UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Higher Polytechnic School of Alcoi

Data Science Methods for Risk Analysis in Natural Disasters

Master's Thesis

Master's Degree in Business Administration

AUTHOR: Tsertsvadze, Verónika

Tutor: Juan Pérez, Ángel Alejandro

Cotutor: Carracedo Garnateo, Patricia

ACADEMIC YEAR: 2023/2024

Abstract

Natural catastrophes pose significant threats to life and property, requiring advanced risk analysis for effective mitigation and preparedness. This work applies statistical techniques to enhance the understanding and prediction of the impact of natural disasters, where the data of study was earthquake-induced losses in Morocco. Specifically, by simulating 13 different earthquake events with varying magnitudes, we employ machine learning models to predict the financial losses each might cause. Additionally, this research proposes a hybrid operational model for seismic risk transfer which combines the cat-in-a-grid and ground motion index methodologies to leverage the advantages of both, enhancing decision-making tools for community resilience. The cat-in-a-grid approach determines payouts under conditions of low to moderate uncertainty, such as extremely large or small magnitudes, while the ground motion index, based on USGS ShakeMaps data, provides additional accuracy for events requiring detailed analysis due to complex geological environments or high variability in exposure. This approach not only yields greater precision in predictions but also offers a strategic means to reduce overall transaction costs, significantly enhancing the efficacy and cost-efficiency of seismic risk mitigation strategies.

KEYWORDS

Data Science, Natural Disasters, Risk Analysis, Predictive Modeling, Disaster Preparedness, Decision Support Systems.

Resumen

Las catástrofes naturales suponen importantes amenazas para la vida y la propiedad, por lo que requieren un análisis de riesgos avanzado para una mitigación y preparación eficaces. Este trabajo aplica técnicas estadísticas para mejorar la comprensión y predicción del impacto de las catástrofes naturales, donde los datos de estudio fueron las pérdidas inducidas por terremotos en Marruecos. En concreto, mediante la simulación de 13 terremotos diferentes con distintas magnitudes, empleamos modelos de aprendizaje automático para predecir las pérdidas económicas que podría causar cada uno de ellos. Además, esta investigación propone un modelo operativo híbrido para la transferencia del riesgo sísmico que combina las metodologías de catástrofe en una red y de índice de movimiento del suelo para aprovechar las ventajas de ambas, mejorando las herramientas de toma de decisiones para la resiliencia de la comunidad. El enfoque cat-in-a-grid determina los pagos en condiciones de incertidumbre baja a moderada, como magnitudes extremadamente grandes o pequeñas, mientras que el índice de movimiento del terreno, basado en los datos de USGS ShakeMaps, proporciona una precisión adicional para los eventos que requieren un análisis detallado debido a entornos geológicos complejos o una alta variabilidad en la exposición. Este enfoque no sólo aporta una mayor precisión en las predicciones, sino que también ofrece un medio estratégico para reducir los costes generales de transacción, mejorando significativamente la eficacia y la rentabilidad de las estrategias de mitigación del riesgo sísmico.

PALABRAS CLAVE

Ciencia de Datos, Catástrofes Naturales, Análisis de Riesgos, Modelos Predictivos, Preparación para Catástrofes, Sistemas de Apoyo a la Toma de Decisiones.

Resum

Les catàstrofes naturals suposen importants amenaces per a la vida i la propietat, per la qual cosa requerixen una anàlisi de riscos avançat per a una mitigació i preparació eficaces. Este treball aplica tècniques estadístiques per a millorar la comprensió i predicció de l'impacte de les catàstrofes naturals, on les dades d'estudi van ser les pèrdues induïdes per terratrèmols al Marroc. En concret, mitjançant la simulació de 13 terratrèmols diferents amb diferents magnituds, emprem models d'aprenentatge automàtic per a predir les pèrdues econòmiques que podria causar cadascun d'ells. A més, esta investigació proposa un model operatiu híbrid per a la transferència del risc sísmic que combina les metodologies de catàstrofe en una xarxa i d'índex de moviment del sòl per a aprofitar els avantatges d'ambdues, millorant les ferramentes de presa de decisions per a la resiliència de la comunitat. L'enfocament cat-in-a-grid determina els pagaments en condicions d'incertesa baixa a moderada, com a magnituds extremadament grans o xicotetes, mentres que l'índex de moviment del terreny, basat en les dades de USGS ShakeMaps, proporciona una precisió addicional per als esdeveniments que requerixen una anàlisi detallada a causa d'entorns geològics complexos o una alta variabilitat en l'exposició. Este enfocament no sols aporta una major precisió en les prediccions, sinó que també oferix un mitjà estratègic per a reduir els costos generals de transacció, millorant significativament l'eficàcia i la rendibilitat de les estratègies de mitigació del risc sísmic.

PARAULES CLAU

Ciència de Dades, Catàstrofes Naturals, Anàlisi de riscs, Models Predictius, Preparació per a Catástrofes, Sistemes de Suport a la Presa de Decisions.

Acknowledgments

First and foremost, I would like to express my deep gratitude to my directors, Angel A. Juan and Patricia Carracedo, whose guidance and support have been fundamental in my personal and professional development. Thank you for giving me the opportunity to be part of the ICSO group and, above all, for placing your trust in me.

I would like to thank my family for teaching me solid values and supporting me in any of the stages of my life, as well as my life partner for his constant support.

Finally, I also thank my ICSO colleagues, from whom I have always learned something valuable.

Contents

List of Figures

List of Tables

1. Introduction

1.1 Motivation

The vulnerability of modern societies to natural disasters has increased significantly, driving the need for efficient and robust risk management strategies. Over the past two decades, according to the Emergency Events Database or EM-DAT (Centre for Research on the Epidemiology of Disasters, 2022), 7,348 natural hazard-related disaster events have been recorded worldwide. These events have had devastating consequences, claiming the lives of approximately 1.23 million people, with an annual average of 60,000 deaths. In addition, these disasters have affected more than 4 billion people, many of them repeatedly, and have generated economic losses estimated at 2.97 trillion dollars, adjusted for 2019 inflation as can be seen in Figure 1, from the report “The human cost of disasters: an overview of the last 20 years (2000-2019)”. This increase in the frequency and severity of natural disasters compares significantly with the records of the previous two decades. Between 1980 and 1999, there were 4,212 disasters that resulted in approximately 1.19 million deaths and affected more than 3 billion people, with total economic losses of about 1.63 trillion dollars. The upward trend is largely attributable to an increase in weather-related disasters, such as meteorological, climatological, and hydrological events.

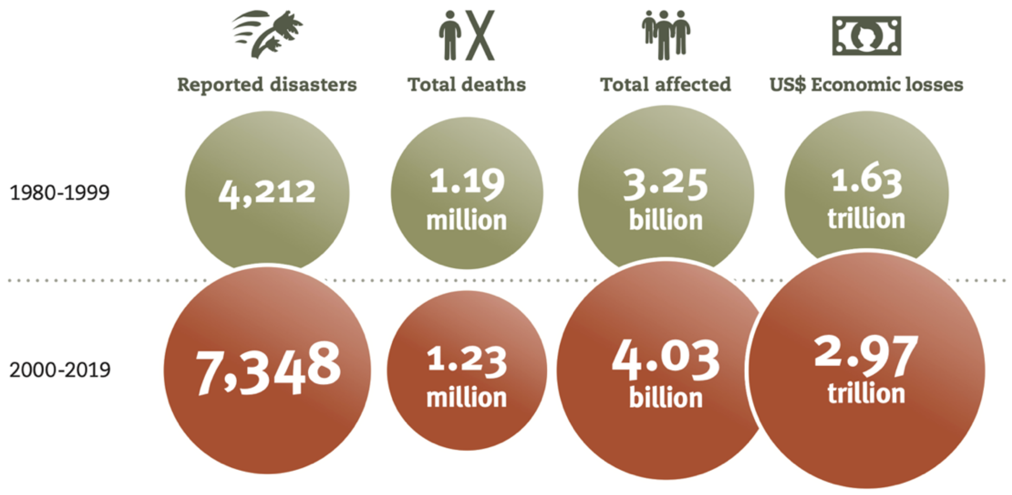


Figure 1. Disaster Impacts.

Earthquakes, together with tsunamis, have positioned themselves as the most lethal catastrophes, accounting for 58% of all recorded deaths. However, unlike other types of disasters, the impacts of earthquakes show great variability. According to EM-

DAT data, there have been years in which earthquakes have caused fewer than 1,000 deaths globally, while in other years, these events have killed more than 100,000 people. In the most recent period, from 2014 to 2019, there have been no earthquakes that have caused more than 10,000 deaths. However, earthquakes in Nepal in 2015, which caused 8,969 deaths, and in Palu, Indonesia, in 2018, with 4,340 deaths, have reminded the world of the destructive potential of these phenomena. Additionally, earthquakes can cause massive damage to infrastructure, as demonstrated by the 2011 earthquake and tsunami in Japan, which resulted in economic losses of approximately 239 billion dollars, the highest figure on record for a catastrophic event.

During the period from 2000 to 2019, 8% of the recorded disasters corresponded to earthquakes, totaling 552 events, and affecting approximately 118 million people. As illustrated in Figure 2, these earthquakes accounted for 21% of the total economic losses caused by natural disasters, amounting to US$636 billion. This figure underscores the significant threat that earthquakes represent at a global level, not only highlighting the direct impact on affected populations, but also emphasizing the profound global economic repercussions, reinforcing the need for comprehensive strategies to mitigate these risks.

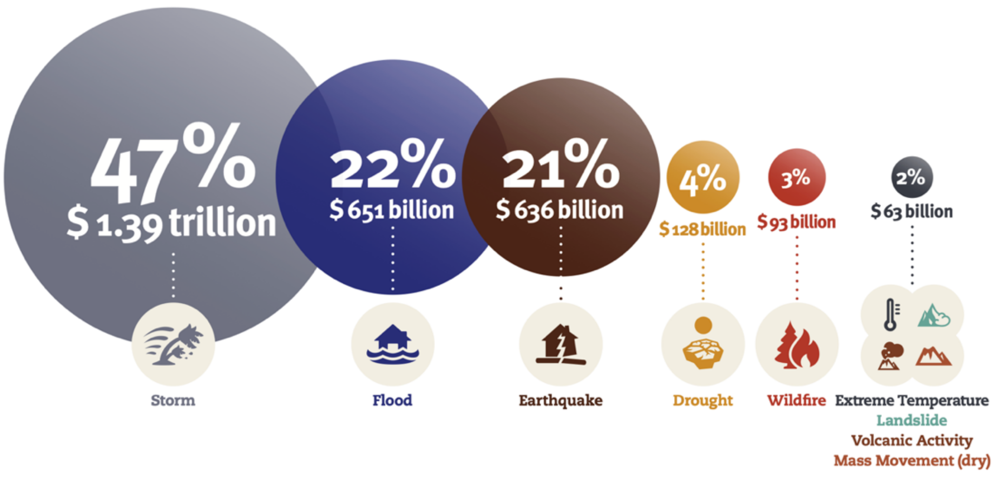


Figure 2. Recorded economic losses per disaster type (2000-2019).

As for today, global economic losses due to disasters reached 250 billion dollars in 2023, equivalent to the total GDP of countries such as New Zealand or Portugal, (Data gaps hide the true human impacts of disasters in 2023). This figure is slightly lower than the previous estimate of 270 billion dollars by 2022. However, these numbers barely outline a portion of the real impact that disasters impose in terms of human lives and economic, developmental, and social effects. The severity of this impact is further underscored by the fact that disasters resulted in 74,000 deaths during the year, significantly above the average for the past five years, with approximately 63,000 of these deaths caused by major earthquakes in Turkey, Syria, Afghanistan, and Morocco. These estimates only partially illustrate the true magnitude of the damage caused by disasters. Many of the impacts of disasters are simply not included in these estimates, such as those associated with slower and smaller-scale development events, and the secondary effects of supply chain issues, which reduce productivity, harm health, and disrupt education in the long term. All these factors contribute to an invisible burden of disasters far greater than the economic estimates provided by insurers.

Given the magnitude of economic losses caused by earthquakes and the devastating impact on affected populations, the critical need for insurers to provide coverage for these catastrophic events is evident. The high cost involved underscores the importance of a robust insurance infrastructure not only for post-disaster economic recovery, but also as an essential risk transfer mechanism.

Since the 1990s, the insurance industry has implemented parametric earthquake solutions to improve the resilience of communities to seismic risk. These products ensure quick and transparent compensation, without the time-consuming processes of traditional claims assessment. Instead, they pay a pre-set amount if the seismic event meets specific physical and measurable characteristics. This introduction builds on previous studies highlighting the effectiveness of such parametric mechanisms for seismic risk transfer, as demonstrated by the research of Guidotti et al. (2024).

In more detail, parametric insurance represents a type of insurance coverage that is triggered upon the fulfillment of pre-established and quantifiable conditions, called parameters, included in the insurance contract. Unlike traditional insurance, which requires an assessment of physical damage in order to proceed to indemnity, parametric insurance is designed around objective variables such as the magnitude of an earthquake, ground acceleration data or the intensity of a hurricane. This type of insurance is particularly useful for rapid disaster response, as it allows for automatic payouts once the agreed-upon parameters are verified, thus facilitating immediate economic recovery without the need for lengthy loss adjustment processes (Desmond, 2023). Furthermore, parametric insurance is characterized by its ability to cover both direct and indirect losses related to catastrophic events. This includes, for example, loss of profits that may happen even though there is no direct physical damage to the insureds’ property. The configuration of these insurances is based on a deep understanding of the clients' needs and on advanced models that predict the probability and impact of specific events.

This type of insurance is especially beneficial for governments that need to deploy resources quickly for disaster response and recovery. By providing immediate liquidity, parametric insurance allows governments to act efficiently in relief and reconstruction operations, thus minimizing the economic and social impact of disasters. In addition, parametric insurance provides an incentive for governments to invest in disaster mitigation and preparedness measures, as premiums can be adjusted according to the level of risk that existing policies and infrastructure are able to mitigate.

To find and guide parametric solutions, this work addresses the integration of a tool such as Machine Learning. Machine Learning (ML) is a branch of artificial intelligence focused on developing algorithms and statistical models that enable computers to perform tasks without explicit programming, explained by Linardos et al. (2022). By leveraging large datasets, these models identify patterns and make predictions or decisions based on new data. In the context of natural disasters, ML has become a relevant tool for guiding parametric solutions. Recent advances in artificial intelligence have proven effective in addressing the severe impacts of disasters, enabling the use of extensive datasets to develop systems that can predict disasters and assist in response and recovery efforts (Singh C. R., 2024). ML applications in disaster management span several phases, from prediction and early detection to damage assessment and post-disaster response. ML models process meteorological and geospatial data to predict the occurrence of hurricanes, floods, or earthquakes with greater accuracy. During and after a disaster, ML analyzes satellite images and sensor data to evaluate impact and coordinate aid efforts more efficiently. Additionally, ML is valuable in creating early warning systems and assessing risks and vulnerabilities, allowing for adequate preparation and timely response. These systems identify patterns in historical data to foresee adverse events and issue alerts in advance. ML enhances the accuracy of predictions and streamlines decision-making processes by analyzing data from several sources, such as satellite images, social media, sensors, and geographic information systems, Sahai et al. (2023). By integrating and processing these data, ML models deliver detailed and precise information that supports informed and effective decision-making. Furthermore, ML enables more accurate damage assessments, facilitating better resource distribution and more effective planning of rescue and reconstruction operations. For instance, ML algorithms can analyze aerial and satellite images to identify affected areas, assess the degree of destruction, and prioritize zones requiring immediate attention. The capability of ML to handle and analyze large volumes of data from multiple sources in real-time enhances the effectiveness of disaster responses. ML systems can process social media data to quickly identify the most impacted areas and efficiently coordinate aid delivery. They can also predict future needs based on historical patterns, allowing authorities to anticipate events and mitigate negative impacts.

Globally, earthquake and catastrophe insurance remain limited. By 2022, only about 42% of economic losses from natural catastrophes were insured, leaving a 58% protection gap (AON, 2023). Moreover, risk is increasing faster than insurance coverage, widening the gap between exposure and protection. The insurance and reinsurance industry are addressing this challenge with a wide range of solutions. Homeowners can purchase earthquake insurance individually or through group mechanisms, which are often mandatory, such as catastrophe funds or group policies. Insurers pool risks and have access to both international reinsurance and new financial instruments, such as catastrophe bonds or derivatives, according to Franco (2014). These tools, which lie at the intersection between traditional insurance and financial markets, also allow governments to access reinsurance through a wider range of capital providers.

Guy Carpenter & Company, founded in 1922, has transformed the reinsurance brokerage industry through its commitment to leveraging comprehensive data analytics. Guy Carpenter has pioneered the use of quantitative analytics to improve risk prediction and management, offering customized solutions adapted to the unique challenges faced by its clients. These services are particularly important in the structuring and placement of reinsurance and in the development of parametric insurance solutions, helping to manage risks associated with natural catastrophes, financial market fluctuations and other global challenges. Today, Guy Carpenter is a prominent leader in the reinsurance industry, known for its strong risk assessment capabilities and strategic reinsurance placement. The company plays a key role in advising governments, especially in catastrophe bond issuance, where it acts as broker of record. Its services encompass risk quantification, global reinsurance structuring and customized development of parametric solutions, enhancing clients' ability to effectively manage severe risks and ensuring sustained growth and resilience in an ever-changing world, (Guy Carpenter, s.f.).

Given the seismic vulnerabilities noted above, this study is based on the development of a parametric insurance solution, in collaboration with Guy Carpenter, specifically adapted to Morocco. Located at the confluence of major tectonic boundaries, Morocco is inherently in a high seismic risk zone. This geographic predisposition, compounded by a history of devastating seismic events, underscores the urgent need to improve the resilience of Moroccan communities and infrastructure to potential earthquakes. Addressing this critical requirement calls for effective risk management strategies that can provide rapid response capabilities and mitigate the economic ramifications of such disasters. In response, Guy Carpenter is leveraging its extensive experience in risk analysis and reinsurance to devise parametric insurance solutions designed to respond to the unique seismic challenges facing Morocco. This initiative is particularly important considering that the seismic landscape of the Mediterranean region is largely determined by interactions between the African and Eurasian plates, along with microplate dynamics, reported by Herman et al. (2015). These interactions give rise to varied and intense seismic activity throughout the area, which has been the cause of several devastating earthquakes and tsunamis over the centuries. Due to this complex tectonic configuration, regions such as Morocco, which are directly influenced by these geodynamic processes, shown in Figure 3 from (United States Geological Survey, n.d.), face a pressing need for effective risk management strategies to mitigate the impact of potential seismic events.

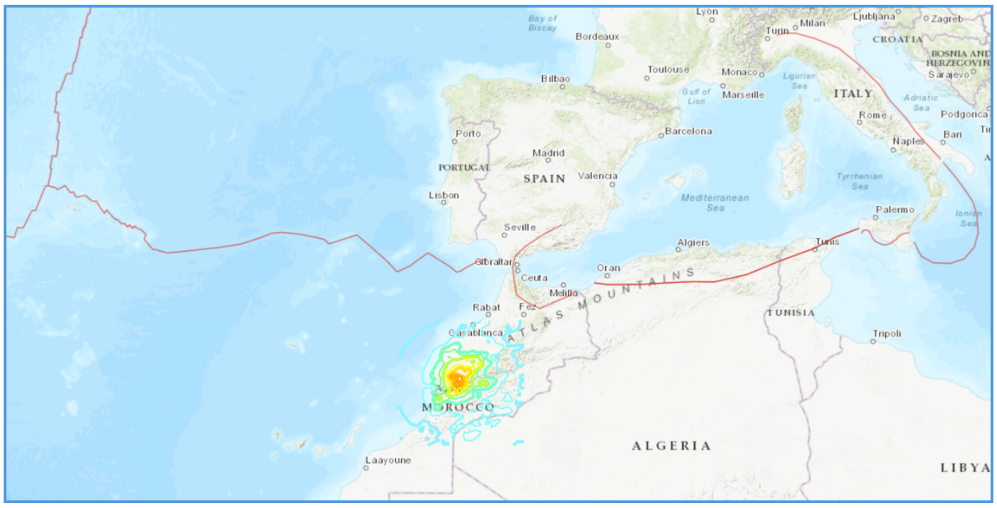


Figure 3. Map of tectonic configuration of Mediterranean region.

The convergence of the African plate with the Eurasian plate at rates between 4 and 10 mm/year has been a continuous process over the last 50 million years, significantly shaping the seismic activity in the region. This process was initially associated with the closure of the Tethys Sea, the remnants of which now form the Mediterranean Sea. High rates of seismicity in Morocco can be attributed to these plate dynamics, including the nearby subduction and transform fault zones. The isoseismic contours over Morocco, also depicted in Figure 3, provide a detailed visualization of the intensity distribution of seismic activity throughout the region. Through this map, the areas' most susceptible to seismic forces are identified based on historical data and predictive models. In other words, it constitutes a tool to better understand the seismic risk profile of Morocco, allowing for more targeted and effective risk management strategies. By integrating this geospatial data, the areas' most likely to be significantly impacted in the event of an earthquake can be better anticipated, thus prioritizing mitigation efforts and resource allocation to improve safety and preparedness.

Having defined the seismic risks facing Morocco, the operational simplified mechanism of risk transfer are then specified, shown in Figure 4. The figure illustrates a comprehensive process for the development and implementation of a parametric insurance solution for Morocco, provided by Guy Carpenter. The process begins with Guy Carpinters' assessment of the seismic risks specific to Morocco. This involves a thorough analysis and quantification of the seismic data, which is used to define the parameters that will trigger the insurance payout. These parameters are carefully chosen based on historical seismic activity and potential impacts to ensure accuracy and relevance. Once the risk parameters are established, Guy Carpenter develops the parametric insurance coverage. This coverage is designed to automatically trigger a payout when the predefined seismic parameters are met, ensuring swift financial response without the need for traditional claims processing. A Special Purpose Vehicle (SPV) is established for the issuance of catastrophe bonds because the risk is significant. This vehicle is independent of the insurance company and protects investors from the insolvency of the insurer. These bonds are a financial instrument to transfer the risk from Morocco to the capital markets. The SPV cedes the risk to the capital markets, which in return provide the necessary funds to cover possible losses. Investors buy these bonds, investing in the risk with the expectation of a return, influenced by the occurrence of the defined seismic event. In the event of a seismic occurrence that meets the defined parameters, the parametric insurance is activated. This activation is based on the data provided by recognized and reliable seismic monitoring systems. The direct release of funds from the SPV to Morocco ensures that financial aid is promptly available to address the damages and initiate recovery processes. Throughout this process, Guy Carpenter acts as the broker of record, overseeing the risk assessment, development of the parametric solution, and the management of the insurance product. They coordinate between the client (Morocco), the SPV, and the capital markets to ensure that all aspects of the insurance solution are properly managed and executed.

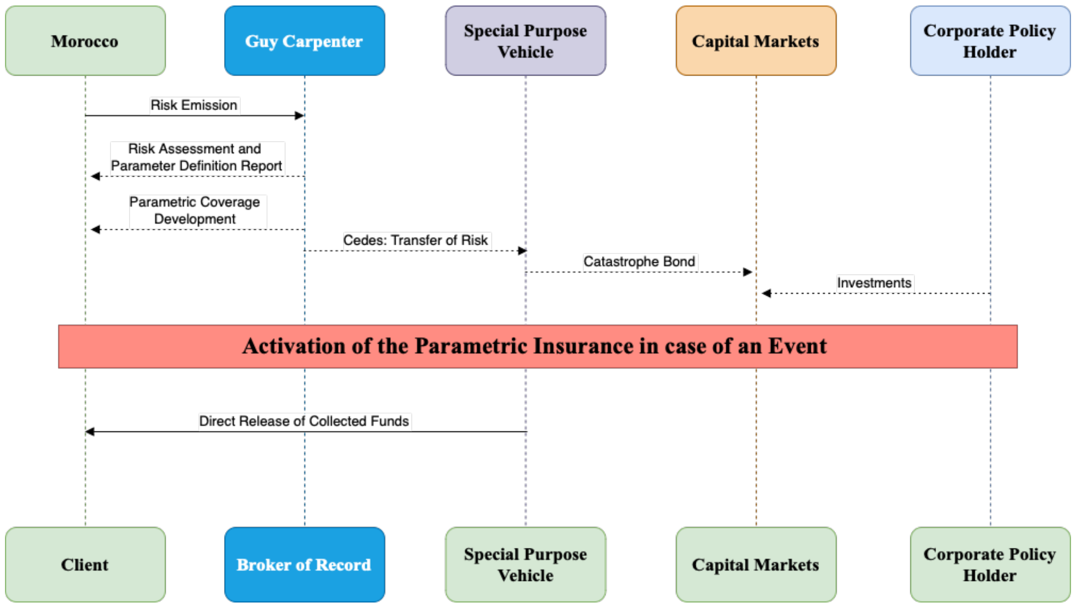


Figure 4. Operational mechanism of risk transfer.

1.2 Objectives

The main objective of this project is the development of statistical and Machine Learning models to predict economic losses derived from simulated earthquakes in Morocco. This process involves the design and implementation of several statistical models and ML algorithms in order to quantify the economic losses associated with simulated seismic events. The development of these models requires the integration of data sets representing seismic and geologic variables, with the objective of building robust models that can identify meaningful patterns and correlations between earthquake characteristics and economic consequences. Different model architectures and regression techniques will be explored to determine which provide the best predictions under diverse conditions.

To ensure the accuracy and applicability of these models, robust statistical methods such as cross-validation or splitting of data into training and test sets will be implemented to assess the ability of the models to generalize to new data. This approach will detect overfitting and assess the reliability of the models in the face of variations in the input data.

Finally, a detailed comparative analysis of the developed models will be carried out to select the most suitable one in terms of predictive accuracy and computational efficiency. This analysis will be based on established performance metrics, such as the mean square error or the coefficient of determination, complemented by an evaluation of the computational complexity of each model. The selection of the optimal model will be oriented not only to technical accuracy but also to the practicality of its implementation in real seismic risk management scenarios.

1.3 Contribution to the Sustainable Development Goals

This section addresses the interaction between the developments presented in this study and the Sustainable Development Goals (SDGs) established by the United Nations in 2015. The SDGs constitute a set of 17 global goals, designed to be a "blueprint for action" towards sustainable development encompassing economic, social, and environmental dimensions (United Nations, 2023). These goals aspire to guide nations towards a future where poverty is eradicated, inequality is reduced, and the environment is protected, fostering peace and prosperity for all.

The research presented in this thesis contributes directly to several of these goals, with a particular focus on improving resilience to natural disasters and strengthening the capacities of communities and cities to respond to emergencies. The proposed innovative solutions, based on data science for natural disaster risk analysis, have the potential to positively impact 5 of the 17 SDGs described in Table 1.

Sustainable Development Goals	**High**	**Medium**	**Low**	**Not applicable**
SDG 1. *No poverty*		**X**		
SDG 2. *Zero hunger*				**X**
SDG 3. *Good health and well-being*				**X**
SDG 4. *Quality education*				**X**
SDG 5. *Gender equality*				**X**
SDG 6. *Clean water and sanitation*				**X**
SDG 7. *Affordable and clean energy*				**X**
SDG 8. *Decent work and economic growth*				**X**
SDG 9. *Industry, innovation, and infrastructure*		**X**		
SDG ***10.*** *Reduced inequalities*				**X**
SDG 11. *Sustainable cities and communities*	**X**			
SDG 12. *Responsible consumption and production*				**X**
SDG 13. *Climate action*		**X**		
SDG 14. *Life below water*				**X**
SDG 15. *Life on land*				**X**
SDG 16. *Peace, justice, and strong institutions*				**X**
SDG 17. *Partnerships for the goals*			**X**	

Table 1. Contribution to the SDG.

The contribution of this work to each of the SDGs mentioned in Table 1 is described below in Figure 5.

1 NO POVERTY

The research develops predictive models that estimate economic losses from natural disasters more accurately. These models can be used by insurers to design parametric insurance products that accelerate compensation payments following a disaster. By receiving funds quickly, vulnerable communities have immediate resources for reconstruction and recovery, thus avoiding prolonged falls into poverty. This rapid response capability is key to post-disaster economic stabilization and to maintaining social cohesion in affected areas.

---

9 INDUSTRY, INNOVATION AND INFRASTRUCTURE

Introducing advanced predictive models into the insurance industry not only drives technological innovation but also improves the viability and efficiency of insurance policies. These models enable insurers to assess risks more accurately and offer products that are better tailored to their customers' needs, which can encourage greater acceptance and use of insurance. Insurance innovation can lead to greater financial stability and operational continuity, especially in critical sectors during and after natural disasters.

---

11 SUSTAINABLE CITIES AND COMMUNITIES

This work promotes the incorporation of scientific data analysis in urban planning and the construction of resilient infrastructure. Through advanced risk assessments, it facilitates the design of urban environments that minimize the impacts of natural disasters. This includes everything from the strategic placement of key structures to the implementation of enhanced building standards, contributing to faster recovery and enhancing the long-term sustainability of communities.

---

13 CLIMATE ACTION

The ability to predict the economic impacts of extreme weather events facilitates the implementation more accurately of more effective risk management and climate change adaptation strategies. While the study focuses more on mitigating economic impacts and less on directly reducing emissions, it strengthens community resilience to climate change by enabling more agile preparedness and response to its consequences, thus supporting global efforts to manage and adapt to changing climatic conditions.

---

17 PARTNERSHIPS FOR THE GOALS

By developing and disseminating knowledge and technologies that facilitate disaster risk assessment, the work promotes collaboration between academia, industry, and policy makers. However, the success and scale of these partnerships depends on external factors, such as political interest and market acceptance of these new technologies.

Figure 5. Detailed analysis of the studys' contribution to the main SDGs.

1.4 Structure of the Project

After the presentation of the motivation for this project, the description of the objectives and the contribution to the SDGs, the rest of the project is structured as follows.

Section 2 describes the literature review on loss prediction models for natural disasters, with a particular focus on earthquakes. This section establishes the theoretical framework and methodological basis necessary for the understanding and application of the modeling techniques used in this research.

In Section 3, the databases and variables used for the development of the predictive models are described in detail. This part is key to understanding the nature and origin of the data that support the analysis performed.

Section 4 outlines the methodology applied for the construction and validation of the predictive models. It details the statistical techniques and modeling approaches used to ensure that the models are robust and applicable in seismic risk management contexts.

Section 5 is focused on the analysis and comparison of the results obtained from the different models implemented to select the optimal one, based on several metrics.

Lastly, Section 6 highlights the findings of this study and outlines potential directions for future research.

2. Theoretical Framework

2.1 Parametric Insurance and Solutions for Seismic Risks

In the context of parametric insurance for seismic risks, the main problem lies in the basis risk, that is, the difference between the actual losses and the calculated compensations based on a predetermined index. The study by Gu et al. (2023) addresses this problem by proposing, mainly, a quantile regression forest method, which considers exposure in terms of human losses, magnitude, population density and GDP, among others, as latent variables, influenced by the geographical location of the epicenter. It is argued in their study that this approach is particularly useful for capturing risks associated with the extremes of the distribution, which is an advantage in the context of natural disasters. On the other hand, Pucciano et al. (2017) opted for the use of strong motion sensor data to estimate losses. The methods used in their estimation vary from simple approaches, which apply third-order polynomials to calculate losses from the closest ground motion measurements or from all available stations to more advanced methods that include detailed estimates of ground motion at asset sites and use damage functions and ground motion prediction equations. Research shows that while simple methods may be sufficient in areas with high sensor density, more sophisticated approaches are needed in areas where sensors are more dispersed or farther away from secured assets.

There is no extensive research literature on the reduction of losses caused by seismic events by means of statistical or Machine Learning models. Different research addresses the accuracy of loss estimation through the calibration of triggering mechanisms for parametric catastrophe bonds (CAT bonds), such as the study by Pai et al. (2022) on obtaining the estimation of the trigger parameters using the Bayesian quintile regression model. These financial instruments serve to transfer seismic risk to the capital market, using triggers based on the physical characteristics of the earthquake, such as magnitude and location. Goda (2014) explains in his study the calibration of these mechanisms through the use of logistic regression and direct ground motion observations, whereby the basis risk, specifically the trigger risk, can be significantly reduced. This improvement allows for more flexible adoption of various payoff structures for CAT bonds, facilitating financial risk management and improving resilience to seismic disasters.

This review of strategies for addressing basis risk in parametric insurance for seismic risks underscores an approach predominantly focused on improving trigger mechanisms and not directly on loss prediction. This orientation highlights a critical need to innovate beyond simple trigger calibration and move toward a more holistic model that also includes effective loss prediction and mitigation.

2.2 Estimation of Seismic Losses through Traditional Approaches

The estimation of earthquake losses within the context of parametric insurance is addressed, on the other hand, through models that integrate a wide range of macroeconomic data together with probabilistic methodologies. These models are designed to assess and manage the risks associated with catastrophic events, focusing on estimating the probability of occurrence of these events and their potential impacts. Unlike the approaches described above, which seek to predict exact post-disaster losses to guide the parametric insurance solution, parametric insurance models themselves aim to establish trigger thresholds based on predefined parameters, allowing for a rapid and effective response when these conditions are met, according to the study by Salgado-Gálvez et al. (2023).

Erdik (2017) highlights in their research that in the analysis of risks associated with earthquakes, most assessment schemes are based on the quantification of seismic shaking, using ground motion intensity parameters as part of probabilistic or seismic hazard models. These models determine direct physical damage through fragility or vulnerability relationships, which calculate the probability of damage or loss as a function of the observed intensity level. These relationships not only estimate direct losses as repair or reconstruction costs, but also consider the loss ratio, i.e., the cost of repair in relation to the replacement value of the affected asset.

Expanding upon these methodologies, the HAZUS software, as detailed in the study by Kircher et al. (2006), offers a comprehensive approach for modeling and assessing earthquake impacts. HAZUS uses geospatial data to create detailed scenarios that predict physical and economic losses from seismic events. The software employs sophisticated engineering models within its Advanced Engineering Building Module (AEBM) to assess damage based on varying seismic intensities, providing insights into structural vulnerabilities and potential repair costs. This detailed loss estimation facilitates more accurate predictions, aligning closely with the needs of parametric insurance models to establish effective and responsive strategies for disaster risk management. The integration of HAZUS in parametric insurance planning allows insurers and policymakers to fine-tune their thresholds and ensure that payouts are triggered precisely, thereby optimizing disaster response and recovery efforts.

Such developments underscore the importance of continued innovation and integration of technology in the insurance industry, aiming to mitigate the economic and social impacts of natural disasters on vulnerable communities.

3. Data Description

In this section, a detailed description of the datasets, supplied by the company, to be used in the study is given. The composition of each dataset is explained, identifying, and describing the specific variables they contain. The datasets have been derived from the simulation of 13 seismic events over 10,000 years of seismicity in Moroccan territory and consists of three main files in a .csv format: the Event Loss Table file, The Exposure by Station file, and the Intensity Measure by Event by Station file.

1. **Event Loss Table**

The Event Loss Table file is generated by running a catastrophe model for the region of interest, using a reference exposure and contains 26,867 observations. This file provides data on the set of stochastic events used in the catastrophe model, including details such as the location and intensity of each event, and the economic losses they cause to the exposure under analysis. Each row of Event Loss Table file corresponds to a simulated event and has 7 columns detailed in Table 2.

Feature	Description	Units	Type
Event ID	Event identifier	N/A	Numerical
Magnitude	Earthquake strength	Moment Magnitude	Numerical
Loss	Monetary losses	USD	Numerical
Centroid Depth	Mean depth of the dislocation earth crust area during an earthquake	Kilometers	Numerical
Longitude	Longitude of epicenter	Decimal Degrees	Numerical
Latitude	Latitude of epicenter	Decimal Degrees	Numerical
Rate	Annual rate of occurrence	N/A	Numerical

Table 2. Event Loss Table dataset features.

2. Exposure by Station

The Exposure by Station file is created by discretizing the exposure into an ordered grid of cells. The exposure is distributed on a regular grid of cells, each with a defined resolution, this is the size of each cell can be 0.05, 0.1, 0.25 or 0.5 decimal degrees (dd). Each grid size corresponds to a different file. The 0.05 decimal degree file contains 22,146 observations, indicating high resolution and detail. In contrast, the 0.5 decimal degree file includes only 327 observations, suitable for less detailed analysis. Intermediate sizes, such as the 0.1 decimal degree file with 6,073 observations and the 0.25 decimal degree file with 1,100 observations, offer moderate resolution suitable for different types of spatial analysis.

The exposure corresponding to each cell is concentrated at the centroid of the cell, known as a station. The numerical value of the exposure at each station is transformed into a weight by a scaling process, so that the sum of the weights of all stations is equal to 1. Each row of the Exposure by Station file contains five columns, which are detailed in Table 3.

Feature	Description	Units	Type
Station ID	Station identifier	N/A	Numerical
Exposure	Exposure by station, buildings which are exposed to the damage.	USD	Numerical
Weight	To each station is assigned a weight, proportional to the exposure associated to the station	N/A	Numerical
Longitude	Longitude of station	Decimal Degrees	Numerical
Latitude	Latitude of epicenter	Decimal Degrees	Numerical

Table 3. Exposure by Station dataset features.

3. Intensity Measure by Event by Station file

The Intensity Measure by Event and by Station file is generated by assigning the intensity measure corresponding to each simulated event to the stations within its area of influence. To perform this assignment, a linear sparse interpolation algorithm that does not perform extrapolations is used, allowing to associate to each station within the area of influence of a simulated event a specific Intensity Measure. Each row of this file contains three columns, and the specific details of each one are presented in Table 4.

Feature	Description	Units	Type
Event ID	Event identifier	N/A	Numerical
Station ID	Station identifier	N/A	Numerical
Intensity Measure	Measurement of ground shaking	Pseudo Spectral Acceleration at 0.3s [%g][1]	Numerical

Table 4. Intensity Measure by Event by Station dataset features.

In order to align the data set with the objectives of the study and due to the existence of overlapping data, the three files have been merged as explained below for each size of grid.

To provide a coherent synthesis of these data, a matrix is constructed where the index is composed of the event ID and the columns represent the individual stations. The values within the matrix are the intensity measurements for each event at each station. Considering that each station has an assigned weight, the next step is the application of these weights to create the WIM (Weighted Intensity Measure) parameter. This weighted measure aggregates the intensity measurement information at each station, reflecting the proportional contribution of each station based on its exposure to risk. By applying the weights to the recorded intensity measures, we obtain WIM, a value that effectively represents the overall earthquake intensity over the analyzed region, considering the variations in the exposure of each station. This ensures that stations with higher exposure have a corresponding influence on the overall assessment of earthquake impact. In the development of the final database for this study, apart from calculating the WIM, higher powers of the WIM are also generated, specifically WIM squared, and WIM cubed. These

[1] Spectral acceleration is a unit measured in g (the acceleration due to Earth's gravity, equivalent to g-force) that describes the maximum acceleration in an earthquake on an object – specifically a damped (USGS).

transformations are performed with the objective of capturing nonlinear effects of earthquake intensity on economic losses. The inclusion of WIM squared, and WIM cubed allows the model to consider how increases in earthquake intensity can have disproportionately larger impacts on losses, thus facilitating a more accurate and nuanced understanding of the relationships between earthquake intensity and economic consequences. In addition, it is important to note that each ground shaking record is associated with estimated economic losses, which are key to the analysis in this study. Therefore, the losses associated with each seismic event are also included in the final dataset. Two other variables, Centroid Depth and Magnitude, are also incorporated in the dataset. The resulting datasets are shown in Table 5.

Grid Size (dd)	Dataset Size	Features
0.05	24,648 rows and 6 columns	Loss, Centroid Depth, Loss, Magnitude, WIM, WIM^2 and WIM^3
0.1	24,652 rows and 6 columns	
0.25	24,115 rows and 6 columns	
0.5	22,061 rows and 6 columns	

Table 5. Resulting datasets.

4. Solving Methodology

4.1 Parametric Problem Formulation

The parametric problem formulation in the field of seismic risk analysis focuses on the creation of predictive models that automatically trigger financial compensation based on objective and measurable criteria. This approach is key in the field of parametric insurance, which is distinguished by its ability to provide rapid and transparent responses following seismic events, without the need for traditional damage assessment processes (Marsh, s.f.).

The main objective of a parametric model is to establish a direct and clear relationship between the physical characteristics of a seismic event, such as magnitude and location, and the estimated economic losses. This allows insurance payments to be made quickly, significantly reducing the time and complexity associated with damage assessment and claims processing. In post-disaster situations, where access to rapid resources is important for recovery and reconstruction, the efficiency of these models offers support to affected communities.

These parametric models are generally based on the results of an existing catastrophe model. Catastrophe models, in their broadest form, use multiple layers or components to estimate asset losses in each region. In the context of seismic risk models, the hazard component is usually composed of a large number of simulated stochastic events, for which both primary parameters (hypocenter locations and magnitude) and simulated ground motion values at exposed locations are available. These values are usually derived from a set of ground motion prediction equations integrated into the model. A vulnerability component probabilistically assesses the expected damage at a given exposure based on the simulated measured intensity values, and a financial module converts the physical damage into economic loss, Guidotti et al. (2024).

Ultimately, the objective of a parametric solution based on the cat-in-a-grid model is to determine an optimal relationship between the magnitude and location of an earthquake and the losses it causes. On the other hand, the objective of an index-based parametric solution is to establish an optimal relationship between the ground motion at a location and the losses it generates. This comprehensive approach allows insurers to offer products that can be quickly activated, providing affected communities with effective and rapid means to initiate their recovery, as mentioned in previous sections.

4.1.1 Cat-in-a-grid Parametric Model

The parametric Cat-in-a-grid solution begins with the division of the region of interest into an organized structure of hexahedra. This process, known as discretization, is used to ensure that each segment of the studied area is properly monitored and evaluated in terms of seismic risk. The underlying catastrophe model provides a solid basis for effectively assigning both risk level and response rate to each hexahedron in the grid. The key component in the Cat-in-a-grid design is the determination of specific magnitude thresholds for each hexahedron. These thresholds are critical because they establish the conditions under which an earthquake will be considered to have reached a significant level of hazard. The selection of these thresholds involves careful analysis of historical earthquake data, geologic characteristics of the region, and structural damage potential.

The optimization problem in the Cat-in-a-grid model seeks to maximize the risk transferred without exceeding a predetermined activation rate, which is directly linked to the insurance policy premium. This objective is mathematically formulated as:

$$\max Z = \sum_i \sum_j Z_j \;\; s.t. \;\; \sum_i \sum_j R_j \leq \hat{R} \tag{1}$$

Here Z_j represents the risk transferred for each stochastic event j in the hexahedron i, and R_j is the rate associated with that risk or the activation rate, or the probability that event j in hexahedron i activates the policy. This value is directly associated with the frequency and magnitude of the stochastic event and its relation to the thresholds established to activate the payments. $\hat{R}$ is the maximum acceptable trigger rate, linked to the policy premium. This means that the cost of the policy can cover risks up to this limit without the insurance company incurring losses. This approach effectively balances risk coverage with the financial sustainability of the policy.

Once the thresholds are established and the insurance policy is in place, the model allows for a rapid post-event assessment to determine whether a payout should be made. This is done by analyzing the primary parameters of the seismic event, such as magnitude and hypocenter location, reported in near real-time by seismic monitoring entities such as the USGS. The ability to respond quickly not only improves the efficiency of disaster management but also reinforces policyholder confidence in the robustness of the insurance system.

4.2.2 Ground Motion Index Parametric Model

In the parametric formulation based on ground motion rates, the design problem is centered on minimizing an error function g, which evaluates the discrepancies between the losses triggered by the parametric model (Parametric Losses, L^P) and the losses predicted by the model (Modeled Losses, L^M). This approach is mathematically stated as an optimization problem:

$$\min \sum_e g\,(L_e^P - L_e^M) \qquad \textbf{(2)}$$

Where index e denotes a stochastic event in the catastrophe model that encompasses all possible events within the stochastic catalog, with L_e^P and L_e^M representing the losses resulting from the parametric model and the stochastic model associated with that stochastic event e, respectively.

In its general form, the parametric loss L_e^P is expressed as a function f of the *N* considered parameters of the event, of the form:

$$L_e^P = f(x_{1,e}, \dots , x_{N,e}) \qquad \textbf{(3)}$$

In the context of catastrophe modeling and parametric solutions in insurance, a stochastic event refers to a possible but uncertain event, whose occurrence and characteristics are modeled from probabilistic distributions based on historical data and statistical analysis. These events are not predicted at a specific time and place but are considered as part of a broader set of possible scenarios that could occur at any time within the model simulations. Stochastic events are fundamental to understanding and preparing insurance risk and financial models, as they allow insurers and risk planners to assess the potential impacts and costs of natural disasters or claims within a probability framework and to make informed policy and coverage decisions.

The design problem for the index-based solution involves determining and calibrating the optimal function f that minimizes the presented objective function. The methodology can be broken down into three steps:

1. Exposure Discretization

Exposure discretization involves dividing the study region into a series of uniformly distributed virtual points or stations. This process is key because it allows for a detailed and accurate representation of the terrain and its seismic characteristics. Each of these virtual stations becomes a node where specific ground motion data are collected and analyzed. For each station, a weight a_s is assigned, which is proportional to the exposure associated with that station. The exposure refers to the population density, critical infrastructure, or economic value of the area covered by the virtual station. The sum of all weights assigned to stations should add up to one ($\sum_s a_s = 1$) ensuring that the distribution of weights is equitable and representative of the total risk across the region.

Figure 6 represents an example of the weight assignment. In this example:

- The station near Agadir has a weight $\boldsymbol{\alpha} = \mathbf{0.027}$
- The station around Fez has a weight of $\boldsymbol{\alpha} = \mathbf{0.042}$
- The station near Casablanca has a weight of $\boldsymbol{\alpha} = \mathbf{0.169}$

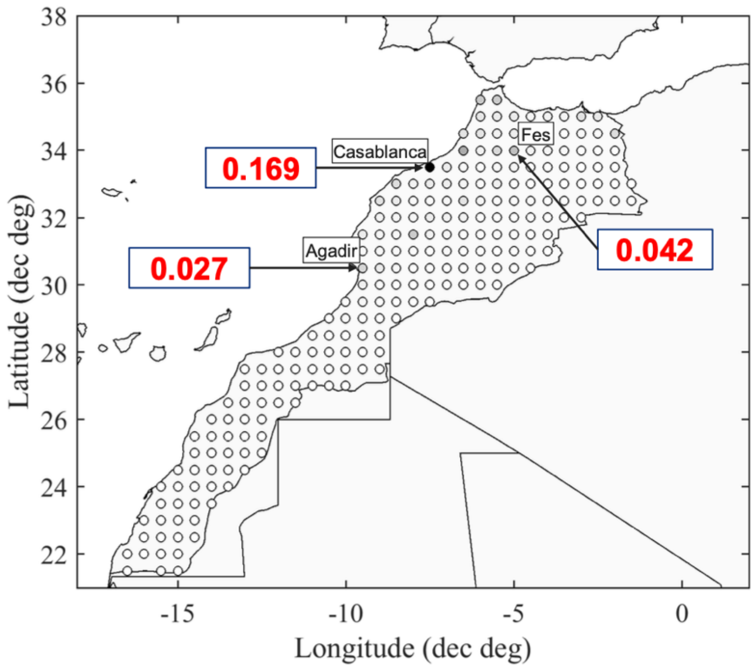


Figure 6. Weights assigned by station.

2. Integration of Catastrophe Model Results

This step directly links seismic hazard parameters to anticipated economic losses using a catastrophe model that simulates thousands of years of potential seismic activity.

In this context, each stochastic event generated by the model is analyzed to determine not only the magnitude of the event but also its ground motion intensity fingerprint, such as Peak Ground Acceleration, Peak Ground Velocity, and Pseudo-Spectral Acceleration. For each stochastic event, estimated losses and intensity measure (IM) values are extracted and recorded at each virtual station. This process involves evaluating how each event would specifically affect the locations represented by the virtual stations, considering the geological and constructive characteristics of the area.

This step uses a catastrophe model to simulate 10,000 years of seismic activity in the Morocco region. The model provides representative data on possible seismic events and their impacts, essential for analyzing and preparing risk mitigation strategies. For each stochastic event, IM footprints of interest are extracted. These footprints indicate how the intensity of ground motion varies across the region for each simulated seismic event. Each virtual station in the discretized network receives information on the MI value and estimated economic losses associated with its specific location. This includes: (1) Each station is assigned the MI value observed at its location during each stochastic event, reflecting the level of seismic risk at the location and (2) the economic losses that each event would cause in the Morocco exposure are calculated and assigned to the corresponding stations.

In the example of three virtual stations located in Casablanca, Agadir, and Fes, represented in Figure 7:

- The station near Agadir records the Intensity Measurement for Stochastic Event 1 only.
- The station near Fes records the Intensity Measurement for Stochastic Event 2 only.
- The station near Casablanca records the Intensity Measurement for both stochastic events.

This example shows how different stations can record different exposures to specific events, highlighting the need for a network that effectively captures variability in seismic intensity and its economic impacts.

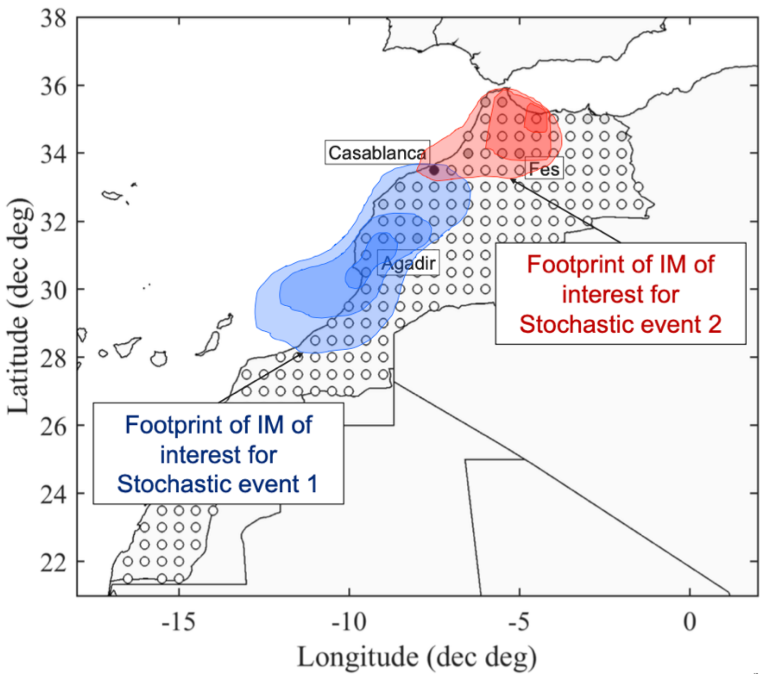

Casablanca
Fes
Agadir
Footprint of IM of interest for Stochastic event 2
Footprint of IM of interest for Stochastic event 1
Latitude (dec deg)
Longitude (dec deg)
38
36
34
32
30
28
26
24
22
-15
-10
-5
0

Figure 7. IM footprints.

3. Index Optimization

The third step in the parametric design based on ground motion indices, it involves the calibration of a mathematical function that models the relationship between the IM and the economic losses at each virtual station. This optimization process seeks to adjust this relationship so that the calculated losses match as closely as possible the actual or modeled losses during seismic events.

The function relating MI to losses, denoted here as f is typically defined as a polynomial combination of the form:

$$f(IM_s) = a \cdot IM_s^3 + b \cdot IM_s^2 + c \cdot IM_s \qquad \textbf{(4)}$$

In the equation, IM_s represents the intensity of ground motion measured at station s, and a, b, c are model parameters that need to be determined through optimization.

Optimization is performed by adjusting parameters a, b, c so that the calculated loss rate L_s or $f(IM_s)$ for each virtual station reflects as closely as possible the losses estimated by the catastrophe model for relevant events. Mathematically, we seek to minimize the weighted sum of the differences between the calculated parametric losses and the modeled losses.

Figure 8 illustrates the correlation between the Ground Intensity Measure, specifically the Pseudo-Spectral Acceleration at 0.3 seconds (PSA 0.3s), and the estimated economic losses for the station located in proximity to Fes. This plot is fundamental to understand how variations in seismic intensity economically affect the specific region. Through a loss function $L_s = f(IM_s)$, it is observed how increases in PSA translate into an exponential increase in losses, underscoring the importance of incorporating accurate ground motion intensity measurements into seismic risk and disaster response planning. This visualization provides a clear basis for the calibration of the loss model used in the study, highlighting the direct relationship between the physical parameters of the earthquake and its economic consequences.

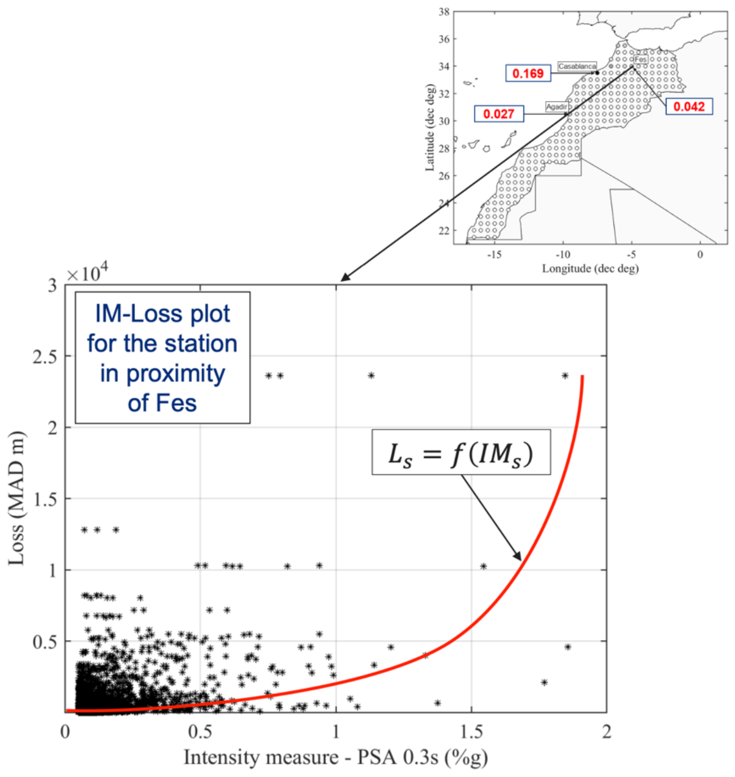


Figure 8. Intensity-Loss ratio at Fes.

4.2 Parametric Solution Based on Hybrid Model

Effective seismic risk management in regions with high tectonic activity, such as Morocco, requires innovative approaches that not only improve community resilience but also optimize the cost and efficiency of post-disaster response. As mentioned above, traditionally, parametric solutions have focused on models that use either direct physical characteristics of earthquakes, such as location and magnitude, or ground motion-based indices. Each of these approaches has its advantages and limitations, which may influence their applicability depending on the specific nature of each seismic event and the needs of the stakeholders.

In this context, a hybrid parametric solution that integrates Cat-in-a-Grid and Ground Motion Indices models has been developed to comprehensively address seismic risk in Morocco. This hybrid solution seeks to combine the strengths of both approaches: the simplicity and speed of the Cat-in-a-Grid model and the accuracy and detail of the Ground Motion Indices-based model. This dual approach not only increases the accuracy of loss estimation, but also allows for a more flexible and tailored response to the complexities of seismic events, from large magnitude earthquakes to those that, without causing major shaking, generate significant damage due to their location or specific geological characteristics.

This section details the application of this hybrid model in Morocco, describing the methodologies used to integrate and calibrate both models into a unified system that improves the capacity for forecasting and management of natural disasters. The specific predictive models developed to estimate losses based on this hybrid approach will then be presented, highlighting how this integration not only reflects an advance in technical accuracy, but also in the practical implementation of insurance and seismic risk mitigation solutions.

4.3.1 Quadratic Ordinary Least Squares Regression Model

The implementation of a quadratic model in seismic risk management in Morocco is part of a broader approach to develop parametric solutions that can respond more accurately and efficiently to the complex dynamics of natural disasters. This statistical model is designed to capture the nonlinear relationship between WIM and the resulting economic losses, allowing a more refined loss estimation than that provided by linear models, Chatterjee and Olkin (2006). The main reason behind running this quadratic model is to address the inadequacy of linear models in capturing the exponential increase in losses as the intensity of ground motion intensifies. In large magnitude events, it is critical to have a model that adequately reflects how small increases in intensity can result in disproportionately large increases in damage and losses, a phenomenon that a linear model may significantly underestimate.

The adjustment of the model has been made using Ordinary Least Squares (OLS) regression. This method works by minimizing the sum of squares of the differences between the observed values and those predicted by the linear model. In other words, it attempts to reduce the squared error between the prediction and the actual data. The model uses as independent variables the WIM and its squared term (WIM^2). This model seeks to capture the nonlinear relationship between the intensity of ground movement and the resulting economic losses, a relationship that simple linear models cannot adequately represent.

The quadratic model is defined as follows:

$$Loss = a \cdot WIM + b \cdot WIM^2 \tag{5}$$

where a, b are coefficients that OLS estimates.

Given the specific behavior of quadratic regression, whose function adopts a U-shape, the OLS model tends to fit the point density close to zero efficiently. This is because most seismic events record relatively low magnitudes and economic losses, which is a recurrent phenomenon in historical records. Events that generate significant losses are, in comparison, much less frequent. The U-shape of the quadratic model assumes that, to encompass both low and high loss magnitudes, the OLS-estimated curve must descend toward negative values before ascending to fit the higher values. It is this dip in the curve that produces negative predicted loss values, which are physically and conceptually unacceptable in this context. To correct this limitation of the OLS model and avoid the generation of negative predicted loss values, a variant called non-negative least squares quadratic regression (NNLS) is used. NNLS modifies the traditional OLS algorithm to impose the additional constraint that all predictions must be non-negative. This means that during the optimization process, the NNLS algorithm adjusts the loss function surface to ensure that all estimated values and their corresponding coefficients remain within realistic, i.e., non-negative, bounds, as explained by Slawski and Hein (2013). The implementation of NNLS in this study addresses the skewness of data concentrated near the origin, avoiding the generation of negative estimates that are conceptually unacceptable. This constraint ensures that all loss estimates remain within a plausible and physically consistent range.

4.3.2 Cubic Ordinary Least Squares Regression Model

The cubic model incorporates a third polynomial term, the WIM^3, which allows for a more complete mathematical representation of how economic losses relate to the WIM. By including this cubic term, the model can capture possible asymmetry in the response of losses to variations in intensity, which might not be evident with linear or quadratic models. Once again, the model is fitted with OLS. By expanding the model to include the cubic term, it seeks to address the shortcomings of lower degree models in representing more complex behavior, such as rapid loss escalation beyond certain points of ground motion intensity.

The structure of the cubic model is described by the equation:

$$Loss = a \cdot WIM + b \cdot WIM^2 + c \cdot WIM^3 \tag{6}$$

where the coefficients a, b, c represent the incremental influence of each increase in WIM power on losses. This model offers several technical advantages: (1) As the WIM increases, especially at high values typical of severe seismic events, the cubic term allows modeling an exponential increase in losses, which is more aligned with actual observations where small variations in intensity can have large impacts on damage and losses. (2) The inclusion of a third-degree term facilitates the detection of points where the relationship between intensity and losses changes significantly. These points, which may indicate critical damage thresholds, are important for disaster mitigation planning. (3) By modeling curvature and changes in the rate of loss increase more effectively, the cubic model can reduce the model residual and improve the overall fit, resulting in more accurate and reliable predictions.

As in the case of quadratic regression, when fitting the cubic regression, negative predicted values are generated. Therefore, the NNLS model is also applied to restrict such behavior.

4.3.3 Piecewise Model

Given the nonlinear relationship between losses and intensity measure, depicted in the example in Figure 8, the piecewise model has been carried out to better capture this complexity. The piecewise model is a technique that allows fitting different sections of the data with different polynomial functions. This offers greater flexibility to capture the nonlinear relationships between the explanatory variables and the dependent variable.

The technique of regression by sections has been studied in several fields such as computational geometry, statistics, and machine learning, according to Lokshtanov, Suri and Jie (2021). In statistics and machine learning, slice approximation refers to the technique of dividing the data set into several sections (or slices) and fitting a simple polynomial function to each section of the data. Unlike other approaches where the fitted functions must be continuously connected over the entire range of data, in the slice approximation each segment can be fitted independently. This means that the functions in each span do not necessarily have to coincide at their junction points. The main idea behind the slice approximation is to facilitate the capture of local behaviors of the data that may vary significantly from one region to another. This is particularly useful in situations where the relationship between variables does not follow a simple global pattern and can change dramatically over different ranges of the data set.

The motivation for using a piecewise model lies in the need to address the nonlinearities evident in the data. In the context of earthquakes and economic losses, the relationship between earthquake intensity measures and losses is not linear across the range of data. Losses may increase disproportionately with magnitude and other measures of intensity in certain ranges. For example, small increases in intensity could have minimal impact on economic losses at low intensities but cause dramatic increases at higher intensities. The piecewise model allows these variations to be captured by fitting different sections of the model to different data behaviors.

For this analysis, both quadratic and cubic piecewise regression models have been applied. For both models, limits have been implemented for clamping or restrict the predicted losses to observed minimum and maximum values, preventing the model from producing unrealistic predictions outside the observed range.

4.3.4 Random Forest Regression Model

In this study, the Random Forest (RF) regression model was also implemented because of its robustness and ability to model complex nonlinear relationships between variables, as described by Borup et al. (2023), such as those between WIM and economic losses resulting from earthquakes. This algorithm, from the supervised machine learning[2]

[2] Supervised learning is an approach in the field of machine learning in which a model learns from a set of labeled data, where the correct answers are provided during training. This method allows the model to generate predictions based on the association between input features and known outputs, as defined by Singh et al. (2016).

branch, builds multiple decision trees from randomly generated subsets of data and combines them to improve the accuracy and stability of the predictions, as shown in Figure 9, from Hunter (2022). Each tree is generated through a bootstrapping process and randomly selects features at its split points, which enriches the diversity among trees and minimizes their correlation. At each node of the tree, a subset of features (e.g., earthquake magnitude, depth, intensity measured at different stations, etc.) is selected and the best split at that node is determined based on a criterion such as maximum variance reduction. This approach is beneficial because it reduces the risk of overfitting (individual trees may have high variance, but the average of many trees is more stable and accurate) and is quite robust to outliers and noise in the data. When talking about overfitting, it means that the model fits the training data too closely to the point of capturing the noise or random fluctuations present in that data rather than the underlying trend or true pattern. This means that the model performs very well on the training data, but its ability to generalize to new data or data not seen during training is poor.

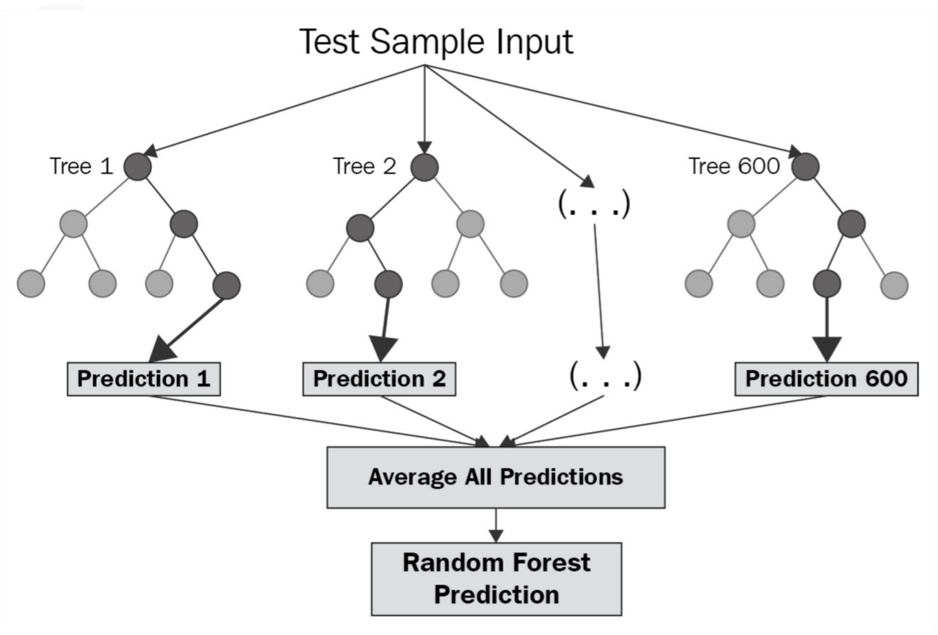


Figure 9. Structure of Random Forest Model.

The accuracy in prediction using multiple decision trees in models such as RF comes from the diversity in the responses of individual trees. This is due to the complexity and subtle characteristics of the data, which may not be captured by a single tree. However, by combining the predictions of all the trees in the model, a final estimate is achieved that is not only more accurate, but also more stable than that provided by any single tree.

Unlike the linear and quadratic regression models used previously, which clarify the direct relationship between the predictor variables and the target variable, RF is considered a “black box” model. This type of model, while providing estimates of the relative importance of each variable, does not provide a transparent explanation of how exactly the inputs are transformed into the final output. Despite this lack of transparency, RF can capture complex patterns in the data, especially in situations where responses to ground motion intensities are highly nonlinear and varied. In any case, in later sections, the ability of this model to predict losses following different events will be validated or determined. The use of RF in seismic risk assessment seeks a balance between the need for accuracy in modeling and the limitations in model interpretability.

4.3.5 XGBoost Regression Model

The XGBoost model is another approach used in this study to predict economic losses resulting from earthquakes. The XGBoost model, short for eXtreme Gradient Boosting, is used in machine learning to make accurate predictions in both regression and classification problems. Its operation, described by Friedman (2001), is based on the sequential construction of decision trees, where each tree tries to correct the errors of the previous one. This process is carried out by means of gradient boosting, a technique that minimizes a specific loss function in each iteration. Initially, a simple decision tree is constructed that performs basic predictions. The errors in this tree are calculated and used to adjust the next tree in the sequence. This process is repeated, with each new tree attempting to correct the accumulated errors of the previous trees. The sum of the individual models generates a final model that is more robust and accurate.

XGBoost employs regularization techniques, which penalize model complexity and help prevent overfitting. This ensures that the model not only fits the training data well, but it also performs well on new and unseen data. Furthermore, XGBoost efficiently handles missing values and performs parallel computations, which speeds up the training process and allows handling large volumes of data (Nguyen, 2023).

4.3.6 Hyperparameter tunning method: Grid Search Cross Validation

Grid Search Cross-Validation is a technique used for hyperparameter optimization in black box predictive models. This systematic and exhaustive method seeks to identify the optimal combination of hyperparameters that maximizes model performance. The technique involves defining a set of possible values for each hyperparameter and evaluating all possible combinations by cross-validation, as defined by Huang, Mao, and Liu (2012) . This approach is broken down into several steps, each important to ensure a complete and efficient exploration of the hyperparameter space.

The process begins with the definition of the hyperparameter space. The hyperparameters to be optimized are selected and the possible values they can take are specified (Budiman, 2019). For each hyperparameter, a range of discrete values to be explored is established, which creates a “grid” of all possible combinations of these values. Once the hyperparameter space is defined, all possible combinations of these values are generated. Each combination represents a unique configuration of the model. For example, if two hyperparameters with three possible values each are being optimized, there will be a total of nine combinations to evaluate (3x3=9). This approach ensures that all possible configurations of the model are explored, providing an exhaustive search for the best set of hyperparameters.

To evaluate each hyperparameter combination, cross-validation is used. In this context, K-Fold Cross-Validation is usually employed, where the data set is divided into K subsets. For each hyperparameter combination, the model is trained K times, using K-1 subsets for training and one for validation in each iteration. The average performance of the model over the K validation subsets is calculated, providing a robust estimate of its generalization ability for that specific hyperparameter configuration. During this process, performance metrics, such as mean square error or coefficient of determination, explained in following sections, are recorded for each hyperparameter combination. At the end of the evaluation of all combinations, the hyperparameter configuration that maximizes model performance on the validation sets is selected. This optimal configuration is the one that best balances the accuracy and generalization capability of the model, avoiding both underfitting and overfitting.

4.3.7 Validation Models

This section is focused on the description of the validation methods applied to the predictive models developed. Techniques such as split validation, K-Fold cross-validation, bootstrapping, and Grid Search CV, among others, will be discussed. Each method will be explained in detail, highlighting how they contribute to evaluate the robustness, accuracy, and generalization of the models against different data sets.

1. K-Fold Cross Validation

K-Fold Cross-Validation is a widely used technique to evaluate the generalization capability of a predictive model. As Mahmood and Khan (2009) explain, this technique is performed by dividing the data set into K subsets or "folds" of approximately equal size. The validation process consists of dividing the data set into K subsets randomly, where in each iteration one of these subsets is used as the validation set and the remaining K-1 are used to train the model. This process is repeated K times, using a different fold for validation in each iteration. Upon completion, the average performance of the model across the K iterations is calculated, providing a more accurate estimate of its generalizability. This method is effective in reducing variance in model evaluation and is especially useful when a limited data set is available. However, the computational cost can increase significantly with increasing K.

Figure 10 represents the process followed in the validation. It should be noted that each validation iteration has an associated error and because of the errors resulting from all iterations the average error is calculated. This average provides a more accurate estimate of the generalization capability of the model compared to a single training and validation partition.

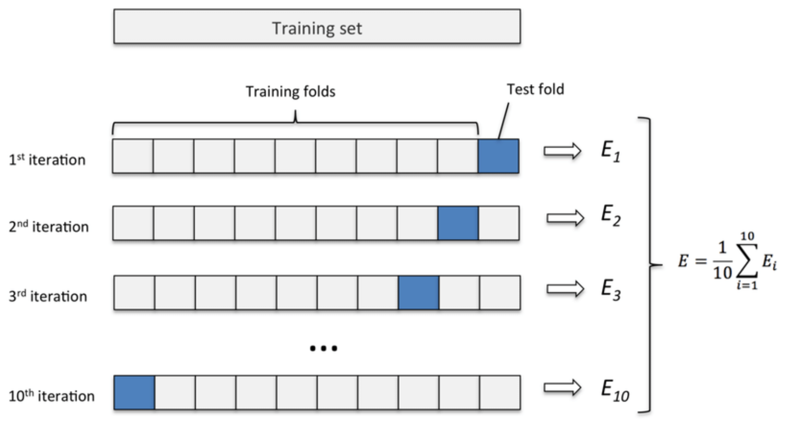


Figure 10. K-Fold Cross Validation process.

2. Bootstrapping

Bootstrapping is a statistical technique for estimating the distribution of a statistic by repeated sampling of the original data set with replacement. This approach is particularly useful for assessing the accuracy and uncertainty of predictive models. According to Egbert and Plonsky (2021), the methodology begins with the generation of multiple subsets from the original data set. Each subset is created by sampling with replacement, which means that the same observation can be selected more than once to be part of the subset. The size of each subset is equal to the size of the original data set, thus ensuring that the structure of the data is maintained.

For each subset generated, the predictive model is trained using all the data in the subset. This allows the model to learn from different combinations of data, capturing the variability present in the original data set. After training the model with each subset, its performance on observations that were not selected for the specific subset, known as “out-of-bag” samples, is evaluated. This evaluation provides a measure of the models’ accuracy and allows estimating the prediction error in a robust manner.

3. Split Validation

The split validation method, often known as holdout validation, is a simple and widely used technique for evaluating the performance of machine learning models. This method involves dividing the available dataset into two distinct subsets, as described by Yadav and Shukla (2016). The first subset is the training set, which is used to train the machine learning model. It typically comprises a significant portion of the original dataset, often around 70-80%. The second subset is the validation set, which is used to evaluate the performance of the model after it has been trained. It helps in tuning model parameters and selecting the best model, and usually comprises about 20-30% of the dataset.

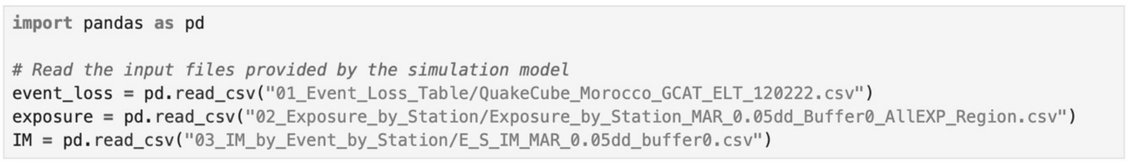
The process of split validation involves several steps. First, the dataset is randomly split into the training and validation sets. This splitting can be done using various methods, such as simple random sampling or stratified sampling to ensure class distribution is maintained in classification problems. Next, the model is trained using only the training set, where the training process involves learning the parameters or weights from the data, minimizing a loss function, and applying any optimization techniques. After the model is trained, it is validated on the validation set.

5. Results

5.1 Descriptive Analysis

First of all, in order to avoid congestion in the analysis and due to the similarity of the data sets and the behavior between variables, the descriptive analysis will be performed using the files corresponding to the 0.05dd cell size. This approach will allow to obtain a detailed and representative view of the data without excessive redundancies. Secondly, the execution of the codes has been carried out with Python 3. 11.

To start with the descriptive analysis of the data, it is necessary to import and load the data sets to be used in this study. Using the pandas library in Python, these files are read and converted to DataFrames as shown in Figure 11.



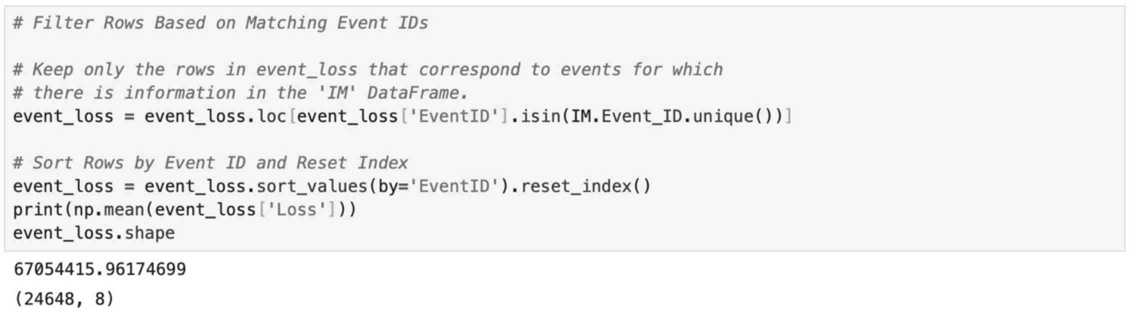
```
import pandas as pd

# Read the input files provided by the simulation model
event_loss = pd.read_csv("01_Event_Loss_Table/QuakeCube_Morocco_GCAT_ELT_120222.csv")
exposure = pd.read_csv("02_Exposure_by_Station/Exposure_by_Station_MAR_0.05dd_Buffer0_AllEXP_Region.csv")
IM = pd.read_csv("03_IM_by_Event_by_Station/E_S_IM_MAR_0.05dd_buffer0.csv")
```

Figure 11. File reading.

Pandas is an open-source Python library widely used for data manipulation and analysis. With pandas, it is possible to perform a wide range of operations, such as reading, cleaning, exploring, or transforming data, among others. On the other hand, a DataFrame consists of a two-dimensional data structure provided by the pandas library in Python. It can be considered similar to a table in a database, a spreadsheet in Excel or a matrix in mathematics, but with additional functionalities that facilitate data manipulation and analysis.

As described in Section 3 of this work, data transformation is performed to avoid overlapping. Once the data is loaded, the rows of the event_loss DataFrame are filtered to keep only those that correspond to events whose IDs are present in the IM DataFrame. This ensures that only events for which complete intensity information is available are analyzed. Subsequently, the rows of the event_loss DataFrame are sorted by the 'EventID' column and the DataFrame index is reset. This facilitates the organization and access to the data during analysis. To provide an overview of economic losses, the average of the 'Loss' column of the DataFrame event_loss is calculated and printed. This basic statistic helps to understand the average magnitude of the estimated economic losses. Finally, the shape of the DataFrame event_loss is shown, indicating the number of rows and columns it contains. This information is useful to understand the dimension of the dataset after applying filters and sorting.

The process is described in Figure 12, where the average earthquake losses are around 67 million USD, and that the dataset has a size of 24,648 rows and 8 columns.



```
# Filter Rows Based on Matching Event IDs

# Keep only the rows in event_loss that correspond to events for which
# there is information in the 'IM' DataFrame.
event_loss = event_loss.loc[event_loss['EventID'].isin(IM.Event_ID.unique())]

# Sort Rows by Event ID and Reset Index
event_loss = event_loss.sort_values(by='EventID').reset_index()
print(np.mean(event_loss['Loss']))
event_loss.shape
```

```
67054415.96174699
(24648, 8)
```

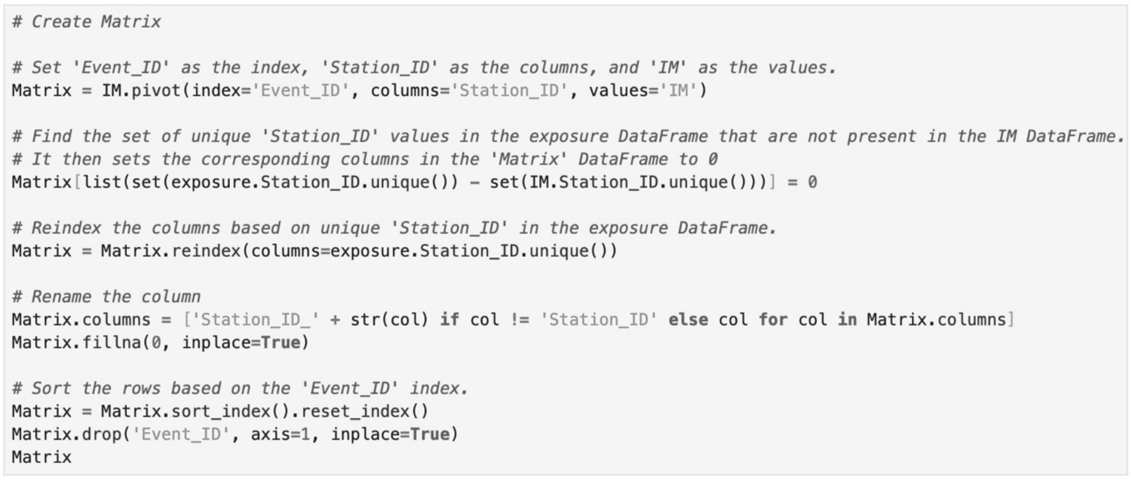
Figure 12. Shape analysis.

Next, a matrix is created that represents the intensity of seismic motions (MI) recorded at various stations for different seismic events. The resulting matrix is structured so that the events are the rows, and the stations are the columns, with the intensity values of the seismic motions as the matrix data.

First, the matrix is set up using the pandas pivot function. This function transforms the DataFrame IM by setting Event_ID as the index (rows), Station_ID as the columns, and IM as the matrix values. The result is a matrix where each row represents a seismic event and each column represents a station, with the cells containing the intensity values of the recorded seismic motions.

Then, the set of unique Station_ID values in the exposure DataFrame that are not present in the IM DataFrame are identified. For these stations that do not have seismic motion intensity records in IM, the corresponding value in the Matrix matrix is set to 0. This ensures that all stations present in exposure are included in the matrix, even if they have no data in IM.

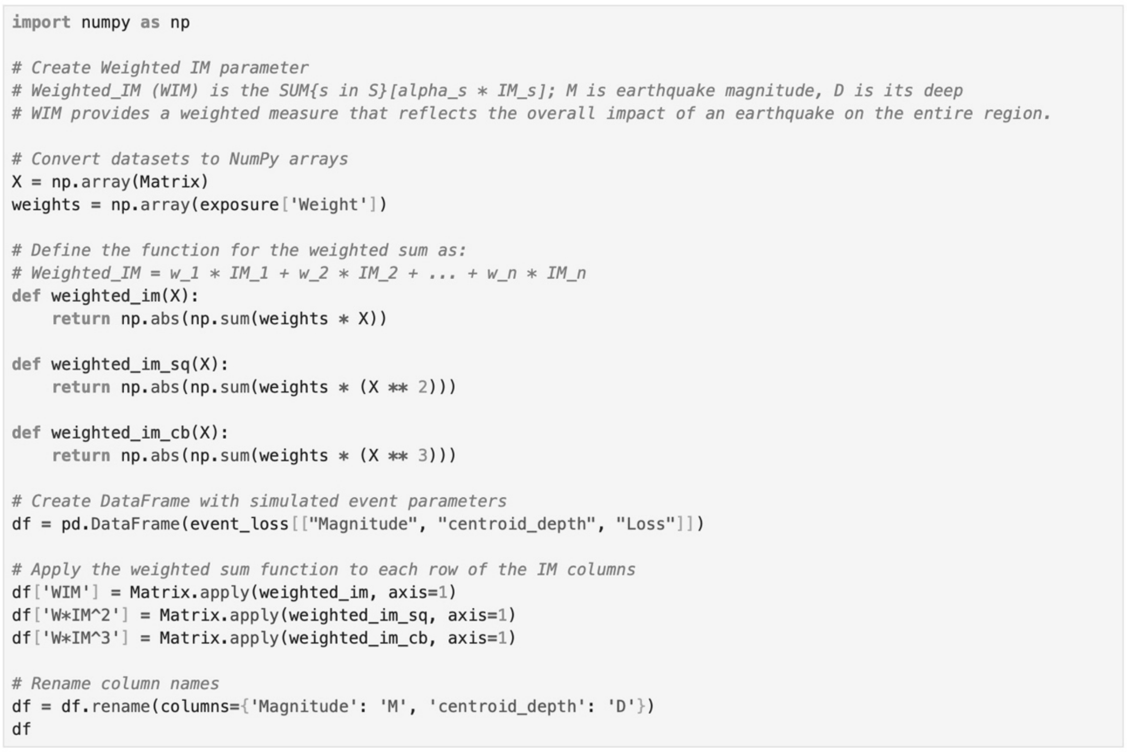
Finally, the rows of the matrix are sorted based on the Event_ID index and the index is reset. Then, the 'Event_ID' column is removed from the DataFrame, as it is no longer needed for the subsequent analysis. The result is a clean and organized matrix ready for further analysis. The resulting matrix contains the seismic intensity values organized in a way that facilitates comparative analysis between different stations and seismic events.



```
# Create Matrix

# Set 'Event_ID' as the index, 'Station_ID' as the columns, and 'IM' as the values.
Matrix = IM.pivot(index='Event_ID', columns='Station_ID', values='IM')

# Find the set of unique 'Station_ID' values in the exposure DataFrame that are not present in the IM DataFrame.
# It then sets the corresponding columns in the 'Matrix' DataFrame to 0
Matrix[list(set(exposure.Station_ID.unique()) - set(IM.Station_ID.unique()))] = 0

# Reindex the columns based on unique 'Station_ID' in the exposure DataFrame.
Matrix = Matrix.reindex(columns=exposure.Station_ID.unique())

# Rename the column
Matrix.columns = ['Station_ID_' + str(col) if col != 'Station_ID' else col for col in Matrix.columns]
Matrix.fillna(0, inplace=True)

# Sort the rows based on the 'Event_ID' index.
Matrix = Matrix.sort_index().reset_index()
Matrix.drop('Event_ID', axis=1, inplace=True)
Matrix
```

Figure 13. Matrix formation.

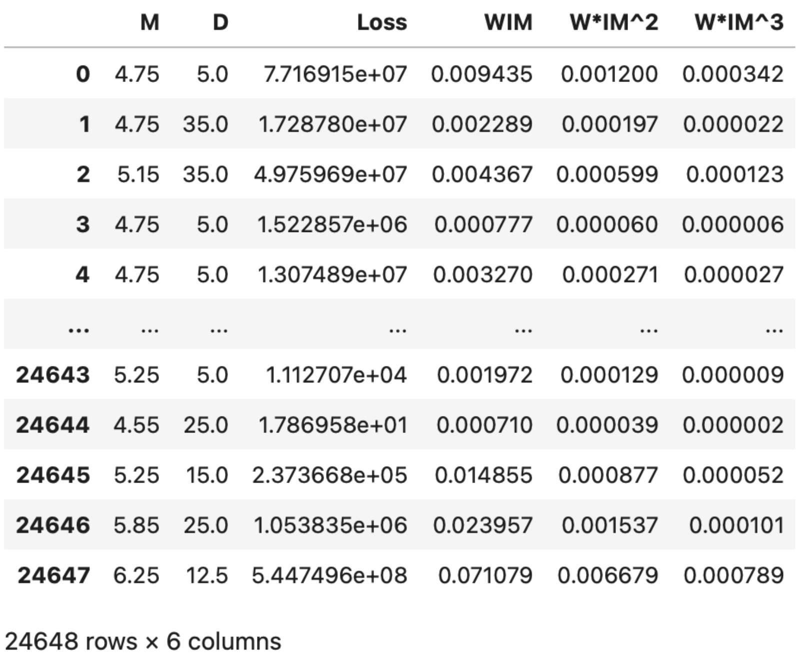
The following step is to create a Weighted IM (Weighted IM, or WIM) parameter that reflects the overall impact of an earthquake on an entire region. To do this, DataFrames are converted into NumPy arrays. An array is a data structure containing a collection of elements, and NumPy is a fundamental Python library for scientific computing that provides support for arrays and efficient mathematical operations. Functions are defined to compute the weighted sum of motion intensity (MI) values. These functions allow different forms of weighted sums to be calculated: basic, squared, and cubed. Then a new DataFrame df containing the simulated event parameters is created, and the weighted sum functions are applied to each row of the motion intensity columns in the Matrix matrix. The results are added as new columns in the DataFrame df. The columns of the DataFrame df are renamed for clarity. The column 'Magnitude' is renamed to 'M' and 'centroid_depth' is renamed to 'D'.



```python
import numpy as np

# Create Weighted IM parameter
# Weighted_IM (WIM) is the SUM{s in S}[alpha_s * IM_s]; M is earthquake magnitude, D is its deep
# WIM provides a weighted measure that reflects the overall impact of an earthquake on the entire region.

# Convert datasets to NumPy arrays
X = np.array(Matrix)
weights = np.array(exposure['Weight'])

# Define the function for the weighted sum as:
# Weighted_IM = w_1 * IM_1 + w_2 * IM_2 + ... + w_n * IM_n
def weighted_im(X):
    return np.abs(np.sum(weights * X))

def weighted_im_sq(X):
    return np.abs(np.sum(weights * (X ** 2)))

def weighted_im_cb(X):
    return np.abs(np.sum(weights * (X ** 3)))

# Create DataFrame with simulated event parameters
df = pd.DataFrame(event_loss[["Magnitude", "centroid_depth", "Loss"]])

# Apply the weighted sum function to each row of the IM columns
df['WIM'] = Matrix.apply(weighted_im, axis=1)
df['W*IM^2'] = Matrix.apply(weighted_im_sq, axis=1)
df['W*IM^3'] = Matrix.apply(weighted_im_cb, axis=1)

# Rename column names
df = df.rename(columns={'Magnitude': 'M', 'centroid_depth': 'D'})
df
```

Figure 14. WIM creation.

The output shown in Figure 15 is a pandas DataFrame containing the processed seismic event data, specifically, earthquake magnitude and depth values, associated losses, and WIM measurements. This DataFrame has 24,648 rows and 6 columns.

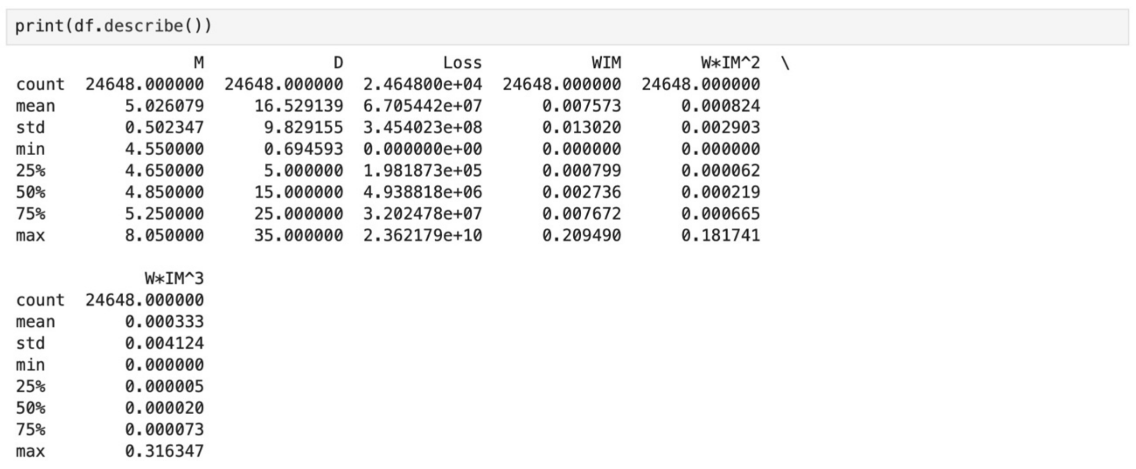


	M	D	Loss	WIM	W*IM^2	W*IM^3
0	4.75	5.0	7.716915e+07	0.009435	0.001200	0.000342
1	4.75	35.0	1.728780e+07	0.002289	0.000197	0.000022
2	5.15	35.0	4.975969e+07	0.004367	0.000599	0.000123
3	4.75	5.0	1.522857e+06	0.000777	0.000060	0.000006
4	4.75	5.0	1.307489e+07	0.003270	0.000271	0.000027
...	...	...	...	...	...	...
24643	5.25	5.0	1.112707e+04	0.001972	0.000129	0.000009
24644	4.55	25.0	1.786958e+01	0.000710	0.000039	0.000002
24645	5.25	15.0	2.373668e+05	0.014855	0.000877	0.000052
24646	5.85	25.0	1.053835e+06	0.023957	0.001537	0.000101
24647	6.25	12.5	5.447496e+08	0.071079	0.006679	0.000789

24648 rows × 6 columns

Figure 15. Output, earthquake event summary.

Once the dataset is prepared, the descriptive process begins. For this, the first step is to generate summary descriptive statistics of the numeric columns in the DataFrame. Using the describe() function of pandas, we obtain a summary statistic for each column, which includes the count, mean, standard deviation (std), minimum values (min), and the 25%, 50% (median) and 75% percentiles, as well as the maximum values (max).



```
print(df.describe())
```

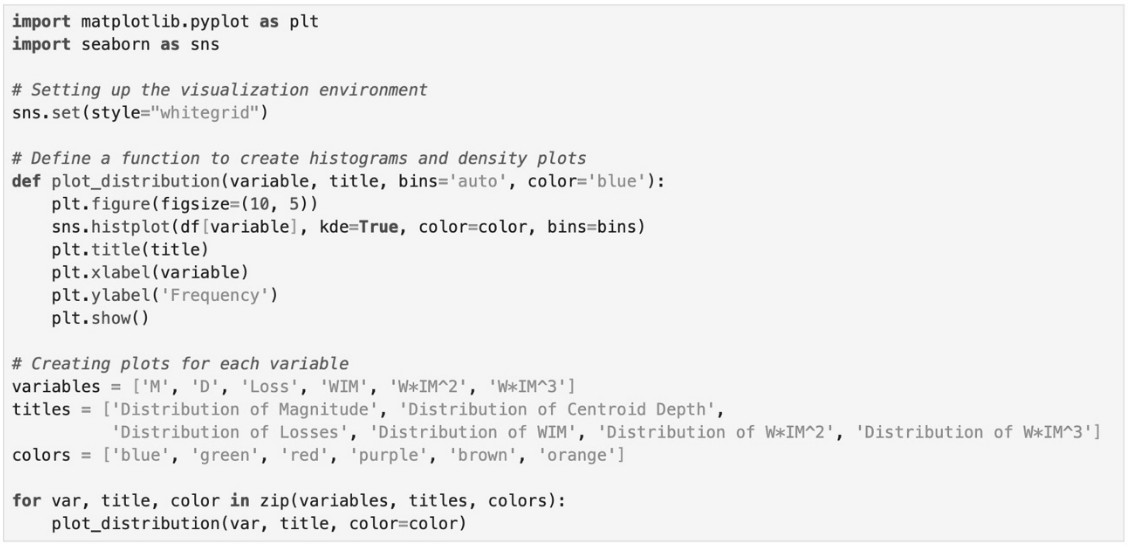
```
                  M             D          Loss           WIM        W*IM^2  \
count  24648.000000  24648.000000  2.464800e+04  24648.000000  24648.000000
mean       5.026079     16.529139  6.705442e+07      0.007573      0.000824
std        0.502347      9.829155  3.454023e+08      0.013020      0.002903
min        4.550000      0.694593  0.000000e+00      0.000000      0.000000
25%        4.650000      5.000000  1.981873e+05      0.000799      0.000062
50%        4.850000     15.000000  4.938818e+06      0.002736      0.000219
75%        5.250000     25.000000  3.202478e+07      0.007672      0.000665
max        8.050000     35.000000  2.362179e+10      0.209490      0.181741

             W*IM^3
count  24648.000000
mean       0.000333
std        0.004124
min        0.000000
25%        0.000005
50%        0.000020
75%        0.000073
max        0.316347
```

Figure 16. Descriptive statistics.

To better understand the distribution of data and visualize how different variables are distributed, it is useful to create distribution and density plots. These plots allow you to observe not only how the data is distributed, but also to identify possible anomalies, biases, and underlying patterns that might not be evident from basic descriptive statistics. Below, in Figure 17, is code that generates these plots for several key variables in a DataFrame. First, the necessary libraries are imported: matplotlib.pyplot as plt and seaborn as sns. matplotlib is a Python 2D graphics library that produces quality figures in a variety of formats. On the other hand, seaborn is a matplotlib-based data visualization library that provides a high-level interface for creating attractive and statistically informative plots.

The style of the graphs is then set using seaborn with sns.set(style="whitegrid"). The "whitegrid" style adds a white grid to the graphs, making it easier to read and visually interpret the data. This setting establishes the visual environment for all plots that will be created later. Then, a function called plot_distribution is defined that creates a histogram and a density plot for a given variable of the DataFrame df. The function takes as parameters the variable to be plotted, the title of the plot, the number of bins for the histogram and the color of the plot.

Within the function, plt.figure(figsize=(10, 5)) sets the figure size to 10 by 5 inches. The sns.histplot(df[variable], kde=True, color=color, bins=bins) line creates a histogram with a density line using seaborn. plt.title(title), plt.xlabel(variable) and plt.ylabel('Frequency') set the title of the plot and label the X and Y axes, respectively. Finally, plt.show() displays the graph. The code then defines three lists: variables, titles, and colors. The variables list contains the names of the DataFrame df columns to be plotted, titles contain the titles to be assigned to each graph and colors contains the colors to be used for each graph. A for loop iterates over these lists using the zip function, which allows iterating in parallel over multiple lists. On each iteration, the plot_distribution function is called with the current values of var, title and color, thus creating a distribution and density plot for each specified variable.



```python
import matplotlib.pyplot as plt
import seaborn as sns

# Setting up the visualization environment
sns.set(style="whitegrid")

# Define a function to create histograms and density plots
def plot_distribution(variable, title, bins='auto', color='blue'):
    plt.figure(figsize=(10, 5))
    sns.histplot(df[variable], kde=True, color=color, bins=bins)
    plt.title(title)
    plt.xlabel(variable)
    plt.ylabel('Frequency')
    plt.show()

# Creating plots for each variable
variables = ['M', 'D', 'Loss', 'WIM', 'W*IM^2', 'W*IM^3']
titles = ['Distribution of Magnitude', 'Distribution of Centroid Depth',
          'Distribution of Losses', 'Distribution of WIM', 'Distribution of W*IM^2', 'Distribution of W*IM^3']
colors = ['blue', 'green', 'red', 'purple', 'brown', 'orange']

for var, title, color in zip(variables, titles, colors):
    plot_distribution(var, title, color=color)
```

Figure 17. Visualization of the distribution of variables.

The generated graphs provide a visual representation of the distribution and density of the DataFrame features. Figure 18 shows the distribution of earthquake magnitude. A histogram with a superimposed density line indicates that most of the seismic events have a magnitude between 4.5 and 5.5. The distribution presents a clear decreasing trend as the magnitude increases, with a long tail to the right indicating the presence of some higher magnitude earthquakes, although they are less frequent. This pattern suggests that the most common earthquakes are of low to moderate intensity, with a few more intense events.

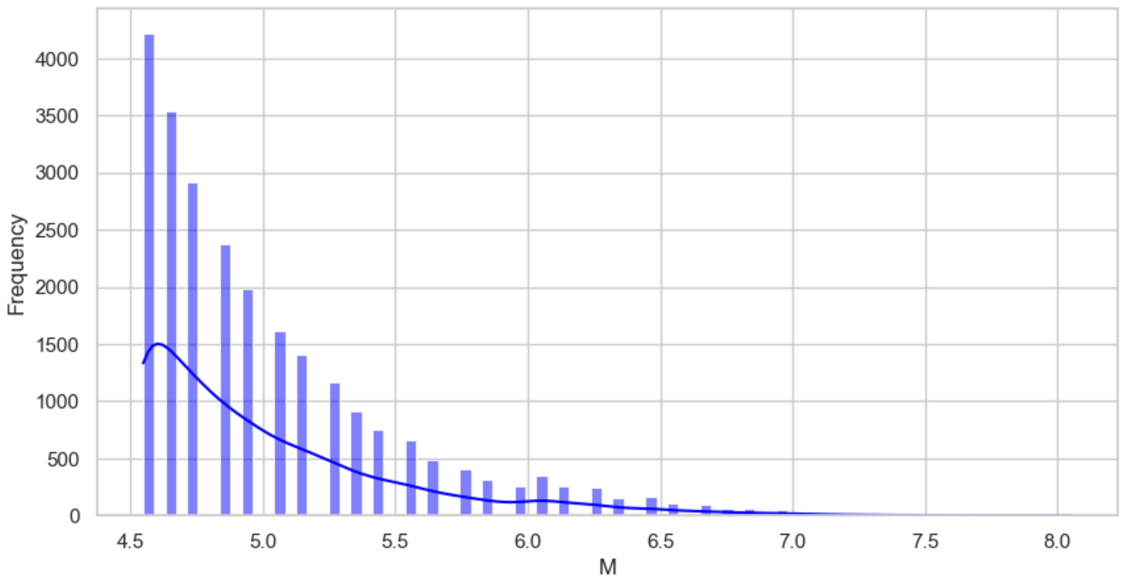


Figure 18. Magnitude distribution.

Continuing with the depth of the earthquake centroid, the second plot reveals pronounced peaks at specific depths, particularly around 5, 15, 25 and 35 kilometers. This concentration at certain depths suggests the existence of subduction zones or geological faults where earthquakes tend to occur more frequently, as explained above.

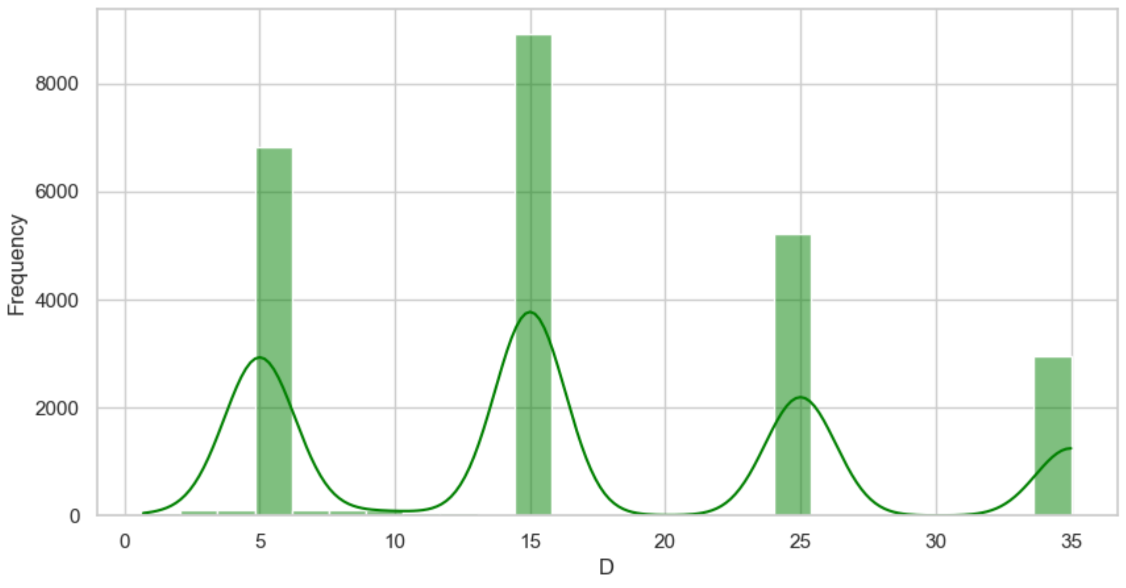


Figure 19. Centroid depth distribution.

The distribution of the estimated economic losses is presented in Figure 20. For better visualization, the X-axis has been delimited from $1\,e^{10}$ to $1\,e^{8}$, which allows a clearer representation of the data. Most of the losses are concentrated at low values, with a very high frequency at the left end of the plot. This indicates that most events result in minor economic losses, although there is a long tail to the right suggesting some events with significantly higher losses. This non-uniform distribution highlights the potentially devastating but rare nature of the more severe earthquakes.

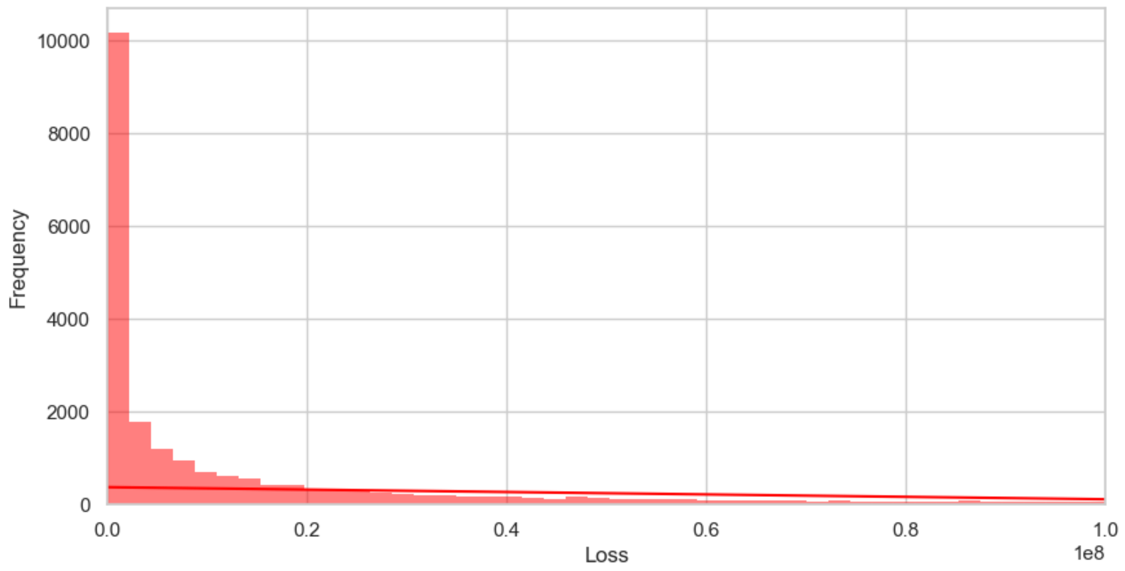


Figure 20. Loss distribution.

Finally, the fourth plot shows the distribution of the weighted intensity measure. Most of the WIM values are very small, concentrating near zero, with the density line decreasing rapidly as the values move away from this point. This distribution indicates that weighted earthquake intensities are generally low, with few events recording significantly higher intensities, as previously shown with the Loss feature.

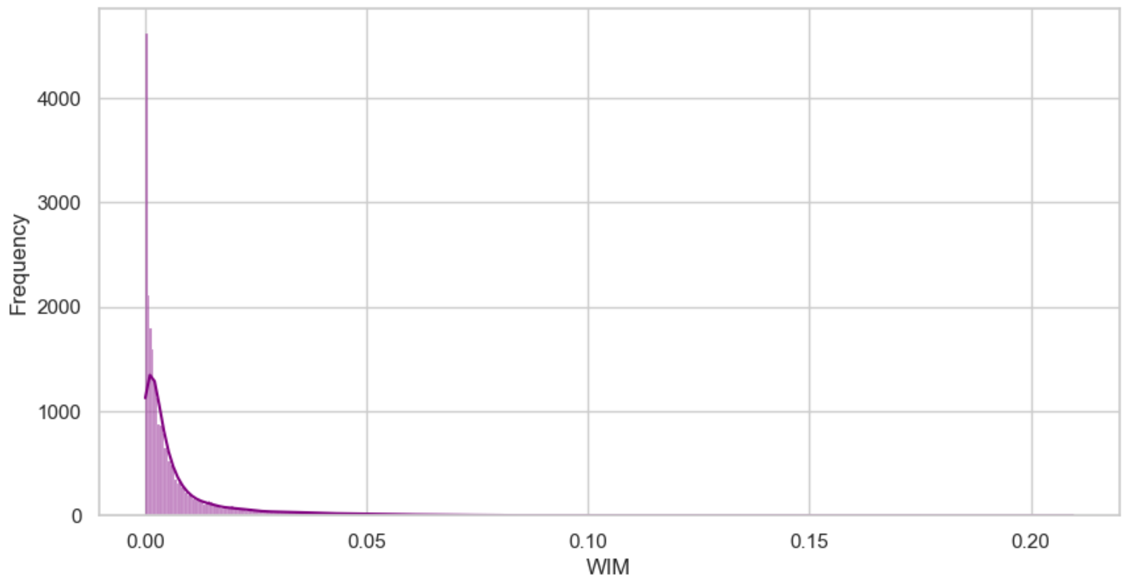
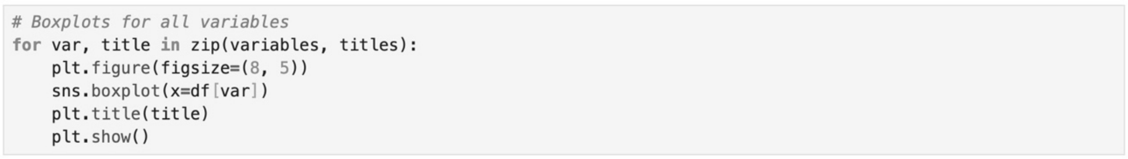


Figure 21. WIM distribution.

To complement the density figures and provide a more complete visual representation, boxplots have been generated. Boxplots are particularly useful for identifying median, quartiles and outliers in a data set, adding an additional layer of information that is not always evident in density figures. The code in Figure 22 generates boxplots for the DataFrame features using matplotlib and seaborn.

The for loop iterates over two lists: variables and titles, which contain the column names of the DataFrame and the titles for each graph, respectively. At each iteration, a new figure is created with plt.figure(figsize=(8, 5)) to set the size of the graph, a boxplot is generated for the current variable var using sns.boxplot(x=df[var]), a title is assigned to the graph with plt.title(title), and the graph is displayed with plt.show(). This process is repeated for each variable in the lists, producing an individual boxplot showing the distribution of each specified feature.



```
# Boxplots for all variables
for var, title in zip(variables, titles):
    plt.figure(figsize=(8, 5))
    sns.boxplot(x=df[var])
    plt.title(title)
    plt.show()
```

Figure 22. Boxplot generation.

The output obtained consists of a figure in which the central box represents the interquartile range (IQR), which is the distance between the first quartile (Q1) and the third quartile (Q3). The IQR contains the central 50% of the data. The line inside the box indicates the median (Q2) of the data set. The boxplot “whiskers” extend from the quartiles to the maximum and minimum values that are not considered outliers. Outliers are typically identified as points that are more than 1.5 times the IQR above the third quartile or below the first quartile. These points are represented by circles or dots outside the whiskers.

Figure 23 shows the boxplots of the dataset variables. Regarding the magnitude variable, numerous outliers are observed on the right, which are significantly larger magnitudes. No treatment of these values has been performed since they represent real events of higher magnitude earthquakes, which, although rare, can occur and have a large impact. In contrast to the magnitude, no outliers are observed in the depth distribution. This is because the depth of the earthquakes does not show extreme variations that are considered anomalous in this context, since there will always be losses when the epicenter is close to areas with exposure, which stabilizes the depth distribution.

Regarding losses, the central box is much smaller compared to the other variables, indicating that the interquartile range (IQR) of economic losses is relatively narrow. This suggests that the central 50% of the economic loss data is highly concentrated at low values. However, a large number of outliers are observed to the right, representing events with significantly higher economic losses. These outliers are more numerous than in the magnitude variable because, although extreme loss events are rare, they can have very high economic impacts. The presence of these outliers reflects the reality that some earthquakes can cause extremely high losses.

Lastly, many outliers are observed in the WIM variable on the far right, indicating that some events have significantly higher intensities. Although these events are rare, as mentioned, they represent situations where an earthquake has a much greater impact due to its location and the specific characteristics of the affected region.

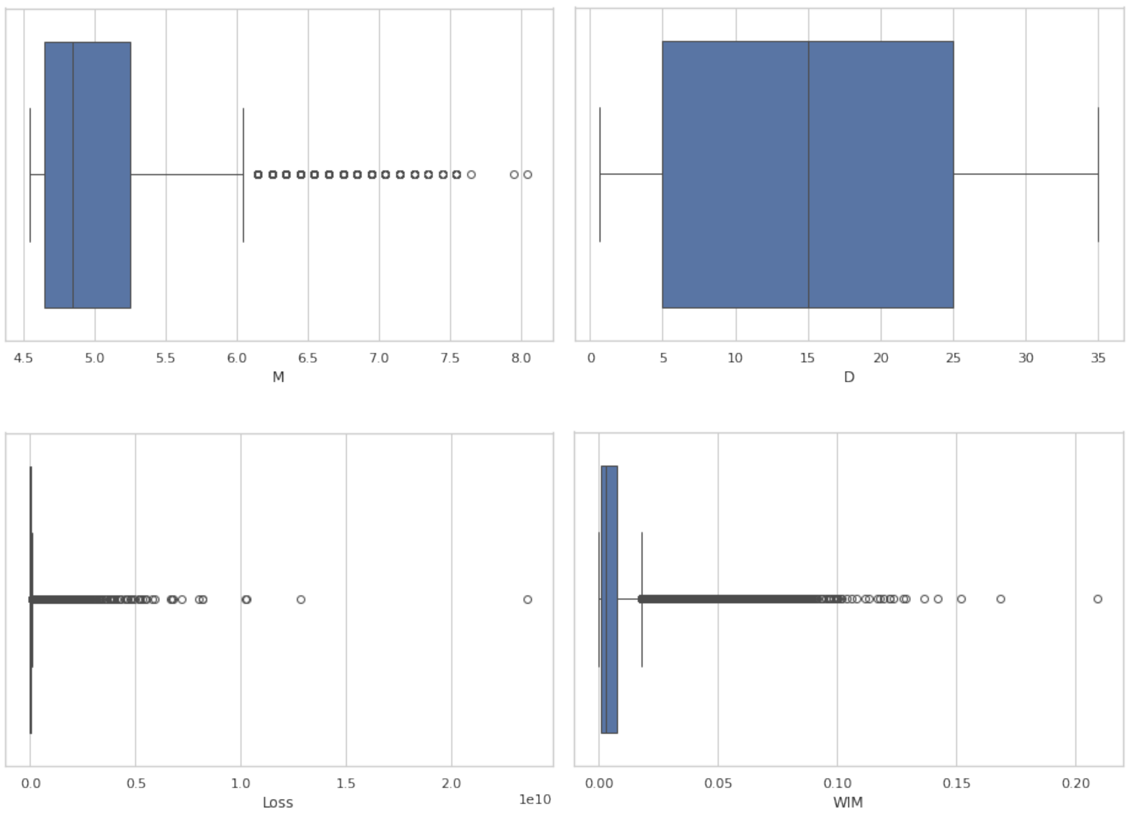
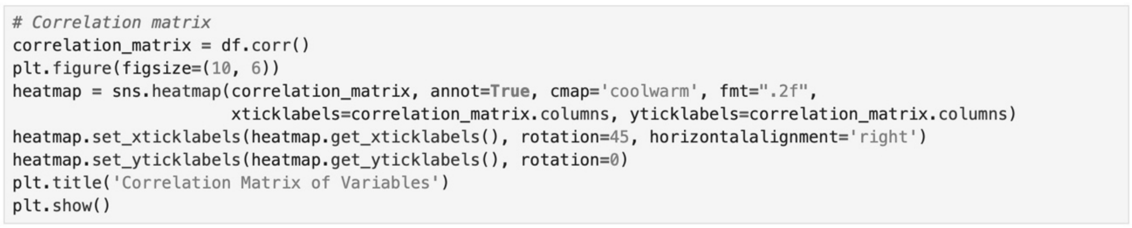


Figure 23. Boxplot of the features.

Continuing with the analysis, the correlation matrix was represented and visualized by means of a heat map using the code shown in Figure 24. Before describing the code, a correlation matrix is a table that shows the correlation coefficient between multiple variables, providing a measure that indicates the strength and direction of a linear relationship between two variables. Correlation coefficient values range from -1 to 1, where -1 indicates a perfect negative correlation, 1 indicates a perfect positive correlation, and 0 indicates no correlation.



```python
# Correlation matrix
correlation_matrix = df.corr()
plt.figure(figsize=(10, 6))
heatmap = sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f",
                      xticklabels=correlation_matrix.columns, yticklabels=correlation_matrix.columns)
heatmap.set_xticklabels(heatmap.get_xticklabels(), rotation=45, horizontalalignment='right')
heatmap.set_yticklabels(heatmap.get_yticklabels(), rotation=0)
plt.title('Correlation Matrix of Variables')
plt.show()
```

Figure 24. Correlation matrix generation.

The first step of the code is to create the correlation matrix using the corr() function of the DataFrame df, which calculates the Pearson correlation coefficient for all combinations of numeric columns in the DataFrame (Faizi, 2023). Next, the size of the figure is configured. To create the heatmap, the seaborn heatmap function is used, passing as parameters the calculated correlation matrix, the option annot=True to display the correlation values in each cell, and the color palette 'coolwarm' to visually represent the correlation values. In addition, the format of numbers displayed to two decimal places is specified by fmt=“.2f”, and the X and Y axis labels corresponding to the DataFrame columns are included. The X-axis labels are set to be rotated 45 degrees and aligned horizontally to the right, while the Y-axis labels are kept vertical, making the graph easier to read. Once the heatmap is set up, the title of the graph is set with plt.title() and the graph is displayed using plt.show().

The results of the correlation matrix show the relationships between the DataFrame variables: magnitude, depth, loss, and the weighted intensity measures. Magnitude and WIM have a moderate positive correlation (0.54), suggesting that an increase in earthquake magnitude is associated with an increase in the weighted intensity measure. Magnitude and loss have a low positive correlation (0.29), indicating that economic losses increase slightly with earthquake magnitude. WIM and loss have a moderate positive correlation (0.59), suggesting that higher weighted intensities are associated with higher economic losses. WIM^2 and loss have a very high correlation (0.95), and $WIM^3$and loss also show a high correlation (0.84), indicating a strong relationship between these variables. Depth shows very low or close to zero correlations with the other variables, indicating little relationship with the magnitude, loss, and weighted intensity measures.

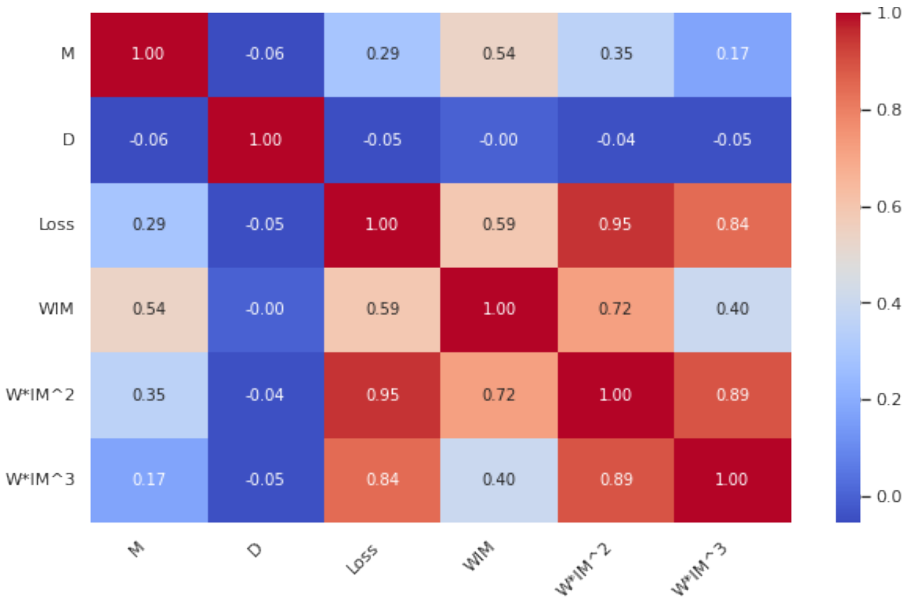
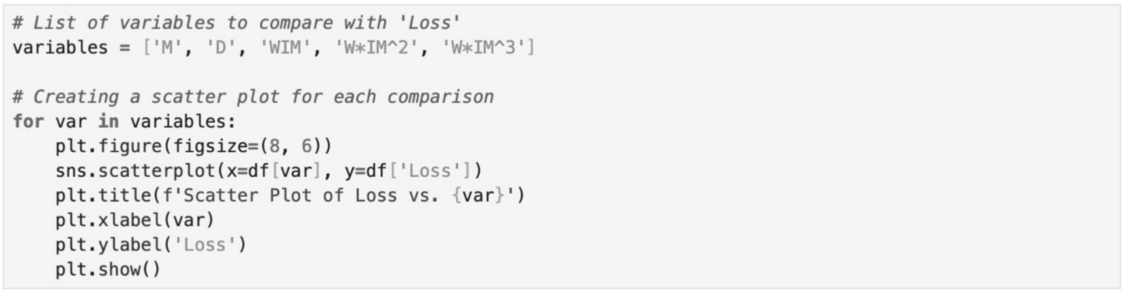


Figure 25. Correlation matrix

Continuing with the analysis, a bivariate analysis was carried out to explore the relationships between the variable 'Loss' and the others. Bivariate analysis, also known as pairwise variable analysis, is a graphical technique that allows visualizing possible correlations and patterns between two continuous features. According to the code shown in Figure 26, a list of variables to be compared with 'Loss' has been defined, including M, D, WIM and their squared and cubed versions. Using a for loop, scatter plots are generated for each of these variables. In each graph, the X-axis represents the variable in question and the Y-axis represents 'Loss'.



```
# List of variables to compare with 'Loss'
variables = ['M', 'D', 'WIM', 'W*IM^2', 'W*IM^3']

# Creating a scatter plot for each comparison
for var in variables:
    plt.figure(figsize=(8, 6))
    sns.scatterplot(x=df[var], y=df['Loss'])
    plt.title(f'Scatter Plot of Loss vs. {var}')
    plt.xlabel(var)
    plt.ylabel('Loss')
    plt.show()
```

Figure 26. Bivariate analysis.

The output of the code is a set of scatter plots where the relationships between Loss and the rest of the variables as mentioned above can be seen in Figure 27. The first figure shows the relationship between economic losses and the magnitude of the earthquake. In this graph, there is no clear linear correlation between these two variables. However, some scattered points can be identified, suggesting that events of greater magnitude do not necessarily result in greater losses, although there are some cases that do show a more significant relationship.

The second figure presents the relationship between economic losses and the depth of the earthquake centroid. This graph shows a large scatter in the data, indicating that earthquake depth does not have a direct and strong relationship with economic loss. The scatter suggests that both shallow and deep earthquakes can cause a wide range of economic losses.

The third figure shows the relationship between economic losses and WIM. An upward trend can be observed here, indicating that as the weighted intensity measure increases, economic losses also tend to increase. In addition, the values are clustered near 0, indicating that small magnitudes do not usually cause significant havoc, but the presence of a weighted intensity measure does not always imply that there will be considerable losses. This graph suggests a moderate positive relationship between these two variables.

The fourth figure presents the relationship between economic losses and WIM^2. Similar to the previous figure, a more pronounced upward trend is observed. As WIM is squared, the relationship appears to be stronger, indicating that the impact of weighted intensity on economic loss increases in a non-linear fashion. This transformation highlights the effects of extreme values, making the relationship between the variables more evident and stronger.

The fifth and final figure shows the relationship between economic losses and WIM^3. This graph shows an even clearer and stronger upward trend. As WIM is raised to the cube, the relationship intensifies, suggesting that economic losses increase significantly with increases in weighted intensity. This cubic transformation further accentuates the extreme values, showing a very strong relationship between the variables, indicating that this variable is highly influential on economic losses.

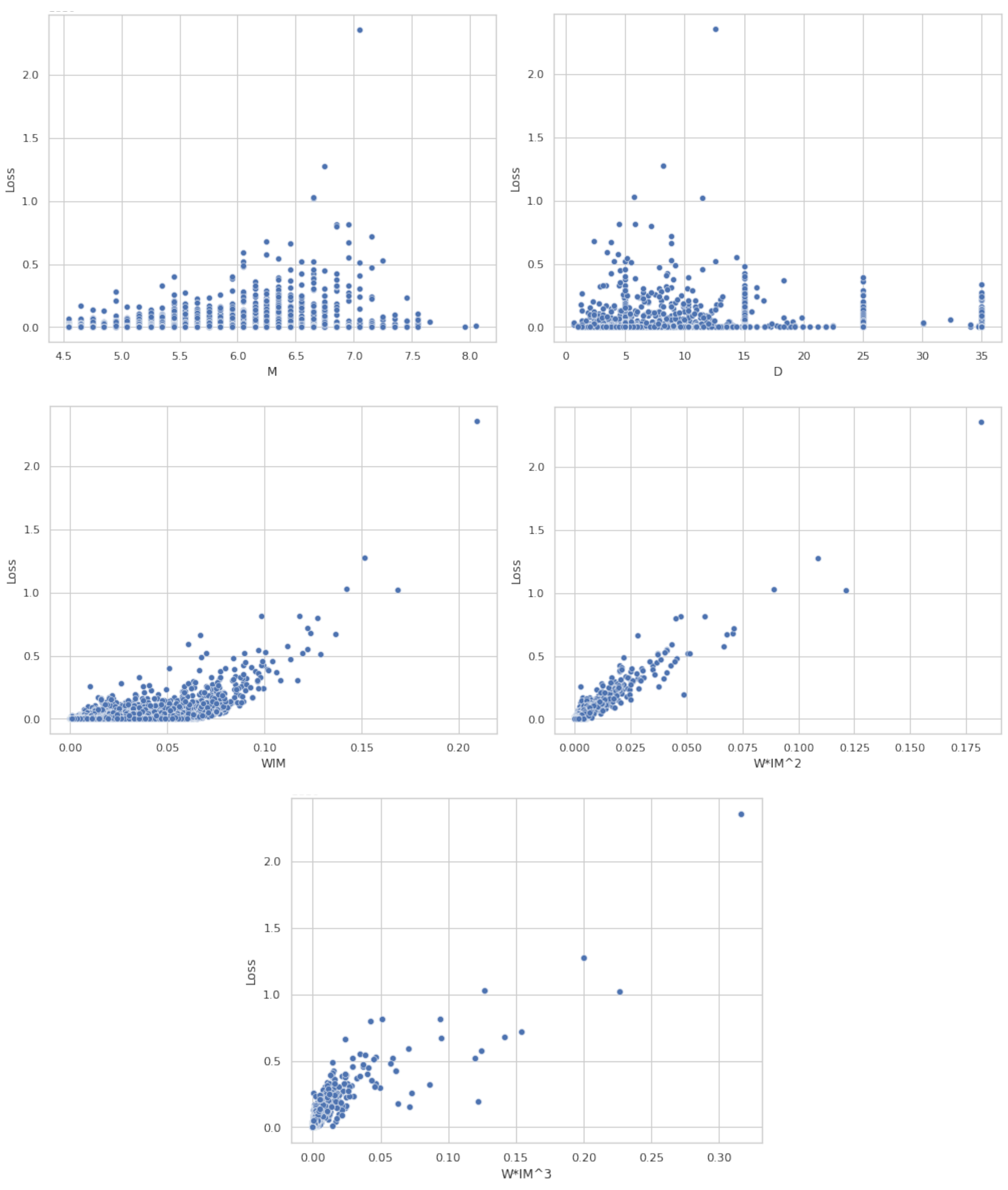


Figure 27. Scatter plots.

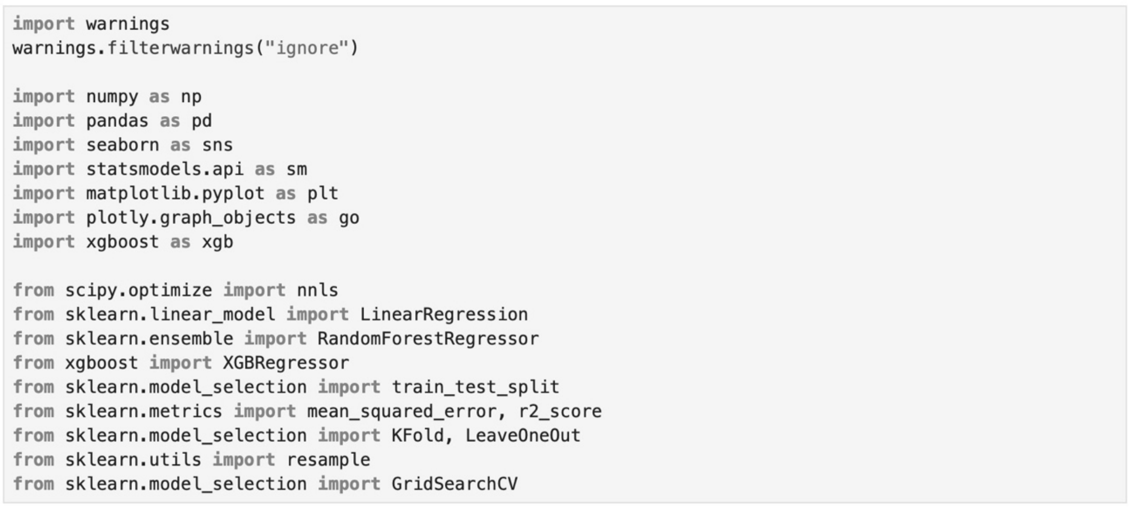
In the descriptive analysis performed, it was observed that the Depth variable does not show a significant correlation with the target variable, while WIM and its derived variables, as well as Magnitude, do show a high correlation. However, despite this high correlation with Magnitude, it was decided to exclude both this variable and Depth from the final model. According to Erdik et al. (2011), 88% of earthquake damage is attributed to ground shaking, a phenomenon that is effectively captured by the MI variable. In this context, Guy Carpenter chooses to model exclusively with WIM and its variants together with Loss, since WIM already indirectly integrates magnitude information through data collected by sensors covering a specific ground size (0.05, 0.1, 0.25 and 0.5dd). In this way, using only ground shaking as a variable, the model not only collects the most critical information to assess the impact of earthquakes, but also avoids redundancies by excluding Depth and Magnitude. The goal of the project is to understand how earthquakes affect structures and systems, not simply to collect or consider data that is more indicative of geographic features than direct impact.

5.2 Model Performance

This section discusses the performance of the models developed to forecast economic losses resulting from simulated events in the territory of Morocco. This evaluation is carried out in order to understand the methodology with which each model processes data and its ability to project results in divergent situations.

The analysis will focus on technical aspects of model fitting and parameter selection, providing a comprehensive view of the configuration and optimization of each implemented model. This study is an important step to validate the applicability of the models in the estimation of economic impacts derived from earthquakes and to ensure that the selected models offer a reliable and effective representation of the reality they attempt to model.

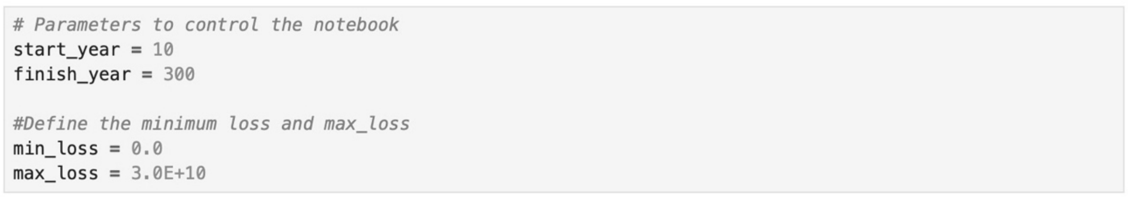
Before explaining the results of the model, some more libraries are imported that will be used and explained throughout the code and are represented in Figure 28.



```
import warnings
warnings.filterwarnings("ignore")

import numpy as np
import pandas as pd
import seaborn as sns
import statsmodels.api as sm
import matplotlib.pyplot as plt
import plotly.graph_objects as go
import xgboost as xgb

from scipy.optimize import nnls
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import KFold, LeaveOneOut
from sklearn.utils import resample
from sklearn.model_selection import GridSearchCV
```

Figure 28. Imported libraries.

A series of control parameters are also defined, as shown in Figure 29.



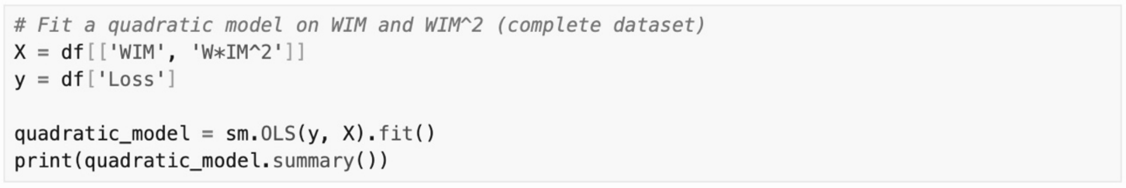
```
# Parameters to control the notebook
start_year = 10
finish_year = 300

#Define the minimum loss and max_loss
min_loss = 0.0
max_loss = 3.0E+10
```

Figure 29. Control parameters.

In the context of earthquake loss modeling, the starting year has been set to 10 to avoid the inclusion of initial outliers or extreme values that do not adequately represent the general behavior of seismic events, which tend to have very long recurrence cycles. On the other hand, the maximum loss value has been set at 30 billion, which corresponds to the maximum loss observed in the historical data, allowing a realistic evaluation framework for worst-case loss scenarios.

5.2.1 Quadratic Ordinary Least Squares Regression Model Performance

The code segment in Figure 30 illustrates the initial process with the quadratic regression model, fitting the entire dataset without previous splitting, and how the model is fitted using OLS with the "X" and "y" variables already defined, and then a statistical summary of the model is printed. This summary provides valuable information that helps interpret the effectiveness and accuracy of the model in terms of its ability to explain the variability in observed losses due to earthquakes.



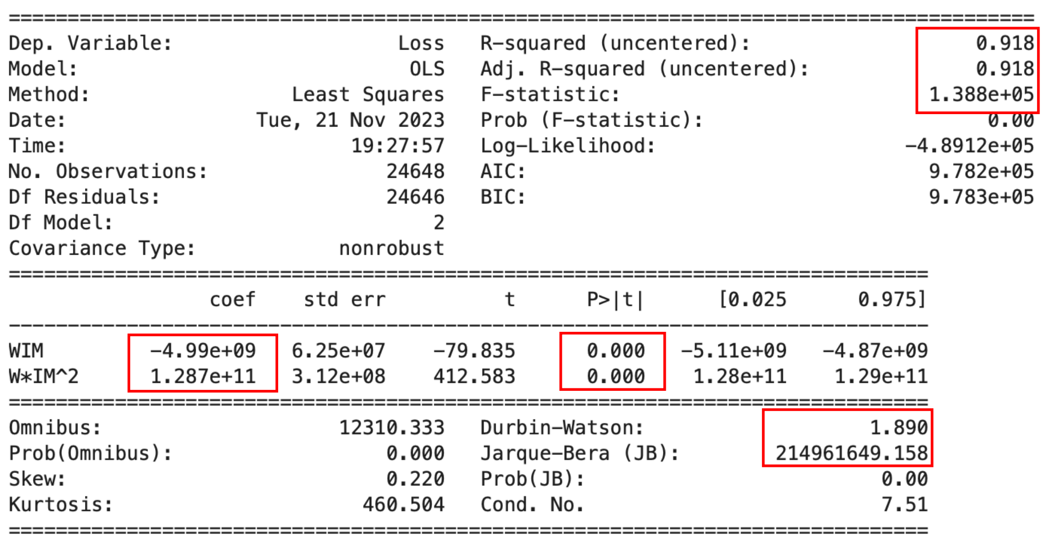
```
# Fit a quadratic model on WIM and WIM^2 (complete dataset)
X = df[['WIM', 'W*IM^2']]
y = df['Loss']

quadratic_model = sm.OLS(y, X).fit()
print(quadratic_model.summary())
```

Figure 30. OLS fitting for quadratic model.

The most relevant details of the analysis are depicted in Figure 31. The coefficient for the linear WIM term is significantly negative (-4.99e+09), which might initially seem counterintuitive. However, this result suggests that there is a threshold effect at lower intensities of ground motion, where further increases in WIM are not associated with an increase in losses, possibly due to the initial resistance of the structures to minor damage. In contrast, the coefficient for the quadratic term WIM^2 is positive (1.28e+11) and highly significant. This result indicates that as WIM increases, the impact on losses grows exponentially, reflecting the nonlinearity of losses in response to higher intensities of ground motion. This is typical in situations where small increases in ground motion intensity can lead to disproportionate increases in damage when certain structural strength thresholds are exceeded.

The F-statistic value is extremely high (1.388e+05), and the associated p-value is 0, indicating that the model is statistically significant at the global level and that the terms included in the model contribute significantly to the explanation of losses. The Durbin-Watson statistic is 1.890, which is close to the ideal value of 2, suggesting that there is no significant correlation between consecutive residuals in the data. This is important to validate the independence assumptions in the regression model. Regarding the Jarque-Bera test it has an extremely high value and a p-value of 0, indicating that the residuals do not follow a normal distribution. This could be a sign that there are other factors or relationships not captured by the current model or possible outliers or extreme influences in the data.

Finally, to measure the effectiveness of the model, we use the R-squared or also known as the coefficient of determination, which is a statistical measure used to assess the goodness of fit of a regression model. This indicator quantifies the proportion of the variability in the dependent variable that is predictable from the independent variables. In other words, the R-squared shows how well the data fit the proposed statistical model. The model has an R-squared of 0.918, indicating that the model explains approximately 91.8% of the variability in observed losses. This is an extremely high level of model fit, suggesting that the inclusion of linear and quadratic WIM terms provides a very accurate approximation for predicting losses based on soil movement intensity.



```
==============================================================================================
Dep. Variable:                   Loss   R-squared (uncentered):                      0.918
Model:                            OLS   Adj. R-squared (uncentered):                 0.918
Method:                 Least Squares   F-statistic:                             1.388e+05
Date:                Tue, 21 Nov 2023   Prob (F-statistic):                           0.00
Time:                        19:27:57   Log-Likelihood:                        -4.8912e+05
No. Observations:               24648   AIC:                                     9.782e+05
Df Residuals:                   24646   BIC:                                     9.783e+05
Df Model:                           2
Covariance Type:            nonrobust
==================================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------
WIM         -4.99e+09   6.25e+07    -79.835      0.000   -5.11e+09   -4.87e+09
W*IM^2      1.287e+11   3.12e+08    412.583      0.000    1.28e+11    1.29e+11
==================================================================================
Omnibus:                    12310.333   Durbin-Watson:                       1.890
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            214961649.158
Skew:                           0.220   Prob(JB):                             0.00
Kurtosis:                     460.504   Cond. No.                             7.51
==================================================================================
```

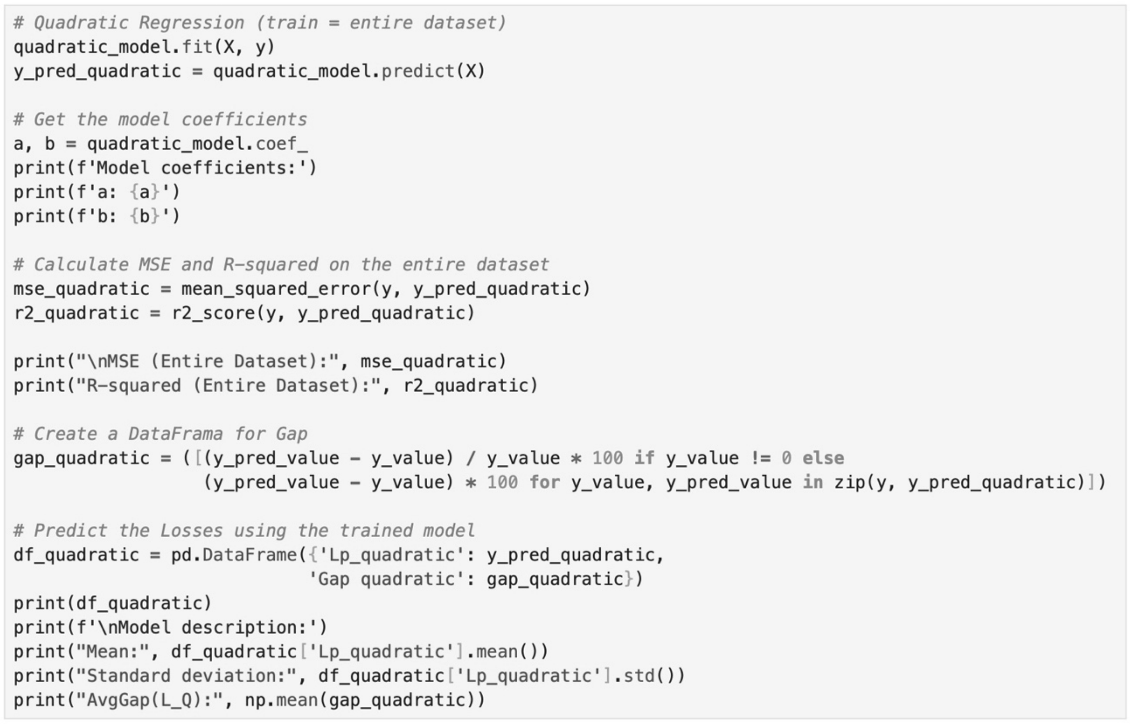
Figure 31. Quadratic regression model results.

Continuing with the analysis, the next step involves implementing a quadratic regression model on the entire dataset. This process is illustrated in the Figure 32, which demonstrates the fitting of the quadratic regression model, the evaluation of its coefficients, and the calculation of several performance metrics to assess its accuracy and reliability.

First, the quadratic regression model is fitted to the complete dataset. Once the model is trained, y_pred_quadratic predictions are generated for the entire data set. This allows the model to capture the underlying relationships between intensity measures and economic losses. After fitting, the coefficients of the quadratic model are extracted and displayed. The coefficients a and b represent the weights associated with the independent variables in the fitted quadratic equation. These coefficients provide information on the influence of each variable in the prediction of economic losses. In particular, a is the coefficient associated with 'WIM' and b is the coefficient associated with 'W*IM^2'.

To evaluate model performance, standard metrics are calculated in ML, such as the Mean Squared Error (MSE), which is the mean square of the differences between the observed and predicted values and provides a measure of the quality of the estimator, with lower values being indicative of a better fit, and the R-squared. Next, a DataFrame is created to analyze the percentage differences (Gap) between the actual losses and the predictions of the quadratic model. The computation of the gap_quadratic allows measuring the percentage difference between the predicted and actual losses for each observation, providing an additional measure of the model's accuracy. This DataFrame facilitates detailed analysis of the models' performance in terms of the accuracy of its predictions.

Finally, the predictions of the quadratic model are statistically described. The mean and standard deviation of the predicted losses are printed to provide an overview of the distribution of the model predictions. In addition, the average gap gives an indication of the overall accuracy of the model.

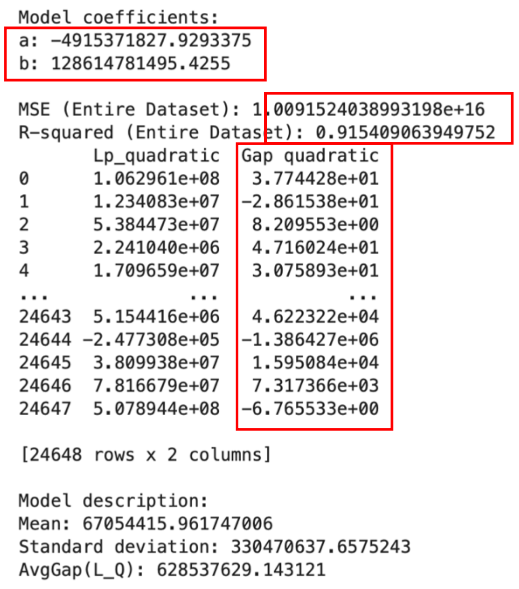


```
# Quadratic Regression (train = entire dataset)
quadratic_model.fit(X, y)
y_pred_quadratic = quadratic_model.predict(X)

# Get the model coefficients
a, b = quadratic_model.coef_
print(f'Model coefficients:')
print(f'a: {a}')
print(f'b: {b}')

# Calculate MSE and R-squared on the entire dataset
mse_quadratic = mean_squared_error(y, y_pred_quadratic)
r2_quadratic = r2_score(y, y_pred_quadratic)

print("\nMSE (Entire Dataset):", mse_quadratic)
print("R-squared (Entire Dataset):", r2_quadratic)

# Create a DataFrama for Gap
gap_quadratic = ([(y_pred_value - y_value) / y_value * 100 if y_value != 0 else
                  (y_pred_value - y_value) * 100 for y_value, y_pred_value in zip(y, y_pred_quadratic)])

# Predict the Losses using the trained model
df_quadratic = pd.DataFrame({'Lp_quadratic': y_pred_quadratic,
                             'Gap quadratic': gap_quadratic})
print(df_quadratic)
print(f'\nModel description:')
print("Mean:", df_quadratic['Lp_quadratic'].mean())
print("Standard deviation:", df_quadratic['Lp_quadratic'].std())
print("AvgGap(L_Q):", np.mean(gap_quadratic))
```

Figure 32. Performance metrics of quadratic regression model.

The output of the quadratic regression model, shown in Figure 33, provides detailed information on its performance and accuracy in predicting economic losses. The coefficients of the model, which indicate the influence of the independent variables in predicting losses, are presented first. The coefficient a is negative for the linear WIM term, suggesting that at low intensities, increases in WIM are not associated with an increase in losses, probably due to the initial strength of the structures. In contrast, the coefficient b for the quadratic WIM term is positive and significantly larger, indicating that at higher intensities of ground motion, losses grow exponentially.

The performance of the model is evaluated by MSE and R-squared. The MSE is 1.01e+16, which may not be a reliable indicator of accuracy in this context because the observed losses reach very high values, around 3.0e+10. In situations with such high values, the MSE may be disproportionately influenced by outliers. The R-squared is 0.915, indicating that the model explains approximately 91.5% of the variability in observed losses. This suggests that the model has an adequate level of fit. The resulting DataFrame shows the loss predictions (Lp_quadratic) and the percentage difference (Gap quadratic) between the predictions and the actual values. For example, the first row shows a predicted loss of 1.06e+08 and a gap of 37.74%, indicating that the prediction was 37.74% greater than the actual loss. This type of analysis helps to identify the accuracy of the model at each data point individually. In addition, a statistical description of the model is provided with the mean and standard deviation (variability in model predictions).



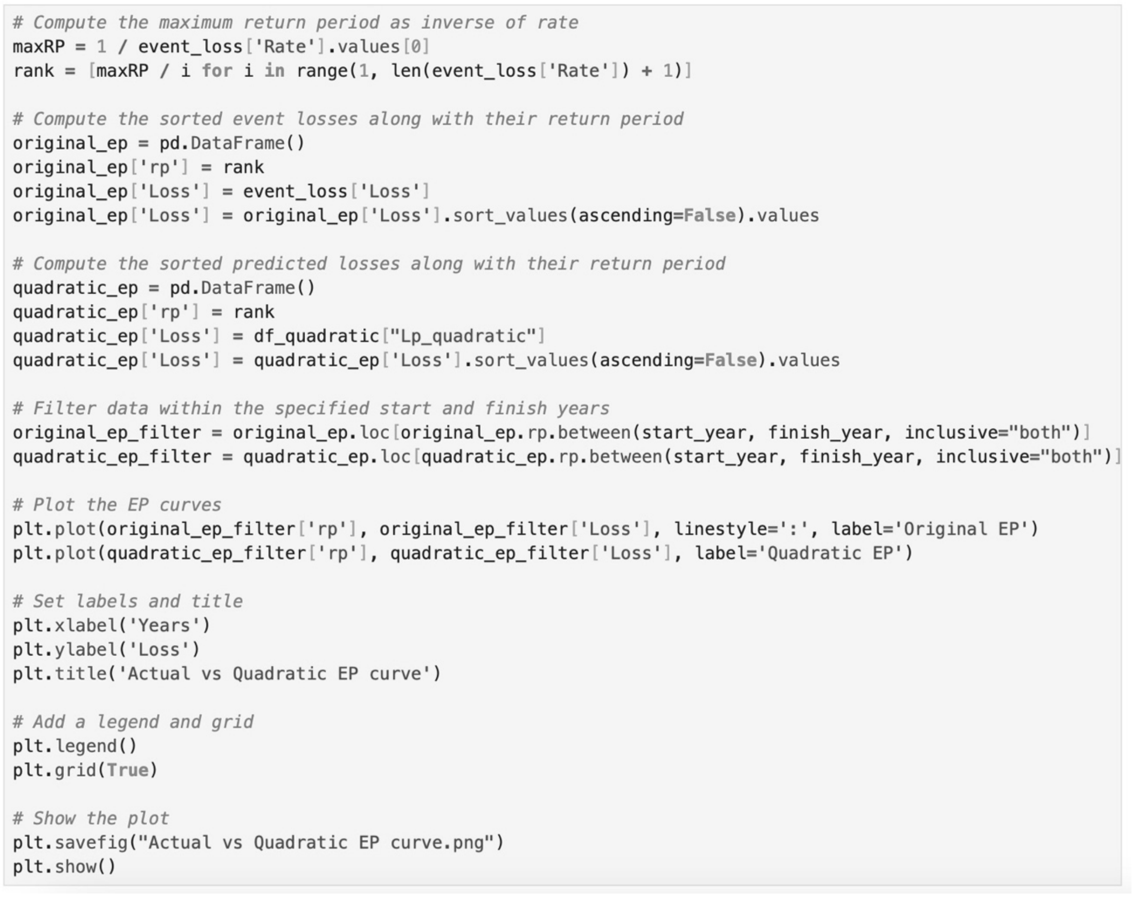
```
Model coefficients:
a: -4915371827.9293375
b: 128614781495.4255

MSE (Entire Dataset): 1.0091524038993198e+16
R-squared (Entire Dataset): 0.915409063949752
       Lp_quadratic  Gap quadratic
0      1.062961e+08   3.774428e+01
1      1.234083e+07  -2.861538e+01
2      5.384473e+07   8.209553e+00
3      2.241040e+06   4.716024e+01
4      1.709659e+07   3.075893e+01
...             ...            ...
24643  5.154416e+06   4.622322e+04
24644 -2.477308e+05  -1.386427e+06
24645  3.809938e+07   1.595084e+04
24646  7.816679e+07   7.317366e+03
24647  5.078944e+08  -6.765533e+00

[24648 rows x 2 columns]

Model description:
Mean: 67054415.961747006
Standard deviation: 330470637.6575243
AvgGap(L_Q): 628537629.143121
```

Figure 33. Performance metrics and statistical summary of quadratic regression model.

Following the detailed analysis of economic loss prediction models, a code is presented that calculates and compares return period curves for both observed losses and losses predicted by the quadratic regression model, shown in Figure 34. An EP curve (Exceedance Probability Curve) is a graphical tool used to represent the probability of losses exceeding a certain value in a specific time, as explained by Grossi and Windeler (2005). This curve is used in risk assessment, especially in sectors such as earthquake engineering, insurance planning and natural disaster management.

First, the maximum return period is determined as the inverse of the rate of the first event in the loss table. Subsequently, return period ranges are generated for each event based on its position. The next step is to calculate the ordered losses along with their corresponding return periods for both the original events and those predicted by the quadratic model. An original_ep DataFrame is created to store the return periods and the original losses, sorting them from highest to lowest. Similarly, a quadratic_ep DataFrame is created to store the return periods and losses predicted by the quadratic model, also sorted from highest to lowest. To fit the analysis to a specific range of years, the data is filtered according to the start_year and finish_year values.

Finally, the EP curves are plotted to compare the original losses with those predicted by the quadratic model. This visual comparison shows how the model predictions align with the observed losses as a function of return periods. This analysis is necessary to understand how the model handles variability in the data and to ensure that the predictions are consistent with the observed reality. By visualizing these EP curves, it is easier to identify possible discrepancies.



```python
# Compute the maximum return period as inverse of rate
maxRP = 1 / event_loss['Rate'].values[0]
rank = [maxRP / i for i in range(1, len(event_loss['Rate']) + 1)]

# Compute the sorted event losses along with their return period
original_ep = pd.DataFrame()
original_ep['rp'] = rank
original_ep['Loss'] = event_loss['Loss']
original_ep['Loss'] = original_ep['Loss'].sort_values(ascending=False).values

# Compute the sorted predicted losses along with their return period
quadratic_ep = pd.DataFrame()
quadratic_ep['rp'] = rank
quadratic_ep['Loss'] = df_quadratic["Lp_quadratic"]
quadratic_ep['Loss'] = quadratic_ep['Loss'].sort_values(ascending=False).values

# Filter data within the specified start and finish years
original_ep_filter = original_ep.loc[original_ep.rp.between(start_year, finish_year, inclusive="both")]
quadratic_ep_filter = quadratic_ep.loc[quadratic_ep.rp.between(start_year, finish_year, inclusive="both")]

# Plot the EP curves
plt.plot(original_ep_filter['rp'], original_ep_filter['Loss'], linestyle=':', label='Original EP')
plt.plot(quadratic_ep_filter['rp'], quadratic_ep_filter['Loss'], label='Quadratic EP')

# Set labels and title
plt.xlabel('Years')
plt.ylabel('Loss')
plt.title('Actual vs Quadratic EP curve')

# Add a legend and grid
plt.legend()
plt.grid(True)

# Show the plot
plt.savefig("Actual vs Quadratic EP curve.png")
plt.show()
```

Figure 34. EP curve generation.

The output is a plot, represented in Figure 3, which shows the comparison between the return period curves of the observed economic losses (Original EP) and the losses predicted by the quadratic regression model (Quadratic EP). The horizontal (x) axis represents the years, which indicate the return periods, while the vertical (y) axis shows the economic losses in values up to 4 billion (1e9) dollars.

The "Original EP" curve, represented by a blue dotted line, shows the observed losses ordered from highest to lowest over the different return periods. This curve serves as a reference to evaluate the accuracy of the predictive model. The "Quadratic EP" line, represented by a continuous orange line, shows the losses predicted by the quadratic regression model, also ordered from highest to lowest.

Comparing both curves, the “Quadratic EP” closely follows the trend of the “Original EP”, although with some differences. For shorter return periods (less than 50 years), the predicted losses are slightly lower than the observed losses. As the return period increases, the difference between the curves narrows, and the model-predicted losses align more closely with observed losses. However, at longer return periods (greater than 250 years), model-predicted losses tend to be slightly less than observed losses.

This visual comparison suggests that the quadratic regression model largely captures the relationship between economic losses and return periods, albeit with some discrepancies at the extremes of the data range. The proximity between the two curves indicates that the model is able to replicate patterns observed in the historical data, which is important for model validation.

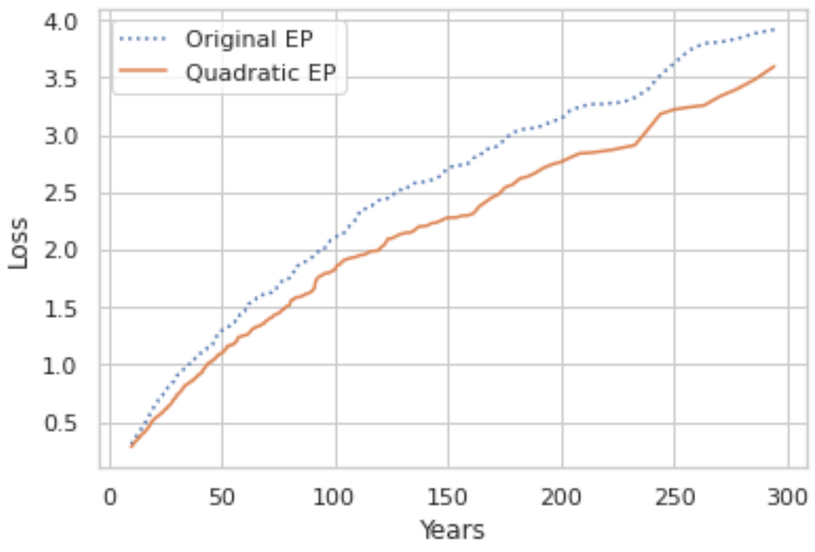
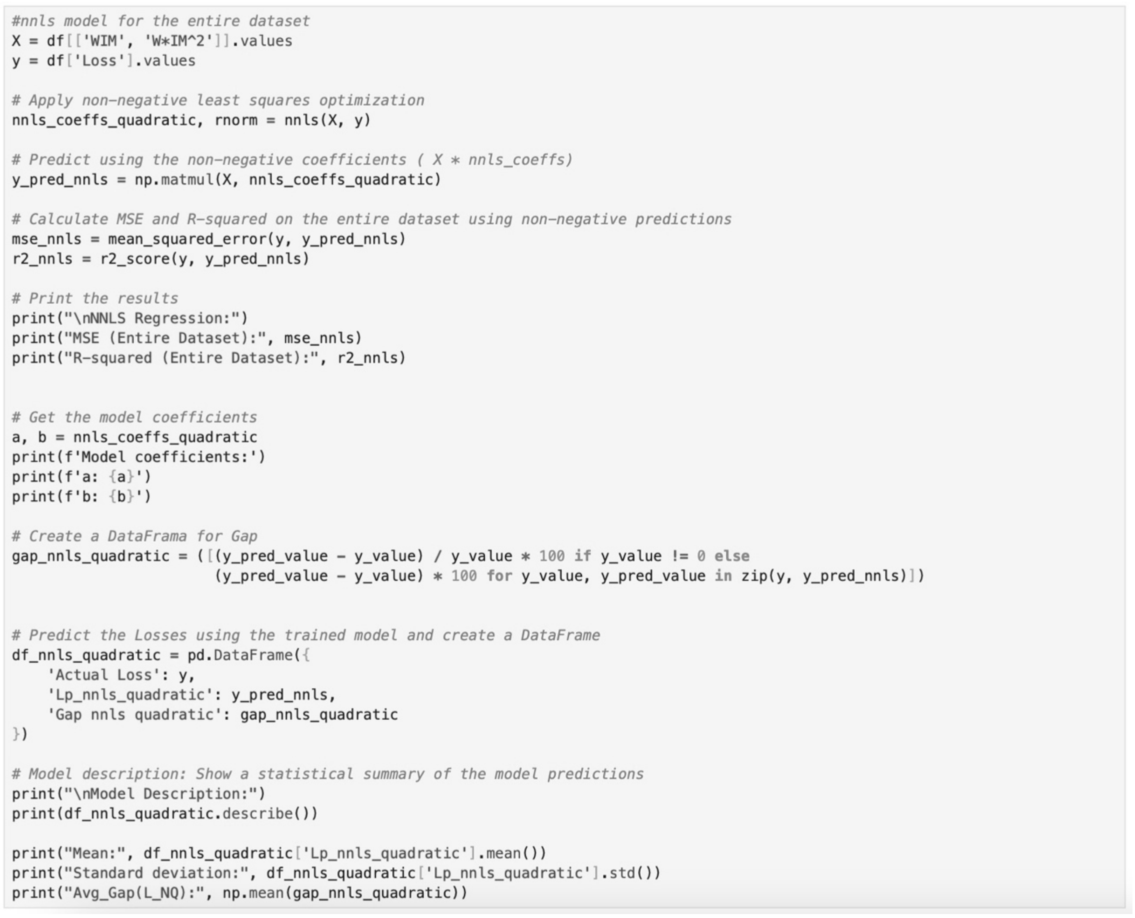


Figure 35. Actual vs quadratic EP curve.

Following quadratic regression, NNLS quadratic regression model was applied for the complete data set. The results of this model are presented below using the code represented in Figure 36. There is only one difference in the code and that is that the non-negative least squares optimization is applied using the nnls function of the scipy[3] library.

[3] SciPy (Scientific Python) is an open-source Python library for mathematics, science and engineering. It provides many useful functions for handling numerical data and performing scientific calculations.

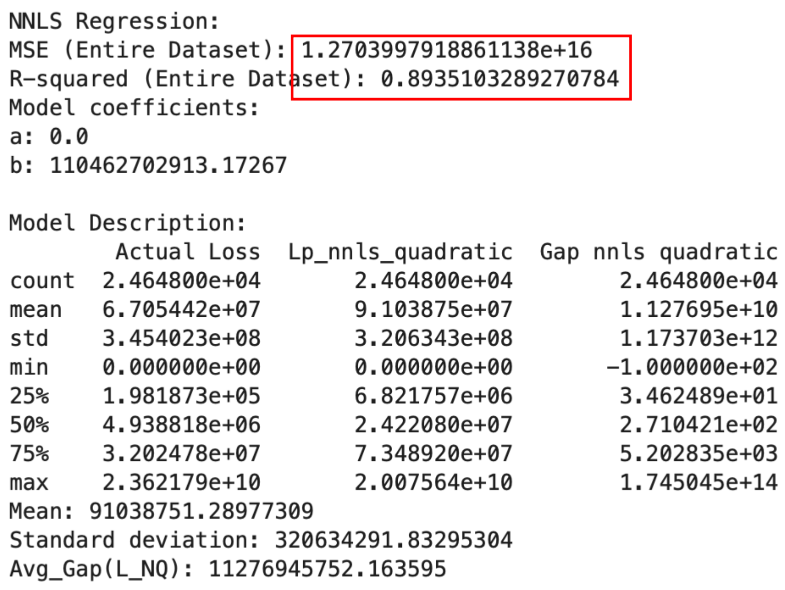


```python
#nnls model for the entire dataset
X = df[['WIM', 'W*IM^2']].values
y = df['Loss'].values

# Apply non-negative least squares optimization
nnls_coeffs_quadratic, rnorm = nnls(X, y)

# Predict using the non-negative coefficients ( X * nnls_coeffs)
y_pred_nnls = np.matmul(X, nnls_coeffs_quadratic)

# Calculate MSE and R-squared on the entire dataset using non-negative predictions
mse_nnls = mean_squared_error(y, y_pred_nnls)
r2_nnls = r2_score(y, y_pred_nnls)

# Print the results
print("\nNNLS Regression:")
print("MSE (Entire Dataset):", mse_nnls)
print("R-squared (Entire Dataset):", r2_nnls)


# Get the model coefficients
a, b = nnls_coeffs_quadratic
print(f'Model coefficients:')
print(f'a: {a}')
print(f'b: {b}')

# Create a DataFrama for Gap
gap_nnls_quadratic = ([(y_pred_value - y_value) / y_value * 100 if y_value != 0 else
                       (y_pred_value - y_value) * 100 for y_value, y_pred_value in zip(y, y_pred_nnls)])


# Predict the Losses using the trained model and create a DataFrame
df_nnls_quadratic = pd.DataFrame({
    'Actual Loss': y,
    'Lp_nnls_quadratic': y_pred_nnls,
    'Gap nnls quadratic': gap_nnls_quadratic
})

# Model description: Show a statistical summary of the model predictions
print("\nModel Description:")
print(df_nnls_quadratic.describe())

print("Mean:", df_nnls_quadratic['Lp_nnls_quadratic'].mean())
print("Standard deviation:", df_nnls_quadratic['Lp_nnls_quadratic'].std())
print("Avg_Gap(L_NQ):", np.mean(gap_nnls_quadratic))
```

Figure 36. Performance metrics of NNLS quadratic model.

The NNLS results show, in Figure 37, some important differences compared to the OLS model. The OLS model has a better fit to the data, reflected in a higher R-squared, indicating that it explains a greater proportion of the variability in observed losses. In addition, the MSE of the OLS model is lower than that of the NNLS model, suggesting higher prediction accuracy. The NNLS model, on the other hand, ensures that the coefficients are non-negative, which is an advantage in this context where the predictor variables are expected to have a positive impact. However, this restriction has affected its ability to fully capture the relationship between predictor variables and losses, resulting in a slightly inferior fit compared to OLS.

In terms of variability and accuracy of individual predictions, the OLS model also shows better performance, with smaller differences between predictions and actual values. In conclusion, although the NNLS model provides a valid alternative with specific constraints, the OLS model fits the data better and provides more accurate predictions in this case.



```
NNLS Regression:
MSE (Entire Dataset): 1.2703997918861138e+16
R-squared (Entire Dataset): 0.8935103289270784
Model coefficients:
a: 0.0
b: 110462702913.17267

Model Description:
        Actual Loss  Lp_nnls_quadratic  Gap nnls quadratic
count  2.464800e+04       2.464800e+04        2.464800e+04
mean   6.705442e+07       9.103875e+07        1.127695e+10
std    3.454023e+08       3.206343e+08        1.173703e+12
min    0.000000e+00       0.000000e+00       -1.000000e+02
25%    1.981873e+05       6.821757e+06        3.462489e+01
50%    4.938818e+06       2.422080e+07        2.710421e+02
75%    3.202478e+07       7.348920e+07        5.202835e+03
max    2.362179e+10       2.007564e+10        1.745045e+14
Mean: 91038751.28977309
Standard deviation: 320634291.83295304
Avg_Gap(L_NQ): 11276945752.163595
```

Figure 37. Performance metrics and statistical summary of NNLS quadratic model.

On the other hand, with respect to the EP curve shown in Figure 38, the NNLS quadratic model, although following the general trend, presents a lower accuracy than OLS, especially in the longer return periods, where it tends to underestimate losses.

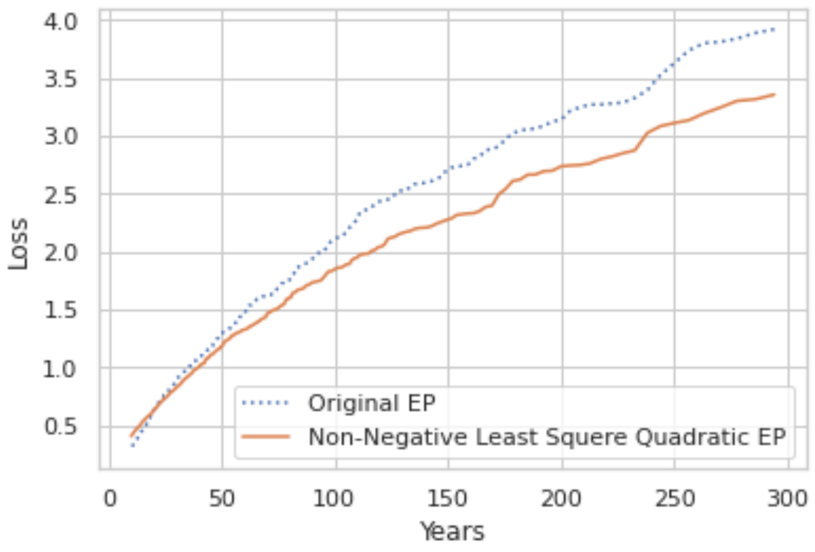
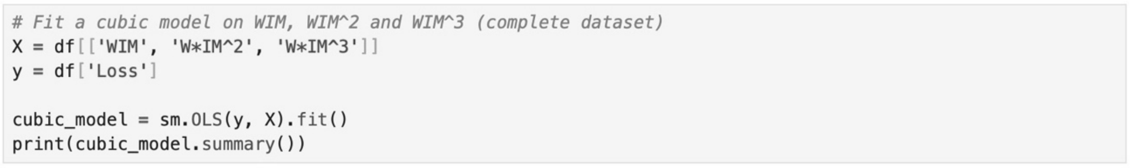


Figure 38. Actual vs NNLS quadratic EP curve.

5.2.2 *Cubic Ordinary Least Squares Regression Model Performance*

The cubic model analysis follows a similar procedure to the quadratic model, except for the model formulation. Figure 36 shows the code on how a cubic model is fitted using the variables 'WIM', 'WIM^2' and 'WIM^3' as predictors, and 'Loss' as the dependent variable. The procedure starts by defining the feature set X and the target variable y. The model is formulated using OLS applied to cubic transformations of the WIM. Once the model is fitted, a summary statistic is generated that provides information on model performance.



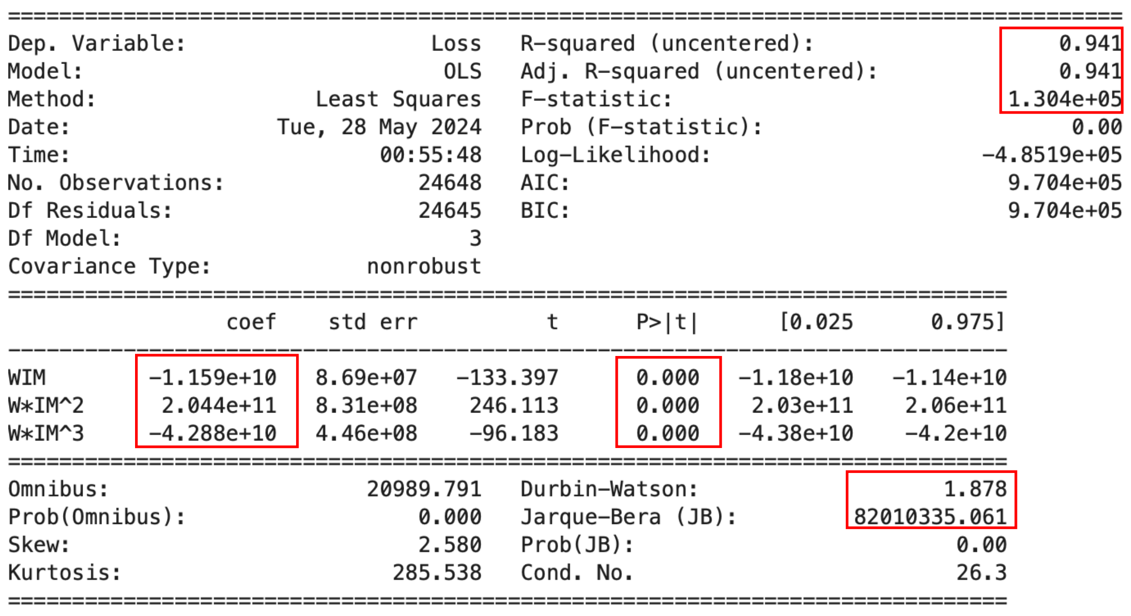
```
# Fit a cubic model on WIM, WIM^2 and WIM^3 (complete dataset)
X = df[['WIM', 'W*IM^2', 'W*IM^3']]
y = df['Loss']

cubic_model = sm.OLS(y, X).fit()
print(cubic_model.summary())
```

Figure 39. OLS fitting for cubic model.

The results of the cubic model applied to the full data set are presented below in Figure 37. The coefficient for the linear WIM term is significantly negative (-1.159e+10), which might seem contradictory at first. However, this result suggests that there is a threshold effect at low intensities of ground motion, where further increases in WIM are not associated with an increase in losses, possibly due to the initial resistance of structures to minor damage. In contrast, the coefficient for the quadratic term WIM^2 is positive (2.044e+11) and highly significant. This indicates that as WIM increases, the impact on losses grows exponentially, reflecting the nonlinearity of losses in response to higher intensities of ground motion. The coefficient for the cubic term WIM^3 is negative (-4.288e+10), introducing additional curvature that may better capture extreme variations in the data.

The F-statistic value is extremely high (1.304e+05), and the associated p-value is 0, indicating that the model is statistically significant at the global level and that the terms included in the model contribute significantly to the explanation of the losses. The Durbin-Watson statistic is 1.878, which is close to the ideal value of 2, suggesting that there is no significant correlation between consecutive residuals in the data, thus validating the assumptions of independence in the regression model. As for the Jarque-Bera test, the extremely high value, and a p-value of 0 indicate that the residuals do not follow a normal distribution.

The cubic model has an R-squared of 0.941, indicating that the model explains approximately 94.1% of the variability in observed losses. This high level of fit suggests that the inclusion of linear, quadratic, and cubic WIM terms provides a very accurate approximation for predicting losses based on ground motion intensity.

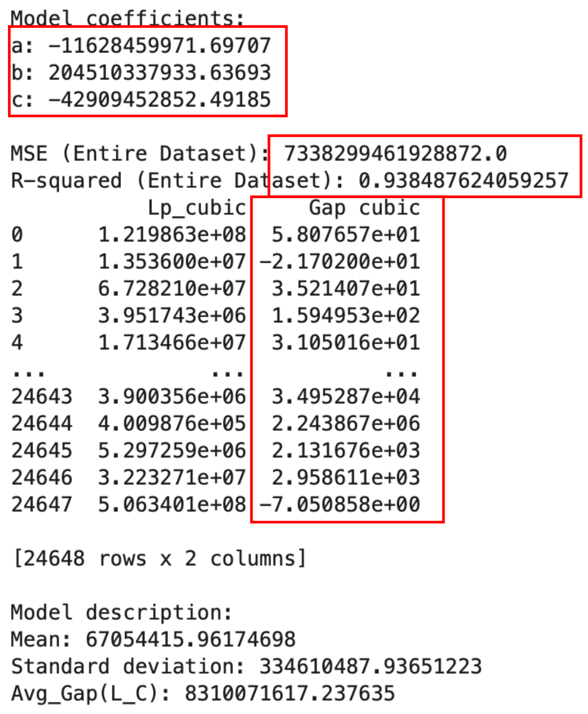


```
==========================================================================================
Dep. Variable:                   Loss   R-squared (uncentered):                   0.941
Model:                            OLS   Adj. R-squared (uncentered):              0.941
Method:                 Least Squares   F-statistic:                          1.304e+05
Date:                Tue, 28 May 2024   Prob (F-statistic):                        0.00
Time:                        00:55:48   Log-Likelihood:                     -4.8519e+05
No. Observations:               24648   AIC:                                  9.704e+05
Df Residuals:                   24645   BIC:                                  9.704e+05
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
WIM        -1.159e+10   8.69e+07   -133.397      0.000   -1.18e+10   -1.14e+10
W*IM^2      2.044e+11   8.31e+08    246.113      0.000    2.03e+11    2.06e+11
W*IM^3     -4.288e+10   4.46e+08    -96.183      0.000   -4.38e+10    -4.2e+10
==============================================================================
Omnibus:                    20989.791   Durbin-Watson:                   1.878
Prob(Omnibus):                  0.000   Jarque-Bera (JB):         82010335.061
Skew:                           2.580   Prob(JB):                         0.00
Kurtosis:                     285.538   Cond. No.                         26.3
==============================================================================
```

Figure 40. Cubic regression model results.

The cubic model analysis is carried out by applying the same method as in the quadratic model but including a cubic term in the formulation. The model is fitted to the complete data using WIM, WIM^2 and WIM^3 as predictors and 'Loss' as the dependent variable. The results obtained are detailed in Figure 38.

The coefficients of the cubic model are a = -11628459971.69707, b = 204510337933.63693, and c = -42909452852.49185. This suggests a nonlinear relationship between losses and intensity measurements, with the negative linear term, the positive quadratic term, and the negative cubic term reflecting a complex relationship. The MSE is 7.33e+16, reflecting the magnitude of the modeled losses. The R-squared of the cubic model is 0.93, indicating that approximately 93.8% of the variability in observed losses is explained. The resulting DataFrame shows the loss predictions (Lp_cubic) and the percentage difference between the predictions and the actual values. For example, a predicted loss of 1.21e+08 with a gap of 58.07% suggests a prediction greater than the actual loss by that percentage. In short, the cubic model shows a high R-squared, suggesting a good fit to the data and effectively capturing the variability in economic losses due to earthquakes.



```
Model coefficients:
a: -11628459971.69707
b: 204510337933.63693
c: -42909452852.49185

MSE (Entire Dataset): 7338299461928872.0
R-squared (Entire Dataset): 0.938487624059257
          Lp_cubic     Gap cubic
0      1.219863e+08  5.807657e+01
1      1.353600e+07 -2.170200e+01
2      6.728210e+07  3.521407e+01
3      3.951743e+06  1.594953e+02
4      1.713466e+07  3.105016e+01
...             ...           ...
24643  3.900356e+06  3.495287e+04
24644  4.009876e+05  2.243867e+06
24645  5.297259e+06  2.131676e+03
24646  3.223271e+07  2.958611e+03
24647  5.063401e+08 -7.050858e+00

[24648 rows x 2 columns]

Model description:
Mean: 67054415.96174698
Standard deviation: 334610487.93651223
Avg_Gap(L_C): 8310071617.237635
```

Figure 41. Performance metrics and statistical summary of cubic regression model.

Regarding the EP curve, Figure 39 shows that the cubic model closely follows the trend of the observed losses, similar to the quadratic model. However, the comparison shows some important differences. At the shortest return periods (less than 50 years), the cubic model predictions are closely aligned with the observed losses, fitting slightly better than the quadratic model in this range.

As return periods are extended, both the cubic and quadratic models maintain close alignment with observed losses. However, the cubic model shows a slight superiority in prediction accuracy in the intermediate periods (between 50 and 200 years). This higher accuracy is due to the ability of the cubic term to capture the additional complexities in the relationship between ground motion intensity and economic losses.

At longer return periods (greater than 250 years), both cubic and quadratic curves tend to converge, although the cubic model has a slight advantage in reflecting extreme variations in the observed data. In general, the cubic model provides a more accurate approximation over various return period ranges, suggesting its better ability to handle nonlinearity in the data.

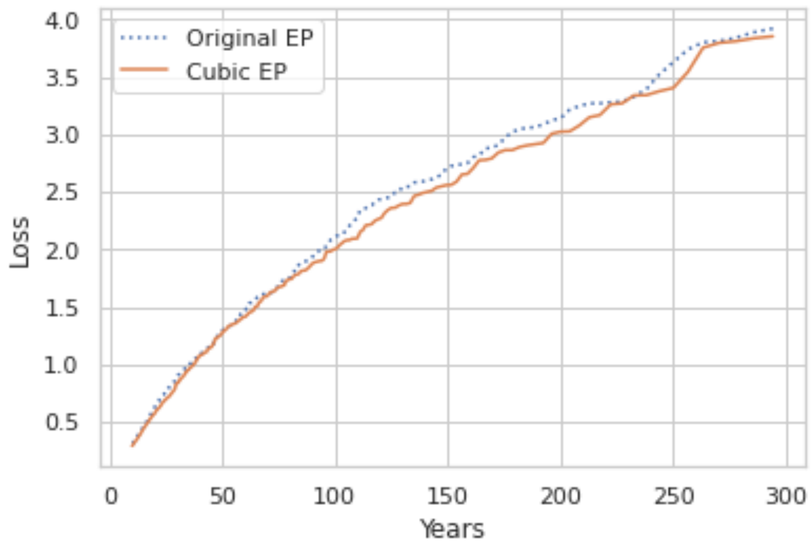
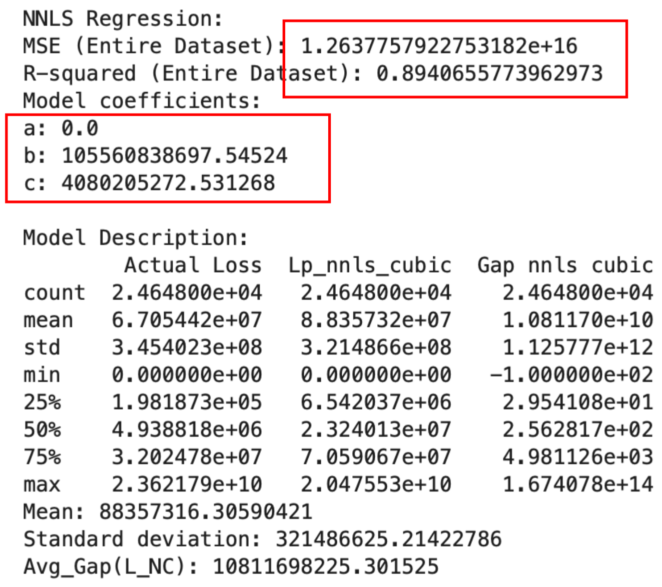


Figure 42. Actual vs cubic EP curve.

As in the quadratic model, the NNLS cubic regression model has also been carried out in the cubic model. When comparing the results of the cubic OLS model with the cubic NNLS, there are also clear differences in terms of performance. As shown in Figure 43, the MSE of the NNLS model is slightly higher than that of the OLS model, indicating that the latter has a better fit. In addition, the R-squared of the NNLS model is 0.89 as opposed to OLS which is 0.93, which leads to a better explanation of the variability of the observed losses by the OLS model.



```
NNLS Regression:
MSE (Entire Dataset): 1.2637757922753182e+16
R-squared (Entire Dataset): 0.8940655773962973
Model coefficients:
a: 0.0
b: 105560838697.54524
c: 4080205272.531268

Model Description:
        Actual Loss  Lp_nnls_cubic  Gap nnls cubic
count  2.464800e+04   2.464800e+04    2.464800e+04
mean   6.705442e+07   8.835732e+07    1.081170e+10
std    3.454023e+08   3.214866e+08    1.125777e+12
min    0.000000e+00   0.000000e+00   -1.000000e+02
25%    1.981873e+05   6.542037e+06    2.954108e+01
50%    4.938818e+06   2.324013e+07    2.562817e+02
75%    3.202478e+07   7.059067e+07    4.981126e+03
max    2.362179e+10   2.047553e+10    1.674078e+14
Mean: 88357316.30590421
Standard deviation: 321486625.21422786
Avg_Gap(L_NC): 10811698225.301525
```

Figure 43. Performance metrics and statistical summary of NNLS cubic model.

On the other hand, if the EP curve in Figure 44 is observed, it can also be seen that the cubic OLS model follows more closely the observed losses, while the cubic NNLS model presents a lower accuracy, especially in the longer return periods.

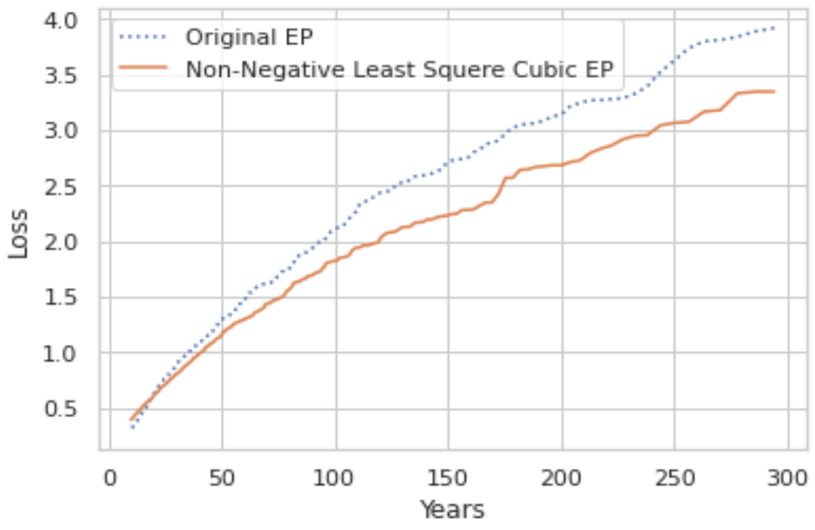


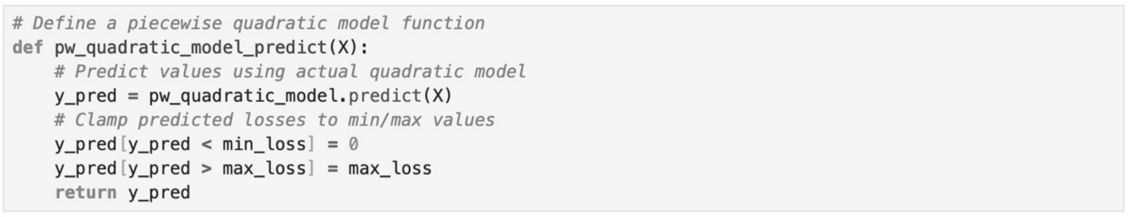
Figure 44. Actual vs NNLS cubic EP curve.

5.2.3 Piecewise Model Performance

In this section, the performance of the piecewise model in predicting economic losses due to seismic events is analyzed. As mentioned above, the piecewise model is employed to capture complex nonlinearities in the relationship between earthquake intensity measurements and economic losses, providing a more accurate and adaptive approximation to extreme variations in the data. In this study, two versions of the piecewise model are implemented and compared: a quadratic and a cubic model, to evaluate which of these approaches offers a better predictive capability.

1. Piecewise Quadratic Regression

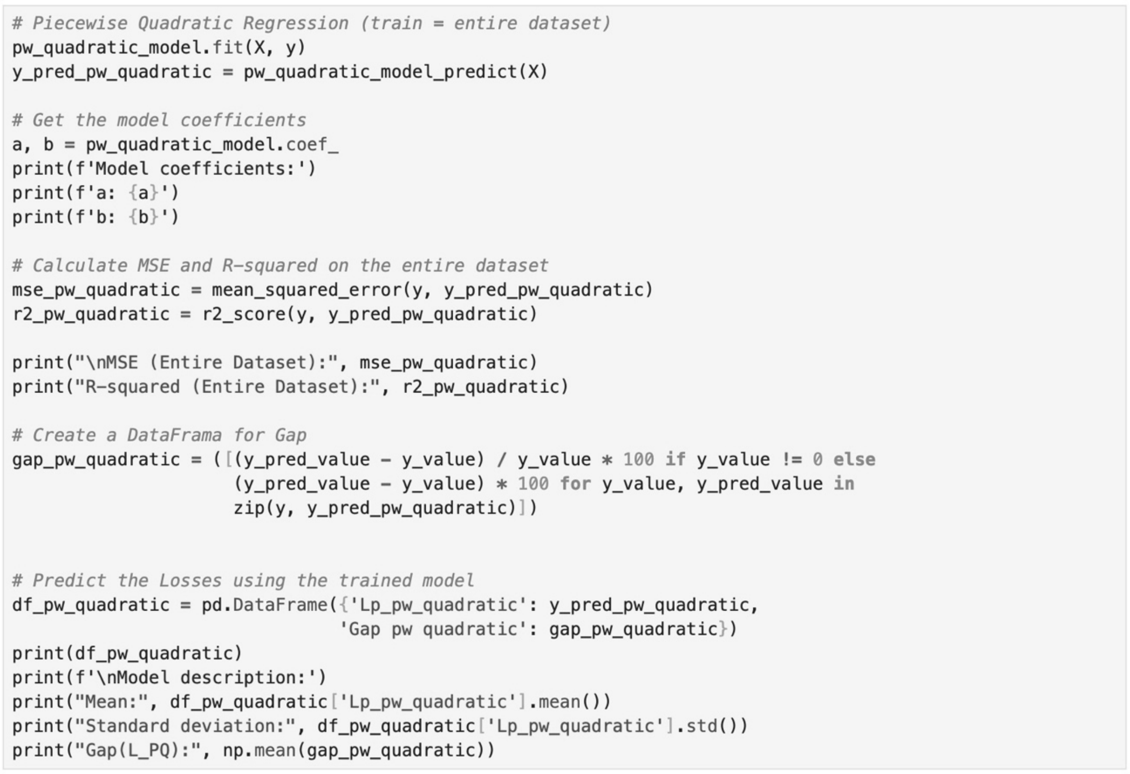
For the implementation of this model, the code in Figure 45 starts with the definition of a prediction function for a piecewise quadratic model. The pw_quadratic_model_predict function is used to predict the economic loss values using the fitted quadratic model, applying minimum and maximum bounds for the predictions, thus ensuring that the predicted losses remain within a realistic range based on the observed losses. The pw_quadratic_model_predict function uses the previously trained quadratic spanwise model (pw_quadratic_model) to make predictions on the X data set. Within this function, y_predict predictions are generated using the fitted quadratic model. Subsequently, a clamping process is applied, which adjusts the predicted losses so that they are neither less than the minimum observed value (min_loss) nor greater than the maximum observed value (max_loss). This ensures that the predictions remain within a realistic range and avoids the generation of implausible predicted values that could disrupt the analysis.



```python
# Define a piecewise quadratic model function
def pw_quadratic_model_predict(X):
    # Predict values using actual quadratic model
    y_pred = pw_quadratic_model.predict(X)
    # Clamp predicted losses to min/max values
    y_pred[y_pred < min_loss] = 0
    y_pred[y_pred > max_loss] = max_loss
    return y_pred
```

Figure 45. Piecewise quadratic model definition.

In addition to the definition and prediction of the piecewise quadratic model, the model is fitted to the entire data set, the coefficients are calculated, and its performance is evaluated in terms of MSE and R-squared. As shown in Figure 46, The quadratic piecewise quadratic model is fitted by a process similar to that followed in the quadratic and cubic models. First, the model is trained with the predictor variables X and the dependent variable y. Then, predictions are made by fitting the values within a realistic range and coefficients are extracted to interpret the relationship between the variables and the losses. A DataFrame is also created with the predicted losses and the percentage difference with respect to the real values.



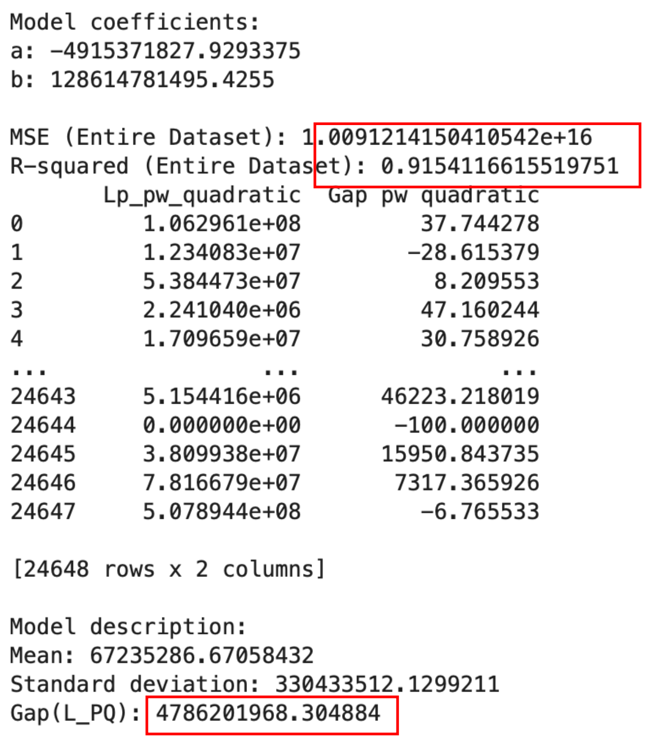
```python
# Piecewise Quadratic Regression (train = entire dataset)
pw_quadratic_model.fit(X, y)
y_pred_pw_quadratic = pw_quadratic_model_predict(X)

# Get the model coefficients
a, b = pw_quadratic_model.coef_
print(f'Model coefficients:')
print(f'a: {a}')
print(f'b: {b}')

# Calculate MSE and R-squared on the entire dataset
mse_pw_quadratic = mean_squared_error(y, y_pred_pw_quadratic)
r2_pw_quadratic = r2_score(y, y_pred_pw_quadratic)

print("\nMSE (Entire Dataset):", mse_pw_quadratic)
print("R-squared (Entire Dataset):", r2_pw_quadratic)

# Create a DataFrama for Gap
gap_pw_quadratic = ([(y_pred_value - y_value) / y_value * 100 if y_value != 0 else
                     (y_pred_value - y_value) * 100 for y_value, y_pred_value in
                     zip(y, y_pred_pw_quadratic)])


# Predict the Losses using the trained model
df_pw_quadratic = pd.DataFrame({'Lp_pw_quadratic': y_pred_pw_quadratic,
                                'Gap pw quadratic': gap_pw_quadratic})
print(df_pw_quadratic)
print(f'\nModel description:')
print("Mean:", df_pw_quadratic['Lp_pw_quadratic'].mean())
print("Standard deviation:", df_pw_quadratic['Lp_pw_quadratic'].std())
print("Gap(L_PQ):", np.mean(gap_pw_quadratic))
```

Figure 46. Performance metrics of piecewise quadratic model.

The output of Figure 47 corresponds to the quadratic piecewise model. When comparing the results of this model with those of the quadratic model and the quadratic NNLS model, it is observed that the quadratic piecewise model presents an R-squared close to the standard quadratic model, indicating a high explanatory capacity of the variability in the observed losses. However, the MSE of the quadratic piecewise model is comparable to that of the quadratic model, but lower than that of the quadratic NNLS model, suggesting better prediction accuracy.

On the other hand, the average gap of the quadratic piecewise model is larger than that of the quadratic model, indicating greater variability in individual predictions. Compared to the quadratic NNLS, which also had a significant average gap, the quadratic piecewise offers intermediate performance, better capturing nonlinear relationships but with greater dispersion in predictions.

In comparison with the quadratic NNLS, the quadratic piecewise model shows a high predictive power similar to the standard quadratic model and improved accuracy, although with a higher variability in individual predictions.



```
Model coefficients:
a: -4915371827.9293375
b: 128614781495.4255

MSE (Entire Dataset): 1.0091214150410542e+16
R-squared (Entire Dataset): 0.9154116615519751
       Lp_pw_quadratic  Gap pw quadratic
0          1.062961e+08         37.744278
1          1.234083e+07        -28.615379
2          5.384473e+07          8.209553
3          2.241040e+06         47.160244
4          1.709659e+07         30.758926
...                 ...               ...
24643      5.154416e+06      46223.218019
24644      0.000000e+00       -100.000000
24645      3.809938e+07      15950.843735
24646      7.816679e+07       7317.365926
24647      5.078944e+08         -6.765533

[24648 rows x 2 columns]

Model description:
Mean: 67235286.67058432
Standard deviation: 330433512.1299211
Gap(L_PQ): 4786201968.304884
```

Figure 47. Performance metrics and statistical summary of piecewise quadratic model.

As for the return period curves, the quadratic piecewise model closely follows the observed losses, Figure 48, although with a slightly lower accuracy than the standard quadratic model. The EP curve of the quadratic OLS model shows a more accurate fit over all periods, while the quadratic NNLS model underestimates the losses, especially at the longer return periods. The quadratic piecewise model, on the other hand, provides a balanced approximation, capturing the general trends but with some underestimation in the longer return periods.

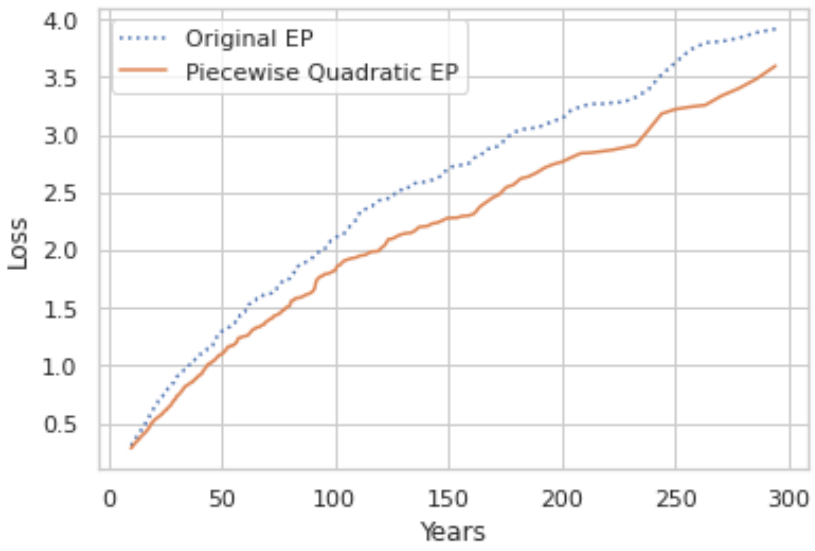
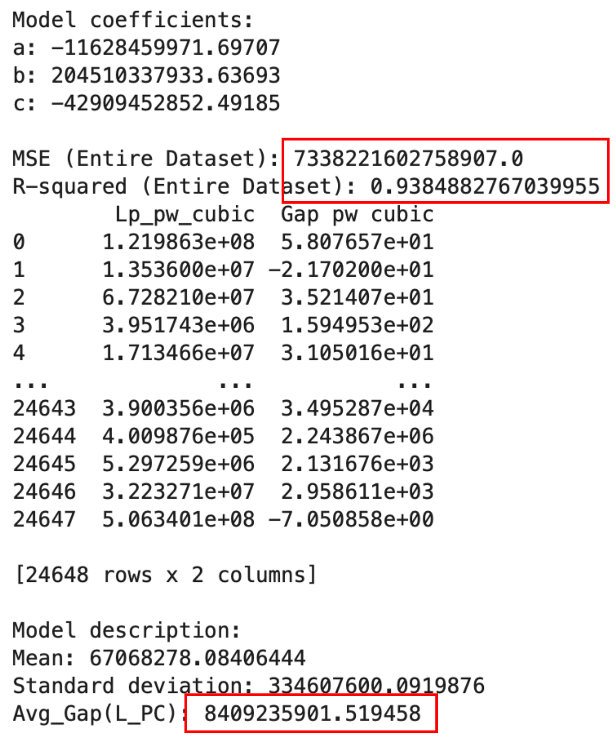


Figure 48. Actual vs piecewise quadratic EP curve.

2. Piecewise Cubic Regression

In this subsection, the performance of the cubic piecewise model is examined, following a process close to that of the quadratic piecewise model. The only significant difference is the inclusion of the cubed WIM term in the model, therefore, only the results obtained will be reviewed.

The results of the cubic piecewise model, represented in Figure 49, show an R-squared of 0.938, close to the cubic OLS model, indicating that both models have a high explanatory power. However, the MSE of the cubic piecewise model is comparable to that of the cubic OLS model, suggesting similar prediction accuracy. The average gap of the cubic piecewise model is larger than that of the cubic OLS model, indicating greater variability in individual predictions. Despite this, the inclusion of the cubic term allows capturing more complex relationships, which may be advantageous in certain contexts.



```
Model coefficients:
a: -11628459971.69707
b: 204510337933.63693
c: -42909452852.49185

MSE (Entire Dataset): 7338221602758907.0
R-squared (Entire Dataset): 0.9384882767039955
        Lp_pw_cubic  Gap pw cubic
0      1.219863e+08  5.807657e+01
1      1.353600e+07 -2.170200e+01
2      6.728210e+07  3.521407e+01
3      3.951743e+06  1.594953e+02
4      1.713466e+07  3.105016e+01
...             ...           ...
24643  3.900356e+06  3.495287e+04
24644  4.009876e+05  2.243867e+06
24645  5.297259e+06  2.131676e+03
24646  3.223271e+07  2.958611e+03
24647  5.063401e+08 -7.050858e+00

[24648 rows x 2 columns]

Model description:
Mean: 67068278.08406444
Standard deviation: 334607600.0919876
Avg_Gap(L_PC): 8409235901.519458
```

Figure 49. Performance metrics and statistical summary of piecewise cubic model.

In terms of return period curves, the cubic piecewise model fits closely to observed losses, as does the cubic OLS model, although it shows some differences at longer return periods. Compared to the cubic NNLS model, the cubic piecewise provides superior fit and accuracy in loss prediction, avoiding the tendency to underestimate observed in the cubic NNLS.

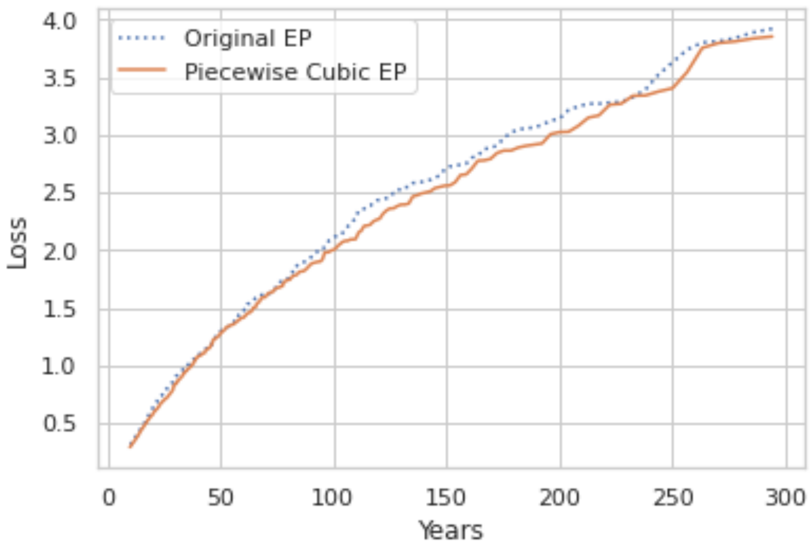
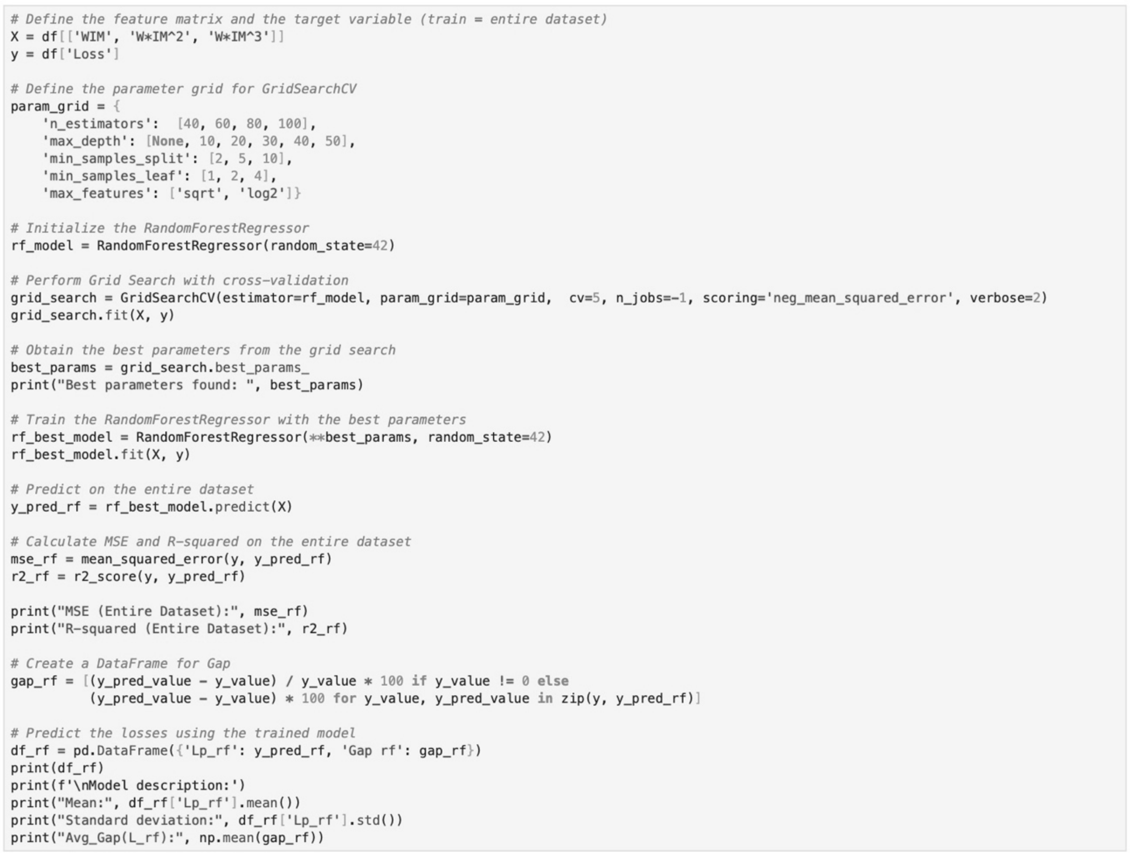


Figure 50. Actual vs piecewise cubic EP curve.

5.2.4 Random Forest Regression Model Performance

In this section, the performance of the Random Forest regression model for predicting economic losses from seismic events is analyzed. Again, the code is quite common to the previous ones already implemented in the different models developed. As shown in Figure 51, the notable differences are that, in this case, a series of hyperparameters is defined in order to optimize them later through the GridSearchCV technique. The hyperparameters considered are: n_estimators, which indicates the number of trees in the forest; max_depth, which is the maximum depth of each tree; min_samples_split, which represents the minimum number of samples required to split a node; min_samples_leaf, which is the minimum number of samples in a leaf; and max_features, which defines the maximum number of features to be considered for the best split.

This search evaluates multiple combinations of hyperparameters to find the best configuration. The best parameters are used to train the model again. Once trained, predictions are made on the entire data set and MSE and R-squared are calculated to assess the accuracy and explanatory performance of the model.

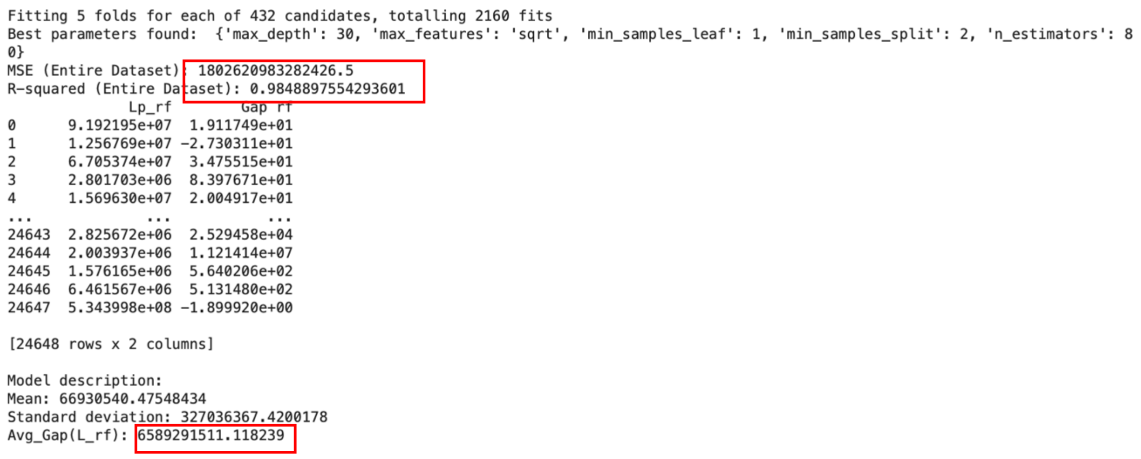


```python
# Define the feature matrix and the target variable (train = entire dataset)
X = df[['WIM', 'W*IM^2', 'W*IM^3']]
y = df['Loss']

# Define the parameter grid for GridSearchCV
param_grid = {
    'n_estimators':  [40, 60, 80, 100],
    'max_depth': [None, 10, 20, 30, 40, 50],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'max_features': ['sqrt', 'log2']}

# Initialize the RandomForestRegressor
rf_model = RandomForestRegressor(random_state=42)

# Perform Grid Search with cross-validation
grid_search = GridSearchCV(estimator=rf_model, param_grid=param_grid,  cv=5, n_jobs=-1, scoring='neg_mean_squared_error', verbose=2)
grid_search.fit(X, y)

# Obtain the best parameters from the grid search
best_params = grid_search.best_params_
print("Best parameters found: ", best_params)

# Train the RandomForestRegressor with the best parameters
rf_best_model = RandomForestRegressor(**best_params, random_state=42)
rf_best_model.fit(X, y)

# Predict on the entire dataset
y_pred_rf = rf_best_model.predict(X)

# Calculate MSE and R-squared on the entire dataset
mse_rf = mean_squared_error(y, y_pred_rf)
r2_rf = r2_score(y, y_pred_rf)

print("MSE (Entire Dataset):", mse_rf)
print("R-squared (Entire Dataset):", r2_rf)

# Create a DataFrame for Gap
gap_rf = [(y_pred_value - y_value) / y_value * 100 if y_value != 0 else
          (y_pred_value - y_value) * 100 for y_value, y_pred_value in zip(y, y_pred_rf)]

# Predict the losses using the trained model
df_rf = pd.DataFrame({'Lp_rf': y_pred_rf, 'Gap rf': gap_rf})
print(df_rf)
print(f'\nModel description:')
print("Mean:", df_rf['Lp_rf'].mean())
print("Standard deviation:", df_rf['Lp_rf'].std())
print("Avg_Gap(L_rf):", np.mean(gap_rf))
```

Figure 51. Performance metrics of Random Forest Regression model.

The results of the RF regression model, represented in Figure 52, show that the best parameters found are max_depth of 30, max_features set to 'sqrt', min_samples_leaf of 1, min_samples_split of 2, and n_estimators of 80. These optimized parameters were determined by a grid search with cross validation, which ensures that the model is efficiently configured for the specific data set.

Most representative of the rest of the output is that the RF model presents both an MSE and an average gap much lower than the previous models and, in turn, presents an R-squared that is well above reaching 0.984, indicating an extremely accurate fit.



```
Fitting 5 folds for each of 432 candidates, totalling 2160 fits
Best parameters found:  {'max_depth': 30, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 8
0}
MSE (Entire Dataset): 1802620983282426.5
R-squared (Entire Dataset): 0.9848897554293601
              Lp_rf        Gap rf
0      9.192195e+07  1.911749e+01
1      1.256769e+07 -2.730311e+01
2      6.705374e+07  3.475515e+01
3      2.801703e+06  8.397671e+01
4      1.569630e+07  2.004917e+01
...             ...           ...
24643  2.825672e+06  2.529458e+04
24644  2.003937e+06  1.121414e+07
24645  1.576165e+06  5.640206e+02
24646  6.461567e+06  5.131480e+02
24647  5.343998e+08 -1.899920e+00

[24648 rows x 2 columns]

Model description:
Mean: 66930540.47548434
Standard deviation: 327036367.4200178
Avg_Gap(L_rf): 6589291511.118239
```

Figure 52. Performance metrics and statistical summary of random forest regression model.

For this model, the EP curve is also obtained and is plotted in Figure 53. In general, the RF curve closely follows the Original EP curve, indicating a good model fit. At the shortest return periods, the model predictions are closely aligned with observed losses, suggesting that the RF model is effective in predicting losses in more frequent events. As return periods increase, the differences between the curves remain minimal, although small deviations can be observed.

These observations confirm that the RF model not only offers a high R-squared and low MSE, but also maintains its accuracy and consistency over different time horizons.

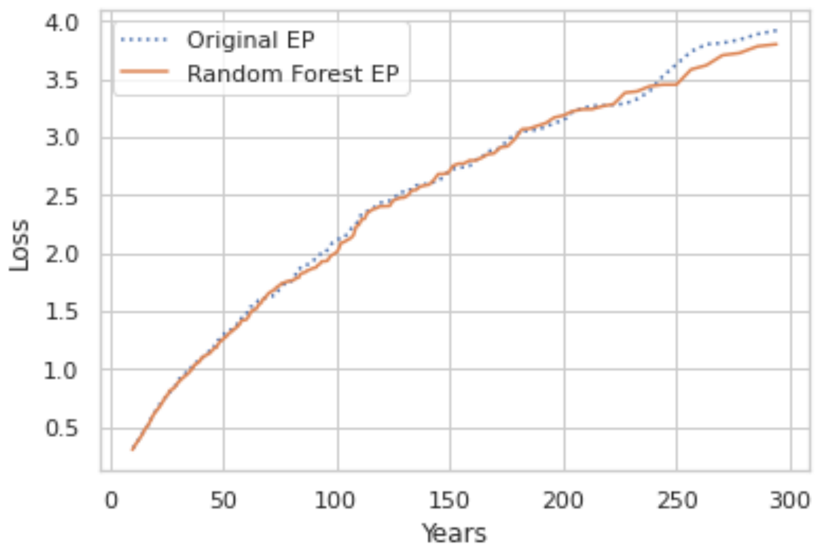
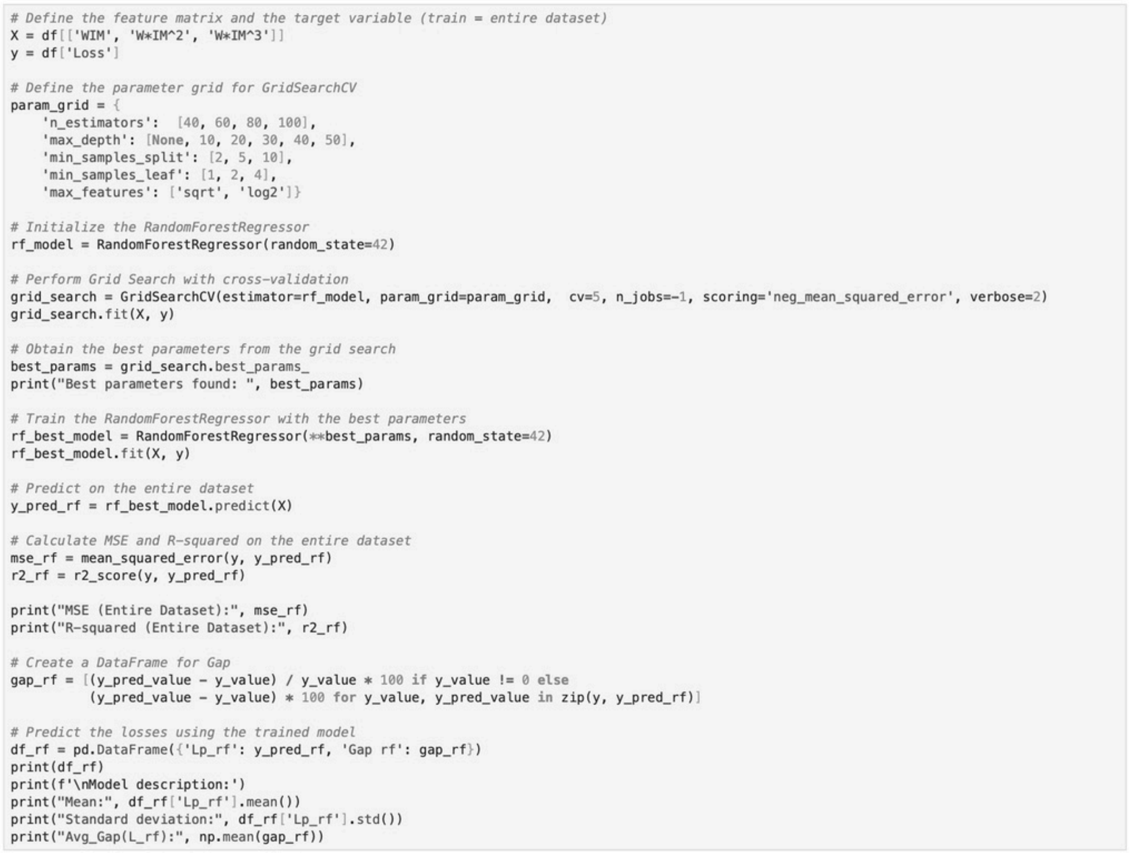


Figure 53. Actual vs random forest EP curve.

5.2.5 *XGboost Regression Model Performance*

As in the previous model, some parameters have been introduced to optimize the XGBoost model, which are represented in Figure 54. In this case these are n_estimators, which is the number of trees in the model, max_depth, which represents the maximum depth of the trees), learning_rate, which is the learning rate, subsample, which consists of a fraction of samples used to train each tree and colsample_bytree, which is a fraction of features used for each tree.

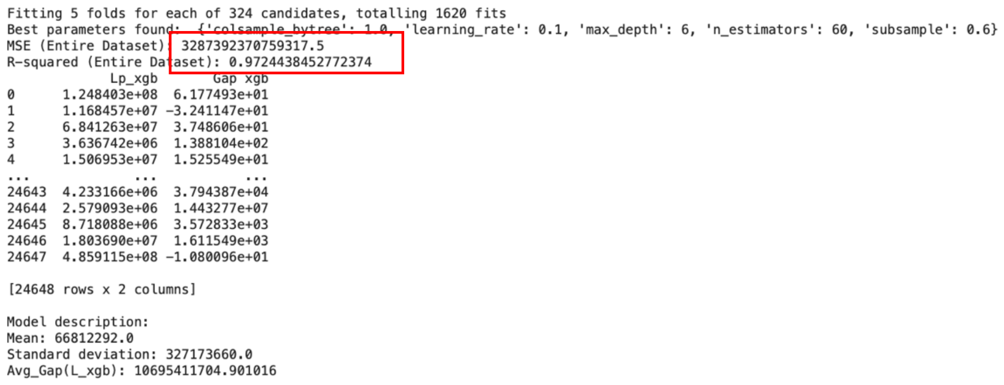
Once the parameters are defined, the XGBRegressor model is initialized, and grid search is performed using GridSearchCV with 5-fold cross-validation. This search evaluates different combinations of parameters to identify the configuration that minimizes the negative mean squared error (neg_mean_squared_error). Once the best parameters are obtained, the XGBoost model is trained with these optimal parameters and the trained model is used to make predictions on the entire dataset.



```
# Define the feature matrix and the target variable (train = entire dataset)
X = df[['WIM', 'W*IM^2', 'W*IM^3']]
y = df['Loss']

# Define the parameter grid for GridSearchCV
param_grid = {
    'n_estimators':  [40, 60, 80, 100],
    'max_depth': [None, 10, 20, 30, 40, 50],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'max_features': ['sqrt', 'log2']}

# Initialize the RandomForestRegressor
rf_model = RandomForestRegressor(random_state=42)

# Perform Grid Search with cross-validation
grid_search = GridSearchCV(estimator=rf_model, param_grid=param_grid,  cv=5, n_jobs=-1, scoring='neg_mean_squared_error', verbose=2)
grid_search.fit(X, y)

# Obtain the best parameters from the grid search
best_params = grid_search.best_params_
print("Best parameters found: ", best_params)

# Train the RandomForestRegressor with the best parameters
rf_best_model = RandomForestRegressor(**best_params, random_state=42)
rf_best_model.fit(X, y)

# Predict on the entire dataset
y_pred_rf = rf_best_model.predict(X)

# Calculate MSE and R-squared on the entire dataset
mse_rf = mean_squared_error(y, y_pred_rf)
r2_rf = r2_score(y, y_pred_rf)

print("MSE (Entire Dataset):", mse_rf)
print("R-squared (Entire Dataset):", r2_rf)

# Create a DataFrame for Gap
gap_rf = [(y_pred_value - y_value) / y_value * 100 if y_value != 0 else
          (y_pred_value - y_value) * 100 for y_value, y_pred_value in zip(y, y_pred_rf)]

# Predict the losses using the trained model
df_rf = pd.DataFrame({'Lp_rf': y_pred_rf, 'Gap rf': gap_rf})
print(df_rf)
print(f'\nModel description:')
print("Mean:", df_rf['Lp_rf'].mean())
print("Standard deviation:", df_rf['Lp_rf'].std())
print("Avg_Gap(L_rf):", np.mean(gap_rf))
```

Figure 54. Performance metrics of XGBoost Regression model.

The XGBoost model has been fine-tuned using GridSearchCV, which identified the optimal parameters as follows, represented in Figure 55: colsample_bytree of 1.0, learning_rate of 0.1, max_depth of 6, n_estimators of 60, and subsample of 0.6. These parameters were chosen because they provided the best performance on the training data during cross-validation. As the RF, XGBoost shows a very high performance with an R-squared of 0.97 and a lower MSE than the OLS and NNLS models.



```
Fitting 5 folds for each of 324 candidates, totalling 1620 fits
Best parameters found:  {'colsample_bytree': 1.0, 'learning_rate': 0.1, 'max_depth': 6, 'n_estimators': 60, 'subsample': 0.6}
MSE (Entire Dataset): 3287392370759317.5
R-squared (Entire Dataset): 0.9724438452772374
             Lp_xgb       Gap xgb
0      1.248403e+08  6.177493e+01
1      1.168457e+07 -3.241147e+01
2      6.841263e+07  3.748606e+01
3      3.636742e+06  1.388104e+02
4      1.506953e+07  1.525549e+01
...             ...           ...
24643  4.233166e+06  3.794387e+04
24644  2.579093e+06  1.443277e+07
24645  8.718088e+06  3.572833e+03
24646  1.803690e+07  1.611549e+03
24647  4.859115e+08 -1.080096e+01

[24648 rows x 2 columns]

Model description:
Mean: 66812292.0
Standard deviation: 327173660.0
Avg_Gap(L_xgb): 10695411704.901016
```

Figure 55. Performance metrics and statistical summary of XGBoost regression model.

Likewise, it can be seen in Figure 56 that in the shorter return periods, the XGBoost predictions are closely aligned with the observed losses, demonstrating its effectiveness in predicting more frequent events. As return periods are extended, the XGBoost curve maintains remarkable proximity to the original curve, suggesting that the model handles both frequent and rare events well.

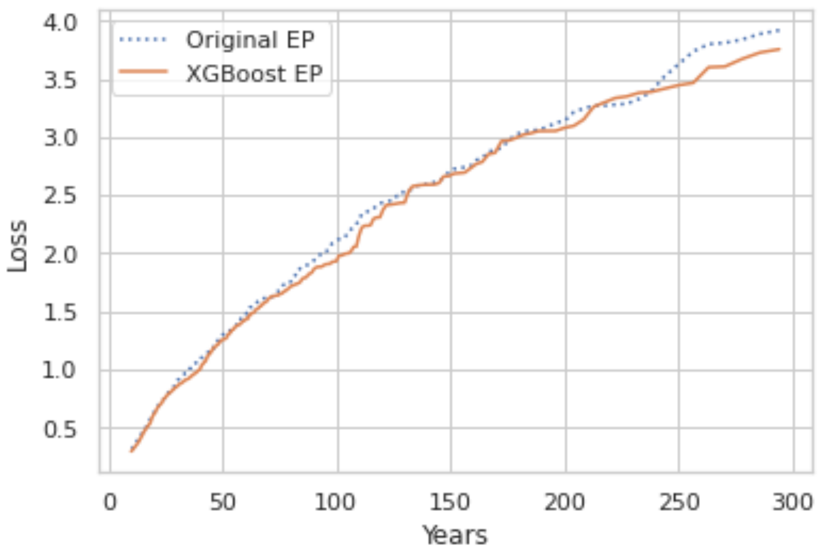


Figure 56. Actual vs XGBoost EP curve.

5.3 Validation Models Performance

This section will evaluate the results obtained from the application of different validation techniques previously described: K-Fold Cross Validation, bootstrapping and split validation. The main objective of this evaluation is to compare the effectiveness of each technique in estimating model performance, providing a complete view on the robustness and reliability of the predictions generated. Through the analysis of these techniques, the aim is to identify which of them offers a better approximation to the real performance of the model on unseen data, thus facilitating the choice of the most appropriate validation method for the specific context of the study.

For clarity and conciseness, only the codes implemented for one model, for example random forest, will be presented, since the codes are quite similar and vary only in specific libraries or hyperparameters.

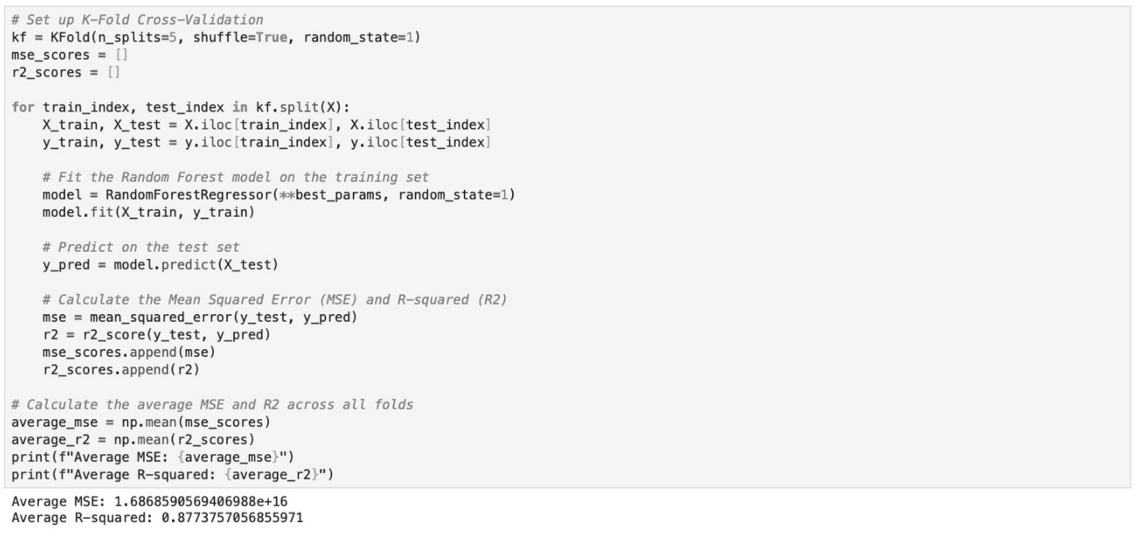
1. K-Fold Cross Validation Performance

The code depicted in Figure 57 implements the K-Fold Cross Validation technique to evaluate a RandomForestRegressor model's performance. It begins by importing the necessary KFold class from sklearn.model_selection for setting up cross-validation. The setup for K-Fold Cross Validation is defined with five splits (n_splits=5), and the data is shuffled before splitting (shuffle=True) to ensure each fold is a representative sample of the dataset, with random_state=1 ensuring the shuffling process is consistent across different runs.

An empty list, mse_scores, is initialized to store the Mean Squared Error (MSE) values from each fold, and another list, r2_scores, to store the R-squared values. In the loop that iterates over training and testing indices generated by kf.split(X), the dataset is partitioned into training (X_train, y_train) and testing (X_test, y_test) subsets according to the current fold's indices.

Within this loop, a RandomForestRegressor model is instantiated with predetermined parameters and a fixed random state for reproducibility. The model is trained using the fit method on the X_train and y_train subsets. Predictions are then made on the X_test subset using the predict method, and the predictions are stored in y_pred.

Following the prediction, the MSE is computed between y_pred and the actual test values y_test, and this value is appended to mse_scores. Similarly, the R-squared value is computed and added to r2_scores. After completing all iterations, the average MSE and average R-squared values are calculated across all folds, providing an assessment of the models' performance across multiple subsets of the dataset. The average R-squared value is about 0.8738, indicating that approximately 87.38% of the variance in the dependent variable is predictable from the independent variables. This is a strong score, suggesting that despite the large MSE, the model does quite well in terms of explaining the variability of the response data around its mean.



```python
# Set up K-Fold Cross-Validation
kf = KFold(n_splits=5, shuffle=True, random_state=1)
mse_scores = []
r2_scores = []

for train_index, test_index in kf.split(X):
    X_train, X_test = X.iloc[train_index], X.iloc[test_index]
    y_train, y_test = y.iloc[train_index], y.iloc[test_index]

    # Fit the Random Forest model on the training set
    model = RandomForestRegressor(**best_params, random_state=1)
    model.fit(X_train, y_train)

    # Predict on the test set
    y_pred = model.predict(X_test)

    # Calculate the Mean Squared Error (MSE) and R-squared (R2)
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)
    mse_scores.append(mse)
    r2_scores.append(r2)

# Calculate the average MSE and R2 across all folds
average_mse = np.mean(mse_scores)
average_r2 = np.mean(r2_scores)
print(f"Average MSE: {average_mse}")
print(f"Average R-squared: {average_r2}")
```

```
Average MSE: 1.6868590569406988e+16
Average R-squared: 0.8773757056855971
```

Figure 57. K-Fold Cross Validation set up.

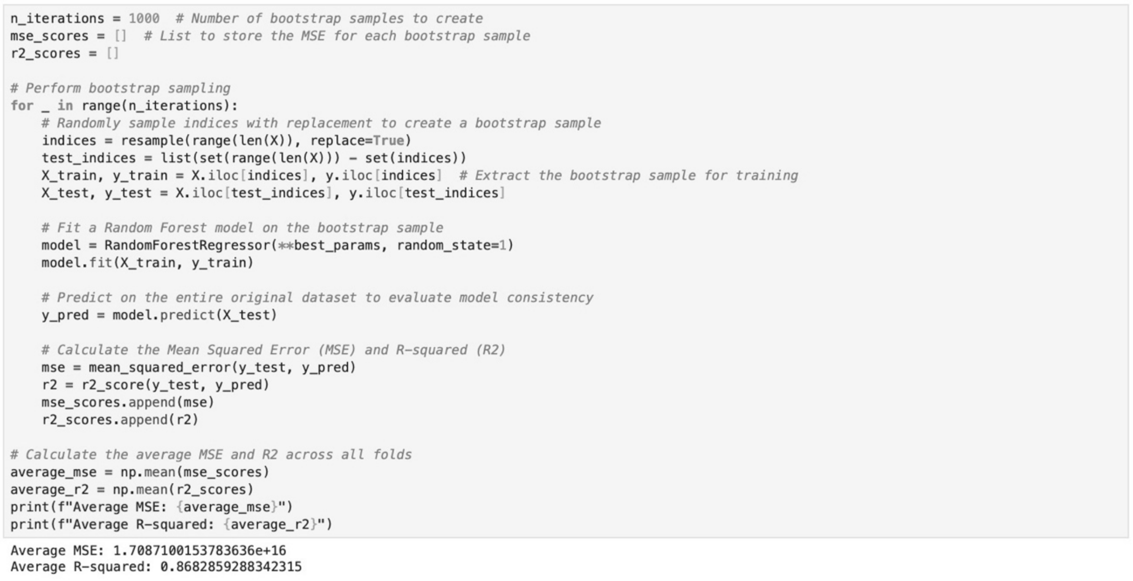
2. Bootstrapping Performance

The code in Figure 58 shows the bootstrapping validation method. In this code, the process begins with setting the number of bootstrap iterations to 1000. For each iteration, indices are randomly sampled to create a bootstrap sample of the original dataset. This sampling allows each iteration to potentially include multiple copies of the same data point, mimicking the effects of sampling from an underlying population. The selected indices determine the training and testing datasets for that bootstrap iteration.

The RandomForestRegressor model is then initialized with specific parameters and a fixed random state to ensure that the results are consistent across different runs. It is trained on the bootstrap sample and subsequently makes predictions on the corresponding test subset. This helps to evaluate the models' performance in terms of prediction accuracy and generalizability.

The bootstrapping results yield an average MSE of approximately 1.788 billion and an R-squared of about 0.8683. Compared to the previous K-Fold Cross Validation results, which showed an MSE of about 1.668 billion and an R- squared of 0.8738, the bootstrapping approach exhibits slightly higher MSE and slightly lower R-squared.

These differences suggest that while the model remains robust and effective in predicting outcomes, the variability introduced by the random sampling with replacement in bootstrapping may lead to marginally less precise predictions than those observed with K-Fold Cross Validation. This implies that the model's performance can slightly fluctuate depending on the sampling technique used, with bootstrapping potentially capturing more variability and real-world uncertainties in the dataset.



```
n_iterations = 1000  # Number of bootstrap samples to create
mse_scores = []  # List to store the MSE for each bootstrap sample
r2_scores = []

# Perform bootstrap sampling
for _ in range(n_iterations):
    # Randomly sample indices with replacement to create a bootstrap sample
    indices = resample(range(len(X)), replace=True)
    test_indices = list(set(range(len(X))) - set(indices))
    X_train, y_train = X.iloc[indices], y.iloc[indices]  # Extract the bootstrap sample for training
    X_test, y_test = X.iloc[test_indices], y.iloc[test_indices]

    # Fit a Random Forest model on the bootstrap sample
    model = RandomForestRegressor(**best_params, random_state=1)
    model.fit(X_train, y_train)

    # Predict on the entire original dataset to evaluate model consistency
    y_pred = model.predict(X_test)

    # Calculate the Mean Squared Error (MSE) and R-squared (R2)
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)
    mse_scores.append(mse)
    r2_scores.append(r2)

# Calculate the average MSE and R2 across all folds
average_mse = np.mean(mse_scores)
average_r2 = np.mean(r2_scores)
print(f"Average MSE: {average_mse}")
print(f"Average R-squared: {average_r2}")
```

```
Average MSE: 1.7087100153783636e+16
Average R-squared: 0.8682859288342315
```

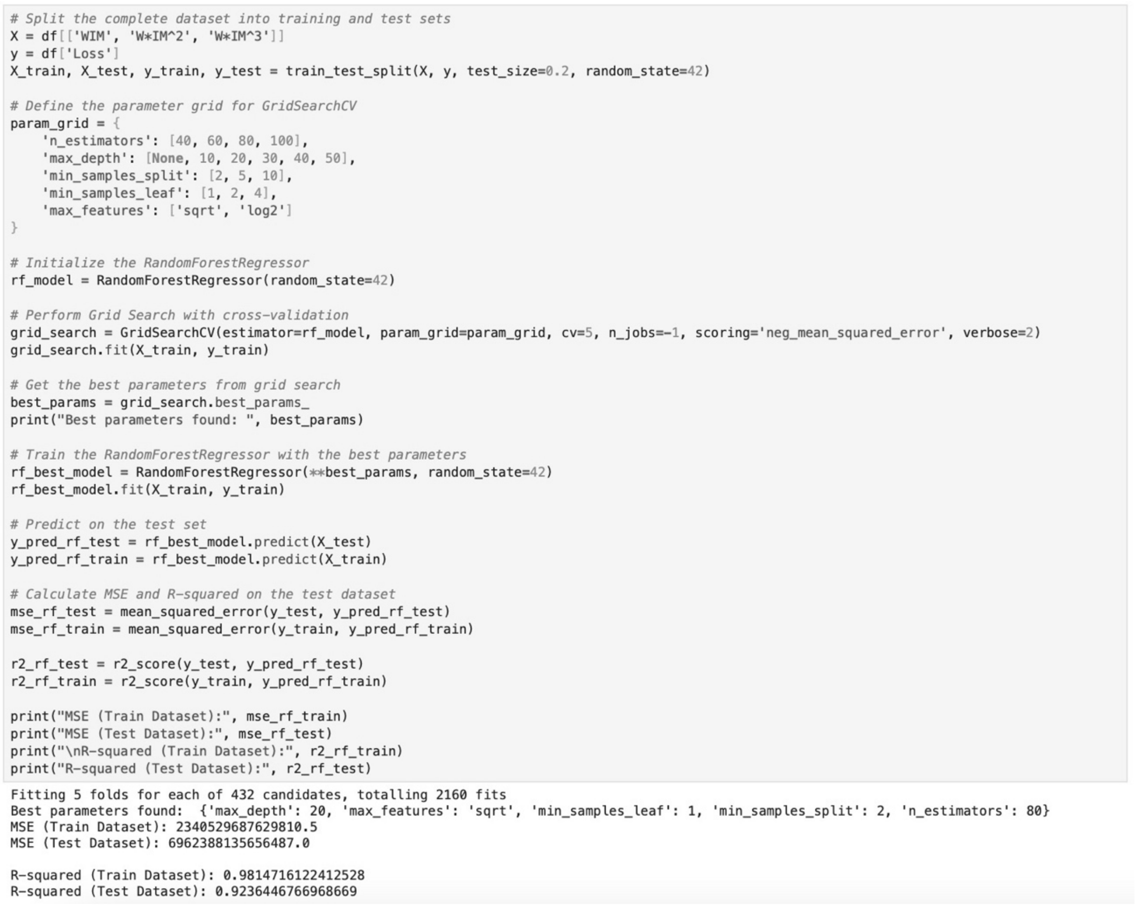
Figure 58. Bootstrapping set up.

3. Split Validation Performance

The code in Figure 59 shows a detailed approach to optimize and evaluate a RandomForestRegressor using Pythons' scikit-learn library. Initially, the dataset is prepared by splitting it into features (X) with 'WIM' and target (y) with 'Loss', followed by splitting into an 80% training set and a 20% test set using a random state to ensure reproducibility. A parameter grid is set for GridSearchCV to find the best settings for the RandomForestRegressor, focusing on key hyperparameters such as n_estimators, max_depth, min_samples_split, min_samples_leaf and max_features. GridSearchCV then performs an exhaustive search for these parameters using 5-fold cross-validation, with the goal of minimizing the negative mean square error.

Once the optimal parameters are identified, the RandomForestRegressor is retrained and used to make predictions on the training and test data sets. Model performance is quantified by an MSE of approximately 2.34 trillion for training and 696 billion for testing, with R-squared values of 0.981 and 0.926 respectively.

Compared to previous models using bootstrapping and K-Fold cross-validation, this GridSearchCV approach potentially offers a more robust and fine-tuned model. This is evidenced by higher R-squared values, indicating higher predictive accuracy and better fit to the data. This model, by systematically exploring a range of parameter settings and validating performance on multiple subsets of data, likely ensures better generalization to unobserved data and provides a more reliable and well-optimized forecasting tool than simpler models or those with less rigorous parameter tuning.



```python
# Split the complete dataset into training and test sets
X = df[['WIM', 'W*IM^2', 'W*IM^3']]
y = df['Loss']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Define the parameter grid for GridSearchCV
param_grid = {
    'n_estimators': [40, 60, 80, 100],
    'max_depth': [None, 10, 20, 30, 40, 50],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'max_features': ['sqrt', 'log2']
}

# Initialize the RandomForestRegressor
rf_model = RandomForestRegressor(random_state=42)

# Perform Grid Search with cross-validation
grid_search = GridSearchCV(estimator=rf_model, param_grid=param_grid, cv=5, n_jobs=-1, scoring='neg_mean_squared_error', verbose=2)
grid_search.fit(X_train, y_train)

# Get the best parameters from grid search
best_params = grid_search.best_params_
print("Best parameters found: ", best_params)

# Train the RandomForestRegressor with the best parameters
rf_best_model = RandomForestRegressor(**best_params, random_state=42)
rf_best_model.fit(X_train, y_train)

# Predict on the test set
y_pred_rf_test = rf_best_model.predict(X_test)
y_pred_rf_train = rf_best_model.predict(X_train)

# Calculate MSE and R-squared on the test dataset
mse_rf_test = mean_squared_error(y_test, y_pred_rf_test)
mse_rf_train = mean_squared_error(y_train, y_pred_rf_train)

r2_rf_test = r2_score(y_test, y_pred_rf_test)
r2_rf_train = r2_score(y_train, y_pred_rf_train)

print("MSE (Train Dataset):", mse_rf_train)
print("MSE (Test Dataset):", mse_rf_test)
print("\nR-squared (Train Dataset):", r2_rf_train)
print("R-squared (Test Dataset):", r2_rf_test)
```

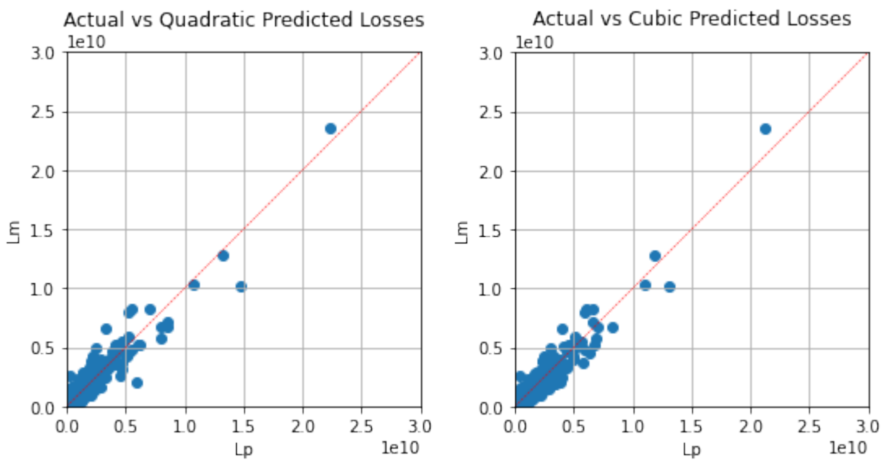
```
Fitting 5 folds for each of 432 candidates, totalling 2160 fits
Best parameters found:  {'max_depth': 20, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 80}
MSE (Train Dataset): 2340529687629810.5
MSE (Test Dataset): 6962388135656487.0

R-squared (Train Dataset): 0.9814716122412528
R-squared (Test Dataset): 0.9236446766968669
```

Figure 59. Split validation set up.

4. Split Validation Performance

Another method for evaluating the performance of a model, particularly in regression tasks such as those discussed above, is by examining residual plots. Residuals, which represent the discrepancies between the observed values of the target variable and the model's predictions, provide significant information about the accuracy and efficiency of a model. A residual plot facilitates the visualization of a models' prediction accuracy across the entire observed data spectrum. Ideally, the residuals are randomly distributed around the horizontal (zero) axis, indicating that the models' predictions are generally accurate over the entire data set. If the residuals show systematic patterns or deviations from this random distribution, they could suggest problems such as nonlinearity, the influence of outliers, or other underlying characteristics of the data that the model may not fully address.

Figure 60 shows the residual plots corresponding to the split validation model of all the models implemented in the study. The figure shows a series of graphs comparing the predicted values with the actual values of several models, each representing a different model or model configuration. The diagonal red line in each graph symbolizes perfect prediction accuracy, where the predicted values exactly match the actual values. In examining the graphs, it is evident that most of the data points tend to cluster near this diagonal line, indicating a general trend of good model performance. However, the degree of clustering and the spread of data points vary from plot to plot, indicating differences in the prediction of the data by each model. The difference is notable in the last row of plots where the RF and XGBoost models are perceived to have better performance.



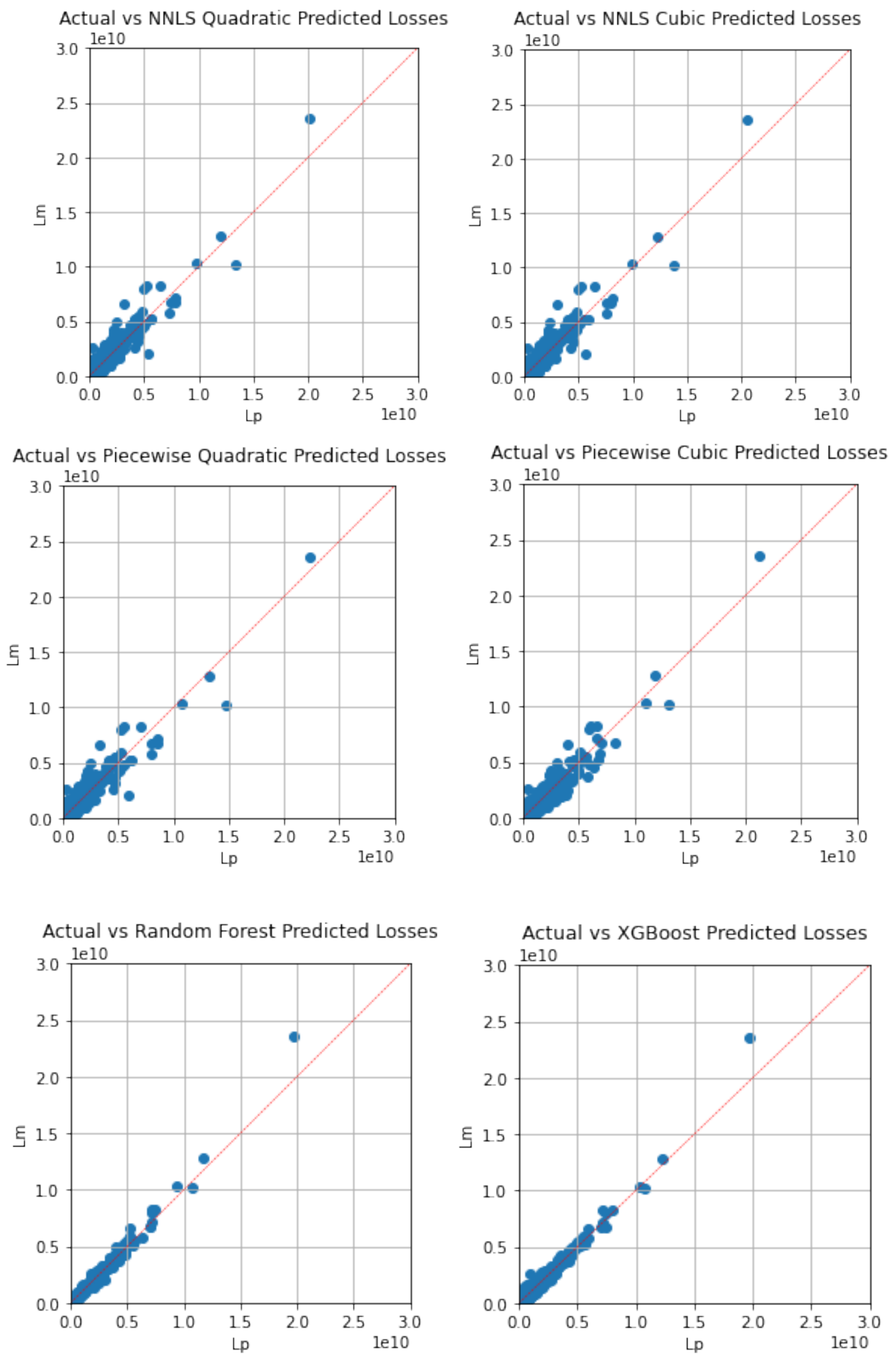


Figure 60. Residual plots.

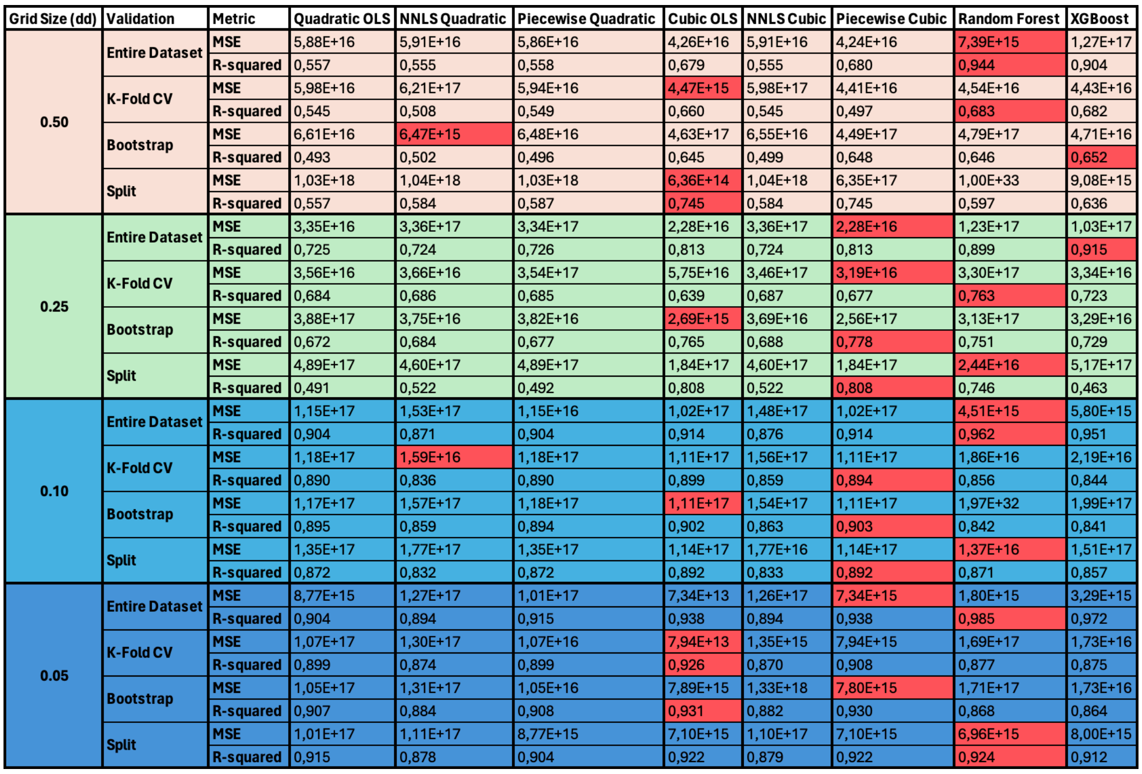
5.4 Discussion of the Results

The comprehensive evaluation of several predictive models across different grid sizes and validation techniques has provided significant insights into their performance, particularly concerning Mean Squared Error and R-squared metrics. This discussion section aims to synthesize these findings, highlighting the models that demonstrated optimal performance in specific settings as indicated by the red cells in the provided table.

The study assessed multiple regression models such as quadratic OLS regression model, cubic OLS regression model, piecewise model, RF and XGboost. These models were tested under varying grid sizes of 0.50, 0.25, 0.10, and 0.05, using validation methods that spanned K-fold Cross-Validation, Bootstrap, and Split techniques. This diverse approach allowed for a mapped analysis of each models' ability to handle different data complexities and partitioning strategies.

Figure 61 presents the final consolidated results of all the models analyzed in the study. It is important to note that the most outstanding values in terms of predictive efficiency are represented in red, denoting the lowest values of MSE and the highest values of R-squared for each validation method used. This coloring facilitates the rapid identification of the models with the most optimal performance. From the analysis of the figure, as the grid size increases, the accuracy of the models tends to decrease. This suggests that models face greater challenges in capturing and modeling data variability at a reduced level of granularity. Therefore, it can be deduced that the accuracy of predictions is inversely proportional to the grid size used in this study context.

Furthermore, it is observed that the model that consistently shows the highest accuracy is the Random Forest model. The Random Forest model, known for its ability to handle large volumes of data and capture nonlinear complexities without overfitting, proved to be particularly robust across multiple validation configurations and grid sizes.



Grid Size (dd)	Validation	Metric	Quadratic OLS	NNLS Quadratic	Piecewise Quadratic	Cubic OLS	NNLS Cubic	Piecewise Cubic	Random Forest	XGBoost
0.50	Entire Dataset	MSE	5,88E+16	5,91E+16	5,86E+16	4,26E+16	5,91E+16	4,24E+16	7,39E+15	1,27E+17
		R-squared	0,557	0,555	0,558	0,679	0,555	0,680	0,944	0,904
	K-Fold CV	MSE	5,98E+16	6,21E+17	5,94E+16	4,47E+15	5,98E+17	4,41E+16	4,54E+16	4,43E+16
		R-squared	0,545	0,508	0,549	0,660	0,545	0,497	0,683	0,682
	Bootstrap	MSE	6,61E+16	6,47E+15	6,48E+16	4,63E+17	6,55E+16	4,49E+17	4,79E+17	4,71E+16
		R-squared	0,493	0,502	0,496	0,645	0,499	0,648	0,646	0,652
	Split	MSE	1,03E+18	1,04E+18	1,03E+18	6,36E+14	1,04E+18	6,35E+17	1,00E+33	9,08E+15
		R-squared	0,557	0,584	0,587	0,745	0,584	0,745	0,597	0,636
0.25	Entire Dataset	MSE	3,35E+16	3,36E+17	3,34E+17	2,28E+16	3,36E+17	2,28E+16	1,23E+17	1,03E+17
		R-squared	0,725	0,724	0,726	0,813	0,724	0,813	0,899	0,915
	K-Fold CV	MSE	3,56E+16	3,66E+16	3,54E+17	5,75E+16	3,46E+17	3,19E+16	3,30E+17	3,34E+16
		R-squared	0,684	0,686	0,685	0,639	0,687	0,677	0,763	0,723
	Bootstrap	MSE	3,88E+17	3,75E+16	3,82E+16	2,69E+15	3,69E+16	2,56E+17	3,13E+17	3,29E+16
		R-squared	0,672	0,684	0,677	0,765	0,688	0,778	0,751	0,729
	Split	MSE	4,89E+17	4,60E+17	4,89E+17	1,84E+17	4,60E+17	1,84E+17	2,44E+16	5,17E+17
		R-squared	0,491	0,522	0,492	0,808	0,522	0,808	0,746	0,463
0.10	Entire Dataset	MSE	1,15E+17	1,53E+17	1,15E+16	1,02E+17	1,48E+17	1,02E+17	4,51E+15	5,80E+15
		R-squared	0,904	0,871	0,904	0,914	0,876	0,914	0,962	0,951
	K-Fold CV	MSE	1,18E+17	1,59E+16	1,18E+17	1,11E+17	1,56E+17	1,11E+17	1,86E+16	2,19E+16
		R-squared	0,890	0,836	0,890	0,899	0,859	0,894	0,856	0,844
	Bootstrap	MSE	1,17E+17	1,57E+17	1,18E+17	1,11E+17	1,54E+17	1,11E+17	1,97E+32	1,99E+17
		R-squared	0,895	0,859	0,894	0,902	0,863	0,903	0,842	0,841
	Split	MSE	1,35E+17	1,77E+17	1,35E+17	1,14E+17	1,77E+16	1,14E+17	1,37E+16	1,51E+17
		R-squared	0,872	0,832	0,872	0,892	0,833	0,892	0,871	0,857
0.05	Entire Dataset	MSE	8,77E+15	1,27E+17	1,01E+17	7,34E+13	1,26E+17	7,34E+15	1,80E+15	3,29E+15
		R-squared	0,904	0,894	0,915	0,938	0,894	0,938	0,985	0,972
	K-Fold CV	MSE	1,07E+17	1,30E+17	1,07E+16	7,94E+13	1,35E+15	7,94E+15	1,69E+17	1,73E+16
		R-squared	0,899	0,874	0,899	0,926	0,870	0,908	0,877	0,875
	Bootstrap	MSE	1,05E+17	1,31E+17	1,05E+16	7,89E+15	1,33E+18	7,80E+15	1,71E+17	1,73E+16
		R-squared	0,907	0,884	0,908	0,931	0,882	0,930	0,868	0,864
	Split	MSE	1,01E+17	1,11E+17	8,77E+15	7,10E+15	1,10E+17	7,10E+15	6,96E+15	8,00E+15
		R-squared	0,915	0,878	0,904	0,922	0,879	0,922	0,924	0,912

Figure 61. Performance comparison of predictive models according to grid size and validation method.

If a more detailed analysis is performed for each grid size, significant conclusions about the performance of the different models can be derived. In deepening the comparative analysis of the different models and validation methods according to the size of the grid, it is important to emphasize that the main parameter for the evaluation is the coefficient of determination. This approach is especially justified in this study since the target variable is economic losses. In contexts where loss data are handled, the MSE may present numerically large and similar values between different models, which complicates direct comparisons and makes differences in performance less evident.

For the grid of size 0.50 represented in Figure 62, it is noticed that the Split validation method generally offers the best results, except for the Random Forest and XGBoost models, where the K-Fold CV method stands out as the most effective. In this context, the Cubic OLS model stands out under the Split validation method as the one with the best performance (74.5%).

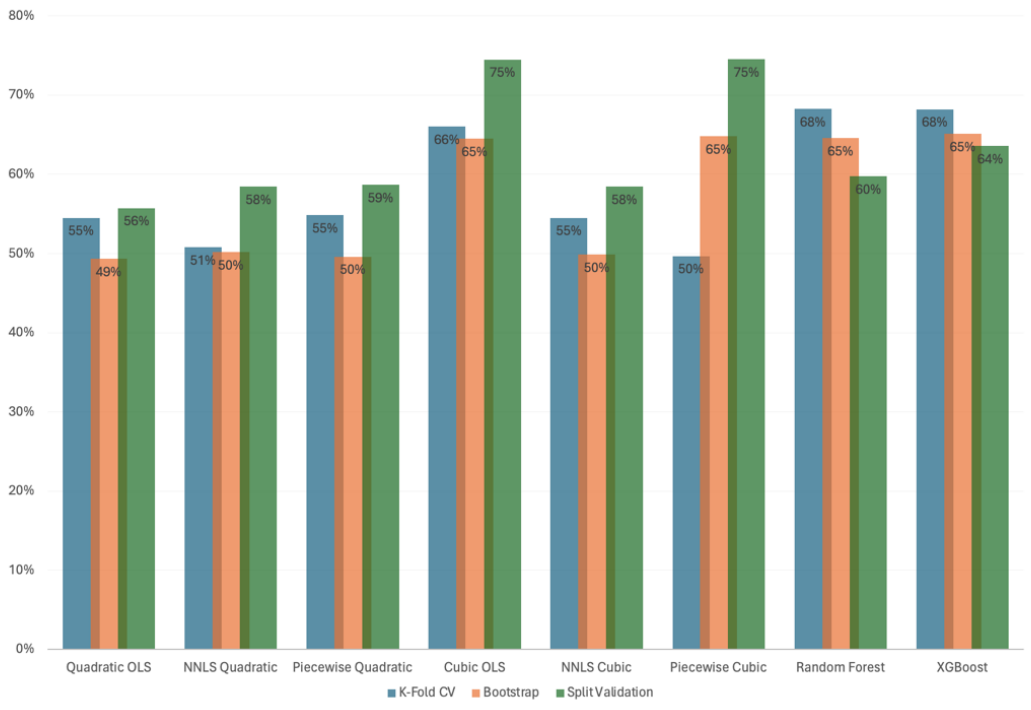


Figure 62. Performance of grid models at grid 0.50dd.

At grid size 0.25, shown in Figure 63, the K-Fold CV technique gains relevance, followed by the Bootstrap method. However, the model that exhibits superior performance is the Piecewise Cubic (80.8%), again under the Split validation method, underlining the consistency of this model in different configurations.

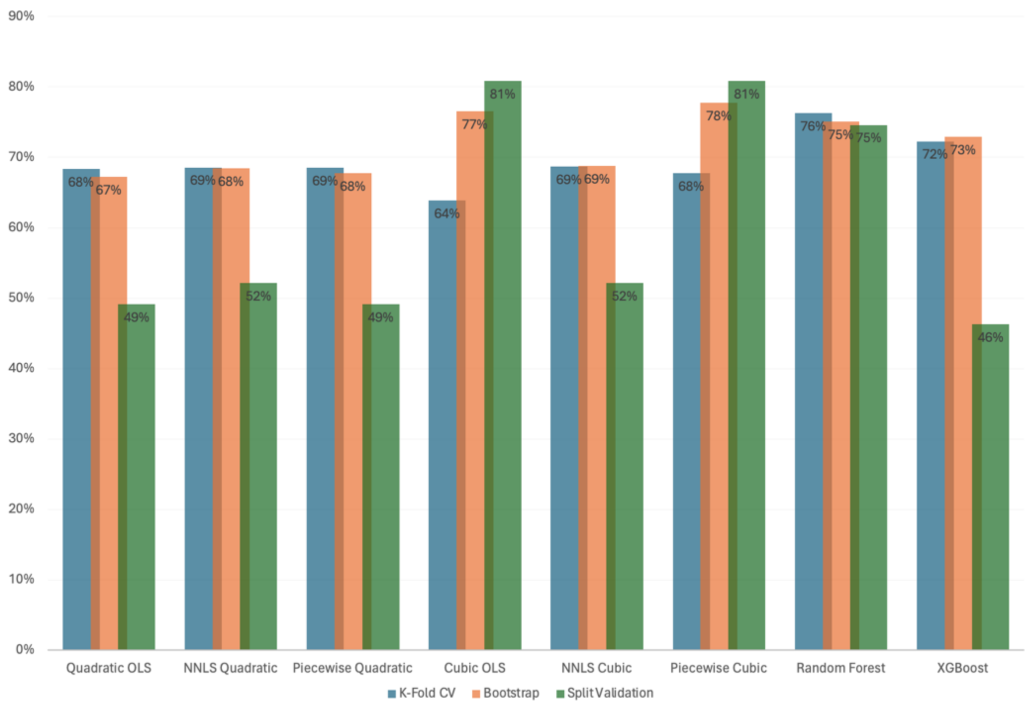


Figure 63. Performance of grid models at grid 0.25dd.

When analyzing the 0.10 grid, in Figure 63, the results between the models are quite competitive, but the Piecewise Cubic model again demonstrates superior performance (89.2%). This is evidence of the robustness of this model in effectively capturing and modeling the variability in the data despite the decrease in grid size.

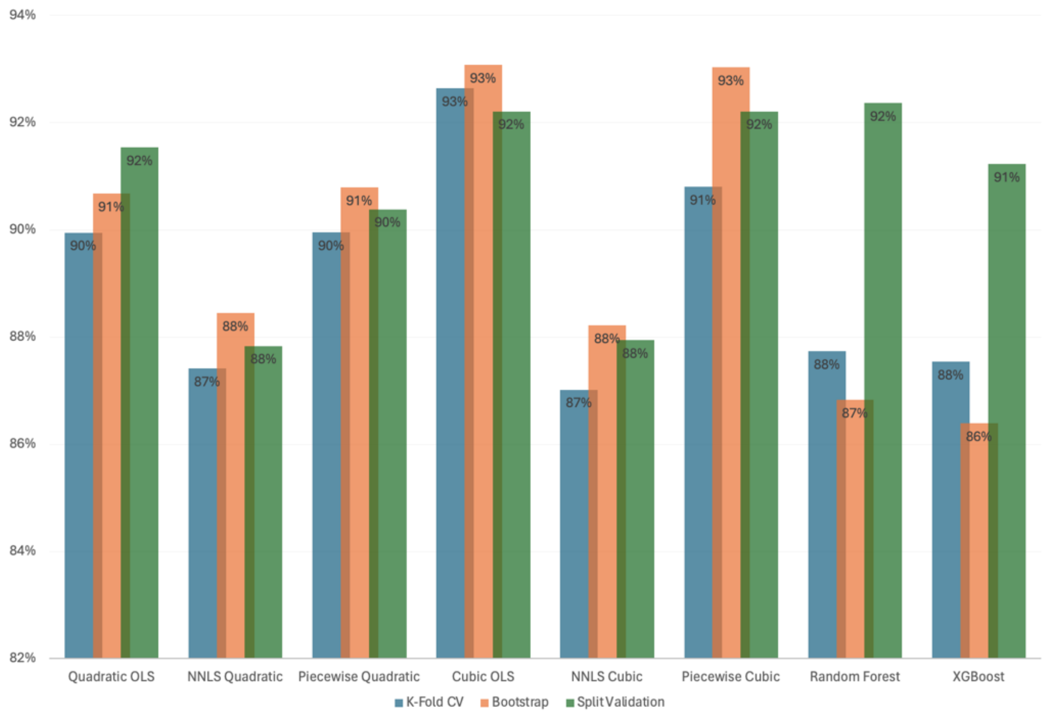


Figure 64. Performance of grid models at grid 0.10dd.

For the smallest grid of 0.05, in Figure 64, a considerable increase in accuracy is observed compared to the larger grids. The models that stand out in this scenario are the Cubic regression model (93.1%) under the Bootstrap validation method and the Random Forest model (92.4%) under the Split method.

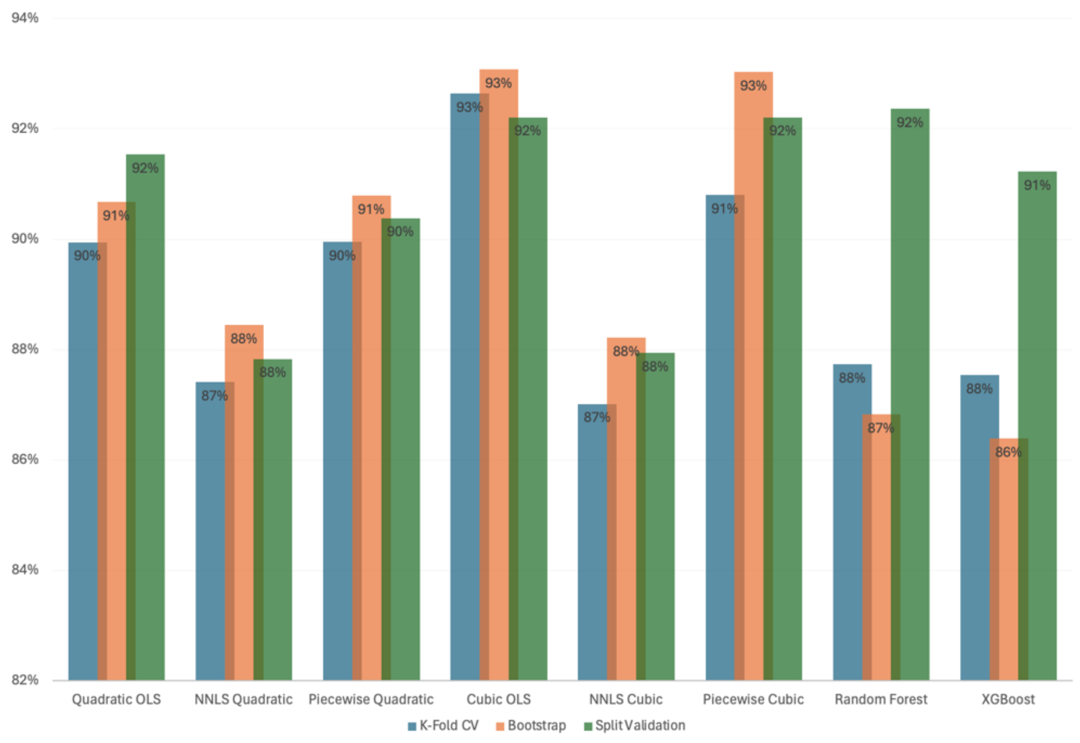


Figure 65. Performance of grid models at grid 0.05dd.

For parametric insurance in Morocco, the cubic model with a grid granularity of 0.10 was selected. This decision was influenced by several strategic and technical considerations. Guy Carpenter chose to avoid including a black box model in the contract, since the lack of transparency in its operation could complicate the explanation and justification of decisions based on its predictions. Instead, the cubic model, being more interpretable and showing outstanding accuracy in the tests, was presented as the most suitable option.

The choice of a granularity of 0.10 for the cubic model was also based on cost-effectiveness considerations. Although a finer grid of 0.05 could theoretically offer greater accuracy, the associated increased costs are not justified by marginal improvements in performance. Therefore, the 0.10 granularity was selected as offering an optimal balance between accuracy and cost, resulting in an efficient and economically viable solution for parametric insurance. This configuration guarantees very good results, providing a robust and reliable model without incurring excessive costs.

6. Conclusions and Future Work

In this project, a detailed investigation into the financial impacts of seismic events in Morocco was undertaken, employing a combination of parametric insurance mechanisms and advanced statistical and machine learning models. The primary accomplishment of this research lies in its methodological innovation, which integrates robust predictive modeling techniques to enhance both accuracy and efficiency in seismic risk management.

The study highlighted several key findings. Initially, simpler regression models such as quadratic and cubic ordinary least squares provided insights into the nonlinear relationships between ground motion intensity and economic losses. However, their limited complexity often failed to capture the intricate dynamics of more significant seismic events. In contrast, piecewise regression models—both quadratic and cubic—demonstrated greater flexibility and effectiveness in modeling variable impacts across different seismic magnitudes. Among the ensemble methods tested, Random Forest and XGBoost models were particularly notable for their robust performance, adeptly handling the complex and nonlinear patterns inherent in seismic data, which simpler models could not adequately address.

These insights are essential for advancing parametric insurance solutions in Morocco, offering a sophisticated framework that merges rapid financial decision-making with comprehensive risk assessment. This integrated approach not only helps mitigate financial risks but also enhances the resilience of communities to future seismic events.

Looking ahead, the methodology developed here lays a solid groundwork for extending the scope of natural disaster risk analysis to encompass other catastrophic events, such as hurricanes. Future research will focus on adapting these models to the unique challenges posed by hurricanes, which differ significantly from seismic risks. This will include adjusting the models to better capture hurricane-specific attributes like wind speed, trajectory, atmospheric pressure changes and so on. Future efforts will also explore the use of more sophisticated machine learning techniques capable of more effectively managing the spatial and temporal dimensions of hurricane data.

These developments aim to not only extend the application of the methodologies established in this thesis but also to contribute substantially to global efforts in managing and mitigating risks associated with diverse natural disasters. The goal is to develop a more comprehensive, adaptable, and resilient framework for disaster risk management and financial mitigation on a global scale.

Bibliography

About pandas. (n.d.). Retrieved 05 2024, from pandas: https://pandas.pydata.org/about/index.html

AON. (2023). *Weather, Climate, and Catastrophe Insight.*

ARTEMIS. (n.d.). *IBRD CAR Mexico 2024*. Retrieved 05 2024, from ARTEMIS: https://www.artemis.bm/deal-directory/ibrd-car-mexico-2024/

Borup, D. C. (2023). Targeting predictors in random forest regression. *Elsevier*, 841-868.

Budiman, F. (2019). SVM-RBF Parameters Testing Optimization Using Cross Validation and Grid Search to Improve Multiclass Classification. *Scientific Visualization*, 80 - 90,.

Centre for Research on the Epidemiology of Disasters, U. O. (2022). *The human cost of disasters: an overview of the last 20 years (2000-2019).*

Chatterjee, S. O. (2006). Nonparametric estimation for quadratic regression. *Elsevier*, 1156-1163.

Data gaps hide the true human impacts of disasters in 2023. (2023). Retrieved 05 2024, from United Nations Office for Disasters Risk Reduction: https://www.undrr.org/explainer/uncounted-costs-of-disasters-2023#:~:text=Regional%20Assessment%20Report%20on%20Disaster,of%20New%20Zealand%20or%20Portugal.

Desmond, A. (2023). *Parametric insurance: How it can benefit construction companies.* Retrieved 05 2024, from Marsh: https://www.marsh.com/en/industries/construction/insights/parametric-insurance-how-it-can-benefit-construction-companies.html

Egbert, J. P. (2021). Bootstrapping Techniques. *Springer*, 593–610.

Erdik, M. Ş. (2011). Rapid earthquake loss assessment after damaging earthquakes. In *Soil Dynamics and Earthquake Engineering* (pp. 247-266). Elsevier.

Erdik, M. (2017). Earthquake risk assessment. *Bulletin of Earthquake Engineering* , 1-38.

Faizi, N. A. (2023). Pearson Correlation. In *Commercial Data Mining* (pp. 109-126). Science Direct.

Franco, G. (2014). Earthquake Mitigation Strategies Through Insurance. *Springer*, 18.

Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *JSTOR*, 1189-1232 .

Ghorbani, E. S. (2024). Enhancing Predictive Models in Insurance: A Feature Selection Analysis. (pp. 1-13). Springer.

Goda, K. (2014). Seismic Risk Management of Insurance Portfolio Using Catastrophe Bonds. In *Special Issue:Risk Assessment of Civil Infrastructures* (pp. 570-582). Computer-Aided Civil and Infrastructure Engineering.

Grossi, P. W. (2005). Sources, Naure, and Impact of Uncertainties on Catastrophe Modeling. In P. K. Grossi, *Catastrophe Modeling: A new approach to managine risk* (pp. 11-38). Springer.

Gu, Z. L. (2023). Modelling economic losses from earthquakes using regression forests: Application to parametric insurance. *Elsevier*, 1-9.

Guy Carpenter. (n.d.). *History*. Retrieved 05 2024, from Guy Carpenter: https://www.guycarp.com/company/history.html

Guy Carpenter. (n.d.). *Post Event: Marrakesh - Safi Earthquake*. Retrieved 05 2024, from Guy Carpenter: https://www.guycarp.com/insights/2023/09/post-event-morocco-earthquake.html

Herman, M. W. (2015). Seismicity of the Earth 1900–2013 Mediterranean Sea and vicinity. *US Geological Survey*.

Huang, Q. M. (2012). An improved grid search algorithm of SVR parameters optimization. *IEEE*, 1-5.

Hunter, D. A. (2022). *Analítica Predictiva 2: Regresión Multivariable con Random Forest.* Retrieved 05 2024, from Quality: https://www.quality.cl/analitica-predictiva-2-regresion-multivariable-con-random-forest/

Introduction to Boosted Trees. (n.d.). Retrieved 05 2024, from dmlc XGBoost: https://xgboost.readthedocs.io/en/stable/tutorials/model.html

Jeon, H. O. (2020). Hybrid-Recursive Feature Elimination for Efficient Feature Selection. *Applied Sciences*, 1-8.

KFold. (n.d.). Retrieved 05 2024, from scikit learn: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html

K-fold cross-validation. (2016). Retrieved from KarlRosean: http://karlrosaen.com/ml/learning-log/2016-06-20/

Kircher, C. A. (2006). HAZUS Earthquake Loss Estimation Methods. *Natural Hazards Review* .

Linardos, V. D. (2022). Machine Learning in Disaster Management: Recent Developments in Methods and Applications. *Machine Learning & Knowledge Extraction*, 1-28.

Lokshtanov, D. S. (2021). Efficient Algorithms for Least Square Piecewise Polynomial Regression. *ESA21: Proceedings of European Symposium on Algorithms*, 1-21.

Mahmood, Z. K. (2009). On the Use of K-Fold Cross-Validation to Choose Cutoff Values and Assess the Performance of Predictive Models in Stepwise Regression. *The International Journal of Biostatistics*.

Marsh. (n.d.). *Parametrics*. Retrieved 05 2024, from Marsh: https://www.marsh.com/en/services/parametrics.html

Nguyen, M. H. (2023). Ensemble XGBoost schemes for improved compressive strength prediction of UHPC. *Elsevier*, 1-17.

Pai, J. L. (2022). Earthquake parametric insurance with Bayesian spatial quantile regression. In *Insurance: Mathematics and Economics* (pp. 1-12). Elsevier.

pandas.DataFrame. (n.d.). Retrieved 05 2024, from pandas: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html

Pucciano, S. F. (2017). Loss Predictive Power of Strong Motion Networks for Usage in Parametric Risk Transfer: Istanbul as a Case Study. *Earthquake Spectra*, 1513-1531.

Quick start guide. (n.d.). Retrieved 05 2024, from matplotlib: https://matplotlib.org/stable/users/explain/quick_start.html#more-reading

R. Guidotti, G. F. (2024). Design of a Hybrid Ground Motion Index and Cat-in-a-grid Parametric Earthquake Trigger for Morocco. *World Conference in Earthquake Engineering*, (p. 12). Milan.

Residual Plot: Definition and Examples. (n.d.). Retrieved 05 2024, from Statistics How To: https://www.statisticshowto.com/residual-plot/

Sahai, R. A.-A.-H.-S. (2023). Insurance Risk Prediction Using Machine Learning. *Springer*, 419–433.

Salgado-Gálvez, M. O.-C. (2023). A Caribbean and Central America Seismic Hazard Model for Sovereign Parametric Insurance Coverage. *Bulletin of the Seismological Society of America*, 1-22.

SciPy User Guide. (n.d.). Retrieved 05 2024, from SciPy: https://docs.scipy.org/doc/scipy/tutorial/index.html

seaborn.boxplot. (n.d.). Retrieved 05 2024, from seaborn: https://seaborn.pydata.org/generated/seaborn.boxplot.html

seaborn.scatterplot. (n.d.). Retrieved 05 2024, from seaborn: https://seaborn.pydata.org/generated/seaborn.scatterplot.html

seaborn: statistical data visualization. (n.d.). Retrieved 05 2024, from seaborn: https://seaborn.pydata.org

Singh, A. T. (2016). A Review of Supervised Machine Learning Algorithms. *IEEE*, 1-6.

Singh, C. R. (2024). Strengthening Resilience: AI and Machine Learning in Emergency Decision-Making for Natural Disasters. *IGI Global*, 249-278.

sklearn.metrics. (n.d.). Retrieved 05 2024, from scikit learn: https://scikit-learn.org/stable/api/sklearn.metrics.html

Slawski, M. H. (2013). Non-negative least squares for high-dimensional linear models: Consistency and sparse recovery without regularization. *Electronic Journal of Statistics*, 3004 - 3056.

The 17 Goals. (n.d.). Retrieved 05 2024, from United Nations: https://sdgs.un.org/goals

train_test_split. (n.d.). Retrieved 05 2024, from scikit learn: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

United Nations. (2023). *The Sustainable Development Goals Report.*

United States Geological Survey. (n.d.). *Earthquake Hazards Program*. Retrieved 05 2024, from M 6.8 - Al Haouz, Morocco: https://earthquake.usgs.gov/earthquakes/eventpage/us7000kufc/region-info

What is NumPy? (n.d.). Retrieved 05 2024, from NumPy: https://numpy.org/doc/stable/user/whatisnumpy.html

Ydav, S. S. (2016). Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification. *IEEE*, 1-6.