Tasación automática de vehículos

Memoria Proyecto Final de Carrera

Autor: Jorge Pujadas Muñoz
Director: Cèsar Ferri Ramírez
2010

UNIVERSIDAD POLITECNICA DE VALENCIA

etsinf

ÍNDICE

1 - INTRODUCCIÓN

DESCRIPCIÓN PROYECTO FINAL DE CARRERA

El objetivo del proyecto final de carrera consiste en el análisis, desarrollo e implementación de una herramienta que permita la tasación automática de automóviles. Esta herramienta se basará en unos modelos de predicción numérica, construidos a través de técnicas de minería de datos.

La minería de datos (DM, Data Mining) se ha definido como la extracción no trivial de información que reside de manera implícita en los datos.

Un proceso típico de minería de datos consta de los siguientes pasos generales:

1. Selección del conjunto de datos, tanto en lo que se refiere a las variables dependientes, como a las variables objetivo, como posiblemente al muestreo de los registros disponibles.

2. Análisis de las propiedades de los datos, en especial los histogramas, diagramas de dispersión, presencia de valores atípicos y ausencia de datos (valores nulos).

3. Transformación del conjunto de datos de entrada, con el objetivo de prepararlo para aplicar la técnica de minería de datos que mejor se adapte a los datos y al problema.

4. Seleccionar y aplicar la técnica de minería de datos, se construye el modelo predictivo, de clasificación o segmentación.

5. Evaluar los resultados contrastándolos con un conjunto de datos previamente reservado para validar la generalidad del modelo.

Para iniciar el proceso de minería de datos, se recopilarán datos de páginas web que contienen automóviles en venta. Estas páginas, además del precio de venta del automóvil, recogen características tales como: modelo, potencia, kilometraje, año de fabricación, etc.

Para ello se construirá un programa que permitirá volcar de manera automática en la base de datos el conjunto de datos disponible en las páginas web de compra venta de automóviles. Este programa además del volcado de la información sobre los vehículos, filtrará los registros con información errónea o poco relevante, transformando los atributos seleccionados de la manera adecuada. Es decir, el programa llevará a cabo las tareas 1, 2 y 3 del ciclo de vida de un proyecto de minería de datos. Como resultado de este proceso se obtendrá una tabla denominada vista minable y que sirve como punto de partida de la tarea 4 (aplicación de la técnica de minería de datos).

Para la fase 4, el aprendizaje y evaluación de modelos, se utilizarán herramientas informáticas de minería de datos de libre distribución, como por ejemplo Weka. Estas herramientas contienen un gran número de técnicas de aprendizaje por lo que se podrían comparar los resultados de diferentes métodos, y de esta manera seleccionar aquella técnica que obtenga el modelo con el menor error. Para evaluar los modelos, dado que en este caso se trata de un problema de regresión, se utilizarán medidas como: error cuadrático medio, error ponderado, etc.

Finalmente, tras la selección del mejor modelo de predicción de acuerdo con el error estimado, se implementará este modelo y se integrará en una aplicación web de manera que un usuario podrá conocer el precio de venta del vehículo introduciendo las características del vehículo en la aplicación. Se contemplarán también otras funcionalidades en la aplicación web como por ejemplo, búsqueda de ofertas preferentes, es decir aquellas ofertas de vehículos cuyo precio ofertado esté muy por debajo del precio estimado por el modelo de tasación.

2 - MINERÍA DE DATOS

2.1 – DEFINICIÓN

A continuación se pasa a explicar en qué consiste la minería de datos, esta es la parte principal en la cual se basa el PFC. Es necesario comprender en qué consiste para la realización del proyecto y el aprovechamiento de este.

Lo primero es una definición formal, la siguiente esta extraída del libro "Introducción a la Minería de Datos".
Se define la minería de datos como el proceso de extraer conocimiento útil y comprensible, previamente desconocido, desde grandes cantidades de datos almacenados en distintos formatos, con lo cual encontrar modelos inteligibles a partir de los datos y el uso de patrones descubiertos debería ayudar a tomar decisiones más seguras que reporten algún beneficio a la organización.

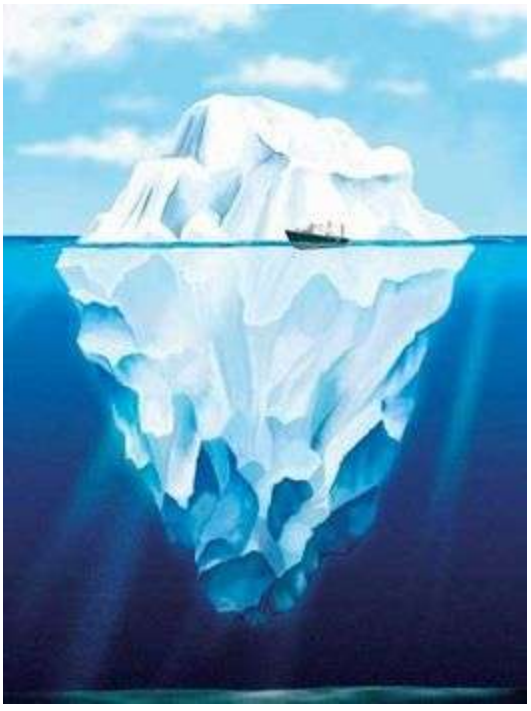

Figura 1: Metáfora de los datos y el conocimiento.

Ejemplo visual de la minería de datos (Figura 1), el hielo del iceberg son los datos almacenados, de los cuales solo se conocen una parte, la parte al descubierto. Debajo del mar existe un gran conocimiento que aparentemente está oculto pero con la minería de datos se puede descubrir y explotar.

La minería de datos no es un campo de los llamados “tradicionales”, se ha ido definiendo en los últimos años, no solo por la investigación y el normal desarrollo de las técnicas, sino también por la necesidad que se ha creado en los últimos años a causa del aumento de la información disponible y los medios técnicos que se tienen. En las grandes organizaciones ya no se borra nada, todo queda registrado para posteriores análisis y seguimientos, pero. ¿Cómo utilizar adecuadamente toda esta información que sobrepasa la capacidad humana de comprensión y análisis? De esta necesidad nació la minería de datos principalmente.

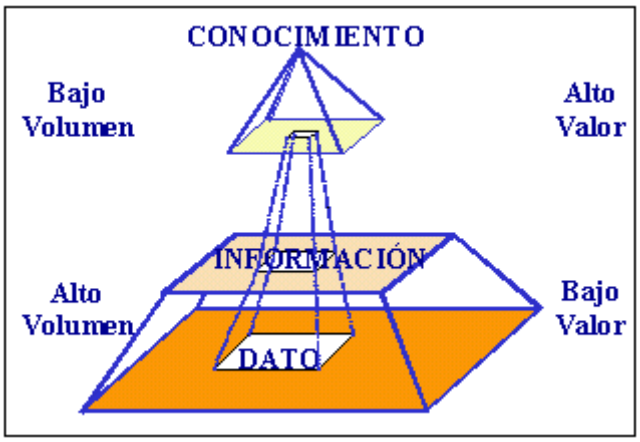

Figura 2: Relación entre conocimiento, información y datos.

La minería de datos no es una disciplina en sí, se basa en otras disciplinas para conseguir sus objetivos tales como: extraer patrones, describir tendencias y regularidades, predecir comportamientos. Todo esto permite comprender el contexto en el que se debe actuar y tomar decisiones más acertadas.

Como hemos comentado antes la minería de datos no forma una disciplina o un campo por sí misma es solo una etapa de un proceso más grande, al cual se le llama extracción de conocimientos a partir de datos, del cual otras técnicas importantes son: los campos del aprendizaje automático, la estadística, las bases de datos, los sistemas de toma de decisión, la inteligencia artificial y otras áreas de la informática y de la gestión de información.

Uno de los cambios que ha traído la minería de datos a los sistemas de información es que los datos ya no son almacenados y utilizados tal cual se registraron, estos datos pasan a ser la materia prima de la cual se puede obtener el conocimiento necesario, el cual es muchísimo más valioso que los datos sin procesar. Para la realización de las tareas necesarias han surgido una nueva generación de herramientas y técnicas para soportar la extracción de conocimientos útil desde la información disponible.

El resultado de la minería de datos son conjuntos de reglas, ecuaciones, arboles de decisión, redes neuronales, grafos probabilísticos…, los cuales pueden usarse para, por ejemplo, responder a cuestiones como ¿existen un grupo de clientes que se comporta de manera diferenciada?

2.2 - RELACIÓN CON OTRAS DISCIPLINAS

Como se ha comentado ya, la minería de datos se nutre de otras disciplinas, por ello la investigación y los nuevos avances de estas también mejoran la minería de datos en sí. Podemos destacar como disciplinas más influyentes las siguientes:

- **Bases de datos:** conceptos como los almacenes de datos y el procesamiento analítico en línea (OLAP) tienen una gran relación con la minería de datos, en las que se basan para extraer conocimiento novedoso y comprensible. Las técnicas de indización y de acceso eficiente a los datos son muy relevantes para el diseño de algoritmos eficientes de minería de datos.
- **Recuperación de información:** consiste en obtener información desde datos textuales, tanto en librerías como en internet. La utilización de búsquedas utilizando palabras clave puede verse como un proceso de clasificación.
- **Estadística:** Esta disciplina ha proporcionado muchos de los conceptos, algoritmos y técnicas que se utilizan en minería de datos. De hecho, algunos paquetes de análisis estadístico se comercializan como herramientas de minería de datos.
- **Aprendizaje automático:** Esta es el área de la inteligencia artificial que se ocupa de desarrollar algoritmos y programas capaces de aprender, y constituye, junto a la estadística, el corazón del análisis inteligente de los datos.
- **Sistemas para la toma de decisión:** Son herramientas y sistemas informatizados que asisten a los directivos en la resolución de problemas y en la toma de decisiones. El objetivo es proporcionar la información necesaria para realizar decisiones efectivas en el ámbito empresarial o en tareas de diagnostico.
- **Visualización de datos:** El uso de técnicas de visualización permite al usuario descubrir, intuir o entender patrones que serian más difíciles de "ver" a partir de descripciones matemáticas o textuales de los resultados.
- **Computación paralela y distribuida:** Actualmente, muchos sistemas de bases de datos comerciales incluyen tecnologías de procesamientos paralelo, distribuido o de computación en grid. En estos sistemas el coste computacional de las tareas más complejas de minería de datos se reparte entre diferentes procesadores o computadores.
- **Otras disciplinas:** Dependiendo del tipo de datos a ser minados o del tipo de aplicación , la minería de datos usa también técnicas de otras disciplinas como el lenguaje natural, el análisis de imágenes, el procesamiento de señales, los graficos por computadora, etc.

Los pasos a seguir en un proceso de minería de datos estándar son iterativos ya que el resultado de alguna fase puede hacer volver a una fase anterior para mejorar el proceso. A menudo son necesarias varias iteraciones para extraer conocimiento de alta calidad.
El proceso se organiza en torno a 5 fases:

- **Fase de integración y recopilación de datos:** Se buscan los recursos de los cuales se va a extraer la información y se transforman a un formato común. En esta fase se recogen datos de diferentes fuentes y se adecuan para su utilización.
- **Fase de selección, limpieza y transformación:** Al recoger los datos de diversas fuentes y estas no estar debidamente estructuradas se pueden encontrar datos incorrectos o incompletos. Por eso es necesaria una limpieza y/o transformación de los datos sin la cual el proceso entero se vería afectado.
- **Fase de minería de datos:** Es la más característica del proceso. Mediante una técnica de minería de datos, se obtiene un modelo de conocimiento, que representa patrones de comportamiento observados en los valores de las variables. El modelo es una descripción de los patrones y relación entre los datos que pueden usarse para hacer predicciones.
- **Fase de evaluación e interpretación:** Con los resultados obtenidos se determina si estos son satisfactorios, de no ser así se vuelve a las fases anteriores y se realiza una nueva iteración. Tres características que deben tener los patrones son que sean precisos, comprensibles e interesantes.
- **Fase de difusión:** Se dan a conocer los resultados y las conclusiones a las que se llega a los interesados. Estos pueden ser un analista para realizar la toma de decisiones, o bien para aplicar el modelo a diferentes conjuntos de datos.

Figura 3: Fases en la minería de datos.

Dentro de la minería de datos hemos de distinguir tipos de tareas, cada una de las cuales puede considerarse como un tipo de problema a ser resuelto por un algoritmo de minería de datos. Las distintas tareas pueden ser predictivas o descriptivas. Entre las tareas predictivas encontramos la clasificación y la regresión, mientras que el agrupamiento (clustering), las reglas de asociación, las reglas de asociación secuenciales y las correlaciones son tareas descriptivas.

Clasificación

La clasificación es una de las tareas de minería de datos más populares. Problemas de negocios, como análisis de clientes, gestión del riesgo y la orientación de anuncios publicitarios por lo general son solucionados con este método.

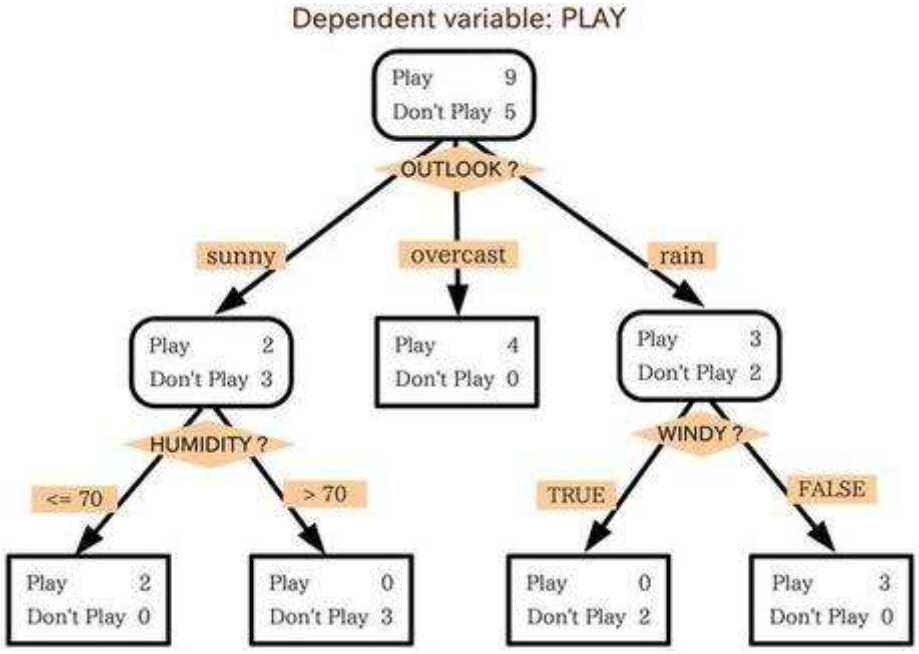

Figura 4: Ejemplo de clasificación.

La clasificación consiste en la asignación de los casos en categorías basadas en un atributo previsible. Cada caso contiene un conjunto de atributos, uno de los cuales es el atributo clase (Atributo predecible). La tarea requiere encontrar un modelo que describa el atributo clase en función de atributos de entrada. Para entrenar un modelo de clasificación, lo que necesita saber es el valor de la clase de los casos de entrada en el conjunto de datos de formación, que suelen ser los datos históricos.

Los algoritmos de clasificación típicos incluyen árboles de decisión, redes neuronales, y Naïve Bayes.

Agrupación (Clustering)

La agrupación es también llamada segmentación. Se utiliza para identificar agrupaciones naturales de casos basados en un conjunto de atributos. Los casos de un mismo grupo tienen más o menos valores de atributos similares.

La mayoría de algoritmos de clustering construyen el modelo a través de un número de iteraciones y se detienen cuando el modelo converge, es decir, cuando los límites de estos segmentos se estabilizan.

Asociación

La asociación es otra tarea popular de minería de datos. La asociación también se denomina análisis de la cesta. Un problema típico de asociación en negocios es analizar una tabla de transacciones de ventas y determinar los productos que a menudo se venden en la misma cesta de la compra. El uso común de las tareas de asociación es identificar grupos comunes de elementos (conjuntos de elementos frecuentes) y las reglas con el fin de conseguir las denominadas ventas cruzadas.

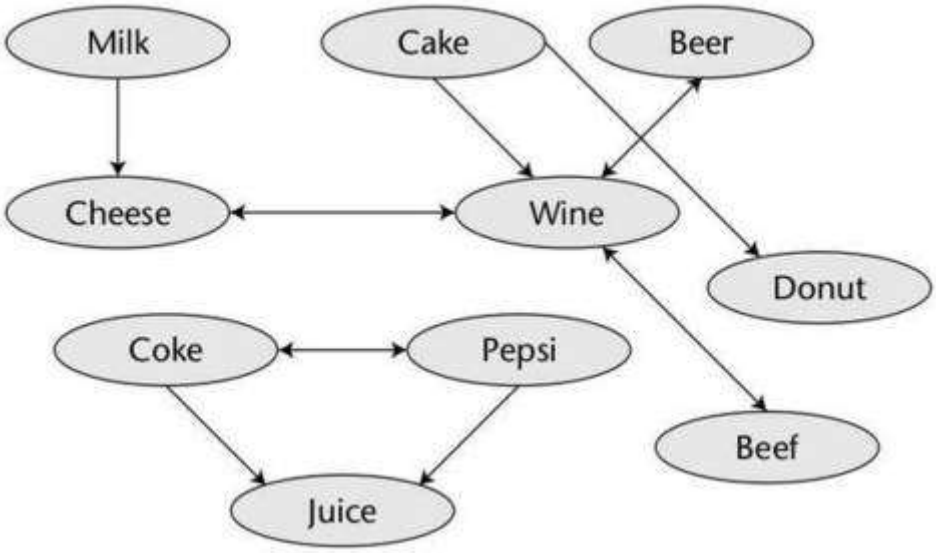

Figura 5: Ejemplo de asociación.

En la asociación, cada producto, o más generalmente, cada par atributo/valor se considera un elemento. Las tareas de asociación tienen dos objetivos: encontrar conjuntos de elementos frecuentes y encontrar reglas de asociación.

Regresión

La tarea de regresión es similar a la de clasificación. La principal diferencia es que el atributo predecible es un número continuo. Las técnicas de regresión han sido ampliamente estudiadas desde hace siglos en el campo de la estadística. La regresión lineal y la regresión logística son los métodos más populares de regresión. Otras técnicas de regresión incluyen árboles de regresión y redes neuronales.

Las tareas de regresión pueden resolver muchos problemas de negocios. Por ejemplo, puede utilizarse para predecir la velocidad del viento basándose en la temperatura, presión atmosférica y humedad.

2.5 - EJEMPLOS EN LA VIDA REAL

Para una mejor compresión del uso de la minería de datos en la vida real se van a exponer una serie de ejemplos prácticos.

Hábitos de compra en supermercados
El ejemplo clásico de aplicación de la minería de datos tiene que ver con la detección de hábitos de compra en supermercados. Un estudio muy citado detectó que los viernes había una cantidad inusualmente elevada de clientes que adquirían a la vez pañales y cerveza. Se detectó que se debía a que dicho día solían acudir al supermercado padres jóvenes cuya perspectiva para el fin de semana consistía en quedarse en casa cuidando de su hijo y viendo la televisión con una cerveza en la mano. El supermercado pudo incrementar sus ventas de cerveza colocándolas próximas a los pañales para fomentar las ventas compulsivas.

Fraudes
Un ejemplo más habitual es el de la detección de transacciones de blanqueo de dinero o de fraude en el uso de tarjetas de crédito o de servicios de telefonía móvil e, incluso, en la relación de los contribuyentes con el fisco. Generalmente, estas operaciones fraudulentas o ilegales suelen seguir patrones característicos que permiten, con cierto grado de probabilidad, distinguirlas de las legítimas y desarrollar así mecanismos para tomar medidas rápidas frente a ellas.

Genética
En el estudio de la genética humana, el objetivo principal es entender la relación cartográfica entre las partes y la variación individual en las secuencias del ADN humano y la variabilidad en la susceptibilidad a las enfermedades. En términos más llanos, se trata de saber cómo los cambios en la secuencia de ADN de un individuo afectan al riesgo de desarrollar enfermedades comunes (como por ejemplo el cáncer). Esto es muy importante para ayudar a mejorar el diagnóstico, prevención y tratamiento de las enfermedades. La técnica de minería de datos que se utiliza para realizar esta tarea se conoce como “reducción de dimensionalidad multifactorial”.

También se utiliza la minería de datos en otros muchos campos como en el análisis de gases, la ingeniería eléctrica, la ciencia, los juegos, la lucha contra el terrorismo, los comportamientos de los usuarios en internet, los recursos humanos y hasta en los patrones de fuga de clientes.

Toda esta explicación sobre la minería de datos se quedaría solo en algo meramente teórico si no fuera aplicado en un ejemplo real y practico. Y para ello necesitamos una herramienta la cual nos permita realizar esta tarea. La herramienta elegida para tal función es **Weka**.

Figura 6: Logo de Weka.

Logotipo de Weka. El animal mostrado en el logo es el pájaro weka, un ave característica de Nueva Zelanda.

Figura 7: Pantalla principal.

Weka es una colección de algoritmos de Máquinas de conocimiento desarrollados por la universidad de Waikato (Nueva Zelanda) implementados en Java, útiles para ser aplicados sobre datos mediante los interfaces que ofrece o para embeberlos dentro de cualquier aplicación. Además Weka contiene las herramientas necesarias para realizar transformaciones sobre los datos, tareas de clasificación, regresión, clustering, asociación y visualización. Weka está diseñado como una herramienta orientada a la extensibilidad por lo que añadir nuevas funcionalidades es una tarea sencilla.

Sin embargo, y pese a todas las cualidades que Weka posee, tiene un gran defecto y éste es la escasa documentación orientada al usuario que tiene junto a una usabilidad bastante pobre, lo que la hace una herramienta difícil de comprender y manejar sin información adicional.

La licencia de Weka es GPL*, lo que significa que este programa es de libre distribución y difusión. Además, ya que Weka está programado en Java, es independiente de la arquitectura, ya que funciona en cualquier plataforma sobre la que haya una máquina virtual Java disponible.

De entre las características más destacables están la gran cantidad de filtros para el preprocesado de datos que corresponden a las funciones de selección de atributos, discretización, tratamiento de valores desconocidos y transformación de atributos numéricos.

También existe una amplia gama de modelos de aprendizaje tales como arboles de decisión, tablas de decisión, vecinos más próximos, maquinas de vectores soporte, reglas de asociación, métodos de agrupamiento y modelos combinados.

En esta página http://www.cs.waikato.ac.nz/ml/weka/ se encuentra la documentación completa sobre Weka, para más información del funcionamiento del programa o las distintas funciones.

En la sección 4 de este documento se explicara con más detalle el funcionamiento de algunas de ellas las cuales se utilizan para el proyecto.

3 - RECOPILACIÓN Y PREPARACIÓN DE DATOS

3.1 – EXTRACCIÓN DE DATOS

Los datos con los cuales se van a trabajar serán extraídos de los anuncios de vehículos de ocasión de la página www.autoscout24.com. Si fuera el caso de ver los datos relativos a un único modelo para una consulta normal como usuario la historia no iría mas allá, pero en el caso de este proyecto se quiere recabar toda la información posible del máximo de vehículos que en la web están anunciados. Para ello la recolección de datos de forma manual sería tan costosa en tiempo y esfuerzo que es descartada solo al plantearse. El ir apuntando todos los datos de cada vehículo en una base de datos de forma manual no es una cosa factible. Asique para este problema la solución es la creación de un wrapper, un programa el cual automáticamente recopile la información correspondiente y la almacene correctamente.
Con esto la recolección de datos es una tarea fácil y automática, quitando la creación del wrapper claro está.

Para la **creación del wrapper** se ha implementado un programa en C#, el programa utiliza la librería System.Net y las funciones WebClient, Steam y SteamReader para conectar con una página web dada y descargar el código fuente de esta. La url de un anuncio responde a la estructura de "www.autoscout24.es/Details.aspx?id=183311157", con lo que el id marca el identificador del anuncio y es único para cada uno, con lo cual para acceder a una serie de anuncios basta con cambiar el numero de id. Una vez ya se tiene el código fuente almacenado en el programa se pasa a analizar este. Observando los patrones de la web nos damos cuenta que para cada campo del anuncio se utiliza una etiqueta de con lo cual se pueden tomar como referencia para localizar cada uno de los campos. Mediantes funciones de procesado de cadenas de texto como IndexOf, LastIndexOf, Split SubString, se logra aislar los caracteres requeridos que corresponden a los datos del vehículo (marca, modelo, potencia, etc.). A continuación mediante los controladores de Access que incorpora C# se inserta el registro en la BD. Después de esto ya se tiene el vehículo con su información y se pasa a un nuevo vehículo lo cual se logra implementando un bucle que recorra diferentes "urls" pero incrementando en cada ciclo el numero de id. Así hasta que el usuario detenga el programa.

Figura 8: Ejemplo de anuncio de vehículo.

En este ejemplo se puede observar los diferentes atributos del vehículo que nos interesan, desde los principales como el precio, fecha matriculación y kilometraje, hasta el equipamiento que lleva el vehículo, pasando por los aspectos referidos al motor y al consumo.

3.2 - WRAPPER

Una posible **definición de wrapper** podría ser esta:

Un wrapper (también llamado como Web Wrapper) es un software de extracción de información. Su cometido es transformar ítems de datos, extraído desde un documento semi-estructurado (por ejemplo, un documento HTML), en una representación auto-descrita (por ejemplo un documento XML) el cual pueda ser utilizado por una variedad de aplicaciones de bases de datos.

El funcionamiento del wrapper consiste en acceder a una página web con información de un vehículo de ocasión, reconocer las características del coche que serán utilizadas como atributos, guardarlas en una base de datos, previamente creada con la estructura deseada, y pasar a otra página web de un nuevo vehículo. Todo esto sin necesidad de la intervención del usuario.

A continuación se muestran unas capturas del programa en ejecución y la explicación de su funcionamiento.

Figura 9: Imagen del recolector.

En esta primera pantalla se muestra el inicio de la aplicación, en el existen 3 botones y 2 cuadros de texto. El primer botón es el de Cargar Base de datos, con el se muestra un cuadro de exploración para seleccionar la base de datos en la cual guardar los nuevos datos. Una vez cargada la base de datos se activa el 2º botón llamado Comenzar exploración, pulsando en el se da inicio a la recolección de datos buscando por los correspondientes anuncios en la web.

En el cuadro de texto Id de inicio se muestra el numero del último vehículo guardado al pulsar el botón de Cargar Base de datos, este número se puede cambiar por otro por el cual se quiera empezar la recolección. A continuación se muestra otro campo de texto donde se debe indicar el número de anuncios a explorar.

Y finalmente un botón de Salir para cerrar la aplicación.

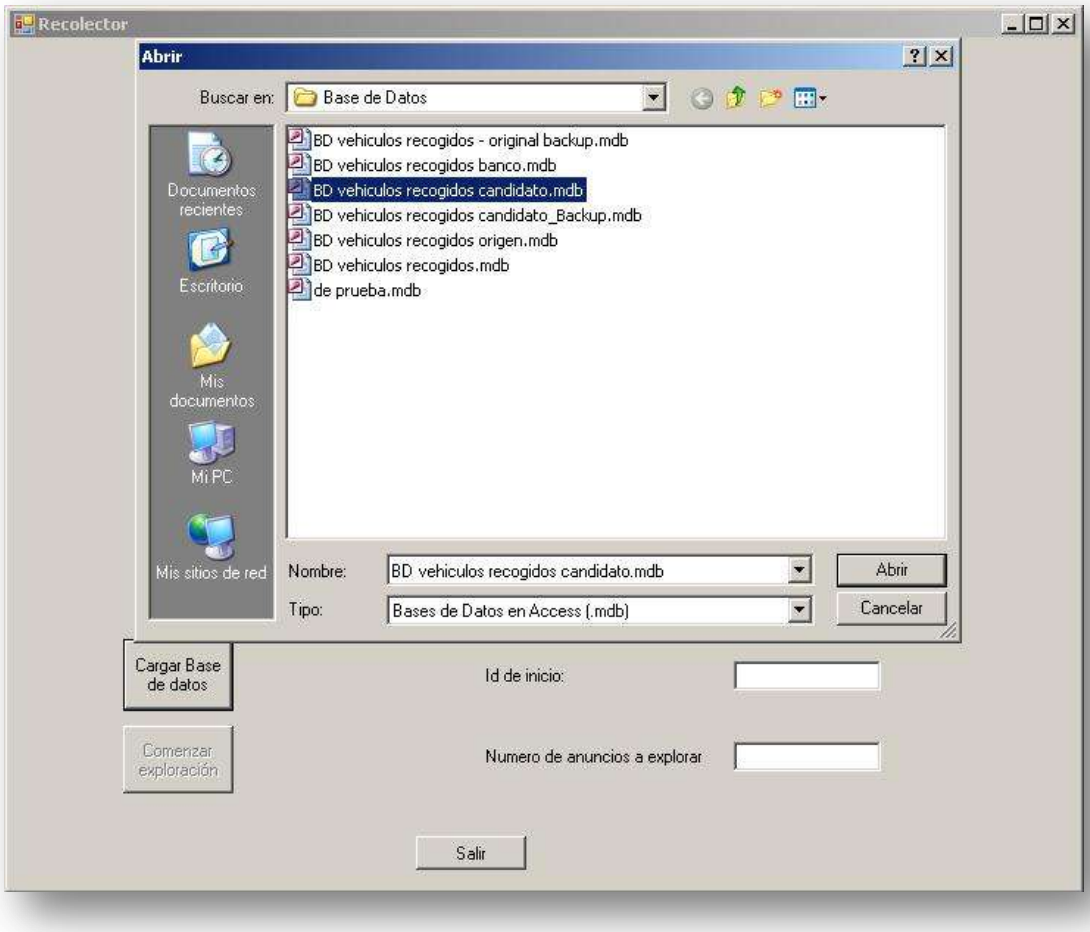

Figura 10: Cuadro de dialogo de selección de base de datos.

ACLARACIÓN DEL PROGRAMA WRAPPER

Uno de los **problemas de los wrappers** como el desarrollado en esta aplicación, está en que se basan en el código fuente de una página web, mientras la estructura de esta página web no cambie el funcionamiento del programa será correcto. Este wrapper se basa en la lectura del código fuente HTML de la página fuente, de él se buscan una serie de etiquetas de estilo para situar los atributos que se precisan. El problema viene cuando la estructura y la forma de marcar los estilos de la pagina web cambia, en ese caso el wrapper ya no funciona correctamente y es incapaz de recolectar nuevos anuncios y guardarlos si no se modifica el código fuente y se adapta a los nuevos cambios.

En este caso se ha tenido la mala suerte que desde el proceso de recolección de los datos de los anuncios a la escritura de esta memoria en la página www.autoscout24.com se ha cambiado el modo de etiquetar los campos de estilo por lo que actualmente el programa recolector no es operativo y para su nueva utilización necesitaría una actualización.

3.3 – BASE DE DATOS

Para almacenar los datos se creara una base de datos, en este caso se ha creado en **MS-Access**, se podría haber realizado en un SGBD más potente pero para el caso no era necesario y además así facilitaba la integración con la aplicación en C# hecha desde Visual Studio.

Esta es la **secuencia SQL** para la creación de la tabla en la cual se guardaran los datos correspondientes a los vehículos que se vayan capturando desde el wrapper.

```
CREATE TABLE vehiculos_recogidos
(Id Número,
Marca Texto (50),
Modelo Texto (50),
Precio Número,
Carrocería Texto (25),
Potencia Número,
Fecha_matriculacion Fecha,
Cambio Texto (25),
Kilómetros Número,
Combustible Texto (25),
Puertas Número,
Consumo Numero Doble,
4WD Sí/No,
Airbag Sí/No,
Airbag_acompañante Sí/No,
Airbag_lateral Sí/No,
Aire_A Sí/No,
Cierre_cent Sí/No,
Climatizador Sí/No,
Direccion_asistida Sí/No,
Elevalunas Sí/No,
Xenon Sí/No,
Llantas Sí/No,
Aparcar_asis Sí/No,
Navegador Sí/No,
Asientos_cuero Sí/No,
Techo_solar Sí/No,
ABS Sí/No,
Adap_disca Sí/No,
Alarma Sí/No,
Asientos_calef Sí/No,
Asientos_electricos Sí/No,
Baca Sí/No,
Bizona Sí/No,
Bola_remolque Sí/No,
Control_traccion Sí/No,
ESP Sí/No,
Anti_niebla Sí/No,
Inmobilizador Sí/No,
Ordenador_bordo Sí/No,
Radio Sí/No,
```

```
Radio_cd Sí/No,
Tunning Sí/No,
PRIMARY KEY (Id)
);
```

3.4 - PREPARACIÓN DE DATOS

A continuación se explican los pasos para la conversión de los datos guardados en la BD con el objetivo de su posterior utilización con Weka. Los datos tal cual están representados no son útiles para su utilización en weka, por lo cual vamos realizar una serie de modificaciones para conseguir un documento con el cual poder trabajar.

Partimos de la versión inicial, tal cual queda después de la utilización del wrapper. En ella existen unos 100.000 vehículos registrados. Estos datos están en bruto y les hacen falta unas modificaciones para su correcta utilización.

- **Se eliminara el campo Edición.** Este campo recoge el sobrenombre del vehículo, la Edición es un campo con un rango de diferentes contenidos muy grande, incluso dentro de un mismo tipo de vehículo, esto viene dado por la falta de formalidad al registrar los anuncios en este campo. En el caso de la Marca y el Modelo la propia página limita las opciones a elegir por lo cual estos campos quedan bien definidos, no como en la Edición que da total libertad para escribir cualquier cosa. Esto a la hora de trabajar en las tareas de clasificación complica mucho el trabajo y empeora los resultados. Al tener ya la Marca y el Modelo del vehículo y las demás características no he creído necesario la inclusión de la Edición para el mejor funcionamiento.
- **Se eliminara el campo Puertas.** Este campo representaba el número de puertas que posee el vehículo. Existe otro campo que es Carrocería que define mejor el tipo de coche y su forma, incluyendo el numero de puertas, y esta mejor limitado los posibles valores, no como en Puertas que es un numero libre el cual puede albergar números exagerados o imposibles.
- A continuación se borraran de la BD todos aquellos **coches que tengan atributos los cuales se salgan de lo normal.** En los anuncios se presupone la veracidad de los datos pero no siempre es así, para llamar la atención o engañar se puede exagerar algún dato, con lo cual no nos serviría para el proyecto porque "ensuciaría" el conjunto de resultados. También serán eliminados aquellos vehículos los cuales sus características, contando que sean reales, **se distancien mucho de la mayoría de las ofertas de otros vehículos**, casos como vehículos de lujo, motores con potencias desorbitadas, vehículos clásicos, etc. Para esta tarea se van a realizar una serie de consultas de borrado en la BD los cuales se especifican a continuación:

  - **Precio :**

```
DELETE *
FROM Vehiculos_recogidos
WHERE precio > 100000 or precio < 350;
```

Con esta consulta se eliminan todos los vehículos cuyo precio exceda los 100.000€ y se sea inferior a 350€.

- **Potencia:**

```
DELETE *
FROM Vehiculos_recogidos
WHERE potencia > 500;
```

Con esta consulta se eliminan todos los vehículos cuya potencia excede los 500 CV.

- **Fecha:**

```
DELETE *
FROM Vehiculos_recogidos
WHERE (fecha_matriculacion < #1/1/1980# Or
fecha_matriculacion > #12/31/2009#);
```

Con esta consulta se eliminan todos los vehículos cuya fecha de matriculación es anterior al año 1980 y posterior que la fecha en la cual se recogieron los datos.

- **Kilómetros:**

```
DELETE *
FROM Vehiculos_recogidos
WHERE kilometros>290000;
```

Con esta consulta se eliminan todos los vehículos con más de 290.000 Km.

- **Consumo:**

```
DELETE *
FROM Vehiculos_recogidos
WHERE consumo >30;
```

Con esta consulta se eliminan todos los vehículos cuyo consumo excede los 30 litros cada 100 Km.

La actualización que se va a llevar a cabo a continuación tiene que ver con el número de marcas y modelos diferentes que se han recogido. El fin de todo este proceso es obtener unos datos útiles para su posterior clasificación con Weka, y la gran variedad de marcas y modelos dificulta este trabajo. Al ser campos muy importantes no se puede prescindir de ellos por lo cual se intenta limitar el abanico de posibilidades. Lo que se ha buscado es quedarse con los vehículos más representativos lo cual se ha comprobado por el número de veces que aparece ese modelo. El criterio que se ha seguido para diferenciar los modelos representativos ha sido de todos los que se han recogido elegir los 100 modelos con mas registros en la BD, con la intención de despreciar los modelos con pocas

apariciones lo que resulta que para estos no se podrían clasificar adecuadamente y además dificultaría lo de los demás modelos.

Para esta modificación en la BD se ha ejecutado la siguiente consulta:

```
DELETE *
FROM Vehiculos_recogidos
WHERE modelo not in (
   SELECT top 100 modelo
   FROM Vehiculos_recogidos
   Group by modelo
   Order by count (*) desc
);
```

Después de comprobar los vehículos guardados en la BD, se ha encontrado un fallo en la recolección de los mismos, existen dos marcas de vehículos Alfa Romeo y Land Rover que tienen la peculiaridad de que el nombre de la marca es un nombre compuesto, para lo cual el wrapper no está preparado y los registra erróneamente, como vehículos con marca "Alfa" y modelo "Romeo". Estos vehículos no corresponden a la realidad y serán borrados de la BD.

Para esta modificación en la BD se ha ejecutado la siguiente consulta:

```
DELETE *
FROM Vehiculos_recogidos
WHERE marca="alfa" OR marca="romeo";
```

Después de esta modificación en la BD nos quedamos con aproximadamente 70.000 vehículos, se ha reducido el número de registros pero también se ha reducido la disparidad de estos.

La BD resultante será la utilizada en un futuro para realizar consultas desde la página web final. Por lo tanto se conservara y las futuras modificaciones se realizaran sobre una copia. Esta versión será llamada "Candidato".

Ver apartado 5.4

El paso siguiente es llevar la BD a una versión en la cual los campos y su contenido estén en el formato adecuado para una vez pasado a texto sean reconocido como un conjunto de valores validos reconocidos por Weka. Para ello se van a realizar una serie de modificaciones:

- Los campos de tipo número se cambiaran a tipo texto. Esto es para poder cambiar el valor asignado para desconocido, que anteriormente era 0, a '?' con el cual es el que trabaja Weka para designar a un valor como desconocido.
- Después de esto se sustituirán todos los caracteres '0' por el carácter '?'.

- Los campos los cuales tengan números decimales el carácter decimal ‘,’ se cambiara por el carácter ‘.’.

Con estos cambios obtenemos una versión de la BD la cual ya puede ser exportada en formato texto para su futura utilización a la cual llamaremos “Banco”.

Existe otra opción para exportar los datos de la BD hasta Weka y es mediante un origen de datos-ODBC, para ellos son necesarios una serie de pasos:

1. Crear el origen de datos-ODBC.
2. Modificar el archivo weka/experiment/ DatabaseUtils.props para tal uso.
3. Abrir desde Weka la base de datos mediante ODBC.

Para información detallada de los pasos consultar el capítulo 14 del manual de Weka.

Utilizando cualquiera de las dos opciones llegamos al mismo estado de los datos.

4 - APRENDIZAJE DE MODELOS

4.1 – ADAPTACIÓN A WEKA

Una vez ya tenemos los datos en un archivo de texto pasamos a adaptarlos para el correcto aprovechamiento de estos en la aplicación Weka.

Al abrir el .txt Weka nos dirá que no es un tipo de archivo valido, con lo cual abra que convertirlo al formato .arff mediante el CSVLoader. En las opciones de este debemos especificar que el carácter para identificar el valor desconocido (missingValue) el cual asignamos el valor '?'. También podemos indicarle los atributos que tendrán valores nominales. Una vez cargado el archivo en Weka y tras comprobar que los atributos responden a lo esperado pasamos a eliminar el atributo ID el cual no es necesario para las funciones de clasificación.

Así ya tendremos una primera versión de los datos para su tratamiento pero esta aun tiene un fallo que es necesario corregir. El atributo Fecha_matriculacion, el cual reconoce como nominal. Para corregir este error guardamos el archivo y pasamos a editarlo manualmente con un editor de texto. El formato para el atributo fecha es el siguiente:

@attribute fecha DATE "dd-MM-yyyy HH:mm"

Ahora ya tenemos el archivo necesario para realizar las siguientes tareas, lo guardamos como *.arff.

Antes de realizar los procesos de clasificación vamos a visualizar distintas estadísticas que se pueden extraer de los datos cargados.

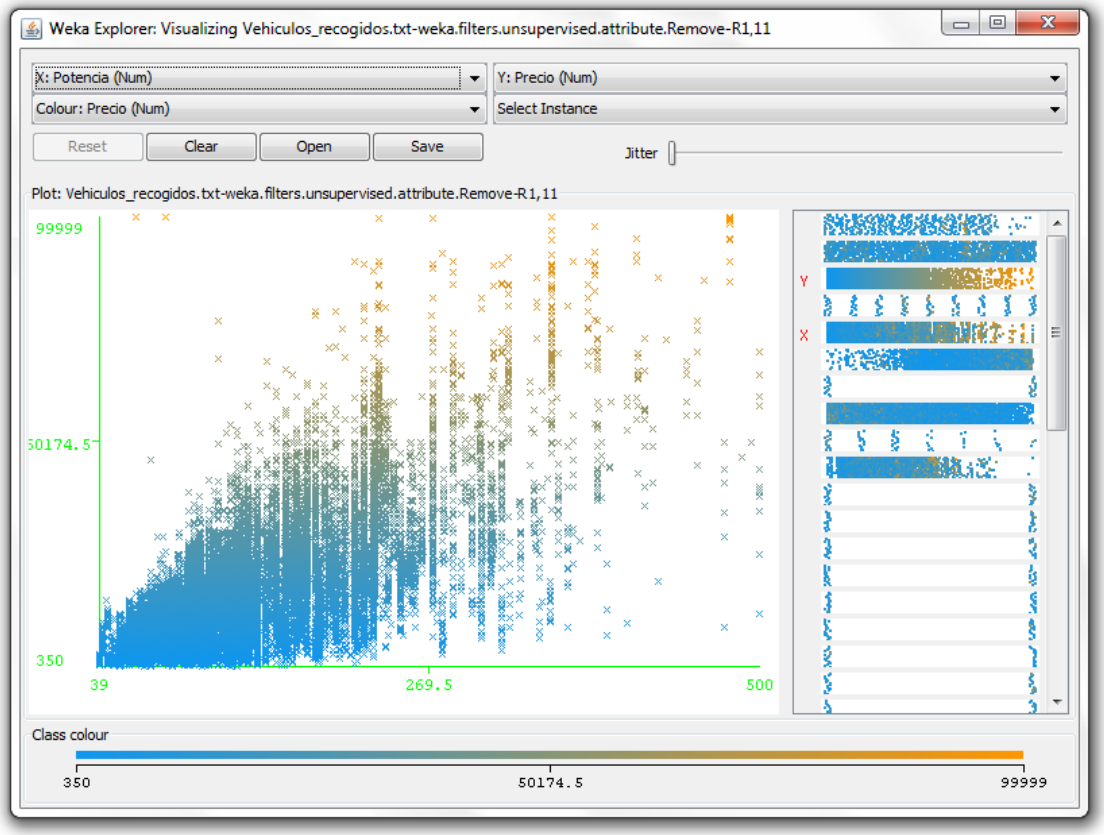

Figura 11: Grafico de la relación entre precio y potencia.

De esta grafica podemos observar que los atributos potencia y precio están bastante relacionados, se nota en la línea ascendente que se obtiene desde los vehículos con menos caballos y más baratos hasta los vehículos más potentes con un aumento claro del precio.

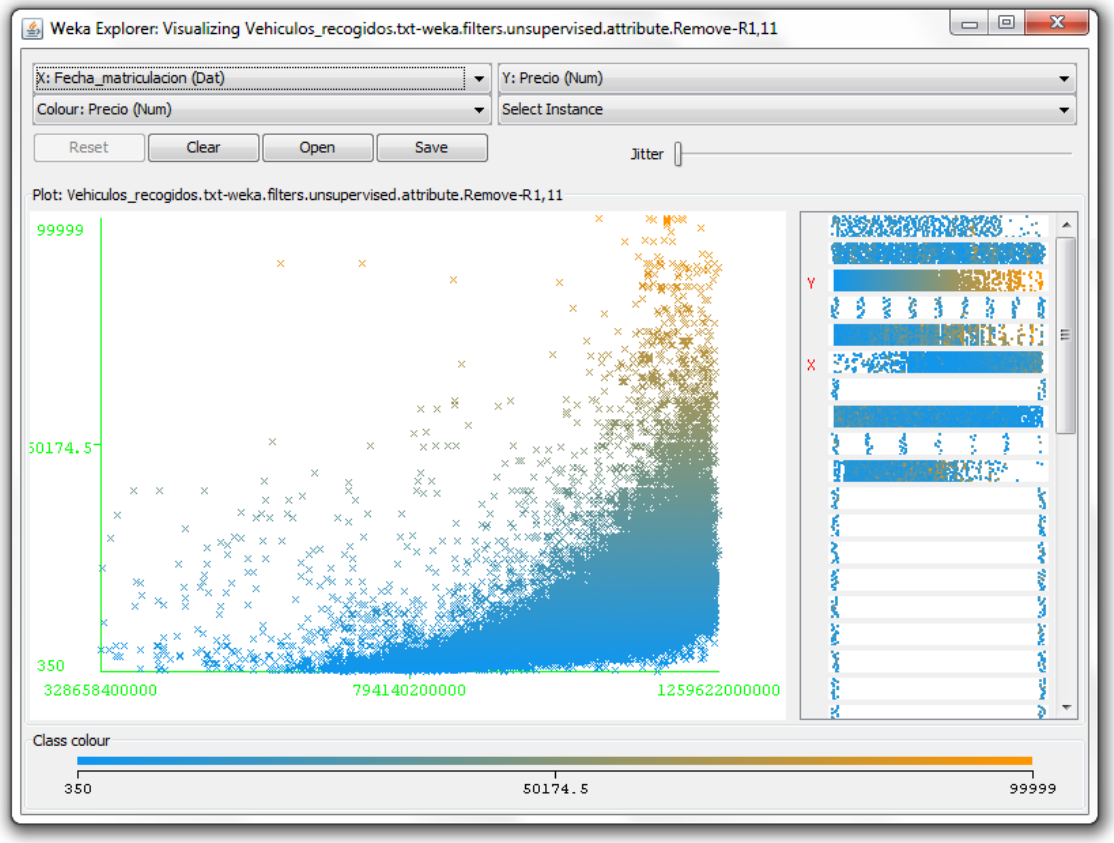

Figura 11: Grafico de la relación entre precio y la fecha de matriculación.

A continuación pasamos a observar la grafica que representan los valores de la fecha de matriculación y el precio del vehículo. Está claro que un coche entre mas nuevo sea mayor valor tendrá, y por el contrario, cuanto más antigüedad tenga ira devaluándose.

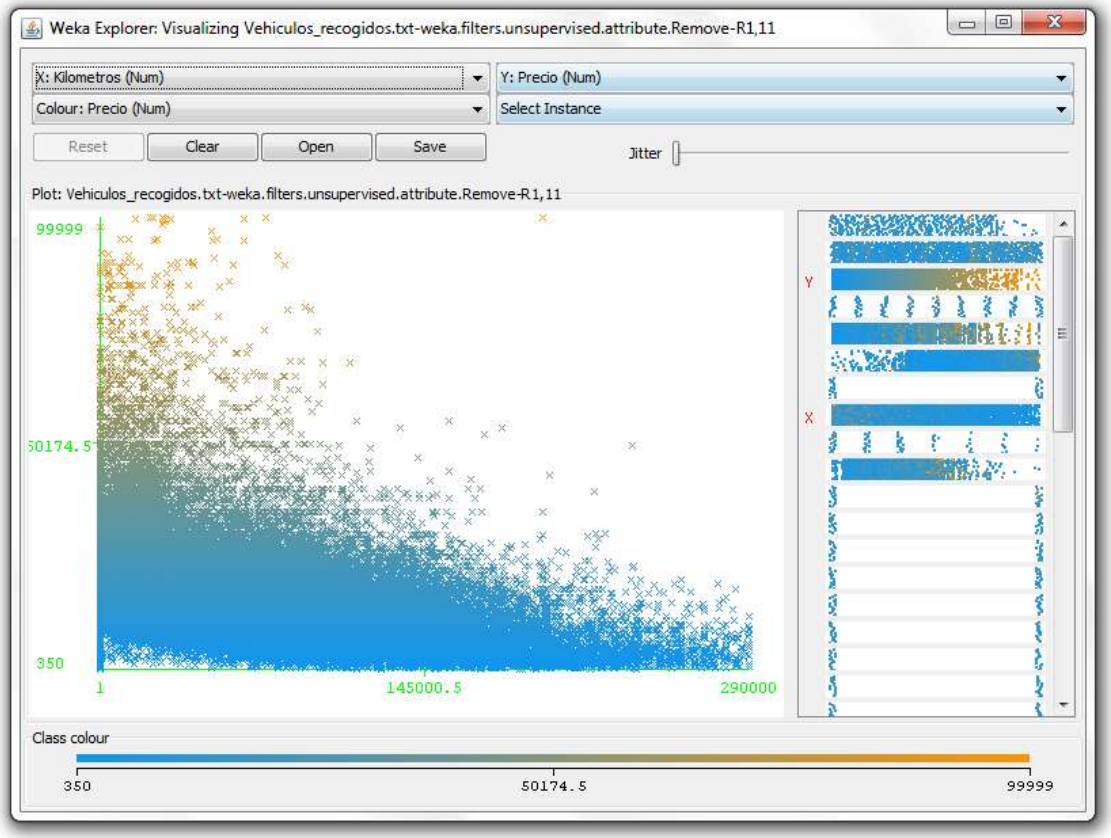

Figura 12: Grafico de la relación entre precio y los kilómetros.

A este grafica le ocurre muy similar a la anterior, en este caso observamos como a mayor kilometraje el valor del vehículo desciende, es normal pues el número de kilómetros es un factor determinante a la hora de la tasación de vehículos de 2º mano.

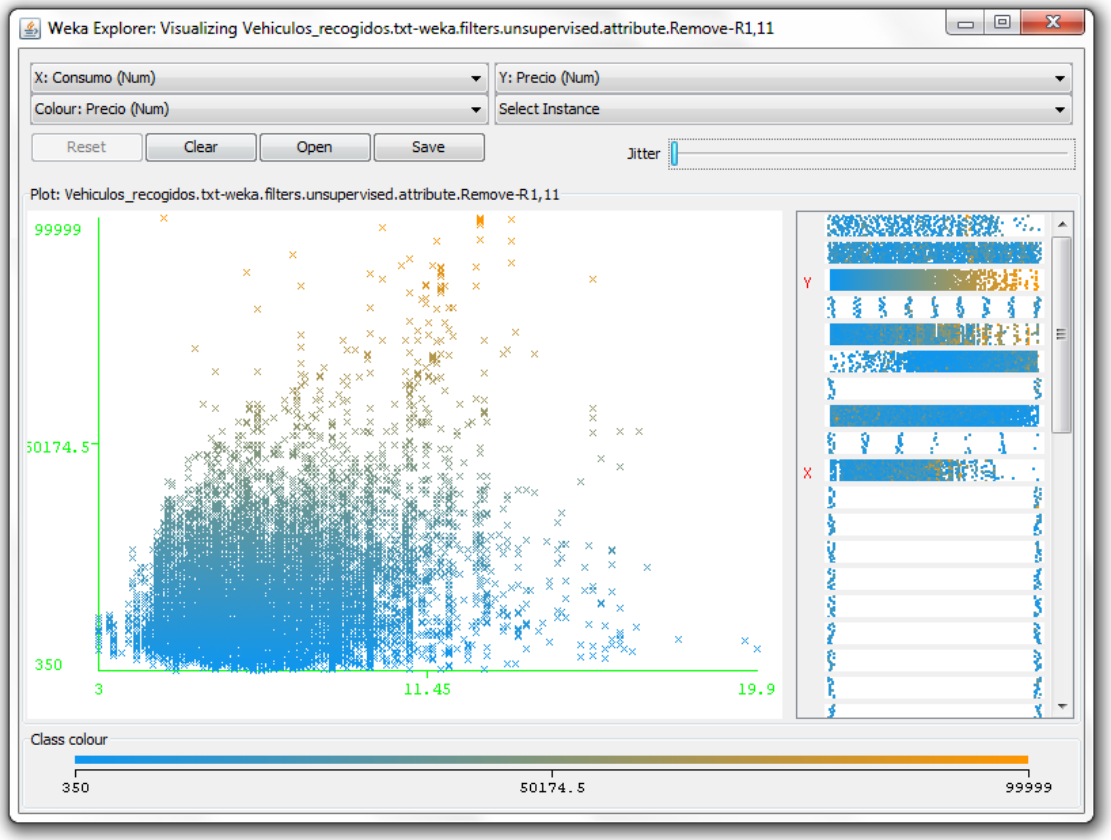

Figura 13: Grafico de la relación entre precio y el consumo.

Y por último la grafica que relación el consumo de combustible y el precio. De esta grafica se puede observar que la mayoría de los vehículos tienen un consumo entre 5 y 12 litros a los 100kms, los vehículos que escapan de este área los conforman los 4x4 y vehículos de gama alta y/o deportivos en los cuales o por exigencias del motor o porque el consumo no es un valor a cuidar.

Después de esto vamos a pasar la parte importante de la sección que es la de la creación de los modelos de predicción.

Weka para la fabricación de los modelos utiliza la práctica estadística de la **validación cruzada (cross-validation).** Esta consiste en partir de una muestra de datos en subconjuntos de tal modo que el análisis es inicialmente realizado en uno de ellos, mientras los otros subconjuntos son retenidos para su uso posterior en la confirmación y validación del análisis inicial. En cada iteración se construirá y evaluara un modelo, usando uno de los conjuntos como test set y el resto como training set. Al final obteniendo la media aritmética de los ratios de error obtenidos conseguiremos el ratio de error para la muestra final.

La elección del valor del número de divisiones dependerá del tamaño y características de la muestra, pero un valor muy utilizado es 10-fold.

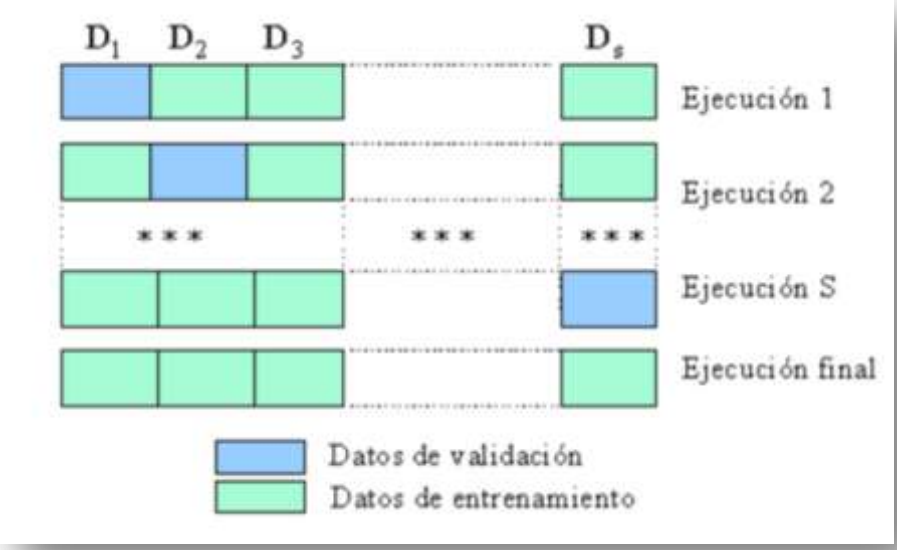

Figura 14: Funcionamiento de la validación cruzada.

Los algoritmos que vamos a utilizar son los siguientes: SMO Reg, SimpleLinealRegresion, IBK (kn=1 y kn=10), ZeroR, M5Rules y M5P.

Antes de mostrar los resultados vamos a definir los parámetros que utiliza Weka para calcular el error del modelo.

- **Correlation coefficient** (Coeficiente correlación): índice que mide la relación lineal entre dos variables aleatorias cuantitativas. A diferencia de la covarianza la correlación es independiente de la escala de medida de las variables.
- **Mean absolute error** (Error absoluto medio): el error absoluto nos indica el grado de aproximación y da un indicio de la calidad de la medida. Indica la media del error producido en cada predicción.
- **Root mean squared error** (Error cuadrático medio): raíz cuadrada de la suma de los cuadrados de los errores individuales de las lecturas, entendiendo por tales a sus diferencias respecto del valor medio medido, que se adopta como valor verdadero convencional.
- **Relative absolute error** (Error absolute relativo): es el cociente entre el error absoluto y el que damos como representativo (la media aritmética).
- **Root relative squared error** (Error cuadrático relativo): es el total de error cuadrático hecho relativo a lo que el error habría sido si la predicción fuese el promedio del valor absoluto.
- **Total Number of Instances** (Número total de instancias): El numero de ejemplos que se han utilizado en la predicción.

Con los datos sobre vehículos y utilizando Weka con los distintos algoritmos estos son los resultados.

SMO Reg:

Este algoritmo no soporta las predicciones con atributos de tipo fecha (DATE).

SimpleLinealRegresion:

Correlation coefficient	0.9029
Mean absolute error	2792.6957
Root mean squared error	4287.6535
Relative absolute error	39.5383 %
Root relative squared error	42.9848 %
Total Number of Instances	68733

IBK (knn=1):

Correlation coefficient	0.7733
Mean absolute error	4003.096
Root mean squared error	6722.1153
Relative absolute error	56.6748 %
Root relative squared error	67.3908 %
Total Number of Instances	68733

IBK (knn=10):

Correlation coefficient	0.843
Mean absolute error	3392.9218
Root mean squared error	5365.5369
Relative absolute error	48.0358 %
Root relative squared error	53.7909 %
Total Number of Instances	68733

ZeroR:

Correlation coefficient	0.0052
Mean absolute error	7063.2703
Root mean squared error	9974.8216
Relative absolute error	100 %
Root relative squared error	100 %
Total Number of Instances	68733

M5Rules:

Correlation coefficient	0.951
Mean absolute error	1888.1222
Root mean squared error	3083.7048
Relative absolute error	26.731 %
Root relative squared error	30.9147 %
Total Number of Instances	68733

M5P:

Correlation coefficient	0.9535
Mean absolute error	1837.5394
Root mean squared error	3007.1848
Relative absolute error	26.0152 %
Root relative squared error	30.1478 %
Total Number of Instances	68733

4.4 – OBSERVACIONES

Viendo los resultados obtenidos con los diferentes modelos se puede llegar a la conclusión de que los algoritmos que mejor resultados dan son el M5P y el M5Rules. A pesar de esto en la aplicación web están disponibles todos los modelos para realizar pruebas y comparar resultados.

Uno de los problemas que puede aparecer al usar Weka con **grandes volúmenes de datos** es que se desborde la memoria virtual de Java, esta memoria tiene un valor inicial por defecto cuando se instala Java pero los al cargar los archivos de vehículos recogidos de esta aplicación esta memoria no es suficiente y como resultado no nos dejara trabajar. El mensaje de error es el siguiente.

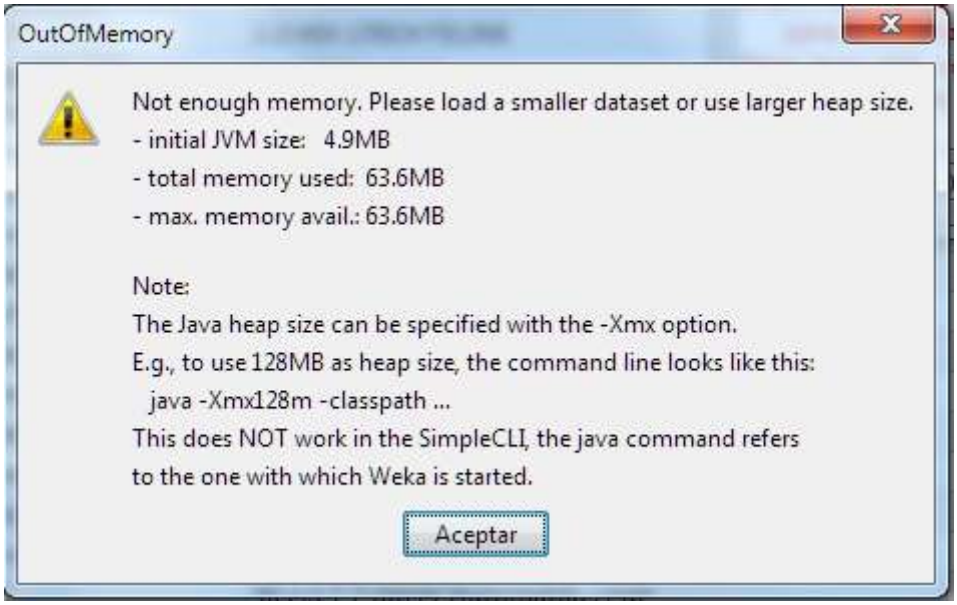

Figura 15: Mensaje de error producido.

Para solucionar esta situación es necesario modificar el archivo RunWeka.ini que se encuentra en la raíz del directorio de instalación. Se puede modificar con un simple editor de texto. El cambio requerido se encuentra en la línea que define el maxheap. En la versión 3.6.3 de Weka el valor inicial es de 256MB pero no es suficiente, para trabajar adecuadamente lo cambiamos a 1024MB (se recomienda utilizar múltiplos de 2 naturales de los tamaños de memoria 256, 512, 1024, 2048, etc.) y dejamos la línea del archivo así:

maxheap=1024m

Con esto ya no tendremos problemas de memoria con la maquina virtual de Java.

5 - APLICACIÓN DE MODELOS

En este apartado se ha desarrollado una página **web con tecnología JSP**, en ella se ha creado una serie de formularios para especificar los atributos de un vehículo concreto el cual será el utilizado para la predicción. Una vez recogidos los datos sobre el vehículo del cual se quiere saber su valor estimado en el mercado se pasa a realizar la predicción y la muestra de resultados. Se ha añadido una lista de vehículos de la base de datos de los cuales más se acercan al vehículo de la consulta y un link del anuncio en www.autoscout24.com .

5.1 - PAGINA DE INICIO

Figura 16: Página de inicio de la aplicación.

En la página de inicio de la web se muestra una lista desplegable en la cual elegir la marca del vehículo de la predicción. Esta elección acotara el número de modelos solo a los correspondientes de la propia marca lo cual facilita mucho la búsqueda e imposibilita las búsquedas de vehículos con marcas y modelos no existentes.

Figura 17: Página de selección de las características del vehículo.

A continuación se muestra una página donde se pasa a seleccionar todos los atributos del vehículo (menos la marca). Primero se da la opción de elegir las características principales como el modelo, fecha de matriculación, etc. Después se muestra una lista de los complementos disponibles para buscar, al existir una gran cantidad no es recomendable seleccionar muchos campos porque dificultaría la búsqueda de coincidencias en la base de datos.

En la parte de abajo se da la elección de elegir uno entre los algoritmos disponibles los cuales son M5P, ZeroR, Linear Regresion, M5Rules y IBK en dos versiones con knn=1 y knn=10.

Audi A3 | El resultado de la predicción es : 10289 €

Vehiculos recomendados

Marca	Modelo	Carroceria	Potencia	Fecha Matriculación	Cambio	Kilometros	Combustible	Consumo	Precio	Link
Audi	A3	2/3 puertas	116.0	07-2004	Manual	152694	Gasolina	0.0	11945 €	Enlace a anuncio
Audi	A3	2/3 puertas	116.0	07-2004	Manual	152694	Gasolina	0.0	11945 €	Enlace a anuncio
Audi	A3	2/3 puertas	116.0	07-2004	Manual	152694	Gasolina	0.0	11945 €	Enlace a anuncio
Audi	A3	2/3 puertas	116.0	07-2004	Manual	152694	Gasolina	0.0	11945 €	Enlace a anuncio
Audi	A3	2/3 puertas	116.0	07-2004	Manual	152694	Gasolina	0.0	11945 €	Enlace a anuncio
Audi	A3	2/3 puertas	116.0	07-2004	Manual	152694	Gasolina	0.0	11945 €	Enlace a anuncio
Audi	A3	2/3 puertas	102.0	07-2004	Manual	69500	Gasolina	7.1	12450 €	Enlace a anuncio
Audi	A3	2/3 puertas	150.0	07-2004	Manual	99841	Gasolina	7.4	14749 €	Enlace a anuncio
Audi	A3	2/3 puertas	250.0	07-2004	Manual	116000	Gasolina	0.0	18890 €	Enlace a anuncio

Nueva Busqueda

Figura 18: Página de resultados de la predicción.

Y por último la página donde se muestran los resultados. En la parte superior se muestra el precio estimado del vehículo con los datos anteriormente facilitados. Esta predicción es orientativa puede variar de la realidad.

En la parte central se lista una serie de vehículos los cuales coinciden con las características del vehículo buscado. Estos vehículos son ejemplos reales almacenados en la base de datos y se muestran los 10 con menor precio, de cada coche se muestran las principales características como un enlace al anuncio.

5.4 – CONSTRUCCIÓN DE LA PAGINA WEB

Para la realización de esta página web se ha utilizado el entorno de **NetBeans** en su versión 6.9.1. Se ha implementado en tecnología JSP lo cual es lo más lógico pensando que **Weka basa su funcionamiento interno en Java.**

Para las funciones de predicción se han utilizado las funciones del **API de Weka**, añadiendo el archivo weka.jar al proyecto. Para una mejor comprensión del API y poder ver al completo la funcionalidad de este se recomienda consultar el manual de Weka el capitulo 16.

Para el correcto funcionamiento de la pagina es necesario tener un servidor JSP activo, como por ejemplo **TomCat**, y crear un origen de datos ODBC de tipo Access llamado "BD" que apunte a la base de datos que preparamos anteriormente para el caso como comentamos en el apartado 3.4 .

En esta memoria no se va a profundizar sobre la instalación de servidores web en un PC, existe gran cantidad de páginas en la red en las cuales tratan del tema, aquí se van a dejar un par de enlaces que permite la descarga del servidor web TomCat, su instalación y configuración.

Descarga:

http://tomcat.apache.org/download-60.cgi

Guía instalación:

http://www.proactiva-calidad.com/java/herramientas/tomcat/index.html

http://chuwiki.chuidiang.org/index.php?title=Instalaci%C3%B3n_de_Tomcat_en_Windows

Para el correcto funcionamiento de la web, es necesario configurar un origen de datos ODBC, este es necesario para la comunicación entre la página web y la base de datos utilizada. Como hemos comentado antes no se va a extender en la explicación de su configuración, en el siguiente enlace se explica cómo se realiza.

http://www.webtaller.com/construccion/lenguajes/java/lecciones/como_conectar_java_access.php

También es necesario copiar la carpeta "Archivos para PFC" a la raíz de la unidad C. En este directorio se encuentran los diferentes modelos de predicción y un ejemplo de test utilizado por la aplicación.

5.5 –DESARROLLO DE LA FUNCION DE PREDICCION

En este apartado se va a explicar cómo se ha llegado a desarrollar la sección encargada de realizar la predicción sobre el precio de los vehículos. La implementación de esta se ha desarrollado mediante el lenguaje de programación JAVA, a continuación se va a mostrar el código necesario para tal cometido acompañado de comentarios explicativos.

El primer paso es cargar el archivo test.arff, este archivo sirve como base para introducir los datos del vehículo a evaluar. No importa su contenido, puesto que será cambiado, solo que contenga la estructura adecuada.

```
        try{
            test = DataSource.read("C:/Archivos para PFC/test.arff");
           }
        catch(Exception e){
            texto+="<br>Test no encontrado. "+e;
        }
```

Con esta instrucción se marca el precio como la clase a evaluar, el precio es el 3º atributo, con lo cual tiene el número 2 en el índice.

```
        if (test.classIndex() == -1) test.setClassIndex(2);
```

Este es un bucle que recorre todos los atributos del test y les atribuye la condición de valor desconocido (missing), que por defecto es el carácter '?'.

```
        for(int i=0;i<test.numAttributes();i++){
            test.firstInstance().setMissing(i);
        }
```

Comprueba que el campo modelo no tenga solo cifras numéricas, de ser así existe un fallo en la evaluación y requiere que se le añada la cadena ".0" al final.

```
            if(esNumero(modelo)){

                modelo_p=modelo+".0";

            }
```

En estas instrucciones se modifica el valor de cada atributo del test con el introducido en el formulario de la página web. Por cada atributo existe una línea que comprueba que existe ese valor introducido y lo asigna en su correspondiente parte del set.

```
            try{

            if(!marca.equals("")) test.firstInstance().setValue(0, marca);

            if(!modelo_p.equals(""))test.firstInstance().setValue(1, modelo_p);

            if(!carroceria.equals("")) test.firstInstance().setValue(3, carroceria);

            if(!potencia.equals(""))
test.firstInstance().setValue(4,Double.parseDouble(potencia));

            if(!mes.equals("") && !año.equals("")) { fecha="01-"+mes+"-"+año+" 00:00";
test.firstInstance().setValue(5, test.attribute(5).parseDate(fecha)); }

            if(!cambio.equals("")) test.firstInstance().setValue(6, cambio);

            if(!kilometros.equals("")) test.firstInstance().setValue(7,
Double.parseDouble(kilometros));

            if(!combustible.equals("")) test.firstInstance().setValue(8, combustible);

            if(!consumo.equals("")) test.firstInstance().setValue(9,
Double.parseDouble(consumo));

            if(wd4!=null) test.firstInstance().setValue(10,"1.0");

            if(airbag!=null) test.firstInstance().setValue(11, "1.0");

            if(airbag_a!=null) test.firstInstance().setValue(12, "1.0");

            if(airbag_l!=null) test.firstInstance().setValue(13, "1.0");

            if(aire_a!=null) test.firstInstance().setValue(14, "1.0");

            if(cierre_c!=null) test.firstInstance().setValue(15, "1.0");

            if(climatizador!=null) test.firstInstance().setValue(16, "1.0");

            if(direccion_a!=null) test.firstInstance().setValue(17, "1.0");

            if(elevalunas!=null) test.firstInstance().setValue(18, "1.0");

            if(xenon!=null) test.firstInstance().setValue(19, "1.0");

            if(llantas!=null) test.firstInstance().setValue(20, "1.0");

            if(aparcar_a!=null) test.firstInstance().setValue(21, "1.0");
```

```
            if(navegador!=null) test.firstInstance().setValue(22, "1.0");
            if(asientos_cuero!=null) test.firstInstance().setValue(23, "1.0");
            if(techo_s!=null) test.firstInstance().setValue(24, "1.0");
            if(abs!=null) test.firstInstance().setValue(25, "1.0");
            if(adap_d!=null) test.firstInstance().setValue(26, "1.0");
            if(alarma!=null) test.firstInstance().setValue(27, "1.0");
            if(asientos_calef!=null) test.firstInstance().setValue(28, "1.0");
            if(asientos_ele!=null) test.firstInstance().setValue(29, "1.0");
            if(baca!=null) test.firstInstance().setValue(30, "1.0");
            if(bizona!=null) test.firstInstance().setValue(31, "1.0");
            if(bola!=null) test.firstInstance().setValue(32, "1.0");
            if(control_tra!=null) test.firstInstance().setValue(33, "1.0");
            if(control_vel!=null) test.firstInstance().setValue(34, "1.0");
            if(esp!=null) test.firstInstance().setValue(35, "1.0");
            if(anti_niebla!=null) test.firstInstance().setValue(36, "1.0");
            if(inmobilizador!=null) test.firstInstance().setValue(37, "1.0");
            if(ordenador_bordo!=null) test.firstInstance().setValue(38, "1.0");
            if(radio!=null) test.firstInstance().setValue(39, "1.0");
            if(radio_cd!=null) test.firstInstance().setValue(40, "1.0");
            if(tunning!=null) test.firstInstance().setValue(41, "1.0");
            }
            catch(Exception e){
                texto+="error al cambiar los datos del test. "+e;
            }
```

En la última sección se carga el algoritmo de clasificación que se ha elegido en el formulario web. Estos archivos tienen que estar creados con Weka como corresponde y situados en el directorio que se muestra.

Y finalmente se invoca la función "classifyInstance()" pasándole como atributo la primera instancia que se encuentra en el test, recordemos que solo existe una instancia en el. La función devuelve un numero doble que se trata para convertirlo en un String para su muestra en la página de resultados.

```
            try{

                Classifier cls = (Classifier) SerializationHelper.read("C:/Archivos para
PFC/"+algoritmo+".model");

                double pred = cls.classifyInstance(test.instance(0));

                int precio = (int)pred;

                texto += String.valueOf(precio)+" €.";

            }

            catch(Exception e){

                texto+="<br>Error al construir la evaluacion.  "+e;

            }


        return texto;
```

6 - CONCLUSIONES

Nuestra capacidad para almacenar datos ha crecido en los últimos años a velocidades exponenciales. Por el contrario, la capacidad del ser humano para procesarlas y asimilarlas sigue siendo constante. Por este motivo la minería de datos se presenta como una tecnología de apoyo para explorar, analizar, comprender y aplicar el conocimiento obtenido usando grandes volúmenes de datos. Y con ello descubrir nuevos caminos que conduzcan al conocimiento deseado.

En el caso de este proyecto final de carrera se ha abordado el tema de los vehículos de ocasión. Existe una gran oferta de vehículos de ocasión en el mercado, páginas de internet, anuncios en periódicos, ofertas de concesionarios, ofertas de particulares, etc. La mayoría de la población no tiene un conocimiento suficiente como para por el mismo llegar a una conclusión razonable y valida sobre el precio actual de un vehículo. Como hemos visto, estos tienen una gran cantidad de características que deben ser tenidas en cuenta para llegar a una valoración. A nadie le gustaría realizar una compra de un vehículo y luego descubrir que ha pagado mucho mas del valor real de este. O saber que has estado delante de una ganga y no la has aprovechado. El fin de esta aplicación es dar la información necesaria para poder realizar decisiones adecuadas.

En cuanto a la realización del proyecto en si ha sido una experiencia positiva. La minería de datos es un campo muy interesante de la cual no se habla lo suficiente durante los años de carrera. Gracias a este proyecto he podido estudiar sobre el tema, no tan profundamente como para ser un experto, pero si como para tener una idea de que trata y de su importancia. En cuanto a aspectos técnicos se han utilizado varios lenguajes de programación (C# y Java), desarrollo de aplicaciones (ejecutable en Windows y pagina web) y utilización de varios programas (Access, Weka, procesadores de texto, etc.) lo cual siempre es positivo para la formación como informático.

7 - BIBLIOGRAFÍA

[1] *Hernandez Orallo, J., Ramírez Quintana, M. J., & Ferrí Ramírez, C. (2004). Introducción a la Minería de Datos. Pearson Educación. S.A.*

[2] *Duque Méndez, N.D., Chavarro Porras, J.C., Moreno Laverde, R. (2007). Integrando información de fuentes heterogéneas enfoques y tendencia.*

[3] *Manual de Weka en castellano:* http://www.metaemotion.com/diego.garcia.morate/

[4] *Sofía J. Vallejos (2006). Trabajo de Adscripción Minería de Datos. Universidad Nacional del Nordeste Facultad de Ciencias Exactas, Naturales y Agrimensura. Argentina.*

[5] *Página de documentación oficial sobre Weka:* http://www.cs.waikato.ac.nz/ml/weka/

[6] http://es.wikipedia.org/wiki/Miner%C3%ADa_de_datos

[7] http://irsweb.blogspot.com/2005/10/qu-son-los-wrappers.html

[8] http://resources.metapress.com/pdf-preview.axd?code=dr5r02w2raqmw1wd&size=largest

[9] *Blog de Luis Domingo García del Pino* http://www.luisdogarcia.com.es

[10] http://www.uoc.edu/web/esp/art/uoc/molina1102/molina1102.html