PROYECTO CARRETERA

Tratamiento Inteligente de Datos
Proyecto Final

Guillermo Esquivel González
Óscar Darias Plasencia
Sergio García de la Iglesia
Joaquín Sanchiz Navarro
Eduardo de la Paz González

24 de Mayo de 2017

ÍNDICE

1. INTRODUCCIÓN

En muchas situaciones, es interesante saber si se puede llegar a tiempo a un determinado destino, conociendo las condiciones que va a tener el viaje que vamos a realizar. Ya sea con aplicaciones comerciales o personales, poder conocer de antemano esta información es algo que podría interesar a muchas entidades. En el caso de las empresas, por ejemplo, sabemos que la optimización es un aspecto clave de prácticamente todas sus operaciones, y por eso consideramos que este proyecto, llevado adecuadamente, tiene un alto potencial de ser realmente útil.

El objetivo de este proyecto de minería de datos es el construir un modelo predictivo que nos permita estimar el tiempo necesario para viajar de un punto a otro de la isla de Tenerife. La minería de datos o tratamiento inteligente de datos nos permite realizar un análisis de grandes cantidades de datos para extraer patrones interesantes hasta ahora desconocidos. Estos patrones pueden ser vistos como una especie de resumen de los datos de entrada, que puede adaptarse para el aprendizaje automático y análisis predictivo, que es lo que nos interesa.

Entonces, para poder estimar el tiempo necesario para realizar un viaje, necesitaremos conocer una serie de parámetros acerca del mismo, como puede ser la calidad del medio de transporte utilizado, las condiciones climáticas, el momento del día, etc. Evidentemente, no es lo mismo realizar un viaje de un punto a otro un fin de semana a mediodía, que un día laboral a las ocho de la mañana. En la sección de datos de este documento se detalla con todo lujo de detalles cada uno de los parámetros que se consideraron necesarios, así como aquellos que fueron descartados y por qué.

Para nuestra primera versión del proyecto hemos decidido realizar la toma de datos en la isla de Tenerife (Islas Canarias), lugar donde vivimos y estudiamos. Básicamente no podemos permitirnos el poder viajar a otros lugares para recopilar datos actualmente. Además, como hemos comentado, partimos con la ventaja de que llevamos 20 años viviendo aquí, por lo que poseemos conocimientos de antemano sobre muchos factores en esta zona, como el clima o el tráfico, por lo que se convierte en el candidato ideal para realizar una primera versión del proyecto.

La isla de Tenerife ha resultado ser un candidato óptimo para la diversificación del proyecto por diferentes factores. Posee diferentes tipos de carreteras por lo que podemos estudiar varios tipos de rutas en vez de tan solo unas pocas. Además, en la isla disponemos de un clima bastante variable, lo cual nos permite poder tomar datos en distintas situaciones. Respecto a las distancias entre los puntos hemos podido tomar datos de trayectorias de más de 100 kilómetros.

Una vez recogidos todos los datos, hemos hecho uso del software Weka para obtener los resultados. Weka (*Waikato Environment for Knowledge Analysis*, en español «entorno para análisis del conocimiento de la Universidad de Waikato»), es una plataforma de software para el aprendizaje automático y la minería de datos escrito en Java y desarrollado por la Universidad de Waikato. Se trata de un software libre distribuido bajo la licencia GNU-GPL, que pone a nuestra disposición las herramientas suficientes como para elaborar el modelo que necesitamos. Todo esto, evidentemente, a partir de una completa base de datos que habrá que reunir.

Los puntos fuertes de Weka son:

- Está disponible libremente bajo licencia pública general de GNU.
- Es muy portable porque está completamente implementado en Java y puede correr en casi cualquier plataforma.
- Contiene una extensa colección de técnicas para preprocesamiento de datos y modelado.
- Es fácil de utilizar por un principiante gracias a su interfaz gráfica de usuario.

Weka soporta varias tareas estándar de minería de datos, especialmente, preprocesamiento de datos, clustering, clasificación, regresión, visualización y selección. Todas las técnicas de Weka se fundamentan en la asunción de que los datos están disponibles en un fichero plano o una relación, en la que cada registro de datos está descrito por un número fijo de atributos.

Concretamente, hemos utilizado el algoritmo REPTree, un algoritmo basado en árboles de regresión que Weka pone a nuestra disposición. Consúltese la sección correspondiente al algoritmo para más información sobre el funcionamiento del mismo y el porqué de su elección.

En este documento vamos a analizar todo el proceso de planificación del modelo para ver las razones por las cuales se seleccionaron finalmente las variables del modelo y cuáles fueron descartadas. Además, veremos los principales métodos utilizados para obtener los datos. Después, hablaremos un poco acerca del algoritmo utilizado para generar el modelo, para finalmente analizar los resultados obtenidos.

2. LOS DATOS

En este apartado vamos a ver qué datos se consideraron necesarios para construir un buen modelo. La gran mayoría de ellos son bastante evidentes y se relacionan directamente con aquellos aspectos que hacen que un viaje dure más o menos tiempo, pero algunos han sido descartados y otros han cambiado en ciertos aspectos a lo largo del desarrollo del proyecto.

DESCRIPCIÓN

La minería de datos o tratamiento inteligente de datos se caracteriza principalmente por el manejo de grandes volúmenes de datos. Esto implica que, cuando vamos a realizar un proyecto en este campo, lo primero que tenemos que hacer es reunir una cantidad de datos lo bastante grande como para poder trabajar convenientemente. Dependiendo del tipo de proyecto que se esté llevando a cabo, se podría necesitar una mayor o menor cantidad de datos.

Sabemos que el objetivo final de nuestro proyecto es obtener una estimación del tiempo necesario para llevar a cabo un viaje de un punto a otro de la isla de Tenerife. Esto hace que el valor objetivo de cada una de las entradas de nuestra base de datos va a ser un valor numérico, no nominal, con un dominio que podría oscilar entre 1 minuto y, digamos como máximo, 180 minutos. Este es un rango bastante grande, que ya nos sugiere que el número de entradas necesarias para generar un buen modelo va a ser bastante alto. La complejidad aumenta si tenemos en cuenta la enorme variedad de viajes posibles y las variaciones entre ellos.

Por tanto, sabemos que para elaborar un modelo 100% fiable, el número de entradas tendría que ser muy alto. En el apartado de *Obtención de los datos* de esta sección veremos un poco más en profundidad el número exacto de entradas reunidas, cómo las obtuvimos y las posibles formas de mejora.

Sin embargo, antes de eso, vamos a ver exactamente qué variables hemos creído necesarias en cada entrada para poder elaborar el modelo. Cada entrada de la base de datos, en minería de datos como en otras áreas de la informática donde se hace uso de teorías de bases de datos, tiene una serie de variables con un dominio bien definido. Las diferencias entre los valores de las variables son lo que diferencia a las entradas entre sí, y son lo que permite a los principales algoritmos de minería de datos identificar patrones y mapear los valores de las entradas en conclusiones acerca del valor objetivo de esas entradas. Este es el caso concreto de los árboles de decisión, de los que ya hablaremos más en profundidad en la sección *Weka* de este informe.

Las variables de nuestras entradas están directamente relacionadas, como no podía ser de otra manera, con las condiciones del viaje. A partir de esas condiciones, se debería poder estimar el tiempo necesario para el viaje. En los siguientes apartados vamos a ver qué variables se consideraron, cuáles fueron descartadas y cuáles forman parte de la base de datos final. Hay que tener en cuenta que, a lo largo del desarrollo del proyecto, la selección

de los datos relevantes para construir el modelo ha sido un aspecto que ha ido evolucionando constantemente.

VARIABLES DESCARTADAS

Las condiciones que pueden afectar a un viaje por carretera son bastante sencillas de intuir. Es bastante evidente que no es lo mismo conducir con un clima soleado que con un clima lluvioso, al igual que no es lo mismo realizar un determinado trayecto un día laboral a las 7 de la mañana que un fin de semana a esa misma hora.

Sin embargo, muchas de las variables que se puede intuir que afectarían al tiempo necesario para el viaje, hemos decidido que deberían desestimarse en este proyecto. Las razones son bastante variadas dependiendo de la variable en cuestión, así que vamos a hablar de cada una de ellas:

- **Origen y destino**

Nuestro primer análisis de requisitos nos llevó a pensar que era fundamental conocer el lugar de origen y lugar de destino del viaje, para así asegurar la estimación en función del resto de parámetros. Más adelante, nos dimos cuenta de que cubrir toda la isla de Tenerife en cuestión de localizaciones habría requerido de miles y miles de entradas en nuestra base de datos. Hay que tener en cuenta que no solo necesitaríamos entradas para todos los municipios de la isla, sino que además dentro de cada municipio los tiempos serían bastante diferentes para cada barrio o incluso cada calle en ciertos casos.

Además, el desplazamiento no es tanto una cuestión de origen y destino, sino de distancia y terreno. Dos viajes con orígenes y destinos diferentes, para una misma distancia y un mismo tipo de terreno, duran aproximadamente lo mismo. Entonces, decidimos eliminar las variables de origen y destino, otorgándole mayor importancia a las variables distancia y terreno, que veremos en el siguiente apartado con más detenimiento.

Para futuras versiones del proyecto, añadir estas variables podría asegurar bastante la precisión de los resultados. Sin embargo, sería estrictamente necesario encontrar una forma más sencilla de reunir los datos, pues como ya veremos en próximos apartados, los métodos de los que hemos dispuesto hasta ahora para reunir datos hacen extremadamente complicado obtener miles de ellos.

- **Medio de transporte**

El medio de transporte utilizado para realizar el viaje también fue un factor que consideramos importante durante bastante tiempo. De hecho, a día de hoy seguimos pensando que sería un añadido bastante bueno al modelo, permitiendo al usuario

especificar si se desplaza en coche o guagua. Las razones del descarte de esta variable no tienen nada que ver con que acabase siendo irrelevante.

El problema que tenía añadir esta variable era bastante similar al problema que nos encontramos con las variables de origen y destino. El desplazamiento en guagua implica necesariamente tener en cuenta otro grupo de factores: el horario disponible para las guaguas, la frecuencia de salida de las mismas en el horario en el que se da el desplazamiento, las posibles escalas o transbordos...

Todo esto hacía que el número de entradas necesarias para construir el modelo aumentase de forma exponencial. Como en el caso de las variables de origen y destino, para incluir esta "funcionalidad extra" necesitaríamos encontrar una forma más sencilla y automatizada de reunir datos. Además, para solventar el problema de los horarios de las guaguas en las distintas localizaciones de la isla, sería necesario incluir previamente las variables de origen y destino.

- **Fuente de los datos y calidad de la misma**

En un principio, también se valoró la posibilidad de diferenciar entre las fuente de datos de la que procede cada entrada, así como la calidad de la misma (entendiendo calidad como fiabilidad). Esto se haría por medio de dos variables nuevas: la primera de ellas nominal, especificando la fuente de donde se obtuvieron los datos de la entrada; otra numérica, con un dominio entre 0 y 1, entendiendo 0 como calidad nula y 1 como calidad máxima.

Esta idea fue descartada rápidamente debido a que todas las entradas de nuestra base de datos las obtuvimos de dos fuentes concretas, ambas bastante fiables. Por tanto, estas dos variables habrían terminado siendo dos constantes, lo que las haría irrelevantes para la construcción del modelo.

VARIABLES ACEPTADAS

Finalmente tras haber estado barajando la utilización de distintas variables, nuestro modelo final cuenta con las siguientes:

Terreno del viaje, con esta variable tratamos de especificar una diferencia entre un trayecto por ejemplo en carretera y uno en montaña. Aunque el terreno puede variar durante un mismo viaje, tomaremos aquel que sea predominante. De esta manera, podemos distinguir diferentes trayectos en función de si vamos por una carretera, por una ciudad o el trayecto es de montaña. Es evidente que el tipo de carretera afectará en la duración del trayecto de una manera mejor o peor, siendo el trayecto por carretera el trayecto estándar, el trayecto de montaña más lento debido al tipo de la calzada y a los diferentes cambios y variaciones en el terreno y el trayecto por ciudad, siendo este último normalmente bastante complicado debido a la alta densidad que normalmente afecta a las zonas de la ciudad.

- El trayecto por **carretera**, como es normal en Tenerife, abarca la mayoría de los trayectos de los cuales hemos recogido datos. Esta clase abarca todas aquellas carreteras que permitan el tráfico a velocidades superiores a los 50 km/h.

- El trayecto en **montaña**, aunque Tenerife sea una isla muy abrupta y con zonas de montaña, es el menos predominante en comparación con el trayecto por carretera. En cambio, decidimos incluir esta categoría debido a que, como es de esperar, un trayecto por montaña contiene muchas más curvas y la velocidad del tráfico es mucho menor, por lo que hay una gran diferencia.

- El trayecto por **ciudad** es similar al trayecto por carretera en algunos casos, pero la presencia de semáforos, peatones y otros elementos propios de las zonas más urbanas hacen que la diferencia entre un trayecto por ciudad y un trayecto por carretera sea un factor a considerar.

La siguiente variable que hemos tenido en cuenta en nuestro proyecto es el **momento del día**. Es evidente que no es lo mismo realizar un trayecto a la 1 de la tarde que en la madrugada, por lo que hemos diferenciado entre 5 posibles categorías que engloban las 24 horas del día. A la hora de definir los rangos de cada categoría nos hemos fijado en la variación del tiempo de viaje en cada franja de horas del día. Por lo tanto se consideran las siguientes posibilidades:

- Horario **matutino**, que abarca desde las 6 a las 9 de la mañana. Un horario que, en días laborales, suele tener bastante importancia dado que el tráfico aumenta por la gente que va al trabajo, colegios, etc.

- Horario de **mañana**. Este engloba las horas desde las 9 a la 1 de la tarde. Suele ser un horario más fluido para el tráfico que el anterior. Podría verse como la franja horaria existente entre los horarios de tráfico más alto en días laborales.

- Horario de **media jornada**. Es uno de los puntos con más congestión del día. Incluye las horas comprendidas entre la 1 y las 6 de la tarde, y debido a que suele ser el horario de salida de trabajos de media jornada o pausas para almorzar, además de salida masiva de alumnos de colegios. Todo esto puede llegar a un grado de congestión casi igual que al del horario matutino.

- Horario de **tarde**. En esta franja de tiempo, comprendida entre las 6 y las 9 de la noche, el tráfico es considerablemente menor que en las anteriores categorías. En días laborales, la gente ha salido ya del trabajo y los colegios están cerrados. La gran mayoría de la gente se encuentra en sus casas y por ello el tráfico es bastante bajo, probablemente el más bajo del día, superado únicamente por el horario de noche.

- Horario de **noche**, que abarca desde las 9 en adelante, hasta las 6 de la mañana. Es, sin duda, el periodo del día donde menos coches hay en la carretera y, por lo tanto, implica que el tiempo de viaje disminuye significativamente si viajamos en horario nocturno.

Cabía la posibilidad de que se introdujera otros rangos distintos, como por ejemplo diferenciar la noche y la madrugada, pero hemos terminado considerando dichas franjas horarias como las más oportunas, debido a que después de darle muchas vueltas, la

diferencia de tráfico, que es el factor más importante en este apartado, tampoco era tan variable entre las franjas actuales, siendo muy estable dentro de ellas.

Por otra parte, para añadir mayor precisión a las estimaciones de nuestro modelo, la siguiente variable que hemos añadido se encargará de evaluar **el clima** que predominará durante el trayecto de un viaje. Se ha hecho una distinción entre **despejado**, **lluvioso** y **tormentoso**. Esto fue algo que desde el principio tuvimos claro que debíamos de contemplar, ya que un mal clima supone una disminución de la velocidad de circulación, con el consecuente aumento del tráfico y posibilidad de accidentes en carretera, lo que conlleva un aumento en el tiempo final del trayecto. Normalmente, las mayores complicaciones aparecen en los días laborales y lluviosos, siendo incluso mayores que en un día de tormenta, donde las condiciones climáticas son peores pero mucha gente opta por no salir a la carretera, ya sea por recomendación pública o por prudencia propia. De esta forma, aunque las condiciones meteorológicas para los días lluviosos sean menos adversas, el tiempo del trayecto aumenta mucho más.

En un principio, también distinguimos entre despejado soleado y despejado nublado, pero esa distinción era absolutamente innecesaria, ya que la diferencia entre ambos casos es inexistente. Por ello, se juntaron en un solo valor de clase.

Es importante valorar que el vehículo en el que realicemos un cierto trayecto influirá en el tiempo total. De esta manera, la **calidad del transporte** es otro de los factores a tener en cuenta, sobre todo en trayectos largos. No es lo mismo viajar por una autopista en un coche de gama alta/media, que en un coche de gama baja o antiguo que no pueda superar los 80 km/h. Debido a sus insignificantes diferencias y para no complicar tampoco mucho la toma de datos, simplemente hemos diferenciado la calidad entre baja y buena. Siendo baja los vehículos antiguos y los actuales de gama baja, y el resto buena.

Tal y como se mencionó anteriormente, en un principio, además de la calidad del vehículo, también buscamos distinguir entre distintos **medios de transporte**, principalmente entre el coche y la guagua. Sin embargo, nos dimos cuenta de que añadir la guagua como vehículo en nuestro modelo complicaría mucho la toma de datos, por lo que acabamos descartando esa opción y optando por dejar el coche como único vehículo posible en nuestro modelo.

En cuanto al **día de la semana**, hemos distinguido dos posibles valores a la hora de recoger los datos: día laborable o fin de semana. En un principio pensamos considerar todos los días de la semana, pero las diferencias entre muchos días eran inapreciables.La variación del tráfico entre por ejemplo un martes y un miércoles era prácticamente inexistente, a no ser que hubiera algún cambio puntual en alguno de los dos días. En cambio, la variación entre los días de la semana que eran laborables y los días de la semana que no lo son, era bastante más notoria. Por eso, finalmente hemos tenido solo en cuenta si el día es laborable o es fin de semana. Es obvio que, durante los días laborales, la congestión del tráfico es mayor en la mayoría de los trayectos, sobretodo en los que tienen como destino grandes ciudades o polígonos industriales. A su vez, los trayectos a destinos menos laborales, como pueden ser lugares de playa, tienen menos congestión.

Por último, hemos incluido una variable que no siempre se sabe de antemano, que es el **tráfico**. Normalmente, cuando salimos de casa no sabemos cuál será el tráfico, pero hoy en día existen ciertas aplicaciones que nos pueden proporcionar datos acerca del tráfico, como podría ser *Google Traffic*; es por eso que hemos decidido incluir esta variable pensando en

una futura implementación en la que podríamos importar datos en tiempo real desde *Google Traffic* y usarlo para nuestro modelo. De esta manera, los resultados obtenidos serían muchos más exactos.

OBTENCIÓN DE LOS DATOS

Para obtener los datos, comenzamos a partir de nuestros **propios trayectos**. Esto es una fuente muy segura para recogerlos ya que se pueden saber cuánto duraron exactamente y en qué condiciones. Sin embargo, nuestro programa necesita muchas entradas para poder generar un árbol con sentido y que las estimaciones sean lo más ajustadas a la realidad posible. Es por ello que hemos aplicado para la generación de datos programas como **Google Maps**, donde hemos podido obtener datos en tiempo real para trayectos entre dos puntos en un instante determinado, con todos los datos acerca del tráfico, clima, tiempo de trayecto y carreteras. Realizar este trabajo ha sido algo verdaderamente tedioso, ya que para que nuestra base de datos fuera realmente útil, hemos tenido que calcular todos estos datos entre diferentes puntos de la isla, llegando a obtener más de 700 datos, todos ellos de forma manual, uno por uno.

Ejemplo de cómo hicimos la toma de datos

Lugar de Partida	Lugar Destino	Distancia	Tráfico	Momento	Clima	Medio de transporte	Calidad Transporte	Dia	Terreno	Tiempo
Tegueste	La Cuesta	11	Bajo	Matutino	Despejado	Coche	Baja	Laboral	Carretera	21
Tegueste	La Cuesta	11	Bajo	Mañana	Despejado	Coche	Baja	Finde	Carretera	18
Tegueste	La Cuesta	11	Alto	Media Jornada	Despejado	Coche	Buena	Laboral	Carretera	22
Tegueste	La Cuesta	11	Bajo	Tarde	Despejado	Coche	Buena	Finde	Carretera	16
Tegueste	La Cuesta	11	Bajo	Noche	Despejado	Bus	Baja	Laboral	Carretera	27
Tegueste	La Cuesta	11	Bajo	Matutino	Despejado	Bus	Baja	Finde	Carretera	28
Tegueste	La Cuesta	11	Bajo	Mañana	Despejado	Bus	Buena	Laboral	Carretera	27
Tegueste	La Cuesta	11	Bajo	Media Jornada	Despejado	Bus	Buena	Finde	Carretera	27
Tegueste	La Cuesta	11	Alto	Tarde	Lluvioso	Coche	Baja	Laboral	Carretera	26
Tegueste	La Cuesta	11	Bajo	Noche	Lluvioso	Coche	Baja	Finde	Carretera	20

El hecho de que este proceso sea tan tedioso es lo que nos ha condicionado el incluir ciertas funcionalidades en el modelo. Como mencionamos en el apartado de los datos descartados de esta sección, el poder distinguir entre medios de transporte diferentes (guagua y coche), o contemplar el lugar de origen y destino del viaje para mejorar la precisión, habría requerido que el número de entradas necesarias aumentase de forma exponencial. Evidentemente, utilizando este sistema de recogida de datos, el proceso habría sido excesivamente largo.

En futuras versiones de este modelo, podríamos tratar de utilizar una manera algo más "automática" de recoger datos, que nos permita llegar a las miles de entradas de forma sencilla. Así, añadir las funcionalidades descartadas para esta versión sería algo factible.

Para la recogida de datos hemos ido introduciendo en un excel compartido todos los datos. Para pasar estos datos a un formato con el cual Weka pudiese trabajar hemos tenido que exportar los datos del excel a .csv. Sin embargo, a pesar de que Weka es capaz de trabajar con este tipo de formato, suele producir algunos errores, con lo cual hemos creado una sencilla aplicación escrita en Java que nos traduce el fichero .csv a un formato .arff con el cual Weka ya es capaz de trabajar perfectamente.

Todos estos datos se podrían recoger mediante técnicas de localización, así como lo hace Google Maps. Si tuviéramos información constante sobre la localización de distintas personas a lo largo de la carretera, podríamos conocer, interpretar y estimar muchísimos datos que de otra manera resultan más difíciles de calcular, tal y como hemos podido experimentar por nuestra propia cuenta con este proyecto. Además, la precisión de los resultados y de los cálculos sería probablemente más cercana a la realidad. El inconveniente de este método de recogida de datos es que por ejemplo en lugares donde la penetración de smartphones no es tan grande, puede ocurrir que no hay suficiente información para proveer información más adecuada del tráfico.

Mediante esta técnica se podrían extraer coordenadas de localización, y, haciendo cálculos simples de distancia, se podría calcular cuánto se tarda en llegar de un punto a otro, la variación del tiempo de viaje para estimar cálculos como el tráfico y la calidad del vehículo, y otros diversos parámetros de utilidad.

3. WEKA

Weka (Waikato Environment for Knowledge Analysis, en español «entorno para análisis del conocimiento de la Universidad de Waikato») es una plataforma de software para el aprendizaje automático y la minería de datos escrito en Java y desarrollado en la Universidad de Waikato. Weka es software libre distribuido bajo la licencia GNU-GPL.

Para generar el modelo, hemos utilizado aprendizaje basado en árboles de decisión, un modelo predictivo que mapea observaciones sobre una entrada a conclusiones sobre el valor objetivo de la misma. Este algoritmo es muy usado en el mundo de la minería de datos. Tenemos un árbol de clasificación en el que cada hoja representa etiquetas de clase y cuyas ramas representan los valores de las variables que concluyen en el valor de dicha hoja.

En el caso de nuestro modelo, los valores de clase no son nominales, es decir, no existe un dominio finito de valores posibles que podrían tomar los valores objetivo de las entradas. El tiempo es una magnitud que podría medirse de forma nominal, a base de intervalos, pero desde luego no es lo recomendable para nuestro caso. Por tanto, necesitamos que nuestras etiquetas de clase adquieran valores continuos, es decir, numéricos. Los árboles de decisión donde la variables de destino puede tomar valores continuos se llaman **árboles de regresión**.

ALGORITMO REPTree

Dentro de los algoritmos de árbol que podemos usar en Weka, hemos decidido usar el algoritmo **REPTree**, que nos proporciona un árbol de regresión, dado que es el que más se acomoda a nuestro problema. Su funcionamiento se divide en dos fases:

1. Fase de aprendizaje. Se crea un conjunto de reglas que se ajusten a los datos utilizados para el aprendizaje. Esta fase es común a prácticamente todos los algoritmos de minería de datos para elaborar modelos predictivos, con evidentes factores exclusivos del algoritmo.
2. Fase de poda. Se poda el conjunto de reglas resultantes utilizando ejemplares que no participaron en el aprendizaje. Esta fase es opcional y Weka nos permite elegir si queremos o no llevarla a cabo, aunque lo cierto es que, en nuestro caso, utilizar o no utilizar poda apenas afectó al resultado final.

El algoritmo REPTree ejecuta de forma automática las dos fases y devuelve como resultado el árbol de regresión que se corresponde con el modelo. A partir del mismo, deberíamos ser capaces de clasificar nuevas entradas, con suerte de forma correcta.

A continuación, vamos a examinar algunos de los argumentos que tiene el algoritmo y el significado de los mismos:

- **Número mínimo de instancias:** establece el número mínimo de instancias que tendrá cada hoja del árbol generado.
- **Poda:** podemos establecer si el algoritmo realizará o no podas a nuestro árbol de regresión. Realizar poda implica obtener un árbol de regresión ligeramente más simple, pero en algunos casos también podría ser menos preciso.
- **Profundidad máxima del árbol:** En caso de que este parámetro esté a menos uno, el algoritmo no establecerá una profundidad máxima.

```
REPTree
============

distance < 47.05
|   distance < 9.65
|   |   traffic = Habitual
|   |   |   distance < 8.25
|   |   |   |   weather = Despejado
|   |   |   |   |   timeOfDay = Matutino
|   |   |   |   |   |   distance < 6.05 : 9.5 (3/0.89) [1/21.78]
|   |   |   |   |   |   distance >= 6.05 : 12 (3/2.67) [1/16]
|   |   |   |   |   timeOfDay = Manana : 11 (2/2.25) [1/2.25]
|   |   |   |   |   timeOfDay = MediaJornada : 9.5 (1/0) [1/1]
|   |   |   |   |   timeOfDay = Tarde : 10 (2/0) [0/0]
|   |   |   |   |   timeOfDay = Noche : 7.6 (3/2.89) [2/0.44]
|   |   |   |   weather = Lluvioso
|   |   |   |   |   timeOfDay = Matutino : 12.5 (0/0) [2/0.29]
|   |   |   |   |   timeOfDay = Manana : 13.4 (4/0.69) [1/0.56]
|   |   |   |   |   timeOfDay = MediaJornada : 13 (1/0) [2/1]
|   |   |   |   |   timeOfDay = Tarde : 13.6 (3/2) [2/14.5]
|   |   |   |   |   timeOfDay = Noche : 10.8 (2/4) [3/1]
|   |   |   |   weather = Nublado : 11.24 (18/8.92) [16/3.74]
|   |   |   |   weather = Tormentoso
|   |   |   |   |   timeOfDay = Matutino : 14.67 (2/0) [1/64]
|   |   |   |   |   timeOfDay = Manana : 13 (2/0) [2/8]
|   |   |   |   |   timeOfDay = MediaJornada : 12 (3/0) [0/0]
|   |   |   |   |   timeOfDay = Tarde : 12 (2/1) [2/1]
|   |   |   |   |   timeOfDay = Noche : 10.75 (3/0.22) [1/2.78]
|   |   |   distance >= 8.25 : 15.14 (16/7.73) [6/1.47]
```

Pequeña fracción de un árbol generado por validación cruzada y 10 pliegues

RESULTADOS OBTENIDOS

Los resultados obtenidos muestran que nuestro modelo es bastante bueno en líneas generales. Hemos obtenido un **coeficiente de correlación** que oscila entre 0.941 y 0.9568. Este coeficiente nos aporta información relativa a cómo de cerca están los valores estimados y los valores reales. La escala del coeficiente va desde -1, en el que hay perfecta correlación negativa, pasando por 0, donde no existe correlación alguna, hasta 1, donde existe perfecta correlación positiva. Al estar tan cercano a uno podemos deducir que el modelo es muy bueno.

Por otra parte, el **error absoluto medio** nos da una idea de la diferencia entre los valores estimados y el valor real, siendo en cierto modo una medida opuesta al coeficiente de correlación. Para nuestro modelo, Weka nos dio como salida una diferencia que varía entre 3.8 y 4.7 puntos, siendo estos minutos de tiempo; es un error bastante pequeño, sobre todo para trayectos de duración media. De la misma manera, el **error relativo absoluto** nos dá una medida similar a la del anterior coeficiente, pero aplicado a un porcentaje. Este

estadístico tomó valores entre el 23 y 29 por ciento a lo largo de nuestras diversas pruebas. Considerando que nuestro modelo trata de medir el tiempo de llegada de un lugar a otro, este tipo de variaciones afectan muy poco a los trayectos cortos, pero algo más a los trayectos largos. Esto probablemente se deba a que los datos que hemos recogido se concentran mayormente en la zona norte de la isla, donde este tipo de variaciones son muy comunes.

A la hora de ejecutar el algoritmo, hemos usado la **validación cruzada** para evaluar el modelo. Los mejores resultados los hemos obtenido ejecutando con entre 7 y 10 pliegues, concretamente con respecto al coeficiente de correlación. Por otra parte, hemos probado el algoritmo con y sin poda del árbol de regresión resultante y los resultados obtenidos han sido bastante similares, como ya mencionamos al inicio de esta sección. En general, la variación de los distintos parámetros del algoritmo apenas ha afectado a los resultados finales obtenidos. Las variaciones más altas las hemos obtenido modificando el número de pliegues a usar.

```
Size of the tree : 151

Time taken to build model: 0.06 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.9568
Mean absolute error                      3.924
Root mean squared error                  5.9659
Relative absolute error                 24.5746 %
Root relative squared error             29.0613 %
Total Number of Instances              747
```

Salida de los resultados obtenidos por el REPTree

4. REPARTO DEL TRABAJO

En primer lugar, la obtención de los datos fue repartida entre los 5 miembros, a razón de 150 entradas por persona. Para tener variedad en las entradas, intentamos repartirnos por zonas de la isla. Así, Sergio García se encargó de la zona noreste hasta metropolitana; Guillermo Esquivel de los trayectos en la zona metropolitana; Óscar Darias, de los desplazamientos en la zona norte de la isla; Eduardo de la Paz, los que unen Santa Cruz con la zona sur; y, por último, Joaquín Sanchíz completó con datos en zonas variadas.

En segundo lugar, creamos un repositorio en GitHub, donde almacenamos todos los recursos necesarios del proyecto, tanto los datos que había recogido cada miembro, así como el informe. Así, podíamos ir sincronizando el trabajo entre todos los miembros del grupo, teniendo siempre actualizada la información a la hora de trabajar. En el repositorio, también se almacenaron la aplicación creada en Java para convertir los archivos .csv a .arff, y la presentación final, expuesta en clase.

El siguiente reparto que nos hicimos, fue el trabajo previo a la presentación. Mientras Óscar Darias se encargaba de la aplicación en Java, los demás miembros se encargaron añadir los datos a Weka y sacar conclusiones de los resultados obtenidos en dicho programa.

Una vez finalizado el trabajo de recogida de datos y aplicación de los mismos al programa, comenzamos con los preparativos de la presentación. En este apartado, nos centramos en la realización de un guión bien detallado de lo que queríamos exponer, así como un conjunto de transparencias para poder apoyarnos en ellas durante la presentación. Sergio García se encargó de las diapositivas para el trabajo. Mientras tanto, Óscar Darias realizaba la introducción al mismo, y Joaquín Sanchíz, Guillermo Esquivel y Eduardo de la Paz, redactaron los apartados relativos a las variables utilizadas, obtención de los datos, resultados obtenidos, etc.

Con la presentación finalizada, nos pusimos a trabajar sobre el informe del proyecto. Debido a que desarrollamos un buen guión para la presentación, pudimos extraer cierta información del mismo, utilizándolo como una base sobre la que añadir más información. Por ello, nos ha servido de apoyo para desarrollar y ampliar en profundidad cada uno de los temas de los que teníamos allí indicados. El reparto de tareas a la hora de desarrollar el informe ha sido el siguiente:

- Óscar Darias: Introducción y Datos. Ligeras aportaciones a Weka.
- Guillermo Esquivel: Datos y Weka.
- Sergio García: Conclusiones e Introducción.
- Eduardo de la Paz: Reparto de Trabajo y Conclusiones.
- Joaquín Sanchíz: Bibliografía e imágenes.

5. CONCLUSIONES FINALES

Respecto a la motivación del proyecto, como dijimos al principio, conocer el tiempo que podemos tardar en ir de un lugar a otro es algo que tiene un gran valor, ya que nos permitirá controlar y optimizar nuestro tiempo disponible mejor. Hoy en día, la rapidez y la optimización del tiempo son aspectos fundamentales en muchas empresas para poder ser competitivas en el mercado. Imaginemos que somos una gran empresa de mercancías: el poder conocer el tiempo que tardaremos en entregar la mercancía en una serie de lugares y la ruta más rápida es algo que nos interesa muchísimo, ya que nos podría ayudar a maximizar el beneficio que podríamos obtener, además de poder optimizar otros numerosos factores como el consumo de combustible, por ejemplo.

En cuanto a los resultados arrojados por el programa, nos dicen que a pesar de no devolver unos valores perfectos, sí que son una buena referencia acerca del tiempo posible que vamos a tardar en llegar a un lugar. Esto hemos podido probarlo personalmente en diferentes ocasiones, sobretodo aquellos componentes del grupo que tenemos coche, ya que nos cronometramos más de una vez al realizar diferentes trayectorias.

Es importante señalar que es casi imposible que nuestro proyecto acierte siempre el **tiempo exacto** que tardaremos en el momento de salida. Durante el recorrido, pueden darse muchos motivos por los cuáles podemos retrasarnos (en la mayoría de casos) o adelantarnos. Por ejemplo, podría pasar que una vez vayamos por la mitad del trayecto se nos pinche una rueda o que pasemos por un tramo en el cual hubo un accidente. Es verdad que la probabilidad de que den uno de estos factores suele ser muy pequeña, pero siempre es importante saber que hay determinados factores casi imposibles de controlar.

Analizando los datos, para distancia cortas (mayoría de nuestros casos, en torno al 65% debido al tamaño de la isla y que nuestros trayectos suelen ser normalmente cortos) el programa se comporta realmente bien, dando unos resultados que, contrastados con nuestras experiencias, suelen ser bastante coherentes. Sin embargo, para distancias más grandes sí que existe un desfase un poco mayor, aunque tampoco nos llevan a la incoherencia. Esto probablemente sea debido a que tomamos bastante menos datos para trayectos de larga distancia y que en esos casos existe mayor probabilidad de que aparezcan factores que hacen que tardemos más o menos. En un trayecto de más de una hora, los factores que pueden afectar al tiempo total del desplazamiento aumentan considerablemente, y hacen que sea mucho más complejo obtener con unos resultados tan fiables como los obtenidos para trayectos cortos.

Para una mejora de nuestros resultados, se deberían incluir más entradas a nuestra fuente de datos con aquellas características en las que los resultados son menos coherentes y menos frecuentes, las cuáles podrían ser sobretodo para distancias muy grandes como dijimos anteriormente (En el caso de la isla sería de La Laguna a Los Gigantes por ejemplo). También nos planteamos introducir una mejora con Google Traffic gracias a la cual sabremos el tráfico exacto en cada punto entre los dos lugares en vez de tan solo el tráfico en el lugar de partida. Esto nos permitirá también, poder indicar desde el lugar de partida el tráfico aproximado que nos vamos a encontrar, lo que mejoraría la precisión de nuestra aplicación. Por último, si hubiésemos dispuesto de alguna herramienta que nos permitiese introducir datos de forma automática, podríamos contemplar también en lugar de salida y lugar de llegada, lo cual desde luego ayudaría bastante al modelo. Sin embargo, al

contemplar dicha variable, aumentaría exponencialmente la complejidad de la toma de datos, ya que no es lo mismo ir desde Santa Cruz a La Laguna, que de Santa Cruz a Taganana, aunque ambos trayectos tengan una distancia similar, un clima similar y se viaje en el mismo vehículo. El modelo debería aprender los trayectos entre cada par de lugares a base de muchas entradas por lo que deberíamos encontrar como mencionamos anteriormente una automatización para la obtención de los datos.

Otra posible mejora sería introducir diferentes funcionalidades nuevas a la aplicación, cómo puede ser que calcule el tiempo caminando o en taxi. Además de mejorar bastante el tema del tiempo que tardan los autobuses. Estas funciones son muy parecidas a las que se proporcionan en Google Maps por ejemplo. En el caso del taxi puede ser bastante parecido a las que ya hemos estimado en coche. Los trayectos de guaguas, incluyen horarios prefijados y diferentes paradas que pueden realizarse o no, por lo que para hacer un sistema fiable, deberíamos de poder contar con la colaboración de las líneas urbanas de autobuses a través de su API o algo por el estilo.

Además de lo comentado en el párrafo anterior, nuestro objetivo es ir desarrollando el proyecto para que funcione en dispositivos móviles. Es en estos dispositivos donde se podría aprovechar su máxima funcionalidad, ya que lo podríamos llevar en el coche, a modo de GPS, calculando de antemano antes de salir hacia nuestro destino, el tiempo estimado de trayecto. Otros objetivos secundarios que tenemos pensado para nuestra aplicación, son que se pueda usar a nivel internacional en diferentes idiomas y que mejore la velocidad de cálculo del tiempo estimado. De esta forma, el cálculo se realizaría de una manera más rápida y óptima. Para ello estudiaremos diferentes lenguajes de programación para comprobar y estudiar con cuál de ellos se computan mejor los datos de la aplicación y ofrecer un sistema eficaz.

En definitiva, respecto al primer prototipo del proyecto consideramos que el modelo es bastante fiable para trayectos entre zonas desde el Puerto de la Cruz o la Orotava, hasta Santa Cruz de Tenerife: la mayor parte de la zona media-norte de la isla. En estos lugares, podemos afirmar que la aplicación predice con un error muy bajo la duración de los trayectos. Hemos comprobado nosotros mismos su funcionamiento y los resultados obtenidos han sido muy buenos para dichos desplazamientos. No obstante, como hemos comentado anteriormente, para zonas más alejadas o en las que pueden influir con más frecuencia otros factores, el programa suele fallar un poco más y hace que no sea tan fiable.

Trabajar en este proyecto ha supuesto una investigación más a fondo de muchísimos factores de la isla de Tenerife respecto a los desplazamientos con vehículos. Como hemos dicho anteriormente, hemos tenido que controlar muchos factores como: clima, tráfico, ..., etc. Esperamos seguir desarrollando nuestro proyecto, además de ir conociendo más los factores que afectan a los desplazamientos en la isla.

6. BIBLIOGRAFÍA

Como hemos comentado, gran cantidad de información que hemos utilizado para componer nuestra base de datos la hemos extraído desde Google Maps. Parte de la información mencionada en la memoria como la de Weka ha sido extraída desde Wikipedia y desde su página web oficial.

Wikipedia: www.wikipedia.com [Información acerca de Weka]

Weka: http://www.cs.waikato.ac.nz/ml/weka/ [Información acerca de Weka]

Google Maps: https://www.google.es/maps [Fuente de datos]