



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Dpto. de Estadística e Investigación Operativa
Aplicadas y Calidad

Modelos predictivos para un problema de reparto modal

Trabajo Fin de Máster

Máster Universitario en Ingeniería de Análisis de Datos, Mejora de
Procesos y Toma de Decisiones

AUTOR/A: Palma Mejía, Luis Enrique

Tutor/a: Vallada Regalado, Eva

Cotutor/a: Martínez Minaya, Joaquín

CURSO ACADÉMICO: 2022/2023

Este trabajo se centra en el análisis del modelo de reparto modal, que se utiliza principalmente en el ámbito del transporte, a través de la teoría de la utilidad aleatoria. En particular, se explora el modelo logit multinomial (MNL) para comprender cómo los usuarios eligen los modos de transporte al desplazarse de un lugar a otro, y cuáles son los criterios que tienen en cuenta para tomar estas decisiones.

Con el objetivo de lograr un mejor ajuste, se llevaron a cabo diversas pruebas y transformaciones en las variables explicativas de los modelos. Sin embargo, estas modificaciones permitieron identificar la combinación de criterios que mejor se ajusta a los comportamientos observados en la elección del modo de transporte. Además, para evaluar la capacidad predictiva del modelo *Logit Multinomial*, se compara con los modelos *Random Forest* y *Redes Neuronales*.

La implementación de los modelos Logit se realizó utilizando el lenguaje de programación Python, en particular, se empleó la librería *Biogeme*, que proporciona las herramientas necesarias para conseguir la estimación de los parámetros de estos modelos. Además, se desarrolló una clase de Python llamada "*Logit_model_pred*", que permite obtener las predicciones generadas por los modelos logit, además de una serie de gráficos e informes de otras librerías y de elaboración propia, un gráfico comparativo que muestra el comportamiento predictivo de los diferentes modelos analizados.

En base a los resultados obtenidos y al análisis comparativo, se llega a la conclusión de que el modelo *Logit Multinomial* presenta una capacidad predictiva significativamente menor en comparación con los modelos *Random Forest* y *Redes Neuronales*. Estos últimos modelos, al utilizar enfoques más complejos y flexibles, logran una mayor precisión en la predicción del modo de transporte elegido por los usuarios.

Palabras clave: Reparto modal; teoría de utilidad aleatoria; funciones de utilidad; modelos predictivos.

ABSTRACT

This work focuses on the analysis of the modal split model, which is primarily used in the field of transportation, through the theory of random utility. In particular, we explore the *Multinomial Logit* (MNL) model to understand how users choose their mode of transportation when traveling from one place to another, and what criteria they consider in making these decisions.

In order to achieve a better fit, several tests and transformations were performed on the explanatory variables of the models. However, these modifications allowed us to identify the combination of criteria that best aligns with the observed behaviors in mode choice. Additionally, to assess the predictive ability of the *Multinomial Logit* model, was compared with the *Random Forest* and *Neural Network* models.

The implementation of the Logit models was done using the Python programming language, specifically utilizing the Biogeme library, which provides the necessary tools to estimate the parameters of these models. Furthermore, a Python class called "*Logit_model_pred*" was developed to obtain the predictions generated by the models, along with a series of graphs and reports from other libraries as well as custom-made ones. Additionally, a comparative graph is included to showcase the predictive behavior of the different models analyzed.

Based on the obtained results and comparative analysis, we conclude that the *Multinomial Logit* model exhibits a significantly lower predictive ability compared to the *Random Forest* and *Neural Network* models. The latter models, by employing more complex and flexible approaches, achieve higher precision in predicting the mode of transportation chosen by users.

Aquest treball se centra en l'anàlisi del model de repartiment modal, que s'utilitza principalment en l'àmbit del transport, a través de la teoria de la utilitat aleatòria. En particular, s'explora el model logit multinomial (MNL) per a comprendre com els usuaris trien la manera de transport en desplaçar-se d'un lloc a un altre, i quins són els criteris que tenen en compte per a prendre aquestes decisions.

Amb l'objectiu d'aconseguir un millor ajust, es van dur a terme diverses proves i transformacions en les variables explicatives dels models. No obstant això, aquestes modificacions van permetre identificar la combinació de criteris que millor s'ajusta als comportaments observats en l'elecció de la manera de transport. A més, per a avaluar la capacitat predictiva del model logit multinomial (MNL), es van comparar amb els models Random Forest i Xarxes Neuronals.

La implementació dels models Logit es va realitzar utilitzant el llenguatge de programació Python, en particular, es va emprar la llibreria Biogeme, que proporciona les eines necessàries per a aconseguir l'estimació dels paràmetres d'aquests models. A més, es va desenvolupar una classe de Python anomenada "Logit_model_pred", que permet obtenir les prediccions generades pels models i a més una sèrie de gràfics i informes d'altres llibreries i de confecció pròpia. Així mateix, s'inclou un gràfic comparatiu que mostra el comportament predictiu dels diferents models analitzats.

Sobre la base dels resultats obtinguts i a l'anàlisi comparativa, s'arriba a la conclusió que el model logit multinomial (MNL) presenta una capacitat predictiva significativament menor en comparació amb els models Random Forest i Xarxes Neuronals. Aquests últims models, en utilitzar enfocaments més complexos i flexibles, aconsegueixen una major precisió en la predicció de la manera de transport triat pels usuaris.

1. CAPÍTULO 1 – INTRODUCCIÓN Y OBJETIVOS	1
1.1. BIG DATA Y TRANSPORTE.....	1
1.2. MODELOS DE ELECCIÓN DISCRETA.....	1
1.3. MOTIVACIÓN.....	2
1.4. OBJETIVOS	3
1.5. ESTRUCTURA DEL TRABAJO.....	3
2. CAPÍTULO 2 – DESCRIPCIÓN DEL PROBLEMA	5
2.1. CONTEXTUALIZACIÓN DEL MODELO DE LAS CUATRO ETAPAS DEL TRANSPORTE	5
2.1.1. <i>Primera etapa o Modelo de Generación de viajes:</i>	6
2.1.2. <i>Segunda etapa o Modelo de distribución de viajes:</i>	6
2.1.3. <i>Tercera etapa o Modelo de Reparto modal</i>	8
2.1.4. <i>Cuarta etapa o Modelo de Asignación</i>	9
2.2. DESCRIPCIÓN DEL PROBLEMA A RESOLVER	9
2.3. BASE DE DATOS A ESTUDIAR.....	11
3. CAPITULO 3 – MARCO TEÓRICO	13
3.1. MODELOS DE ELECCIÓN DISCRETA.....	13
3.2. TEORÍA DE LA UTILIDAD ALEATORIA APLICADO AL MODELO LOGIT MULTINOMIAL	13
3.2.1. <i>Funciones de utilidad</i>	17
3.2.2. <i>Máxima verosimilitud</i>	19
3.3. MODELO LOGIT ANIDADO, LOGIT JERÁRQUICO O <i>NESTED LOGIT (NL)</i>	20
3.4. MEDIDAS DE BONDAD DE AJUSTE.....	23
3.4.1. <i>Métricas de evaluación en predicciones discretas</i>	24
4. CAPITULO 4 – OTROS MÉTODOS DE CLASIFICACIÓN.....	28
4.1. ÁRBOL DE CLASIFICACIÓN	28
4.2. RANDOM FOREST	29
4.3. REDES NEURONALES PARA CLASIFICACIÓN	30
4.3.1. <i>Modelos de Aprendizaje profundo o Deep learning</i>	30
4.3.2. <i>Modelo de Redes neuronales</i>	31
5. CAPITULO 5 – EXPERIENCIA COMPUTACIONAL	32
5.1. ETAPA 1. EXPLORACIÓN DE DATOS Y ANÁLISIS DESCRIPTIVO DE LOS DATOS	32

5.1.1.	<i>Depuración de datos</i>	32
5.1.2.	<i>Inspección de variable dependiente</i>	33
5.1.3.	<i>Inspección de variables explicativas continuas</i>	34
5.1.4.	<i>Inspección de variables explicativas categóricas</i>	36
5.2.	ETAPA 2. AJUSTE DE MODELOS DE REPARTO MODAL.....	39
5.2.1.	<i>Preparativos para ajuste de modelos de reparto modal</i>	39
5.2.2.	<i>Modelo 1: MNL</i>	41
5.2.3.	<i>Modelo 2: MNL</i>	42
5.2.4.	<i>Modelo 3: MNL</i>	42
5.2.5.	<i>Modelo 4: MNL</i>	42
5.2.6.	<i>Modelo 5.1: MNL con transformación cuadrática</i>	43
5.2.7.	<i>Modelo 5.2: MNL con transformación raíz cuadrada</i>	43
5.2.8.	<i>Modelo 5.3: MNL con transformación Box-Cox</i>	44
5.2.9.	<i>Modelo 5.4: MNL con Componentes Principales (PCA)</i>	45
5.2.10.	<i>Modelos NL: NEST_NOCAR, NEST_NOBUS y NEST_NOBP</i>	46
5.2.11.	<i>Obtención de predicciones de modelos logit</i>	48
5.2.12.	<i>Modelo 6: Red Neuronal</i>	49
5.2.13.	<i>Modelo 7: Random Forest</i>	55
6.	CAPITULO 6 – EXPOSICIÓN DE RESULTADOS	58
6.1.	COMPARACIÓN DE MÉTRICAS DE EVALUACIÓN DE MODELOS DE CLASIFICACIÓN.....	58
6.1.1.	<i>Comparación de modelos por matrices de confusión, curva ROC y sensibilidad</i>	62
6.1.2.	<i>Comparación de modelos por la métrica exactitud (accuracy)</i>	64
6.1.3.	<i>Comparación de modelos por la métrica F1-score</i>	65
6.1.4.	<i>Comparación de modelos por comportamiento de probabilidades para cada clase predicha respecto a las observadas</i>	66
6.2.	DISCUSIÓN DE RESULTADOS.....	79
7.	CAPÍTULO 7 – CONCLUSIONES Y RECOMENDACIONES	80
7.1.	CONCLUSIONES	80
7.2.	RECOMENDACIONES	81
	BIBLIOGRAFÍA	83
	ANEXOS	89

INDICE DE TABLAS

Tabla 1. Representación del modelo de distribución de viajes mediante una matriz de dos dimensiones	8
Tabla 2. Matriz de confusión de dos clases	24
Tabla 3. Conjunto de datos de EPR	33
Tabla 4. Costes y tiempos por modos entre las zonas i-ésimas a las zonas j-ésimas	33
Tabla 5. Resumen estadístico descriptivo de variables continuas explicativas	34
Tabla 6. Frecuencias de clases de variable SEXO	36
Tabla 7. Frecuencias de clases de variable ESTUDIOS	37
Tabla 8. Frecuencias de clases de variable MOTIVO	37
Tabla 9. Frecuencias de clases de variable VIAJE_OBLIGADO	38
Tabla 10. Modelos a ajustar	40
Tabla 11. Parameter grid de Random Forest	55
Tabla 12. Exactitud de modelos	64
Tabla 13. Simbologías utilizadas cuando se inspecciona cada clase	67
Tabla 14. Matriz de confusión y curvas ROC multiclase de modelos 1 y 2	58
Tabla 15. Matriz de confusión y curvas ROC multiclase de modelos 3 y 4	59
Tabla 16. Matriz de confusión y curvas ROC multiclase de modelos 5,1, 5,2 y 5,3	60
Tabla 17. Matriz de confusión y curvas ROC multiclase de modelos 6 y 7	61
Tabla 18. Comportamiento de las probabilidades respecto a las clases reales obtenidas con el modelo 1	68
Tabla 19. Comportamiento de las probabilidades respecto a las clases reales obtenidas con el modelo 2	69
Tabla 20. Comportamiento de las probabilidades respecto a las clases reales obtenidas con el modelo 3	70
Tabla 21. Comportamiento de las probabilidades respecto a las clases reales obtenidas con el modelo 4	71
Tabla 22. Comportamiento de las probabilidades respecto a las clases reales obtenidas con el modelo 5,1	72
Tabla 23. Comportamiento de las probabilidades respecto a las clases reales obtenidas con el modelo 5,2	73

Tabla 24. Comportamiento de las probabilidades respecto a las clases reales obtenidas con el modelo 5,3.....	74
Tabla 25. Comportamiento de las probabilidades respecto a las clases reales obtenidas con el modelo 5,4.....	75
Tabla 26. Comportamiento de las probabilidades respecto a las clases reales obtenidas con el modelo 6.....	76
Tabla 27. Comportamiento de las probabilidades respecto a las clases reales obtenidas con el modelo 7.....	77

INDICE DE FIGURAS

Ilustración 1. Representación del modelo de transporte con nodos (zonas) y arcos (rutas)	9
Ilustración 2. Comparación entre las distribuciones normal, Gumbel y logística	16
Ilustración 3. Ejemplo de un modelo logit anidado	21
Ilustración 4. Ejemplo del modelo logit sin agrupaciones: MNL.....	21
Ilustración 5. Introducción a la curva ROC	26
Ilustración 6. Esquema de la Curva ROC.....	26
Ilustración 7. Esquema de partición dicotómica en la construcción de un árbol de decisión	29
Ilustración 8. Diagrama de Redes neuronales artificiales.....	31
Ilustración 9. Gráfico de barras de la variable dependiente: modo de transporte	34
Ilustración 10. Boxplots de tiempo y costes de modos de transporte	35
Ilustración 11. Boxplots de distancia del viaje y edad del usuario	35
Ilustración 12. Gráficos de frecuencias relativas de las variables categóricas	39
Ilustración 13. Configuración del modelo logit con la agrupación NEST_NOCAR	46
Ilustración 14. Configuración del modelo logit con la agrupación NEST_NOBUS.....	47
Ilustración 15. Configuración del modelo logit anidado NEST_NOBP	47
Ilustración 16. Configuración de red neuronal de clasificación	49
Ilustración 17. Esquema de una función de activación reLU	51
Ilustración 18. Gráficos de error y accuracy de la Red Neuronal.....	54

1. CAPÍTULO 1 INTRODUCCIÓN Y OBJETIVOS

El modelo de elección discreta es una herramienta muy útil para evaluar los criterios que tienen en cuenta las personas al momento de tomar decisiones. Existen diversas aplicaciones de modelización de decisión sobre qué productos y/o servicios seleccionan ciertos tipos de usuarios. Se puede aplicar en la creación de campañas de publicidad, perfilamiento de públicos objetivo de marketing, como herramienta de medición de cuantificación de demandas y un sinnúmero más de utilidades, sin embargo, este trabajo se centrará en el estudio de los modelos de elección discreta y su aplicación en transporte.

1.1. Big data y transporte

Actualmente, con el crecimiento de los medios electrónicos, como billetes de transporte con sensores, recolección de datos de ubicación de personas por el uso de telefonía móvil, recolección de datos de movilidad de carácter público a través de plataformas del Ministerio de Transporte Movilidad y Agenda Urbana (en España) [1], puede registrarse la movilidad de las personas en bases de datos. Esta información almacenada puede ser tan específica, que hasta puede saberse en qué modo de transporte (vehículo privado, autobús, tren, aéreo, etc.) se han desplazado las personas. Con esta información se puede caracterizar la movilidad de las masas.

Una de las ventajas que trae consigo el uso del *Big Data* en el transporte es que en la actualidad existen más datos de la movilización de personas y mercancías disponibles. Los datos aprovechables se pueden encontrar para diferentes tamaños de espacios geográficos, como zonas censales, distritos, municipios y hasta sus formas más agregadas tal como comunidades autónomas y/o países.

1.2. Modelos de elección discreta

Para sacarle el máximo provecho a estos datos existen diversas técnicas, *softwares* y lenguajes de código abierto con los que se puede modelizar a las personas y/o vehículos transitando por un determinado lugar, así como los orígenes y destinos de sus viajes, pero sobre todo el modo de transporte que usan para desplazarse.

En 1973 McFadden [2] colaborando en proyectos de transporte de California vincula los modelos de elección discreta con los modelos logit empleados en el reparto modal (modelización de la toma de decisiones). Ese trabajo le hizo acreedor de un premio Nobel en economía en el año 2000, en el que inicialmente se planteó un escenario de elección dicotómico, es decir, dos alternativas de decisión: por ejemplo, se ejecutaba una acción o no, se votaba o no, etc. [3]. Este Trabajo Final de Máster (TFM) en su mayoría se apoya en el trabajo citado de McFadden.

En transporte la modelización de elecciones discretas se conoce como modelo de reparto modal, y considera una serie de cuestiones:

- Modos de transporte disponibles por escoger: andar, bicicleta, coche, autobús, tren, avión, entre otros.
- Los atributos medibles de cada modo de transporte para el usuario: tiempo de viaje, tiempo de espera, frecuencia, coste, confort ofrecido, disponibilidad, entre otros.
- Los atributos medibles del usuario: características socioeconómicas y demográficas siempre que estén los datos disponibles: sexo, edad, ingresos medios, posesión de vehículo propio, motivo del viaje, entre otros.

1.3. Motivación

Una correcta modelización servirá de insumo para mejoras y optimizaciones de los sistemas de transporte, por lo que ha de comprenderse que con las modelizaciones y optimizaciones en la oferta de sistemas de transporte puede aumentarse la calidad de vida de los habitantes y usuarios, reducción de costes públicos y privados asociados al transporte, disminución del tráfico vehicular, mover a más personas y/o mercancías, evitar congestiones de tráfico y/o aglomeraciones en el transporte público, reducir emisiones de gases de CO₂, en general, se podrá estar en mayor armonía entre la oferta y demanda de transporte.

Actualmente, existen diversos estudios que explican el reparto modal desde una perspectiva teórica y con suficiente rigor matemático, sin embargo, son pocos los que abordan la parte práctica, por lo que, considerando el desarrollo y notorio uso actual de lenguajes de código abierto como *Python* y *R*, este trabajo pretende aportar una guía práctica para la modelización de los criterios que tienen en cuenta los usuarios de transporte de un lugar en el momento de escoger un medio de transporte.

1.4. Objetivos

El objetivo global de este trabajo es el desarrollo de una metodología que permita modelizar los criterios que tienen en cuenta los usuarios de transporte de una determinada región de estudio al momento de escoger un medio de transporte. Para ello se plantean los siguientes objetivos específicos:

- Desarrollar una metodología práctica que permita el ajuste y predicciones con el modelo logit multinomial con funciones de utilidad sin y con transformaciones a las variables para comprobar que variables son significativas para los usuarios al momento de decidir qué modo de transporte usar.
- Desplegar mediante Python la confección de una clase que permita obtener las predicciones de modelos logit multinomial, comprobar la significatividad de las variables y bondad de ajuste del modelo.
- Comparar los resultados con otros modelos de *Machine Learning* para clasificación multinomial como son: Redes neuronales y el *Random Forest*, empleando las métricas clásicas de evaluación de modelos de clasificación para comparar los rendimientos de los modelos.
- Evaluar la capacidad predictiva del modelo desarrollado.

1.5. Estructura del Trabajo

A continuación, se explica la distribución de contenidos que tiene este trabajo. En el capítulo 1 se aborda una introducción general, una motivación para realizar este trabajo y por último se enuncian los objetivos de la investigación.

En el capítulo 2 se explican tanto el problema por resolver, como los datos disponibles con las respectivas explicaciones de las variables de la base de datos.

En el capítulo 3 se presenta el marco teórico de la metodología para poder resolver los modelos de reparto modal, incluida la revisión bibliográfica y matemática.

En el capítulo 4 se incluye la explicación teórica de “otros métodos de clasificación”, específicamente de los modelos: redes neuronales y *Random Forest*.

En el capítulo 5, se comentará la experiencia computacional obtenida al explorar y hacer un análisis descriptivo de los datos, acto seguido se explica cómo se resuelve cada uno de los modelos planteados mediante *Python*.

Posteriormente, en el capítulo 6, se abordará una exposición de los resultados de cada modelo, comparación de métricas de rendimientos y discusión de resultados.

Para cerrar, en el capítulo 7 se incluyen unas conclusiones y recomendaciones para continuar esta línea de investigación.

2. CAPÍTULO 2 – DESCRIPCIÓN DEL PROBLEMA

Para identificar el problema que se procura resolver en este trabajo es necesario contextualizar al lector acerca de cuándo y cómo se utiliza el reparto modal en la modelización del transporte. Se introduce “el método de las cuatro etapas” de modelización de demanda del transporte.

Para confeccionar la modelización del transporte o modelización de la demanda del transporte, que tiene como objetivo la previsión, mejora, optimización y planificación de los sistemas de transporte, se utiliza habitualmente el método de las cuatro etapas de transporte.

A continuación, se hace una breve explicación de los procesos y conceptos implementados en cada una de las etapas de la modelización de la demanda del transporte. Tener en cuenta que el objetivo de este trabajo es encontrar una metodología práctica para la resolución para la tercera etapa o modelo de reparto modal.

2.1. Contextualización del modelo de las cuatro etapas del transporte

Según Barreno [4], la modelización para el análisis y planificación de los sistemas de transporte consiste en una representación de las interacciones de la demanda y oferta del transporte, siendo la demanda representada por la cantidad de viajes que los usuarios realizan en cada uno de los medios de transporte disponibles. Mientras que la oferta se representa por las vías, servicios y medios de transporte disponibles.

La modelización comienza dividiendo el área ámbito de estudio en n zonas. Estas zonas pueden ser provincias, municipios o zonas censales según el tamaño y precisión que el estudio requiera. Luego, definiendo el espacio temporal en el que se medirán la cantidad de viajes que se produzcan y atraigan en cada una de las n zonas del área ámbito de estudio que puede ser: horario o diario en la mayoría de los casos.

También, se tiene en consideración que un viaje tiene un origen i y un destino j . Con esta información presente se procede a la explicación resumida de las cuatro etapas que se explican a continuación:

2.1.1. Primera etapa o Modelo de Generación de viajes:

Según Ortuzar [5], quien explica las etapas de la modelización de la demanda del transporte, indica que en la primera etapa o modelo de generación de viajes se cuantifica la cantidad de viajes O_i que tienen como origen cada i -ésima zona, a estos viajes se les conoce como viajes producidos o generados. Simultáneamente, se cuantifican los viajes D_j que tienen como destino cada j -ésima zona, a estos viajes se les conoce como viajes atraídos.

Se utilizan modelos de regresión lineal múltiple (*MLR*) que explican la producción y atracción mediante variables exógenas socioeconómicas y demográficas X_k :

$$MLR \text{ Producción: } O_i = \beta_0 + \beta_1 X_1 + \dots + \beta_P X_P$$

$$MLR \text{ Atracción: } D_j = \gamma_0 + \gamma_1 X_1 + \dots + \gamma_A X_A$$

En donde:

β y γ son los parámetros de los modelos de producción y atracción respectivamente.

Los subíndices P y A hacen referencia a las P-ésima y γ -ésima variables de los modelos *MLR* en que se está usando las variables explicativas sean de producción o atracción respectivamente.

X_1, \dots, X_P son las variables exógenas que explican la producción de viajes.

X_1, \dots, X_A son las variables exógenas que explican la atracción de viajes.

Se debe tener en cuenta que cada uno de los modelos tiene que superar los criterios de validación habitualmente usados: ausencia de multicolinealidad, homocedasticidad, autocorrelación de residuos, observaciones atípicas e influyentes, debe tener la forma funcional adecuada, normalidad de residuos, etc.

2.1.2. Segunda etapa o Modelo de distribución de viajes:

Siguiendo con la estructura propuesta por Ortuzar [5], una vez calculados los viajes producidos y atraídos por cada zona en la etapa anterior, se requiere conocer cuáles son los destinos j de los viajes que cada zona i produce. Y desde que zonas i partieron los viajes que fueron atraídos a las zonas j del estudio.

En esta etapa se usan modelos gravitatorios doblemente restringidos (orígenes y destinos) partiendo del coste intrínseco C_{ij} que representa viajar desde la zona i a la zona j , este coste no es necesariamente el coste monetario, puede ser: gasto económico, distancia, tiempo, incomodidad o *discomfort*, etc.

El modelo de distribución parte de la analogía a la ley de gravitación de Newton que describe la fuerza o interacción gravitatoria entre distintos cuerpos con masa.

$$F = G \frac{m_1 m_2}{r^2}$$

En donde F representa el módulo de la fuerza ejercida entre los cuerpos 1 y 2, G es la constante de gravitación, m_1 y m_2 las masas de los cuerpos 1 y 2 y r la distancia entre los centros de masas de los dos cuerpos. Análogamente, este principio se utilizó por primera vez por Casey en 1955 [6], quien propuso que para hacer la estimación de los viajes que se producen entre las zonas i y j se podía usar una expresión como la que sigue:

$$V_{ij} = \alpha \frac{P_i P_j}{d_{ij}^2}$$

En donde P_1 y P_2 son las poblaciones de las i -ésima y j -ésima zona respectivamente, α un factor y d_{ij}^2 la distancia al cuadrado entre las zonas i -ésima y j -ésima. Luego, este modelo fue considerado muy simplista y según Ortuzar [5] ha sido actualizado partiendo del mismo principio de gravitación, pero en este caso han cambiado las variables de población por los viajes producidos O_i y D_j y la distancia por una función de coste que representa el coste o dificultad de viajar entre las zonas i -ésima y j -ésima.

$$V_{ij} = \alpha O_i D_j f(c_{ij})$$

La función de coste $f(c_{ij})$ es una función con uno o más parámetros de ajuste. Usualmente, esta función recibe el nombre de *Deterrence function* (función de disuasión) porque representa el desincentivo a viajar cuando se aumenta la distancia (tiempo) o coste y suele estar representada por una de las siguientes versiones:

$$f(c_{ij}) = e^{-\beta c_{ij}} \quad \text{función exponencial}$$

$$f(c_{ij}) = c_{ij}^{-n} \quad \text{función potencial}$$

$$f(c_{ij}) = c_{ij}^n e^{-\beta c_{ij}} \quad \text{combinación de funciones}$$

La tabla 1 sirve como una representación visual del resultado del modelo de distribución en el que se calculan los viajes V_{ij} .

Tabla 1. Representación del modelo de distribución de viajes mediante una matriz de dos dimensiones

Orígenes	Destinos						$\sum_j V_{ij}$
	1	2	. . .	j	. . .	n	
1	V_{11}	V_{12}	. . .	V_{1j}	. . .	V_{1n}	O_1
2	V_{21}	V_{22}	. . .	V_{2j}	. . .	V_{2n}	O_2
.
.
i	V_{i1}	V_{i2}	. . .	V_{ij}	. . .	V_{in}	O_i
.
.
n	V_{n1}	V_{n2}	. . .	V_{nj}	. . .	V_{nn}	O_n
$\sum_i V_{ij}$	D_1	D_2	.	D_j	.	D_n	$\sum_{ij} V_{ij} = V$

Nota: En esta tabla se explica la distribución de los viajes entre cada origen y destino. Ha sido tomada del libro de Modelos de Transporte, por Juan de Dios Ortúzar. 2011.

En donde:

V_{ij} : Representa los viajes que se han originado en la i-ésima zona y que han terminado en j-ésima zona.

2.1.3. Tercera etapa o Modelo de Reparto modal

Mediante encuestas se recopila información de la movilidad a personas de cada una de las zonas del ámbito de estudio, cuáles son sus preferencias de uso de modo de transporte, motivos de viajes, orígenes y destinos, características socioeconómicas, con el fin de calcular cuál es el modo de transporte que sería más probable que elijan.

En esta etapa se hace uso del modelo de la teoría de utilidad aleatoria con modelos logit multinomial y/o logit anidados. El objetivo de este trabajo es determinar una metodología práctica para el ajuste de los parámetros de esta etapa del modelo, se profundiza más a detalle en el próximo capítulo.

2.1.4. Cuarta etapa o Modelo de Asignación

En esta etapa también llamada de asignación de rutas [7], hacen referencia al envío de artículos desde un origen hasta un destino, pero en este caso se estima la cantidad de viajeros que usarán cada una de las rutas disponibles (carreteras, vías de tren, calles alternas, etc.), determinado por el objetivo que se pretende modelizar, sea poner la red en equilibrio, maximizando los viajeros según la capacidad de las rutas, reducción de costes o haciendo usos de modelos de asignación que satisfagan los límites de oferta y demanda. La red que se presenta en la ilustración 1, muestra una representación visual de la manera más simple que puede tener el modelo de asignación. Hay que destacar que origen y destino puede ser el mismo (i -ésima zona = j -ésima zona), por ejemplo, aquellas personas que tienen su hogar y trabajo en la misma zona.

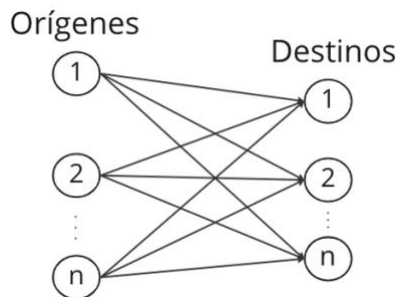


Ilustración 1. Representación del modelo de transporte con nodos (zonas) y arcos (rutas)

Nota: Fuente: elaboración propia

2.2. Descripción del problema a resolver

Una vez introducido el modelo de transporte de las cuatro etapas, se explicará más en detalle la problemática que se resuelve en la tercera etapa o de reparto modal. Empezando por la explicación de las variables que tienen los datos, para continuar con el propósito a desarrollar.

Habitualmente, los datos de las preferencias del modo de transporte de los usuarios se recolectan por medio de dos tipos de encuestas que son:

- **Encuesta de preferencias reveladas (EPR):** el objetivo de la implementación de este tipo de encuesta es recolectar datos para modelizar el reparto modal de los usuarios de una zona estudiada. Se recaba información real de desplazamientos de los usuarios [8], esto corresponde a una ventaja en la modelización, dado que se apoya en el comportamiento real de los viajeros. Sin embargo, obtiene limitaciones, como su alto coste de ejecución. La información recabada puede ser: origen y destino, motivo del viaje, modo de transporte empleado, condiciones del viaje, datos socioeconómicos del usuario, etc.
- **Encuesta de preferencias declaradas (EPD):** estas encuestas hacen referencia a un conjunto de procedimientos que se basan en lo que los individuos declaran acerca de sus preferencias cuando se le presentan escenarios hipotéticos [9]. El objetivo de este tipo de encuestas es el de evaluar el impacto que puede tener la implantación de cambios en los atributos de los modos de transporte, conocer si hay interés en la ejecución de nuevos proyectos en sistema de transporte, etc. Por su naturaleza hipotética, el primer desafío es definir las variables que intervienen en el modelo, así como los niveles que estas pueden tener. Estos niveles deben asignarse de manera realista y no entrar en contradicciones con la experiencia de los encuestados.

Los resultados de estas encuestas pueden usarse para el cálculo de elasticidades en la demanda. El tipo de preguntas que suelen hacerse son:

- “¿estaría dispuesto a viajar en un determinado modo de transporte con ciertas condiciones (específicas)?” (refiriéndose a un modo de transporte no existente),
- ¿“cuánto estaría dispuesto a pagar, esperar, o que durase el viaje en un nuevo modo de transporte?”,
- ordenar las preferencias de distintos modos de transporte,
- etc.

Los datos que se utilizarán en este trabajo no son reales, pero mantienen una estructura semejante a los resultados que provienen de una encuesta de preferencia revelada para un estudio de movilidad de una ciudad mediana de España.

2.3. Base de datos a estudiar

La primera está basada en los resultados de una EPR en la que se les preguntó a habitantes del ámbito de estudio acerca de sus movilizaciones. La base de datos consta de 2240 observaciones. Cada observación corresponde a la respuesta de una persona. Las variables se explican a continuación:

- Id Encuesta: [numérico]: 1 al 2240.
- Sexo: [categórico]: MUJER; HOMBRE.
- Edad: [numérico]: edad del encuestado.
- Estudios: [categórico]: nivel de estudios según las categorías siguientes: Estudios primarios acabados; Estudios secundarios acabados; Estudios universitarios medios acabados; Estudios universitarios superiores acabados; Formación profesional; Posgrado, Máster, Doctorado; Sin Estudios.
- Zona_origen: [numérico]: zona de origen de la 1 a la 120.
- Zona_destino: [numérico]: zona de destino de la 1 al 120.
- Viaje_obligado: [categórico]: si el viaje es o no una obligación, por ejemplo: viajes obligados pueden ser: trabajo, estudio o similar y consta de dos categorías: Obligadoy; No obligado.
- Motivo_Viaje: [categórico]: según el motivo del viaje: HBW; HBS; HBO; NHB. En donde cada categoría tiene los siguientes significados por sus siglas en inglés: HBW: Basado en el Hogar con motivo de Trabajo; HBS: Basado en el hogar con motivo de estudio; HBO: Basado en el hogar con motivo distinto a estudio o trabajo; NHB: viaje no basado en el Hogar sin importar el motivo.
- Modo_txt: [categórico]: modo en que viajaron y puede ser: A pie; coche; autobús; bici_patinete.

Adicionalmente, existe una base de datos, en la que se han estimado las distancias, tiempos y costes de los modos “coche”, “autobús” y “bici_patinete” entre cada una de las zonas del ámbito de estudio:

- From: [numérico]: zona del ámbito de origen de viaje.
- To: [numérico]: zona del ámbito de destino viaje.
- Distancia entre zonas: [numérico]: distancia entre la zona de origen y destino.
- Tiempo en coche: [numérico]: tiempo en coche entre la zona de origen y destino.

- Tiempo en autobús: [numérico]: tiempo en autobús entre la zona de origen y destino.
- Tiempo en bici_patinete: [numérico]: tiempo en bici_patinete entre la zona de origen y destino.
- Coste en coche: [numérico]: coste en coche entre la zona de origen y destino.
- Coste en bus: [numérico]: coste en autobús entre la zona de origen y destino.
- Coste a bici_patinete: [numérico]: coste en bici_patinete entre la zona de origen y destino.

El modelo que se llevará a cabo es uno que prediga el modo en que las personas viajan a partir de la utilidad que cada usuario le otorga al medio de transporte. Hay que tener en cuenta que solo pueden incluirse en el modelo las variables con las que puedan hacerse predicciones, es decir, las que están o puedan estar disponibles referentes al ámbito de estudio.

3. CAPITULO 3 – MARCO TEÓRICO

3.1. Modelos de elección discreta

Los modelos de elección discreta explican cuáles son los atributos o características más significativos de un producto o servicio para un usuario o grupo de usuarios al momento de la toma de decisiones. Los modelos de elección discreta se utilizan para explicar o predecir una elección dentro de un conjunto de dos o más alternativas disponibles, discretas y mutuamente excluyentes. Por ejemplo, la preferencia de ciertos usuarios hacia un modo de transporte para ir al trabajo, escuela, etc., otro ejemplo muy habitual es el de la elección de cierto tipo de chocolates en función de sus características por ejemplo el precio, calidad, origen etc.

Para introducir formalmente los modelos de elección discreta [10] nos remontamos a 1973 en donde McFadden plantea que, al momento de tomar una decisión, la alternativa que se escoge es aquella que maximiza la utilidad del usuario de entre un conjunto de alternativas disponibles. Para conseguir modelizar lo anterior se hace uso de la teoría de la utilidad aleatoria a través de modelos logit multinomial o alguna de sus variaciones como son: Modelo Logit Multinomial (MNL); Logit anidado, Logit jerárquico o *Nested Logit* (NL); Logit condicional o *Conditional Logit* (CL); etc.

3.2. Teoría de la utilidad aleatoria aplicado al Modelo logit multinomial

Para estudiar al consumidor y sus decisiones, empleando la teoría de la utilidad aleatoria se hacen las siguientes suposiciones.

- Los individuos actúan de forma racional.
- Cuentan con la información completa respecto a las alternativas de elección y sus atributos.
- Escogen la alternativa que les proporciona la máxima utilidad.

Cada individuo asocia una utilidad a cada una de las alternativas y sus atributos que son los aspectos medibles que el individuo tiene en cuenta al momento de tomar una decisión. A continuación, se define la teoría de la utilidad aleatoria: sea un conjunto de J alternativas disponibles y mutuamente excluyentes $A = \{A_1, A_2, \dots, A_J\}$ para el individuo

n , y sea el conjunto de K atributos, $X = \{X_{1n}, X_{2n}, \dots, X_{Kn}\}$ del usuario n ; en donde los atributos del usuario pueden ser: sexo, edad, nivel de ingresos, y los atributos de sus alternativas disponibles, que pueden ser: tiempos, costes, disponibilidades de las distintas alternativas A_j . Luego, cada una de las alternativas tendrá asociada una utilidad U_{jn} para cada uno de los usuarios, es decir, la utilidad que cada persona n le asigna a cada alternativa disponible A_{jn} .

La teoría de la utilidad aleatoria explica que la utilidad U_{jn} tiene dos componentes, una que es observable, medible y generalizable V_{jn} y otra que es estocástica ε_{jn} , que revela los errores de medición, gustos, costumbres de cada individuo. Esta parte que es estocástica explica porque los individuos con iguales características pueden llegar a escoger alternativas distintas, prefiriendo a veces la que a priori es menos conveniente. Se define la alternativa i como la alternativa que maximiza la utilidad del usuario, y se expresa la utilidad con la siguiente Eq 1.

$$U_{in} = V_{in} + \varepsilon_{in} \quad (\text{Eq. 1}).$$

Cada individuo escoge la alternativa A_{in} que maximiza su utilidad, en otras palabras, la utilidad de la alternativa escogida es mayor a la utilidad del resto de alternativas:

$$U_{in} > U_{jn}, \forall A_j \in A(n) \quad (\text{Eq. 2}).$$

Siendo $A(n)$ el conjunto de alternativas disponibles que tiene el individuo n . Por lo que, se puede reescribir la Eq. 1 y la Eq. 2 así:

$$V_{in} - V_{jn} > \varepsilon_{jn} - \varepsilon_{in} \quad (\text{Eq. 3}).$$

Dado que $(\varepsilon_{in}, \varepsilon_{jn})$ son valores estocásticos y no es posible determinarlos, deben asignarse unas probabilidades a la elección de cada una de las alternativas, por lo que la probabilidad de que el individuo n escoja la alternativa A_i viene dada por la Eq. 4:

$$P_{in} = P\{\varepsilon_{in} \geq \varepsilon_{jn} + (V_{jn} - V_{in})\} \forall A_j \in A(n) \quad (\text{Eq. 4})$$

Para estimar la probabilidad P_{in} , se supone que las componentes estocásticas $(\varepsilon_i, \varepsilon_j)$ se comportan con una distribución equivalente, específicamente, se comportan como una distribución de Gumbel independiente e idéntica, es decir, que tienen la misma varianza: $(\varepsilon_i, \varepsilon_j) \sim \text{Gumbel iid}$. La función de distribución Gumbel está dada por:

$$F(\varepsilon) = e^{-e^{-\mu(\varepsilon-\eta)}}, \mu > 0, \quad (\text{Eq. 5})$$

cuya función de densidad es:

$$f(\varepsilon) = \mu e^{-\mu(\varepsilon-\eta)} e^{-e^{-\mu(\varepsilon-\eta)}}, \quad (\text{Eq. 6})$$

siendo η el parámetro de localización y μ el parámetro de escala. Esta distribución tiene algunas propiedades como que η es la moda y la media es $\eta + \gamma/\mu$, siendo γ la constante de Euler ($\sim 0,577$), la varianza es $\pi^2/6\mu^2$.

Por tanto, pueden omitirse $(\varepsilon_i, \varepsilon_j)$ [11] y se obtiene un modelo logit, únicamente, a partir de las componentes medibles. La Eq. 7 representa V_{in} que es la utilidad de la alternativa i para el individuo n :

$$V_{in} = \sum_{k=1}^K \theta_{ik} X_{ikn} = \beta_0 + \beta_1 X_{1n} + \dots + \beta_K X_{Kn} \quad (\text{Eq. 7}).$$

En donde β_0, \dots, β_K son parámetros por estimar, los cuales representan la utilidad marginal de cada uno de los atributos de los modos de transporte, por ejemplo: tiempo, distancia, coste, confort también pueden incluirse los atributos de los usuarios que toman las decisiones: sexo, edad, categorías como estudiante, trabajador, viajero frecuente, etc. Usualmente los parámetros difieren por alternativas y atributos, pero no por individuos. El modelo logit emplea una distribución Gumbel [11] con parámetro de localización cero, $\eta = 0$, y como cabe esperar sus errores tienen media nula, sin embargo, en realidad la media viene dada por γ/μ (al ser una distribución sesgada), aunque este valor no es observable pues se emplean diferencias entre utilidades y la diferencia entre dos términos aleatorios con la misma media resulta en media cero. La diferencia entre dos distribuciones de Gumbel, con parámetro de escala μ y parámetro de localización nulo

$\eta = 0$, distribuye logísticamente, es decir, $\varepsilon_{nji}^* = \varepsilon_{nj} - \varepsilon_{ni}$ tiene la función de distribución:

$$F(\varepsilon_{nji}^*) = \frac{e^{\mu\varepsilon_{nji}^*}}{1+e^{\mu\varepsilon_{nji}^*}} \quad (\text{Eq. 8}).$$

Luego, la distribución logística, es muy similar a la normal, cuya varianza es el doble de la de Gumbel, por lo que la logística estándar tiene una varianza de $\pi^2/3$. Los errores Gumbel en modelos de la familia logit, se refieren al error de cada alternativa, siendo observable la diferencia con la distribución logística. La ilustración 2 muestra la similitud entre ambas distribuciones y el sesgo de la distribución Gumbel, relevante en la selección y calibración de modelos de elección discreta.

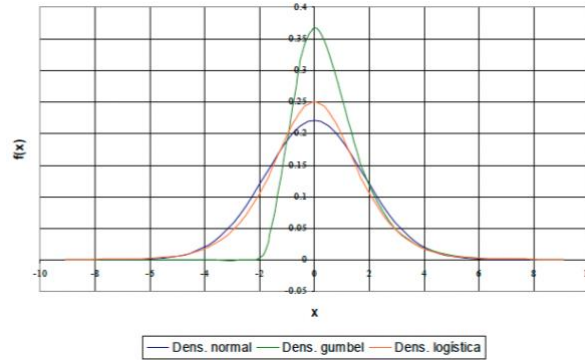


Ilustración 2. Comparación entre las distribuciones normal, Gumbel y logística

Nota: Gumbel estándar, Logística estándar y normal $(0, \pi^2/3)$

Fuente: imagen extraída de [11]

Luego, la probabilidad de que el individuo n escoja la alternativa A_i viene dado por la siguiente expresión (modelo logit):

$$P_{in} = \frac{e^{V_{in}}}{e^{V_{1n}} + \dots + e^{V_{jn}}} = \frac{e^{\mu V_{in}}}{\sum_j e^{\mu V_{jn}}} = \frac{e^{\sum_k \theta_{ik} X_{ikn}}}{\sum_j e^{\sum_k \theta_{jk} X_{jkn}}}$$

(Eq. 9).

El parámetro de escala μ no es identificable, por lo que habitualmente, se fija como $\mu = 1$, por ello los parámetros se establecen de manera lineal.

En la [sección 3.2.1](#) se incluye un ejemplo de un caso de elección binaria para que el lector pueda comprender el uso de las variables, funciones y conceptos que se han introducido hasta este punto.

El modelo asume la inexistencia de correlación entre las distintas alternativas disponibles. Esta propiedad es la de independencia de alternativas irrelevantes y puede llevar a que el modelo estime resultados sesgados si las alternativas se encontraran correlacionadas entre sí.

Siguiendo el planteamiento de García [10], el modelo puede ser probit o logit. El primero tiene la ventaja de que teóricamente es más general, pero su calibración es más compleja y suele no ser eficiente al momento de la optimización y si se utilizan modelos de máxima verosimilitud es posible encontrarse con múltiples óptimos de verosimilitud. Es por lo anterior que es más habitual utilizar modelos logit multinomial con sus variantes cuando son necesarias como el modelo logit jerárquico o logit mixto. Así pues, en este trabajo nos focalizaremos en el uso de los modelos logit.

3.2.1. Funciones de utilidad

Las funciones de utilidad se construyen a partir de los atributos significativos de cada uno de los modos de transporte. Para ilustrar mejor al lector sobre la parametrización de las funciones de utilidad se hará uso de un ejemplo de elección binaria por su simpleza, sin embargo, un modelo puede incluir dos o más alternativas de elección.

Un individuo que tiene que desplazarse hacia su trabajo, tiene que escoger entre dos modos de transporte disponibles: coche, autobús. El individuo que quiere escoger el medio de transporte que maximiza su utilidad tendrá en cuenta entre otros, el tiempo (t_{coche} y t_{bus}) y coste de viaje (c_{coche} y c_{bus}).

$$V_{coche} = \beta_{coche} + \beta_{t_{coche}} t_{coche} + \beta_{c_{coche}} c_{coche} \quad (\text{Eq. 10}).$$

$$V_{bus} = \beta_{bus} + \beta_{t_{bus}} t_{bus} + \beta_{c_{bus}} c_{bus} \quad (\text{Eq. 11}).$$

Primero que todo, puede notarse que la función de utilidad tiene la forma de un modelo de regresión lineal múltiple ($Y = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k + \varepsilon$). Los parámetros iniciales β_{coche} y β_{bus} , cuya interpretación, cuando son distintos de cero, sería la propensión inicial de tender a escoger o no a priori cada modo según el signo resultante del parámetro.

Posteriormente, se considera que las utilidades V_{coche} y V_{bus} no tienen unidades, por tanto, cada parámetro β tiene la unidad inversa del regresor a que acompaña, por ejemplo. Sea t_{coche} el tiempo en minutos que toma el viaje en coche, por ende $\beta_{t_{coche}}$ tiene unidades $\frac{1}{min}$ así sucesivamente, para cada parámetro β .

Luego, que en la representación de las Eq. 10 y Eq. 11 existe un parámetro de tiempo para coche $\beta_{t_{coche}}$ y otro para el autobús $\beta_{t_{bus}}$, lo que esto significa es que el tiempo en el coche y en el autobús tienen una distinta valoración para el usuario, en otras palabras, que tiene en cuenta la comodidad, libertad, es decir, aprovecha el tiempo que un modo le ofrece más que el otro, por ejemplo, mientras conduce no puede hacer otra actividad simultáneamente, pero en el autobús, sí que puede hacer algunas otras cosas mientras se desplaza o espera. Si el usuario no adjudica un valor adicional al tiempo de un modo u otro, el parámetro de tiempo sería el mismo tanto para el tiempo en coche como para el tiempo en autobús: $\beta_{t_{coche}} = \beta_{t_{bus}} = \beta_t$. Lo mismo respecto al parámetro asociado al coste y demás si existiesen.

Una vez definidos los parámetros con los que se entrenará el modelo de reparto modal, se introducirán las Eq. 10 y Eq.11 en la Eq. 9 para calcular las probabilidades de elegir cada modo. La probabilidad de que el usuario escoja el coche:

$$P(coche|t_{coche}, c_{coche}) = \frac{e^{V_{coche}}}{e^{V_{coche}} + e^{V_{bus}}}$$

$$P(coche|t_{coche}, c_{coche}) = \frac{e^{\beta_{coche} + \beta_{t_{coche}} t_{coche} + \beta_{c_{coche}} c_{coche}}}{e^{\beta_{coche} + \beta_{t_{coche}} t_{coche} + \beta_{c_{coche}} c_{coche}} + e^{\beta_{bus} + \beta_{t_{bus}} t_{bus} + \beta_{c_{bus}} c_{bus}}}$$

(Eq. 12).

La probabilidad de que escoja el autobús:

$$P(bus|t_{bus}, c_{bus}) = \frac{e^{V_{bus}}}{e^{V_{coche}} + e^{V_{bus}}}$$

$$P(bus|t_{bus}, c_{bus}) = \frac{e^{\beta_{bus} + \beta_{t_{bus}} t_{bus} + \beta_{c_{bus}} c_{bus}}}{e^{\beta_{coche} + \beta_{t_{coche}} t_{coche} + \beta_{c_{coche}} c_{coche}} + e^{\beta_{bus} + \beta_{t_{bus}} t_{bus} + \beta_{c_{bus}} c_{bus}}}$$

(Eq. 13).

En las bases de datos de este tipo de problemas, cada observación suele ser un viaje con los datos de cada variable y su respectiva decisión, por lo que a partir de esta información se entrenan los modelos. El siguiente paso es estimar los parámetros que explican el comportamiento de los usuarios en función de estas variables, con el objetivo de hacer predicciones. El método que utilizamos para ello es el de la máxima verosimilitud.

3.2.2. Máxima verosimilitud

Dadas tanto una variable aleatoria representada por unos parámetros, como una muestra poblacional en la base de datos existente, se consideran estimadores Máximo-verosímiles de los parámetros de una población determinada a aquellos parámetros que formarían con mayor probabilidad la muestra observada. En otras palabras, los estimadores Máximo-verosímiles son los que para una función de densidad conjunta alcanza un máximo [12]. El método de estimación de máxima verosimilitud es exactamente el mismo para logit con múltiples alternativas, como para el modelo logit binario, sabemos que:

$$P_n(A_{in}|A_n) = \frac{e^{V_{in}}}{\sum_{j \in A_n} e^{V_{jn}}}, \quad (\text{Eq. 14})$$

en donde $A_n = \{A_{1n}, \dots, A_{jn}\}$, es el conjunto de alternativas de elección disponibles para el individuo n.

La función de verosimilitud (\mathcal{L}^*) está dada por:

$$\mathcal{L}^*(x_i; \beta_K) = f(x_1, x_2, \dots, x_K) \quad (\text{Eq. 15}).$$

Donde, x_1, x_2, \dots, x_K representan los K atributos del usuario y alternativas de transporte, es decir: sexo, edad, ingresos, coste, tiempo.

$$\mathcal{L}^*(\beta_K|x_i) = \prod_{i=1}^N f(x_i; \beta)$$

(Eq. 16).

Dado que los parámetros pueden tener distintos valores en función de cada alternativa los parámetros están dados por: $\beta = \{ \beta_{11}, \dots, \beta_{JK} \}$.

Luego, el logaritmo de la verosimilitud $\mathcal{L} = Ln(\mathcal{L}^*)$ de una muestra sería como sigue:

$$\mathcal{L}(\beta|x_i) = \sum_{n=1}^N \left(\sum_{i \in C_n} y_{in} Ln(P_n(i|A_n)) \right)$$

(Eq. 17).

Donde $y_{in} = 1$ si el usuario n escoge la alternativa i , 0 si no. Luego se introduce la Eq. 13 en la Eq. 19 y se obtiene:

$$\mathcal{L}(\beta_1, \dots, \beta_K) = \mathcal{L}(\beta|x_i) = \sum_{n=1}^N \sum_{i \in A_n} y_{in} \left(V_{in} - Ln \sum_{j \in A_n} e^{V_{jn}} \right)$$

(Eq. 18).

La estimación de máxima verosimilitud consiste en encontrar el vector β , resolviéndolo como problema de optimización [13].

3.3. Modelo Logit anidado, Logit jerárquico o *Nested Logit* (NL)

Este modelo es similar al MNL, con la salvedad de que en este las alternativas se agrupan en conjuntos anidados [14], donde cada conjunto comparte unas características comunes que las distingue de las opciones de los demás conjuntos.

Por ejemplo: en la ilustración 3 se observa un modelo logit anidado que está considerando S conjuntos de “transporte privado” con sus alternativas (coche y motocicleta), “transporte público” (autobús y tren) y “transporte no motorizado” (bicicleta y caminar). En la ilustración 4 se muestra el mismo modelo sin agrupaciones, tal como sería un modelo logit multinomial. El modelo logit anidado asume que los consumidores primero

eligen entre los diferentes conjuntos de alternativas y luego, dentro de cada conjunto, eligen el que maximiza su utilidad. Este modelo permite modelizar las relaciones complejas que puede haber entre las opciones y las preferencias del usuario y puede ser útil para analizar las elecciones en contextos con opciones agrupadas.

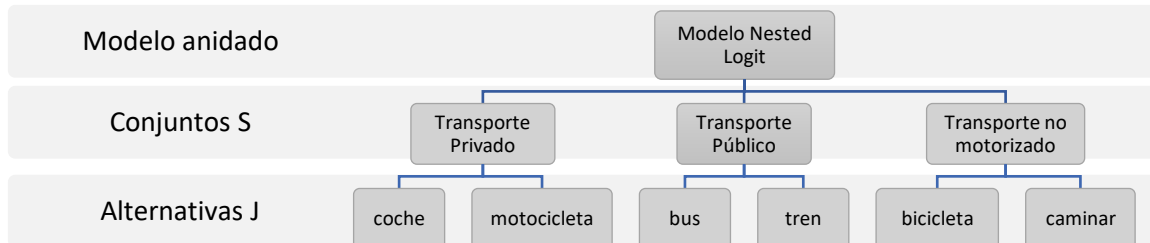


Ilustración 3. Ejemplo de un modelo logit anidado

Fuente: elaboración propia

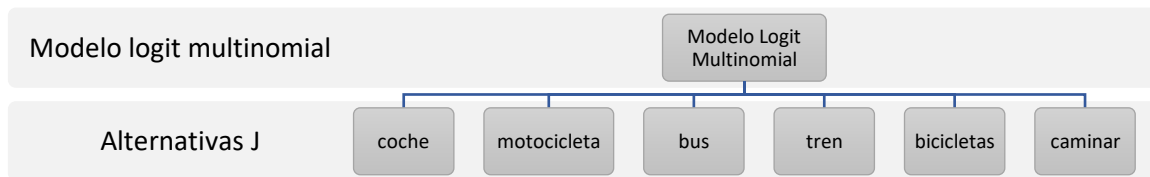


Ilustración 4. Ejemplo del modelo logit sin agrupaciones: MNL

Fuente: elaboración propia

Para definir formalmente el modelo logit anidado [15] se parte de que existen J alternativas. Se agrupan las alternativas en S conjuntos $\{B_1, B_2, \dots, B_S\}$, las alternativas pueden estar correlacionadas dentro de cada conjunto, pero es necesario que los conjuntos sean independientes. Luego, como antes, el individuo escoge la alternativa que le dé la mayor utilidad, donde la utilidad de escoger la alternativa j en el conjunto B_s para el individuo n esta dada por:

$$U_{nj} = x'_{nj}\beta + Z'_s\alpha + \varepsilon_{nj}, \quad (\text{Eq. 19})$$

donde Z_s representa las características del conjunto y ε_{nj} una variable aleatoria que no se ha tenido en cuenta en el estudio.

Siguiendo el razonamiento de [15] el modelo logit anidado asume que el vector no observado de la función de utilidad esta dado por: $\varepsilon_n = \{\varepsilon_{n1}, \dots, \varepsilon_{nJ}\}$ y tiene una distribución acumulada siguiente:

$$\exp\left(\sum_{s=1}^S \left(\sum_{j \in B_s} e^{-\varepsilon_{nj}/\lambda_s}\right)^{\lambda_s}\right).$$

(Eq. 20)

Esta distribución es un tipo de distribución de valor extremo generalizada (GEV). En el modelo logit cada ε_{nj} es independiente con una distribución de extremo univariada. Para este GEV, los valores pueden estar correlacionados dentro de cada grupo en el conjunto B_s , sin embargo, los valores de diferentes conjuntos no están correlacionados. Para cualesquiera dos alternativas j y m dentro del conjunto B_s , ε_{nj} esta correlacionada con ε_{nm} . Por otra parte, para dos alternativas en distintos conjuntos, tendrán correlación nula en la parte no observada ε_n .

El parámetro λ_s es una medida del grado de independencia de la utilidad no observada entre las alternativas del conjunto S , mientras mayor sea este parámetro indica una mayor independencia y menos correlación, por otra parte, el estadístico $(1 - \lambda_s)$ es una medida de correlación. Cuando $\lambda_s = 1$, para todos los conjuntos S , representa independencia entre todas las alternativas en todos los conjuntos, luego, el logit anidado se reduce al modelo logit estándar.

La probabilidad de elección para la alternativa $i \in B_s$ esta dada por la siguiente expresión:

$$P_{ni} = \frac{e^{v_{ni}/\lambda_s} \left(\sum_{j \in B_s} e^{v_{nj}/\lambda_s}\right)^{\lambda_s - 1}}{\sum_{\ell=1}^S \left(\sum_{j \in B_\ell} e^{v_{nj}/\lambda_\ell}\right)^{\lambda_\ell}}$$

(Eq. 21).

Donde λ_ℓ , representa el parámetro de cada conjunto considerado ℓ . Se considera la alternativa $i \in B_s$ y $m \in B_\ell$ y dado que en la ecuación 22 el denominador es el mismo para todas las alternativas la ratio de probabilidades es la ratio de numeradores:

$$\frac{P_{ni}}{P_{nm}} = \frac{e^{v_{ni}/\lambda_S} \left(\sum_{j \in B_S} e^{v_{nj}/\lambda_S} \right)^{\lambda_S - 1}}{e^{v_{nm}/\lambda_\ell} \left(\sum_{j \in B_\ell} e^{v_{nj}/\lambda_\ell} \right)^{\lambda_\ell - 1}}$$

(Eq. 22).

Si $s = \ell$ (i y m están en el mismo conjunto) se pueden cancelar sus términos entre paréntesis y se tiene:

$$\frac{P_{ni}}{P_{nm}} = \frac{e^{v_{ni}/\lambda_S}}{e^{v_{nm}/\lambda_\ell}}$$

(Eq. 23)

Esta relación es independiente de todas las demás alternativas. Para $s \neq \ell$ (i y m están en los distintos conjuntos), los términos en los paréntesis no se cancelan y la relación de probabilidades depende de los atributos de todas las alternativas en el conjunto que contiene a i y m . Notar que la relación no depende de los atributos de las alternativas en los conjuntos que contienen las alternativas i y m .

El parámetro λ_S puede variar en los diferentes conjuntos, reflejando distintas correlaciones entre los factores no observados en cada conjunto. Si todos los $\lambda_S \forall S$ están entre 0 y 1, significa que el modelo es consistente con la maximización de la utilidad para todos los posibles valores de las variables explicativas. Para λ_S mayores que uno, el modelo es consistente con el comportamiento de maximización de la utilidad para exclusivamente un rango de las variables explicativas. Modelos con $\lambda_S < 0$ son inconsistentes con la maximización de la utilidad.

3.4. Medidas de bondad de ajuste

Una vez entrenados los distintos modelos con sus respectivos parámetros debe de hacerse la comprobación de cuál de estos modelos simula de manera óptima a los datos observados y para valorar lo anterior es necesario medir su eficacia predictiva mediante las medidas de bondad de ajuste [16].

3.4.1. Métricas de evaluación en predicciones discretas

Usualmente los modelos de clasificación generan dos tipos de predicciones [20]: una es continua y que es una probabilidad que varía entre 0 y 1, por otra parte, la que es discreta y que es una clase. Por lo que, a continuación, se presentan una serie de medidas que evalúan la confianza del modelo en el momento de hacer predicciones en un conjunto de observaciones nuevas para el modelo llamado conjunto de prueba.

3.4.1.1. Matriz de confusión

Suponiendo que el problema logra predecir sobre dos clases: positivo y falso, en la matriz de confusión pone la cantidad de positivos y falsos correcta e incorrectamente predichos, comparándolos con los observados tal y como se muestra en la tabla 2:

Tabla 2. Matriz de confusión de dos clases

Predichos	Observados	
	Positivo	Falso
Positivo	TP	FP
Falso	FN	TN

Fuente: Tabla extraída de [20]

En donde:

TP: indica la cantidad de verdaderos positivos.

FN: indica la cantidad de falsos negativos.

FP: indica la cantidad de falsos positivos.

TN: indica la cantidad de verdaderos negativos.

A partir de los indicadores anteriores y una serie de relaciones se pueden obtener otros estadísticos con los que puede conocerse el rendimiento del modelo y/o sesgo que pueda tener. Simplificando, para interpretar la matriz de confusión se utilizarán las métricas sensibilidad, especificidad y precisión [21]:

$$\text{Sensibilidad} = \frac{TP}{TP + FN} = \frac{TP}{\text{Total Positivos Observados}}$$

(Eq. 24)

$$Especificidad = \frac{TN}{TN + FP} = \frac{TN}{Total\ Negativos\ Observados}$$

(Eq. 25)

$$Precisión = \frac{TP}{TP + FP} = \frac{TP}{Total\ Positivos\ Predichos}$$

(Eq. 26).

Siendo la sensibilidad la tasa de verdaderos positivos; la especificidad la tasa de verdaderos negativos. Otras métricas que evalúan los componentes de la matriz de confusión como el *accuracy* o exactitud refleja la relación entre las clases predichas correctamente versus el total de observaciones, esta métrica puede presentar una desventaja por no distinguir el tipo de errores que se presentan y puede devolver resultados engañosos [20], especialmente cuando se presentan clases con desbalanceo. Si los datos presentan desbalanceo se recomienda usar la métrica *F1-score* que corresponde a una media armónica entre precisión y sensibilidad.

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} = \frac{Predichos\ correctamente}{Total\ observados}$$

(Eq. 27)

$$F1\ Score = \frac{2 \times Precisión \times Sensibilidad}{Precisión + Sensibilidad} = \frac{2TP}{2TP + FP + FN}$$

(Eq. 28).

3.4.1.2. Curva Roc

La curva de características de funcionamiento del receptor (ROC por sus siglas en inglés) es una representación gráfica de la sensibilidad versus la especificidad de un modelo de clasificación [22]. En otras palabras, la curva sirve para interpretar la tasa de verdaderos positivos versus la tasa de falsos positivos según se varíe el umbral de discriminación. En general, la curva ROC sirve para interpretar que tan bien el modelo es capaz de discriminar tanto una clase como la otra o del resto para modelos de elección múltiple. Se explica la construcción de la curva ROC tomando como base la ilustración 5, donde se observan las distribuciones de positivos y negativos, adicionalmente, hay un umbral que discrimina las clases en función de la probabilidad de pertenecer a la clase positiva.

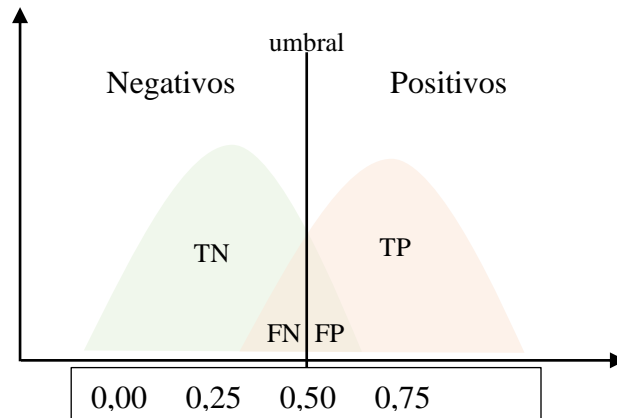


Ilustración 5. Introducción a la curva ROC

Fuente: elaboración propia

Observar que, al disminuir el umbral, el modelo predecirá una mayor cantidad de positivos aumentando con eso la sensibilidad, pero disminuyendo la especificidad; en un sentido opuesto, al aumentar el umbral, el modelo predecirá una mayor cantidad de negativos disminuyendo con eso la sensibilidad, pero aumentando su especificidad.

Si variamos los umbrales y representamos diferentes valores de sensibilidad frente a (1-especificidad) obtenemos la curva ROC. Es por eso, por lo que puede decirse que la curva ROC está compuesta por la tasa de verdaderos positivos y por la tasa de falsos positivos.

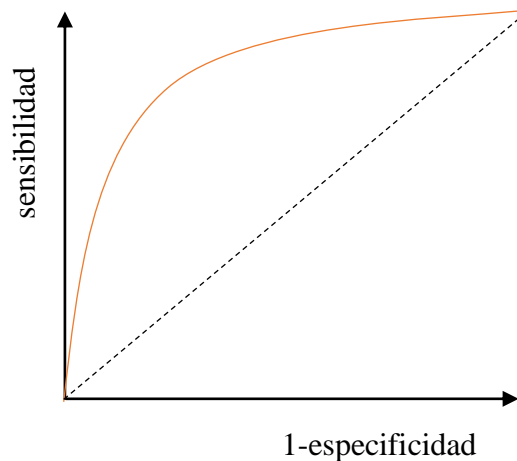


Ilustración 6. Esquema de la Curva ROC

Fuente: elaboración propia

Dado que la curva ROC permite visualizar para infinitos umbrales como funcionan las infinitas formas de clasificar, es una herramienta muy eficaz para comparar distintos modelos, basta con observar el área bajo la curva (AUC por sus siglas en inglés).

Un método que es capaz de distinguir perfectamente las distribuciones de los positivos y negativos, tendrá un AUC de 1, conforme el área se va reduciendo, el clasificador reduce su capacidad de clasificar. En el peor de los casos, tendría un modelo que clasifica completamente al azar, es decir $AUC = 0,5$ (línea punteada en la ilustración 6).

4. CAPITULO 4 – OTROS MÉTODOS DE CLASIFICACIÓN

Uno de los beneficios que presenta la teoría de la utilidad aleatoria es que esta tiene en cuenta cuales son las variables significativas para cada modo de transporte, por lo que se hace más fácil y práctica la interpretación de sus parámetros, pues las funciones de utilidad están establecidas de manera lineal. Sin embargo, puede ser que la capacidad predictiva (*accuracy* en datos de prueba) de los modelos clásicos utilizados en ella se vea inferior frente a la de otros modelos de clasificación clásicos del *machine learning*, como son: redes neuronales o el *Random Forest*.

Hasegawa et. al. (2013) [23] modelizaron un modelo híbrido de *Random Forest* y un algoritmo genético para modelizar la elección del modo de transporte que frente al modelo multinomial logit presentaba un mayor potencial de clasificación.

Para conceptualizar el *Random Forest*, es necesario introducir el concepto de árbol de clasificación.

4.1. Árbol de clasificación

El principio detrás de los árboles de clasificación es el de dividir un conjunto de datos en subconjuntos cada vez más pequeños y homogéneos, utilizando criterios de selección de variables y valores de corte. De esta manera, se pueden identificar patrones y relaciones entre las variables que permiten clasificar los objetos en diferentes categorías.

El concepto surgió basándose en la estructura de un árbol invertido [24] que se compone de una raíz, luego extendiéndose hacia abajo a sus ramas y generalmente se interpreta o lee de izquierda a derecha. El nodo inicial se llama nodo raíz, siguiendo a sus nodos internos que de donde se derivan las ramas del árbol, llegando así, hasta los nodos extremos llamados nodos hoja.

En la ilustración 2, se muestra un esquema sencillo de un árbol de clasificación, que se explica así: primero que todo, viendo el esquema izquierdo, se definirán los puntos negros (•) como puntos N y los que tiene el símbolo de suma en un círculo (\oplus) como puntos S. Luego, en el esquema derecho se observa el árbol de clasificación construido a partir de los datos del esquema izquierdo. Se observa que en la raíz está escrito el total de puntos $N=20$ y puntos $S=10$, posteriormente se subdivide en los nodos internos [$X \leq 3$] con puntos $\{N=5$ y $S=10\}$ del que se divide hacia los nodos hoja [$Y \leq 3$] con puntos $\{N=4$ y

$S=1$ y $[Y > 3]$ con puntos $\{N=9$ y $S=1\}$, el nodo $[X > 3]$ con puntos $\{N=15$ y $S=0\}$ como nodo hoja pues es un nodo extremo.

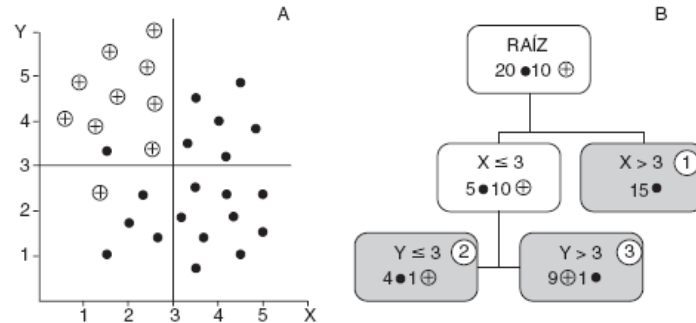


Ilustración 7. Esquema de partición dicotómica en la construcción de un árbol de decisión

Nota: estructura simple de un árbol de clasificación dicotómica en la construcción de un árbol de decisión.

Fuente: imagen extraída de [25]

4.2. Random Forest

El *Random Forest* (Bosque aleatorio) puede ser un modelo tanto de regresión, como de clasificación. En este trabajo nos centraremos en la parte de clasificación [26]. Este está formado por un conjunto (*ensemble*) de árboles de clasificación individuales, a cada árbol se le entrena a partir de muestreos aleatorios obtenidos de los datos de entrenamiento mediante *bootstrapping*, que es un método de aleatorización de muestras basado en una variante del algoritmo del vecino más cercano k , en la que solo las observaciones que forman parte del mismo nodo terminal tienen influencia en la observación predicha. Lo que significa que cada árbol que integra el *Random Forest* se entrena con datos ligeramente distintos. Como se explicó en la sección anterior, en cada árbol individual las observaciones se van distribuyendo por los nodos, generando la estructura de un árbol hasta alcanzar los nodos hojas o extremos. El *Random Forest* puede combinar diferentes métodos para crear los conjuntos de árboles de decisión, los dos más utilizados son el *Bagging* y el *Boosting*.

El *Bagging* [27] es un tipo estrategia de aprendizaje conjunto que “ensambla” múltiples modelos de aprendizaje sencillos llamados *weak learners* para construir un modelo de aprendizaje más robusto.

El *Boosting* [28] es un algoritmo de aprendizaje conjunto que construye clasificadores sucesivos. Al igual que el método *Bagging*, el *Boosting* funciona como una estrategia de aprendizaje conjunta que construye un clasificador con mayor rendimiento combinando modelos de aprendizaje sencillos, en contraste.

En este trabajo se incluirá el clasificador *Random Forest* clásico para compararlo con el modelo logit multinomial.

4.3. Redes Neuronales para clasificación

Se ha podido demostrar que el sistema nervioso humano se compone de células individuales, a las que se les llama neuronas [29], las cuáles se conectan entre sí, creando redes neuronales por donde los seres humanos y otros seres vivos transfieren y procesan la información y aprendizaje. En 1936, Allan Turing inició a trabajar con redes neuronales de forma artificial para diferentes aplicaciones. Hoy en día, son abundantes las publicaciones de trabajos, investigaciones y avances en la materia de redes neuronales artificiales.

4.3.1. Modelos de Aprendizaje profundo o *Deep learning*

Los modelos introducidos en este trabajo pertenecen al paradigma del Aprendizaje de máquinas o *Machine Learning*, cuyo principio [30] es el de “crear algoritmos que analizan datos, aprenden de ellos y luego aplican lo aprendido para tomar decisiones informadas”. Cabe aclarar que los algoritmos de aprendizaje de máquinas pueden ser intervenidos o ajustados para mejorar su rendimiento.

Por otra parte, el aprendizaje profundo, puede ser considerado una rama o extensión del aprendizaje de máquinas, su diferencia radica en que este tipo de algoritmos que funcionan con redes neuronales se autoajustan para alcanzar mayores niveles de rendimiento, pudiendo alcanzar predicciones tan potentes que son capaces de analizar datos etiquetados en tablas, imágenes, videos, sonidos, texto, etc.

Se introducirá un modelo de aprendizaje profundo de clasificación para comparar el rendimiento del modelo logit multinomial.

4.3.2. Modelo de Redes neuronales

A una red neuronal [31] se le puede considerar como un sistema que permite establecer una relación de entradas y salidas inspiradas en el sistema nervioso. Las unidades básicas de procesamiento son las llamadas neuronas, que generalmente están organizadas en capas. En la ilustración 8, se observa una red neuronal. A la izquierda está la capa de entrada con sus neuronas en color naranja, que representa los campos de entrada (datos de entrada o variables explicativas), sucesivamente la red tiene tres capas ocultas con sus neuronas en color verde, para finalizar con una capa de salida con una neurona en color rojo, que representa el campo de salida o campo de destino.

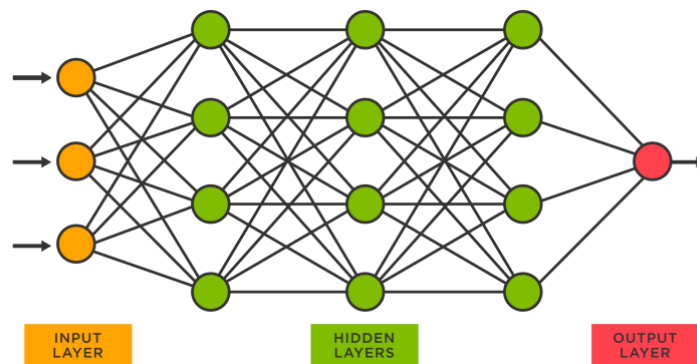


Ilustración 8. Diagrama de Redes neuronales artificiales

Nota: estructura simple de una red neuronal con sus capas de entrada, capas ocultas y capa de salida.

Fuente: imagen extraída de [32]

Cada conexión entre neuronas es un parámetro de la red y cada capa tiene unos parámetros llamados *bias*. La red inicia estableciendo unos parámetros iniciales aleatorios para todas sus neuronas, posteriormente, los datos de entrada se propagan por toda la red mediante operaciones simples con sus respectivos parámetros, hasta obtener unos valores predichos. Internamente, con una función de pérdida [33], la red hace una comparación del valor predicho y la variable de salida, de la que obtiene un error que posteriormente se retropropaga por la red, ajustando así sus parámetros para volver a comenzar, a todo este proceso se le llama entrenamiento y es el que la red utiliza para “aprender”. Este entrenamiento se repite hasta alcanzar la convergencia de aprendizaje, teniendo como objetivo minimizar la función de pérdida.

En la [sección 5.2.12](#) se explican en detalle cuales fueron los hiperparámetros y supuestos que se tuvieron en cuenta para la configuración de la red neuronal utilizada.

5. CAPITULO 5 – EXPERIENCIA COMPUTACIONAL

En este capítulo, se hará una exposición de la experiencia de haber implementado los conceptos de los [capítulo 3](#) y [capítulo 4](#) de este trabajo en el lenguaje de programación Python implementando distintas librerías o paquetes estadísticos y/o de apoyo matemático y visual. Con el fin de poder medir el rendimiento de los modelos logit en comparación con los modelos *Random Forest* y redes neuronales. Para explicar mejor el procedimiento y los resultados obtenidos se dividirá este capítulo en las siguientes etapas:

- Etapa 1: se presentarán los datos junto con una breve exploración y un análisis descriptivo de la distribución de sus variables.
- Etapa 2: se explicarán cuáles son los modelos implementados, hallazgos interesantes, problemas y soluciones.

5.1. Etapa 1. Exploración de datos y análisis descriptivo de los datos

En la [sección 2.3](#) fueron enumeradas las variables explicativas y la variable dependiente, por lo que se procede a hacer una inspección del conjunto de datos.

5.1.1. Depuración de datos

Como se mencionó antes, existen dos conjuntos de datos, el primero contiene la información proveniente de la EPR (encuesta de preferencias reveladas) y una base de datos con la información de costes, tiempos y distancias de los modos de transporte. Existen cuatro modos de transporte, pero se eliminarán todas aquellas observaciones del modo “*A_pie*”, debido a que las distancias entre las zonas del ámbito de estudio no son cortas, por lo que el modelo consistirá en predecir el modo que un usuario escogería de entre: *coche*, *bus* y *bici_patinete*. El nuevo conjunto de datos se observa en la tabla 3 y queda con 1587 observaciones, después de haber eliminado las observaciones correspondientes al modo “*A_pie*”.

Tabla 3. Conjunto de datos de EPR

ID_ENCUESTA	SEXO	EDAD	ESTUDIOS	Zona_Origen	Zona_Destino	Viaje_obligado	Motivo	Modo_txt	Orig_Dest_zona	
0	26	MUJER	33	Estudios primarios acabados	1	4	No Obligado	HBO	Coche	1_4
1	27	MUJER	35	Estudios universitarios superiores acabados	1	15	No Obligado	HBO	Coche	1_15
2	28	MUJER	35	Estudios universitarios superiores acabados	1	15	No Obligado	HBO	Coche	1_15
3	29	HOMBRE	52	Formación profesional	1	19	Obligado	HBW	Coche	1_19
4	30	MUJER	69	Estudios universitarios superiores acabados	1	19	No Obligado	HBO	Coche	1_19
...
1582	2229	HOMBRE	57	Estudios primarios acabados	120	107	Obligado	HBW	Coche	120_107
1583	2230	MUJER	40	Estudios secundarios acabados	120	107	Obligado	HBW	Coche	120_107
1584	2232	HOMBRE	38	Estudios universitarios superiores acabados	120	111	Obligado	HBW	Coche	120_111
1585	2235	HOMBRE	57	Estudios universitarios superiores acabados	120	116	Obligado	HBW	Coche	120_116
1586	2236	HOMBRE	57	Estudios universitarios superiores acabados	120	116	Obligado	HBW	Coche	120_116

1587 rows x 10 columns

Fuente: elaboración propia

El segundo conjunto de datos contiene las variables que describen la información de tiempo y coste en función de la distancia que tienen las zonas i -ésimas, respecto a las j -ésimas como se muestra en la tabla 4.

Tabla 4. Costes y tiempos por modos entre las zonas i -ésimas a las zonas j -ésimas

	o	d	distance_km	Tiempo_CAR	Tiempo_BUS	Tiempo_BP	Coste_CAR	Coste_BUS	Coste_BP	o_d
0	1	1	0.00	14.38	18.46	5.00	2.50	1.00	0.00	1_1
1	1	2	101.97	77.77	136.73	334.36	27.99	16.30	1.02	1_2
2	1	3	14.51	19.54	32.72	51.85	6.13	3.18	0.15	1_3
3	1	4	16.37	24.47	38.35	57.86	6.59	3.46	0.16	1_4
4	1	5	40.09	36.79	62.46	134.48	12.52	7.01	0.40	1_5
...
14395	120	116	12.25	20.48	37.75	56.45	6.94	3.66	0.18	120_116
14396	120	117	10.95	24.11	30.14	50.96	6.47	3.38	0.16	120_117
14397	120	118	1.30	11.69	18.08	10.46	2.97	1.28	0.02	120_118
14398	120	119	7.20	19.47	23.48	35.22	5.11	2.57	0.10	120_119
14399	120	120	0.00	11.31	15.15	5.00	2.50	1.00	0.00	120_120

14400 rows x 10 columns

Fuente: elaboración propia

El próximo paso consiste en agregar al primer conjunto de datos la información de distancia, tiempo y coste para cada modo de transporte en función del origen y destino.

5.1.2. Inspección de variable dependiente

La ilustración 9 muestra un gráfico de barras con las cantidades de usuarios que han escogido cada modo de transporte en el conjunto de datos.

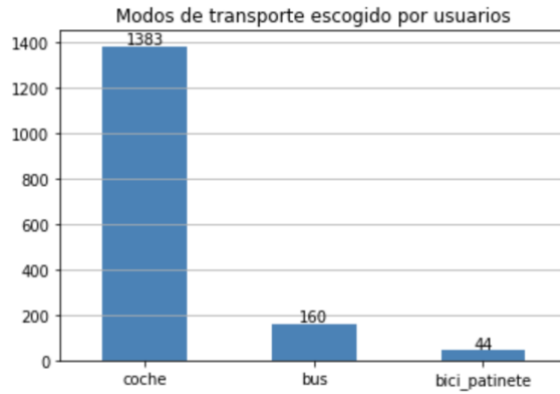


Ilustración 9. Gráfico de barras de la variable dependiente: modo de transporte

Fuente: elaboración propia

Se puede apreciar un desbalance en las clases, pues la clase “*coche*” representa el 87% de las respuestas. En la siguiente etapa, específicamente en la [sección 5.2.1](#), se hará un balanceo de las clases.

5.1.3. Inspección de variables explicativas continuas

Las variables continuas explicativas son las siguientes: *EDAD*, *DIST_KM*, *TIEMPO_CAR*, *COSTE_CAR*, *TIEMPO_BUS*, *COSTE_BUS*, *TIEMPO_BP*, *COSTE_BP*. Para iniciar la inspección se revisa en la tabla 5, que detalla un resumen estadístico descriptivo de las variables.

Tabla 5. Resumen estadístico descriptivo de variables continuas explicativas

	EDAD	DIST_KM	TIEMPO_CAR	COSTE_CAR	TIEMPO_BUS	COSTE_BUS	TIEMPO_BP	COSTE_BP
count	1587.00	1587.00	1587.00	1587.00	1587.00	1587.00	1587.00	1587.00
mean	42.88	6.46	17.33	4.50	24.36	2.20	29.20	0.08
std	12.63	7.91	5.48	2.03	10.36	1.22	25.91	0.08
min	16.00	0.00	10.03	2.50	10.06	1.00	5.00	0.00
25%	33.00	1.50	13.61	3.04	17.55	1.33	11.29	0.02
50%	43.00	4.13	16.13	4.00	22.25	1.90	22.35	0.06
75%	51.00	8.28	19.68	5.50	28.91	2.80	39.76	0.12
max	80.00	69.95	59.89	19.99	104.84	11.49	230.94	0.70

Fuente: elaboración propia

Revisando los valores máximos, mínimos y promedios, intuitivamente no se observan valores atípicos extremos, para confirmar lo anterior se procede con la inspección visual mediante la ilustración 10.

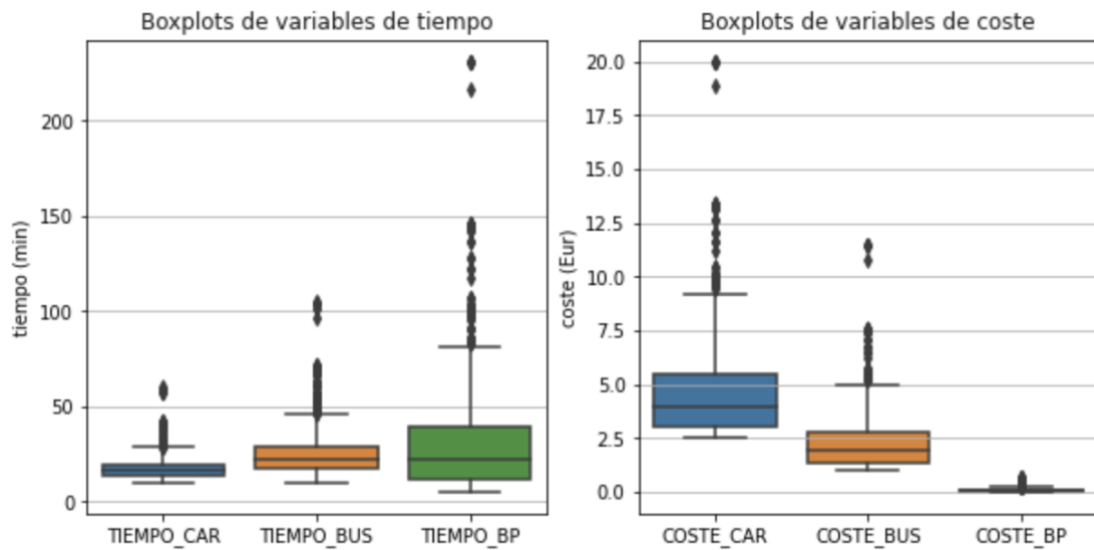


Ilustración 10. Boxplots de tiempo y costes de modos de transporte

Fuente: elaboración propia

Según los *boxplots*, el *coche*, *bus* y *bici_patinete* tienen en el mismo orden un menor rango de tiempos, pero opuestamente en el coste. También, que los viajes en *coche* y *bus* tienen un coste mínimo distinto de cero, como era de esperar.

Continuando con el análisis exploratorio visual en la ilustración 11, se muestran los *boxplots* de las variables distancia y edad del usuario.

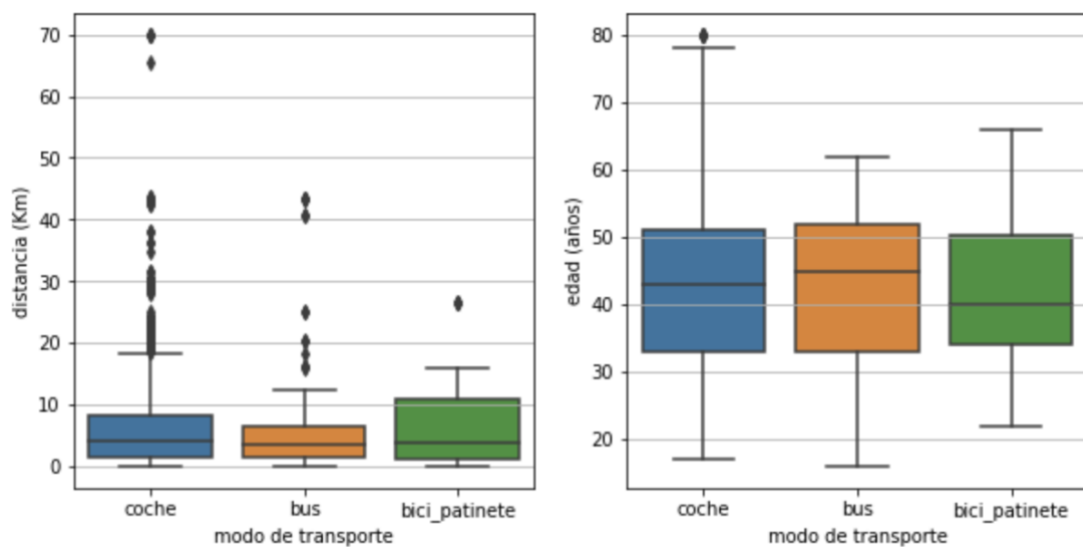


Ilustración 11. Boxplots de distancia del viaje y edad del usuario

Fuente: elaboración propia

No se observa algún comportamiento especial o anormal, por lo que se confirma que los datos de las variables continuas no tienen que ser modificados y/o ajustados y se procede a inspeccionar las variables categóricas explicativas.

Se puede observar que el 75% de las personas que viajan en *coche* recorren una distancia menor de 10 Km, el 75% de las personas que viajan en *bus*, recorren una distancia de aproximadamente 7 KM y los que viajan en *bici_patinete* recorren una distancia de 11Km. No se aprecia una diferencia marcada en los percentiles 25 a 75 de la edad de las personas que viajan en los distintos modos de transporte.

5.1.4. Inspección de variables explicativas categóricas

Las variables categóricas están definidas por: *SEXO*, *ESTUDIOS*, *VIAJE_OBLIGADO*, *MOTIVO*.

En la tala 6, se representan las frecuencias de la variable *SEXO*, observando que en la muestra existen 945 personas de sexo masculino y 642 de sexo femenino.

Tabla 6. Frecuencias de clases de variable *SEXO*

clase	frecuencias
<i>m</i>	945
<i>f</i>	642

Fuente: elaboración propia

En la tabla 7 se representan las frecuencias de la variable *ESTUDIOS*. En esta tabla se observan dos columnas: clase inicial y clase definitiva, la primera contiene el antiguo nombre de las clases y la segunda el nombre al que se ha actualizado, más que todo para abreviar la longitud de los nombres y/o agregar varias clases en una sola. Las clases que inicialmente estaban planteadas como estudios universitarios superiores y medios se han agregado en una sola clase llamada *Univers_terminada*. A excepción de las clases *sin_estudios* y *formacion_profesional* se puede notar que las frecuencias van disminuyendo conforme aumenta el nivel de estudios.

Tabla 7. Frecuencias de clases de variable ESTUDIOS

Clase inicial	Clase definitiva	frecuencias
<i>Sin estudios</i>	<i>sin_estudios</i>	46
<i>Estudios primarios acabados</i>	<i>Primaria_terminada</i>	455
<i>Estudios secundarios acabados</i>	<i>Secundaria_terminada</i>	402
<i>Formación profesional</i>	<i>Formación_profesional</i>	315
<hr/>		
<i>Estudios universitarios superiores acabados</i>	<i>Univers_terminada</i>	336
<i>Estudios universitarios medios acabados</i>		
<hr/>		
<i>Posgrado, Máster, Doctorado</i>	<i>posgrado</i>	33

Fuente: elaboración propia

En la tabla 8 se representa los motivos de los viajes, se han agregado los motivos hbw (*home based work*: basado en el hogar y trabajo) y hbs (*home based study*: basado en el hogar y estudios) en uno solo hbws (*home based work/study*: basado en el hogar, trabajo/estudio), para tener en una sola clase los viajes con los motivos que usualmente se dan a la misma hora y frecuencia, este nuevo motivo tiene 749 observaciones, seguido del motivo hbo (*home based other*: basado en el hogar distinto al trabajo o estudio) con 649 observaciones y por último, el motivo nhb (*non home based*: que son viajes no basados en el hogar) con 189 observaciones. Es importante recalcar que este comportamiento se da por que la encuesta ha sido realizada en un día de semana, evidentemente los fines de semana tendrán un comportamiento distinto.

Tabla 8. Frecuencias de clases de variable MOTIVO

Clase inicial	Clase definitiva	frecuencias
<i>hbw</i>	<i>hbws</i>	749
<i>hbs</i>		
<hr/>		
<i>hbo</i>	<i>hbo</i>	649
<i>nhb</i>	<i>nhb</i>	189

Fuente: elaboración propia

En la tabla 9 se observan las frecuencias de la variable VIAJE_OBLIGADO, de la que 808 personas realizan viajes obligados, por el estudio, trabajo, responsabilidades

adquiridas, y que 779 personas realizaron viajes no obligados, es decir, que tenían la libertad de viajar o no.

Tabla 9. Frecuencias de clases de variable VIAJE_OBLIGADO

clase	frecuencias
<i>Si</i>	808
<i>no</i>	779

Fuente: elaboración propia

En la ilustración 12, están representadas las variables categóricas con gráficos de tipo mosaico, en el que se observan las frecuencias relativas. Pueden observarse, tanto las frecuencias relativas, como el modo de transporte escogido por cada usuario y es interesante notar algunas situaciones, por ejemplo: en la variable *SEXO*: no se observaron personas de género *femenino* utilizando *bici_patinete*. En la variable *ESTUDIOS*, no se observaron usuarios de *bici_patinete* en las clases *sin_estudios* y *posgrado*. El resto de las clases tienen comportamientos semejantes entre sí. En la variable *VIAJE_OBLIGADO* existen comportamientos esperables y en la variable *MOTIVO*: específicamente en el motivo *nhb*, menos del 5% utilizan el *bus*, el resto usa el *coche*. Después de los comentarios anteriores se observa que las clases de todas las variables se comportan de manera similar y en función de lo que se puede esperar cuando se tiene una variable de modo de transporte (*coche*, *bus* y *bici_patinete*) desbalanceada.

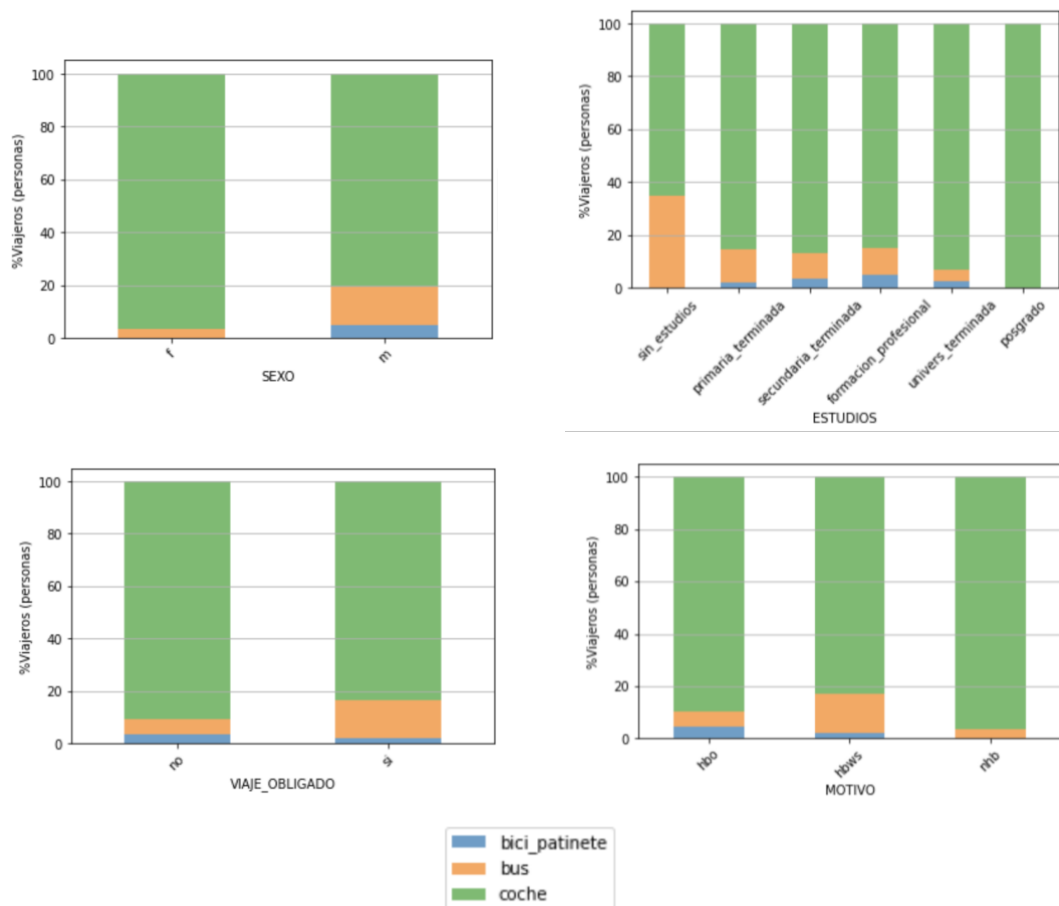


Ilustración 12. Gráficos de frecuencias relativas de las variables categóricas

Fuente: elaboración propia

Debido a que los modelos que se implementarán deben recibir datos de tipo numéricos para que sean correctamente interpretadas, se procede a convertir las variables categóricas en variables *dummy* [34], que es un proceso mediante el cual se crean variables ficticias, convirtiendo las variables categóricas en múltiples variables binarias que indican la presencia o ausencia de una clase específica para cada observación. Por cada variable categórica con K categorías se ha de crear $K - 1$ variables dummy quedando en el intercepto la categoría de referencia.

5.2. Etapa 2. Ajuste de Modelos de reparto modal

5.2.1. Preparativos para ajuste de modelos de reparto modal

Para iniciar con la calibración o ajuste de los modelos de reparto modal se dividirá aleatoriamente el conjunto de datos en dos subconjuntos, el primero, el de entrenamiento (*train*) que tendrá el 70% de los datos iniciales, y el segundo, el de prueba (*test*) con el

30% de los datos restantes, los datos mantendrán la misma estratificación de la variable dependiente en el conjunto de datos antes de hacer la división.

Los modelos se entrenarán con el primer subconjunto mencionado, posteriormente se probará con el subconjunto de prueba para medir la capacidad predictiva de los mismos. Para conseguir el balanceo de los datos, se procederá a aplicar la función *RandomOverSampler* de la librería de Python *imbalanced learn* [35].

Los datos desbalanceados se utilizarán tanto en los primeros dos modelos, como en el modelo 5, cuyos datos serán transformados mediante el PCA (*Principal Component Analysis* por sus siglas en inglés), luego en todos los demás se utilizarán los datos balanceados. Los primeros cuatro modelos serán logit multinomial sin transformaciones, en los próximos tres, se harán distintas transformaciones para encontrar el mejor ajuste posible. Y, por último, se ajustarán unos modelos Random Forest y una red neuronal para comparar los rendimientos de las predicciones de los modelos logit. En la tabla 10, se describe cuáles serán los ajustes que se realizarán a cada uno de los modelos.

Tabla 10. Modelos a ajustar

Modelo	Tipo de modelo	Datos balanceados	Datos transformados	Parámetros no significativos depurados
1		No	No	No
2	MNL	No	No	Sí
3		Sí	No	No
4		Sí	No	Sí
5.1	MNL: cuadrados	Sí	Sí	Sí
5.2	MNL: raíces cuadradas	Sí	Sí	Sí
5.3	MNL: Box Cox	Sí	Sí	Sí
5.4	MNL: PCA	No	Sí	N/A
6	Red neuronal	Sí	No	N/A
7	Random Forest	Sí	No	N/A

Fuente: elaboración propia

Nota: MNL significa Modelo Logit Multinomial

Los parámetros no significativos depurados son todos aquellos parámetros que al ajustarse los modelos resultaron no significativos y por eso se eliminaron de los modelos, para comparar la capacidad predictiva de los mismos con y sin parámetros no significativos. Únicamente, en los modelos logit 1 y 3 no se eliminó ningún parámetro, tampoco se eliminaron parámetros de la red neuronal ni del *Random Forest*.

5.2.2. Modelo 1: MNL

El modelo 1, consiste en un modelo logit multinomial, que será entrenado con los datos desbalanceados, sin transformaciones y sin hacer depuraciones, es decir, no se eliminarán todas aquellas variables que no sean significativas.

Las funciones de utilidad V_{Jn} de modelo se expresan con las siguientes ecuaciones:

$$\begin{aligned}
 V_{CAR} = & ASC_{CAR} + \beta_{COST_{CAR}} * COSTE_{CAR} + \beta_{TIME_{CAR}} * TIEMPO_{CAR} + \\
 & \beta_{EDAD_{CAR}} * EDAD + \beta_{SEXO_m_{CAR}} * SEXO_m + \beta_{DIST_{CAR}} * DIST_{KM} + \\
 & \beta_{MOTIVO_{hbo_{CAR}}} * MOTIVO_{hbo} + \beta_{MOTIVO_{hbws_{CAR}}} * MOTIVO_{hbws} + \\
 & \beta_{ESTUDIOS_{pri_termi_{CAR}}} * ESTUDIOS_{primaria_terminada} + \\
 & \beta_{ESTUDIOS_{sec_termi_{CAR}}} * ESTUDIOS_{secundaria_terminada} + \\
 & \beta_{ESTUDIOS_{forma_pro_{CAR}}} * ESTUDIOS_{formacion_profesional} + \\
 & \beta_{ESTUDIOS_{uni_termi_{CAR}}} * ESTUDIOS_{univers_terminada} + \\
 & \beta_{ESTUDIOS_{posgrado_{CAR}}} * ESTUDIOS_{posgrado} + \varepsilon
 \end{aligned}$$

$$\begin{aligned}
 V_{BUS} = & ASC_{BUS} + \beta_{COST_{BUS}} * COSTE_{BUS} + \beta_{TIME_{BUS}} * TIEMPO_{BUS} + \\
 & \beta_{EDAD_{BUS}} * EDAD + \beta_{SEXO_m_{BUS}} * SEXO_m + \beta_{DIST_{BUS}} * DIST_{KM} + \\
 & \beta_{MOTIVO_{hbo_{BUS}}} * MOTIVO_{hbo} + \beta_{MOTIVO_{hbws_{BUS}}} * MOTIVO_{hbws} + \\
 & \beta_{ESTUDIOS_{pri_termi_{BUS}}} * ESTUDIOS_{primaria_terminada} + \\
 & \beta_{ESTUDIOS_{sec_termi_{BUS}}} * ESTUDIOS_{secundaria_terminada} + \\
 & \beta_{ESTUDIOS_{forma_pro_{BUS}}} * ESTUDIOS_{formacion_profesional} + \\
 & \beta_{ESTUDIOS_{uni_termi_{BUS}}} * ESTUDIOS_{univers_terminada} + \\
 & \beta_{ESTUDIOS_{posgrado_{BUS}}} * ESTUDIOS_{posgrado} + \varepsilon
 \end{aligned}$$

$$\begin{aligned}
 V_{BP} = & ASC_{BP} + \beta_{COST_{BP}} * COSTE_{BP} + \beta_{TIME_{BP}} * TIEMPO_{BP} + \\
 & \beta_{EDAD_{BP}} * EDAD + \beta_{SEXO_m_{BP}} * SEXO_m + \beta_{DIST_{BP}} * DIST_{KM} + \\
 & \beta_{MOTIVO_{hbo_{BP}}} * MOTIVO_{hbo} + \beta_{MOTIVO_{hbws_{BP}}} * MOTIVO_{hbws} + \\
 & \beta_{ESTUDIOS_{pri_termi_{BP}}} * ESTUDIOS_{primaria_terminada} + \\
 & \beta_{ESTUDIOS_{sec_termi_{BP}}} * ESTUDIOS_{secundaria_terminada} + \\
 & \beta_{ESTUDIOS_{forma_pro_{BP}}} * ESTUDIOS_{formacion_profesional} + \\
 & \beta_{ESTUDIOS_{uni_termi_{BP}}} * ESTUDIOS_{univers_terminada} + \\
 & \beta_{ESTUDIOS_{posgrado_{BP}}} * ESTUDIOS_{posgrado} + \varepsilon
 \end{aligned}$$

Luego, para el cálculo de probabilidad de cada uno de los modos para cada usuario se introducen las funciones de utilidad en la Eq. 9 y se obtiene la siguiente expresión:

$$P(J_n | \beta, X_{k_n}) = \frac{e^{V_{Jn}}}{e^{V_{COCHE_n}} + e^{V_{BUS_n}} + e^{V_{BP_n}}}$$

(Eq. 29).

Recordando que J_n , representa las J alternativas disponibles para el usuario n ; X_{k_n} los K regresores. Para el cálculo de los parámetros se han utilizado dos librerías: Pylogit [36] Biogeme [37] y se ha constatado que devuelven los mismos valores para los parámetros, encontrando en algunos casos unas pequeñas diferencias en las centésimas.

Este modelo queda planteado con todas las variables, incluyendo aquellas no significativas.

5.2.3. Modelo 2: MNL

Este modelo parte de las mismas asunciones que el modelo 1, con la diferencia que en este se han eliminado las variables no significativas. Para conseguir la depuración de las variables no significativas se han eliminado manualmente, una a una, iniciando por la peor, es decir, aquella que tenía el mayor p-valor y finalizando hasta que todas las variables no tenían un p-valor mayor que el 5% de significancia.

Se han eliminado los parámetros: $\beta_{TIME_{CAR}}$, $\beta_{COST_{CAR}}$, $\beta_{TIME_{BUS}}$, $\beta_{EDAD_{BUS}}$, $\beta_{COST_{BP}}$, $\beta_{TIME_{BUS}}$, $\beta_{DIST_{BUS}}$, $\beta_{DIST_{CAR}}$, $\beta_{DIST_{CAR}}$, $\beta_{EDAD_{BP}}$, $\beta_{TIME_{BP}}$, $\beta_{MOTIVO_{hbws_{BUS}}}$, $\beta_{DIST_{BP}}$, $\beta_{EDAD_{CAR}}$. Es interesante notar que, para este conjunto de datos, se han eliminado los parámetros de tiempo, distancia y edad para todos los modos. El parámetro de coste únicamente resulta significativo para el modo autobús. Los parámetros iniciales ASC_j y los parámetros tanto de estudios como de sexo masculino han resultado significativos.

5.2.4. Modelo 3: MNL

Este modelo es una réplica del modelo 1, con la diferencia que en este se utilizarán datos de entrenamiento con las clases de la variable dependiente balanceadas tal y como se explica en la [sección 5.2.1](#).

5.2.5. Modelo 4: MNL

Este modelo consiste en el modelo 3, con los parámetros no significativos depurados. Para este caso se han eliminado los parámetros siguientes: $\beta_{EDAD_{BP}}$, β_{BP} , β_{BUS} , $\beta_{SEXO_m_{BUS}}$, $\beta_{EDAD_{CAR}}$, $\beta_{ESTUDIOS_{posgrado_{BP}}}$, $\beta_{MOTIVO_{hbws_{BUS}}}$, $\beta_{COST_{BP}}$.

5.2.6. Modelo 5.1: MNL con transformación cuadrática

Se utilizará el conjunto de datos de entrenamiento balanceados para ajustar los modelos con transformaciones. Los parámetros no significativos se han eliminado siguiendo la misma metodología que se ha seguido anteriormente.

Las variables de tipo continuo se transforman por cuadrados, así:

$$V_j = ASC_j + \beta_{1j}X_{1j}^2 + \dots + \beta_{Kj}X_{Kj}^2 + \beta_{Lj}\delta_{Lj}$$

(Eq. 30).

En donde:

j : los distintos modos de transporte.

X_{Kj} : las K variables de tipo continuo para el modo j .

δ_{Lj} : las L variables de tipo binarias para el modo j .

Con esta transformación se busca exagerar un poco el comportamiento de las variables continuas, así como captar posibles relaciones no lineales.

5.2.7. Modelo 5.2: MNL con transformación raíz cuadrada

Antes de introducir las variables de tipo continuo en el modelo se hace una transformación de raíz cuadrada, como se observa a continuación:

$$V_j = ASC_j + \beta_{1j}\sqrt{X_{1j}} + \dots + \beta_{Kj}\sqrt{X_{Kj}} + \beta_{Lj}\delta_{Lj}$$

(Eq. 31).

En donde:

j : los distintos modos de transporte.

X_{Kj} : las K variables de tipo continuo para el modo j .

δ_{Lj} : las L variables de tipo binarias para el modo j .

Con esta transformación se suaviza un poco las diferencias que pueda haber entre las distintas observaciones para cada variable continua, así como captar relaciones no lineales.

5.2.8. Modelo 5.3: MNL con transformación Box-Cox

La transformación Box Cox fue propuesta en 1964 por Cox & Box [38], su objetivo es transformar una serie de datos que no sigue una distribución normal. La transformación de Box-Cox utiliza una función que eleva los valores de la serie de datos a una potencia λ . La fórmula de transformación para $y_i > 0$ es la siguiente:

$$y_i^{(\lambda)} = \begin{cases} (y_i^\lambda - 1)/\lambda; & \lambda \neq 0 \\ \text{Log}(y_i); & \lambda = 0 \end{cases}$$

(Eq. 32).

Luego, a las variables de tipo continuo, se hizo la transformación de Box-Cox después de sumar un uno para evitar valores nulos, como sigue:

$$V_j = ASC_j + \beta_{1j}(X_{1j} + 1)^{(\lambda_1)} + \dots + \beta_{Kj}(X_{Kj} + 1)^{(\lambda_K)} + \beta_{Kj}\delta_{Lj}$$

(Eq. 33).

En donde:

j : los distintos modos de transporte.

X_{Kj} : K variable de tipo continuo para el modo j .

δ_{Lj} : L variable de tipo binarias para el modo j .

λ_K : parámetro lambda de transformación de la K variable de tipo continuo.

Con esta transformación se busca normalizar las distribuciones de las variables de tipo continuo. Orro et al [39] resumen las justificaciones realizadas por diversos autores sobre la implementación de la transformación Box-Cox en modelos logit a alguna o todas las variables continuas siempre y cuando sea positivas. Afirman que al aplicarse la transformación se obtiene una mayor flexibilidad, y como consecuencia esto supone un mejor ajuste (puede ser que esta mejoría en el ajuste no siempre sea significativa). En resumen, serán los datos los que establezcan si conviene o no la transformación, pero es importante mencionar que el modelo no debería limitarse a funciones de utilidad lineales. Pero, se aclara que la conclusión anterior se encuentra bajo discusión entre algunos expertos.

5.2.9. Modelo 5.4: MNL con Componentes Principales (PCA)

El cálculo de componentes principales es un método utilizado, entre otras cosas para reducir la complejidad de un conjunto de datos, manteniendo la mayor cantidad de información posible. A continuación, se explica el procedimiento de cálculo de componentes principales [40].

1. **Estandarización de datos:** se inicia estandarizando el conjunto de datos, es decir, se resta la media de cada variable y se divide por su desviación estándar. Dado que el PCA se basa en la varianza y covarianza de las variables, esta primera etapa garantiza que las variables estén en la misma escala, para que variables con unidades con mayor escala de medida que otras no tengan un mayor peso en el cálculo que las que tienen una escala de medida menor, por ejemplo: minutos frente a unidades monetarias.
2. **Cálculo de matriz de covarianzas:** se calcula la matriz de covarianza a partir de los datos estandarizados. La matriz de covarianza es una matriz cuadrada que indica la covarianza entre dos variables. La diagonal de la matriz indica la varianza de cada variable.
3. **Cálculo de vectores y valores propios:** se calcula a partir de la matriz de covarianza. Los vectores propios representan las direcciones principales en las que los datos varían, mientras que los valores propios indican la cantidad de varianza en los datos que se explica por cada vector propio.
4. **Selección de componentes principales:** se seleccionan las componentes principales a partir de los vectores y los valores propios. Las componentes principales se ordenan en función de su valor propio, de mayor a menor, y se seleccionan las primeras componentes principales que retengan la mayor cantidad de varianza posible.

Freire et al [41] exponen su experiencia utilizando las componentes principales de las variables explicativas como argumentos en un modelo logit binario. La utilización del PCA tiene beneficios al extraer estructuras latentes de los datos mediante las componentes principales. Una de estas ventajas es que puede ayudar a mitigar las debilidades del modelo logit multinomial, como la alta correlación entre variables.

Para este modelo se calcularán las componentes principales de las variables explicativas se concatenarán al conjunto de datos al que se implementará y se utilizarán como argumento de la función de utilidad, como se ve a continuación:

$$V_j = ASC_j + \beta_{1j}PC_{1j} + \beta_{2j}PC_{2j}$$

(Eq. 34).

En donde:

j: los distintos modos de transporte.

PC_{1j} , PC_{2j} : componentes principales de variables explicativas del conjunto de datos desbalanceado.

5.2.10. Modelos NL: NEST_NOCAR, NEST_NOBUS y NEST_NOBP

Tal como se explicó en la [sección 3.3](#), para que el modelo *Nested Logit* tenga sentido debe de haber una correlación entre las clases de las agrupaciones posibles, de otro modo el modelo se vuelve inconsistente.

Se realizaron tres pruebas para este modelo: dado que solo existen tres clases posibles (coche, bus y bici_patinete), la opción era agrupar dos clases y dejar libre el resto, es decir, para el modelo en el que se utilizó NEST_NOCAR, se agruparon las clases que no son coche: bus y bici_patinete y se dejó libre la clase coche, como se observa en la ilustración 13.

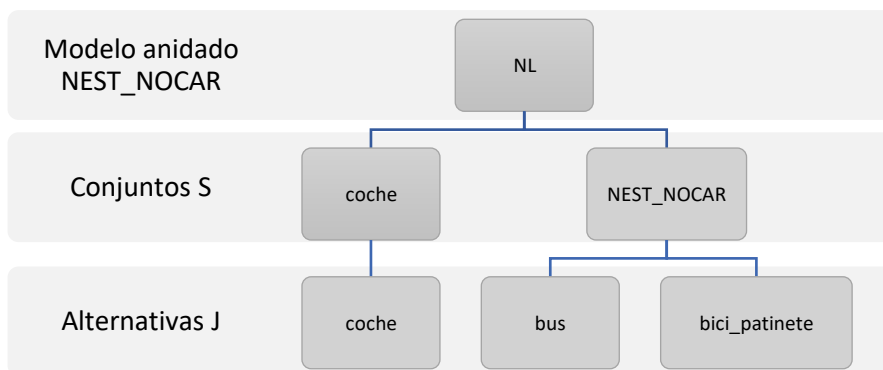


Ilustración 13. Configuración del modelo logit con la agrupación NEST_NOCAR

Fuente: elaboración propia

De la misma manera para el modelo en el que se utilizó la agrupación NEST_NOBUS se agruparon las clases que no son bus: coche y bici_patinete y se dejó libre la clase bus, como se observa en la ilustración 14.

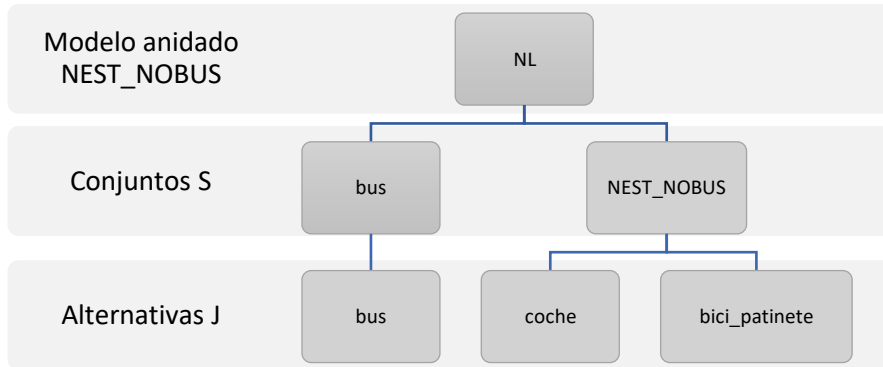


Ilustración 14. Configuración del modelo logit con la agrupación NEST_NOBUS

Fuente: elaboración propia

Y, por último, el modelo en el que se utilizó la agrupación NEST_NOBP, se agruparon las clases que no son bici_patinete: coche y bus y se dejó libre la clase bici_patinete, tal y como se observa en la ilustración 15.

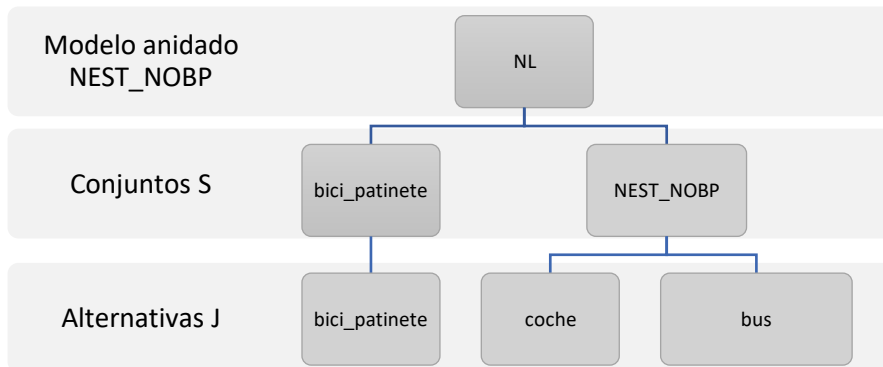


Ilustración 15. Configuración del modelo logit anidado NEST_NOBP

Fuente: elaboración propia

Como se mencionó al principio de esta sección, por la falta de correlación entre las clases agrupadas de los datos estudiados, el modelo *Nested Logit* no es apropiado. Sin embargo, se deja un enlace a un [notebook de Python](#) con una prueba de la misma metodología explicada del *Nested Logit*, pero que se ha realizado en otro conjunto de datos llamado *optima*, estos datos han sido provistos por la librería *Biogeme*.

5.2.11. Obtención de predicciones de modelos logit

Python es un lenguaje de programación orientado a objetos, y que está basado en librerías (o paquetes) de código abierto que han sido creadas con “clases”. Las clases permiten la empaquetación de datos [43] y funcionalidades juntos. Al crear una clase se crea un objeto, que es el modelo en este caso, permitiendo crear nuevas instancias de este tipo.

Se ha elaborado una clase propia en Python, llamada *Logit_model_pred* que se encarga de realizar predicciones de probabilidades de elección para modelos logit multinomiales y anidados. La clase contiene dos métodos principales: *probs_pred_MNL()* y *probs_pred_NL()*, que calculan las probabilidades de elección para modelos logit multinomiales y anidados, respectivamente que almacena en unos *dataframe* de *pandas*. En ambos métodos, la clase utiliza los coeficientes estimados del modelo (almacenados en un *dataframe* de *pandas*), los regresores y los modos de elección para generar las funciones de utilidad de cada alternativa. A continuación, estas funciones de utilidad se utilizan para calcular las probabilidades de elección. Para el modelo multinomial, mientras que para el modelo anidado se utiliza una versión modificada de la función que tiene en cuenta la estructura anidada del modelo.

La clase también utiliza varias funciones de la biblioteca *sklearn*, como la matriz de confusión, la curva ROC y el informe de clasificación, para evaluar el rendimiento de los modelos.

Y por último el gráfico *Comparative_plot* (se explica en [sección 6.1.4](#)) tiene como argumento *based_on* que corresponde a la clase a la que se desea observar el comportamiento de las predicciones y compararlo con las probabilidades del resto de las clases. Lo que hace este método es devolver una gráfica con las probabilidades del modo elegido en *based_on*, ordenadas de menor a mayor y coloreadas por el verdadero modo según el conjunto de datos de prueba. Esta gráfica es muy útil como herramienta de exploración para conocer los comportamientos de las predicciones (probabilidades de cada modo) de cada observación. Dado que las gráficas están ordenadas por las probabilidades de un modo especificado de menor a mayor, se grafican los modos reales y adicionalmente las probabilidades de los demás modos. Por cada observación existirán tres puntos que representa la probabilidad de cada modo, la suma de los tres lógicamente debe sumar uno.

5.2.12. Modelo 6: Red Neuronal

En esta sección se explicará cuáles fueron las configuraciones, criterios seguidos y los hiperparámetros utilizados para construir la red neuronal.

5.2.12.1. Configuración de la red neuronal

Este modelo de red neuronal se ha elaborado con la librería *TensorFlow* [48] en Python. En la ilustración 16 se muestra la configuración de la red neuronal. Se procede con la explicación.

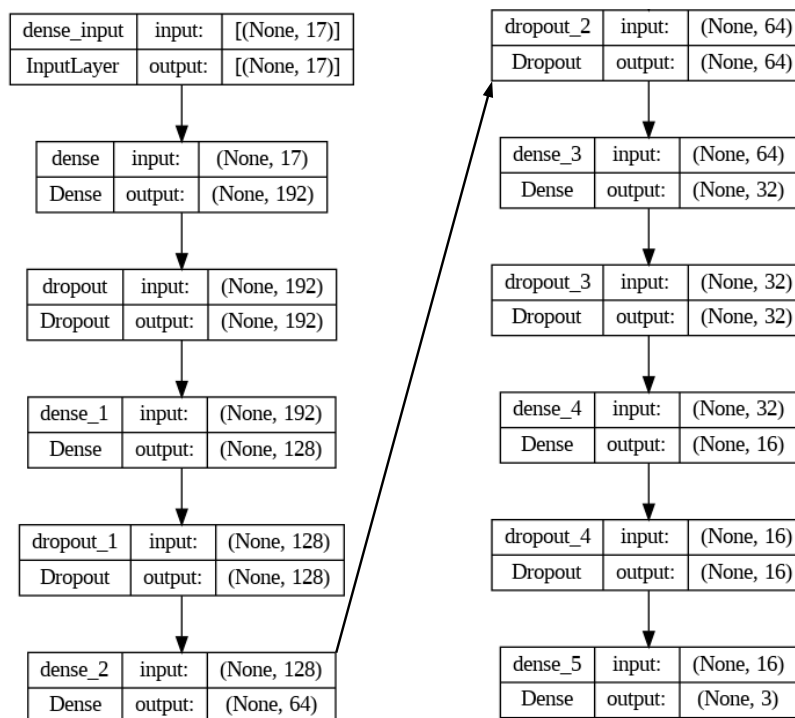


Ilustración 16. Configuración de red neuronal de clasificación

Fuente: elaboración propia con función *plot_model*

La red tiene una capa de entrada *dense_input* a la que se le introducen 17 variables, *SEXO_m*, *ESTUDIOS_primaria_terminada*, *ESTUDIOS_secundaria_terminada*, *ESTUDIOS_formacion_profesional*, *ESTUDIOS_univers_terminada*, *ESTUDIOS_posgrado*, *VIAJE_OBLIGADO_si*, *MOTIVO_hbo*, *MOTIVO_hbws*, *EDAD*, *DIST_KM*, *TIEMPO_CAR*, *COSTE_CAR*, *TIEMPO_BUS*, *COSTE_BUS*, *TIEMPO_BP*, *COSTE_BP*. Las variables continuas fueron normalizadas para introducirse en la red neuronal.

Consecutivamente, la red está compuesta de 5 capas densas ocultas con 192, 128, 64, 32 y 16 neuronas. Para evitar el sobreajuste del modelo se han agregado unas capas *Dropout* [49], cuya función consiste en eliminar aleatoriamente las conexiones entre neuronas de una capa y la siguiente, según la tasa de eliminación que se desee, para este modelo se ha utilizado una tasa de eliminación *Dropout* de 10%, es decir, entre capa y capa solo existe el 90% de las conexiones activas, aleatoriamente existe un 10% de conexiones apagadas. Por último, una capa de salida *dense_5*, la que devuelve las probabilidades para los modos de transporte a predecir, es decir probabilidades de *coche*, *bus* y *bici_patinete* para cada observación. La red consta de 39.075 parámetros, es decir las conexiones entre las neuronas de las capas y adicionalmente un parámetro por cada capa llamado *bias*. Todos estos parámetros son entrenables.

Los parámetros de la red neuronal no siempre pueden coger cualquier valor, más bien, su comportamiento está restringido a la función de activación que tenga. La función de activación [50] es la manera en la que se transmite la información por las conexiones de salida entre las neuronas. Existen distintos tipos de funciones de activación, para el caso de las capas de entrada y las capas ocultas de la red se utiliza la función de activación *reLU*.

La función de activación *reLU* [51] (*Rectified Linear Unit*) es un método no paramétrico que devuelve el mayor valor $R(z) = \max(0, z)$. Lo que hace la función es devolver el mismo valor si Z es positivo, en cambio, si es negativo devuelve un cero. La función *reLU* tiene algunas propiedades útiles como la esparsidad, lo que significa que solo un subconjunto de las neuronas en una capa dada se activa para una entrada dada. Otra propiedad útil de la función *reLU* es la inhibición lateral, con la que la red se asegura que cada neurona en una capa oculta se especialice en una característica específica, pero no a expensas de las neuronas vecinas, lo que ayuda a evitar la coactivación innecesaria de neuronas. En general, esta función es utilizada comúnmente en las capas ocultas debido a la eficiencia computacional y su capacidad para no introducir linealidad, en la ilustración 17 puede observarse la función.

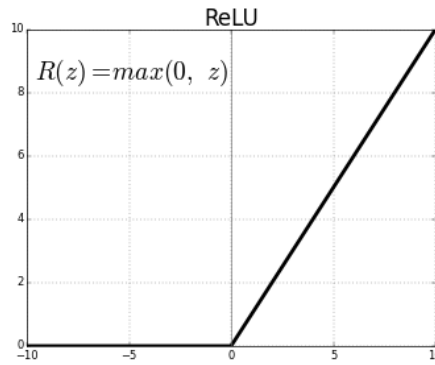


Ilustración 17. Esquema de una función de activación reLU

Fuente: imagen extraída de Casallas, 2020 [52]

La capa de salida tiene una función de activación distinta, esta es la *Softmax* [53], esta función se utiliza comúnmente en problemas de clasificación multiclase para transformar la salida de la capa actual en una distribución de probabilidades sobre las K clases. Formalmente para una entrada $Z = (Z_1, Z_2, \dots, Z_K)$, la función Softmax se define como:

$$Softmax(Z_j) = \frac{e^{Z_j}}{\sum_{k=1}^K e^{Z_k}}, \text{ para } j = 1, \dots, K$$

(Eq. 35).

En donde:

Z_j representa la entrada en el nodo o neurona actual.

J : es el nodo de la clase actual.

La función *Softmax* [53] genera una distribución de probabilidad sobre las posibles clases de salida, el valor más alto de esta distribución es la que se considera como la predicción de la red neuronal para la entrada dada. La función *Softmax* tiene la propiedad de ser diferenciable, lo que la hace adecuada para la optimización mediante algoritmos de descenso de gradiente.

Para finalizar con la explicación de la configuración de la red neuronal, en las capas de entrada se han utilizado regularizadores. Los regularizadores [54] son técnicas para evitar que las redes aprendan particularidades o ruido presente en los datos de entrenamiento, es decir, que incurra en sobreajuste. La estrategia de los regularizadores es que exista un intercambio apropiado entre la fiabilidad de los datos de entrenamiento y las bondades

de la robustez de un modelo de red neuronal. La regularización consiste en añadir una penalización en la función de pérdida del modelo, lo que obliga a que los pesos de la red neuronal sean más pequeños. Específicamente, se han utilizado regularizadores *L1* y *L2*. La función de los regularizadores *lasso regularization* o *L1* [55] es la de añadir una penalización proporcional al valor absoluto de los pesos de la red neuronal, de esta manera se provoca que algunos de los pesos tengan un valor pequeño y/o que algunos pesos se reduzcan a cero lo que puede ser útil para reducir la complejidad del modelo y mejorar su capacidad de generalización. El rango del hiperparámetro de penalización *L1* usualmente, oscila entre *0,0001* y *0,01*, valores más grandes pueden conducir a una regularización excesiva, lo que puede conducir a limitar la capacidad del modelo para ajustarse a los datos de entrenamiento. Se aplica mediante la función *regularizers.l1* de la librería *keras* de Python.

Por otra parte, la regularización *L2* [55] (también conocida como *ridge regularization*) añade una penalización proporcional a la suma de los cuadrados de los valores de los pesos de la red neuronal. De esta manera, al minimizar la función de pérdida, los pesos del modelo son penalizados si son demasiado grandes, lo que fomenta a que tengan valores más pequeños. El rango de los hiperparámetros de penalización *L2* comúnmente utilizados varía en el rango de *0,001* a *0,1*. Con la librería *keras* en Python se implementa mediante la función *regularizers.l2*.

5.2.12.2. Explicación de hiperparámetros de la red neuronal

Los hiperparámetros son aquellos parámetros que no son aprendidos en la etapa de entrenamiento del modelo [56], más bien son preestablecidos antes de comenzar el entrenamiento. Los hiperparámetros en función de sus valores, hacen que el modelo de red neuronal funcione mejor, de acuerdo con el objetivo buscado.

Los hiperparámetros que serán explicados son el resultado de una serie de pruebas realizadas, hasta que se encontró la combinación de ellos que devolvía el mejor rendimiento del modelo, en función de la métrica *accuracy* más alta posible, teniendo en cuenta que el procedimiento ha sido un poco iterativo, ósea que se realizaron pruebas con distintas cantidades de capas y tamaños e hiperparámetros diferentes. Es posible que este modelo sea aún mejorable, sin embargo, los resultados, que se verán en el próximo capítulo son bastante prometedores.

A continuación, se explicarán los hiperparámetros utilizados en el modelo.

- Tasa de aprendizaje (*learning rate* = 0,001) : es la tasa con la que se actualizan los pesos (parámetros) del modelo durante el entrenamiento. Se utiliza una función llamada *ReduceOnPlateau* [57] de la librería *keras*, con la que se actualiza la tasa de aprendizaje durante el entrenamiento para mejorar la convergencia, la técnica consiste en que cada vez que el modelo deja de aprender, se ajusta la tasa de aprendizaje.
- Numero de épocas (*epochs* = 2500): es el número de veces que se entrena el modelo, con cada época el modelo se va ajustando o “aprendiendo”, por tanto, mejora sus parámetros.
- Tamaño de lote (*batch size* = 128): es el número de observaciones (entradas) de entrenamiento que se procesan a la vez durante el entrenamiento, esto significa que la red neuronal se entrena por grupos de observaciones del conjunto de datos de entrenamiento, el tamaño del grupo con el que se van haciendo las propagaciones hacia adelante y hacia atrás y ajustando los pesos corresponden al tamaño del lote. Por ejemplo, dado que el tamaño del conjunto de datos de entrenamiento es de 2901 observaciones y el *batch size* = 128, significa, que serán necesarias $2901 / 128 \sim 23$ iteraciones para completar una época.

5.2.12.3. Configuración del proceso de entrenamiento del modelo de red neuronal

Para configurar el proceso de entrenamiento del modelo se utiliza el método *compile*, al que se le deben de especificar los siguientes argumentos:

- Función de pérdida (*loss* = *sparse_categorical_crossentropy*): la función de pérdida es una medida de que tan bien se está ajustando el modelo para ejecutar la tarea dada, en este caso, la clasificación multiclase. El optimizador es el algoritmo que se usa para ajustar los pesos de la red neuronal durante el entrenamiento al minimizar la función de pérdida. Para este modelo se ha utilizado la función de pérdida *sparse_categorical_crossentropy*, cuyo objetivo es el de calcular la pérdida de entropía cruzada categórica dispersa (diferencia entre la distribución de probabilidad predicha y la distribución real de las clases), se suele usar esta función en problemas de clasificación multiclase [58].
- Optimizador (*optimizer* = *Adam*): el optimizador es el algoritmo que se utilizar para ajustar los pesos de las conexiones entre las neuronas para minimizar la función de pérdida, el proceso de ajuste de los pesos se llama optimización. En

este modelo se ha utilizado el optimizador *Adam* [59] es un método de Gradiente Descendente Estocástico (*SGD*) que se basa en la estimación adaptativa de momentos de primer y segundo orden.

- Métrica de evaluación (*metrics = accuracy*): lista de métricas con la que se evalúa el rendimiento durante el entrenamiento y la evaluación. La métrica *accuracy* (exactitud) ver [sección 3.6.4](#), es una medida que se suele utilizar para evaluar el rendimiento de modelos de clasificación.

5.2.12.4. Entrenamiento del modelo de red neuronal

Se procede a hacer el entrenamiento del modelo con los datos de entrenamiento, la estructura (configuración) del modelo, sus hiperparámetros y demás anteriormente explicado se obtiene las siguientes gráficas de error y *accuracy* que puede verse en la ilustración 18:

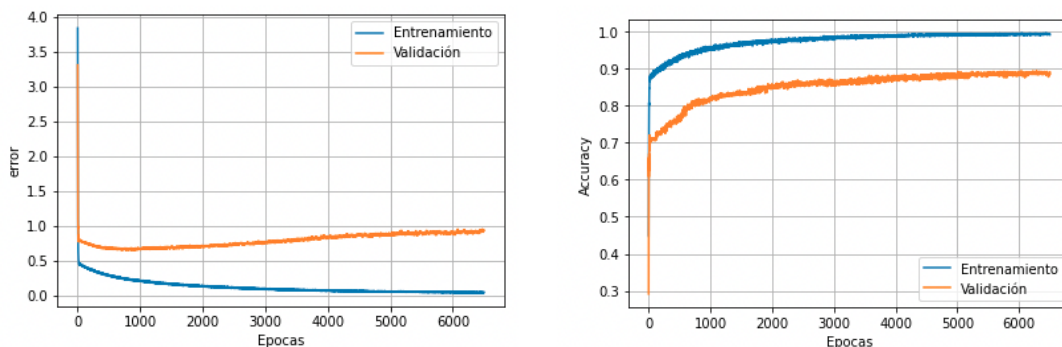


Ilustración 18. Gráficos de error y accuracy de la Red Neuronal

Fuente: elaboración propia

Con cada época se van registrando los rendimientos (errores y *accuracy*) que son graficados. Estas gráficas son muy importantes, pues en ellas se puede observar, primero que todo, si el modelo ha convergido. Para comprobar la convergencia del modelo, la pendiente en la parte final de las gráficas debería ser nula, eso significa que el modelo no aprenderá más, aunque se utilicen más épocas. Adicionalmente, si se observaran cambios bruscos o muchos movimientos erráticos en su comportamiento podría deberse a que tiene una incorrecta tasa de aprendizaje o que los datos tienen valores anómalos, puede corroborarse en los gráficos que no se observa tal cosa. Por último, se espera que la gráfica de validación sea inferior que la de entrenamiento, porque, los datos de validación no son

conocidos por el modelo, y ese será el rendimiento con el que el modelo hará predicciones.

5.2.13. Modelo 7: *Random Forest*

5.2.13.1. Definición de Malla de valores de parámetros

Para ajustar el modelo de *Random Forest* se ha de utilizar una función llamada *ParameterGrid* [60] de la librería *sklearn*, cuya aplicación es crear una “malla con valores de los parámetros discretos” de un modelo, y con ello encontrar la combinación de los valores de estos parámetros que devuelva el mejor ajuste del modelo en función del objetivo inicial planteado. En la tabla 11 se han tabulado los parámetros con los que se ha realizado las iteraciones para encontrar el mejor modelo *Random Forest*.

Tabla 11. *Parameter grid de Random Forest*

Parámetro	Malla de valores
n_estimators	50, 100, 150
max_features	5, 7, 9
max_depth	None, 3, 10, 20
criterion	Gini, entropy

Fuente: elaboración propia

Los hiperparámetros mostrados en la tabla 11 se han extraído de la explicación de la implementación de código en Python para el desarrollo de modelos *Random Forrest* propuesta por Amat, 2020 [61].

5.2.13.2. Ajuste de parámetros de modelo *Random Forest*

La explicación de la configuración de parámetros del clasificador *Random Forest*, mediante el método *RandomforestClassifier* [62] de la librería *sklean* es la que sigue:

- *oob_score* (*oob_score = True*): utiliza muestras “*Out-of-the-bag*” OOB, para estimar la puntuación del ajuste, solo está disponible si el *Bootstrap* está habilitado. Como se explicó en la [sección 4.2.1](#), al construir un árbol de decisión en *Random Forest* los datos se muestrean al azar con reemplazo, lo que significa que algunos datos pueden ser seleccionados varias veces, quedando algunos sin ser seleccionados, estos datos son los llamados OOB. La técnica OOB se utiliza para estimar la precisión del modelo utilizando estos datos para cada árbol individual

en el conjunto de *Random Forest*, para calcular una medida de precisión, sin necesidad de un conjunto de validación adicional.

- `n_jobs` (`n_jobs = -1`): corresponde al número de trabajos que se ejecutan en paralelo, “-1” en este caso significa utilizar la máxima cantidad de recursos de procesamiento disponibles para acelerar la construcción del modelo.
- `random_state` (`random_state=123`): controla la aleatoriedad del bootstrapping de las muestras en la construcción de árboles. Se espera que los resultados sean reproducibles al ejecutarse en distintos ordenadores o tiempos.
- `n_estimators` (`n_estimators = 150`): representa la cantidad de árboles en el bosque. El método tiene predeterminada la cantidad de 100 árboles.
- `max_features` (`max_features=9`): controla la cantidad de características que se consideran al construir cada árbol en el conjunto. Al limitar el número de características que se consideran, se puede reducir la correlación entre los árboles y mejorar la precisión del modelo.
- `max_depth` (`max_depth=None`): determina la profundidad máxima o número de divisiones de cada árbol, si es nulo (*None*), los nodos se expanden hasta que todas las hojas sean puras. Cuando el número es pequeño, los árboles son más limitados lo que puede evitar el sobreajuste, sin embargo, al permitir que el árbol tenga mayor profundidad podrá captar patrones complejos que puedan existir en los datos. Por lo que es requerido encontrar la profundidad óptima.
- `criterion` (`criterion= “entropy”`): esta es la medida con la que se selecciona la mejor división en cada nodo del árbol. La diferencia de ellas radica en la forma en que se mide la pureza de un nodo. La impureza de Gini se define como la probabilidad de que una muestra seleccionada aleatoriamente de un nodo sea clasificada incorrectamente si se clasifica aleatoriamente según la distribución de clases en ese nodo. Es decir, a mayor impureza de Gini en un nodo, más mixtas serán las clases presentes en ese nodo, esta técnica suele funcionar mejor cuando las clases están más diferenciadas en el conjunto de datos. Por otra parte, la entropía mide la cantidad de información necesaria para describir la distribución de clases en un nodo. Cuanto mayor sea la entropía en un nodo, más incertidumbre hay sobre la distribución de clases en ese nodo, esta técnica funciona mejor cuando las clases están mezcladas y la diferenciación de clases es menos clara.

Se dan por concluidas las etapas 1 y 2 introducidas al principio de este capítulo, los resultados pueden observarse en el siguiente capítulo, en el que se explicarán los resultados obtenidos por cada modelo explicados.

6. CAPITULO 6 – EXPOSICIÓN DE RESULTADOS

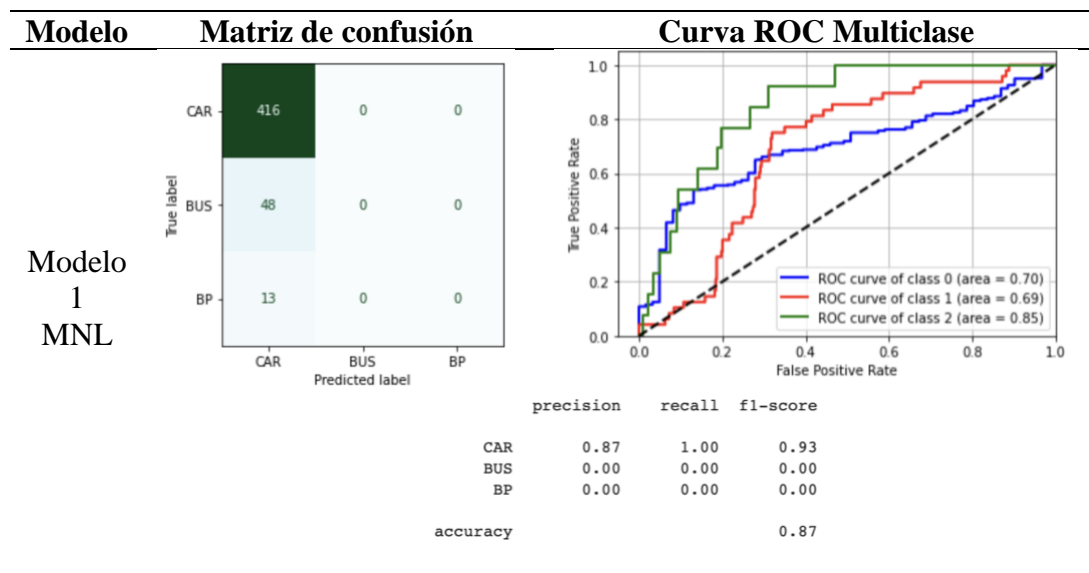
Como se explicó en la [sección 5.2.1](#), los datos se dividieron en dos subconjuntos: uno de entrenamiento (*train*), que fueron los datos utilizados para ajustar o calibrar los modelos explicados anteriormente y los datos de prueba (*test*), que son los que se han utilizado para evaluar la calidad de las predicciones de los modelos.

6.1. Comparación de métricas de evaluación de modelos de clasificación

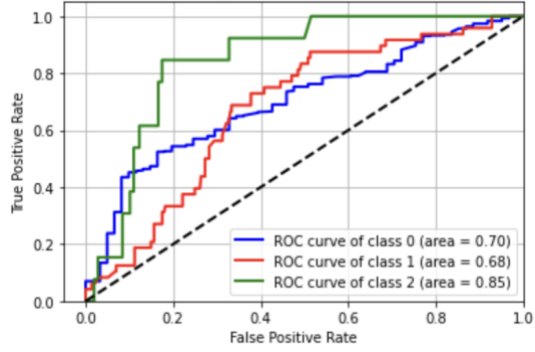
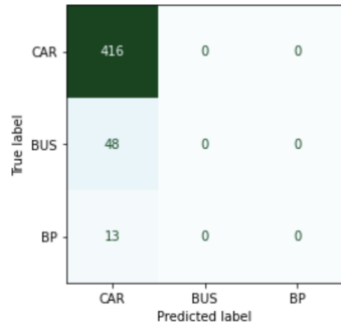
En esta sección se hará una comparación entre los grupos de modelos con similitudes. Se recomienda observar la tabla 10, ya que en ella se explica en que consiste cada uno de los modelos.

A continuación, se muestran las tablas 14 a la 16 en la que se ponen los resultados de las métricas de modelos de clasificación: matriz de confusión, curvas ROC, informe de clasificación (precisión, sensibilidad, F1-score, exactitud) de los modelos evaluados.

Tabla 12. Matriz de confusión y curvas ROC multiclase de modelos 1 y 2



Modelo
2
MNL



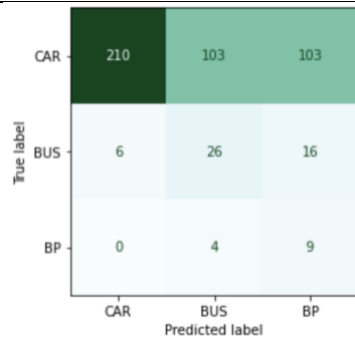
	precision	recall	f1-score
CAR	0.87	1.00	0.93
BUS	0.00	0.00	0.00
BP	0.00	0.00	0.00
accuracy			0.87

Fuente: elaboración propia

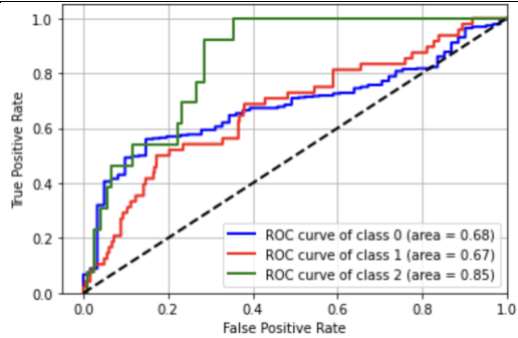
Tabla 13. Matriz de confusión y curvas ROC multiclase de modelos 3 y 4

Modelo
3
MNL

Modelo **Matriz de confusión**

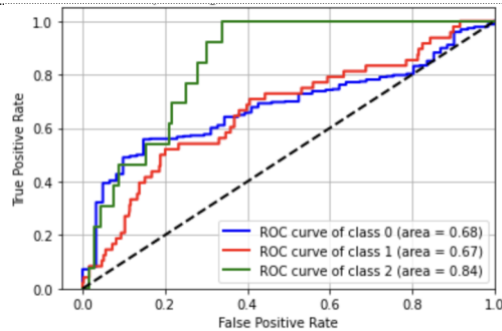
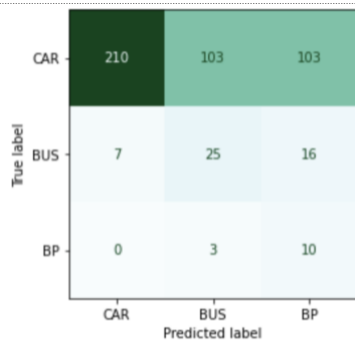


Curva ROC Multiclase



	precision	recall	f1-score
CAR	0.97	0.50	0.66
BUS	0.20	0.54	0.29
BP	0.07	0.69	0.13
accuracy			0.51

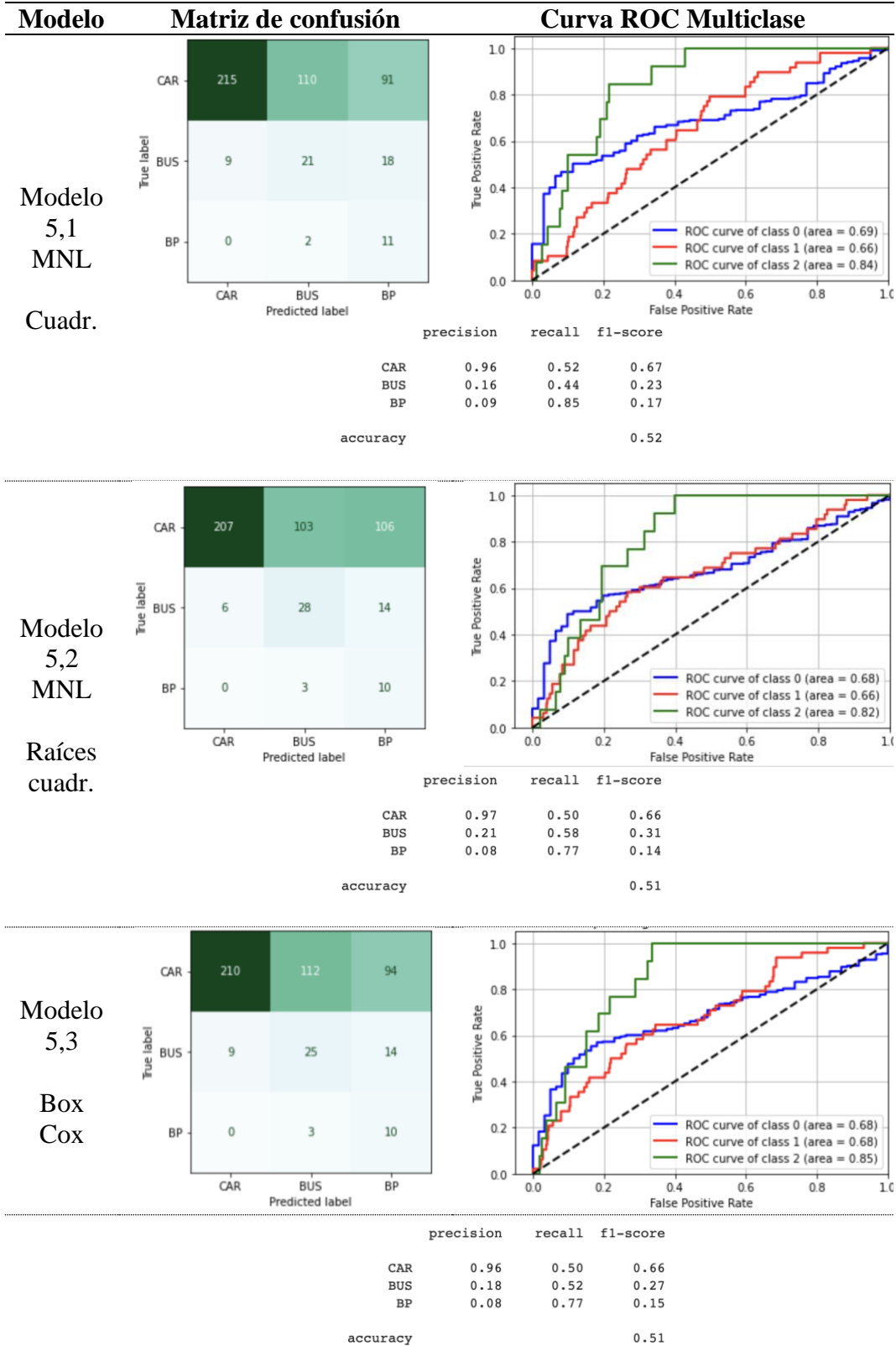
Modelo
4
MNL



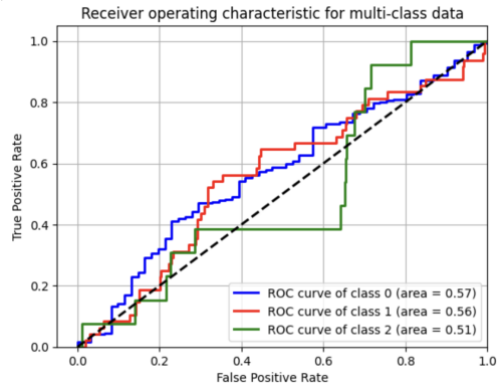
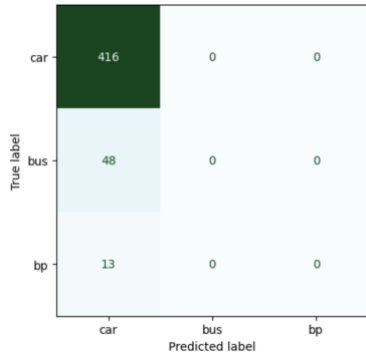
	precision	recall	f1-score
CAR	0.97	0.50	0.66
BUS	0.19	0.52	0.28
BP	0.08	0.77	0.14
accuracy			0.51

Fuente: elaboración propia

Tabla 14. Matriz de confusión y curvas ROC multiclase de modelos 5,1, 5,2 y 5,3



Modelo
5,4
PCA



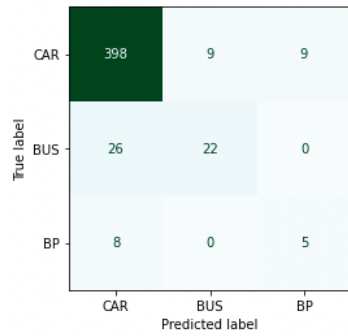
	precision	recall	f1-score	support
car	0.87	1.00	0.93	416
bus	0.00	0.00	0.00	48
bp	0.00	0.00	0.00	13
accuracy			0.87	477

Fuente: elaboración propia

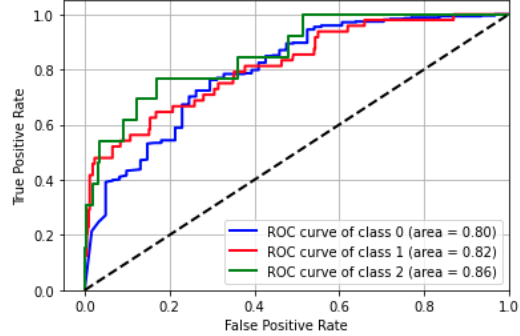
Tabla 15. Matriz de confusión y curvas ROC multiclase de modelos 6 y 7

Modelo
6
NN

Modelo Matriz de confusión



Curva ROC Multiclase

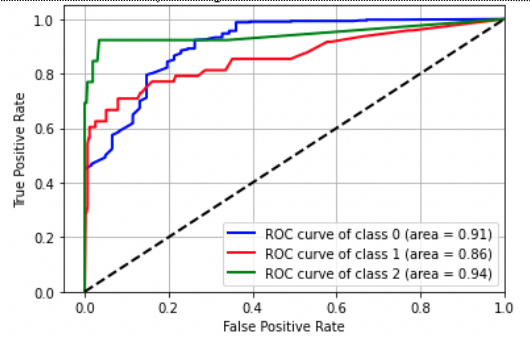
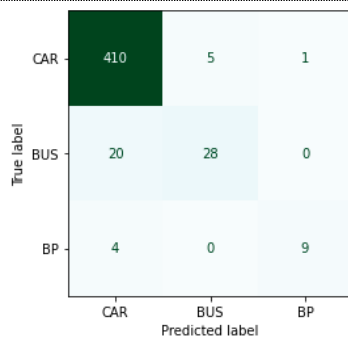


	precision	recall	f1-score
CAR	0.92	0.96	0.94
BUS	0.71	0.46	0.56
BP	0.36	0.38	0.37

accuracy

0.89

Modelo
7
RF



	precision	recall	f1-score
CAR	0.94	0.99	0.96
BUS	0.85	0.58	0.69
BP	0.90	0.69	0.78

accuracy

0.94

Fuente: elaboración propia

6.1.1. Comparación de modelos por matrices de confusión, curva ROC y sensibilidad

Como se explicó en el apartado de tratamiento de datos, la variable dependiente fue sustituida por valores numéricos: 0, 1, 2 para *coche*, *bus* y *bici_patinete* respectivamente. En las ilustraciones de curvas ROC, están puestos las clases por su número, que representan las clases estudiadas.

- *Matrices de confusión, Curvas ROC y sensibilidad de modelos 1 y 2 (MNL) con datos desbalanceados, sin transformaciones*

Para iniciar con el análisis, se comparan los modelos 1 y 2, que como se explicó antes, estos modelos fueron entrenados con los datos de entrenamiento desbalanceados. La diferencia radica en que en el modelo 1 se han incluido todas las variables explicativas y en el modelo 2 han sido eliminado los parámetros no significativos. Los resultados se observan en la tabla 14.

En esta comparación, puede observarse que las matrices de confusión se mantienen iguales, ocurre aproximadamente lo mismo con las curvas ROC, aunque pueden apreciarse algunas leves diferencias en las curvas que no llegan a ser representativas. Las AUC (*Area under the curve*) se mantienen iguales entre sí, obteniendo 0,70, 0,69 y 0,85 para *coche*, *bus* y *bici_patinete* respectivamente. Por lo que específicamente, para esta comparación de modelos, se preferiría usar el modelo 2 por ser más parsimonioso (utiliza menos variables y devuelve los mismos resultados), sin embargo, se deberían usar unos umbrales distintos, porque con los resultados actuales los modelos no llegan a discriminar las clases coches respecto a las demás, dato que se confirma al observar la sensibilidad (*recall*), que la clase *coche* da uno, por otra parte, las clases *bus* y *bici_patinete* son cero.

- *Matrices de confusión, Curvas ROC y sensibilidad de modelos 3 y 4 (MNL) con datos balanceados, sin transformaciones*

Siguiendo con el análisis comparativo, los modelos 3 y 4 se han ajustado con los datos de entrenamiento balanceados, la diferencia entre estos modelos es que el modelo 3 se ha ajustado con todas las variables y en el modelo 4 han sido depuradas sus variables no significativas. En la tabla 15 se puede observar los rendimientos en sus predicciones.

Nuevamente, no se observan diferencias significativas entre un modelo y otro, es decir, sus resultados son relativamente los mismos. Por otra parte, ha sido útil hacer el balanceo de las clases minoritarias, porque puede observarse que se logran identificar correctamente algunos buses y *bici_patinetes*. De acuerdo con la sensibilidad, el modelo 4 está presentando mejores resultados que el modelo 3, sin embargo, no son grandes diferencias, por lo que podría entenderse como equivalente. En conclusión, estos modelos hacen una mejor discriminación de las clases que los modelos 1 y 2.

- *Matrices de confusión, Curvas ROC y sensibilidad de modelos 5.1, 5.2, 5.3 y 5.4 (MNL) datos balanceados, con transformación cuadrática, raíz cuadrada, Box-Cox y PCA respectivamente*

Las relaciones con las variables explicativas no siempre son lineales. Esa es la razón por la que se han introducido transformaciones no lineales en algunas variables explicativas continuas. Se explican los resultados de los modelos 5.1 (cuadrados), 5.2 (raíces cuadradas), 5.3 (Box Cox) y 5.4 (PCA).

Según las matrices de confusión y las curvas ROC de los modelos con transformaciones, en la tabla 16 no tienen una diferencia importante, basándose en la sensibilidad, primeros los tres modelos tienen aproximadamente la misma capacidad para predecir si un usuario usará el coche. Por otra parte, dado que las clases *bus* y *bici_patinete* son clases minoritarias, cada acierto tiene un mayor impacto en su sensibilidad, es por eso por lo que para la clase *bus* el orden de la mejor capacidad de predicción de los modelos sería el 5.3, 5.2 y 5.1, sin embargo, las diferencias siguen siendo poco relevantes y por último para la clase *bici_patinete* el mejor modelo es el 5.1 seguido de los modelos 5.2 y 5.3 con igual sensibilidad para esta clase. Respecto a las AUC de los modelos son relativamente parecidos entre sí, por lo que convendría ajustar los umbrales en función de la clase que se quisiera tener mayor certidumbre de predicción, teniendo en cuenta que esto conllevaría a tener más falsos positivos.

Respecto al modelo 5.4 tiene las mismas matrices de confusión y sensibilidad que los modelos 5.1 y 5.2, porque al igual que esos modelos también está prediciendo todas las observaciones como *coches*, sin embargo, la curva ROC demuestra que este modelo es un poco peor que el modelo original (modelos 1 y 2), porque sus curvas oscilan cerca de la diagonal que representa la aleatoriedad.

- *Matrices de confusión, Curvas ROC y sensibilidad de modelos 7 y 8 (NN y RF), con datos balanceados*

Los modelos 6 y 7 corresponden a modelos de tipo redes neuronales y *Random Forest*, respectivamente. El objetivo de la inclusión de estos modelos es el de comparar los rendimientos y predicciones de los modelos logit contra estos modelos que actualmente, han demostrado tener un comportamiento sobresaliente en el área de clasificación.

Se observa en la tabla 17, que, de acuerdo con las matrices de confusión, por simple inspección parece que los modelos 6 y 7 tienen un comportamiento correcto en la predicción de las diferentes clases. Basándose en la sensibilidad, curvas *ROC*, *AUCs*, el modelo 7, tiene mejores resultados, para cada una de las clases en comparación con el modelo 6.

6.1.2. Comparación de modelos por la métrica exactitud (*accuracy*)

En la [sección 3.4.1](#) se explicó que la tasa de exactitud está dada por la relación entre cantidad de predichos correctamente y la cantidad de observaciones. Esta métrica puede ser una buena medida de referencia para conocer que tan bien están los modelos prediciendo, pero no se llega a saber que clases se están prediciendo correcta o incorrectamente. Y dado que el conjunto de datos de prueba está desbalanceado, la cantidad de predichos correctamente de la clase mayoritaria puede dar como resultado una exactitud, que, al interpretarse, da como resultado una engañosa apariencia de buen modelo.

Es por lo anterior, que solo se mencionarán, sus resultados, pero no se harán conclusiones sobre esta métrica. En la tabla 12 pueden observarse las exactitudes obtenidas por cada modelo.

Tabla 16. Exactitud de modelos

Modelo	Tipo de modelo	Accuracy (exactitud)
1		0,87
2	MNL	0,87
3		0,51
4		0,51
5.1	MNL: cuadrados	0,52
5.2	MNL: raíces cuadradas	0,51
5.3	MNL: Box Cox	0,51
5.4	MNL: PCA	0,87
6	Red neuronal	0,89
7	Random Forest	0,94

Fuente: elaboración propia

- *Exactitud (accuracy) de los modelos logit multinomial: 1, 2 (MNL) y 5.4 (MNL PCA) con datos desbalanceados*

Los modelos 1, 2 y 5.4 tienen una alta exactitud debido a que estos modelos predicen la mayoría de las observaciones como *coche* (la clase mayoritaria). Sin embargo, esta métrica no es fiable.

- *Exactitud (accuracy) de los modelos logit multinomial: 3, 4 (MNL), 5.1 (MNL cuadrático), 5.2 (MNL raíz cuadrada) y 5.3 (MNL Box-Cox) con datos balanceados*

Siguiendo con el análisis, los modelos 3, 4, 5.1, 5.2 y 5.3 tienen una exactitud se aproximadamente el 51%.

- *Exactitud (accuracy) de los modelos 6 (NN) y 7 (RF)*

Por último, los modelos 6 y 7 tienen unas exactitudes altas y además tienen un comportamiento adecuado de las predicciones en todas las clases, por tanto, únicamente, para estos modelos se podría confiar en el resultado de esta métrica.

Debido a la incapacidad de la exactitud para medir el rendimiento de cada clase de manera individual se utilizará la métrica F1-score para evaluar el rendimiento en la predicción de los modelos para cada clase.

6.1.3. Comparación de modelos por la métrica F1-score

Tal como se explicó en la [sección 3.4.1](#), la métrica F1-score es una combinación de dos medidas la precisión (*precision*) y sensibilidad (*recall*). Una alta puntuación del F1-score indica que el modelo es capaz de identificar correctamente una gran proporción de muestras positivas y al mismo tiempo minimizar los falsos positivos (para este efecto, muestras de otras clases que se predicen incorrectamente como la clase evaluada) y falsos negativos (para este efecto, muestras de la clase evaluada predichas incorrectamente como otras clases).

- *F1-score de los modelos 1, 2 (MNL) y 5.4 (MNL PCA) con datos desbalanceados*

Los modelos 1, 2 y 5.3 tienen una valoración del F1-score para la clase *coche*, sin embargo, para el resto de las clases es nula.

- *F1-score de los modelos 3, 4 (MNL), 5.1 (MNL cuadrático), 5.2 (MNL raíz cuadrada) y 5.3 (MNL Box-Cox) con datos balanceados*

Los modelos 3, 4, 5.1, 5.2 y 5.3 tienen una valoración bastante similar entre sí, cambiando por algunas centésimas, siendo en promedio para la clase *coche* 0,66; para la clase *bus*, 0,28; y para la clase *bici_patinete* 0,15.

- *F1-score de los modelos 6 y 7*

Y, por último, el modelo 6 (redes neuronales) devuelve una alta valoración para la clase *coche*, por otra parte, para las clases *bus* y *bici_patinete* devuelve aproximadamente el doble de lo obtenido por los modelos 3, 4, 5.1, 5.2 y 5.3.

Finalmente, con el modelo 7 (*Random Forest*) se obtienen valores altos para cada clase respecto a los modelos anteriores.

6.1.4. Comparación de modelos por comportamiento de probabilidades para cada clase predicha respecto a las observadas

En la [sección 6.1](#) se explicó que se confeccionó una clase de Python para la obtención de las probabilidades, métricas de rendimiento y varias gráficas de los modelos de clasificación. Una de las gráficas que se idearon fue la gráfica de comparación (*Comparative _plot*) que sirve para inspeccionar el rendimiento de las probabilidades por cada clase y su comparación con el resto de las clases.

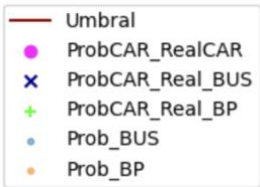

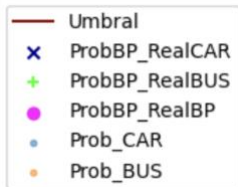
Esta gráfica tiene como objetivo conocer el comportamiento de las probabilidades (predichos) y su capacidad para discriminar entre clases. Además, es posible conocer cambios en los comportamientos de las predicciones producto de la significancia o aporte de ruido de variables, sobre todo las variables dicotómicas. Otra de las ventajas de la gráfica de comparación es que pueden detectarse agrupaciones de clases incorrectamente predichas (falsos positivos o negativos), además es muy útil para definir un umbral de separación óptimo de clases. También, puede descubrirse si los modelos están correctamente planteados y cuando podría descartarse un modelo al notarse que sus resultados tienden a la aleatoriedad en sus predicciones.

El procedimiento para obtener esta gráfica consiste los siguientes pasos.

- Del conjunto de datos de prueba se obtienen las probabilidades para cada modo obtenidas con el modelo evaluado.
- Definir una clase (categoría) a inspeccionar.
- Ordenar de menor a mayor el conjunto de datos de probabilidades por la clase definida.
- Crear un gráfico de dispersión de las probabilidades ordenadas de la clase inspeccionada. Los tipos de puntos se graficarán en función de la clase real, es decir, cada clase tendrá un distinto tipo de punto para identificar si las clases están correctamente discriminadas.
- Agregar al gráfico de dispersión las probabilidades de las demás clases simplemente para observar comportamientos que puedan ser notables.

Para acompañar la explicación anterior se introduce la tabla 13 en la que se presentan las simbologías utilizadas para los gráficos de dispersión cuando se inspecciona cada clase. El umbral en este caso es 0,50, pero no representa nada más que identificar la mitad entre 0 y 1. Para cada observación la clase predicha se escoge como aquella que devuelve la mayor probabilidad, por tanto, en la gráfica será la que tenga la mayor ordenada por cada trio ordenado.

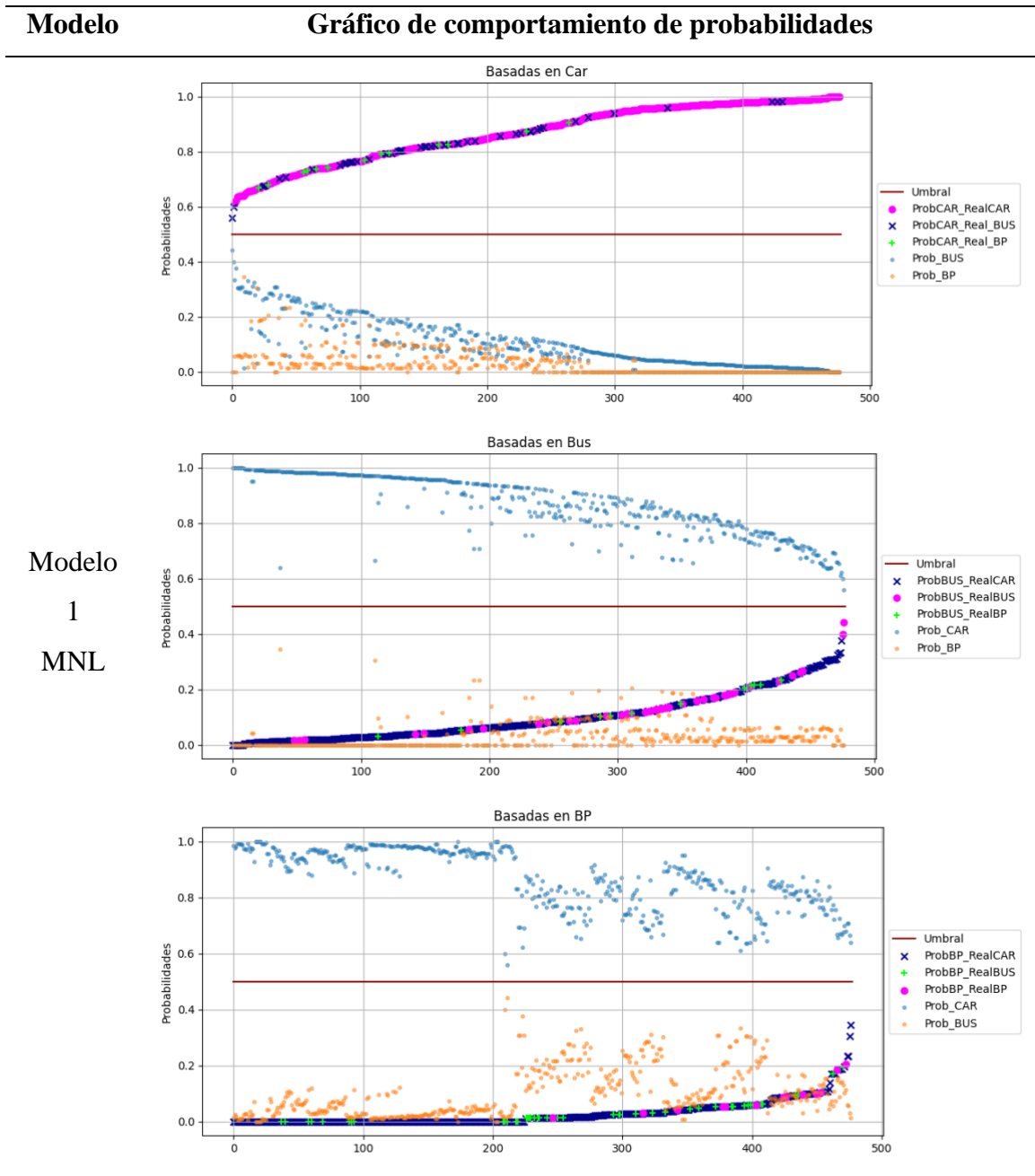
Tabla 17. Simbologías utilizadas cuando se inspecciona cada clase

Clase para inspeccionar		
<i>coche</i>	<i>bus</i>	<i>bici_patinete</i>
		

Fuente: elaboración propia

Una vez explicado cómo funciona este gráfico, en las tablas 18 a la 27 están contenidos los *Comparative_plots* de cada modelo. A continuación, se harán comentarios sobre hallazgos que puedan parecer interesantes de los modelos.

Tabla 18. Comportamiento de las probabilidades respecto a las clases reales obtenidas con el modelo 1



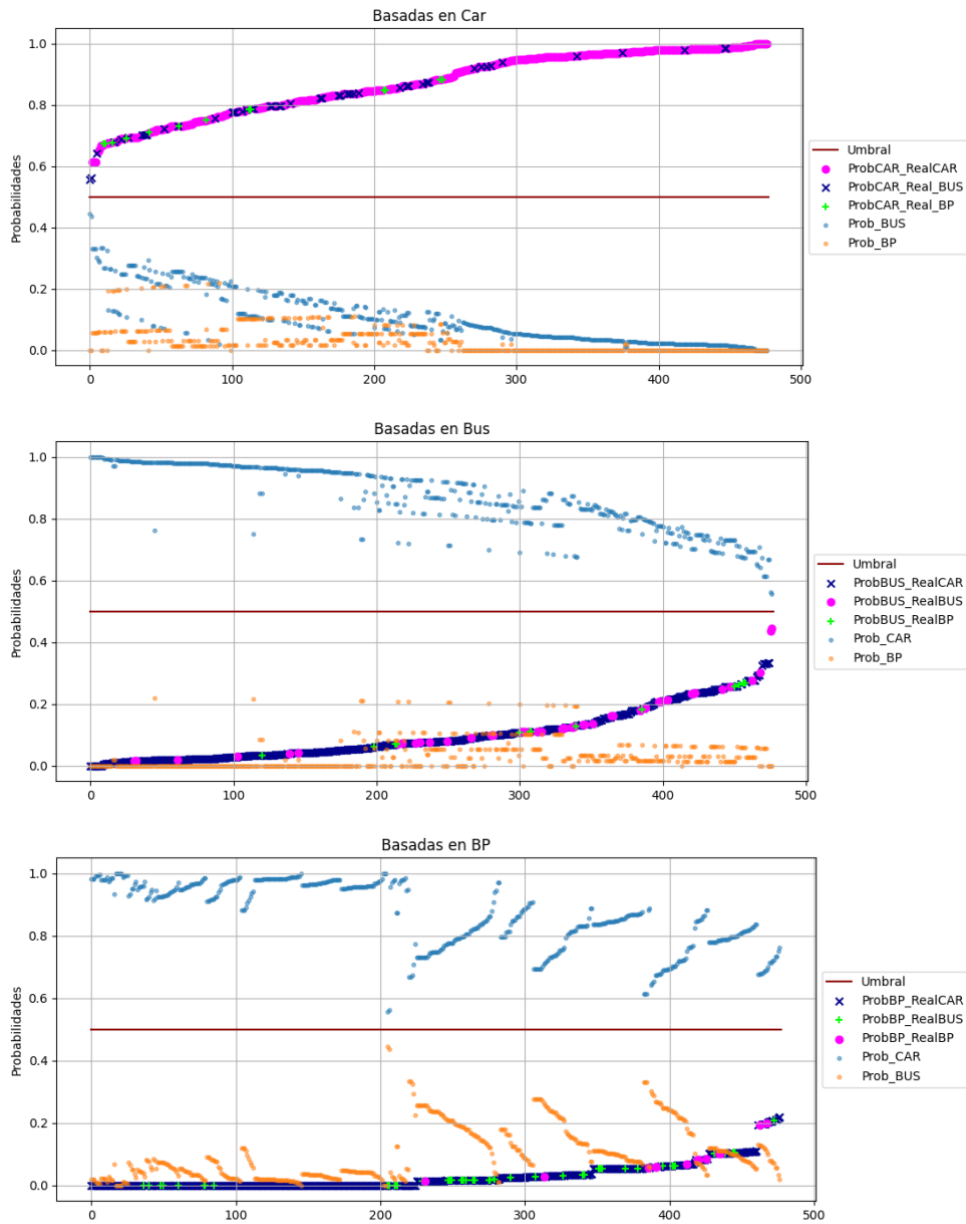
Fuente: elaboración propia

Tabla 19. Comportamiento de las probabilidades respecto a las clases reales obtenidas con el modelo 2

Modelo

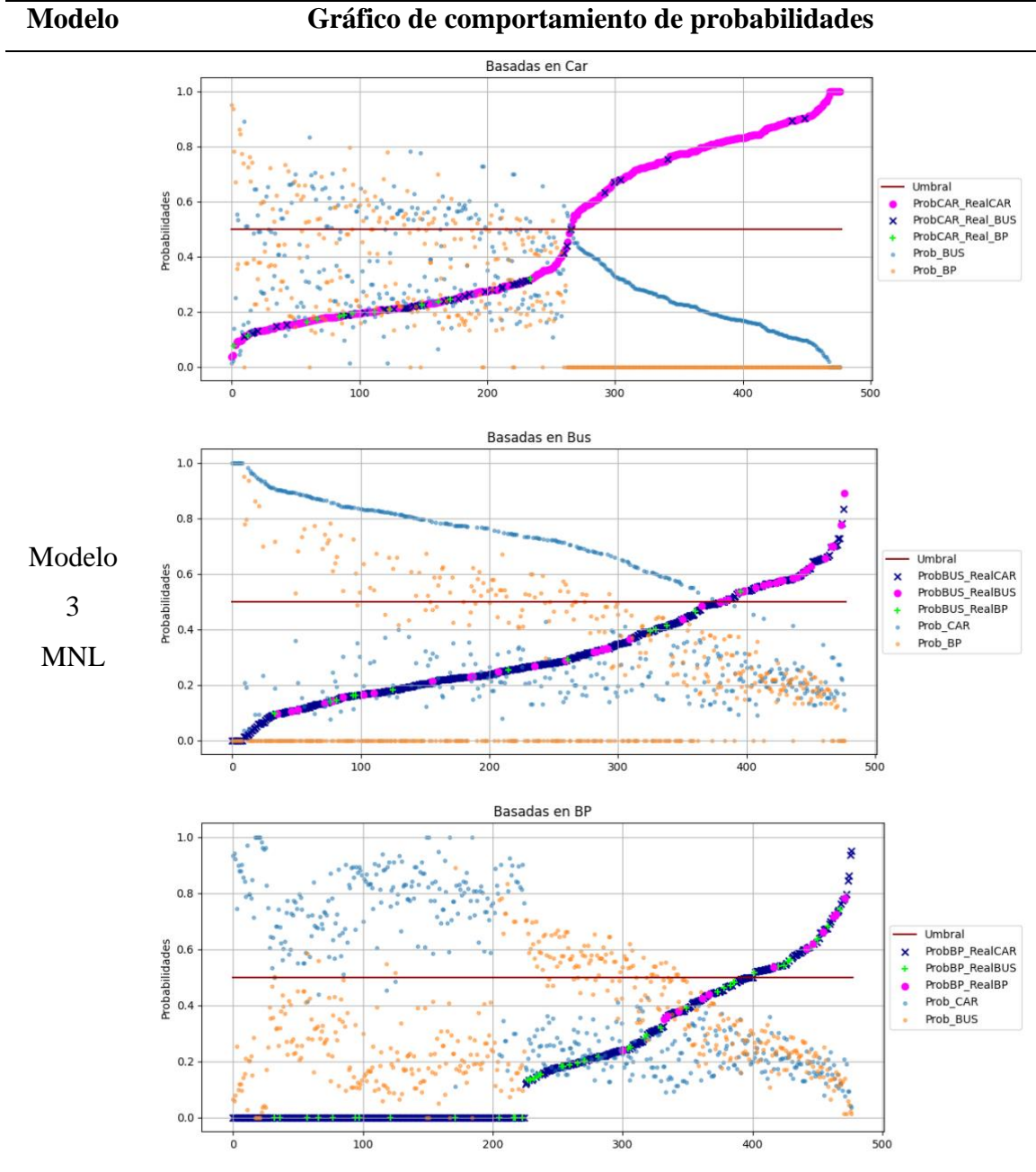
Gráfico de comportamiento de probabilidades

Modelo
2
MNL



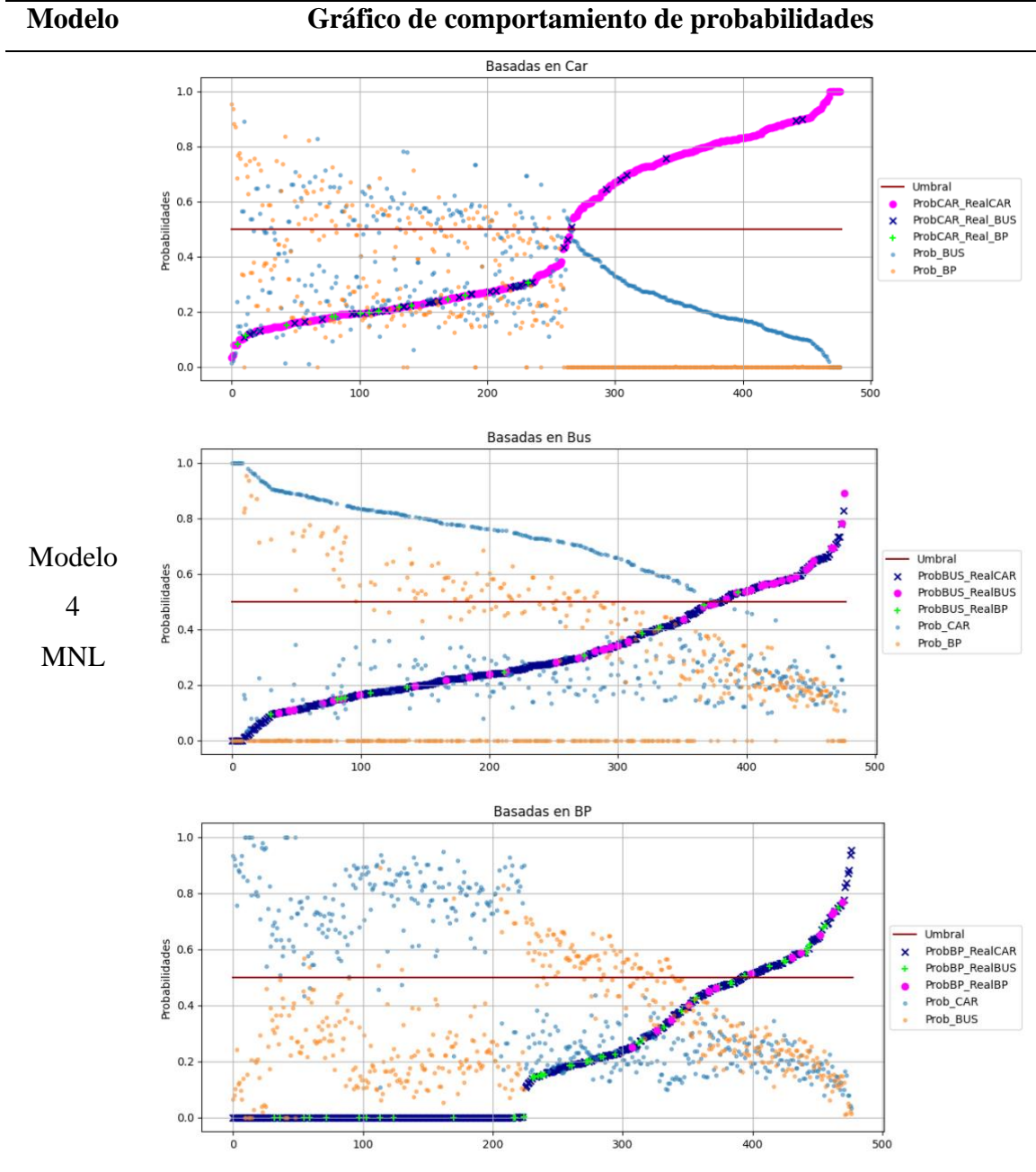
Fuente: elaboración propia

Tabla 20. Comportamiento de las probabilidades respecto a las clases reales obtenidas con el modelo 3



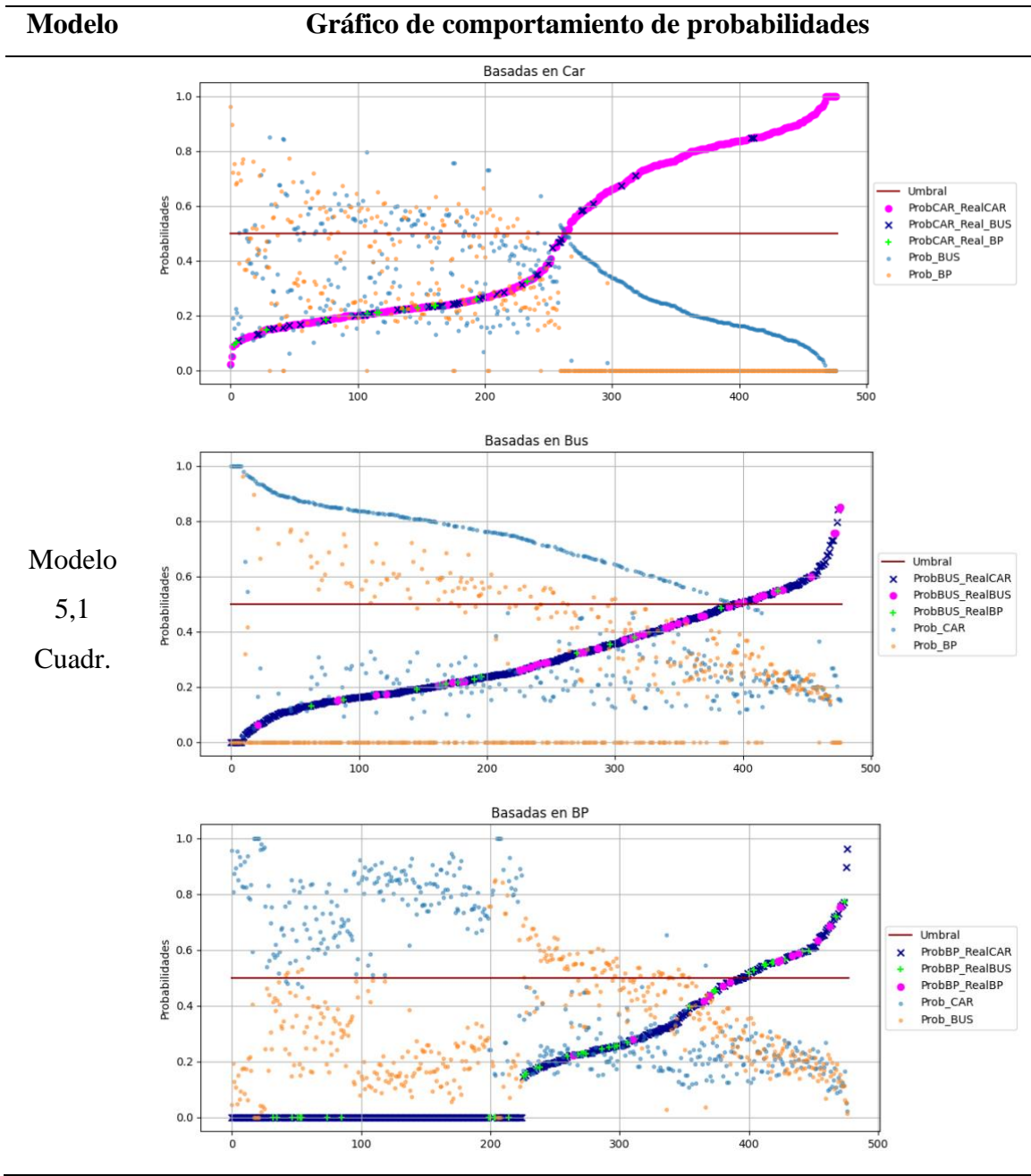
Fuente: elaboración propia

Tabla 21. Comportamiento de las probabilidades respecto a las clases reales obtenidas con el modelo 4



Fuente: elaboración propia

Tabla 22. Comportamiento de las probabilidades respecto a las clases reales obtenidas con el modelo 5,1



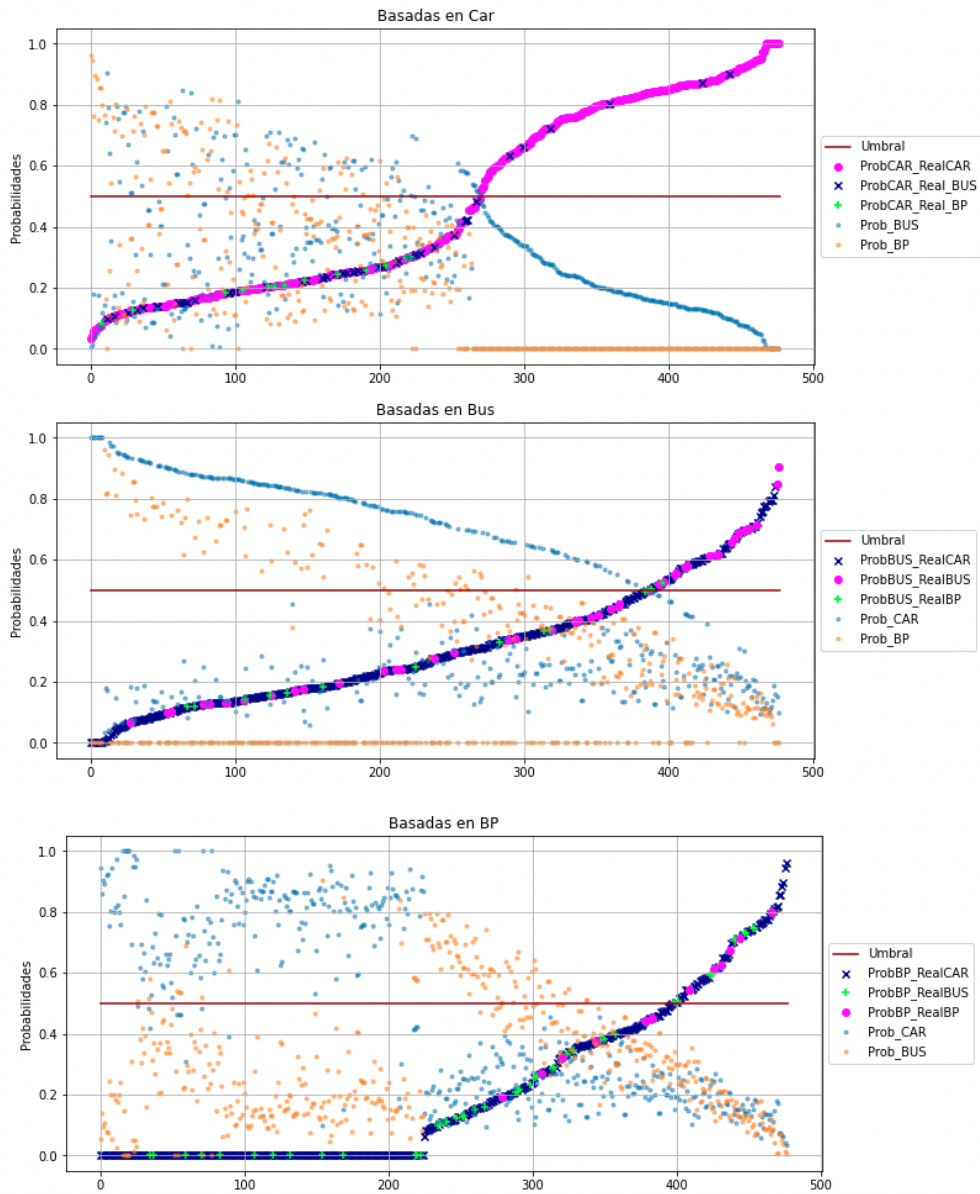
Fuente: elaboración propia

Tabla 23. Comportamiento de las probabilidades respecto a las clases reales obtenidas con el modelo 5,2

Modelo

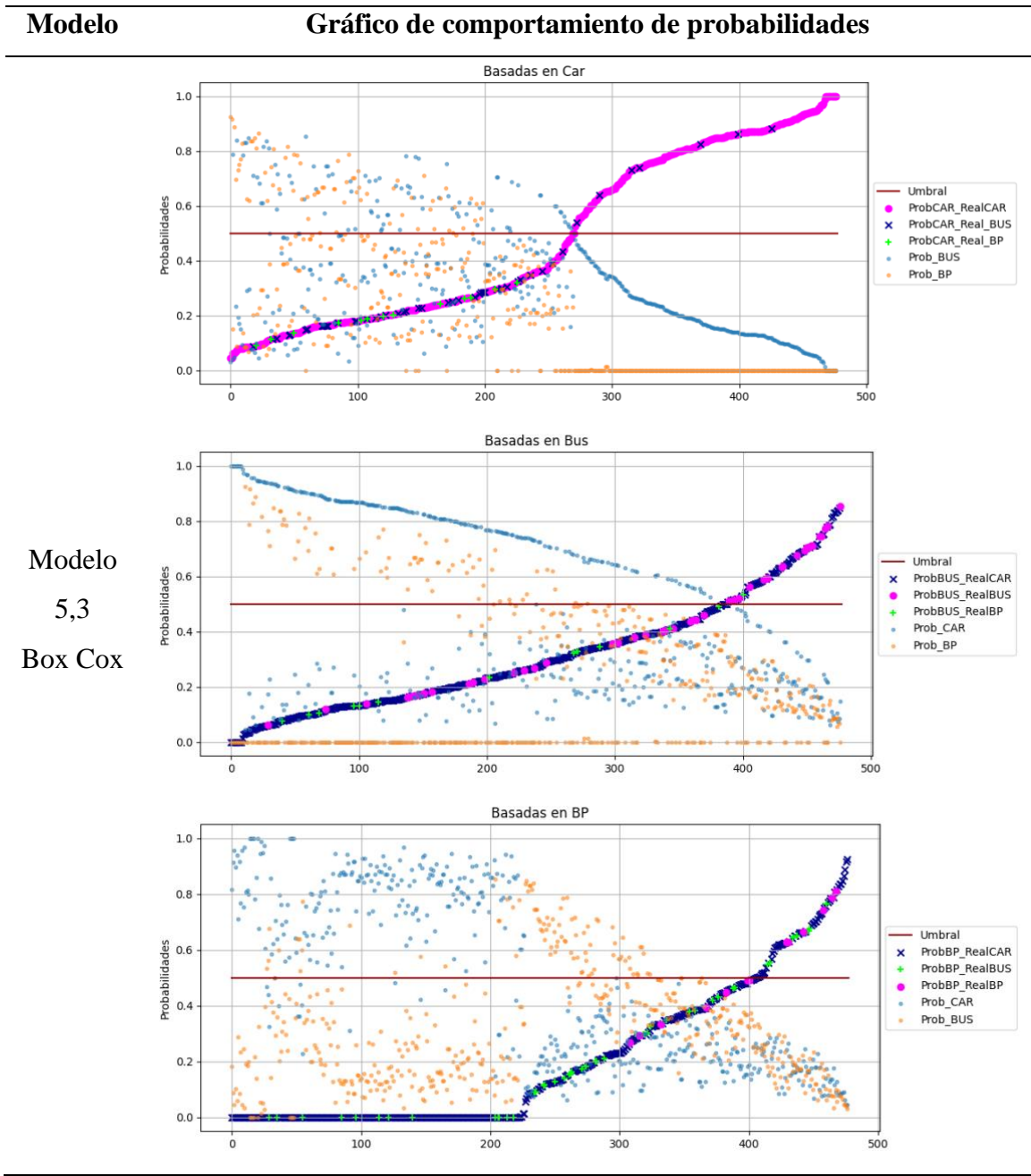
Gráfico de comportamiento de probabilidades

Modelo
5,2
raíces



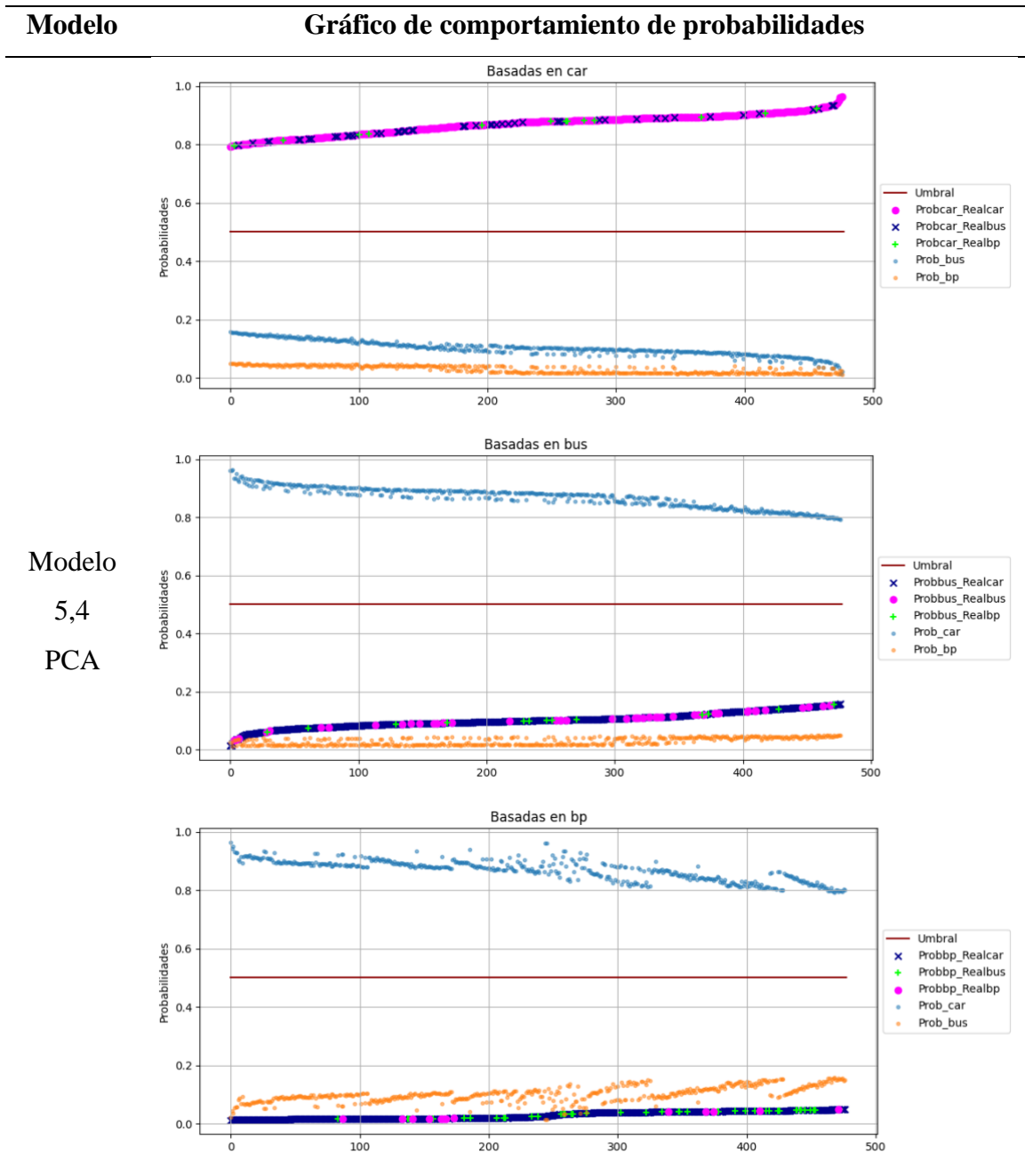
Fuente: elaboración propia

Tabla 24. Comportamiento de las probabilidades respecto a las clases reales obtenidas con el modelo 5,3



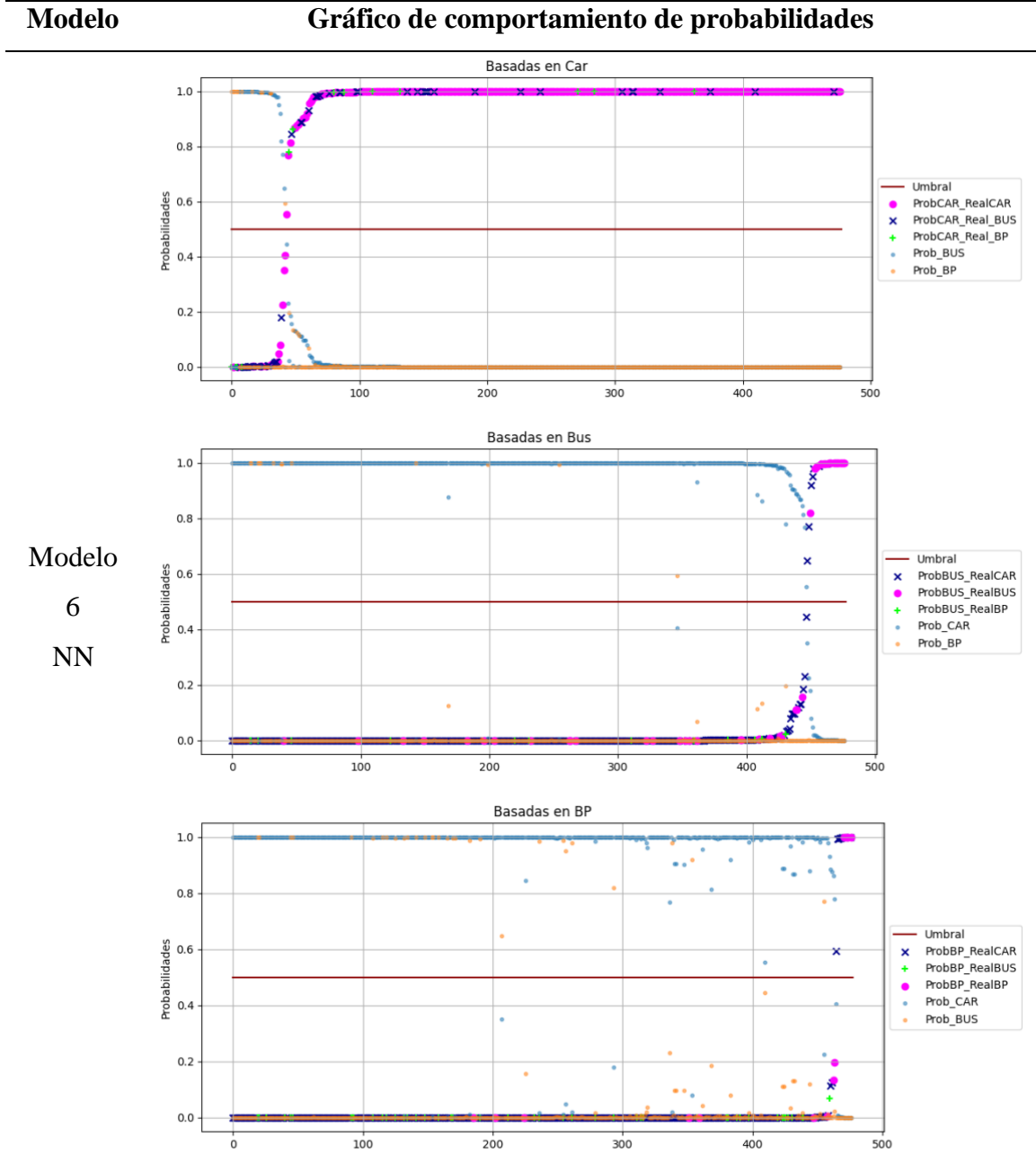
Fuente: elaboración propia

Tabla 25. Comportamiento de las probabilidades respecto a las clases reales obtenidas con el modelo 5,4



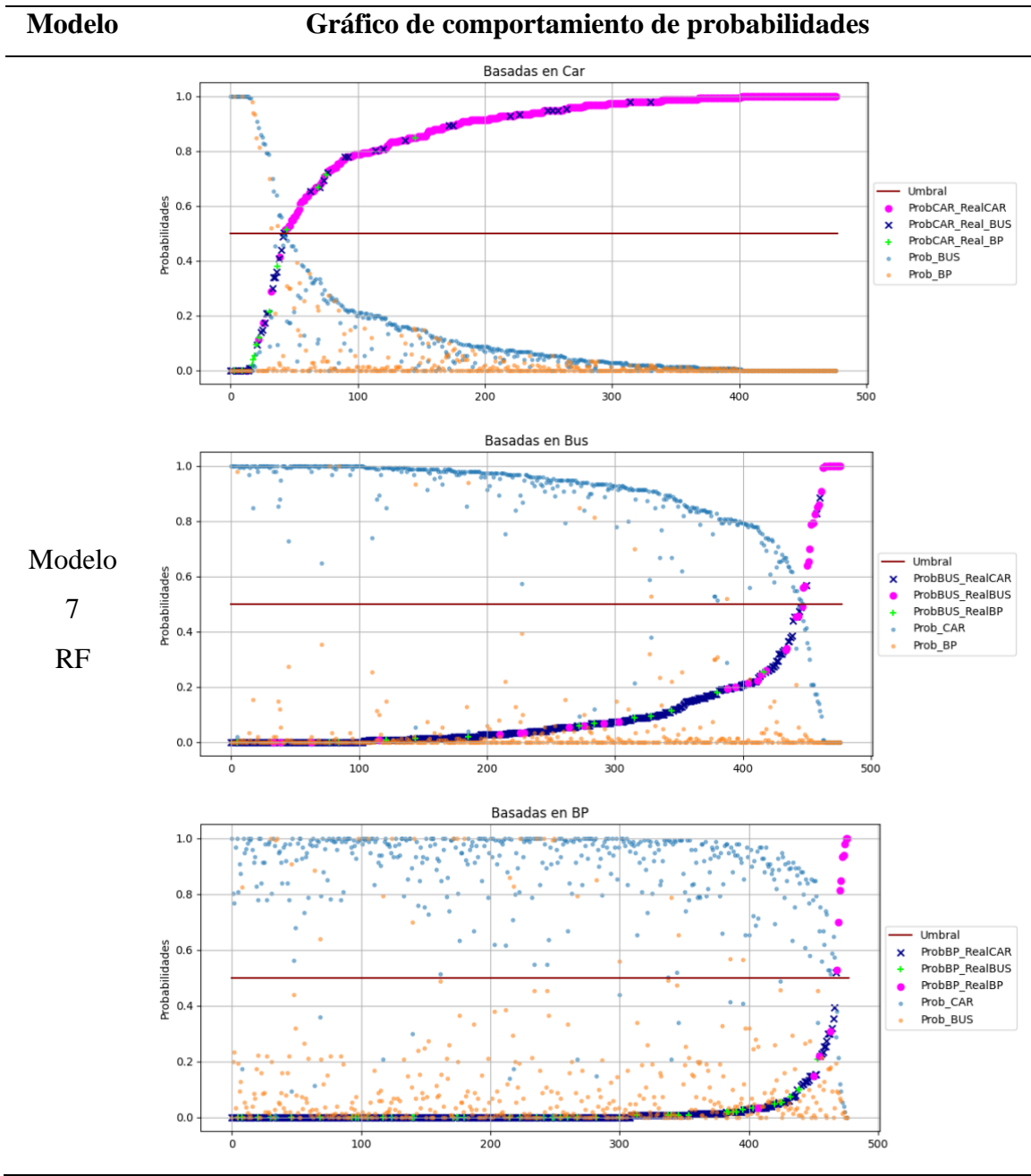
Fuente: elaboración propia

Tabla 26. Comportamiento de las probabilidades respecto a las clases reales obtenidas con el modelo 6



Fuente: elaboración propia

Tabla 27. Comportamiento de las probabilidades respecto a las clases reales obtenidas con el modelo 7



Fuente: elaboración propia

- Revisión de "Comparative_plot" de modelos 1, 2 (MNL) y 5.4 (MNL PCA)

En las tablas 18 y 19 correspondientes a los modelos 1, 2 tienen los mismos comportamientos, para cada clase, salvo en las gráficas de BP, en las que se aprecia que en el modelo 2 las probabilidades de coche y bus (puntos azules y naranjas) están menos dispersos que en el modelo 1. Esto es interesante ya que se interpreta como que el modelo

2 tiene un poco menos de ruido, aunque la dispersión solo se reduce a unas pequeñas áreas de la gráfica, siguen las mismas tendencias.

Al observar la gráfica "*Basadas en Car*" puede notarse que todas las probabilidades oscilan entre aproximadamente 0,55 hasta 1, y están por encima del resto de las probabilidades de las otras clases, es por eso por lo que estos modelos predicen mayoritariamente como coche. Pueden apreciarse algunas agrupaciones de observaciones predichas como coches cuando realmente son buses, pero no pasa lo mismo con las verdaderas clases *bici_patinete*.

Las gráficas de comparación del modelo 5.4 (tabla 25) permiten visualizar que su comportamiento es similar al de los modelos 1 y 2. Sin embargo, hay una diferencia en el rango de probabilidades de las clases, que es más pequeño y homogéneo en el modelo 5.4. Por ejemplo, en los modelos 1 y 2, el rango de probabilidades de las clases de coche y bus es de alrededor del 40%, mientras que en el modelo 5.4 es de alrededor del 20%, manteniendo el mismo comportamiento.

- *Revisión de "Comparative_plot" de modelos 3, 4 (MNL), 5.1 (MNL cuadrático), 5.2 (MNL raíz cuadrada) y 5.3 (MNL Box-Cox) con datos balanceados*

Los modelos 3, 4, 5.1, 5.2 y 5.3 (tablas 20 a 24) tienen tendencias de agrupación bastante similares presentando algunas suavizaciones muy pequeñas en la dispersión de los datos, esto tiene sentido por qué es equivalente con lo obtenido por las métricas F1-score.

Al interiorizar en las gráficas de cada modo se puede notar que para "Basadas en Car" existe un cambio de tendencia a partir de aproximadamente la observación 260 para, luego en la gráfica "Basadas en BP", puede notarse otro cambio de tendencia a partir de la observación 217 (que sería el complemento del tamaño del conjunto de prueba: $477 - 260 = 217$). Se puede interpretar como una fuerte significatividad en alguna de las variables de tipo binarias. Adicionalmente, en la gráfica "Basadas en Car" antes del límite, no está muy clara la distinción de clases, sin embargo, posterior al límite, las clases están ordenadas correctamente, con algunas excepciones. Por otra parte, en la gráfica "Basadas en BP", las probabilidades cambian de predominancia, ósea que antes del límite, al modelo predice la mayoría como coche, después del límite, predice la mayoría como bus. Esta gráfica ha ayudado a interpretar que tan bien se está comportando el modelo en el cálculo de las probabilidades de cada clase.

- *Revisión de “Comparative_plot” de modelos 6 y 7 (NN y RF)*

En las tablas 26 y 27 se puede observar en las gráficas de comparación de probabilidades que el modelo 6 provee una transición más suave de las probabilidades respecto al modelo 7, cuya transición ocurre de manera abrupta y repentinamente. Luego, los dos modelos presentan una apropiada diferenciación entre clases, también pueden observarse más claramente los falsos positivos.

6.2. Discusión de resultados

En esta sección se hará un recuento de los hallazgos más representativos en las comparaciones de resultados, métricas y gráficas de la sección anterior, de cada modelo propuesto.

En primer lugar, ha sido de valioso haber realizado el balanceo de clases, pues incrementa la segregación de clases respecto a los modelos cuyos datos de entrenamiento estaban desbalanceados (modelos 1, 2 y 5.4).

En segundo lugar, no se ha visto una mejora representativa al aplicar las transformaciones a las variables explicativas de tipo continuo en el modelo logit multinomial. Por lo que no se consideran significativas.

En tercero y último lugar, tampoco se observaron grandes diferencias en los modelos cuyas variables no significativas habían sido depurados respecto a los modelos que las conservaban. Por tanto, siempre se recomienda implementar el modelo que sea más parsimonioso, es decir que devuelva los mismos resultados con menos argumentos.

Respecto a los modelos logit multinomial tuvieron en promedio un rendimiento de predicciones menores que la red neuronal y el *Random Forest*, esto es consistente con la conclusión de Ye et al [63], que comparan el modelo logit multinomial con los modelos *Random Forest* y *Support Vector Machine (SVM)* y concluye que los modelos logit multinomial son muy sensibles a la multicolinealidad de las variables explicativas y dado que generalmente, los atributos de los viajeros suelen ser multicolineales, la aplicabilidad de este modelo suele considerarse pobre, comparado con modelos como el *Random Forest* y el *SVM*, cuyas ventajas son muy notorias. Se utilizó la exactitud (*accuracy*) como métrica de comparación entre los modelos.

7. CAPÍTULO 7 – CONCLUSIONES Y RECOMENDACIONES

El presente trabajo se centró en la construcción y comparación de diferentes modelos de aprendizaje automático para predecir el comportamiento de los usuarios al momento de escoger un modo de transporte. Para ello, se aplicaron técnicas de preprocesamiento de datos, selección de variables y evaluación de modelos.

El objetivo de esta investigación fue identificar el/los modelo(s) que ofrezcan el mejor desempeño en la predicción del comportamiento de los usuarios, lo que permitiría mejorar la planificación y gestión del servicio de transporte público. Para lograr este objetivo, se implementaron varios modelos, como el logit multinomial, la red neuronal y el Random Forest, y se compararon a través de diversas métricas y gráficos.

En las secciones anteriores, se presentaron los resultados obtenidos en la evaluación de los diferentes modelos. En esta sección de conclusiones, se sintetizan y analizan estos resultados para identificar las principales fortalezas y debilidades de cada modelo y establecer recomendaciones para la selección del modelo óptimo. Con esta información, se espera contribuir al mejoramiento de la toma de decisiones en el contexto del servicio de transporte público y sentar las bases para futuras investigaciones en este campo.

7.1. Conclusiones

La técnica de balanceo de clases demostró ser una estrategia muy útil y valiosa, ya que permitió mejorar significativamente la predicción de las clases de manera segregada en comparación con los modelos que no fueron balanceados. Este enfoque es particularmente importante en situaciones donde existe una gran desigualdad en la cantidad de datos disponibles para cada clase, lo que puede llevar a una falta de representatividad y a resultados sesgados. Al lograr un equilibrio entre las clases, se pueden evitar estos problemas y garantizar una mayor precisión en las predicciones y decisiones basadas en los resultados del modelo. En resumen, el balanceo de clases es una técnica fundamental para garantizar la calidad y la fiabilidad de los modelos de aprendizaje automático y mejorar su capacidad para manejar conjuntos de datos desafiantes y desequilibrados.

Por otro lado, no se encontró una mejora significativa al aplicar transformaciones a las variables explicativas de tipo continuo en el modelo logit multinomial, por lo que se concluye que estas transformaciones no fueron notables para este conjunto de datos. Además, no se observaron grandes diferencias en los modelos cuyas variables no significativas fueron eliminadas en comparación con los modelos que conservaron las variables no significativas, por lo que se sugiere implementar siempre el modelo más parsimonioso, ya que esto reduce el trabajo de recolección de variables explicativas para hacer predicciones.

En cuanto al rendimiento de los modelos, se encontró que los modelos logit multinomial tuvieron en promedio un rendimiento de predicciones menor que la red neuronal y el Random Forest. Estos resultados son consistentes con la literatura, que sugiere que los modelos logit multinomial son muy sensibles a la multicolinealidad de las variables explicativas, lo que reduce grandemente su aplicabilidad en la predicción de la elección modal de los usuarios frente a otros modelos.

7.2. Recomendaciones

Aunque la metodología se ha comprobado en datos simulados corroborando que se obtienen resultados con coherencia a los resultados obtenidos en este trabajo, y, dado que cada caso y conjunto de datos reales tienen sus particularidades y limitaciones, se recomienda probar y evaluar el rendimiento de los modelos en conjuntos de datos con clases balanceadas. De esta manera, se podrán confirmar las conclusiones logradas.

En este sentido, es recomendable llevar a cabo diferentes pruebas y evaluaciones para encontrar el conjunto de modelos que mejor se adapte a las necesidades específicas de cada caso, ya que no existe una única solución y generalizable que funcione para todos los conjuntos de datos. Además, se sugiere considerar la calidad y cantidad de las variables explicativas utilizadas en cada modelo, ya que esto suele tener un gran impacto en la eficiencia del investigador al momento de recolectar las variables explicativas.

Otro aspecto a tener en cuenta es la necesidad de evaluar y monitorizar continuamente el rendimiento de los modelos en producción a lo largo del tiempo, ya que los datos pueden cambiar y los modelos pueden perder precisión.

Además, se sugiere probar un modelo logit anidado con un conjunto de datos con más clases dependientes en el que se puedan agrupar aquellas clases que tengan correlación

entre sí. Este enfoque de modelo anidado podría ofrecer una mayor versatilidad y utilidad en su aplicación.

Por último, pero no menos importante, a quienes deseen continuar esta línea de investigación se recomienda incluir posibles interacciones entre variables explicativas, ya que al combinarse las variables pueden potenciar su importancia en los modelos.

BIBLIOGRAFIA

- [1] «Observatorio MITMA,» 2023. [En línea]. Available: <https://observatoriotransporte.mitma.es/>.
- [2] D. McFadden, «The Measurement of Urban Travel demand,» *Journal of Public Economics* , vol. 3, pp. 303-328, 1974.
- [3] E. M. Rodríguez, «Logit model como modelo de elección discreta: origen y evolución,» *Anuario Jurídico y Económico Escurialense*, vol. XLI, pp. 469-484, 2008.
- [4] E. Barreno Vereau, «Metodología de modelamiento de un sistema de transporte urbano,» *Red de Revistas Científicas de América Latina, el Caribe, España y Portugal*, n° 26, pp. 11-44, 2008.
- [5] J. d. D. Ortúzar, *Modelos de Transporte*, West Sussex, UK: John Wiley & Sons, Ltd, 2011.
- [6] H. Casey, «Applications to traffic engineering of the law of retail gravitation,» *Traffic Quarterly*, vol. 4, pp. 23-35, 1955.
- [7] L. Guzmán, «Modelo de asignación de transporte para la minimización de costos de cosecha en un trapiche panelero,» *Encuentro Nacional de Investigación Formativa - Memorias*, vol. VI, n° 1, pp. 95-117, 2015.
- [8] K. Montenegro, «Valoración de un asiento de bus usando datos mixtos de preferencias reveladas y declaradas,» Universidad de Chile, Santiago, Chile, 2012.
- [9] R. González et al, *Metodología de Encuestas*, vol. 14, pp. 65-80, 2012.
- [10] P. M. García, «Una aproximación microeconómica a los determinantes de la elección del modo de transporte,» *ScieELO Analytics*, pp. 11-40, 2005.
- [11] A. Orro, «Tesis doctoral: Modelos de elección discreta en transportes con coeficientes aleatorios,» 2005.
- [12] B. A. Garcia, «Modelos de elección binaria,» 2007. [En línea]. Available: chrome-

extension://efaidnbmnnnibpcajpcgglefindmkaj/http://alvarez.webs.uvigo.es/teaching_archivos/ectria2_0708/binary.pdf.

- [13] D. McFadden, «Conditional logit analysis of qualitative choice behavior,» *Frontiers in Econometrics, Academic Press*, pp. 105-142, 1974.
- [14] «The Generalized Nested Logit Model,» *Dept. of Civil Engineering Northwestern University*, pp. 1-28, 2000.
- [15] K. Train, «Discrete Choice Methods with Simulation,» *Cambridge University Press*, 2002.
- [16] J. Cabrera, «Calibración de Modelos Hidrológicos,» Instituto para la Mitigación de los Efectos del Fenómeno El Niño, Universidad Nacional de Ingeniería, Perú, 2012.
- [17] M. Kuhn y K. Johnson, *Applied Predictive Modeling*, New York: Springer, 2013.
- [18] F. Sierra, «La sensibilidad y especificidad: entendiendo su origen y utilidad real,» *Revista Colomb Gastroenterol*, vol. 18, pp. 180-182, 2003.
- [19] T. Fawcett, «An introduction to ROC Analysis,» *Science Direct*, nº 27, pp. 861-874, 2005.
- [20] H. Hasegawa, T. Naito, M. Arimura y T. Tamura, «Hybrid Model of Random Forests and Genetic Algorithms for Commute Mode Choice Analysis,» *In proceedings of the Eastern Asia Society for Transportation Studies*, vol. 9, 2013.
- [21] R. Medina y C. Nique, «Random Forest como extensión de los árboles de clasificación con los programas R y Python,» Lima, Perú, 2017.
- [22] J. Trujillano, A. Sarria, A. Esquerda, M. Badía, M. Palma y J. March, «Approach to the methodology of classification and regression trees,» *Gaceta Sanitaria (SESPAS)*, vol. 22, nº 1, 2008.
- [23] W. Lin, Z. Wu, L. Lin, A. Wen y J. Li, «An Ensemble Random Forest Algorithm for Insurance Big Data Analysis,» *IEEE Access*, vol. 5, pp. 16568-16575, 2017.
- [24] L. Breiman, «Bagging Predictors,» *Machine Learning*, nº 24, pp. 123-140, 1996.
- [25] Y. Mishina y e. al, «Boosted Random Forest,» *IECE TRANS. INF. & SYST.*, vol. 98, nº 9, pp. 1630-1636, 2015.
- [26] E. Serna, *Desarrollo e Innovación en Ingeniería*, Medellín - Antioquia: Editorial Instituto Antioqueño de Investigación, 2017.

- [27] B. Grossfeld, «Zendesk,» 2020. [En línea]. Available: <https://www.zendesk.es/blog/machine-learning-and-deep-learning/>. [Último acceso: 2023].
- [28] IBM, «IBM SPSS Modeler,» 2021. [En línea]. Available: <https://www.ibm.com/docs/es/spss-modeler/saas?topic=networks-neural-model>. [Último acceso: 2023].
- [29] Tibco, «Tibco,» 2023. [En línea]. Available: <https://www.tibco.com/es/reference-center/what-is-a-neural-network>. [Último acceso: 2023].
- [30] L. Hernández, «Modelaje de inundaciones en el departamento de Antioquia a partir de redes neuronales,» *Universidad de los Andes*, 2023.
- [31] J. Alonso, «Tutorial para la estimación de un modelo con variables dummy en EasyReg,» *Apuntes de economía*, n° 14, 2008.
- [32] G. Lema, F. Nogueira y C. Aridas, «Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning,» *Journal of Machine Learning Research*, vol. 18, n° 17, pp. 1-5, 2017.
- [33] T. Brathwaite, «Github,» 2016. [En línea]. Available: <https://github.com/timothyb0912/pylogit#readme>. [Último acceso: 2023].
- [34] M. Bierlaire, «A short introduction to PandasBiogeme,» *Technical report TRANSP-OR*, vol. 05, 2006.
- [35] R. Sakia, «The Box-Cox transformation technique: a review,» *Journal of the Royal Statistical Society: Series D (The Statistician)*, vol. 41, n° 2, pp. 169-178, 1992.
- [36] A. Orro, M. Novales y F. García, «El modelo Logit Mixto Box-Cox para la elección modal en transportes».
- [37] L. Colina y R. López, «El análisis de componentes principales: aplicación al análisis de datos secundarios,» *Papers: revista de sociología*, n° 37, pp. 31-63, 1991.
- [38] M. J. Freire, B. López y P. Carlos, «Employability profiles: The case of a Spanish university,» *Cuadernos de Economía*, n° 43, pp. 341-352, 2020.
- [39] python.org, «Documentos de Python,» [En línea]. Available: <https://docs.python.org/es/3/tutorial/classes.html#:~:text=Las%20clases%20prov>

een%20una%20forma,adjuntos%20para%20mantener%20su%20estado.. [Último acceso: 2023].

- [40] A. M., A. Agarwal, P. Barham y E. Brevdo, «TensorFlow: Large-scale machine learning on heterogeneous systems,» TensorFlow, 2015. [En línea]. Available: www.tensorflow.org. [Último acceso: 2023].
- [41] Keras, «Layer dropout,» [En línea]. Available: https://keras.io/api/layers/regularization_layers/dropout/. [Último acceso: 2023].
- [42] Telefónica, «AI of Things: Telefonica Tech,» [En línea]. Available: <https://aiofthings.telefonicatech.com/recursos/datapedia/funcion-activacion>. [Último acceso: 2023].
- [43] J. Schmidt-Hieber, «Nonparametric regression using deep neural networks with ReLU activation function,» *Annals of Statistics*, pp. 1875-1897, 2020.
- [44] A. Casallas, C. Ferro y N. Mayorga, Llenado de series de datos de 2014 a 2019 de PM2.5 por medio de una red neuronal y una regresión lineal, 2020.
- [45] S. Martinchich y P. Soto, «Clasificación de Bosques por información cartográfica,» *Instituto de Ingenieria Eléctrica*, 2015.
- [46] F. Villa, J. Velásquez y R. Souza, «Una aproximación a la regularización de redes cascada-correlación para la predicción de series de tiempo,» *Investigación operacional*, vol. 28, pp. 151-161, 2008.
- [47] Keras, «Layer weight regularizers,» [En línea]. Available: <https://keras.io/api/layers/regularizers/>. [Último acceso: 2023].
- [48] D. Galvañ, «Comparativa de técnicas para la prevención del sobreajuste en redes neuronales,» 2021.
- [49] keras, «keras,» [En línea]. Available: https://keras.io/api/callbacks/reduce_lr_on_plateau/. [Último acceso: 2023].
- [50] TensorFlow, «Categorical crossentropy,» [En línea]. Available: https://www.tensorflow.org/api_docs/python/tf/keras/metrics/sparse_categorical_crossentropy. [Último acceso: 2023].
- [51] TensorFlow, «Adam optimizers,» [En línea]. Available: https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/Adam. [Último acceso: 2023].

- [52] scikit learn, «Parameter grid model selection,» [En línea]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.ParameterGrid.html. [Último acceso: 2023].
- [53] J. Amat, «Ciencia de Datos,» 2020. [En línea]. Available: https://www.cienciadedatos.net/documentos/py08_random_forest_python.html. [Último acceso: 2023].
- [54] scikit learn, «Random Forest Classifier,» [En línea]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>. [Último acceso: 2023].
- [55] Y. Ye, H. Xiong y Jiayu Tian, «Research on Transportation Mode Selection Based on Machine Learning,» *CICTP*, 2019.
- [56] H. Akaike, «A New Look at the Statistical Model Identification,» *IEEE Transactions on Automatic Control*, vol. 19, n° 6, pp. 716-723, 1974.
- [57] G. Schwarz, «Estimating the Dimension of a Model,» *The Annual of Statistics*, vol. 6, n° 2, pp. 461-464, 1978.
- [58] M. Bierlaire, 2022. [En línea]. Available: <https://groups.google.com/g/biogeme/c/UFhJKYk2N8I>. [Último acceso: 2023].
- [59] scikit learn, «Confusion Matrix Display,» [En línea]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.metrics.ConfusionMatrixDisplay.html>. [Último acceso: 2023].
- [60] scikit learn, «Roc Curve Display,» [En línea]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.metrics.RocCurveDisplay.html>. [Último acceso: 2023].
- [61] scikit learn, «Precision-Recall,» [En línea]. Available: https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html#:~:text=The%20precision%2Drecall%20curve%20shows,a%20low%20false%20negative%20rate.. [Último acceso: 2023].
- [62] C. Cameron y P. Trivedi, *Microeconometrics Methods and Applications*, New York: Cabridge University Press, 2005.

- [63] scikit learn, «Classification metrics,» [En línea]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html. [Último acceso: 2023].

Anexos: Código

El código utilizado para la elaboración de este trabajo en la dirección:

<https://github.com/palmaluisen/discrete-choice-model-predictions.git>