Trabajo Fin de Máster
Máster en Ingeniería Industrial

Modelos de elección modal mediante aprendizaje automático. Aplicación a los accesos a la Escuela Técnica Superior de Ingenieros de Sevilla

Autor: Miguel Cabeza Garrido
Tutor: Luis Miguel Romero Pérez

Dpto. Ingeniería y Ciencia de los Materiales y del Transporte
Escuela Técnica Superior de Ingeniería

Sevilla, 2022

Trabajo Fin de Máster
Máster en Ingeniería Industrial

Modelos de elección modal mediante aprendizaje automático. Aplicación a los accesos a la Escuela Técnica Superior de Ingenieros de Sevilla

Autor:
Miguel Cabeza Garrido

Tutor:
Luis Miguel Romero Pérez
Profesor Ayudante Doctor

Dpto. Ingeniería y Ciencia de los Materiales y del Transporte
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2022

Trabajo Fin de Máster: Modelos de elección modal mediante aprendizaje automático. Aplicación a los accesos a la Escuela Técnica Superior de Ingenieros de Sevilla

Autor: Miguel Cabeza Garrido

Tutor: Luis Miguel Romero Pérez

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2022

El Secretario del Tribunal

A mi familia por apoyarme durante toda la carrera.

A mi madre por seguir cada detalle de mi evolución y estar en todos los momentos difíciles para animarme a no abandonar.

A mi padre por ser mi ejemplo y mi inspiración para estudiar ingeniería.

A mi hermano por demostrarnos que nunca es tarde para cumplir los objetivos mediante el esfuerzo y la constancia.

Resumen

Actualmente existe una necesidad a nivel mundial por el desarrollo de modelos de transporte más sostenibles, lo que lleva a analizar los diferentes modos de transporte en las ciudades. Para ello, existen distintas herramientas entre las que se encuentra la elaboración de un modelo de reparto modal.
El presente trabajo se apoya en este modelo, el cuál (mediante un filtrado previo de datos) clasificará el número de usuarios en los medios de transporte analizados para la ciudad de Sevilla.

Asimismo, somos conscientes de que formamos parte de un entorno en el que, hasta ahora, basamos nuestras decisiones comerciales y personales en experiencias y estadísticas de pequeños conjuntos de información; nos hemos trasladado a un mundo en el que tenemos un gran número de ellos y son tan amplios que necesitamos la ayuda de algoritmos que nos ayuden a categorizar correctamente esta información.

Estos se denominan algoritmos de inteligencia artificial y aprendizaje automático. En el presente trabajo, se desarrollarán y compararán algunos algoritmos de predicción aplicados a la toma de decisiones a la hora de seleccionar un medio de transporte para trasladarse a la ETSI.

Índice

ÍNDICE DE FIGURAS

1 INTRODUCCIÓN

Este documento presenta la memoria del trabajo de fin de máster ‘‘Modelos de elección modal mediante aprendizaje automático. Aplicación a los accesos a la Escuela Técnica Superior de Ingenieros de Sevilla’’, consistente en el desarrollo, a través del aprendizaje supervisado, de un modelo que permita estimar el modo de transporte que empleará un usuario para desplazarse a la Escuela Técnica Superior de Ingeniería (en adelante ETSI) en función de una serie de características y variables.

El presente capítulo contextualizará el trabajo desarrollado, mostrando la motivación que ha servido de base para su desarrollo, los objetivos marcados y la estructura que sigue este documento.

1.1 Motivación y ámbito de estudio

Es una realidad que las distancias entre los individuos y sus lugares de destino ha ido acrecentándose de manera significativa con la sucesiva aparición de nuevos modos de transporte. Esto ha derivado en que el transporte necesario para desplazarse a ubicaciones cada vez más distantes se haya convertido en un proceso muy complejo, desarrollándose un sistema multimodal que ofrece diferentes alternativas, ya sea en el ámbito urbano o el interurbano.

El italiano Cesare Marchetti realizó un estudio en el que formuló lo que actualmente se conoce como “la constante de Marchetti”. Este físico planteó que el tiempo de viaje diario no sobrepasa la hora, esto se aplica en todo el mundo y a través de la historia, sin importar que la planificación urbana o la ubicación de los lugares de residencia y de trabajo varíen. Esto implica que, con la evolución de los medios de transporte, de media se viaja más distancia al realizar estos desplazamientos y, por lo tanto, es una buena explicación a la expansión urbana.

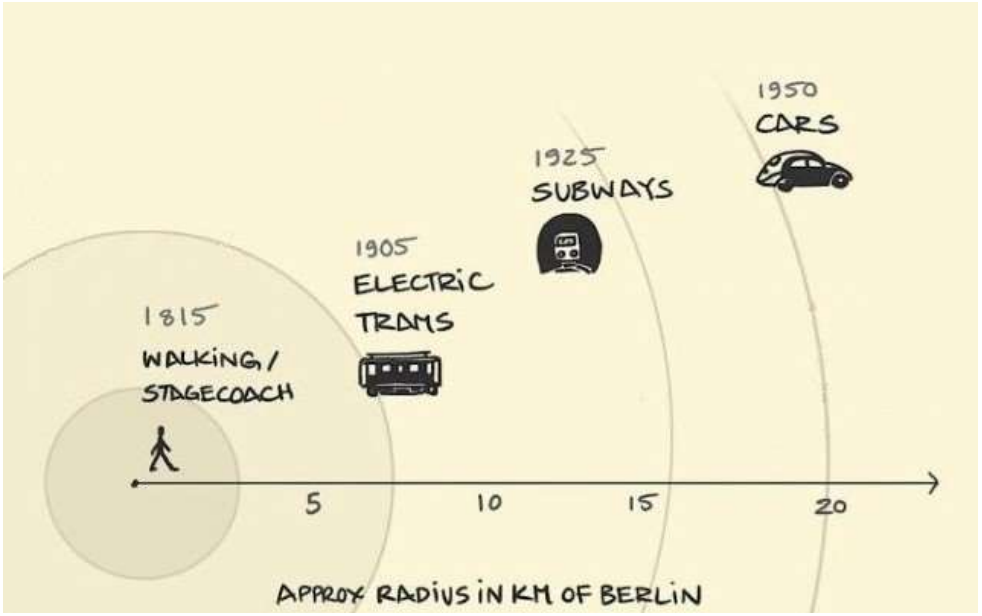


Figura 1-1: Evolución de la ciudad de Berlín.

Fuente: Anthropological Invariants in Travel Behavior, Technological Forecasting and Social Change [1]

La evolución de estos sistemas de transporte, tanto en términos de individuos como de mercancías, está estrechamente ligada a los cambios socioeconómicos de un país, originados a partir de la revolución industrial, que afectan a todos los niveles de accesibilidad territorial. Todo ello ha contribuido a que el transporte (público o privado) sea considerado algo de interés general, independientemente del motivo que lo genere (trabajo, ocio, estudios, entre otros).

En este orden de ideas, el presente documento pretende generar indicios del comportamiento de los usuarios de medios de transporte, en concreto para realizar el desplazamiento desde su domicilio hasta la ETSI, como una herramienta que contribuya al análisis y mejora de la situación actual.

Durante el desarrollo de este estudio se tendrán en cuenta distintas zonas de la ciudad diferenciándolas según su código postal. En la ilustración 1- 2 pueden verse las zonas que abarcará el análisis.

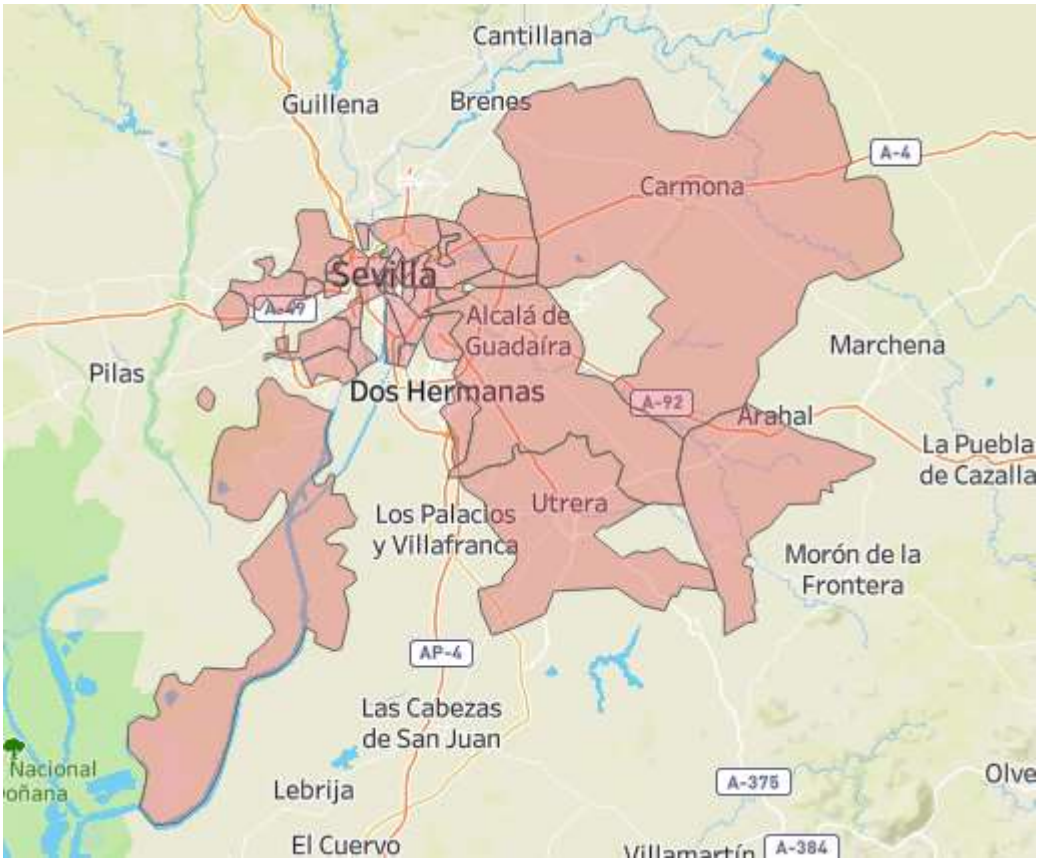


Figura 1-2: Municipio de Sevilla.

Fuente: elaboración propia. Software: Tableau.

Como se verá más adelante, para la generación de todas las variables dependientes de los modos de transporte se ha empleado la información recopilada mediante una encuesta dirigida a los profesores y estudiantes de la Escuela.

1.2 Objetivos

En el presente trabajo se persigue el desarrollo de un modelo de aprendizaje supervisado, el cual ha de estimar siendo lo más fiel posible a la realidad el modo de transporte que empleará un individuo en función de determinadas variables socioeconómicas y otras dependientes de cada tipo de transporte.

Se han considerado medios de transporte tanto privados como públicos. Las opciones seleccionadas han sido el autobús, el coche, la bicicleta y el desplazamiento a pie.

Para el desarrollo del modelo se empleará el entorno de cálculo Matlab. Previamente, se requerirán una serie de datos y variables de cada uno de los modos de transporte (principalmente tiempos y distancias), los cuales se generarán mediante Python.

La finalidad de este modelo es comparar diferentes modelos de reparto modal y así poder reproducir (considerando ciertas limitaciones) el proceso de elección del modo de transporte de un individuo, adaptándose a cambios en las variables de las que depende para poder estimar su respuesta ante cambios hipotéticos de sus características en escenarios futuros, optimizando así el proceso de implantación de un nuevo modo o la modificación de los ya existentes.

1.3 Estructura del trabajo

Con la intención de alcanzar los objetivos planteados en el apartado anterior (1.2) y resolver los diferentes problemas que se plantearán, el presente trabajo consta de la estructura que se muestra a continuación.

Durante el capítulo 1, se plantean los diferentes factores que influyen en la logística de los medios de transporte y de su relación con el aprendizaje supervisado, a la vez que se muestran los objetivos fundamentales que se pretenden lograr.

A lo largo del capítulo 2, se exponen los datos que se van a emplear y el método escogido para su recopilación. De igual manera, se muestran los métodos escogidos para realizar la depuración de esta información y los criterios empleados.

Durante los siguientes capítulos, concretamente del 3 al 5, se trata toda la metodología y teoría que envuelven el presente trabajo. En primer lugar, se expone el modelo clásico de elección y finalmente se desarrolla todo el marco teórico de modelos más actuales.

Una vez establecido lo que se pretende conseguir y las técnicas necesarias para ello, se procederá a la ejecución, de forma que en el capítulo 6 se muestran los resultados obtenidos, mostrando ejemplos y comentando lo más relevante.

Por último, en el capítulo 7 se establecerán las conclusiones obtenidas durante el desarrollo y ejecución del trabajo, así como futuras aplicaciones y posibles mejoras.

2 DATOS DE PARTIDA

En este apartado se expondrá el método para obtener la información necesaria y las herramientas empleadas para el tratamiento de la misma. Se analizarán los resultados obtenidos y, finalmente, se justificará el empleo de los diferentes softwares, así como el uso de ciertos aspectos clave a la hora de evaluar e interpretar el conjunto de datos.

2.1 Encuesta

Los alumnos de la asignatura de Planificación de Transporte lanzaron una encuesta de movilidad dirigida a los usuarios que acceden frecuentemente a la ETSI con el objetivo de estudiar el modo de acceso a la misma.

Para la elaboración de las preguntas y sus respectivas respuestas se tuvo en cuenta que era primordial la caracterización, por una parte, de los diferentes modos de acceso a la universidad y, por otro lado, la de los usuarios tipo según el tipo de transporte empleado como, por ejemplo, los usuarios de bicicletas en sus dos formatos, público y privado. Es necesario responder a la pregunta de cuál es el elemento diferenciador entre los usuarios para que opten por un modo frente a otro, que puede ser específico del usuario (vehículo propio no disponible o estacionado en origen y/o destino), o viaje (tiempo de viaje o costo, entre otros), etc. Los alumnos diseñaron el cuestionario permitiendo lograr el objetivo de caracterizar a los usuarios y los viajes, incluyendo el origen/destino y la duración/tiempo, de la forma menos intrusiva posible.

Originalmente se consideraron las siguientes variables para realizarla:

- Sexo
    - Hombre
    - Mujer
- Edad
- Rol en la Escuela:
    - Estudiante grado
    - Estudiante máster
    - PDI
    - PAS
    - Otros
- ¿Dispones de coche/moto?
    - Si
    - No
- Medio de transporte usual en tus desplazamientos a la Escuela
    - Coche solo
    - Coche compartido
    - Moto
    - Autobús

    - Bus lanzadera cartuja
    - Bicicleta Privada
    - SEVICI/Bicicleta consorcio
    - Cercanías
    - Bus interurbano
    - A pie
- Código postal/Barrio de origen habitual (si fuera de Sevilla, localidad)
- Calle y número de origen (opcional)
- Tiempo de desplazamiento en el trayecto:
    - Menos de 10 minutos
    - De 10 a 20
    - De 20 a 30
    - De 30 a 40
    - De 40 a 50
    - De 50 a 60
    - Más de 60 minutos
- Razón por la que se usa ese medio de transporte:
    - Distancia
    - Velocidad
    - Economía
    - Comodidad
    - Otro
- Tramo horario de desplazamiento:
    - Antes de las 12 pm
    - De 12 a 15 horas
    - De 15 a 18 horas
    - Después de las 18 horas
- Días de desplazamiento a la ETSI (sólo ida):
    - Lunes
    - Martes
    - Miércoles
    - Jueves
    - Viernes
    - Sábado
- Número de desplazamientos por día a la ETSI:
    - 1
    - 2

- 3
- 4
- 5

La recolección de las respuestas se hizo tanto de manera presencial como online. Este trabajo se ha realizado a partir del fichero generado mediante las respuestas telemáticas, las cuales conforman un total de 623.

2.2 Depuración y tratamiento de datos

Después de recabar información por diversos medios y métodos, se ha recopilado la base de datos principal que utilizaremos en el presente documento.

Como fase inicial de depuración, se ha realizado un tratamiento de las respuestas en bruto generadas de manera automática por el sistema de encuestas de Google. De esta forma, mediante el desarrollo de un código Python llamado "Depuración Inicial" (ya que el volumen da datos es bastante considerable y hacerlo de forma manual requeriría demasiado tiempo) se pasa de un formato tosco y complicado de leer (Ilustración 2-1) a uno más visual y que supone mayor comodidad a la hora de trabajar (Figura 2-2), en el que cada posible variable ocupa una celda propia.

	A	B	C	D	E	F	G	H	I	J	K
1	Marca temporal,"Edad","Sexo","Rol en la Escuela","Â¿Dispones de coche/moto propio?","Medio de transporte usual en tus desplazamie										
2	2017/12/05	MiÃ®rcoles	Jueves	Viernes","1"							
3	2017/12/05	Bus Interurbano","41009","Calle Talgo NÂº2","De 10 a 20 minutos","Distancia","De 12 a 15 horas","De Lunes a Viernes","1"									
4	2017/12/05	MiÃ®rcoles	Jueves	Viernes","1"							
5	2017/12/05	Martes	MiÃ®rcoles	Jueves	Viernes","1"						
6	2017/12/05 7:23:47 p. m. CET,"25","Hombre","Estudiante mÃjster","SÃ","Coche compartido","41927","","De 10 a 20 minutos","Comodid										
7	2017/12/05 7:27:52 p. m. CET,"26","Hombre","Otros trabajadores","SÃ","Coche solo","41927","","De 10 a 20 minutos","Comodidad","De										
8	2017/12/05 7:27:57 p. m. CET,"26","Hombre","Estudiante mÃjster","A veces","Bicicleta privada","41002","Cristo del Buen Fin nº4","Meno										
9	2017/12/05	Lunes	MiÃ®rcoles","2"								
10	2017/12/05	A pie","41002","","De 20 a 30 minutos","Velocidad","De 12 a 15 horas","De Lunes a Viernes","4"										
11	2017/12/05 7:33:16 p. m. CET,"25","Hombre","Estudiante mÃjster","A veces","Bicicleta privada","41001","","De 10 a 20 minutos","Comodi										
12	2017/12/05 7:51:18 p. m. CET,"25","Mujer","Estudiante mÃjster","No","Coche compartido","41013","","De 10 a 20 minutos","Velocidad","D										
13	2017/12/05 7:58:10 p. m. CET,"25","Mujer","Estudiante grado","A veces","AutobÃ's (C1, C2,...)","41003","","De 20 a 30 minutos","No teng										
14	2017/12/05	Martes	MiÃ®rcoles","1"								
15	2017/12/05 8:02:40 p. m. CET,"24","Mujer","Estudiante grado","No","AutobÃ's (C1, C2,...)","41013","","De 40 a 50 minutos","Distancia","A										
16	2017/12/05 8:02:50 p. m. CET,"24","Mujer","Estudiante grado","No","AutobÃ's (C1, C2,...)","41013","","De 40 a 50 minutos","Distancia","A										
17	2017/12/05 8:04:31 p. m. CET,"25","Hombre","Estudiante mÃjster","No","A pie","41009","","De 20 a 30 minutos","Comodidad","De 12 a 15										

Figura 2-1. Fichero Excel Generado por Google Encuestas en Bruto.

Fuente: Google Encuestas. Software: Microsoft Excel.

Adicionalmente, se han eliminado de forma manual aquellas respuestas que carecían de sentido lógico y que podrían desvirtuar su posterior análisis , como las repetidas, las que indicaban un código postal o una calle inexistente o las que carecían de sentido lógico, por ejemplo, un individuo que indicaba un código postal de Vitoria. A continuación, se ha decidido contar sólo con aquellas que decidieron indicar su dirección de residencia ya que gracias a esa información y mediante el programa Python, obtendremos las latitudes y longitudes de origen, para así poder georreferenciar tanto el origen como el destino de los viajes y poder inferir tiempos y distancias de viaje en las alternativas no escogidas por el individuo.

	A	B	C	D	E	F	G	H	I	J	K	L	
1						Coche solo	Coche compartido	Moto	Bicicleta privada	SEVICI/Bicicleta consorcio	Bus Interurbano	autobus (C1, C2,...)	Bus Lanz Cart
38	2017/12/05 11:11:53 p. m. CET	27	Hombre	estudiante master	No								
39	2017/12/05 11:29:27 p. m. CET	24	Hombre	Estudiante grado	No							autobus (C1, C2,...)	
40	2017/12/05 11:34:05 p. m. CET	25	Mujer	estudiante master	No							autobus (C1, C2,...)	
41	2017/12/05 11:36:45 p. m. CET	23	Hombre	estudiante master	si	Coche solo							
42	2017/12/05 11:38:00 p. m. CET	25	Hombre	estudiante master	si	Coche solo	Coche compartido					autobus (C1, C2,...)	
43	2017/12/05 11:38:57 p. m. CET	25	Hombre	estudiante master	si					SEVICI/Bicicleta consorcio			
44	2017/12/05 11:39:01 p. m. CET	25	Mujer	estudiante master	si	Coche solo							

Figura 2-2. Fichero Excel Generado por Google Encuestas Depurado.

Fuente: Google Encuestas. Software: Microsoft Excel.

2.2.1 Obtención de coordenadas mediante Python y API de Google Maps

En primer lugar, el objetivo de esta parte del proyecto es generar latitudes y longitudes a partir de una serie de direcciones (calle, número, código postal, etc.).

Para poder obtenerlos se ha recurrido al API de Google Maps ya que dispone de un servicio llamado Geocoding con el que usando como dato de entrada una dirección, se puede diferentes salidas como la latitud y longitud exactas de la misma. Ha sido necesario seguir los siguientes pasos:

- Registrarse en Google Cloud para obtener un API KEY ya que la API es de pago y genera un cobro por cada petición que se realice por lo que se necesita esta clave para identificarse cuando realizas peticiones, en nuestro caso una petición sería lo equivalente a darle una dirección (las primeras 40000 peticiones son gratuitas).
- En el entorno de desarrollo Pycharm es necesario instalar el módulo "googlemaps" ya que es la librería que trabaja con el API de Google.
- Diseñar el código de programación (este modelo recibe el nombre de "Generar latitudes y longitudes a partir de direcciones") que nos permitirá reflejar en un fichero Excel, que llamaremos "final_calle_lat_lon.xlsx", los resultados obtenidos.

address	LAT	LON	LAT,LON
Calle Talgo numero 2 CP 41009	37.4060326	-5.9937767	37.4060326,-5.9937767
Calle Estrella Canopus numero 1 CP 41015	37.4224524	-5.9716094	37.4224524,-5.9716094
Calle cristo del buen fin numero 4 CP 41002	37.3981752	-5.9991335	37.3981752,-5.9991335
Avenida Italia numero 17 CP 41012	37.35051	-5.982	37.35051,-5.982
Calle cueva de menga numero 3 CP 41020	37.3993565	-5.9437029	37.3993565,-5.9437029
Calle baltasar gracian numero 1 CP 41007	37.39052	-5.97147	37.39052,-5.97147
calle hermenegildo casas jimenez numero 1 CP 41020	37.3931767	-5.9230634	37.3931767,-5.9230634
Calle dolores ibarruri numero 8 CP 41806	37.364957	-6.1545777	37.364957,-6.1545777

Figura 2-3. Ejemplo de coordenadas generadas.

Fuente: elaboración propia.

2.2.2 Obtención de distancia y tiempos de transporte mediante Python y API de Google Maps

Una vez obtenidas las coordenadas el siguiente paso es el de obtener los tiempos y distancias medias de desplazamiento de los diferentes individuos que realizaron la encuesta hasta la universidad.

Para ello, se ha programado en Python un nuevo código que recibe el título de "Distancia y Tiempo Code" que nos permite obtener estos datos para tres tipos de desplazamiento: andando, vehículo particular (coche o moto) y bicicleta. En este caso, se ha empleado la misma API de Google que para la obtención de las coordenadas y el funcionamiento es similar, la diferencia radica en que los datos de entrada serán las coordenadas de la residencia de los encuestados y de la Escuela y como salida se generarán el tiempo y distancia empleado para realizar el trayecto.

Cabe destacar que los datos referentes al autobús se obtuvieron a partir de la modelización del sistema de transporte público de la ciudad de Sevilla, realizada con el programa TransCAD y facilitad por el tutor del presente trabajo.

KM_bicy	TIME_bicy_min	KM_driving	TIME_driving_min	KM_walking	TIME_walking_min	KM_bus	TIME_bus_min
2.8	8	4.3	7	2	25	2.5	13
3.8	13	5.3	10	3.8	47	6.1	38
2.5	8	2.8	9	2.4	30	2.5	19
8.8	27	12.6	17	8	100	9.2	45
9.3	30	9.3	13	7.4	94	13.2	62
6	19	8.2	18	4.9	61	5.7	26
11	35	12.5	18	9.5	119	10.7	67
18.9	58	20.3	18	18.4	224	22	90
2.5	10	4	7	2.5	31	3	25
8.8	29	9.3	15	7.5	94	11	59

Figura 2-4. Ejemplo de tiempos y distancias generadas.

Fuente: elaboración propia.

3 MODELO CLÁSICO DE ELECCIÓN Y REPARTO MODAL

El presente capítulo está basado y expone la teoría desarrollada por los profesores Luis Miguel Romero Pérez y Jose María del Castillo, concretamente en el tema 5 "Modelos de elección y reparto modal" de la asignatura "Compl. De Transporte y Serv. Urbanos".

Los modelos de transporte constituyen una herramienta muy beneficiosa durante la toma de decisiones durante el proceso de planificación de transporte. Estos, se emplean en la definición y en el diseño de los distintos criterios de transporte, de forma que cuantifican los resultados y valoran las posibles alternativas. Como añadido, también aportan valor estableciendo diseños adecuados para algunos parámetros del sistema con el objetivo de obtener una eficiencia óptima atendiendo a algún criterio a definir.

Existen unos pasos durante el análisis de un sistema de transporte que ayudan a identificar los elementos que lo componen y como se relacionan entre ellos. Estos son:

- Identificación de las dimensiones espaciales relevantes:
    - Definición del área de estudio
    - Zonificación
    - Modelización de la red de transporte
- Identificación de las dimensiones temporales relevantes:
    - Definición del horizonte de estudio
    - Hipótesis y metodologías para la estimación de la evolución de las variables implicadas
    - Estudio de la variabilidad de los parámetros del sistema en el período de estudio
- Identificación de las componentes relevantes de la demanda de viajes

A pesar de que cada vez es más común en el modelo tradicional elementos adicionales, estos modelos presentan una estructura clásica conocida como "Modelo de Cuatro Etapas", como se puede ver en la figura adjunta a continuación.

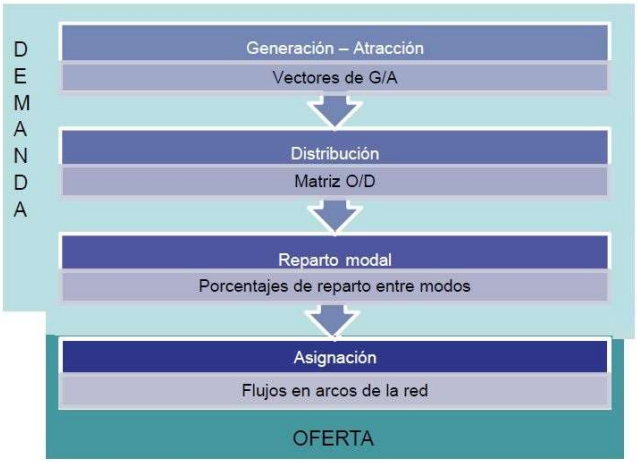


Figura 3-1. Estructura del Modelo Clásico de Cuatro Etapas

Fuente: Tema 5, Modelos de Elección y Reparto Modal, Apuntes de Compl. De Transporte y Serv. Urbanos.

3.1 Teoría de los modelos de reparto modal

Cuando se habla de reparto modal, este se refiere a la cantidad de viajes realizadas en un modo de transporte o, por otra parte, al porcentaje de viajeros que usan ese medio para desplazarse.

Se puede hablar de forma genérica de un agente que se enfrenta un problema de seleccionar la mejor alternativa entre un conjunto discreto de opciones. Esta decisión la toma de forma racional, con información completa y perfecta. De esta manera, trata de alcanzar objetivos muy específicos y predeterminados en la mayor medida posible con el menor coste posible. [2]

Para estos modelos, existe un conjunto de alternativas disponibles que deberán reunir las siguientes características para su empleo:

- Mutuamente excluyentes: escoger una opción implica que se han desechado las demás. Si estamos ante un caso real que no sea excluyente (caso A y B) podrían escogerse solo una de las alternativas, pero también ambas como si formasen un conjunto.
- Exhaustivo: se han de incluir todas las posibilidades en el conjunto de elección.
- El conjunto de elección debe ser finito.

Cada individuo q asocia a cada alternativa j una utilidad percibida U_{jq}. Dicha utilidad es asignada teniendo en cuenta una serie de características tanto suyas propias como de la alternativa j. Se puede resumir en que el individuo persigue maximizar su utilidad, seleccionando aquella alternativa que más beneficios o utilidad le reporte. El modelo de conducta supuesto implica que el individuo q escoge la alternativa i si y solo si $U_{iq} > U_{jq}$ para cualquier $j \neq i$.

Sin embargo, el modelador, que es un observador del sistema y que no cuenta con toda la información que lleva a cada individuo al realizar sus elecciones, tiene que procurar cuantificar dicha utilidad en términos objetivos.
Por ello considera que la utilidad de la alternativa j para un individuo q determinado, U_{jq}, está formada por dos componentes, atendiendo a la siguiente expresión:

$$U_{jq} = V_{jq} + \varepsilon_{jq}$$

Siendo:

- V_{jq} la parte medible por el observador y por tanto determinista, que se expresa en función de una serie de atributos.
- ε_{jq} una componente aleatoria que representa la idiosincrasia y/o gustos o preferencias de cada

individuo, incluyendo también los errores de medición y observación que pueda cometer el modelador.

No se conocen los errores, pero sí se puede suponer que responden a una función de distribución de probabilidad determinada y estimar probabilidades.

Los Modelos de elección discreta son modelos desagregados. Cada individuo representa una observación (un punto en las gráficas). La aplicación del modelo sí puede dar lugar a resultados agregados. Los Modelos desagregados están menos expuestos a distorsiones debidas a correlaciones entre unidades agregadas. Cuando se agrega información puede ocultarse el comportamiento individual por características no identificadas asociadas a las zonas. Este problema es conocido como falacia ecológica.

3.1.1 Modelos de elección aplicados al reparto modal

Como se ha mencionado, el objetivo principal de estos modelos es mostrar la probabilidad de elegir una alternativa en función de una serie de parámetros o variables relativas a los usuarios y, en nuestro caso, al modo de transporte.

En principio la forma concreta en la cual la utilidad depende de los atributos deber ser también asumida como hipótesis. La forma funcional más común en los modelos de elección es aquella que asume que el nivel de utilidad varía linealmente con los rasgos, entonces:

$$V_i = \sum_{k=1}^{m_i} \theta_{ik} X_{ik}$$

Donde

- m_i= número de atributos o variables de la alternativa i
- θ_{ik} = coeficiente del atributo k de la alternativa i
- X_{ik} = variable k de la alternativa i

3.1.2 Modelo Logit

Este modelo se obtiene suponiendo que el componente ε_{jq} de la expresión de la utilidad, se distribuye independientemente y de forma idénticamente distribuida según una densidad de probabilidad de tipo valor extremo, conocida como Gumbel. De esta forma, se pretende obtener un modelo capaz de explicar el efecto de los cambios de las variables independientes sobre la probabilidad de que *i* valga 1 (es decir, sobre la probabilidad de éxito).

Asumiendo las hipótesis del modelo Logit, se obtiene una expresión para la probabilidad de elegir la alternativa *i*:

$$p_i = \frac{\exp(V_i)}{\sum_{k=1}^{n} \exp(V_k)}$$

Siendo V_i la utilidad de la alternativa *i* que muestra lo útil o beneficioso que es para un individuo esa alternativa. En la especificación del modelo, la utilidad depende de las variables cuantificables y de los coeficientes aún por inferir.

En el contexto de transporte, las alternativas se refieren a los modos de transporte que se le ofrecen al usuario potencial. Estos conjuntos de alternativas dependerán de la situación en consideración, para el caso del presente trabajo existe el siguiente conjunto de alternativas:

- Coche
- Autobús
- Bici/Andando

De esta manera, partiendo de todo lo desarrollado, se establecen las siguientes expresiones de utilidad para la parte determinista del problema que nos ocupa:

$$V^{q}_{Coche} = \theta_{Coche} + \theta_{CS} \cdot S^{q} + \theta_{CR} \cdot R^{q} + \theta_{D} \cdot D^{q} + \theta_{t} \cdot (t^{q}_{Coche} - t^{q}_{Bus})$$

$$V^{q}_{Andando} = \theta_{Andando} + \theta_{AS} \cdot S^{q} + \theta_{AR} \cdot R^{q} + \theta_{t} \cdot (t^{q}_{Andando} - t^{q}_{bus})$$

$$V^{q}_{Bus} = 0$$

Donde:

- q: se refiere al individuo
- S^q: sexo del individuo
- R^q: rol del individuo q
- D^q: disponibilidad de coche (solo aparece para este medio de transporte)
- θ_t: coeficiente del tiempo (es el mismo para todas las alternativas)

4 APRENDIZAJE AUTOMÁTICO Y ALGORITMOS

El concepto de aprendizaje automático no es algo nuevo, aunque se ha convertido en uno de los temas más actuales tanto en el campo de la investigación como en el ámbito profesional debido al gran desarrollo y difusión que ha experimentado de la mano de grandes empresas tecnológicas como Google, Microsoft o Facebook. En la actualidad, se cuenta con una gran cantidad de conjuntos de datos. Hace relativamente pocos años, ni siquiera se tenía la capacidad para almacenarlos, y mucho menos analizarlos. El costo de almacenamiento y cómputo eran demasiado altos, pero hoy, estos datos pueden ser automáticamente almacenados y analizados debido a la gran capacidad de cómputo y del almacenamiento a un precio reducido que existe hoy en día. Por todo ello, el aprendizaje automático es una tecnología en auge.

Para empresas este tipo de empresas los costos de infraestructura informática y almacenamiento son insignificantes si se comparan con los beneficios obtenidos como resultado del análisis de estos datos almacenados. De hecho, cabe destacar que en ocasiones existe información que se almacena por la utilidad y ventaja que puede aportar en el futuro ya que en el instante de su almacenamiento carece de valor.

Por otra parte, nos encontramos con que la mayoría de las pymes españolas, aunque dispongan de una gran cantidad de información y de datos valiosos, aún no apuestan por el Machine Learning debido al desconocimiento que existe sobre este tipo de herramientas. Según estudios, los sectores más potencian sus negocios gracias al empleo de esta tecnología son los de la banca y los seguros, permitiéndoles implementar proyectos muy ambiciosos. [4]

Durante la última década, la técnica que ha tenido gran relevancia es la de redes neuronales, su importancia radica en que son capaces de aprender de forma jerarquizada, es decir, por niveles. De esta manera, en las primeras capas aprenden conceptos concretos (tornillo, rueda, etc.) y en las posteriores se usa la información asimilada anteriormente para aprender conceptos más abstractos (coche, camión) por lo que, con cada capa añadida, la información aprendida es más extensa. La ventaja es que el número de capas posible es ilimitado por lo que se pueden obtener algoritmos cada vez más complejos. Este tipo de conocimiento interno de los datos es una nueva tecnología operativa conocida como Deep Learning. La diferencia entre esta y el machine Learning es que tiene mayor flexibilidad gracias a la existencia de grupos neuronales más especializados para resolver problemas específicos, por lo que estos grupos de neuronas pueden resolver problemas más complejos. En términos de seguridad de redes, ambos métodos se aplican a la detección y clasificación de correos no deseados, malware, bots, fraude con tarjetas de crédito, ciber terrorismo en redes sociales, reconocimiento de voz y lenguaje natural, emociones, etc. [5]

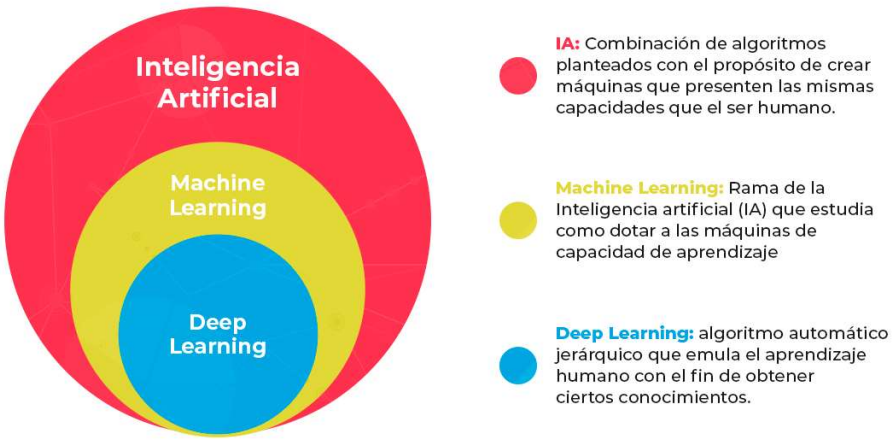


Figura 4-1. ¿Qué es el Machine Learning?

Fuente:://www.masterdatascienceucm.com/que-es-machine-learning/

Cuando hablamos de Machine Learning ya no nos referimos al futuro, es algo latente en el presente, que nos ayuda a enfrentarnos a un gran número de problemas apoyándose en cada avance tecnológico y en otras tecnologías como big data, IoT, etc. Se puede aplicar a un amplio abanico de entornos entre los que destaca, desde la ciberseguridad hasta algo más cotidiano, como la música o películas que te recomienda tu teléfono a raíz del patrón que haya observado en nuestro consumo

4.1 Motivos para el empleo del Aprendizaje Automático

Las razones para usar el Machine Learning a todos los niveles son muy variadas. A continuación, se enumeran las más destacadas:

- La cantidad de datos que podemos administrar hoy crece día a día y es tan grande que ya no podemos procesarlos con herramientas convencionales.
- Los costos de almacenamiento y computación se han reducido drásticamente, lo que ha dado como resultado un aumento significativo en la capacidad de almacenamiento y la potencia informática tanto en el hogar como a escala empresarial.
- Los beneficios obtenidos del análisis de un gran volumen de datos. Esto se observa en que las empresas de hoy en día ya no toman decisiones comerciales basadas únicamente en la experiencia, sino que van un paso más allá y se están convirtiendo en empresas Data Driven, es decir, basadas en datos que toman decisiones comerciales fundamentadas en este análisis. Por este motivo, los algoritmos analizados en el presente TFM son un buen ejemplo para este tipo de tareas. [6]
- Las soluciones de inteligencia artificial se pueden aplicar e implementar de forma relativamente sencilla, desarrollándolas en múltiples dispositivos y máquinas. De esta forma, podríamos emplear un algoritmo de clasificación o una red neuronal correctamente entrenada y validada para realizar diversas tareas.

Gracias al aprendizaje automático, hoy podemos analizar datos a gran escala con varios algoritmos, permitiéndonos reconocer patrones en el menor tiempo posible, identificar oportunidades rentables y evitar riesgos que serían imperceptibles basándonos solo en la experiencia. La gran potencia de esta es la posibilidad de que las máquinas aprendan por sí solas y puedan entender, e incluso anticipar, el comportamiento de los usuarios. Por lo tanto, los campos de aplicación son muy amplios: medicina, banca, logística, etc.

4.2 Tipologías Principales en Aprendizaje Automático

El aprendizaje automático es una rama de la inteligencia artificial que permite que los sistemas aprendan automáticamente y mejoren en función de la experiencia. Estos sistemas convierten datos en información útil para la toma de decisiones. Para que un modelo haga predicciones sólidas, debe alimentarse con datos; cuanto mayor sea el volumen de estos, mejor funcionará el algoritmo. Afortunadamente, Internet hoy en día está lleno de recursos de datos. En muchos casos, estos son recopilados por empresas privadas para su propio beneficio, pero existen otras iniciativas como los portales de datos abiertos.

Una vez que tenemos los datos, podemos iniciar el proceso de aprendizaje. Este proceso, llevado a cabo por un algoritmo, intenta analizar y explorarlos en busca de patrones ocultos. El resultado de este aprendizaje a veces no es más que una función que actúa sobre ellos para calcular una predicción particular.

A continuación, se muestran las principales tipologías que abarca el campo del Machine Learning:

- Aprendizaje supervisado:

  Este tipo de aprendizaje consiste en la construcción de unos modelos que son entrenados a partir de

ejemplos con unas características determinadas y que se asocian a una clase concreta, es decir, son modelos que predicen el resultado de salida en base a ejemplos históricos de esa misma variable de salida.

Este es el tipo de aprendizaje que nos aplica ya que necesita un conjunto de datos etiquetados, lo que se traduce en que hay que indicarle al modelo qué es lo que queremos que aprenda. En nuestro problema se han estado registrando datos referentes a los desplazamientos de diferentes individuos, distancia, tiempo, día de la semana, etc., también se ha hecho lo propio con el número y tipo de modo de transporte empleados cada día. En este caso, interesa entrenar un modelo que, a partir de los datos (características del modelo) de un día específico, muestre qué modo es el más adecuado (la etiqueta a predecir) para realizar el desplazamiento.

El aprendizaje supervisado consta una serie de pasos esenciales:

- Seleccionar el tipo de datos de entrenamiento: el primer paso en el aprendizaje supervisado es determinar cuál es la naturaleza de los datos que se utilizarán para el entrenamiento (datos train). Por ejemplo, en el caso del análisis de escritura a mano, esto podría ser una sola letra, una palabra o una oración. A los datos que no han sido seleccionados para el entrenamiento se le denomina “datos test”.
- Recopilar y limpiar los datos de entrenamiento: en este paso, los datos de entrenamiento se recopilan de varias fuentes y se homogeneizan sometiéndose a una rigurosa depuración de los datos.
- Elegir el algoritmo de aprendizaje supervisado: según la naturaleza de los datos de entrada y el uso deseado, se debe utilizar un algoritmo de clasificación o uno de regresión. Como ejemplo de estos algoritmos se pueden citar los árboles de decisión, SVM, Naïve Bayes o bosques aleatorios. La consideración principal al seleccionar un algoritmo es la velocidad de entrenamiento, el uso de la memoria, la precisión de la predicción de nuevos datos y la transparencia/interpretación del algoritmo.
- Entrenar el modelo: se emplean los datos train para crear el modelo.
- Realizar predicciones y evaluar el modelo: en esta fase se emplean los datos test para evaluar el modelo creado anteriormente. El motivo de que se realice de esta forma es que si entrenamos y evaluamos el modelo con los mismos datos hay una probabilidad muy alta de que obtengamosun modelo que haya sobreaprendido de los datos de la muestra, obtenendo un resultado engañosamente perfecto pero que no sería representativo de lo que ocurriría si se aplicara al resto de la población.
- Optimizar y volver a entrenar el modelo: la degradación de datos es una parte natural de Machine Learning. Por lo tanto, los modelos se deberán volver a entrenar periódicamente con datos actualizados para garantizar la precisión.

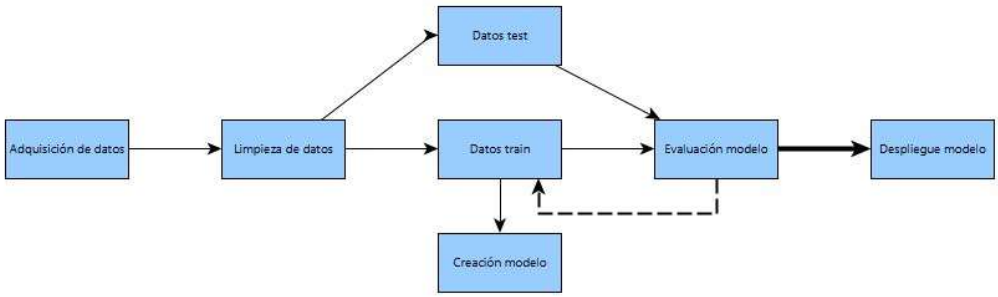


Figura 4-2. Etapas del Aprendizaje Supervisado.

Fuente: elaboración propia

Dependiendo del tipo de etiqueta, dentro del aprendizaje supervisado se puede dividir en:

- Clasificación: la salida del sistema debe ser asociada a una de entre un conjunto discreto de clases C_k con $k=1, 2, ..., e$. Es decir, son útiles cuando la respuesta a la pregunta sobre la empresa se aloja dentro de un conjunto finito de resultados posibles.

  A su vez, este tipo de modelos pueden ser binarios si tenemos que predecir entre dos clases o etiquetas (coche o no coche, clasificación de correos electrónicos como "spam" o no "spam") o multiclase, cuando se tiene que clasificar más de dos clases (clasificación de imágenes de plantas, análisis de sentimientos, etc.).

- Regresión**:** producen como salida un valor real, es decir, representa a los valores de una variable continua, como el ejemplo que anterior de los medicamentos.

Los algoritmos enfocados a la clasificación trabajan generalmente sobre la información entregada por un conjunto de muestras, patrones, ejemplos o prototipos de entrenamiento que son tomados como representantes de las clases, y los mismos conservan una etiqueta de clase correcta. A este conjunto de prototipos correctamente etiquetados se les llama conjunto de entrenamiento, y es el conocimiento

disponible para la clasificación de nuevas muestras. El objetivo de la clasificación supervisada es determinar, según lo que se tenga conocimiento, cual es la clase a la que debería concernir una nueva muestra, teniendo en cuenta la información que se pueda extraer.

- Aprendizaje no supervisado:

  Esta es una técnica que trabaja con datos que no han sido etiquetados y solo están disponibles los datos de entrada. Por lo tanto, la máquina debe poder encontrar la estructura de datos actual. Este tipo de aprendizaje es muy útil para reducir o simplificar el tamaño de los datos, minimizando la pérdida de información. Un buen ejemplo, sería el de aplicar este aprendizaje para segmentar los pacientes que han sido atendidos en urgencias en grupos homogéneos, pero sin un conocimiento previo de los grupos que queremos obtener ya que se haría a partir de estructuras no evidentes subyacentes en los datos.

- Aprendizaje semi-supervisado:

  El aprendizaje semi-supervisado se encuentra a medio camino entre el aprendizaje supervisado y el no supervisado. En este caso, se dispone tanto de datos etiquetados como de no etiquetados, es decir, además de tener tuplas (X, Y), tenemos datos sólo de X de los que no sabemos su respuesta Y.

  La dificultar reside en combinar datos etiquetados y no etiquetados para formar un modelo supervisado que sea mejor ya que:

  - La cantidad de datos etiquetados se puede aumentar, lo que generalmente mejora los resultados de los modelos.
  - Hay un alto coste de etiquetar los datos X sin etiquetas.
  - Supone que los datos etiquetados y no etiquetados provienen de la misma distribución. Por otro lado, puede existir un sesgo en la elección de datos no etiquetados.
  - Básicamente, entrenamos un algoritmo de aprendizaje supervisado empleando como etiquetas las etiquetadas manualmente más las generadas por los modelos anteriores.

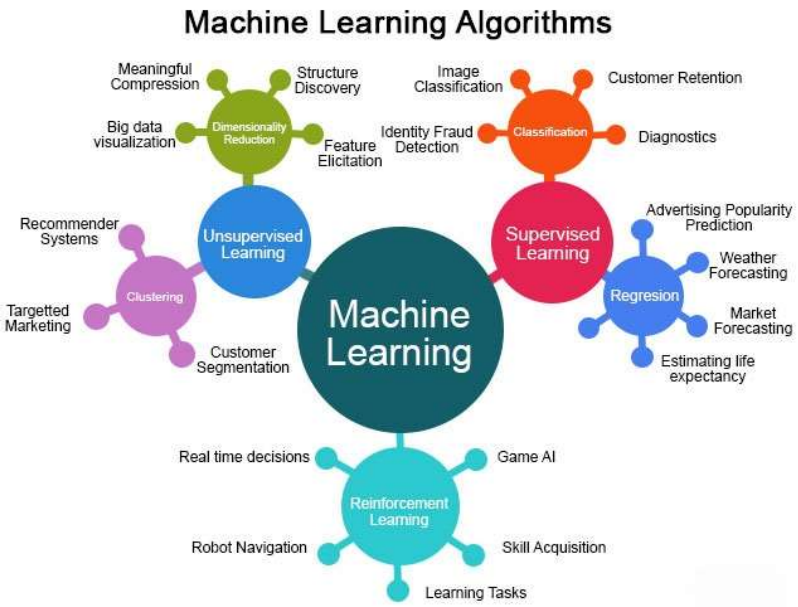


Figura 4-3. Machine Learning Algorithms.

Fuente: www.educba.com

- Aprendizaje por refuerzo:

  Este último es un método de aprendizaje automático basado en recompensar los comportamientos deseados y penalizar los no deseados. Los datos de entrada se obtienen a través del feedback o retroalimentación del entorno. Si la máquina no lo hace bien se le da un premio igual a 0 o negativo. En cambio, si toma una acción acertada se le dan premios con valor positivo para que finalmente encuentre una buena solución.

  AlphaGo o Pacman son algunos juegos donde se emplea esta técnica. En estos casos, el agente recibe información sobre las reglas del juego y aprende a jugar por sí mismo. Al principio, lógicamente, se comporta de manera aleatoria, pero con el tiempo empieza a aprender movimientos más sofisticados. Este tipo de aprendizaje se aplica también en otras áreas como la robótica, la optimización de recursos o sistemas de control.

5 APRENDIZAJE SUPERVISADO

Como se ha comentado, el aprendizaje supervisado se refiere al subconjunto de Aprendizaje Automático donde se generan modelos para predecir el resultado de salida en base a ejemplos históricos de esa variable de salida. Los modelos se construyen a partir de los algoritmos de Aprendizaje Automático y características o atributos de los datos de entrenamiento para que se pueda predecir el valor utilizando otros valores obtenidos a partir de datos de entrada.

Los algoritmos de aprendizaje supervisado intentan modelar relaciones y dependencias entre el resultado predictivo objetivo y las características de entrada para que podamos predecir los valores de salida para datos nuevos en función de las relaciones que aprendió de los conjuntos de datos anteriores.

El conjunto de este capítulo está basado en libros y artículos los cuales están referenciados a lo largo del mismo.

A continuación, se presentarán los principales tipos de algoritmos supervisados de aprendizaje automático, desarrollando sus características principales y enfrentando sus ventajas e inconvenientes [7]. Los diferentes algoritmos a tratar son:

- Árbol de decisión
- Métodos de conjunto o ensemble:
  - Decision Tree Bagging
  - Random Forest
  - Decision Tree Boosting
- Clasificación de Naïve Bayes
- Support Vector Machines (SVM)
- K vecinos más cercanos

5.1 Árbol de decisión

Los métodos basados en árboles son considerados una de las mejores y más empleadas opciones de aprendizaje supervisado. Este tipo de algoritmos de aprendizaje impulsan modelos predictivos con una gran estabilidad, precisión y sencillez de interpretación.

A diferencia de los modelos lineales, estos árboles mapean bastante bien las relaciones no lineales. Son adecuados para resolver cualquier tipo problemas, independientemente de si son de regresión o clasificación.

En los problemas de clasificación [8] (como es el caso que nos ocupa) se pueden distinguir dos pasos, el aprendizaje y el paso de predicción. En el primero, el modelo se desarrolla en base a datos de capacitación existentes mientras que, en el paso de predicción, el modelo se emplea para predecir la respuesta de esos datos previamente dados.

Examinando este tipo de algoritmo desde una perspectiva de machine learning se puede afirmar que es uno de los más sencillos y populares de entender e interpretar. Un árbol de decisión en aprendizaje automático consiste en una estructura similar a un diagrama de flujo que imita el pensamiento a nivel humano; en él existen unos nodos internos que representan unos atributos, unas ramas que implican unas reglas de decisión y unos nodos hojas que son los resultados. El nodo superior, del cual nace el árbol, es conocido como nodo raíz. Este divide el árbol en otros subgrupos y lo particiona en función del atributo.

Los árboles de decisión clasifican los ejemplos distribuyéndolos por el árbol desde la raíz hasta algún nodo

hoja, el cual proporciona la clasificación al ejemplo, esto se llama Enfoque de arriba hacia abajo o top-down. Cada nodo en él actúa como un caso de prueba para un atributo, y cada borde que desciende de ese nodo supone una de las posibles respuestas al caso de prueba. Este proceso es recursivo y se repite para cada subárbol enraizado en los nuevos nodos.

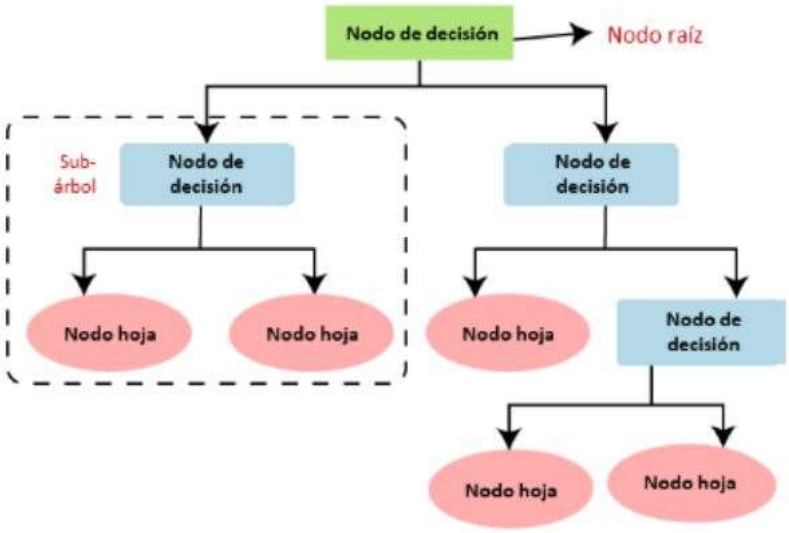


Figura 5-1. Estructura de un árbol de decisión.

Fuente: [9]*http://repositorio.espe.edu.ec/handle/21000/24174*

Con la intención de profundizar y a modo de resumen, se exponen las terminologías básicas en el tema a tratar:

- Nodo raíz (nodo de decisión superior): Representa a todo el conjunto de la población o muestra y este se divide en dos o más subconjuntos homogéneos.
- División: Es un proceso de segmentación de un nodo en dos o más subnodos.
- Nodo de decisión: Es un subnodo adicional fruto de la división de otro subnodo.
- Nodo de hoja / terminal: Son aquellos nodos sin hijos.
- Poda: Proceso que consiste en reducir el tamaño de los árboles de decisión eliminando nodos (opuesto a la división).
- Rama / Subárbol: Una subsección del árbol de decisión.
- Nodo padre e hijo: El nodo padre es aquel nodo que se divide en subnodos o nodos hijo; se denomina nodo principal de subnodos.

5.1.1 Funcionamiento del Árbol

La idea básica detrás de cualquier algoritmo de árbol de decisión es la siguiente:

Seleccionar el mejor atributo utilizando Medidas de selección de atributos (ASM) para dividir los registros.

Convertir ese atributo en un nodo de decisión que divida el conjunto de datos en subconjuntos más pequeños recursivamente para cada hijo hasta que coincida una de las condiciones:

- Todas las duplas pertenecen al mismo valor de atributo.
- No quedan más atributos.
- No hay más instancias.

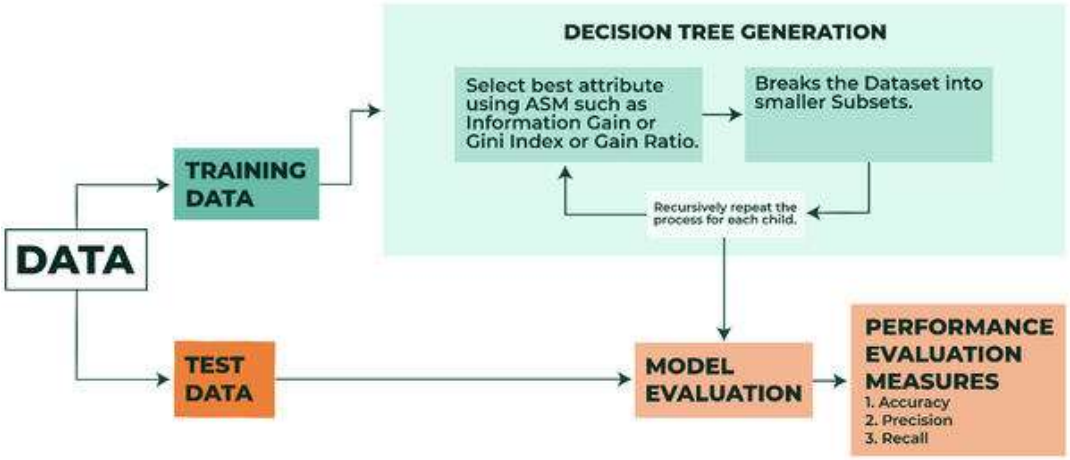


Figura 5-2. Algoritmo del Árbol de Decisión.

Fuente: https://www.godatadrive.com/blog/data-science-for-business-leaders-decision-trees

5.1.2 *Ventajas y Desventajas*

En este apartado se mostrarán los beneficios de emplear esta metodología para el aprendizaje automático:

- Los árboles de decisión se pueden usar para regresión o clasificación, aunque son más comunes para el segundo problema. En general, si desea utilizar árboles de decisión para modelos de regresión, se debe utilizar una metodología de conjunto.
- No son paramétricos, que es una forma de decir que no existen suposiciones sobre cómo se distribuirán los datos; se determina la estructura (parámetros) de nuestro modelo a partir de la entrada del usuario y el estudio de la muestra, es decir, se modela a través de los datos. Los modelos no paramétricos son excelentes cuando hay un gran volumen de datos.
- Son interpretables ya que, además de las predicciones, podemos interpretar el propio modelo después de construirlo.
- Por último, debido a su simpleza son rápidos y esto, aunque suponga renunciar a priorizar la velocidad en un modelo final se contrarresta a la hora de construir un modelo inicial que permita tener una comprensión inicial de los datos.

Es obvio que hay muchos aspectos positivos al usar árboles de decisión, pero en determinados casos puede que no sea la mejor elección. Esto ocurre si el tamaño de muestra existente es pequeño, y en el caso de la regresión, puede que no sea la mejor opción si se estima que se van a predecir valores finales que estarán fuera de los que contiene la muestra de entrenamiento. Además de estos puntos, los árboles de decisión, como todos los modelos, presentan las siguientes desventajas:

- Tienen tendencia al sobreajuste ya que pueden finalizar con un nodo hoja para cada valor objetivo en sus datos de entrenamiento.
- Este tipo de algoritmo busca minimizar el valor escogido para la selección de atributos por lo que está optimizado localmente, es decir, que no piensa en el futuro a la hora de decidir como dividirse en un nodo concreto.

- En cada partición o división, el árbol elige como agrupar las diferentes clases en los dos nodos siguientes por lo que la existencia de clases desequilibradas plantea un problema a tener en cuenta cuando se trata de clasificación ya aquellas clases con una representación muy baja (clases minoritarias) pueden perderse entre las mayoritarias. Esto supone que la predicción de estas sea menos probable de lo que debería, en el caso de que alguno de los nodos la prediga.

5.1.3 Tamaño del Árbol

La dimensión final que adquiere un árbol puede gestionarse mediante reglas de parada que detengan la división de los nodos en función de si se cumplen o no determinadas condiciones [10]. Estas pueden variar según el software o librería empleados, pero en general están presentes en los más comunes o comerciales y son las siguientes:

- Número máximo de nodos terminales: define el número máximo de nodos terminales que pueden existir en el árbol. Una vez alcanzado el límite, se detienen las divisiones.
- Profundidad máxima del árbol: define la profundidad máxima del árbol, entendiendo por profundidad máxima el número de divisiones de la rama más larga (en sentido descendente) del árbol.
- Reducción mínima de error: define la reducción mínima de error que tiene que conseguir una división para que se lleve a cabo.
- Observaciones mínimas para división: define el número mínimo de observaciones que debe tener un nodo para poder ser dividido. Cuanto mayor el valor, menos flexible es el modelo.
- Observaciones mínimas de nodo terminal: define el número mínimo de observaciones que deben tener los nodos terminales. Su efecto es muy similar al de observaciones mínimas para división.

Todos estos parámetros se denominan hiperparámetros porque no se aprenden durante el entrenamiento del modelo. Su valor debe ser establecido por el usuario en función de su conocimiento del problema y mediante el uso de validación cruzada.

5.2 Métodos Ensemble

Actualmente, existen los los algoritmos conocidos como métodos de conjunto o métodos combinados, que hacen entrenan árboles de decisión para diferentes subconjuntos de datos y obtiene respuesta para cada uno de ellos, de forma que finalmente se escoge el resultado con mayor número de votos. En el campo del aprendizaje automático, estos métodos (métodos de ensemble) utilizan múltiples algoritmos de aprendizaje para obtener un rendimiento predictivo que mejore el que podría obtenerse por medio de cualquiera de los algoritmos de aprendizaje individuales que lo constituyen.

En concreto, para los métodos de conjunto de árboles de decisión [11]se combinan varios de estos algoritmos para obtener un mejor rendimiento predictivo que si se utilizase un solo árbol de decisión. Sumado a esto, se reduce uno de los principales problemas del árbol de decisión, que es el sobreajuste. El principio fundamental detrás del modelo de conjunto es que muchos aprendizajes débiles se unen para formar uno fuerte. A pesar de todo, estos métodos de conjunto pueden tornarse bastante complejos y llegar a convertirse en modelos de caja negra, es decir, que perdamos la interpretación lógica del modelo. Por otra parte, podemos correr el riesgo de obtener un modelo sobreajustado si no se tienen algunas precauciones. En la práctica, la forma en que se seleccionan los modelos individuales que se combinan hacen uso de algunas técnicas que tienden a reducir los problemas relacionados con el exceso de ajuste de los datos de entrenamiento y mejoran la predicción conjunta.

Empíricamente, se ha comprobado que cuando existe una diversidad significativa entre los modelos individuales, las combinaciones tienden a obtener mejores resultados, por lo que muchos de los métodos existentes buscan promover la diversidad entre los modelos que se combinan, y ello provoca a veces que se

usen como modelos aquellos que hacen un uso fuerte de la aleatoriedad, en vez de modelos más dirigidos y que funcionan mejor individualmente. [12]

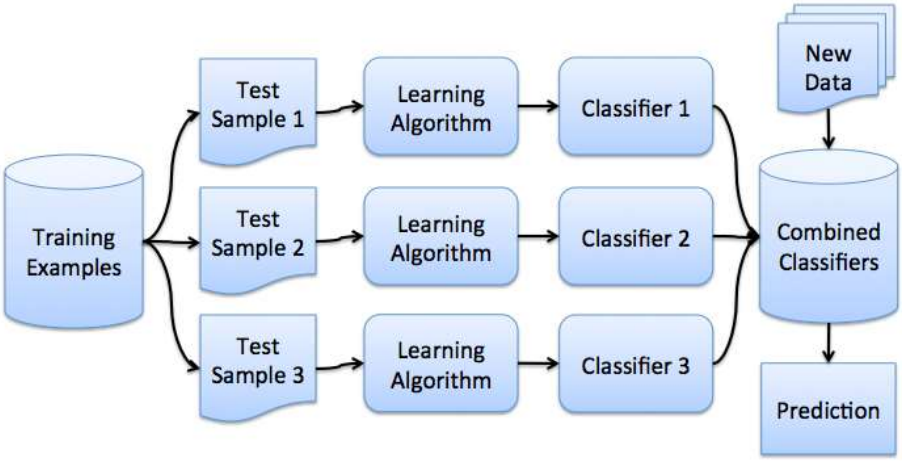


Figura 5-3: Esquema Métodos de Conjunto.

Fuente: http://www.cs.us.es/~fsancho/?e=106

5.2.1 Principales Métodos de Conjunto

A continuación, se mostrarán algunos de los métodos de combinación típicos y que más se emplean en la industria actualmente. [13]

Antes de profundizar en cada técnica de agregación, debemos definir los términos "muestreo repetido" o "bootstrapping" en inglés. Este término es importante tanto en el Bagging como en las técnicas de Random Forest.

Bootstrap hace referencia al remuestreo aleatorio con reemplazo y nos permite entender mejor el sesgo y la varianza con el conjunto de datos. Este método supone el muestreo aleatorio de un subconjunto de datos del conjunto de datos original. Como dato adicional, la selección de todos los ejemplos en el conjunto de datos tiene la misma probabilidad y, por otra parte, el proceso de muestreo permite el reemplazamiento por lo que, si un dato ha sido ya seleccionado, puede volver a ser elegido en el mismo subconjunto.

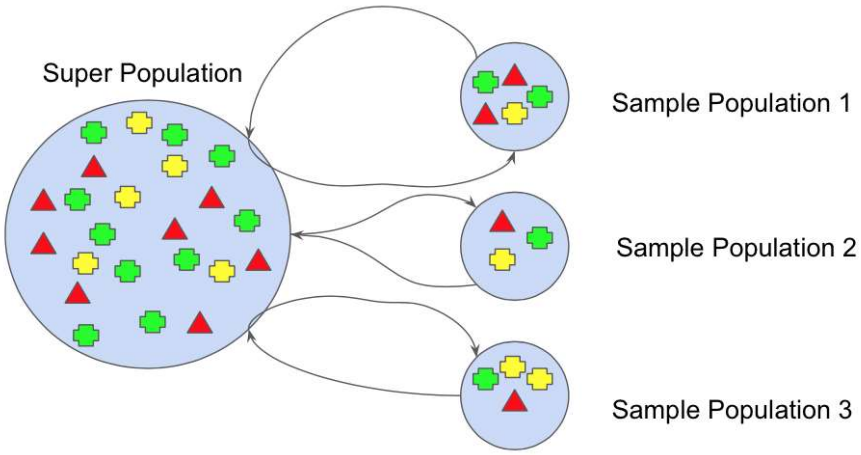


Figura 5-4. Ejemplo de Bootstrapping.

Fuente: https://www.i2tutorials.com/machine-learning-tutorial/machine-learning-bagging-boosting/

Para que esta aproximación se válida es necesario que se verifique una hipótesis que tiene carácter doble:

- Independencia: El tamaño *N* del conjunto de datos debe ser lo suficientemente grande en comparación con el tamaño *B* de las muestras de arranque para que las muestras no estén demasiado correlacionadas.

- Representatividad: El tamaño *N* del conjunto de datos inicial debe ser lo suficientemente grande como para capturar la mayor parte de la complejidad de la distribución subyacente, de modo que el muestreo del conjunto de datos sea una buena aproximación del muestreo de la distribución real.

El muestreo de Bootstrap se usa en los algoritmos de conjunto de aprendizaje automático bagging y Random Forest. Con esta técnica se ayuda a evitar el sobreajuste y mejora la estabilidad de los algoritmos de aprendizaje automático.

5.2.2 Decision Tree Bagging (Bootstrap Aggregation)

Esta técnica reduce la varianza y ayuda a reducir el sobreajuste. En general cuando se promedian varios modelos se obtiene un mejor ajuste que cuando se utiliza un solo modelo de algoritmo de aprendizaje. La idea básica es crear varios subconjuntos de datos a partir de una muestra de entrenamiento elegida de forma aleatoria con reemplazo, remuestrearlos y calcular las predicciones sobre el conjunto de datos remuestreados. Al promediar varios modelos conjuntamente se obtiene un mejor ajuste debido a que se suavizan tanto los modelos con sesgo como los que tienen alta varianza. [14]

Este método supone una mejora sobre la precisión de cualquiera de los clasificadores individuales utilizados, sobre todo si el modelo se encuentra sobreajustado. Además, es más robusto ya que el modelo compuesto reduce la varianza de los clasificadores individuales. La mejora que produce es más significante para clasificadores que varían de manera relevante con leves variaciones del conjunto de datos de entrenamiento. Por ejemplo, bagging produce una importante mejora en árboles de decisión, pero no en KNN (algoritmo K Vecinos Más Próximos).

Hay que tener en cuenta que, con el bagging, no dividimos los datos originales en subconjuntos más pequeños. Por el contrario, si tuviéramos una muestra inicial de tamaño *N*, continuaríamos alimentando cada árbol débil con un conjunto de entrenamiento de tamaño *N* (a menos que se indique lo contrario). Pero en lugar de los datos de entrenamiento originales, tomamos una muestra aleatoria de tamaño *N*. Por ejemplo, si nuestros datos de entrenamiento son [*A, B, C, D, E, F*], podemos darle a uno de nuestros árboles la siguiente lista [*A, B, B, C, F, F*]. Tenga en cuenta que ambas listas tienen una longitud de seis y que "*B*" y "*F*" se repiten en los datos de entrenamiento seleccionados al azar que alimentan nuestro árbol, esta es la propiedad de reemplazo.

Al hacer bagging [15]con árboles de decisión, nos podemos permitir tener árboles de decisión más sobreajustados. Por esta razón, cada uno de los árboles se hacen profundos (es decir, pocas muestras de entrenamiento en cada nodo de hoja del árbol) y no se podan. Estos árboles tendrán una alta varianza y un bajo sesgo. Pero, así, mejoramos la eficacia del modelo bagging.

La técnica de bagging se considera una técnica en paralelo ya que cada modelo individual de árbol de decisión se crea únicamente con los datos seleccionados con boostrapping.

5.2.2.1 Ventajas y Desventajas

La técnica de bagging resulta muy eficiente en muchos casos. Las ventajas que presenta son:

- Reducir la varianza respecto un modelo único.
- Elimina el sobreajuste que puede producirse si se usa un único modelo.

Pero como en cualquier técnica, también existen algunas desventajas:

- Introduce una pérdida de interpretabilidad.
- El modelo resultante puede experimentar muchos sesgos cuando se ignora el procedimiento adecuado.
- A pesar de destacar por su elevada precisión, puede ser computacionalmente costoso y esto puede desalentar su uso en ciertos casos.

5.2.3 Random Forest

Este método emplea una gran cantidad de árboles de decisión individuales que funcionan como un conjunto, de ahí que se le llame bosque [16]. Dentro de este, cada árbol individual origina una predicción de clase y la clase con mayor número de votos se convierte en la predicción final del modelo.

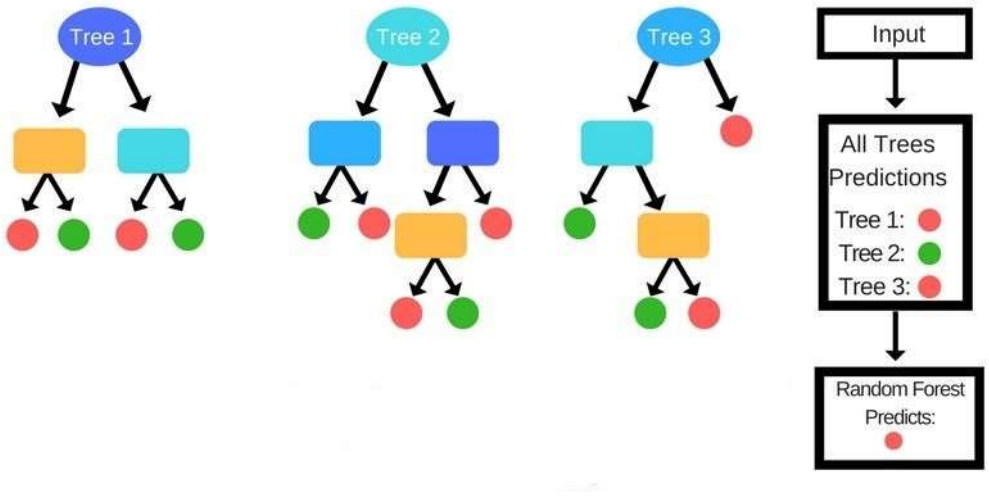


Figura 5-5. Random Forest.

Fuente: www.dataspirant.com

El concepto básico detrás de Random Forest es sencillo pero poderoso: la sabiduría de la multitud. En la ciencia de datos, la razón por la que un modelo de bosque aleatorio funciona tan bien es que una gran cantidad de modelos (árboles) casi no correlacionados, trabajando juntos como un comité, funcionarán mejor que todos los modelos que se ejecutan individualmente.

La baja correlación entre los modelos es clave. Los modelos no correlacionados pueden producir predicciones agregadas que son más precisas que cualquier predicción individual. La razón de este efecto es que los árboles se protegen unos a otros de sus errores individuales (siempre y cuando no se desplacen demasiado en la misma dirección errónea). Si bien algunos árboles pueden estar equivocados, muchos otros estarán en lo correcto, ya que un grupo de árboles se moverá en la dirección correcta.

Para asegurarse de que el comportamiento de cada árbol individual no esté demasiado relacionado con el de los otros árboles en el modelo, el Random Forest presenta las siguientes características:

- Bagging: Los árboles de decisión son muy sensibles a los datos en los que se entrenan, y pequeños cambios en el conjunto de entrenamiento pueden conducir a estructuras de árbol completamente diferentes. Random Forest se aprovecha de esto al permitir que cada árbol tome muestras aleatorias de forma individual de un conjunto de datos con alternativas, lo que da como resultado diferentes árboles.
- En un árbol de decisión normal, cuando llega el momento de dividir un nodo, consideramos todas las características posibles y elegimos la que produce la mayor división entre las observaciones del nodo derecho y las del nodo izquierdo. Por el contrario, cada árbol en un bosque aleatorio solo puede elegir características de un subconjunto aleatorio. Esto conduce a mayores diferencias entre los árboles más débiles en el modelo y es un factor que conduce a una asociación entre los árboles más débiles y una mayor diversidad.

Se puede concluir con que, en el Bosque Aleatorio, acabamos con árboles débiles que están entrenados en distintos conjuntos de datos (gracias al empleo del bagging) y que, además, también usan diferentes características para tomar sus decisiones. Esta es la razón por la cual los árboles débiles tienen muy poca correlación entre sí, pudiendo así, protegerse de sus errores.

5.2.3.1 Ventajas y Desventajas

Random Forest se basa en el algoritmo de bagging y emplea técnicas de aprendizaje por conjunto. Crea árboles en el subconjunto de datos y combina la salida de todos ellos. De esta forma, es lógico que algunas ventajas y desventajas coincidan con las del bagging. Las ventajas a destacar son:

- Aminora el problema de sobreajuste en los árboles de decisión, disminuye la variación y, como consecuencia, optimiza la precisión
- Puede trabajar con miles de variables de entrada e identificar las más representativas.
- La existencia de muy pocas suposiciones implica que la preparación de los datos es mínima (no hay que escalar los valores, por ejemplo).
- Incorpora métodos efectivos para estimar valores faltantes.
- Generalmente, Random Forest es robusto para los valores atípicos y puede operar con ellos de manera automática.

Entre las desventajas de los bosques aleatorios se pueden encontrar:

- Pérdida o falta de interpretación.
- Poco control en lo que hace el modelo (modelo caja negra para modeladores estadísticos).
- Mayor complejidad del modelo, se necesitan muchos más arboles débiles. Este algoritmo requiere una potencia más elevada y mayores recursos computacionales. Frente al árbol de decisión es simple y no requiere tantos recursos computacionales.
- Se incrementa el período de entrenamiento, el Bosque Aleatorio necesita mucho más tiempo para entrenar en comparación con los árboles de decisión normales, ya que genera muchos árboles y toma la decisión en función de la mayoría de los votos.

5.2.4 Decision Tree Boosting

El término Boosting se refiere a generar un algoritmo que nos permita obtener un modelo con menos errores de predicción a través del entrenamiento secuencial de varios modelos [17]. Para el presente desarrollo, los modelos débiles serán arboles de decisión.

El presente algoritmo pretende minimizar los errores del árbol predecesor. Se considera que cada árbol en el boosting es un modelo bastante débil (baja precisión de predicción), añadir varios árboles débiles en serie y cada uno centrado en los errores del anterior hace que el boosting sea un modelo altamente eficiente y preciso.

Como apunte, es importante saber que cada vez que se agrega un árbol nuevo, este emplea una versión moderadamente modificada del conjunto de datos inicial.

De esta manera, la base principal del boosting, al igual que el bagging, consiste en combinar varios modelos débiles para generar uno nuevo más fuerte. La diferencia entre ellos radica en que en el algoritmo boosting se crean los modelos débiles de manera secuencial, donde cada modelo intenta subsanar los errores del modelo anterior a él.

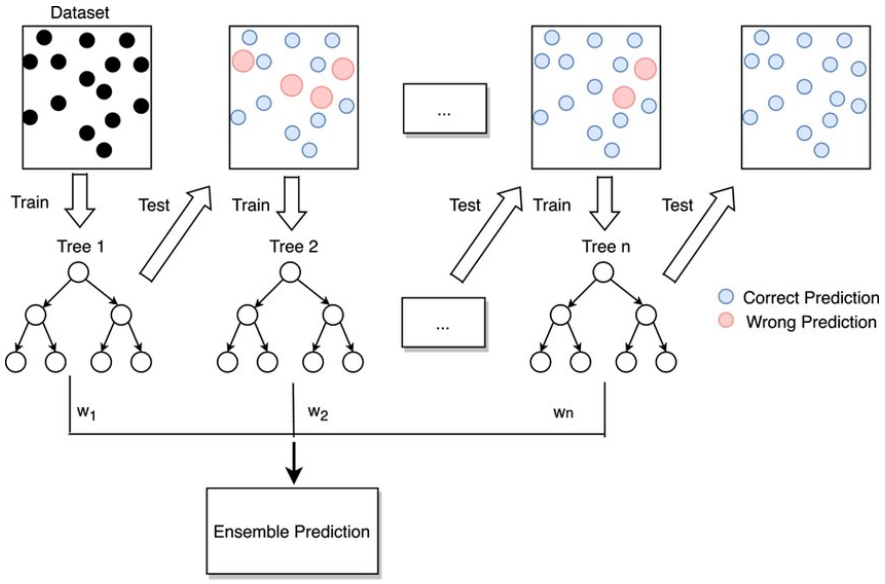


Figura 5-6. Diagrama de una Máquina Boosting.

Fuente: www.researchgate.net/figure/Flow-diagram-of-gradiente-boosting-machine-learning-method-The-ensemble-classifiers-fig-1-351542039

Existen diferentes tipos de algoritmos que funcionan mediante las técnicas de boosting. Todos ellos utilizan la misma idea base de realizar modelos débiles en serie, pero empleando técnicas con pequeñas diferencias entre ellas. Los modelos más populares de boosting son:

- AdaBoost (Adaptive Boosting): este método emplea el algoritmo para corregir al modelo anterior prestando mayor atención a las instancias incorrectas de entrenamiento de su predecesor. Debido a esto, en cada nuevo predictor, el modelo se centrará en aquellos casos que hayan sido más erróneos en el modelo previo.
- Gradient Boosting: su funcionamiento es muy similar al del AdaBoost. La diferencia fundamental se encuentra en cómo ajusta los valores erróneos del modelo anterior. Este método busca ajustar el nuevo predictor a los errores residuales cometidos por el predictor anterior, modificando los pesos de las instancias en cada iteración.
- XG Boost algorithm: es un método avanzado de Gradient Boosting. Este tiene un elevado poder predictivo y es del orden de diez veces más veloz que cualquier otra técnica de aumento de gradiente. Por otra parte, incluye una variedad de regularización que mejora el rendimiento general y reduce el sobreajuste.

5.2.4.1 Adaboost

También conocido como Adaptative Boosting, este método de conjunto iterativo en serie consiste en una combinación de múltiples clasificadores débiles con el fin de incrementar la precisión del clasificador final [18].

Generalmente, cualquier algoritmo de aprendizaje automático que acepte pesos en el conjunto de entrenamiento puede utilizarse como clasificador base. Adaboost se apoya en el concepto básico de establecer el peso de cada clasificador y, en cada iteración, entrenar la muestra de datos con el fin de garantizar la precisión de las predicciones de observaciones anteriores inusuales. En el caso de Adaboost se deben cumplir dos condiciones:

- El clasificador debe entrenarse de forma interactiva en varios ejemplos de entrenamiento ponderados.
- En cada iteración, debe intentar proporcionar un excelente ajuste para estos ejemplos minimizando el error de entrenamiento.

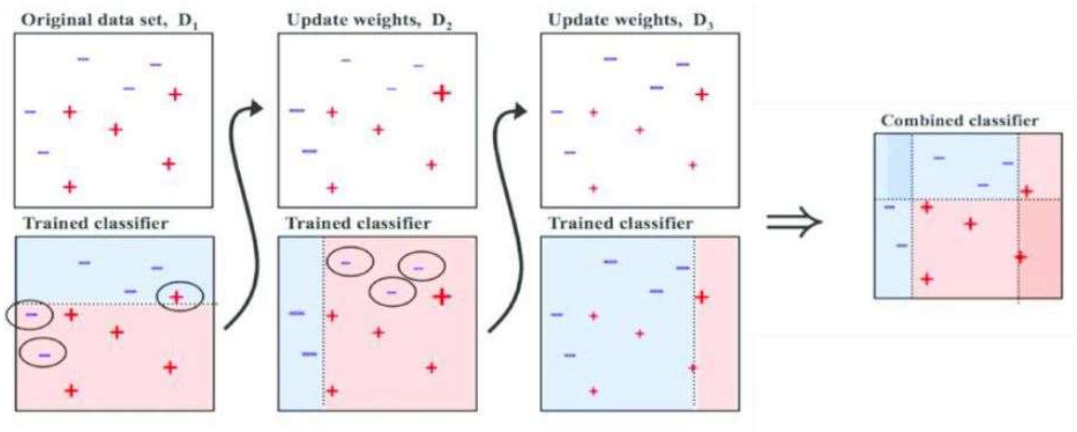


Figura 5-7. Adaboost.

Fuente: https://towardsdatascience.com

Como se puede ver en la Figura 4-7, el algoritmo sigue los siguientes pasos:

- Inicialmente, escoge un subconjunto de entrenamiento al azar.
- El siguiente paso consiste en ir seleccionando, de forma iterativa, el conjunto de entrenamiento basándose en la predicción precisa de último entrenamiento.
- El tercer paso consiste en asignar el mayor peso a aquellas iteraciones clasificadas como incorrectas, de manera que, en la siguiente iteración, estas tengan una alta probabilidad de clasificación. Como esta asignación de peso al clasificador débil entrenado en cada iteración la hace de acuerdo con la precisión del clasificador, se consigue que clasificador más preciso tenga un alto peso.
- El algoritmo iterará hasta que los datos de entrenamiento se ajusten sin errores o hasta llegar al número máximos de estimadores definido.
- Una vez que todos los aprendices han sido entrenados, AdaBoost realiza la predicción final dando a cada aprendiz un voto ponderado según el peso relativo que haya recibido.

En el caso del Adaboost aplicado en árboles de decisión, existen tres ideas fundamentales para su aplicación:

- Busca conseguir una clasificación final precisa mediante la combinación de modelos débiles. De esta manera, cada árbol de decisión forma un modelo débil y de profundidad 1 por lo que son muy simples.
- No todos tendrán la misma relevancia ya que, de los árboles resultantes, se identifican los pesos que hay que asignarle a cada uno mediante la fórmula indicada en la Figura 4-8, donde el error es la suma de los pesos de los datos que han sido clasificados como incorrectos. Se aprecia que, si el error es muy pequeño, el árbol obtendrá una ponderación elevada, de manera que se ocupará una posición importante en las votaciones finales.

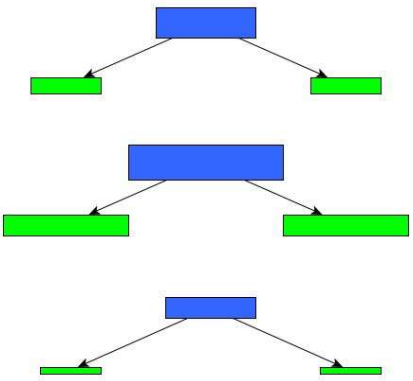


Figura 5-8. Árboles resultantes tras asignación de pesos a los simples.

Fuente: elaboración propia.

$$A = \frac{1}{2}\log\left(\frac{1 - Error\ total}{Error\ Total}\right)$$

- Para cada árbol débil se tienen en cuenta los errores del modelo anterior y cuando el dato anterior es correcto, el peso decrece mientras que, si se ha obtenido un dato incorrecto, se asigna un valor mayor del nuevo peso.

5.3 Clasificación de Naive Bayes

Este clasificador, también conocido como "Bayesiano ingenuo" es una aproximación simple pero que, por lo general, suele obtener buenos resultados en aplicaciones reales. Este modelo es un tipo de red Bayesiana denominada "Naive" y tiene en cuenta dos premisas de simplificación [19]. Estas son:

- Las variables predictoras son condicionalmente independientes entre ellas dada la clase.
- Asume que no hay atributos ocultos o latentes que puedan influenciar en el proceso de predicción.

Como su nombre indica, se basa en el teorema de Bayes:

$$\mathrm{P(etiqueta|características)} = \frac{\mathrm{P(c|e)P(e)}}{\mathrm{P(c)}}$$

Donde *P(etiqueta)* es la probabilidad inicial de una etiqueta y, *P(etiqueta | características)* es la probabilidad que la etiqueta *e* sea verdadera, conociendo la característica *c*. Finalmente, *P(características)* es la probabilidad a priori de que se den un conjunto determinado de características. Unido a las suposiciones descritas anteriormente y que Naïve asume, se podría reescribir la ecuación como:

$$\mathrm{P(etiqueta|características)} = \frac{\mathrm{P(e)P(c_1|e)\ldots P(c_n|e)}}{\mathrm{P(c)}}$$

5.3.1 Ventajas y Desventajas

Naïve Bayes es uno de los algoritmos más sencillos y potentes que, a pesar de los continuos avances de Machine Learning, ha demostrado su valía. Destaca por las siguientes ventajas:

- Es simple y rápido predecir la clase de conjunto de datos de prueba y también trabaja bien en la predicción multiclase.
- Funciona bien en el caso de variables de entrada categóricas comparada con variables numéricas.
- Teniendo en cuenta la suposición de independencia, este clasificador funciona mejor que otros modelos que necesitan más datos de entrenamiento.

Al igual que en el resto de los modelos, existen una serie de desventajas a tener en cuenta:

- Si la variable categórica tiene una categoría en el conjunto de datos de prueba, la cual no se observó en el conjunto de datos de entrenamiento, el modelo asignará una probabilidad de 0 y no será capaz de realizar una predicción. Esto es conocido como frecuencia cero y puede resolverse utilizando la técnica de aislamiento.
- Asunción de predictores independientes ya que, en la realidad, es muy complicado que obtengamos un conjunto de predictores que sean completamente independientes.

5.4 Support Vector Machines

También conocido con las siglas SVM, se trata de un algoritmo de clasificación binario, aunque también permite la clasificación multiclase a través del método uno contra el resto [20]. Este método se basa en que, dado un conjunto de puntos en el lugar *N* dimensional, el algoritmo SVM genera un hiperplano (*N-1*) que divide todos los puntos en dos grupos, de forma que maximiza la distancia de los puntos más cercanos de cada clase al hiperplano (caso linealmente separable). Esto se puede ver en la siguiente figura para un problema de dimensión 2:

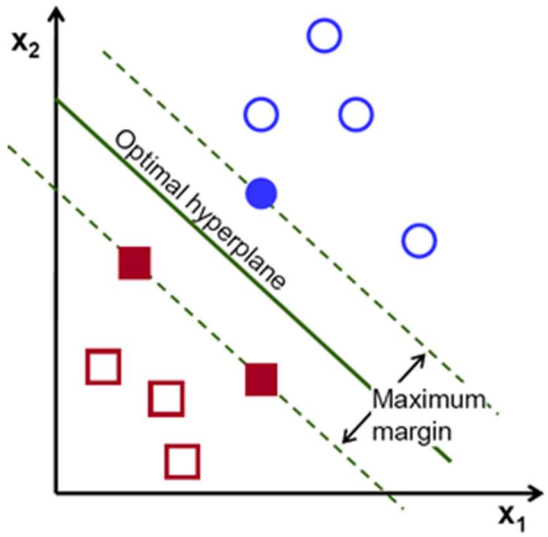


Figura 5-9. Problema en SVM para dimensión 2.

Fuente: www.aitrends.com

Aquellos datos de naturaleza dispersa son ideales para la clasificación SVM ya que, a pesar de que algunas características son irrelevantes, tienden a relacionarse unas con otras y, generalmente, tienden a organizarse en categorías separables. Este método es capaz de generar una superficie de decisión no lineal en el espacio de características original, asignando las instancias de datos de manera no lineal a un espacio mediante una transformación implícita, en la cual las clases se puedan separar linealmente con un hiperplano.

5.4.1 Ventajas y Desventajas

Como se ha visto, las máquinas de vectores de soporte son un conjunto de métodos de aprendizaje supervisado que pueden ser empleados para la clasificación. Presenta las ventajas expuestas a continuación:

- Es eficaz en espacios de grandes dimensiones.
- Funciona bien en aquellos casos donde el número de dimensiones es mayor que el de muestras.
- Emplea un subconjunto de puntos de entrenamiento en la función de decisión, por lo tanto, es eficiente en memoria.

Sus desventajas más destacables son:

- Cuando el número de características es mucho mayor que el de muestras hay que evitar el exceso de ajuste al elegir las funciones.
- Este método no proporciona directamente estimaciones de probabilidad, estas se deben calcular utilizando una validación cruzada quíntuple.

5.5 K-Vecinos más Cercanos

Este algoritmo, también conocido como K-NN (K-Nearest Neightbors), el de los K vecinos más cercanos, pertenece al grupo de algoritmos conocidos como "Lazy Learning Methods", estos no generan un modelo como resultado del aprendizaje con datos de entrenamiento, el modelo va aprendiendo en el momento en que se testean los datos.

El objetivo de este método es clasificar cada dato nuevo en el grupo que le corresponda, dependiendo de los K vecinos de un grupo que tenga más cerca y de la clase mayoritaria de estos *K* vecinos más cercanos [21]. Por lo tanto, el algoritmo realiza un cálculo de la distancia del elemento nuevo a cada uno de los existentes y ordena dichas distancias de menor a mayor para seleccionar el grupo al cual pertenece.

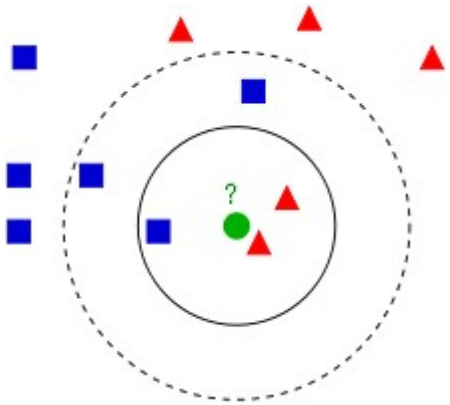


Figura 5-10. Algoritmo K-NN para K=3 y K=5.

Fuente: www.github.com

Como se puede observar en la Figura 4-10, la circunferencia más pequeña representa el caso para *K=3*, donde el punto verde queda asociado al grupo de los triángulos ya que son mayoría. Por otra parte, en el caso de *K=5* cambiaría al grupo de los cuadrados al tener 3 vecinos más cercanos en comparación con los dos triángulos.

6 RESULTADOS

Durante este apartado se implementará una comparación y estudio de los algoritmos de clasificación más relevantes en términos de prestaciones. Como es natural, para realizar la comparación, se empleará el conjunto de datos sobre de la encuesta de movilidad.

Los modelos seleccionados para su estudio son los siguientes:

- SVM
- Bagged Trees
- Boosted Trees
- Naive Bayes

Cada uno de los modelos mencionados se entrenará con los hiperparámetros que el modelo establece por defecto, lo que nos permitirá tener una visión más genérica del comportamiento de este. Posteriormente, estos parámetros serán susceptibles de algún ajuste con el objetivo de optimizar sus prestaciones.

Es importante subrayar que, dentro de todo el conjunto de datos existentes, se han seleccionado diferentes muestras para analizar el comportamiento del modelo y se ha realizado un depurado de la misma. Por lo tanto, según el criterio aplicado, se obtendrán dos muestras diferentes para su estudio. En primer lugar, se han agrupado los diferentes datos seleccionando aquellos individuos que a lo largo de la semana únicamente se desplazaban exclusivamente mediante una de las diferentes opciones, se ha denominado "Datos Transportes Únicos". A continuación, se analizará otra muestra llamada "Datos Transportes Agrupados" y, en tercer lugar, se obtendrán resultados a partir del estudio de la muestra mediante el método tradicional de clasificación

Por último, se analizarán los resultados obtenidos tras aplicar un método tradicional de clasificación para poder realizar una comparación con los anteriores modelos.

Para entender el estudio realizado es necesario aclarar los siguientes conceptos:

- Precisión (accuracy): este concepto se puede calcular en base a la matriz de confusión en la que se almacenan los valores de verdaderos negativos, verdaderos positivos, falsos negativos y falsos positivos. Se puede definir la exactitud como el número de casos que el modelo ha acertado (los positivos y los negativos), que matemáticamente se expresa de la siguiente manera:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}}$$

TP = Verdaderos positivos (True Positive)

FN = Falsos negativos (False negative)

FP = Falsos positivos (False Positive)

TN = Verdades negativos (True negative)

Cabe destacar que, al ser un porcentaje, sólo puede tener valor entre 0 y 1. Como es de esperar, los modelos ideales son aquellos cuyo valor de la exactitud sea mayor.

No obstante, para saber con certeza la calidad de un modelo es necesario una visión conjunta de todos los parámetros relativos al mismo ya que, en ocasiones, la exactitud puede llevar a engaño. Esto ocurre cuando las clases están desbalanceadas, es decir, si para el caso a estudiar tuviésemos 3 personas que van en bici y 150 que van en coche.

- Matriz de confusión:

  En el campo de la inteligencia artificial y el aprendizaje automático una matriz de confusión es una herramienta que permite visualizar el desempeño de un algoritmo de aprendizaje supervisado. Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila reproduce las instancias en la clase real., o sea en términos prácticos nos permite ver qué tipos de aciertos y errores está teniendo nuestro modelo a la hora de pasar por el proceso de aprendizaje con los datos.

- Curva ROC:

  Es una herramienta estadística utilizada en el análisis de clasificación de la capacidad discriminante de una prueba diagnóstica dicotómica. Es decir, basada en una variable de decisión, cuyo principal objetivo es clasificar a los individuos de una población en dos grupos: uno que presente un evento de interés y otro que no. Esta capacidad discriminante está sujeta al valor umbral elegido de entre todos los posibles resultados de la variable de decisión, es decir, la variable por cuyo resultado se clasifica a cada individuo en un grupo u otro. De esta forma, se genera un gráfico capaz de representar, para cada valor umbral, las medidas de sensibilidad (eje Y) y especificidad (eje X) de la prueba diagnóstica. Por una parte, la sensibilidad cuantifica la proporción de individuos que presenta el evento de interés y que son clasificados por la prueba como portadores de dicho evento. En segundo lugar, la especificidad cuantifica la proporción de individuos que no lo presentan y son clasificados por la prueba como tal.

6.1 Depuración de datos

Durante esta etapa se han tomado una serie de decisiones para evitar desbalances en la muestra; estos aparecen cuando en una muestra clasificada en grupos, alguna de las clases es dominante, es decir, tiene un número de individuos mucho mayor que los demás. Para el presente caso se parte de una muestra con las siguientes 12 clases:

- Coche
- Bus
- Bicicleta
- Andando
- Coche - Bici
- Coche - Bus
- Coche – Bici - Bus
- Coche - Bus - Andando
- Bici - Bus
- Bici - Andando
- Bus – Andando
- Coche – Bus – Bici - Andando

Una vez establecidas todas las clases existentes, se han llevado a cabo las siguientes acciones:

- Eliminación de los dos únicos individuos que empleaban los cuatro medios de transporte posibles (coche, autobús, bici, andando).
- Unión de las clases *bicicleta* y *andando.* Esto se debe a que, por separado el número de individuos era notablemente menor frente al resto de clases. Por lo que quedan las siguientes:
    - Coche
    - Bus
    - Bicicleta/Andando
    - Coche – Bici/Andando
    - Coche – Bus
    - Coche – Bus – Bici/Andando
    - Bus – Bici/Andando

Con esto se consigue una distribución de clases más equilibrada, esta servirá como base para la ejecución de los diferentes métodos de clasificación.

El modelo se entrena para que nos aporte el modo de desplazamiento que escogería para cada individuo. Para entrenar el modelo, de toda la información disponible, se escogen las siguientes variables de entrada como atributos ya que se considera que son las más representativas desde un punto de vista de la movilidad:

- Tiempo de desplazamiento según el medio de transporte
- Distancia de desplazamiento según el medio de transporte
- Sexo
- Edad
- Rol en la E.T.S.I.
- Disponibilidad de vehículo propio
- Código postal

A continuación, se ejecuta el software de Matlab para obtener los resultados en función de la base de datos empleada.

6.2 Resultados con Modo de Transportes Únicos Mediante Aprendizaje Automático

Para este análisis únicamente se han tenido en cuenta aquellos individuos que empleaban únicamente uno de los siguientes medios de transporte:

- Coche
- Bicicleta/Andando
- Autobús

El tratamiento del fichero Excel necesario para que el programa Matlab pueda trabajar con los datos ha sido realizado mediante un código Python recogido en Punto *7.4 Tratamiento de Datos* .

Por cada modelo y técnica se mostrará la matriz de confusión seguida de la curva ROC y el resto de las métricas empleadas para estudiar los resultados.

6.2.1 Método SVM

Para este algoritmo Matlab ha conseguido una exactitud (accuracy) del 80% lo que implica que nuestro modelo acierta en ese porcentaje de casos y que se obtiene de dividir las predicciones correctas entre el número total de predicciones. Por otra parte, ha generado la siguiente matriz de confusión.

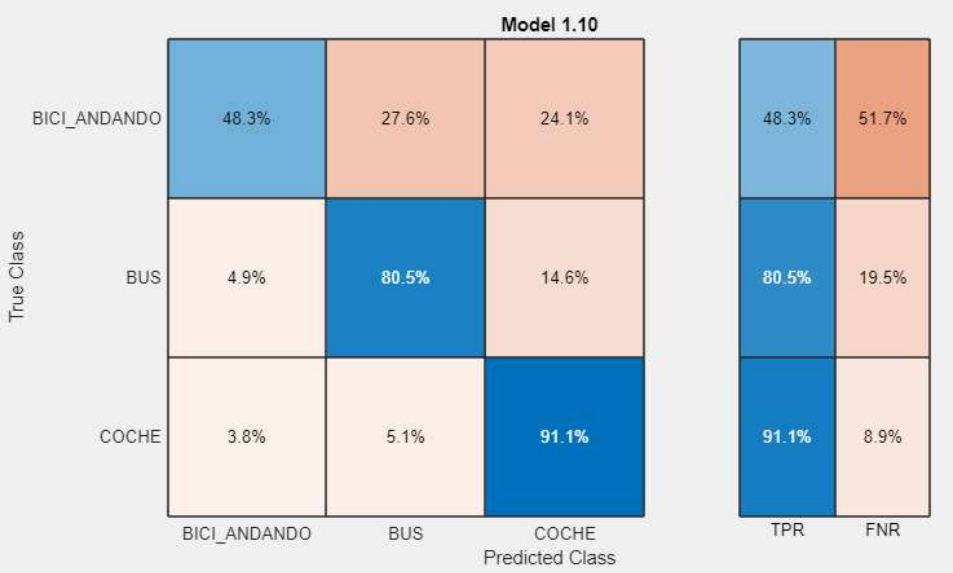


Figura 6-1. Matriz de confusión para SVM.

Fuente: MATLAB (elaboración propia).

Se explicará brevemente esta figura para que sirva de precedente para todo el resto que se muestran en el TFM.

Nos centraremos en los valores denominados TPR, FNR.

- True Positive Rates (TPR): también conocido como sensibilidad o exhaustividad de una clasificación es el ratio entre los verdaderos positivos y la suma de los verdaderos positivos y los falsos negativos. En otras palabras, es el ratio entre los verdaderos positivos detectados y los positivos reales. Su valor ideal sería 1 (en nuestro caso el 100%) y, como es lógico, el peor clasificador posible tendría una exhaustividad de 0.
- False Negative Rates (FNR): se define como el ratio entre el número de falsos negativos y el número de negativos (reales). Lógicamente, el clasificador ideal tendría un FNR de cero (pues no tendría falsos negativos), y el peor clasificador posible tendría un FNR de uno (todos los positivos reales serían identificados erróneamente como negativos).

Se puede observar que, para este caso, el clasificador “coche” ha obtenido el mejor TPR de todos.

A continuación, se expone el resultado obtenido basándose en la curva ROC:

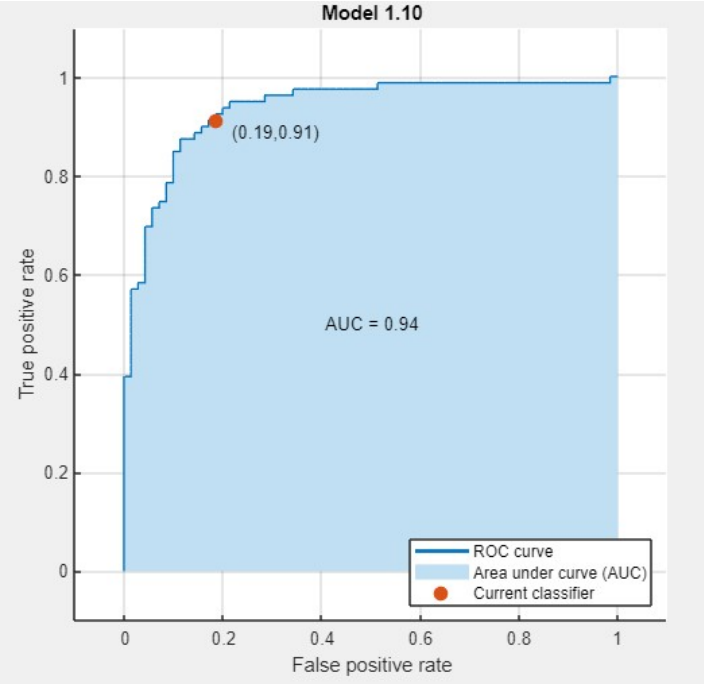


Figura 6-2. Curva ROC para la clase positiva Coche.

Fuente: MATLAB (elaboración propia).

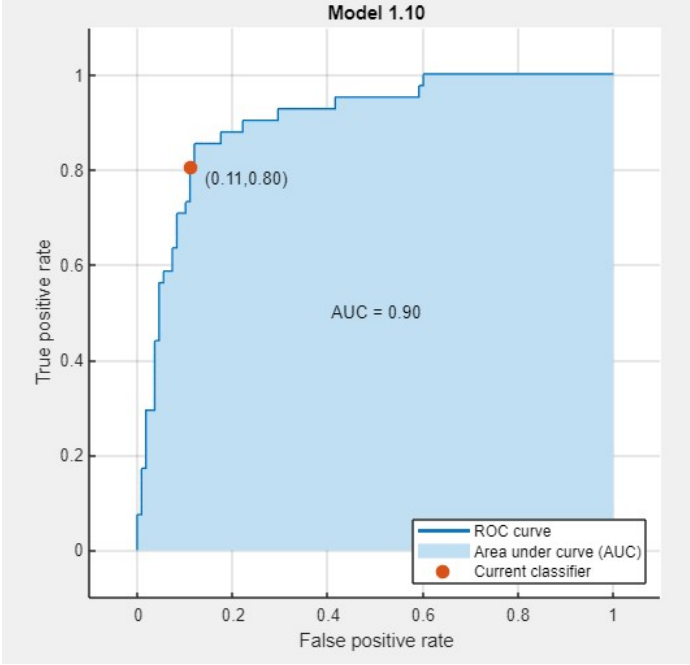


Figura 6-3. Curva ROC para la clase positiva Bus.

Fuente: MATLAB (elaboración propia).

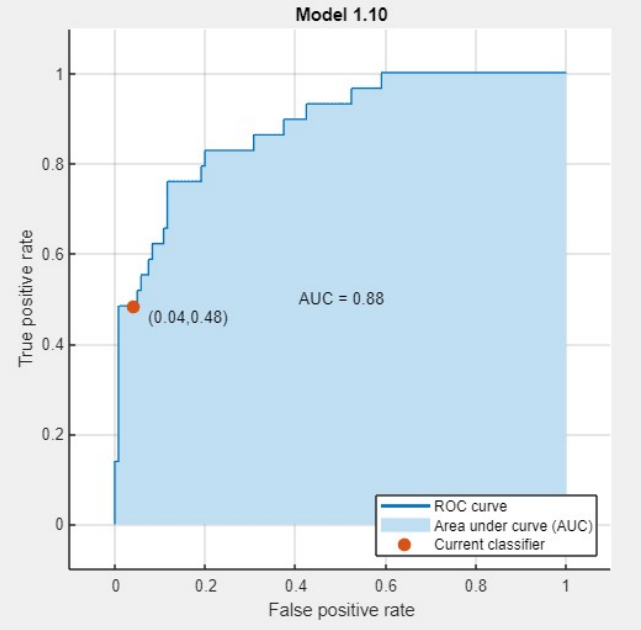


Figura 6-4. Curva ROC para la clase positiva Bici/Andando.

Fuente: MATLAB (elaboración propia).

- AUC: conocido como el Área Bajo la Curva, nos da una buena idea de qué tan bien funciona el modelo para cada clasificador y será mejor cuanto más próximo sea a 1, lo que implicaría tener una tasa de verdaderos positivos perfecta y una de falsos positivos de valor cero. En general, cuanto más "arriba y a la izquierda" del diagrama se encuentre la curva ROC, mejor será el clasificador. Este valor nos indica la probabilidad de que el modelo pueda distinguir entre clase positiva y negativa, es decir, para el caso de la figura 6-4 será capaz de predecir con un 88% entre la clase positiva y negativa del clasificador Bici/Andando, por lo que se puede afirmar que este modelo diferenciará bien entre los que si utilizan este modo de transporte y los que no. El eje Y representa la sensibilidad, mientras que el X la diferencia de 1 menos la especificidad.

6.2.2 Método Bagged Trees

En esta ocasión, la exactitud obtenida ha sido del 79'2 % y se ha obtenido la siguiente matriz de confusión junto con las curvas ROC correspondientes.

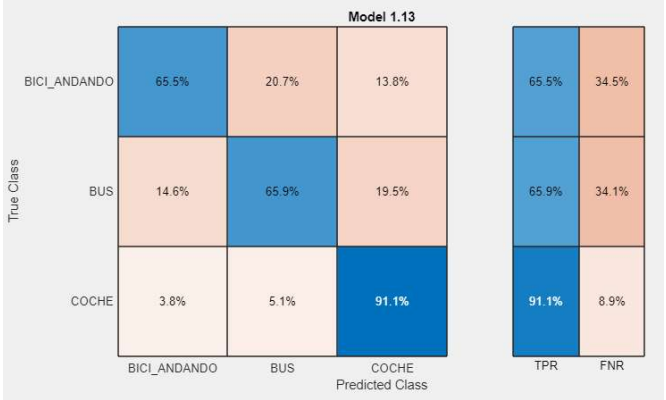


Figura 6-5. Matriz de confusión para Bagged Tree.

Fuente: MATLAB (elaboración propia).

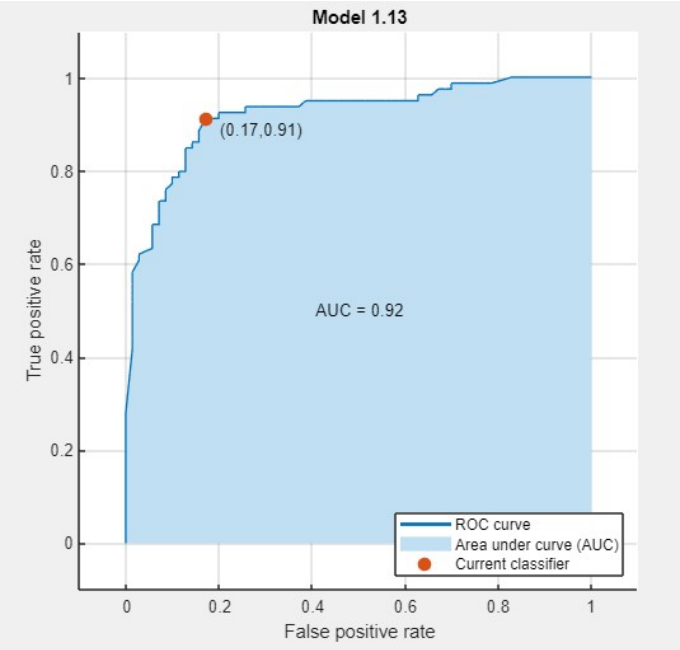


Figura 6-6. Curva ROC para la clase positiva Coche.

Fuente: MATLAB (elaboración propia).

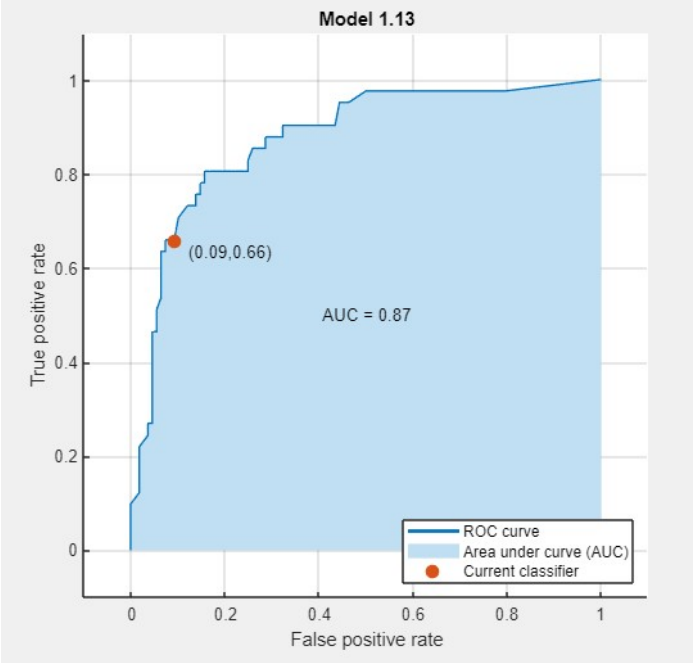


Figura 6-7. Curva ROC para la clase positiva Bus.

Fuente: MATLAB (elaboración propia).

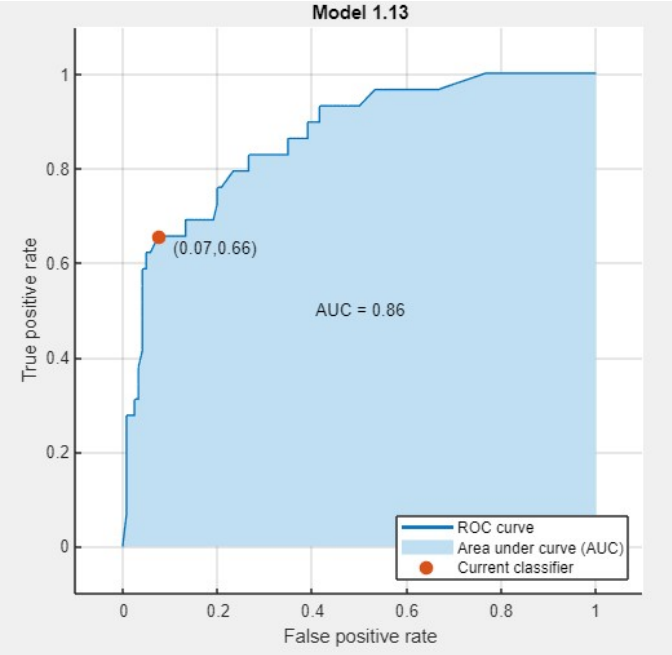


Figura 6-8. Curva ROC para la clase positiva Bici/Andando.

Fuente: MATLAB (elaboración propia).

6.2.3 Método Boosted Trees

En este caso, la exactitud que nos da este algoritmo es del 73’2%.

A continuación, se exponen el resto de los resultados obtenidos.

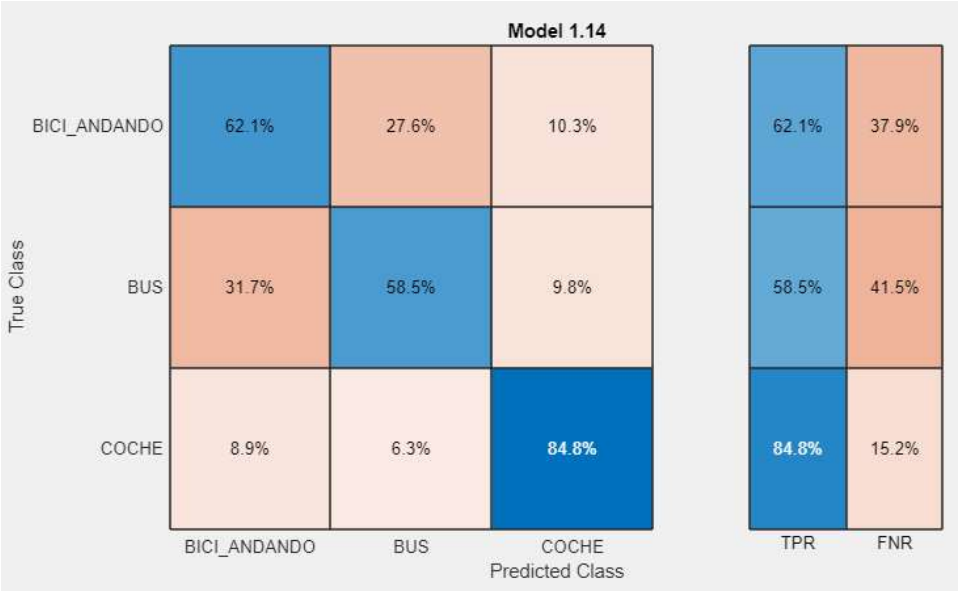


Figura 6-9. Matriz de confusión para Boosted Tree.

Fuente: MATLAB (elaboración propia).

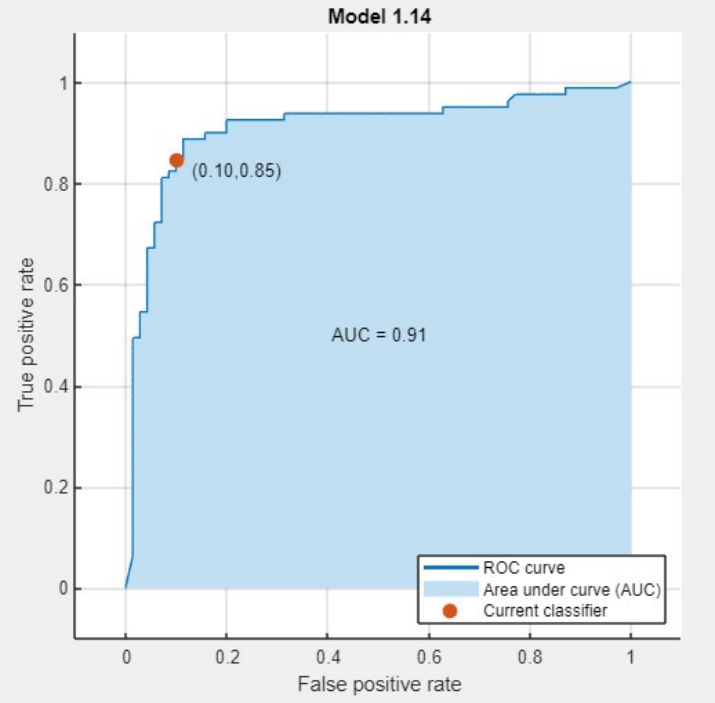


Figura 6-10. Curva ROC para la clase positiva Coche.

Fuente: MATLAB (elaboración propia).

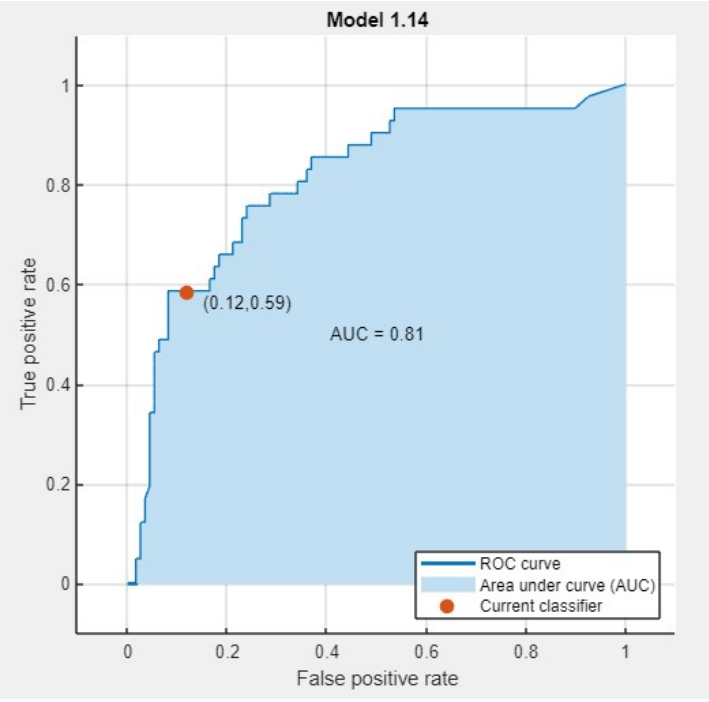


Figura 6-11. Curva ROC para la clase positiva Bus.

Fuente: MATLAB (elaboración propia).

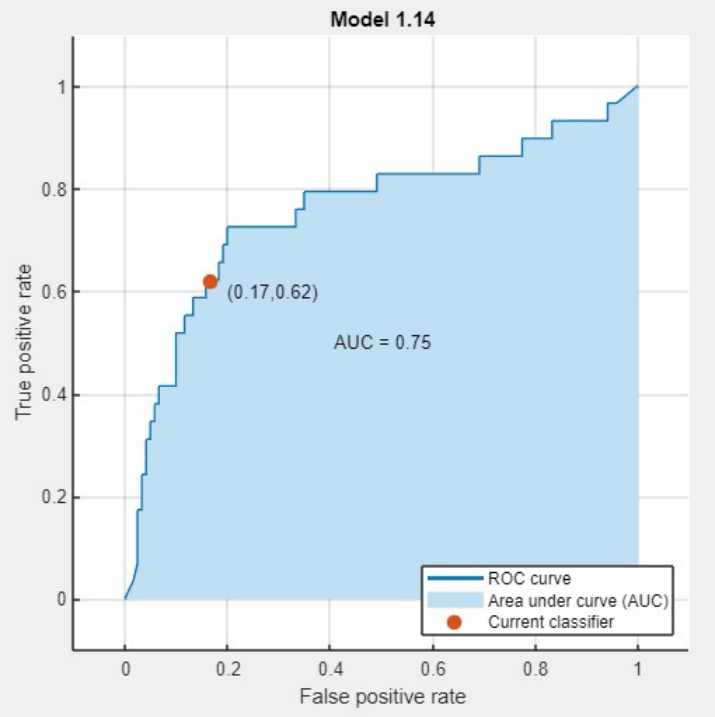


Figura 6-12. Curva ROC para la clase positiva Bici/Andando.

Fuente: MATLAB (elaboración propia).

6.2.4 Método Naive Bayes

Para este caso, se obtiene la exactitud más baja de los métodos analizados, con un valor de 72’5%.

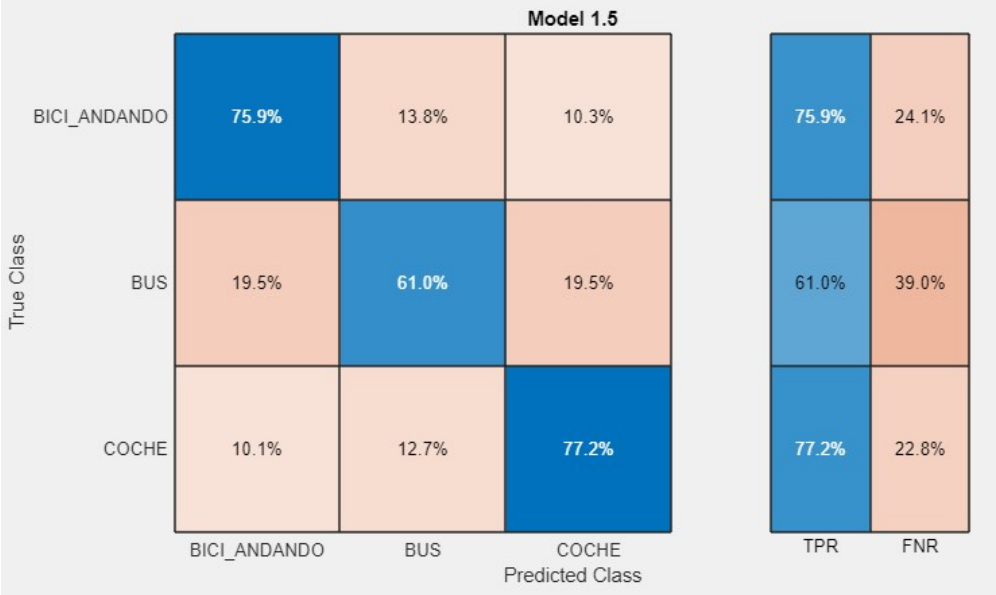


Figura 6-13. Matriz de confusión para Naive Bayes.

Fuente: MATLAB (elaboración propia).

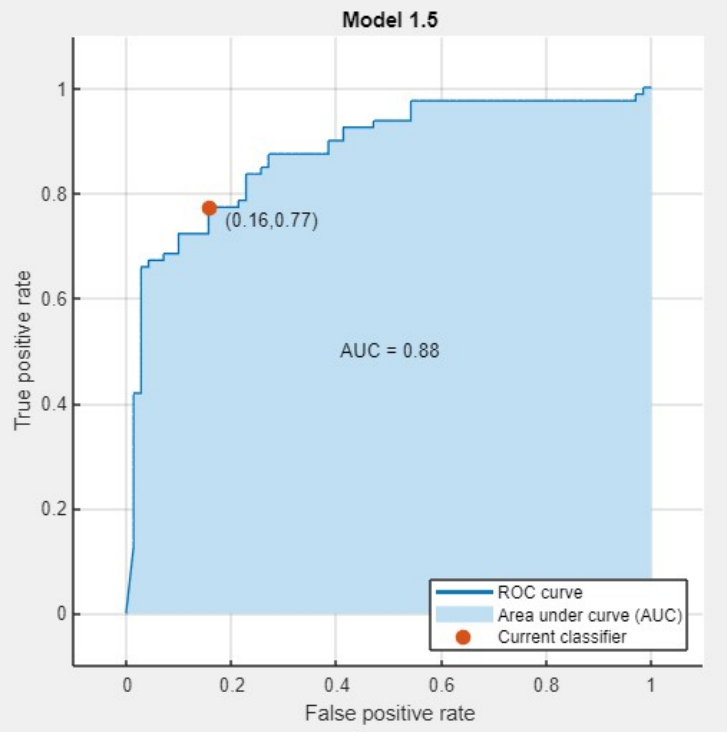


Figura 6-14. Curva ROC para la clase positiva Coche.

Fuente: MATLAB (elaboración propia).

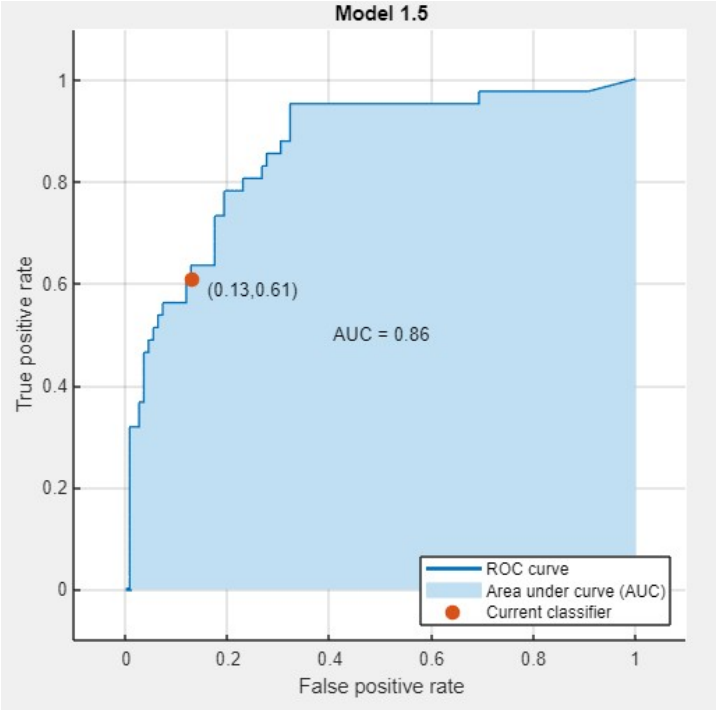


Figura 6-15. *Curva ROC para la clase positiva Bus.*

Fuente: MATLAB (elaboración propia).

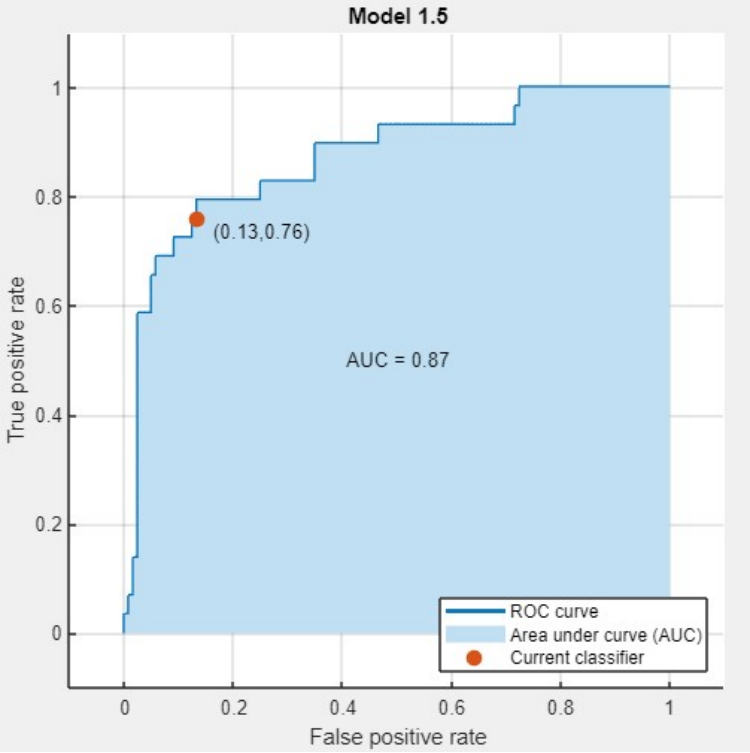


Figura 6-16. Curva ROC para la clase positiva Bici/Andando.

Fuente: MATLAB (elaboración propia).

6.3 Resultado con Modo de Transportes Agrupados Mediante Aprendizaje Automático

Durante este análisis, se han tenido en cuanta tanto aquellos individuos que utilizan un único método de transporte como los que utilizan dos. De esta forma, se ha tomado las siguientes decisiones:

- Aquellos que se desplacen unas veces andando y otras en autobús se considerarán como individuos que van andando.
- Para aquellos que emplean el coche y el autobús, se asumirá que se desplazan exclusivamente en bus.

Con esto, se consigue mayor equilibrio entre el número de individuos existentes para cada alternativa.

El tratamiento del fichero Excel necesario para que el programa Matlab pueda trabajar con los datos ha sido realizado mediante un código Python recogido en Punto *7.5 Tratamiento de Datos* .

Para este apartado se han seguido los mismos pasos que en el anterior, presentado los algoritmos analizados de mayor a menor precisión.

6.3.1 Método Bagged Trees

Este método obtiene una exactitud del 74%, siendo esta la mayor obtenida para el presente caso. Se puede observar que la precisión máxima obtenida es menor que la del caso anterior. Esto se debe a que se han tomado algunas decisiones erróneas a la hora de elegir un solo medio de transporte en aquellos casos en los que el individuo indicó que empleaba más de uno.

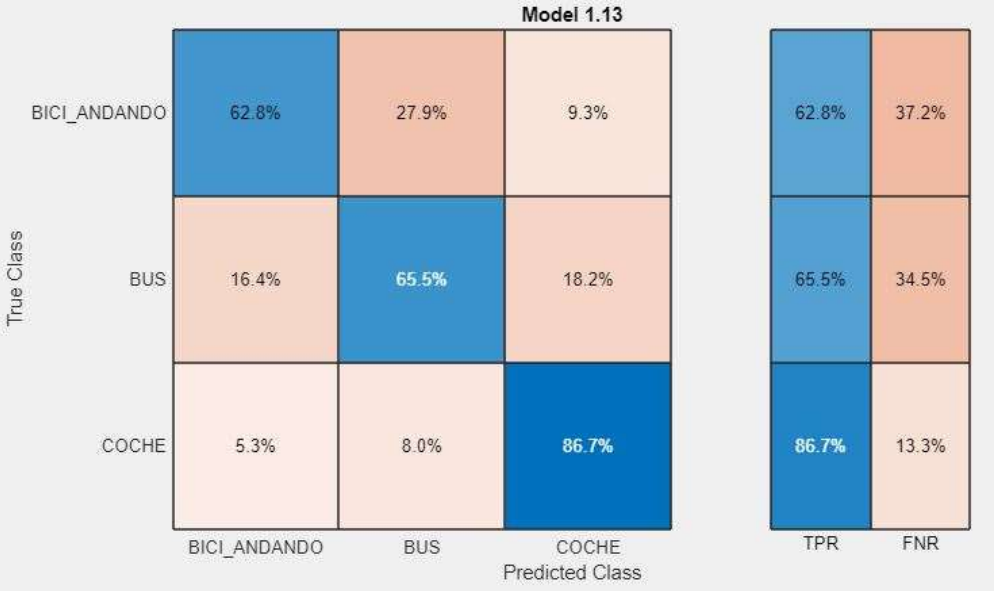


Figura 6-17. Matriz de confusión para Bagged Tree.

Fuente: MATLAB (elaboración propia).

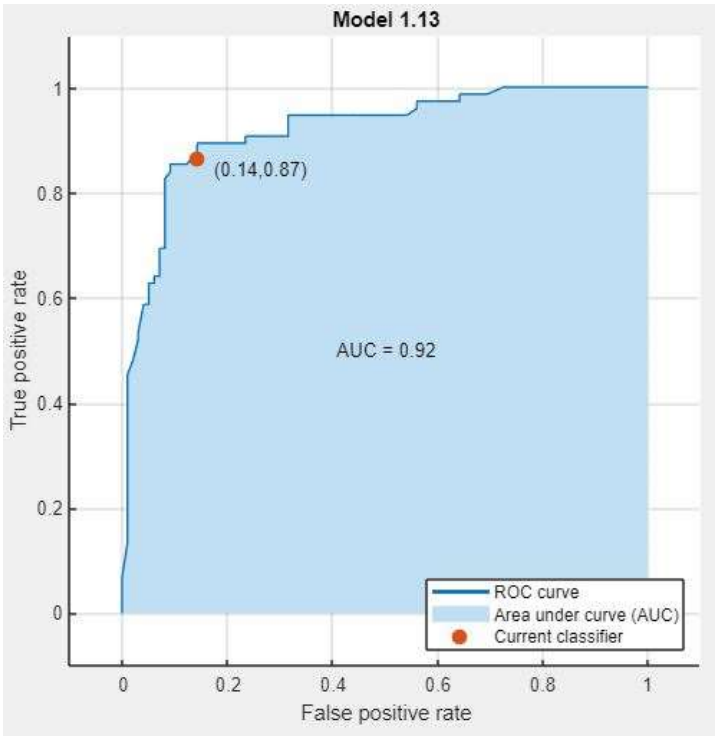


Figura 6-18. Curva ROC para la clase positiva Coche.

Fuente: MATLAB (elaboración propia).

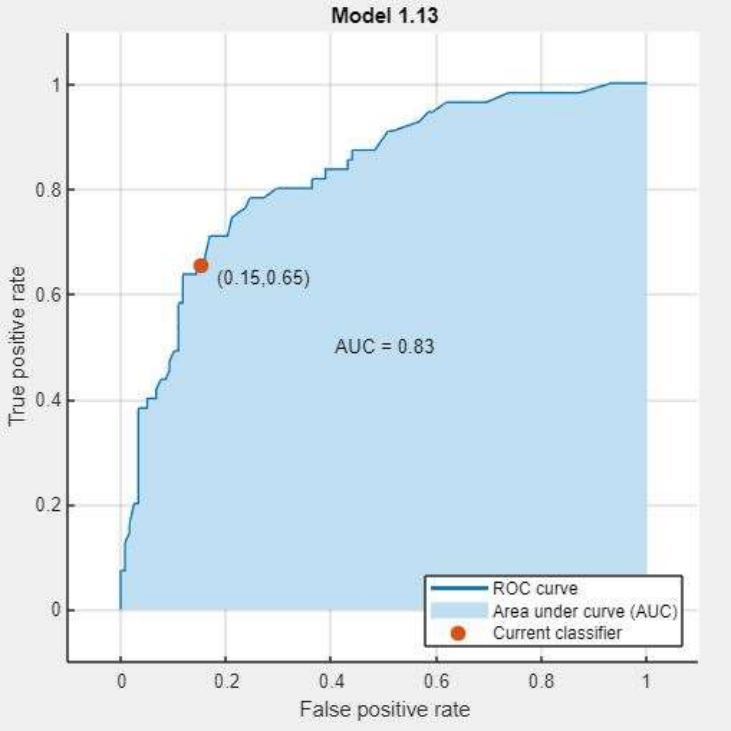


Figura 6-19. Curva ROC para la clase positiva Bus.

Fuente: MATLAB (elaboración propia).

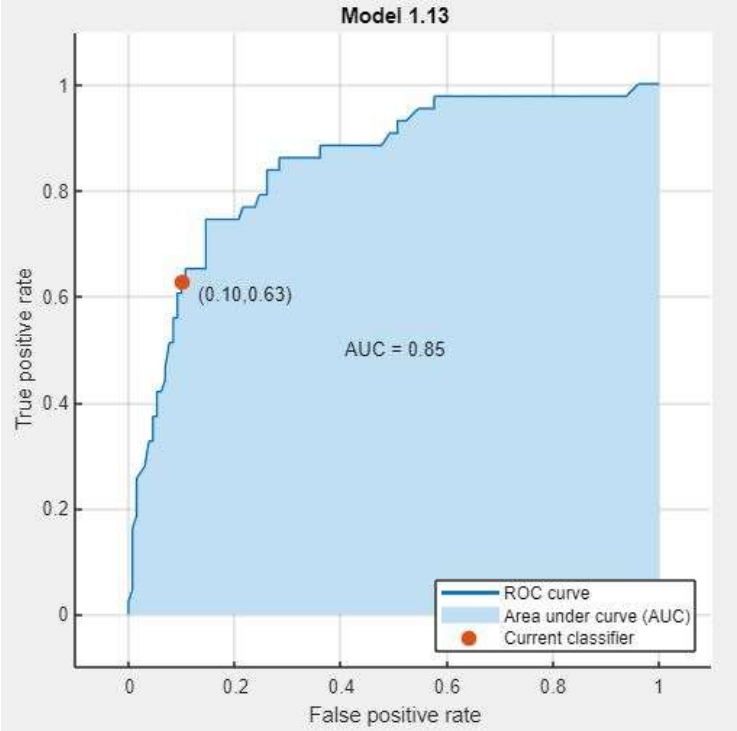


Figura 6-20. Curva ROC para la clase positiva Bici/Andando.

Fuente: MATLAB (elaboración propia).

6.3.2 Método SVM

La exactitud obtenida ha sido del 71’7%. A continuación, se exponen la matriz de confusión y las curvas ROC generadas.

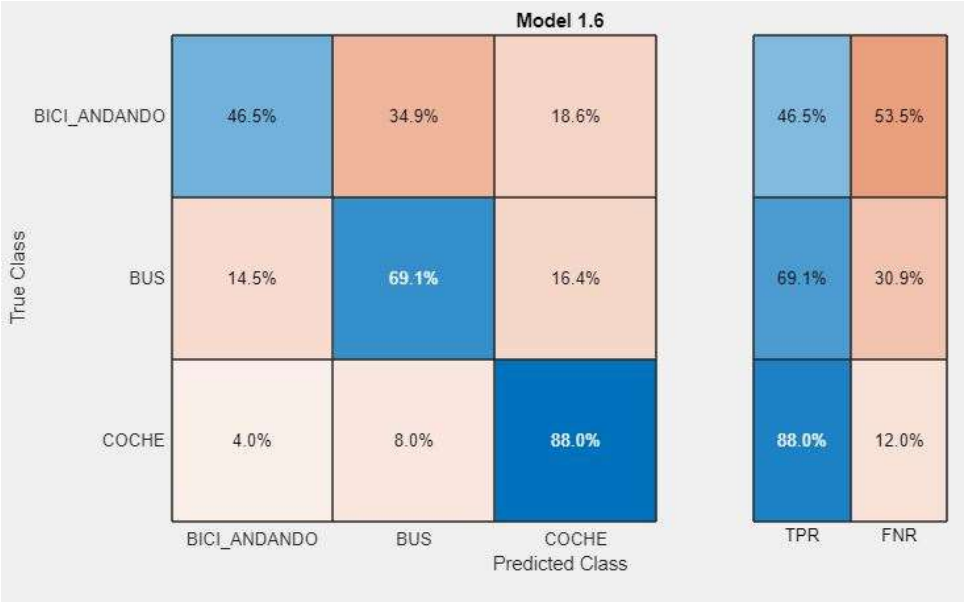


Figura 6-21. Matriz de confusión para SVM.

Fuente: MATLAB (elaboración propia).

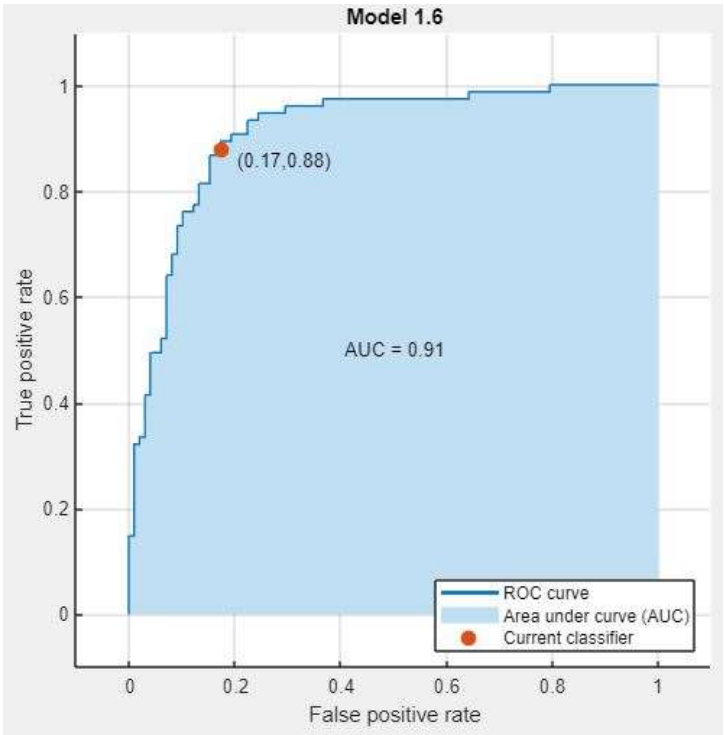


Figura 6-22. Curva ROC para la clase positiva Coche.

Fuente: MATLAB (elaboración propia).

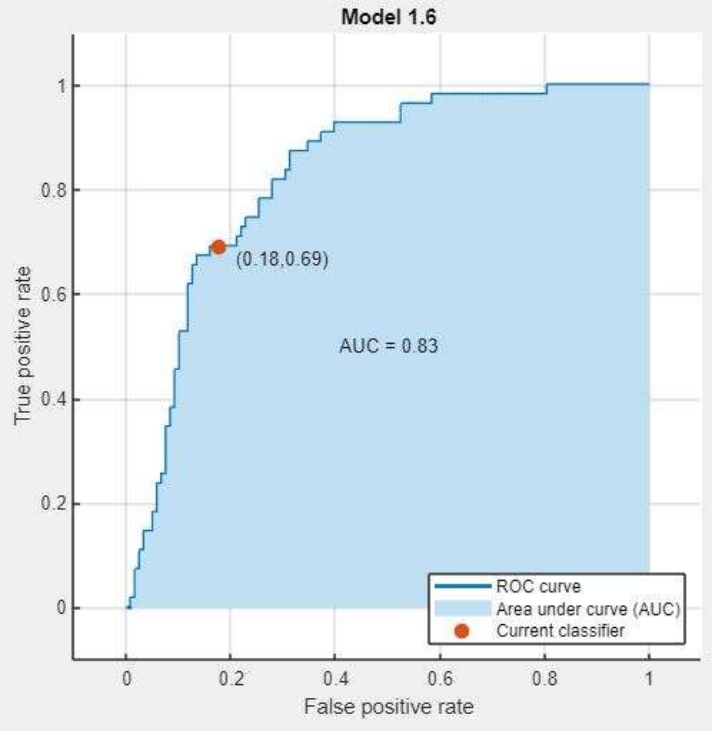


Figura 6-23. Curva ROC para la clase positiva Bus.

Fuente: MATLAB (elaboración propia).

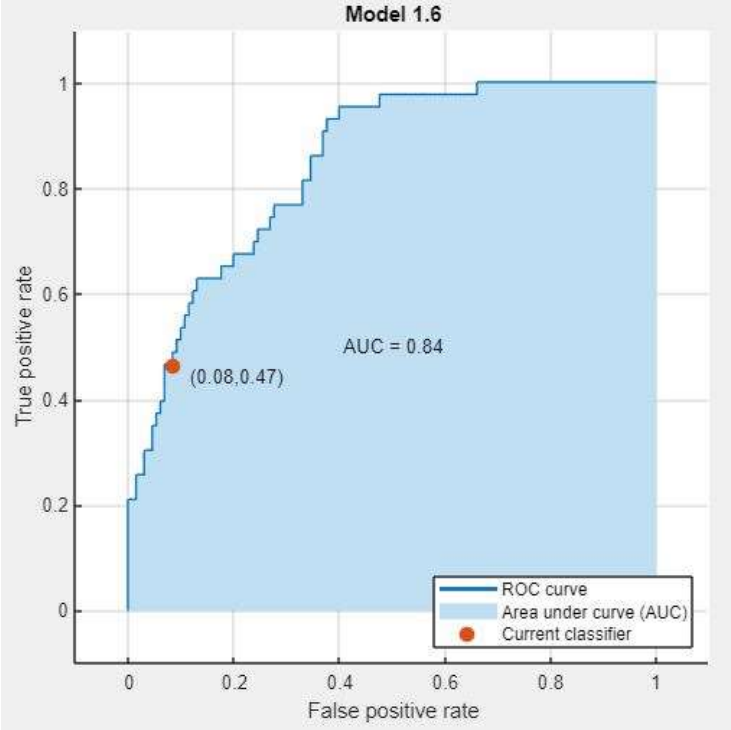


Figura 6-24. Curva ROC para la clase positiva Coche.

Fuente: MATLAB (elaboración propia).

6.3.3 Método Boosted Trees

La exactitud para este caso es de 65’9%.

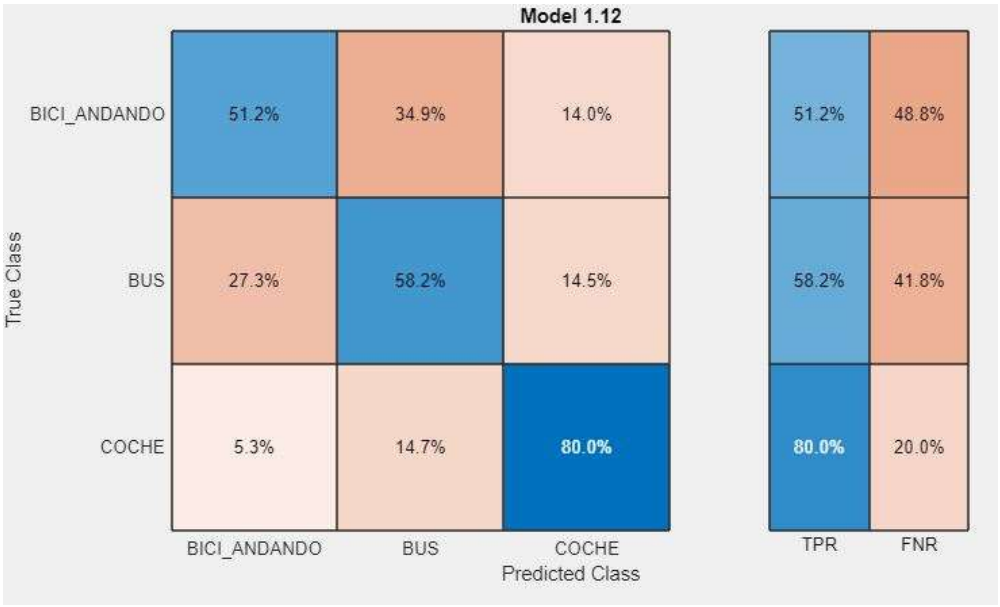


Figura 6-25. Matriz de confusión para Boosted Tree.

Fuente: MATLAB (elaboración propia).

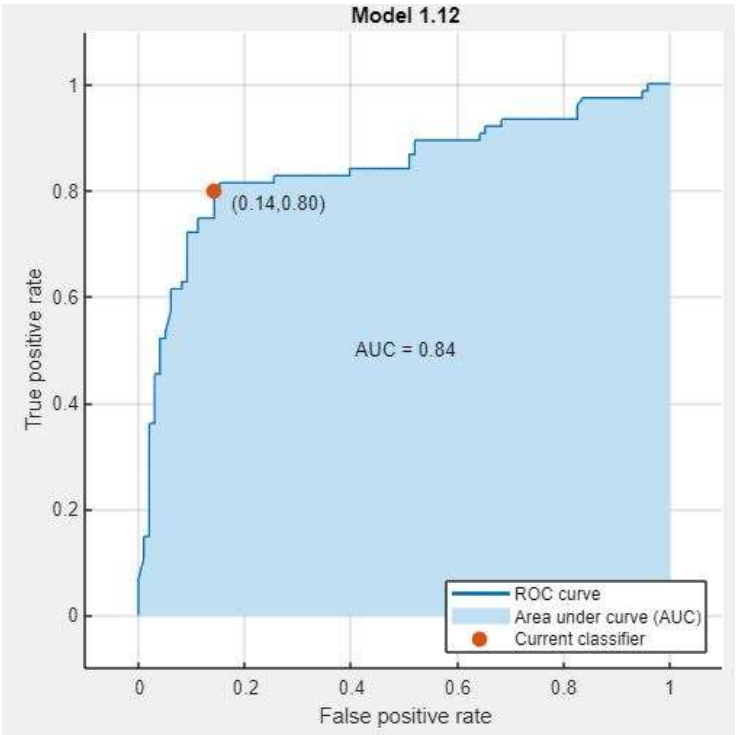


Figura 6-26. Curva ROC para la clase positiva Coche.

Fuente: MATLAB (elaboración propia).

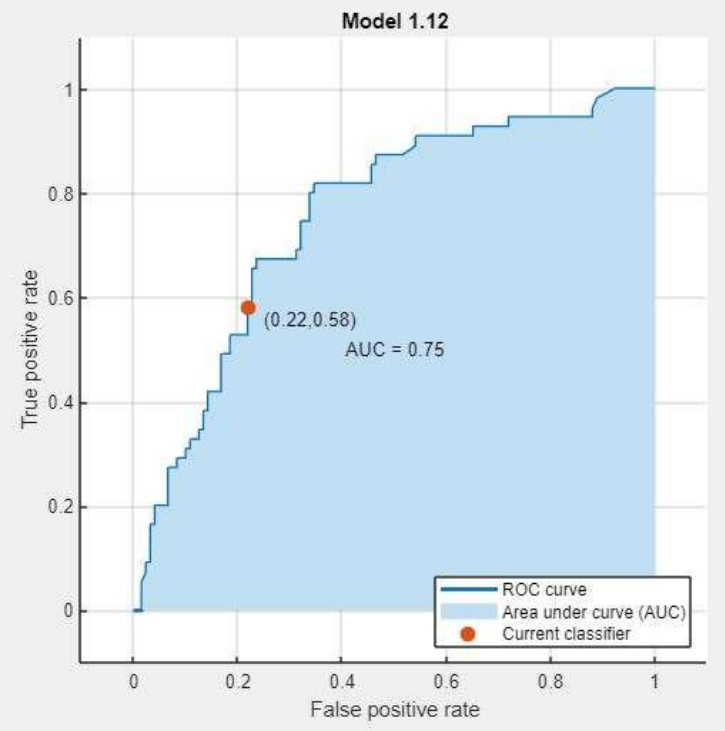


Figura 6-27. Curva ROC para la clase positiva Bus.

Fuente: MATLAB (elaboración propia).

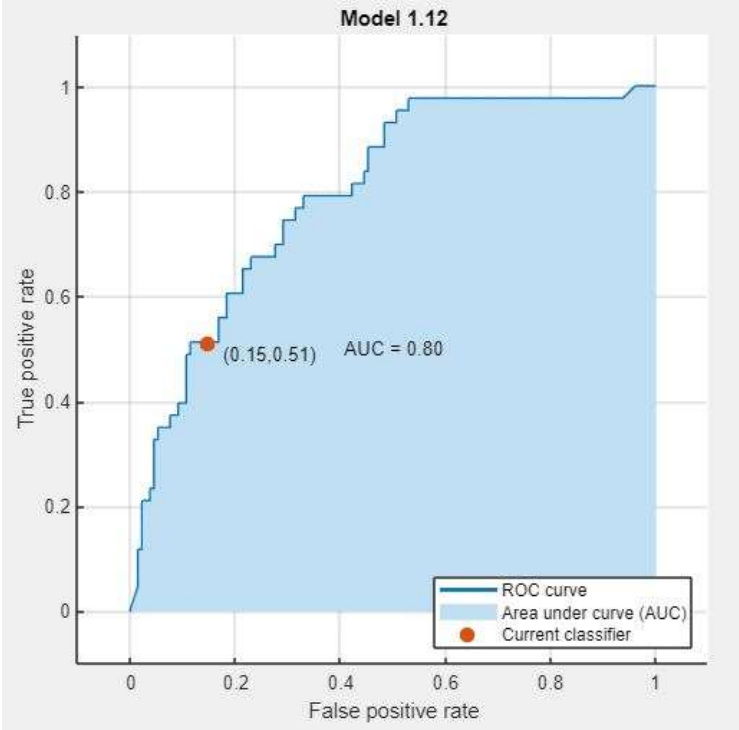


Figura 6-28. Curva ROC para la clase positiva Bici/Andando.

Fuente: MATLAB (elaboración propia).

6.3.4 Método Naive Bayes

Al igual que en el caso de los datos de transportes únicos, este algoritmo vuelve a ser el que menor exactitud ofrece, con un valor del 65'3%.

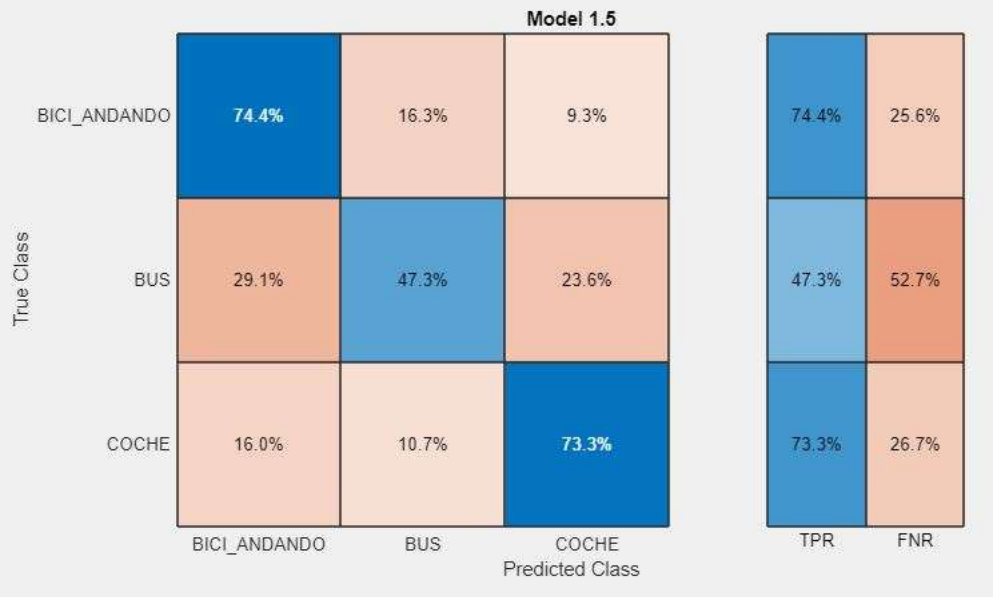


Figura 6-29. Matriz de confusión para Naive Bayes.

Fuente: MATLAB (elaboración propia).

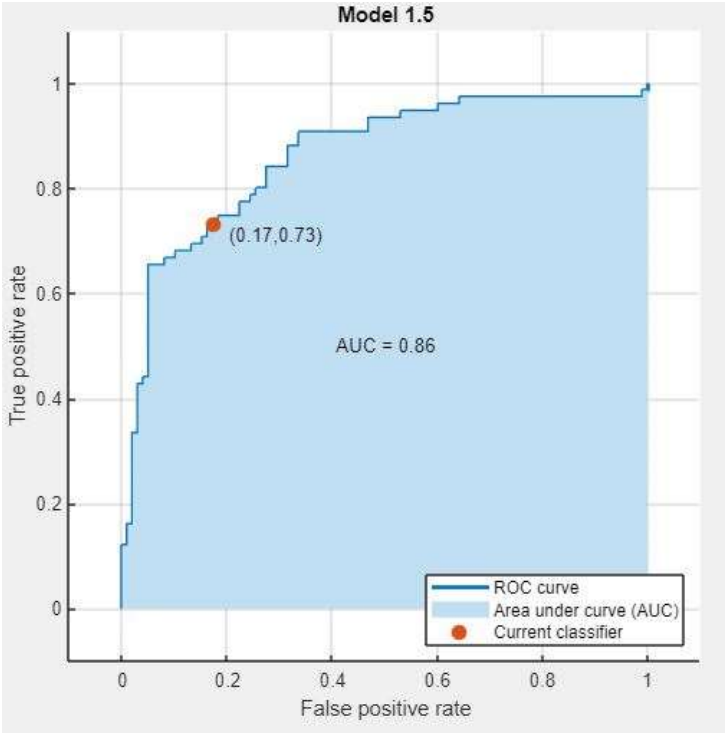


Figura 6-30. Curva ROC para la clase positiva Coche.

Fuente: MATLAB (elaboración propia).

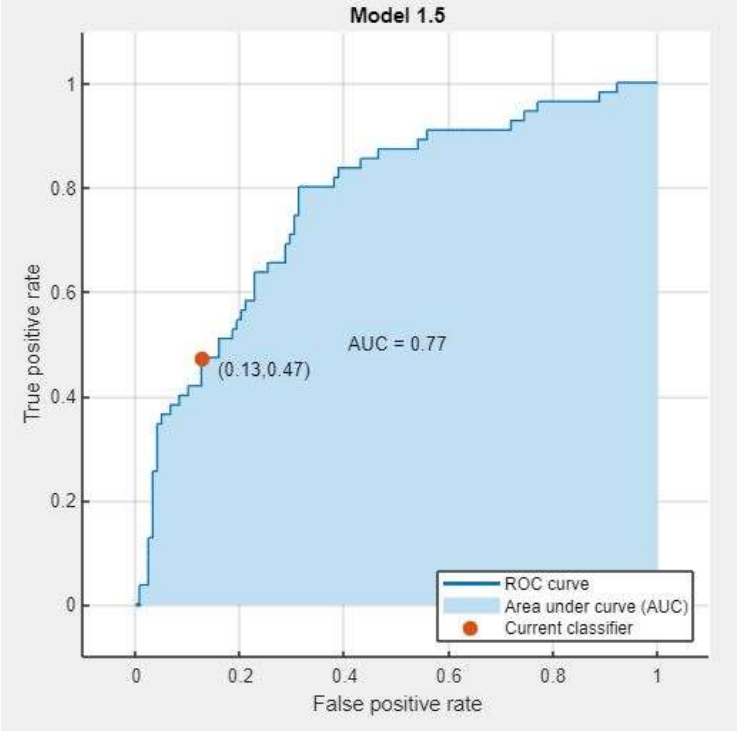


Figura 6-31. Curva ROC para la clase positiva Bus.

Fuente: MATLAB (elaboración propia).

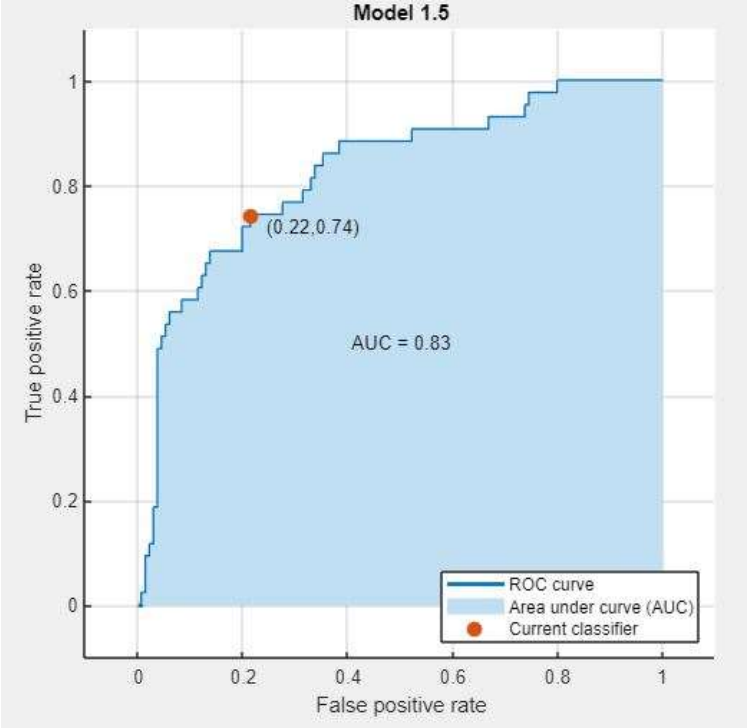


Figura 6-32. Curva ROC para la clase positiva Bici/Andando.

Fuente: MATLAB (elaboración propia).

6.4 Resultado con Modo de Transportes Únicos Mediante el Método Clásico

Durante el presente apartado, se desarrollará el proceso en el entorno del Mathematica, con las funciones de utilidad definidas en el apartado 3.1.2. Los coeficientes se han obtenido mediante la maximización de la log-verosimilitud de la muestra, utilizando las funciones de optimización no lineal integradas en el software. Se ha trabajado con el mismo conjunto de datos que en el apartado 6.2, por lo que se analizarán aquellos individuos que únicamente utilizaban uno de los siguientes medios de transporte:

- Coche
- Bicicleta/Andando (también se le llamará modo Activo)
- Autobús

Al tratarse del método clásico, el tratamiento de datos es más tedioso por lo que, de todas las variables existentes, se tendrán en cuenta las siguientes:

- Sexo
- Rol
- Disponibilidad de vehículo (exclusiva del coche)

Los coeficientes inferidos se recogen la figura 6.33

VARIABLE	Alt1	Alt2	Alt3
ASC	26.1409	30.8433	0
Sex	0.178078	-0.434946	0
Rol	-30.9582	-31.8492	0
Disposición	2.0277	0	0
tiempo	-0.0295776	-0.0295776	-0.0295776

Índice	
1	Coche
2	Bici/Andando
3	Bus

Figura 6-33. Valor asociado a los coeficientes.

Una vez que se obtienen esos valores, se calcula el valor de la función de utilidad descrita en el punto 3.1.2 para cada medio de transporte. Con esto, se obtienen la probabilidad de cada uno de los modos mediante las siguientes expresiones:

$$p_{Coche} = \frac{\exp(V_{Coche})}{\exp(V_{Coche}) + \exp(V_{Activo}) + \exp(V_{Bus})}$$

$$p_{Activo} = \frac{\exp(V_{Activo})}{\exp(V_{Coche}) + \exp(V_{Activo}) + \exp(V_{Bus})}$$

$$p_{Bus} = \frac{\exp(V_{Bus})}{\exp(V_{Coche}) + \exp(V_{Activo}) + \exp(V_{Bus})}$$

Aquel modo de transporte con la probabilidad más alta será el seleccionado por nuestro método. De esta forma, si coincide con la elección real del individuo significará que el modelo funciona bien para ese caso.

Se obtiene la siguiente matriz de confusión:

Clase real						
	Coche	72	2	5	72	7
	Bici Andando	9	3	18	3	27
	Bus	4	0	39	39	4
		Coche	Bici/Andando	Bus	TPR	FNR
		Predicción Clase				

Figura 6-34. Matriz de confusión numérica para el método clásico.

Fuente: elaboración propia.

Clase real						
	Coche	91.14%	2.53%	6.33%	91.14%	8.86%
	Bici Andando	30.00%	10.00%	60.00%	10.00%	90.00%
	Bus	9.30%	0.00%	90.70%	90.70%	9.30%
		Coche	Bici/Andando	Bus	TPR	FNR
		Predicción Clase				

Figura 6-35. Matriz de confusión porcentual para el método clásico.

Fuente: elaboración propia.

Se puede ver que la clase con mayor porcentaje de acierto es la del coche y que este modelo presenta cierto desequilibrio en lo que al porcentaje de aciertos y fallos se refiere. En términos generales, el número total de aciertos ha sido de 114 frente a 38 errores lo que implica un acierto del 75%.

7 CONCLUSIONES Y LÍNEAS FUTURAS

El contenido desarrollado en el presente trabajo persigue el estudio de diversos datos con los siguientes objetivos:

- Reunir la información relevante, desde una perspectiva de movilidad, para crear bases de datos que sirvan para modelar la distribución de patrones de viaje y realizar comparativas entre los resultados obtenidos tras aplicar diferentes modelos de reparto modal empleando aprendizaje automático y, por otra parte, el método clásico o tradicional.
- Estudio de la mejor toma de decisiones a la hora de depurar los datos, así como valorar qué información es la más representativa durante la fase de recopilación.

Toda la información se ha obtenido mediante una encuesta donde los individuos participantes comparten sus preferencias para realizar el desplazamiento a la universidad de ingeniería desde su domicilio. De esta forma, se generan una serie de variables de cada modo de transporte analizado (coche, autobús, bicicleta y andando). Como variables de entrada al modelo se emplearon diferentes variables socioeconómicas.

La totalidad de los datos recogidos han de ser tratados mediante la ayuda de programas informáticos como Excel o Matlab ya que el formato inicial no permite su tratamiento con softwares de aprendizaje automático y, por otro lado, es necesaria la depuración de estos para optimizar los diferentes modelos. Como añadido, se toman una serie de decisiones que permiten obtener dos bases de datos diferentes partiendo de la inicial. Este proceso es una parte importante del presente trabajo ya que es el proceso que más tiempo ha requerido y donde se aplicaron nociones más profundas de programación.

Una vez recogida toda la información, es el momento de aplicar algoritmos de aprendizaje automático mediante el software *Classification Learner* de Matlab y empleando el método Logit, que es lo que en este documento se ha denominado método tradicional o clásico.

Lo que se persigue es obtener diferentes resultados, centrándonos en la exactitud, la matriz de confusión y las curvas ROC para estudiar como varían en función del método empleado y de las decisiones tomadas en el tratamiento de la información. Por lo tanto, se estudian dos bases de datos, ambas mediante el método clásico y, por otra parte, aplicando aprendizaje automático..Se pueden sacar los siguientes puntos clave:

- Para el conjunto de datos del modo de transporte único, aparentemente, la máxima exactitud obtenida mediante aprendizaje automático, del 80%, es muy similar a la del método clásico, que es del 75%. Esto nos puede llevar a pensar que no hay gran diferencia entre uno y otro, pero no es correcto. En primer lugar, con el método clásico se han manejado muchas menos variables y, por otra parte, si observamos la matriz de confusión de ambos resultados, se puede ver que los resultados del método clásico están muy poco equilibrados por lo que su porcentaje de precisión es poco representativo.
- Mediante aprendizaje automático se pueden realizar cálculos mucho más rápidos que con los métodos clásicos, lo que permite trabajar con un volumen de datos mayor e ir tomando decisiones que permitan equilibrar la muestra de una forma más sencilla.
- Teniendo en cuenta exclusivamente los modelos obtenidos mediante aprendizaje automático, se puede observar que las precisiones obtenidas para el modo de transporte agrupados son algo menores que para el modo de transportes únicos. Esto se puede asociar a las decisiones que se han tomado a la hora de asignar un medio de transporte a aquellos individuos que empleaban dos o más, por lo tanto, sería conveniente estudiar otras alternativas.

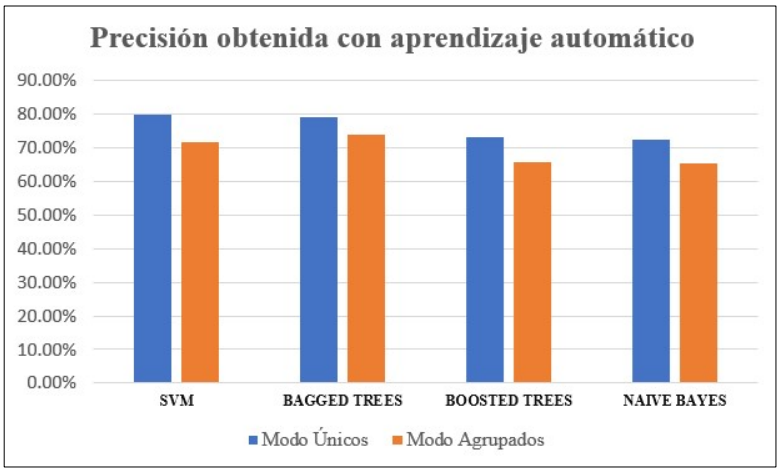


Figura 7-1. Valores de precisión de los métodos por aprendizaje automático.

Fuente: elaboración propia.

Por otra parte, en lo referente al estudio y recopilación de información se obtienen conclusiones y posibles acciones futuras desde un punto de vista de la movilidad que resultan de gran interés. Se exponen a continuación:

- Se echa en falta una categoría de clases que incluya otros medíos de transporte, como el metro, el patinete eléctrico (cada vez más empleado). Podría optarse por realizar una división entre transporte público, privado o peatonal y más adelante, una que especificase cada medio de transporte.
- Durante la fase de recopilación de datos sería conveniente limitar las respuestas, es decir, según el tipo de preguntas existentes, que se abra algún tipo de desplegable para no dar opción a la escritura libre, evitando así respuestas inconsistentes o carentes de sentido.
- Podría asignarse un porcentaje de uso a aquellos individuos que utilicen más de un medio de transporte.
- Estudiar la posibilidad de incorporar otro tipo de variables, como puede ser el estado civil (los individuos solteros suelen ser más jóvenes y puede ser que empleen más el transporte público y viceversa).
- Puede resultar interesante hacer una división entre transporte motorizado y el que no lo está, de esta manera se puede tener un indicativo interesante sobre la sostenibilidad.
- Aunque parezca algo evidente, del estudio de los datos recopilados se observa que la tendencia en individuos que realizan un solo viaje al día es emplear transporte público mientras que si el número de viajes es mayor el método elegido es el coche.
- El transporte en coche es el más habitual. Esto se manifiesta de forma más clara en lo relativo a viajes desde alguna zona del área metropolitana de Sevilla. La ubicación de la Escuela Técnica Superior de Ingenieros en un extremo de la ciudad de Sevilla puede ser determinante para este comportamiento en un entorno universitario.

Todo lo expuesto pretende formar parte de un futuro desarrollo de un modelo de reparto modal que pueda trabajar de la forma más exacta posible, aprendiendo de los datos que puedan ir integrándose y que optimice la forma de desplazarse.

REFERENCIAS

[1] A. A. Andrés Monzón, El transporte en España, Madrid: Fundación Encuentro, 2012.

[2] L. M. R. y. J. M. d. Castillo, Tema 5. Modelos de elección y reparto modal., Sevilla: Escuela Técnica Superior de Ingeniería.

[3] M. V. Jiménez, *Modelo de Regresión Logística,* Departamento de Estadística e Investigación Operativa: Universidad de Granada.

[4] E. Bermejo y A. Martínez, «Raona,» 2017. [En línea]. Available: https://www.raona.com/machine-learning-whitepaper/.

[5] TELEFONICA TECH, «PathsTalks 3x15: Inteligencia Artificial y Machine Learning,» 2017. [En línea]. Available: https://www.youtube.com/watch?v=M-Ou7p0_FcA.

[6] J. N. Kutz y S. Brunton, «Data-Driven Science and Engineering Machine Learning, dynamical systems, and control.,» 2019.

[7] S. S.-S. &. S. Ben-David, Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press.

[8] S. R. S. &. D. Landgrebe, *A Survey of a Decision Tree Classifier Methodology,* 1990.

[9] M. M. P. Escobar, Implementación de un modelo para predecir la resistencia a carbapenémicos en klebsiella pneumoniae mediante un algoritmo de machine learning, Universidad de las Fuerzas Armadas ESPE. Carrera de Ingeniería en Biotecnología, 2021.

[10] J. A. Rodrigo, «Cienciadedatos.net,» Octubre 2020. [En línea]. Available: https://www.cienciadedatos.net/documentos/py07_arboles_decision_python.html.

[11] L. O. H. K. W. B. &. W. P. K. Robert E. Bnafield, «A Comparison of a Decision Tree Ensemble Creation Techniques,» 2007.

[12] F. S. Caparrini, «cs.us,» 26 Diciembre 2018. [En línea]. Available: http://www.cs.us.es/~fsancho/?e=106.

[13] T. G. Dietterrich, «cs.orst.edu,» Oregon State University, [En línea]. Available: http://wwwcsorstedu/"tgd.

[14] C. B. F. H. a. F. K. Michael R. Berthold, Guide to Intelligent Data Analysis: How to Intelligently Make, Springer Publishing Company, 2010.

[15] L. Breiman, *Bagging Predictors,* Berkeley: Uniersity of California, 1994.

[16] R. K. N. A. &. I. M. Jehad Ali, *Random Forest and Decision Trees,* 2012.

[17] «Amazon ¿Qué es el Boosting?,» [En línea]. Available: https://aws.amazon.com/es/what-is/boosting/. [Último acceso: 2022].

[18] R. Mendoza, «Boosting en el Modelo de Aprendizaje PAC,» *Institución Universitaria Politécnico Grancolombiano,* 2013.

[19] T. Mitchell, *Machine Learning,* 1997.

[20] M. P. Dr., «Multiclass Approaches for Support Vector Machine Based Land Cover Classification,» de *MapIndia Conference*, 2005.

[21] S. García, *Principios Básicos de Machine Learning,* Granada, España: Instituto Andaluz de Investigación en Data Science and Computational Intelligence.

[22] R. S. C. Sergio Cobo Pérez, *Estudio del modo de acceso de la comunidad universitaria a la ETSI,* Sevilla, 2018.

[23] C. Alonso, «Chema Alonso: Tus datos mueven el mundo y tú sin saberlo,» 2019. [En línea]. Available: https://www.youtube.com/watch?v=sGRJt_8mb5s.

ANEXO A: TABLAS EXCEL

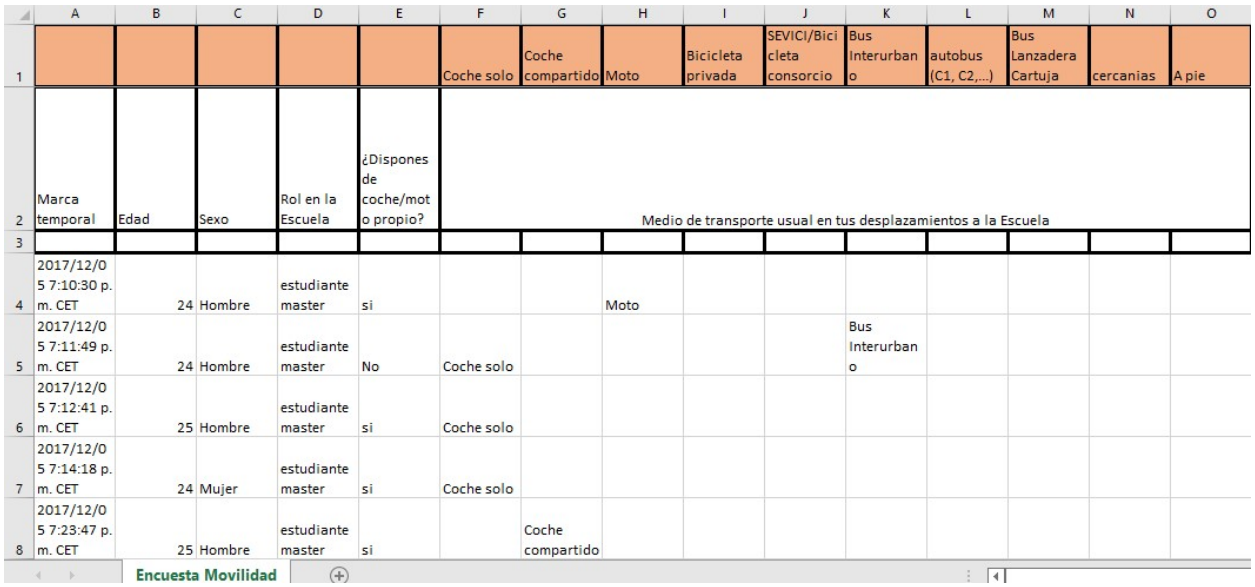
En este anexo quedan recogidas todas aquellas utilizadas a la hora de ejecutar los distintos modelos realizados en el presente trabajo, así como aquellas generadas después de la ejecución de estos.

En el título asignado a cada imagen, al igual que en todo el trabajo, se añade une breve descripción de la figura.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	Marca temporal,"Edad","Sexo","Rol en la Escuela","Â¿Dispones de coche/moto propio?","Medio de transporte usual en tus desplazamientos a la Escuela","CÃ³digo Postal/Barrio de origen habitual (si fuera de Sevilla, localidad)","[D																		
2	2017/12/05	MiÃ®rcoles Jueves		Viernes","1"															
3	2017/12/05	Bus Interurbano","41009","Calle Talgo NÃ·2","De 10 a 20 minutos","Distancia","De 12 a 15 horas","De Lunes a Viernes","1"																	
4	2017/12/05	MiÃ®rcoles Jueves		Viernes","1"															
5	2017/12/05	Martes	MiÃ®rcoles Jueves	Viernes","1"															
6	2017/12/05 7:23:47 p. m. CET,"25","Hombre","Estudiante mÃjster","SÃ·","Coche compartido","41927","","De 10 a 20 minutos","Comodidad","Antes de las 12 pm","De Lunes a Viernes","2"																	
7	2017/12/05 7:27:52 p. m. CET,"26","Hombre","Otros trabajadores","SÃ·","Coche solo","41927","","De 10 a 20 minutos","Comodidad","De 12 a 15 horas","De Lunes a Viernes","5"																	
8	2017/12/05 7:27:57 p. m. CET,"26","Hombre","Estudiante mÃjster","A veces","Bicicleta privada","41002","Cristo del Buen Fin n4","Menos de 10 minutos","Distancia","Antes de las 12 pm","De Lunes a Viernes","1"																	
9	2017/12/05	Lunes	MiÃ®rcoles","2"																
10	2017/12/05	A pie","41002","","De 20 a 30 minutos","Velocidad","De 12 a 15 horas","De Lunes a Viernes","4"																	
11	2017/12/05 7:33:16 p. m. CET,"25","Hombre","Estudiante mÃjster","A veces","Bicicleta privada","41001","","De 10 a 20 minutos","Comodidad","De 12 a 15 horas","De Lunes a Viernes","1"																	
12	2017/12/05 7:51:18 p. m. CET,"25","Mujer","Estudiante mÃjster","No","Coche compartido","41013","","De 10 a 20 minutos","Velocidad","De 12 a 15 horas","De Lunes a Viernes","1"																	
13	2017/12/05 7:58:10 p. m. CET,"25","Mujer","Estudiante grado","A veces","AutobÃ*s (C1, C2,...)","41003","","De 20 a 30 minutos","No tengo mÃjs remedio","Antes de las 12 pm","De Lunes a Viernes","1"																	
14	2017/12/05	Martes	MiÃ®rcoles","1"																
15	2017/12/05 8:02:40 p. m. CET,"24","Mujer","Estudiante grado","No","AutobÃ*s (C1, C2,...)","41013","","De 40 a 50 minutos","Distancia","Antes de las 12 pm","Lunes","1"																	
16	2017/12/05 8:02:50 p. m. CET,"24","Mujer","Estudiante grado","No","AutobÃ*s (C1, C2,...)","41013","","De 40 a 50 minutos","Distancia","Antes de las 12 pm","Lunes","1"																	
17	2017/12/05 8:04:31 p. m. CET,"25","Hombre","Estudiante mÃjster","No","A pie","41009","","De 20 a 30 minutos","Comodidad","De 12 a 15 horas","De Lunes a Viernes","1"																	
18	2017/12/05 8:09:37 p. m. CET,"20","Hombre","Estudiante grado","A veces","AutobÃ*s (C1, C2,...)","41003","","De 20 a 30 minutos","Velocidad","Antes de las 12 pm","De Lunes a Viernes","1"																	
19	2017/12/05 8:10:46 p. m. CET,"20","Mujer","Estudiante grado","No","A pie","41008","","De 10 a 20 minutos","EconomÃ·a","Antes de las 12 pm","De Lunes a Viernes","1"																	
20	2017/12/05	SEVICI/Bicicleta consorcio","41009","","De 10 a 20 minutos","Distancia","De 15 a 18 horas","De Lunes a Viernes","4"																	
21	2017/12/05 8:18:43 p. m. CET,"21","Hombre","Estudiante grado","SÃ·","Coche compartido","41020","Cueva de Menga 3","De 10 a 20 minutos","Comodidad","Antes de las 12 pm","De Lunes a Viernes","1"																	
22	2017/12/05	MiÃ®rcoles Jueves		Viernes","1"															
23	2017/12/05	A pie","41009","Calle Resolana, 24","De 10 a 20 minutos","Distancia","Antes de las 12 pm","De Lunes a Viernes","2"																	
24	2017/12/05	A pie","10001","Doctor maraÃ±on, 13","De 20 a 30 minutos","EconomÃ·a","Antes de las 12 pm","De Lunes a Viernes","1"																	
25	2017/12/05 9:01:26 p. m. CET,"20","Hombre","Estudiante grado","A veces","AutobÃ*s (C1, C2,...)","41007","C/Baltasar GraciÃjn","De 20 a 30 minutos","EconomÃ·a","Antes de las 12 pm","De Lunes a Viernes","5"																	
26	2017/12/05 9:03:31 p. m. CET,"20","Hombre","Estudiante grado","SÃ·","Coche solo","41004","","De 10 a 20 minutos","Comodidad","Antes de las 12 pm","De Lunes a Viernes","2"																	
27	2017/12/05	SEVICI/Bicicleta consorcio","41007","","De 30 a 40 minutos","EconomÃ·a","De 15 a 18 horas","De Lunes a Viernes","1"																	
28	2017/12/05 9:13:41 p. m. CET,"24","Hombre","Estudiante grado","A veces","Coche compartido","41907 / Valencina de la ConcepciÃ*n","","De 10 a 20 minutos","Comodidad","Antes de las 12 pm","De Lunes a Viernes","1"																	
29	2017/12/05 9:24:49 p. m. CET,"23","Mujer","Estudiante grado","No","A pie","41009","","De 20 a 30 minutos","EconomÃ·a","De 15 a 18 horas","De Lunes a Viernes","2"																	
30	2017/12/05 9:28:55 p. m. CET,"20","Hombre","Estudiante grado","No","AutobÃ*s (C1, C2,...)","41003","","De 20 a 30 minutos","Distancia","Antes de las 12 pm","De Lunes a Viernes","2"																	
31	2017/12/05	A pie","41009","Leon X","De 20 a 30 minutos","Comodidad","Antes de las 12 pm","De Lunes a Viernes","2"																	
32	2017/12/05	A pie","410(	MiÃ®rcoles Jueves	Viernes","1"															
33	2017/12/05 10:00:11 p. m. CET,"21","Hombre","Estudiante grado","No","A pie","41009","Calle Esmeralda","De 10 a 20 minutos","No disponibilidad de coche y tambiÃ®n por economÃ·a","Antes de las 12 pm","De Lunes a Viernes","1"																	

Figura 0-1. Fichero Excel en bruto generado a través de los datos recogidos en la encuesta.

Fuente: elaboración propia.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1						Coche solo	Coche compartido	Moto	Bicicleta privada	SEVICI/Bicicleta consorcio	Bus Interurbano	autobus (C1, C2,...)	Bus Lanzadera Cartuja	cercanias	A pie
2	Marca temporal	Edad	Sexo	Rol en la Escuela	¿Dispones de coche/moto propio?	Medio de transporte usual en tus desplazamientos a la Escuela									
3															
4	2017/12/05 7:10:30 p. m. CET	24	Hombre	estudiante master	si			Moto							
5	2017/12/05 7:11:49 p. m. CET	24	Hombre	estudiante master	No	Coche solo					Bus Interurbano				
6	2017/12/05 7:12:41 p. m. CET	25	Hombre	estudiante master	si	Coche solo									
7	2017/12/05 7:14:18 p. m. CET	24	Mujer	estudiante master	si	Coche solo									
8	2017/12/05 7:23:47 p. m. CET	25	Hombre	estudiante master	si		Coche compartido								

Encuesta Movilidad

Figura 0-2. Fichero Excel después de depurar y filtrar los datos

Fuente: elaboración propia.

A pie											
	Código Postal/Barrio de origen habitual (si fuera de Sevilla, localidad)	[OPCIONAL] Calle y número de origen	Tiempo de desplazamiento en el trayecto	Razón por la que usa ese medio de transporte	Tramo horario del desplazamiento	Días de desplazamiento a la ETSI (solo ida) [Seleccionar días]					Número de desplazamientos por día a la ETSI
						Lunes	Martes	Miercoles	Jueves	Viernes	
	41010		Menos de 10 minutos	Velocidad	De 12 a 15 horas		Martes	Miercoles	Jueves	Viernes	1
	41009	Calle Talgo 2	De 10 a 20 minutos	Distancia	De 12 a 15 horas	Lunes	Martes	Miercoles	Jueves	Viernes	1
	41703		De 30 a 40 minutos	Comodidad	De 12 a 15 horas		Martes	Miercoles	Jueves	Viernes	1
	41015	Estrella canopus	De 10 a 20 minutos	Velocidad	De 12 a 15 horas	Lunes	Martes	Miercoles	Jueves	Viernes	1
	41927		De 10 a 20 minutos	Comodidad	Antes de las 12 pm	Lunes	Martes	Miercoles	Jueves	Viernes	2

Figura 0-3. Detalle fichero Excel después de depurar y filtrar los datos.

Fuente: elaboración propia.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1													
2													
3				Generados a través de Python									Marca temporal
4	address	LAT	LON	LAT,LON	KM_bicy	TIME_bicy	KM_driving	TIME_driving	KM_walking	TIME_walking	KM_bus	TIME_bus	
5	Calle Talgo numero 2 CP 41009	37.4060326	-5.9337767	37.4060326,-5.9337767	2,8 km	8 min	4,3 km	7 min	2,0 km	25 min			2017/12/05 7:11:43 p. m. CET
6	Calle Estrella Canopus numero 1 CP 41015	37.4224524	-5.9716094	37.4224524,-5.9716094	3,8 km	13 min	5,3 km	10 min	3,8 km	47 min			2017/12/05 7:14:18 p. m. CET
7	Calle cristo del buen fin numero 4 CP 41002	37.3981752	-5.9991335	37.3981752,-5.9991335	2,5 km	8 min	2,8 km	9 min	2,4 km	30 min			2017/12/05 7:27:57 p. m. CET
8	Avenida Italia numero 17 CP 41012	37.35051	-5.982	37.35051,-5.982	8,8 km	27 min	12,6 km	17 min	8,0 km	1h 40 min			2017/12/05 7:30:20 p. m. CET
9	Calle cueva de menga numero 3 CP 41020	37.3993565	-5.9437029	37.3993565,-5.9437029	9,3 km	30 min	9,3 km	13 min	7,4 km	1h 34 min			2017/12/05 8:18:43 p. m. CET
10	Calle baltasar gracian numero 1 CP 41007	37.39052	-5.97147	37.39052,-5.97147	6,0 km	19 min	8,2 km	18 min	4,9 km	1h 1 min			2017/12/05 9:01:26 p. m. CET
11	calle hermenegildo casas jimenez numero 1 CP 41020	37.3931767	-5.9230634	37.3931767,-5.9230634	11,0 km	35 min	12,5 km	18 min	9,5 km	1h 59 min			2017/12/06 10:12:23 a. m. CET
12	Calle dolores ibarruri numero 8 CP 41806	37.364957	-6.1545777	37.364957,-6.1545777	18,9 km	58 min	20,3 km	18 min	18,4 km	3h 44 min			2017/12/06 12:33:45 p. m. CET
	Calle traviesa numero 9 CP 41015												2017/12/07 11:21:28 a

Figura 0-4. Fichero Excel generado mediante Python y API de Google Maps.

Fuente: elaboración propia.

	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF
1																				
2						Coche	Bicicleta privada	Interurbano	A pie											
3	Marca temporal	Edad	Sexo	Rol en la Escuela	¿Dispones de coche/mot	Medio de transporte usual en tus desplazamientos a la Escuela				Código Postal/Barrio de	[OPCIONAL] Calle y número de	Tiempo de desplazamiento en	Razón por la que usa ese medio	Tramo horario del	Días de desplazamiento a la ETSI (solo ida) [Seleccionar días]					Numero de desplaza
4															Lunes	Martes	Miercoles	Jueves	Viernes	
5	2017/12/05 7:11:49 p. m. CET	24	Hombre	estudiante master	No	SI		SI		41009	Calle Talgo 2	De 10 a 20 minutos	Distancia	De 12 a 15 horas	Lunes	Martes	Miercoles	Jueves	Viernes	1
6	2017/12/05 7:14:18 p. m. CET	24	Mujer	estudiante master	si	SI				41015	Estrella canopus	De 10 a 20 minutos	Velocidad	De 12 a 15 horas	Lunes	Martes	Miercoles	Jueves	Viernes	1
7	2017/12/05 7:27:57 p. m. CET	26	Hombre	estudiante master	A veces		SI			41002	Cristo del Buen Fin 4	Menos de 10 minutos	Distancia	Antes de las 12 pm	Lunes	Martes	Miercoles	Jueves	Viernes	1
8	2017/12/05 7:30:20 p. m. CET	25	Mujer	Estudiante grado	si	SI				41012	Avda italia 17	De 10 a 20 minutos	Distancia	Antes de las 12 pm	Lunes	Martes	Miercoles	Jueves	Viernes	2
9	2017/12/05 8:18:43 p. m. CET	21	Hombre	Estudiante grado	si	SI				41020	Cueva de Menga 3	De 10 a 20 minutos	Comodidad	Antes de las 12 pm	Lunes	Martes	Miercoles	Jueves	Viernes	1
10	2017/12/05 9:01:26 p. m. CET	20	Hombre	Estudiante grado	A veces			SI		41007	Calle Baltasar GraciÃ¡n	De 20 a 30 minutos	economia	Antes de las 12 pm	Lunes	Martes	Miercoles	Jueves	Viernes	5
11	2017/12/06 10:12:23 a. m. CET	22	Mujer	estudiante master	si	SI				41020	Hermenegildo	De 20 a 30 minutos	Comodidad	Antes de las 12 pm	Lunes	Martes	Miercoles	Jueves	Viernes	1

Figura 0-5. Fichero Excel generado mediante Python y API de Google Maps.

Fuente: elaboración propia.

	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Rol	Posesión	t1	t2	t3	V1	V2	V3	SumExp	P1	P2	P3	MaximaP	Acierto??
2	1	3	-28	-25	0	2.272051	-0.70141	0	11.19516	0.866381	0.044295	0.089324	1	1
3	1	2	-10	-11	0	-0.46612	-0.68055	0	2.13377	0.294047	0.237298	0.468654	3	0
4	1	3	-28	-18	0	2.272051	-0.90845	0	11.10242	0.873618	0.036312	0.09007	1	1
5	1	3	-49	-32	0	2.715102	-0.05942	0	17.04847	0.886071	0.055273	0.058656	1	1
6	1	2	-8	-7	0	-0.52528	-0.79886	0	2.041233	0.289722	0.220378	0.4899	3	1
7	1	3	-49	-32	0	2.89318	-0.49436	0	19.66059	0.918112	0.031024	0.050863	1	1
8	1	3	-72	-32	0	3.573465	-0.49436	0	37.24984	0.956779	0.016375	0.026846	1	1
9	0	3	-18	-15	0	32.7564	31.28696	0	2.07E+14	0.812971	0.187029	4.83E-15	1	1
10	1	3	-44	-30	0	2.567214	-0.11857	0	14.91767	0.873426	0.059539	0.067035	1	1
11	1	1	-33	-27	0	-1.63546	-0.64225	0	1.720969	0.113228	0.305704	0.581068	3	1
12	1	3	-27	-22	0	2.064395	-0.35519	0	9.581569	0.822468	0.073165	0.104367	1	1
13	1	2	-39	-32	0	0.391626	-0.05942	0	3.421699	0.432354	0.275394	0.292252	1	0
14	0	3	-43	-25	0	33.49584	31.58274	0	4.04E+14	0.871367	0.128633	2.47E-15	1	1
15	0	1	-15	-14	0	28.61226	31.25739	0	4.02E+13	0.06629	0.93371	2.48E-14	2	1
16	1	1	-18	-17	0	-2.2572	-0.50308	0	1.709308	0.061219	0.353748	0.585032	3	1
17	1	3	-59	-41	0	3.010878	0.206782	0	22.53494	0.901055	0.054569	0.044376	1	1
18	1	3	-20	-18	0	2.03543	-0.90845	0	9.058692	0.845105	0.044504	0.110391	1	1
19	1	3	-59	-38	0	3.010878	0.118049	0	22.43053	0.90525	0.050168	0.044582	1	1
20	1	3	-15	-13	0	1.887542	-1.05634	0	7.950845	0.830493	0.043735	0.125773	1	0
21	1	3	-20	-16	0	2.03543	-0.9676	0	9.035536	0.847271	0.042055	0.110674	1	1
22	1	1	-19	-16	0	-2.22763	-0.53266	0	1.694826	0.063596	0.346373	0.590031	3	1
23	1	2	-27	-25	0	0.036695	0.26646	0	2.803463	0.370034	0.273264	0.356702	1	0

VsLM | CoefLM | Matrix Confusion

Figura 0-6. Fichero Excel para cálculos de probabilidad del método clásico.

Fuente: elaboración propia.

ANEXO B: CÓDIGO PYTHON

En este anexo se presenta el código realizado para desarrollar los distintos modelos planteados. Como se indica anteriormente, el entorno de programación utilizado es Pycharm.

Se han añadido algunas notas o comentarios con la intención de facilitar su comprensión.

7.1 Código Depuración Inicial

```
#DEPURACION INICIAL

import pandas as pd    #importar la librería pandas con el nombre "pd" como es normal
import matplotlib.pyplot as plt
from matplotlib import pyplot

xls = pd.ExcelFile('encuestaV1.xlsx')
df = xls.parse('Encuesta Movilidad')   #lectura  del archivo excel indicando el nombre de la hoja
columns_name = df.columns.values  #con esta función se conocen los nombres de las columnas del excel
print(columns_name)

#Contabilizar los datos
num_mujer = len(df[df['sexo'] == 'Mujer'])
print(num_mujer)
num_hombre = len(df[df['sexo'] == 'Hombre'])

num_rol_grado = len (df[df['rol']=='Estudiante grado'])
num_rol_master = len (df[df['rol']=='estudiante master'])
num_rol_otro = len (df[df['rol']=='Otros trabajadores'])

num_si_vehiculo_propio = len(df[df['vehiculo_propio'] == 'si'])
num_no_vehiculo_propio = len(df[df['vehiculo_propio'] == 'No'])

num_coche_solo = len(df[df['coche'] == 'Coche solo'])
num_coche_compartido = len(df[df['coche_compartido'] == 'Coche compartido'])
num_moto = len(df[df['moto'] == 'Moto'])
num_bici_privada = len(df[df['bicicleta_privada'] == 'Bicicleta privada'])
num_sevici = len(df[df['sevici'] == 'SEVICI/Bicicleta consorcio'])
num_bus_inter = len(df[df['bus_interurbano'] == 'Bus Interurbano'])
num_bus_C1_C2 = len(df[df['bus_c1_c2'] == 'autobus (C1, C2,...)'])
num_bus_lanzadera = len(df[df['bus_lanzadera'] == 'Bus Lanzadera Cartuja'])
num_cercanias = len(df[df['cercanias'] == 'cercanias'])
num_andando = len(df[df['andando'] == 'A pie'])

num_tiempo_menos_10 = len(df[df['tiempo_desplazamiento'] == 'Menos de 10 minutos'])
num_tiemp0_10_20 = len(df[df['tiempo_desplazamiento'] == 'De 10 a 20 minutos'])
num_tiempo_20_30 = len(df[df['tiempo_desplazamiento'] == 'De 20 a 30 minutos'])
num_tiempo_30_40 = len(df[df['tiempo_desplazamiento'] == 'De 30 a 40 minutos'])
num_tiempo_40_50 = len(df[df['tiempo_desplazamiento'] == 'De 40 a 50 minutos'])
num_tiempo_50_60 = len(df[df['tiempo_desplazamiento'] == 'De 50 a 60 minutos'])
num_tiempo_mas_60 = len(df[df['tiempo_desplazamiento'] == 'mas de 60 minutos'])

num_razon_velocidad = len(df[df['razon'] == 'Velocidad'])
num_razon_distancia = len(df[df['razon'] == 'Comodidad'])
num_razon_economia = len(df[df['razon'] == 'Economia'])
```

```
num_tramo_antes_12 = len(df[df['tramo_horario'] == 'Antes de las 12'])
num_tramo_12_15 = len(df[df['tramo_horario'] == 'De 12 a 15 horas'])
num_tramo_15_18 = len(df[df['tramo_horario'] == 'De 15 a 18 horas'])
num_tramo_despues_18 = len(df[df['tramo_horario'] == 'Despuess de las 18 horas'])

num_lunes = len(df[df['Lunes'] == 'Lunes'])
num_martes = len(df[df['Martes'] == 'Martes'])
num_miercoles = len(df[df['Miercoles'] == 'Miercoles'])
num_jueves = len(df[df['Jueves'] == 'Jueves'])
num_viernes = len(df[df['Viernes'] == 'Viernes'])

#Media del número de desplazamientos
med_desplazamientos = df['num_dias'].mean()
print(med_desplazamientos)  #comprobación

#Se comprueban resultado
dias = ('Lunes', 'Martes', 'Miercoles', 'Jueves', 'Viernes')
slices = (num_lunes, num_martes, num_miercoles, num_jueves, num_viernes)
colores = ('red', 'blue', 'green', '#DD98AA','#18492D')
pyplot.pie (slices, colors=colores, labels = dias, autopct='%1.1f%%')
pyplot.axis ('equal')
pyplot.savefig('pastel_dias_semana')

medio_transporte = ('num_coche_solo','num_coche_compartido', 'num_moto',
'num_bici_privada','num_sevici','num_bus_inte',
            'num_bus_C1_C2','num_bus_lanzadera','num_cercanias','num_andando')
slices_2 = (num_coche_solo, num_coche_compartido, num_moto, num_bici_privada, num_sevici,
num_bus_inter,
        num_bus_C1_C2, num_bus_lanzadera, num_cercanias, num_cercanias)
pyplot.pie (slices_2, labels = medio_transporte, autopct= '%1.1f%%')
pyplot.axis ('equal')
```

7.2 Código Coordenadas

```
#GENERAR LATITUDES Y LONGITUDES A PARTIR DE DIRECCIONES

#usaremos la librería "pandas"
import pandas
df = pandas.read_csv('encuesta_calles_prueba.csv')
df

#llamada a google maps
import googlemaps
gmaps_key = googlemaps.Client(key = 'AIzaSyDsnm6tzm6ADfFEF_2lK4vCR0otlF2Q58c')
#Generar dataframe a partir del excel
df['LAT'] = None
df['LON'] = None

#recorrido en bucle de las diferentes direcciones para obtener sus coordenadas
for i in range (0, len(df), 1):
  geocode_result = gmaps_key.geocode(df.iat[i,0])

  try:
    lat = geocode_result [0]["geometry"]["location"]["lat"]
    lon = geocode_result[0]["geometry"]["location"]["lng"]
    df.iat[i, df.columns.get_loc("LAT")] = lat
    df.iat[i, df.columns.get_loc("LON")] = lon
```

```
    except:
        lat = None
        lon = None
print(df)

#generación del fichero excel que contiene las latitudes y las longitudes
#asociadas a las distintas direcciones
df.to_excel('final_calle_lat_lon.xlsx')
```

7.3 Código Tiempos y Distancias

```
#DISTANCIA Y TIEMPO CODE

#se genera un dataframe con las coordenadas
import pandas
df1 = pandas.read_excel('address_lat_lon_3.xlsx')
print(df1)

#se añaden columnas al dataframe vacías para rellenarlas con los datos que nos interesan
df1['KM_bicy'] = None
df1['TIME_bicy'] = None
df1['KM_driving'] = None
df1['TIME_driving'] = None
df1['KM_walking'] = None
df1['TIME_walking'] = None
print (len(df1))
import requests
import json

API_KEY = "AIzaSyDsnm6tzm6ADfFEF_2lK4vCR0otlF2Q58c"
dest_coord = "37.4108681,-6.0028718"

for i in range (0, len(df1), 1):
    #modo bici
    mode_bicy = "bicycling"
    orig_coord = str(df1["LAT,LON"][i])
    url_request_bicy =
"https://maps.googleapis.com/maps/api/distancematrix/json?units=metric&origins={0}&destinations={1}&mode={2}&language=es-ES&key={3}".format(orig_coord, dest_coord, mode_bicy, API_KEY)
    rsp_bicy = requests.get(url_request_bicy)
    a_bicy = rsp_bicy.json()
    data_string_bicy = json.dumps(a_bicy)
    jsondecoded_bicy = json.loads(data_string_bicy)
    print(jsondecoded_bicy)
    print(i)
    dist_bicy = str(jsondecoded_bicy["rows"][0]["elements"][0]["distance"]["text"])
    time_bicy = str(jsondecoded_bicy["rows"][0]["elements"][0]["duration"]["text"])
    df1.iat[i, df1.columns.get_loc("KM_bicy")] = dist_bicy
    df1.iat[i, df1.columns.get_loc("TIME_bicy")] = time_bicy

    #modo conducción
    mode_driving = "driving"
    url_request_driving =
"https://maps.googleapis.com/maps/api/distancematrix/json?units=metric&origins={0}&destinations={1}&mode={2}&language=es-ES&key={3}".format(orig_coord, dest_coord, mode_driving, API_KEY)
    rsp_driving = requests.get(url_request_driving)
    a_driving = rsp_driving.json()
    data_string_driving = json.dumps(a_driving)
    jsondecoded_driving = json.loads(data_string_driving)
```

```
    dist_driving = str(jsondecoded_driving["rows"][0]["elements"][0]["distance"]["text"])
    time_driving = str(jsondecoded_driving["rows"][0]["elements"][0]["duration"]["text"])
    df1.iat[i, df1.columns.get_loc("KM_driving")] = dist_driving
    df1.iat[i, df1.columns.get_loc("TIME_driving")] = time_driving

    #modo andando
    mode_walking = "walking"
    url_request_walking =
"https://maps.googleapis.com/maps/api/distancematrix/json?units=metric&origins={0}&destinatio
ns={1}&mode={2}&language=es-ES&key={3}".format(orig_coord, dest_coord, mode_walking, API_KEY)
    rsp_walking = requests.get(url_request_walking)
    a_walking = rsp_walking.json()
    data_string_walking = json.dumps(a_walking)
    jsondecoded_walking = json.loads(data_string_walking)
    dist_walking = str(jsondecoded_walking["rows"][0]["elements"][0]["distance"]["text"])
    time_walking = str(jsondecoded_walking["rows"][0]["elements"][0]["duration"]["text"])
    df1.iat[i, df1.columns.get_loc("KM_walking")] = dist_walking
    df1.iat[i, df1.columns.get_loc("TIME_walking")] = time_walking

df1.to_excel('Distancia_tiempo_all_tranport.xlsx')
```

7.4 Tratamiento de Datos de Transportes Únicos

```
U = readtable('Final_V5_coordenadas_tiempos_transporte_puros.xlsx');

data_puro = U (: , {'LAT', 'LON', 'KM_bicy', 'TIME_bicy_min', 'KM_driving', 'TIME_driving_min',
'KM_walking', 'TIME_walking_min', 'KM_bus', 'TIME_bus_min', 'RANGOEdad', 'Sexo', 'RolEnLaEscuela',
'x_DisponesDeCoche_motoPropio_', 'C_digoPostal_BarrioDeOrigenHabitual_siFueraDeSevilla_Localidad_',
'TiempoDeDesplazamientoEnElTrayecto', 'TramoHorarioDelDesplazamiento',
'N_meroDeDesplazamientosPorD__aALaETSI', 'TotalDiasSem', 'CODIGORESULTADOMedioTransp'});

data_puro.RANGOEdad = categorical (data_puro.RANGOEdad);

data_puro.Sexo = categorical (data_puro.Sexo);

data_puro.RolEnLaEscuela= categorical (data_puro.RolEnLaEscuela);

data_puro.x_DisponesDeCoche_motoPropio_ = categorical (data_puro.x_DisponesDeCoche_motoPropio_);

data_puro.TramoHorarioDelDesplazamiento= categorical (data_puro.TramoHorarioDelDesplazamiento);

data_puro.N_meroDeDesplazamientosPorD_aALaETSI = categorical
(data_puro.N_meroDeDesplazamientosPorD__aALaETSI);

data_puro.TotalDiasSem = categorical (data_puro.TotalDiasSem);

data_puro.LAT = str2double (data_puro.LAT);

data_puro.LAT = -1 * data_puro.LAT;

data_puro.LON = str2double (data_puro.LON);

data_puro.LON = -1 * data_puro.LON;

DATOS_PUROS = data_puro;

DATOS_PUROS.CODIGORESULTADOMedioTransp =
string(DATOS_PUROS.CODIGORESULTADOMedioTransp);

DATOS_PUROS.CODIGORESULTADOMedioTransp
(DATOS_PUROS.CODIGORESULTADOMedioTransp == 'A00') = 'COCHE';
```

```
 DATOS_PUROS.CODIGORESULTADOMedioTransp
(DATOS_PUROS.CODIGORESULTADOMedioTransp == '0B0') = 'BICI_ANDANDO';

 DATOS_PUROS.CODIGORESULTADOMedioTransp
(DATOS_PUROS.CODIGORESULTADOMedioTransp == '00C') = 'BUS';
```

7.5 Tratamiento de Datos de Transportes Agrupados

```
U = readtable('Final_V5_coordenadas_tiempos_transporte_AGRUPADOS.xlsx');

data_agrupado = U (: , {'LAT', 'LON', 'KM_bicy', 'TIME_bicy_min', 'KM_driving', 'TIME_driving_min',
'KM_walking', 'TIME_walking_min', 'KM_bus', 'TIME_bus_min', 'RANGOEdad', 'Sexo', 'RolEnLaEscuela',
'x_DisponesDeCoche_motoPropio_', 'C_digoPostal_BarrioDeOrigenHabitual_siFueraDeSevilla_Localidad_',
'TiempoDeDesplazamientoEnElTrayecto', 'TramoHorarioDelDesplazamiento',
'N_meroDeDesplazamientosPorD__aALaETSI', 'TotalDiasSem', 'CODIGORESULTADOMedioTransp'});

data_agrupado.RANGOEdad = categorical (data_agrupado.RANGOEdad);

data_agrupado.Sexo = categorical (data_agrupado.Sexo);

data_agrupado.RolEnLaEscuela= categorical (data_agrupado.RolEnLaEscuela);

data_agrupado.x_DisponesDeCoche_motoPropio_=
categorical(data_agrupado.x_DisponesDeCoche_motoPropio_);

data_agrupado.TramoHorarioDelDesplazamiento=
categorical(data_agrupado.TramoHorarioDelDesplazamiento);

data_agrupado.N_meroDeDesplazamientosPorD_aALaETSI=
categorical(data_agrupado.N_meroDeDesplazamientosPorD__aALaETSI);

data_agrupado.TotalDiasSem = categorical (data_agrupado.TotalDiasSem);

data_agrupado.LAT = str2double (data_agrupado.LAT);

data_agrupado.LAT = -1 * data_agrupado.LAT;

data_agrupado.LON = str2double (data_agrupado.LON);

data_agrupado.LON = -1 * data_agrupado.LON;

DATOS_AGRUPADOS = data_agrupado;

DATOS_AGRUPADOS.CODIGORESULTADOMedioTransp =
string(DATOS_AGRUPADOS.CODIGORESULTADOMedioTransp);

DATOS_AGRUPADOS.CODIGORESULTADOMedioTransp
(DATOS_AGRUPADOS.CODIGORESULTADOMedioTransp == 'A00') = 'COCHE';

DATOS_AGRUPADOS.CODIGORESULTADOMedioTransp
(DATOS_AGRUPADOS.CODIGORESULTADOMedioTransp == '0B0') = 'BICI_ANDANDO';

DATOS_AGRUPADOS.CODIGORESULTADOMedioTransp
(DATOS_AGRUPADOS.CODIGORESULTADOMedioTransp == '00C') = 'BUS';
```