Máster en Data Science

Trabajo Fin de Máster

PREDICCIÓN DE OCUPACIÓN DE PARQUÍMETROS SEGÚN MODELOS PREDICTIVOS ESPACIO-TEMPORALES

Alumnos:

Eva Carbón
Emilio Delgado
Cintia García
Paloma Panadero
Pedro Sánchez

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

1.1. Objetivo

En este documento de trabajo fin de máster analizamos en profundidad el uso de técnicas de Big Data y Aprendizaje Automático para la predicción de porcentajes de ocupación de las zonas de aparcamiento reguladas por parquímetros de la ciudad de Seattle.

El resultado de este trabajo se puede aprovechar para la creación o mejora de aplicaciones móviles (apps) asociadas al uso de aparcamientos regulados por parquímetros.

Son muchas las apps para aparcar disponibles para smartphones pero el propio mercado de oferta y demanda ha eliminado competencia, algunas han ido desapareciendo, y otras tienen un ámbito de actuación restringido (sólo en algunos municipios y ciudades). Una de las funcionalidades más demandadas por los usuarios y por los operadores de aparcamiento a los desarrolladores de las apps es que puedan dar información de la situación de ocupación.

En el caso de la ciudad de Barcelona, la consultora *AIS Group* ha logrado desarrollar una app para informar a los conductores sobre las plazas disponibles en el momento de la conducción, sea ese momento presente o futuro, en modo predictivo. También como en nuestro caso, los aparcamientos objeto de predicción son aquellos de estacionamiento regulado [1].

Find & Pay es el nombre de otra app que se encuentra ahora mismo en fase de prueba y testeo. Es la mayor app de aparcamiento en Europa, con presencia en casi 600 ciudades de once países distintos, con más de 500 probadores en 31 ciudades europeas que testean, validan y mejoran su capacidad predictiva [2]. Esta app, desarrollada por *EasyPark*, utiliza algoritmos avanzados para procesar diversas fuentes de datos, incluidos datos de transacciones, datos de seguimiento de dispositivos, datos de sensores y datos de automóviles en circulación, entre otros [3].

También hemos encontrado que la app *OPnGO* utiliza modelos predictivos para ayudar a los conductores a encontrar plaza en las zonas de estacionamiento regulado en distintas ciudades de Francia, España, Bélgica, Luxemburgo y Brasil [4].

Por último, mencionar la app *Telpark* que permite hasta el pago de denuncias, como servicio adicional a los mencionados anteriormente. Sus servicios están ya consolidados en decenas de ciudades españolas y también utiliza los modelos predictivos para su funcionamiento [5].

1.2. Background del problema

Creemos en la bondad de este estudio y de su desarrollo futuro para ayudar a la población en general, debido a todos los **beneficios** que puede aportar el hecho de anticipar el conocimiento de las plazas libres en una determinada zona.

Uno de los beneficios más evidentes es el ahorro de tiempo para el propio conductor. La población pierde numerosos minutos de su vida buscando aparcamiento, dando vueltas a la misma manzana esperando que se libere una plaza. Ésto repercute negativamente en la vida de las personas, ya que deben prever un tiempo suplementario que perderán en buscar aparcamiento para poder llegar a la hora a su cita. Y llegar puntual sería el mejor de los casos, ya que los retrasos en las citas son frecuentes debido a las dificultades para aparcar. Algunas cifras a modo de ejemplo:

- un 30 % del volumen del tráfico del centro de las ciudades es causado por coches buscando aparcamiento
- en media un usuario pierde 20 minutos cada vez que busca aparcamiento
- el 32 % de las multas que se extienden en Madrid son por estacionamiento incorrecto [6]
- en la ciudad de Londres un conductor pierde de media 67 horas al año buscando aparcamiento [7]
- en EEUU la media es de 17 horas perdidas, lo que resulta en un montante económico de 345$ por persona, teniendo en cuenta el coste de las emisiones, gasolina y tiempo
- y concretamente en la ciudad de Seattle se pierden 58 horas al año en esta búsqueda, lo que monetariamente se traduce en 1.205$ por persona perdidos al año [8]

Otra ventaja asociada al uso de un predictor de ocupación en zonas reguladas por parquímetros es la reducción de contaminación. Una persona que no tiene a su disposición esta información daría vueltas por la zona deseada hasta encontrar aparcamiento, dando lugar a un gasto extra de gasolina y también un alto nivel de contaminación asociado, ya que precisamente cuando buscamos plazas libres conducimos en marchas cortas, que son las que más efectos contaminantes tienen. Por ilustrar este dato, en la ciudad alemana de Freiburg el 74 % del tráfico de la ciudad se debe a conductores buscando aparcamiento [9]. En Los Ángeles este nivel de tráfico llega al 30 % [10]. Las emisiones de gases de efecto invernadero se verían reducidas considerablemente si se consigue reducir el tiempo de búsqueda de aparcamiento. Los expertos en movilidad tienen incluso un nombre para este fenómeno: tráfico de agitación.

También altera el humor de las personas, la espera en general hace que nos pongamos más nerviosos y perjudica nuestra actividad cardíaca. La felicidad de la población se ve afectada por esta espera en la búsqueda de aparcamiento, generando además peleas entre conductores que se disputan una misma plaza. En este sentido, predecir la ocupación en una determinada zona hará que el conductor sepa si por ejemplo tiene que irse a otra zona colindante para aumentar sus posibilidades de aparcar más rápido, haciendo que no tenga que perder la paciencia en la zona con nivel de ocupación más alto. De media dos tercios de las personas que se ven obligadas a buscar aparcamiento confiesan sentirse estresadas en esos momentos [11].

Y no hay que olvidar el beneficio comercial, pues el hecho de que una zona suela tener problemas para poder aparcar ahuyenta a posibles compradores de acudir a esa zona a visitar los comercios

locales. Así, si se sabe con antelación la ocupación de una determinada área, será más fácil animar al consumidor a acudir a los comercios en ese área.

El transporte público existente en la ciudad también se vería beneficiado de la puesta a disposición del público de las predicciones sobre ocupación que vamos a exponer, ya que en el caso de que una zona urbana esté masivamente ocupada, el usuario podría tender a dejar aparcado el coche en casa y optar por los servicios públicos de transporte para llegar a su punto de destino.

Capítulo 2

Fuentes de Datos

En este capítulo presentamos las múltiples fuentes de datos que hemos considerado para el análisis.

Hemos comenzado buscando en Internet datasets públicos con datos de uso de parquímetros, concretamente sus tickets o transacciones. Aunque como preveíamos la disponibilidad pública de este tipo de información es muy escasa, el dataset elegido como fuente de datos para el TFM no ha sido el único que hemos encontrado. Hemos descartado por comparación en número de registros el uso de un dataset de la ciudad de Melbourne, porque su tamaño es mucho menor (más de 342 mil registros) y por estar limitado temporalmente a un único año [12]. Y también hemos descartado el uso de otro dataset con más registros porque la información de ocupación corresponde a la utilización de aparcamientos privados en la ciudad de Bath, y nuestro objetivo era obtener datos del uso de aparcamientos públicos, es decir, de parquímetros en la calle [13].

El dataset con transacciones de uso de parquímetros públicos que hemos seleccionado como fuente principal de datos de nuestro TFM lo hemos encontrado en un Github con la documentación publicada por Rex Thompson como proyecto final de sus estudios de Data Science en la Universidad de Washington en 2017, y cuyo objetivo de estudio es totalmente diferente del nuestro. El objetivo de Rex Thompson era el análisis de los registros de los parquímetros para calcular el dinero total recaudado por la ciudad de Seattle durante las horas en las que hay fijadas restricciones de aparcamiento [14].

Los datos en crudo originales recopilados en el Github mencionado pertenecen al departamento de transportes de Seattle, conocido como SDOT (the city of Seattle Deparment of Transportation), que indica en su web que pone a disposición pública esta información con el objetivo de animar a los desarrolladores a crear aplicaciones que puedan ayudar a los usuarios a encontrar aparcamiento más rápidamente y pasar menos tiempo circulando o en atascos.

SDOT indica que los parquímetros de Seattle operan de Lunes a Sábado entre las 8am y las 8pm, con límites de tiempo de uso que pueden variar entre las 2, 4 o 10 horas. Y en periodos de desplazamientos al trabajo por la mañana y por la tarde-noche no está permitido el aparcamiento en algunas calles principales del centro de la ciudad. Añade también que el cálculo de la ocupación (número de transacciones dividido por el número de plazas disponibles en un periodo) no reflejaría la situación real ya que hay vehículos que no pagan (por causas justificadas o no).

En el proyecto de Rex Thompson se utilizan dos datasets que publica SDOT mediante APIs:

- *Paid Parking information data* [15], que contiene un histórico de transacciones desde Enero de 2012 a Septiembre de 2017, y en el que cada registro contiene como variables de interés para nuestro proyecto las siguientes:
  - *TransactionId*: identificador único de la transacción realizada en el parquímetro
  - *TransactionDateTime*: fecha y hora de la transacción
  - *Duration_mins*: duración en minutos reservada para el aparcamiento
  - *ElementKey*: identificador del segmento de la calle donde se ubica el parquímetro
- *Parking Blockface information data* [16], que descarga un fichero pequeño llamado *'Blockface.csv'* que complementa al dataset anterior y contiene como variables de interés para nuestro proyecto las siguientes:
  - *ElementKey*: coincide con el dataset anterior
  - *ParkingSpaces*: el número de plazas de parking disponibles en el segmento de calle
  - *PaidParkingArea*: el barrio o distrito de la ciudad al que está asociado el segmento de calle

Hemos aprovechado el trabajo laborioso ya realizado y compartido por Rex Thompson de recogida, limpieza y consolidación de los datos del primer dataset de transacciones, ya que SDOT sólo permite consultas que obtienen como respuesta ficheros con información de un máximo de 7 días. El fichero global de transacciones consolidado por Rex Thompson para el periodo entre el 1 de Enero de 2012 y el 30 de Septiembre de 2017 tiene un tamaño de 5.32GB y más de 62 millones de registros, y se llama *'ParkingTransaction_20120101_20170930_cleaned.csv'* (**DATASET-1**). También aprovechamos la limpieza realizada sobre el segundo dataset y utilizamos el fichero disponible llamado *'Blockface_cleaned.csv'* (**DATASET-2**).

	TransactionId	TransactionDateTime	TransactionDate	timeStart	timeExpired	Duration_mins	Amount	PaymentMean	MeterCode	ElementKey
0	13968676	2012-01-01T22:07:59Z	2012-01-01	22:07	23:22	75	2.50	COINS	10015002	25706
1	13968818	2012-01-01T23:30:59Z	2012-01-01	23:30	01:30	120	4.00	CREDIT CARD	10023002	25710
2	13968824	2012-01-01T22:45:59Z	2012-01-01	22:45	00:45	120	4.00	CREDIT CARD	10096002	9357
3	13968660	2012-01-01T22:51:59Z	2012-01-01	22:51	00:51	120	4.00	CREDIT CARD	10210002	25718
4	13968821	2012-01-01T23:28:59Z	2012-01-01	23:28	00:38	70	2.25	CREDIT CARD	10223002	2789

Figura 2.1: Extracto de las primeras muestras del DATASET-1

	PayStationBlockfaceId	ElementKey	ParkingSpaces	PaidParkingArea	ParkingTimeLimitCategory	PeakHourStart1	PeakHourEnd1	PeakHourStart2
0	7644	70865	8.0	Pioneer Square	120.0	NaN	NaN	NaN
1	7645	69085	10.0	Pioneer Square	120.0	06:00:00	09:00:00	NaN
2	7646	8870	1.0	Pioneer Square	120.0	NaN	NaN	NaN
3	7647	36093	8.0	Pioneer Square	120.0	NaN	NaN	NaN
4	7648	88630	5.0	Pioneer Square	120.0	NaN	NaN	NaN

Figura 2.2: Extracto de las primeras muestras del DATASET-2

Para nuestro proyecto necesitábamos añadir a los dos datasets mencionados los datos geoespaciales de localización de los parquímetros. Por ello hemos buscado en Internet las localizaciones GPS con latitud y longitud asociadas a los parquímetros de la ciudad de Seattle y hemos encontrado que SDOT publica también esa información a través de una API [17] [18]. Hemos creado un notebook de Python llamado *'FD_SDOT_PayStations.ipynb'* para realizar las consultas a esa API y descargar la información en dos ficheros json (*'paystations_ids_1_1000.json'* y *'paystations_ids_1001_1800.json'*). Son necesarios dos ficheros debido a que la API fija un límite de respuesta de 1000 registros por consulta. En la segunda parte del mismo notebook seleccionamos los tres parámetros que nos interesan de los ficheros json:

- *ELMNTKEY*: identificador del segmento de calle que coincide con los 2 datasets de partida
- *SHAPE_LAT*: latitud de coordenadas GPS
- *SHAPE_LNG*: longitud de coordenadas GPS

Luego hemos unido los datos en un dataframe de Pandas calculando la media de las distintas coordenadas existentes para un mismo *element key* antes de escribirlo en un fichero csv llamado *'Coord_EK.csv'* que reutilizaremos como dataset en otros notebooks (**DATASET-3**). Como habíamos indicado anteriormente el identificador *element key* hace referencia a un segmento de calle, y dependiendo de la longitud del segmento podemos tener hasta 3 coordenadas distintas para un mismo *element key*, por eso calculando la media de los valores de coordenadas existentes para un mismo *element key* obtenemos las coordenadas asociadas al punto central del segmento de calle asociado al *element key*. Hemos asumido por simplicidad que un *element key* identifica a un único parquímetro.

	element_key	latitude	longitude
0	1001	47.602862	-122.334703
1	1002	47.602997	-122.334538
2	1005	47.603602	-122.335382
3	1006	47.603725	-122.335171
4	1009	47.605010	-122.336669

Figura 2.3: Extracto de las primeras muestras del DATASET-3

Asociado también a la ciudad de Seattle hemos encontrado en Kaggle [19] un dataset que contiene información meteorológica histórica desde 1948 hasta 2017 (**DATASET-4**: fichero *'seattle-Weather_1948-2017.csv'*) [20]. Cada registro de este dataset meteorológico contiene las siguientes variables de interés para nuestro proyecto:

- *DATE*: fecha de observación
- *PRCP*: cantidad de precipitación medida en pulgadas
- *TMAX*: temperatura máxima del día medida en grados Farenheit
- *TMIN*: temperatura mínima del día medida en grados Farenheit

	DATE	PRCP	TMAX	TMIN	RAIN
0	1948-01-01	0.47	51	42	True
1	1948-01-02	0.59	45	36	True
2	1948-01-03	0.42	45	35	True
3	1948-01-04	0.31	45	34	True
4	1948-01-05	0.17	45	32	True

Figura 2.4: Extracto de las primeras muestras del DATASET-4

También relacionado con la meteorología hemos encontrado un dataset con registros asociados a sensores de temperatura ubicados en la ciudad de Seattle que recogen datos de temperatura ambiente y del asfalto por minuto desde Marzo de 2014 hasta hoy (**DATASET-5**: fichero *'Road_Weather_Information_Stations.csv'* [21]).

	StationName	StationLocation	DateTime	RecordId	RoadSurfaceTemperature	AirTemperature
0	35thAveSW_SWMyrtleSt	(47.53918, -122.37658)	03/03/2014 12:42:00 PM	672560	53.88	53.88
1	35thAveSW_SWMyrtleSt	(47.53918, -122.37658)	03/03/2014 12:43:00 PM	672561	54.05	54.05
2	35thAveSW_SWMyrtleSt	(47.53918, -122.37658)	03/03/2014 12:44:00 PM	672562	54.21	54.21
3	35thAveSW_SWMyrtleSt	(47.53918, -122.37658)	03/03/2014 12:45:00 PM	672563	54.38	54.38
4	35thAveSW_SWMyrtleSt	(47.53918, -122.37658)	03/03/2014 12:46:00 PM	672564	54.54	54.54

Figura 2.5: Extracto de las primeras muestras del DATASET-5

A diferencia del dataset anterior en el que los datos son diarios, en este dataset se dispone de datos medidos cada minuto y recogidos en 10 estaciones con distintas ubicaciones. Hemos creado un notebook de Python llamado *'FD_Road_Weather_Information_Stations.ipynb'* que transforma el dataset original para poder combinarlo con el dataset de transacciones. Entre las transformaciones necesarias destacamos las siguientes:

- filtrado de las horas en el rango de horas hábiles de uso de los parquímetros y agregación por horas de las medidas por minuto
- conversión de medidas de grados Farenheit a Celsius
- cálculo de la estación meteorológica de medida más próxima a cada parquímetro utilizando la distancia Haversine que es la que se usa habitualmente para calcular distancias entre puntos ubicados con coordenadas GPS ya que tiene en cuenta la curvatura de la tierra. Generamos el fichero *'Coord_EK_stations.csv'* (**DATASET-6**)
- completado de la serie temporal realizando interpolación porque faltan datos para algunos días y horas que provocarían nulos indeseados en la combinación con el dataset de transacciones, creando el fichero *'RWIS_completed.csv'* (**DATASET-7**)

	element_key	latitude	longitude	station_closest
0	1001	47.602862	-122.334703	5
1	1002	47.602997	-122.334538	5
2	1005	47.603602	-122.335382	5
3	1006	47.603725	-122.335171	5
4	1009	47.605010	-122.336669	5

Figura 2.6: Extracto de las primeras muestras del DATASET-6

	station_closest	timestamp	air_temp	road_temp
0	5	2016-01-01 08:00:00	2.03	-2.63
1	5	2016-01-01 09:00:00	1.99	-2.27
2	5	2016-01-01 10:00:00	2.08	-0.89
3	5	2016-01-01 11:00:00	2.28	2.44
4	5	2016-01-01 12:00:00	2.57	4.87

Figura 2.7: Extracto de las primeras muestras del DATASET-7

En relación con las ubicaciones de los parquímetros en la ciudad de Seattle hemos buscado información sobre su proximidad a puntos de interés cultural o deportivo en la ciudad, ya que su ocupación puede estar condicionada por esa situación. En la web de datos públicos de la ciudad de Seattle hemos encontrado ambas informaciones. Por un lado con un dataset que ubica teatros, cines, museos, bibliotecas, galerías, clubs de música, etc [22] (**DATASET-8**: fichero *'Seattle_Cultural_Space_Inventory.csv'*).

	Name	Phone	URL	Square Feet Total	Neighborhood	Organization Type	Dominant Discipline	Year of Occupation	Rent vs Own	Age of Current Building	...	Stability Index (5=very stable, 1=very uncertain)
0	Bulldog News	(206) 632-6397	http://www.bulldognews.com/	500.0	University District	N	Literary	1985.0	R	1930.0	...	4.0
1	METHOD Gallery	(206) 769-1151	http://www.methodgallery.com/	800.0	Pioneer Square	Y	Visual	2013.0	R	1907.0	...	2.0
2	The Makery	(206) 954-3497	https://themakeryseattleblog.wordpress.com	500.0	Seward Park	N	Arts/Cultural Training or Education	2.0	R	1940.0	...	4.0
3	SEEDArts Studios	(206) 760-4286	http://www.seedseattle.org/seedarts-studios/	10200.0	Hillman City	Y	Studios	2014.0	R	1920.0	...	4.0
4	The Royal Room	(206) 906-9920	NaN	3000.0	Columbia City	N	Music	2011.0	R	1917.0	...	4.0

Figura 2.8: Extracto de las primeras muestras del DATASET-8

Y por otro lado con varios datasets que ubican instalaciones deportivas en la ciudad para practicar diferentes deportes [23]:

- baseball: fichero *'Baseball_Field.csv'* (**DATASET-9**)
- tenis: fichero *'Tennis_Court_Point.csv'* (**DATASET-10**)
- natación: fichero *'Swimming_Pools.csv'* (**DATASET-11**)
- baloncesto: fichero *'Basketball_Court_Point.csv'* (**DATASET-12**)
- futbol: fichero *'Soccer_Field.csv'* (**DATASET-13**)
- atletismo: fichero *'Track_Fields.csv'* (**DATASET-14**)

	PMAID	the_geom	RESERVED1	GIS_AREA	GIS_LENGTH	GIS_EDT_DT	BALLFIELD_	NAME	SPORT_TYPE	FIELD_SURF	...	FACILITY_N
0	422	MULTIPOLYGON (((-122.27259129399673 47.5260192...	NaN	77303.320339	1135.795831	10/21/2014 12:00:00 AM +0000	801	Rainier Beach	Baseball	Grass	...	Ballfield 01
1	391	MULTIPOLYGON (((-122.3019285062559 47.66868012...	NaN	37512.326072	749.471305	10/21/2014 12:00:00 AM +0000	801	Ravenna	Baseball	Grass	...	Ballfield 01
2	400	MULTIPOLYGON (((-122.31492060232524 47.5861955...	NaN	42033.954263	813.444406	10/21/2014 12:00:00 AM +0000	801	Beacon Hill	Baseball	Grass	...	Ballfield
3	292	MULTIPOLYGON (((-122.34217620405622 47.6687435...	NaN	63942.730131	1038.732456	10/21/2014 12:00:00 AM +0000	801	Lower Woodland	Baseball	Grass	...	Ballfield 06
4	361	MULTIPOLYGON (((-122.32555601354544 47.7200755...	NaN	38644.145115	797.354520	10/21/2014 12:00:00 AM +0000	801	Northacres	Baseball	Grass	...	Ballfield 01

Figura 2.9: Extracto de las primeras muestras del DATASET-9

	PMAID	the_geom	GIS_AREA	GIS_LENGTH	GIS_EDT_DT	SPORTCOURT	NAME	SPORT_TYPE	COURT_ID	FACILITY_I	FACILITY_N
0	322	POINT (-122.35535254024099 47.6312628217014 6)	7008.017763	353.107404	04/01/2015 12:00:00 AM +0000	0	NaN	Tennis	NaN	NaN	NaN
1	322	POINT (-122.35505634945147 47.6315881843275 1)	5919.709996	334.916999	04/01/2015 12:00:00 AM +0000	0	NaN	Tennis	NaN	NaN	NaN
2	488	POINT (-122.30446228709225 47.6765633271014 4)	1440.812667	151.848564	04/01/2015 12:00:00 AM +0000	0	NaN	Tennis	NaN	NaN	NaN
3	292	POINT (-122.34340864856209 47.669367127426725)	6562.129583	350.175016	04/01/2015 12:00:00 AM +0000	0	NaN	Tennis	NaN	NaN	NaN
4	292	POINT (-122.3431911505528 47.669035558387705)	6527.883015	349.449458	04/01/2015 12:00:00 AM +0000	0	NaN	Tennis	NaN	NaN	NaN

Figura 2.10: Extracto de las primeras muestras del DATASET-10

	the_geom	COORDINATO	ADDRESS	NAME	PHONE	OFFICIAL_N	INDOOR_OUT	FULL_NAME	POINT_X	POINT_Y	GIS_EDT_DT
0	POINT (-122.35795026785668 47.63626286559663)	Janet Wilson	1920 1st Ave West	Queen Anne Pool	386-4282	Queen Anne Pool	Indoor	Queen Anne Pool	1.264556e+06	235812.484537	11/30/1899 12:00:00 AM +0000
1	POINT (-122.30240182803597 47.606887600677034)	Kristen Schuler	500 23rd Ave	Evers Pool	684-4766	Evers Memorial Pool	Indoor	Medgar Evers Pool	1.278044e+06	224832.999934	11/30/1899 12:00:00 AM +0000
2	POINT (-122.27033751275907 47.52476641504047 4)	Donna Sammons	8825 Rainier Ave S	Rainier Beach Pool	386-1944	Rainier Beach Pool	Indoor	Rainier Beach Pool	1.285393e+06	194733.953177	11/30/1899 12:00:00 AM +0000
3	POINT (-122.36916224498255 47.52800132362959)	Nancy Eisner	2801 SW Thistle St	Southwest Pool	233-7295	Southwest Pool	Indoor	Southwest Pool	1.261005e+06	196385.281239	11/30/1899 12:00:00 AM +0000
4	POINT (-122.376161943172 47.677540264436196)	Angela Eddy	1471 NW 67th Street	Ballard Pool	684-4094	Ballard Pool	Indoor	Captain William R. Ballard Pool	1.260369e+06	250955.515603	11/30/1899 12:00:00 AM +0000

Figura 2.11: Extracto de las primeras muestras del DATASET-11

	GIS_LENGTH	GIS_AREA	PMAID	the_geom	GIS_EDT_DT	SPORTCOURT	NAME	SPORT_TYPE	COURT_ID	FACILITY_I	FACILITY_N
0	327.852253	6237.155112	114	POINT (-122.30849123459399 47.56944067743952)	04/01/2015 12:00:00 AM +0000	0	NaN	Basketball	NaN	NaN	NaN
1	321.136129	6197.019558	390	POINT (-122.30789666684107 47.600349509910195)	04/01/2015 12:00:00 AM +0000	0	NaN	Basketball	NaN	NaN	NaN
2	182.821568	1822.460392	382	POINT (-122.31416450879583 47.71649703913903)	04/01/2015 12:00:00 AM +0000	0	NaN	Basketball	NaN	NaN	NaN
3	238.815565	3452.310432	450	POINT (-122.36327480720777 47.56132920631473)	04/01/2015 12:00:00 AM +0000	0	NaN	Basketball	NaN	NaN	NaN
4	357.406488	7142.272128	458	POINT (-122.36979006956886 47.53334748473792)	04/01/2015 12:00:00 AM +0000	0	NaN	Basketball	NaN	NaN	NaN

Figura 2.12: Extracto de las primeras muestras del DATASET-12

	NAME	the_geom	ADDRESS	DIVISION	SOCCER	OVERLAPPIN	E_SURFACE	E_LIGHTS	PMAID	LOCID	AMWO_ID	RES1	RES2
0	Decatur EL	POINT (-122.28435384474307 47.68561364506833)	7711 43rd Ave NE	SSD	2	Y	Grass	No	NaN	NaN	NaN	NaN	NaN
1	East Queen Anne Playground	POINT (-122.35349681556443 47.636140546290584)	1912 Warren Ave N	Central	1	N	Grass	No	329.0	NaN	NaN	NaN	NaN
2	Pinehurst Playground	POINT (-122.31463878544118 47.71625503063297)	12029 14th Ave NE	North	1	N	Grass	No	382.0	NaN	NaN	NaN	NaN
3	Ravenna-Eckstein Park	POINT (-122.30540806725641 47.67716865127399)	NaN	North	1	N	Grass	No	488.0	NaN	NaN	NaN	NaN
4	Green Lake Park	POINT (-122.32766609679587 47.6794764516775 7)	7201 E Green Lake Way	North	1	Y	Grass	No	307.0	NaN	NaN	NaN	NaN

Figura 2.13: Extracto de las primeras muestras del DATASET-13

	ADDRESS	NAME	the_geom	DIVISION	TRACK	OVERLAPPIN	E_SURFACE	E_LIGHTS	PMAID	LOCID	AMWO_ID	RES1	RES2
0	3013 S Mt Baker Blvd	Franklin HS	POINT (-122.29541428116494 47.57676103339843 4)	SSD	1	Y	Synthetic	No	NaN	NaN	NaN	NaN	NaN
1	11051 34th Ave NE	Jane Addams	POINT (-122.29345154595929 47.70896175799827)	SSD	1	Y	Synthetic	Yes	NaN	NaN	NaN	NaN	NaN
2	550 Phiney Ave N	Woodland Park - Field 7	POINT (-122.3416118198449 47.66970572000668)	North	1	N	Synthetic	Yes	292.0	NaN	NaN	NaN	NaN
3	5511 15th Ave S	Cleveland Playfield	POINT (-122.31558010418237 47.5520865432037)	South	1	N	Grass	No	404.0	NaN	NaN	NaN	NaN
4	4432 35th Ave Sw	West Seattle Stadium	POINT (-122.37415252022458 47.56307498209784)	South	1	N	Grass	Yes	472.0	NaN	NaN	NaN	NaN

Figura 2.14: Extracto de las primeras muestras del DATASET-14

Hemos creado un notebook de Python llamado *'FD_Cultural_And_Sports_Points.ipynb'* para combinar estos 7 datasets con el **DATASET-3** y crear un nuevo dataset a partir de éste último. Con el notebook descargamos los distintos ficheros csv a partir de APIs, seleccionamos las variables de interés de cada dataset, realizamos una pequeña limpieza y combinamos los registros con el **DATASET-3** para calcular la distancia Haversine entre los parquímetros y los puntos de interés (culturales y deportivos). En el nuevo dataset contenido en el fichero *'Coord_cult_&_sport.csv'* (**DATASET-15**) hemos creado una nueva columna binaria por cada tipo de punto de interés en el que señalamos con un valor 1 aquellos parquímetros que tienen un punto de interés a una distancia inferior a 75 metros, y con un valor 0 al resto. Observamos que con esa distancia se descartan los datasets asociados a natación y atletismo porque no hay ningún parquímetro cerca.

	element_key	latitude	longitude	poi	baseball	tennis	basket	soccer
0	1001	47.602862	-122.334703	1	0	0	0	0
1	1002	47.602997	-122.334538	1	0	0	0	0
2	1005	47.603602	-122.335382	0	0	0	0	0
3	1006	47.603725	-122.335171	0	0	0	0	0
4	1009	47.605010	-122.336669	1	0	0	0	0

Figura 2.15: Extracto de las primeras muestras del DATASET-15

Además también hemos buscado información sobre eventos de interés en la ciudad que pudieran influir en el uso de los parquímetros, y en la web de datos públicos de la ciudad de Seattle hemos encontrado un dataset con algunos eventos para varios meses del año 2016 [24] (fichero *'City_of_Seattle_Events.csv'*). Hemos creado un notebook de Python llamado *'FD_Eventos_Seattle_2016.ipynb'* para descargar el fichero a través de API y combinar esa información con un dataframe manual que hemos creado con otros eventos que hemos encontrado en internet de forma independiente. Este dataset (**DATASET-16**: fichero *'Events_2016.csv'*) se combinará con el dataset de transacciones para calcular la distancia Haversine e identificar por parquímetro y fecha las transacciones con un evento cerca y en ese día concreto.

	Latitude	Longitude	day_year
0	47.611543	-122.33263	105
1	47.611543	-122.33263	106
2	47.601130	-122.32980	137
3	47.628560	-122.33979	137
4	47.636290	-122.35922	137

Figura 2.16: Extracto de las primeras muestras del DATASET-16

Y por último hemos encontrado en internet un formulario para consultar información sobre la calidad del aire medida en las principales ciudades de Estados Unidos [25]. Hemos seleccionado la ciudad de Seattle, el año 2016 y los cuatro parámetros siguientes que son los considerados más relevantes para medir la polución del aire y que se relacionan con el motor y tubo de escape de los vehículos:

- monóxido de carbono (CO)
- dióxido de nitrógeno (NO_2)
- ozono (O_3)
- partículas en suspensión de 2,5 micrómetros o menos ($PM_{2,5}$)

Hemos combinado el resultado de las cuatro consultas en un dataset mediante un sencillo notebook llamado *'FD_Air_Quality_Data_Seattle_conversion.ipynb'* en el que además hemos unificado la unidad de medida de las tres primeras variables como $\mu g/m^3$ a partir de fórmulas encontradas en internet [26] (**DATASET-17**: fichero *'Air_Quality_Data_Seattle_2016.csv'*).

	day_year	no2	co	pm2_5	o3
0	1	75.262667	858.75	26.685000	61.0
1	2	71.001333	1030.50	19.875000	83.0
2	3	72.568000	801.50	14.281250	62.0
3	4	64.860000	629.75	11.047368	49.0
4	5	70.938667	1030.50	14.912500	45.0

Figura 2.17: Extracto de las primeras muestras del DATASET-17

En resumen, hemos recopilado diversas fuentes de datos que podemos dividir en dos grupos. Los tres primeros datasets son los necesarios para construir una serie espacio-temporal del porcentaje de ocupación de plazas de parking. Y los datasets restantes son complementarios para añadir a esa serie variables adicionales que pueden tener influencia en el porcentaje de ocupación mencionado y por tanto ayudar a su predicción futura.

Capítulo 3

ETL de los Datos

[27] Hemos creado un notebook de Python llamado *'ETL_Seattle_serie_2016.ipynb'* para realizar las tareas de preprocesamiento de las fuentes de datos que podríamos asimilar a los procesos de Extracción, Transformación y Carga, en los que se extraen datos desde múltiples fuentes, se limpian, manipulan o reformatean para luego cargarlos en este caso en otro fichero final que es el que se utilizará para crear los modelos de predicción.

Decidimos acotar el análisis de las fuentes de datos al año 2016 porque es el año más reciente para el que tenemos datos todos los meses en el dataset inicial de transacciones (**DATASET-1**) y para simplificar el tamaño del dataset ya que sólo con ese año tiene casi 11 millones de registros.

En el notebook mencionado realizamos las siguientes acciones destacables sobre el **DATASET-1** (transacciones):

- Extracción del dataset del fichero csv origen a un dataframe de Pandas, filtrado de las transacciones correspondientes al año 2016 y creación de una nueva variable llamada *final_date_time* a partir de las columnas *transaction_date_time* y *duration_mins*.
- Eliminación de transacciones con duración incorrecta (negativa o nula) que son menos de un 0,1 % del total.
- Transformación de las transacciones con distinto día de inicio y fin que son un poco más de un 0,1 % del total. Como para el análisis de ocupación sólo hay que tener en cuenta el rango de operación de los parquímetros (8-20h), es necesario duplicar las transacciones de larga duración para tener en cuenta los dos días, origen y final, de forma independiente, modificando sus fechas y horas para adaptarlas al rango de análisis. Así para la primera mitad de las transacciones duplicadas modificamos su fecha final para que coincida con la fecha inicial y su hora final a las 20h. Y para la segunda mitad de las transacciones duplicadas modificamos su fecha origen para que coincida con la fecha final y su hora origen a las 8h siempre que su hora final sea superior a esa hora, porque si es inferior borramos la segunda parte de la transacción duplicada.
- Borrado de las transacciones existentes con hora inicial y final inferior a las 8h o con hora inicial y final posterior a las 20h. Aunque no tienen sentido desde el punto de vista de uso de los parquímetros, existen en el dataset y es necesario borrarlas, suponiendo más de un 0,65 % del total.

- Transformación de las horas iniciales y finales de las transacciones al rango horario de funcionamiento de los parquímetros. Para su posterior agregación redondeamos las horas eliminando los minutos y segundos y modificamos a las 08:00h las que tienen una hora inicial inferior y a las 20:00h las que tienen una hora final superior.

- Borrado de las transacciones realizadas por error en domingos. Aunque también consideramos inicialmente la eliminación de los días festivos, finalmente no lo hicimos para facilitar la predicción por parte de los modelos considerando que la serie puede tener una estacionalidad de lunes a sábado.

Comprobamos que después de la extracción y transformación del dataset su tamaño se ha reducido en algo más de un 0.76 %.

Sobre el **DATASET-2** (segmentos de calle - capacidad de plazas y distritos) destacamos:

- Extracción del dataset del fichero csv origen a un dataframe de Pandas con más de 13.700 registros. En este dataset observamos que hay bastantes valores nulos y que para un mismo valor de *element_key* hay distintos valores en la columna *parking_spaces*, por lo que decidimos quedarnos con el máximo valor de capacidad de plazas agrupando por *element_key*. Adicionalmente encontramos dos parquímetros cada uno con dos valores distintos de barrio-distrito de la ciudad. Necesitaremos utilizar la información del tercer dataset para poder discriminar cuál es el valor más adecuado.

- Eliminación de aquellos parquímetros con valores de capacidad de plazas igual a cero (sólo 6). Comprobamos que después de la extracción y transformación del dataset su tamaño se ha reducido a 1703 registros.

Combinamos los primeros datasets para construir una serie espacio-temporal del porcentaje de ocupación de plazas de parking:

- Los tres primeros datasets comparten entre sí la variable *element_key*, habiendo 1514 valores distintos en el primer dataset, 1517 en el segundo y 1701 en el tercero. Al combinarlos entre sí en un nuevo dataframe observamos que disponemos de más de 10,6 millones de transacciones asociadas a 1443 parquímetros distintos.

- Creamos dos nuevos dataframes como copia del último generado, donde añadimos una nueva columna llamada *timestamp_sign*. En el primer dataframe consideramos sólo la fecha-hora (*timestamp*) de inicio de la transacción y la nueva columna toma el valor 1, y en el segundo dataframe consideramos sólo la fecha-hora de fin de la transacción y la nueva columna toma el valor -1. Uniendo ambos dataframes, la variable *timestamp_sign* nos ayudará a calcular el número de plazas ocupadas para cada parquímetro y hora.

- Agrupando el dataframe anterior por *element_key* y *timestamp*, creamos una nueva columna llamada *occupation* calculada como la suma acumulada de la columna *timestamp_sign*. Luego eliminamos duplicados y nos quedamos con el último registro que es el que contiene la suma acumulada total y por tanto contabiliza las transacciones de inicio y fin registradas en una misma franja horaria. Agrupando luego por *element_key* y *day_year* (ordinal del día del año asociado al *timestamp*), calculamos la suma acumulada total de la columna *occupation* para obtener el total de plazas ocupadas para ese día y hasta esa hora. Por último convertimos el valor absoluto de ocupación en porcentaje dividiendo por el total de plazas disponibles.

- Observamos que tenemos casi un 5 % de registros con un porcentaje de ocupación superior al 100 % y que además no corresponde a casos puntuales sino que casi el 82 % de los parquímetros tiene algún registro en esa situación. Una vez que revisamos que no hay ningún error en la generación de las cifras de ocupación acumuladas y que las transacciones realizadas en el mismo día y tramo horario se contabilizan correctamente, el problema sólo es atribuible a la cifra de plazas de parking disponibles.

Corrección de la capacidad de plazas de parking disponibles:

- Hemos encontrado en la web de SDOT otra API [28] que contiene el campo *ELMNTKEY* asociado a otros campos con información de plazas de aparcamiento, aunque no hemos conseguido localizar información explicativa al respecto. Hemos creado un notebook de Python aparte llamado *'ETL_SDOT_StreetParkingCategory.ipynb'* para realizar las consultas a esa API y descargar la información en varios ficheros json (como ya habíamos mencionado anteriormente todas las APIs de SDOT fijan un límite de respuesta de 1000 registros por consulta). De cada fichero json seleccionamos los parámetros que nos interesan y consolidamos los datos en un dataframe de Pandas que volcamos finalmente en un fichero csv llamado *'Street_Parking.csv'* (**DATASET-18**). Obtenemos más de 46 mil registros sin duplicidad de *element_key*. Para cada segmento de calle identificado por el *element_key* tenemos las siguientes variables: *parking_category*, *parking_spaces*, *total_zones*, *total_nopark* y *total_spaces*, donde la cifra de la columna *total_spaces* es la suma de las 3 variables anteriores.

- Eliminamos aquellos segmentos de calle con valores de capacidad de plazas igual a cero (97 registros).

- Comparamos los datos de la columna *parking_spaces* del nuevo dataset con el **DATASET-2** y encontramos que hay un 62 % de coincidencias, que es un valor alto teniendo en cuenta que en el **DATASET-2** teníamos valores diversos de capacidad para un mismo segmento de calle y habíamos seleccionado el valor máximo de los disponibles. Dado que hemos encontrado un nuevo valor de capacidad (la variable *total_spaces* que engloba a la que teníamos), decidimos utilizarla a pesar de no conocer el significado de las otras 2 variables y del sospechoso nombre de una de ellas (*total_zones* y *total_nopark*).

- Recalculamos los porcentajes de ocupación considerando los nuevos valores de capacidad de plazas disponibles y observamos en este caso que tenemos sólo un 0,3 % de registros con un porcentaje de ocupación superior al 100 % y que además corresponden a sólo 194 segmentos de calle, por lo que procedemos a eliminarlos de la serie manteniendo su elevado tamaño total.

- Adicionalmente decidimos acotar la serie teniendo en cuenta los valores de la variable *parking_category*. Menos de un 7 % de los parquímetros corresponden a categorías especiales (*No Parking Allowed*, *Restricted Parking Zone* o *Carpool Parking*) que pueden perjudicar el objetivo de generalización de la predicción de nuestro proyecto, por lo que decidimos quedarnos únicamente con la categoría mayoritaria.

Completamos la serie con las horas intermedias faltantes:

- Observamos que es necesario completar la serie porque hay muchos casos en los que no tenemos transacciones en el primer dataset durante alguna franja horaria. Por ejemplo, para un parquímetro podemos tener transacciones en la franja de las 12h (de 12:00 a 12:59) y no

tener transacciones nuevas hasta las 16h, por lo que al construir la serie nos faltan las franjas de las 13h, 14h y 15h que tendrán la misma ocupación que la franja de las 12h porque no ha habido transacciones en ese rango horario y por tanto no hay cambios.

- Observamos también que sólo 217 del total de 1110 parquímetros (menos del 20 %) tienen transacciones todos los hábiles del año. Pero no completamos esos casos porque en ese caso estaríamos falseando los datos.
- Obtenemos una serie espacio-temporal de más de 3,8 millones de registros.

Añadimos variables adicionales a la serie combinándola con el resto de datasets mencionados en el apartado anterior:

- Combinamos la serie con el **DATASET-4** (información meteorológica diaria) mediante la variable *day_year* añadiendo a la serie las columnas de cantidad de precipitación diaria, temperatura máxima y temperatura mínima diaria.
- Añadimos el **DATASET-5** (sensor de temperatura más próxima por parquímetro) y **DATASET-6** (serie de medidas por hora de los sensores de temperatura) a la serie espacio-temporal, combinando el primer dataset por la variable *element_key* y el segundo por las variables *timestamp* y *station-closest*.
- Añadimos el **DATASET-15** (indicadores booleanos de proximidad de puntos de interés cultural y deportivo a cada parquímetro) a la serie combinando los datasets por la variable *element_key*.
- Con el **DATASET-16** (lista de día y coordenadas de eventos) creamos en la serie una nueva columna booleana que indica para cada transacción si ese día hay un evento o no, y si dicho evento además está próximo al parquímetro de cada transacción. Y como en casos anteriores la proximidad está calculada con la distancia Haversine y definida por un valor inferior a 75 metros.
- Combinamos la serie con el **DATASET-17** (parámetros de calidad del aire) mediante la variable día del año (columna *day_year*).
- Exportamos la serie a un fichero csv llamado *'Serie_Total2016_ext.csv'*.

Y por último filtramos la serie global completa para seleccionar la información de la serie asociada a 30 parquímetros con los que realizamos la evaluación del mejor modelo de predicción. Para elegir los 30 parquímetros seleccionamos primero aquellos parquímetros con menos días del año sin transacciones. Y por otra parte seleccionamos también aquellos parquímetros que tienen un mayor número total de transacciones en el año. Con la intersección de esas dos selecciones obtenemos los 30 parquímetros elegidos para la evaluación de los modelos y exportamos sus transacciones al fichero csv llamado *'SerieTotal2016_ext_selected.csv'*.

Capítulo 4

Análisis Exploratorio de los Datos (EDA)

En este capítulo realizamos un análisis exploratorio (EDA) de la serie espacio-temporal construida en el capítulo anterior. El objetivo es estudiar el dataset en dos niveles, para encontrar sus características más relevantes y describir su estructura:

1. Análisis descriptivo estático, donde se estudian las covariables (variables que no son coordenadas espaciales o temporales y pueden describir o predecir el resultado), estableciendo relaciones entre ellas y extrayendo conclusiones con impacto en capítulos posteriores.

2. Análisis descriptivo dinámico, donde se analiza la estructura temporal y espacial de los datos, describiendo sus principales parámetros y características.

	day_year	element_key	hour	timestamp	occupation_perc	latitude	longitude	paid_parking_area	prcp	tmax
0	1	1001	8	2016-01-01 08:00:00	0.0	47.602862	-122.334703	Pioneer Square	0.0	7.78
1	1	1001	9	2016-01-01 09:00:00	0.0	47.602862	-122.334703	Pioneer Square	0.0	7.78
2	1	1001	10	2016-01-01 10:00:00	0.0	47.602862	-122.334703	Pioneer Square	0.0	7.78
3	1	1001	11	2016-01-01 11:00:00	0.0	47.602862	-122.334703	Pioneer Square	0.0	7.78
4	1	1001	12	2016-01-01 12:00:00	0.0	47.602862	-122.334703	Pioneer Square	0.0	7.78

	tmin	air_temp	road_temp	poi	baseball	tennis	basket	soccer	event	no2	co
0	-2.22	2.03	-2.63	1	0	0	0	0	0	75.262667	858.75
1	-2.22	1.99	-2.27	1	0	0	0	0	0	75.262667	858.75
2	-2.22	2.08	-0.89	1	0	0	0	0	0	75.262667	858.75
3	-2.22	2.28	2.44	1	0	0	0	0	0	75.262667	858.75
4	-2.22	2.57	4.87	1	0	0	0	0	0	75.262667	858.75

Figura 4.1: Extracto de las primeras muestras de la serie espacio-temporal

El conjunto de datos bajo análisis consta de 22 columnas y 4.182.480 observaciones. En la Figura 4.1 se muestra una captura de los cinco primeros registros. El dataset consta de dos columnas con coordenadas espaciales (*latitude* y *longitude*), una columna con coordenadas temporales (*timestamp*), la variable a predecir (el porcentaje de ocupación, columna *occupation_perc*), y 15 covariables.

4.1. Análisis descriptivo estático

La serie bajo estudio presenta las siguientes 9 covariables numéricas continuas:

- Temperatura máxima diaria
- Temperatura mínima diaria
- Cantidad de precipitación diaria
- Temperatura media horaria de la superficie del asfalto
- Temperatura media horaria ambiente
- Cantidad de dióxido de nitrógeno
- Cantidad de monóxido de carbono
- Cantidad de ozono
- Cantidad de partículas en suspensión de 2.5 micrómetros o menos

4.1.1. Temperatura máxima diaria

La temperatura máxima registrada durante los días en los que se ha producido una transacción sigue la distribución que se muestra en la Figura 4.2 que es aproximadamente una distribución normal centrada en la media y con desviación típica la de la muestra. A lo largo de los días que recoge el dataset, la temperatura máxima media es de 17.14^oC, mientras que su desviación típica es 7.06^oC.

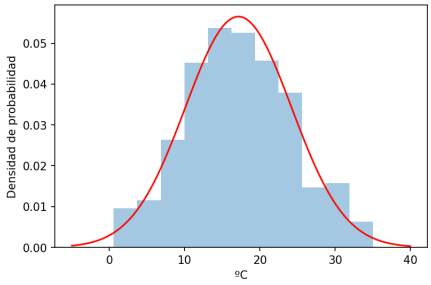

Figura 4.2: Distribución de temperaturas máximas

En la Tabla 4.1 se presentan los intervalos de confianza para la media y varianza de la temperatura máxima de la muestra ($\alpha = 0{,}05$). Como puede observarse, el intervalo de confianza para ambas medidas es muy estrecho, debiéndose al tamaño elevado de la muestra.

Parámetro	2.5 %	97.5 %
Media (°C)	17.136	17.150
Desv. Est. (°C)	7.050	7.060

Tabla 4.1: Intervalo de confianza para la temperatura máxima

4.1.2. Temperatura mínima diaria

Las mismas conclusiones que se han presentado sobre la temperatura máxima pueden realizarse sobre la temperatura mínima. La temperatura media mínima es 8.74^oC, mientras que su desviación típica es 4.54^oC. En la Figura 4.3 se muestra la distribución estadística y en la Tabla 4.2 los intervalos de confianza para cada parámetro, calculados para $\alpha = 0{,}05$.

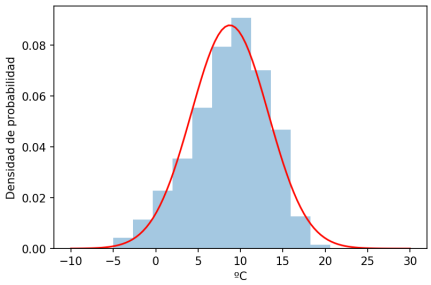

Figura 4.3: Distribución de temperaturas mínimas

Parámetro	2.5 %	97.5 %
Media (°C)	8.731	8.740
Desv. Est. (°C)	4.536	4.542

Tabla 4.2: Intervalo de confianza para la temperatura mínima

De nuevo, debido al tamaño de la muestra, los valores de media y desviación típica calculados son muy precisos. Además, también puede asumirse que la distribución es normal. La normalidad tanto de temperatura máxima como de temperatura mínima puede ayudar con el desarrollo de los modelos predictivos posteriores, debido a que muchas veces exigen normalidad en las variables que se utilizan para la predicción.

4.1.3. Precipitaciones

La distribución de precipitaciones se muestra en la Figura 4.4, y los intervalos de confianza para media y desviación típica en la Tabla 4.3. Presenta una media de 0.13 pulgadas y una desviación típica muestral de 0.26 pulgadas. Al igual que en secciones anteriores, los intervalos de confianza son estrechos, como corresponde a una muestra de un gran número de datos. Sin embargo, la distribución en este caso no es gaussiana, principalmente debido a no ser simétrica.

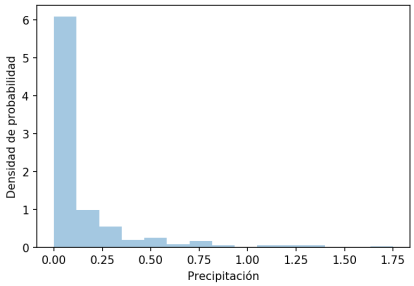

Figura 4.4: Distribución de precipitaciones

Parámetro	2.5 %	97.5 %
Media	0.1273	0.1278
Desv. Est.	0.2557	0.2561

Tabla 4.3: Intervalo de confianza para las precipitaciones

Este parámetro tiene mucha dispersión, pues su coeficiente de variación (C_V), calculado como

$$C_V = \frac{\sigma}{\bar{x}} \approx \frac{s}{\bar{x}},$$

da como resultado $C_V = 2$. En porcentaje, el coeficiente de variación es del 200 %, lo que implica que estamos ante una característica con gran variabilidad.

4.1.4. Temperatura media horaria del asfalto

La temperatura media horaria de la superficie del asfalto sigue la distribución que se muestra en la Figura 4.5. Su media es de 18.28^oC, mientras que su desviación típica es 9.16^oC.

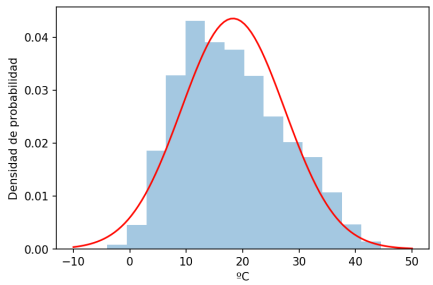

Figura 4.5: Distribución de temperatura media horaria del asfalto

En la Tabla 4.4 se presentan los intervalos de confianza para la media y varianza de la temperatura media del asfalto en la muestra ($\alpha = 0{,}05$). Como en las variables anteriores el intervalo de confianza para ambas medidas es muy estrecho, debiéndose al tamaño elevado de la muestra.

Parámetro	2.5 %	97.5 %
Media (°C)	18.265	18.284
Desv. Est. (°C)	9.154	9.167

Tabla 4.4: Intervalo de confianza para la temperatura media horaria del asfalto

4.1.5. Temperatura media horaria ambiente

La temperatura media horaria ambiente sigue la distribución que se muestra en la Figura 4.6. Su media es de $14.65°C$, mientras que su desviación típica es $6.39°C$. En la Tabla 4.5 se presentan los intervalos de confianza para la media y varianza de la temperatura media ambiente en la muestra ($\alpha = 0{,}05$).

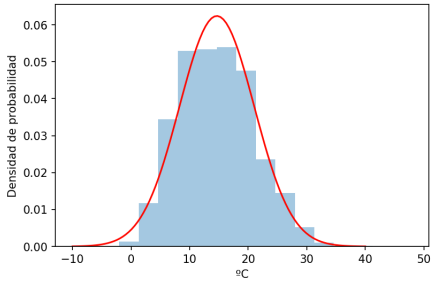

Figura 4.6: Distribución de temperatura media horaria ambiente

Parámetro	2.5 %	97.5 %
Media (°C)	14.646	14.659
Desv. Est. (°C)	6.389	6.398

Tabla 4.5: Intervalo de confianza para la temperatura media horaria ambiente

4.1.6. Dióxido de nitrógeno

La cantidad de dióxido de nitrógeno sigue la distribución que se muestra en la Figura 4.7. Su media es de 56.65 $\mu g/m^3$, mientras que su desviación típica es 14.79 $\mu g/m^3$.

En la Tabla 4.6 se presentan los intervalos de confianza para la media y varianza de la medida de dióxido de nitrógeno en la muestra ($\alpha = 0{,}05$).

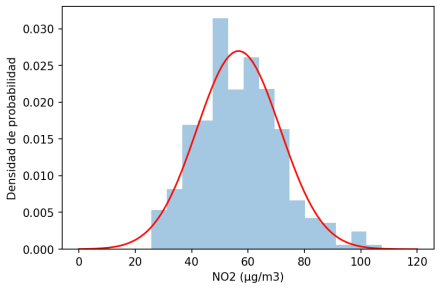

Figura 4.7: Distribución de cantidad de dióxido de nitrógeno

Parámetro	2.5 %	97.5 %
Media	56.633	56.663
Desv. Est.	14.779	14.800

Tabla 4.6: Intervalo de confianza para la cantidad de dióxido de nitrógeno

4.1.7. Monóxido de carbono

La cantidad de monóxido de carbono sigue la distribución que se muestra en la Figura 4.8, que a diferencia de la variable anterior no es gaussiana. Su media es de 512.55 $\mu g/m^3$, mientras que su desviación típica es 194.61 $\mu g/m^3$. En la Tabla 4.7 se presentan los intervalos de confianza para la media y varianza de la medida de monóxido de carbono en la muestra ($\alpha = 0{,}05$).

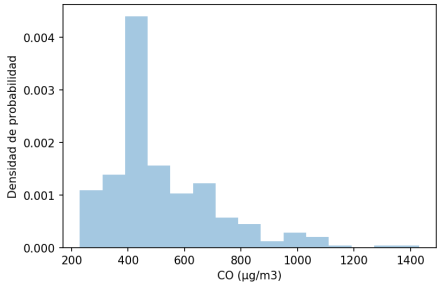

Figura 4.8: Distribución de cantidad de monóxido de carbono

Parámetro	2.5 %	97.5 %
Media	512.354	512.743
Desv. Est.	194.472	194.747

Tabla 4.7: Intervalo de confianza para la cantidad de monóxido de carbono

4.1.8. Ozono

La cantidad de ozono sigue la distribución que se muestra en la Figura 4.9, que es gaussiana. Su media es de 67.16 $\mu g/m^3$, mientras que su desviación típica es 16.31 $\mu g/m^3$. En la Tabla 4.8 se presentan los intervalos de confianza para la media y varianza de la medida de ozono en la muestra ($\alpha = 0{,}05$).

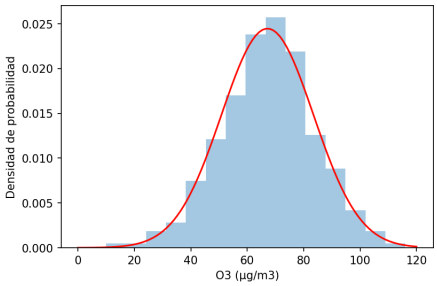

Figura 4.9: Distribución de cantidad de ozono

Parámetro	2.5 %	97.5 %
Media	67.142	67.174
Desv. Est.	16.294	16.317

Tabla 4.8: Intervalo de confianza para la cantidad de ozono

4.1.9. Partículas en suspensión

La cantidad de partículas en suspensión de tamaño inferior a 2.5 micras sigue la distribución que se muestra en la Figura 4.10, que no es gaussiana. Su media es de 5.63 $\mu g/m^3$, mientras que su desviación típica es 2.95 $\mu g/m^3$. En la Tabla 4.9 se presentan los intervalos de confianza para la media y varianza de la medida de partículas en suspensión en la muestra ($\alpha = 0{,}05$).

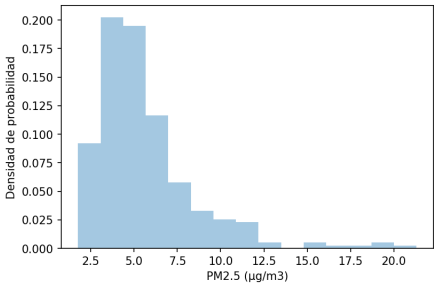

Figura 4.10: Distribución de cantidad de partículas en suspensión

Parámetro	2.5 %	97.5 %
Media	5.623	5.629
Desv. Est.	2.945	2.950

Tabla 4.9: Intervalo de confianza para la cantidad de partículas en suspensión

4.1.10. Porcentaje de ocupación

Finalmente, se presenta la distribución del porcentaje de ocupación de los parquímetros, tanto gráficamente (Figura 4.11) como con los intervalos de confianza para un 5 % de significación (Tabla 4.10). La media del porcentaje de ocupación es de 18.45 %, mientras que la desviación típica es 16.10 %.

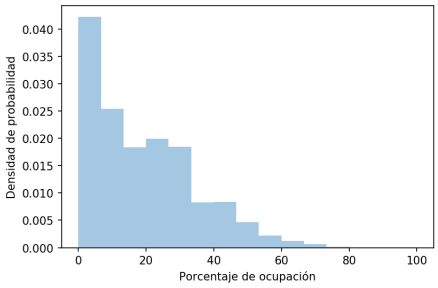

Figura 4.11: Distribución del porcentaje de ocupación medio de los parquímetros

La distribución no es gaussiana y está muy polarizada hacia los valores inferiores. Del mismo modo que en las variables anteriores, se puede comprobar que los intervalos de confianza son muy estrechos debido al gran número de muestras de que consta el dataset.

Parámetro	2.5 %	97.5 %
Media (%)	18.437	18.469
Desv. Est. (%)	16.086	16.108

Tabla 4.10: Intervalo de confianza para el porcentaje de ocupación

Por otro lado, tiene una varianza considerable, especialmente en relación con la media, como consecuencia de la variabilidad de la disponibilidad de plazas de aparcamiento. Sin embargo, la variabilidad de esta característica no es tan grande como habíamos visto para la variable de precipitaciones: su coeficiente de variación es del 87 %.

4.1.11. Análisis de correlaciones entre las covariables y el target

A continuación, se presenta un análisis de correlaciones entre las covariables y el target, que permitirá disponer de información más precisa y detallada sobre la serie. Además, debido a que muchos modelos espacio-temporales constan de una parte regresiva, se podrá utilizar el resultado obtenido para predecir con mejor precisión.

Para el análisis de correlaciones, dado que todas las variables númericas son continuas, se utiliza el coeficiente de correlación de Pearson, definido como

$$r_{X_1X_2} = \frac{E[(X_1 - \bar{x}_1)(X_2 - \bar{x}_2)]}{s_{X_1} s_{X_2}}.$$

La significación estadística de este valor se estudia mediante un test T [43], que determina si el valor calculado es significativamente distinto de cero. Para ello, se calcula el estadístico T,

$$T = \frac{r}{\sqrt{1 - r^2}} \cdot \sqrt{N - 2},$$

que se distribuye según una t de Student de $N - 2$ grados de libertad. El p-valor se calcula de forma bilateral, utilizando las tablas de la t de Student, como la probabilidad de obtener un valor más extremo del estadístico T que se ha calculado con la muestra dada. Es decir,

$$p = \text{Prob}(|t| \geq |T|)$$

Establecemos el nivel de significación (α) en el 5 %, por lo que el p-valor deberá ser menor de 0.05 para que sea válido y el resultado tenga significación estadística.

Mantendremos la estructura original del test, tal y como está establecido en [42], pero debemos tener en cuenta que N es muy grande, y por lo tanto:

- La t de Student se podría aproximar a una distribución normal.
- Los resultados saldrán muy significativos, pues el valor de T será muy elevado, situándose muy a la derecha o muy a la izquierda de la distribución t, quedando muy lejos del valor crítico definido por $\alpha = 0{,}05$.

Los resultados se presentan en la Tabla 4.11, donde se muestra tanto el coeficiente de correlación de Pearson (r) como el p-valor asociado a cada uno de ellos. No hay evidencia de que haya correlación entre las covariables y el target. Además, esta conclusión estadísticamente es bastante significativa, pues todos los p-valores calculados son menores que el intervalo de significación establecido ($p < 0{,}05$).

Como se comentó anteriormente, todos los resultados son significativos porque la muestra es muy grande. Por eso, aunque los coeficientes de correlación están próximos a cero, son estadísticamente distintos de cero, lo que es lógico teniendo en cuenta el tamaño de la muestra.

También hemos analizado la correlación del porcentaje de ocupación con las variables espaciales (latitud y longitud) y las variables temporales (mes, día de la semana, día del año) y los coeficientes de correlación también están próximos a cero.

Correlación con el porcentaje de ocupación	r	p-valor
Temperatura máxima	0.005	2.79e-25
Temperatura mínima	0.006	2.34e-29
Precipitación	-0.006	5.23e-35
Temperatura asfalto horaria	0.027	0.0
Temperatura ambiente horaria	0.012	1.76e-120
Dióxido de nitrógeno	-0.004	6.3e-12
Monóxido de carbono	0.003	4.13e-10
Ozono	0.010	5.23e-93
Partículas en suspensión	-0.001	1.4e-02

Tabla 4.11: Correlaciones entre las covariables y el target

La conclusión, por lo tanto, es que no hay evidencias de un grado de correlación alto entre las covariables y el target. Esto podría dificultar el análisis de tipo regresivo, puesto que no hay relaciones lineales directas entre las variables presentadas y el porcentaje de ocupación. Este aspecto se tiene en cuenta a la hora de realizar la evaluación del mejor modelo de predicción espacio-temporal, pues algunos permiten incluir regresores.

4.1.12. Análisis de correlaciones mutuas entre las covariables

En esta sección, se repite el análisis anterior, pero para estudiar las posibles correlaciones entre cada una de las covariables. De esta forma, se analiza la posible existencia de multicolinealidad, que pudiera influir en la parte regresiva de algunos de los modelos espacio-temporales. Por otro lado, se puede determinar si existen variables que están tan relacionadas que en realidad pertenecen a la misma distribución, con lo que debe tenerse en cuenta para eliminar alguna de ellas. Los resultados del análisis de correlaciones mutuas entre las covariables se muestran en la Tabla 4.12 y en la Figura 4.12.

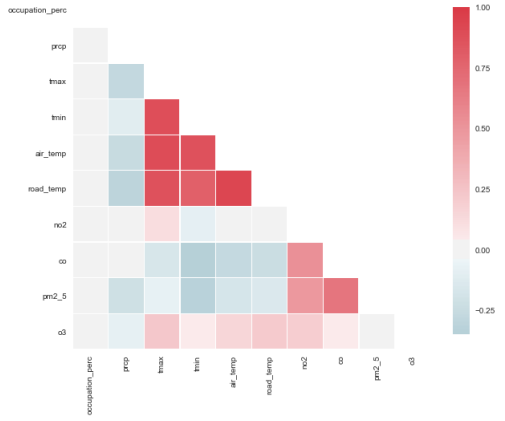

Figura 4.12: Matriz de correlación de las covariables

Correlación mutua	r	p-valor
Temperatura máxima - Temperatura mínima	**0.879**	0
Temperatura máxima - Precipitaciones	-0.273	0
Temperatura máxima - Temperatura asfalto	**0.873**	0
Temperatura máxima - Temperatura ambiente horaria	**0.901**	0
Temperatura máxima - Dióxido de nitrógeno	0.111	0
Temperatura máxima - Monóxido de carbono	-0.164	0
Temperatura máxima - Ozono	0.230	0
Temperatura máxima - Partículas en suspensión	-0.076	0
Temperatura mínima - Precipitaciones	-0.106	0
Temperatura mínima - Temperatura asfalto	**0.795**	0
Temperatura mínima - Temperatura ambiente horaria	**0.867**	0
Temperatura mínima - Dióxido de nitrógeno	0.086	0
Temperatura mínima - Monóxido de carbono	-0.347	0
Temperatura mínima - Ozono	0.047	0
Temperatura mínima - Partículas en suspensión	-0.319	0
Precipitaciones - Temperatura asfalto	-0.301	0
Precipitaciones - Temperatura ambiente horaria	-0.257	0
Precipitaciones - Dióxido de nitrógeno	-0.006	0
Precipitaciones - Monóxido de carbono	-0.025	0
Precipitaciones - Ozono	-0.082	0
Precipitaciones - Partículas en suspensión	-0.215	0
Temperatura asfalto - Temperatura ambiente horaria	**0.934**	0
Temperatura asfalto - Dióxido de nitrógeno	-0.012	0
Temperatura asfalto - Monóxido de carbono	-0.237	0
Temperatura asfalto - Ozono	0.215	0
Temperatura asfalto - Partículas en suspensión	-0.142	0
Temperatura ambiente horaria - Dióxido de nitrógeno	-0.034	0
Temperatura ambiente horaria - Monóxido de carbono	-0.267	0
Temperatura ambiente horaria - Ozono	0.149	0
Temperatura ambiente horaria - Partículas en suspensión	-0.173	0
Dióxido de nitrógeno - Monóxido de carbono	0.525	0
Dióxido de nitrógeno - Ozono	0.202	0
Dióxido de nitrógeno - Partículas en suspensión	0.487	0
Monóxido de carbono - Ozono	0.046	0
Monóxido de carbono - Partículas en suspensión	0.671	0
Ozono - Partículas en suspensión	0.018	0

Tabla 4.12: Correlaciones mutuas entre las covariables

De nuevo, debido al tamaño de la muestra, las conclusiones son muy significativas. Se puede observar que hay una correlación muy fuerte entre las cuatro variables de temperatura. Esta conclusión es lógica pues las cuatro variables forman parte de una misma información, si aumenta la temperatura media ambiente, por ejemplo en verano, suben tanto las temperaturas mínimas como las máximas (salvo casos extremos), y además con el mismo signo. Y la temperatura ambiente

afecta directamente a la temperatura del asfalto. Sin embargo, las cuatro variables no provienen de la misma distribución estadística, pues su media es claramente diferente (algo menos entre la temperatura máxima y la temperatura media ambiente), y la muestra es suficientemente grande. Esta afirmación se demuestra mediante la aplicación del test de Kolmogorov-Smirnov (KS), que analiza las diferencias entre las dos funciones de distribución que se están comparando. Se calcula el estadístico D, como

$$D = \text{máx}[F_1(x) - F_2(x)],$$

donde F_1 y F_2 son las funciones de distribución de las dos variables bajo comparación. El resultado obtenido es que las funciones de distribución de las cuatro variables difieren en los valores de D que se muestran en la Tabla 4.13, con una significación estadística altísima, de nuevo debido al tamaño de la muestra.

Variables	D	p-valor
Temperatura máxima - Temperatura mínima	0.54	0
Temperatura máxima - Temperatura media asfalto	0.11	0
Temperatura máxima - Temperatura media ambiente	0.15	0
Temperatura mínima - Temperatura media asfalto	0.54	0
Temperatura mínima - Temperatura media ambiente	0.43	0
Temperatura media asfalto - Temperatura media ambiente	0.21	0

Tabla 4.13: Test de Kolmogorov-Smirnov para las variables de temperatura

Los mismos resultados pueden observarse en la Figura 4.13, donde se aprecian que las diferencias máximas entre las funciones de distribución de la temperatura máxima, temperatura media ambiente y del asfalto son muy pequeñas (entre 0.1-0.2 aproximadamente).

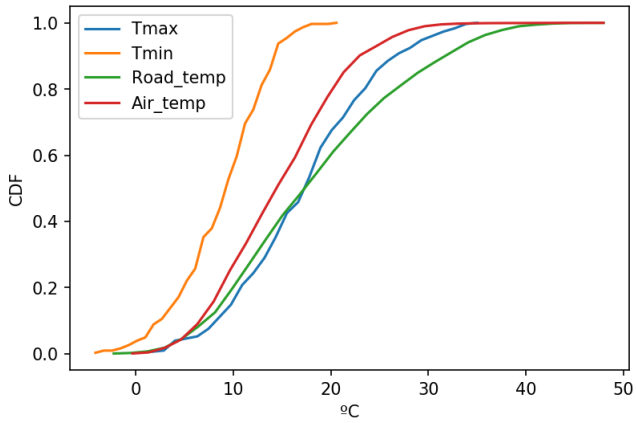

Figura 4.13: Funciones de distribución de las variables de temperatura

Este test se presenta únicamente como comprobación formal de lo que se comentó con anterioridad: la temperatura máxima y la mínima están muy relacionadas, pero no provienen de la misma

distribución, por lo que tendrán impactos diferentes en el porcentaje de ocupación. Por ejemplo, es posible que las temperaturas mínimas no afecten del mismo modo a los desplazamientos y ocupación de los parquímetros en Seattle que las temperaturas máximas.

4.2. Análisis descriptivo dinámico

A continuación se presenta un análisis dinámico del dataset, donde se describen efectos y propiedades del mismo, pero en función del lugar y tiempo en el que se produjeron. Se realiza primero un estudio temporal, donde se relaciona la variable de ocupación (y la distribución de transacciones) con la temporalidad del fenómeno bajo estudio. Después, se analiza de forma geográfica, presentando las distribuciones de ocupación por localización (parquímetro). Por último, se explican cuestiones relativas a la frecuencia de actualización de los parquímetros, que es muy relevante a la hora de decidir qué parquímetros utilizar para realizar la predicción.

4.2.1. Análisis temporal

En primer lugar, en la Figura 4.14 se muestra una representación gráfica en la que aparece la distribución estadística de las transacciones (tickets) en función de las horas en las que se produjeron (inicio y fin). Cabe destacar que las horas centrales del día (11h-13h) son las de mayor actividad para el inicio del ticket, y para la hora de fin además de la última hora (19h) también destaca el rango entre las 13-15h.

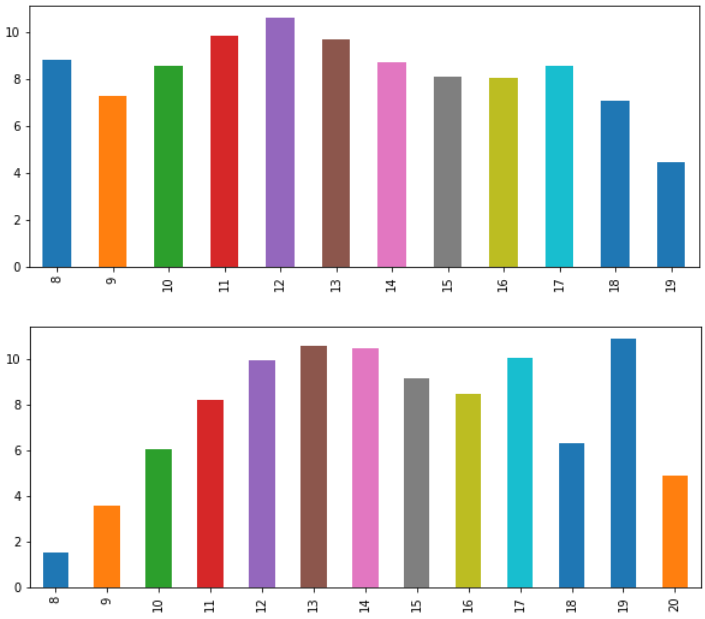

Figura 4.14: Distribución de la hora de inicio y de la hora de fin de las transacciones

Continuando con el análisis, la Figura 4.15 presenta la distribución del porcentaje de ocupación

de los parquímetros en función de la hora del día. Se observa que las horas centrales del día suelen constituir las horas más relevantes para el análisis.

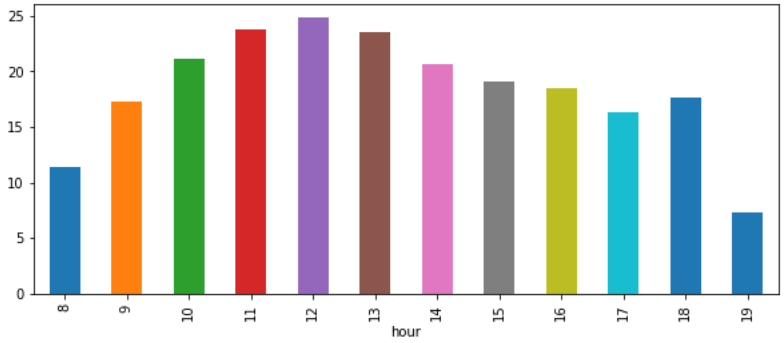

Figura 4.15: Distribución de la ocupación de los parquímetros en función de la hora del día

Dentro de una misma semana, tiende a haber mayor ocupación en los días finales de la semana (Jueves, etiquetado como 3, Viernes, etiquetado como 4, y Sábado, etiquetado como 5), según se representa en la Figura 4.16. Es lógico que se obtenga este resultado, pues los días cercanos al fin de semana suelen llevar aparejados mayores desplazamientos. Nótese que el domingo no aparece representado por no estar activo el sistema de pago por aparcamiento en domingos y días festivos.

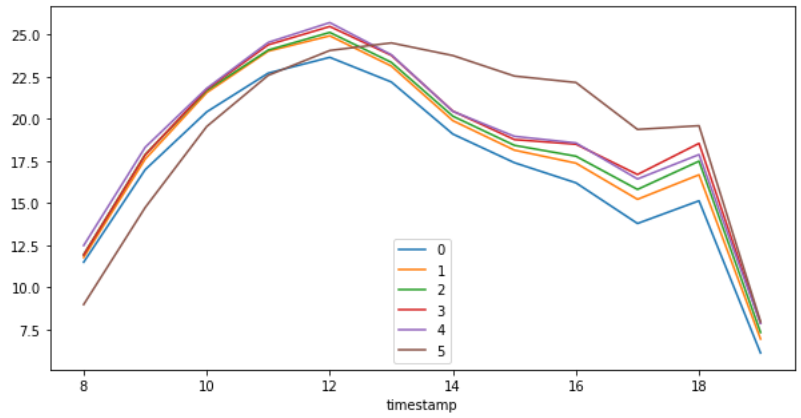

Figura 4.16: Distribución de la ocupación de los parquímetros en función del día de la semana

Si extendemos el análisis a los días dentro de un mes (Figura 4.17), se observa que la distribución es relativamente uniforme: no se aprecia una diferencia significativa entre los días de principio de mes (días 1 a 7, etiquetados como 'begin'), los días de final de mes (días 25 a 31, etiquetados como 'end'), y el resto (etiquetados como 'rest'). Además, también se aprecia que la distribución por horas se mantiene tanto a lo largo de una semana como a lo largo de un mes, siempre observando ocupaciones mayores en las horas centrales del día.

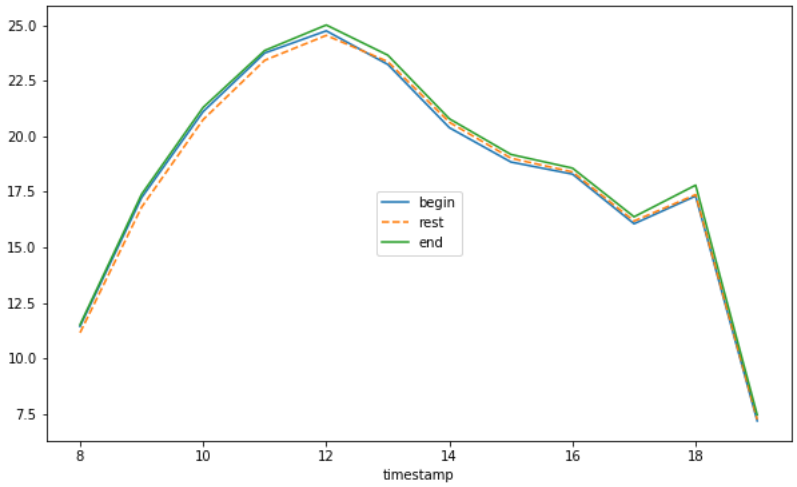

Figura 4.17: Distribución de la ocupación de los parquímetros en función del día del mes

Y si consideramos un año completo, vemos que no hay gran diferencia entre los meses y se sigue manteniendo la tendencia horaria de mayores ocupaciones entre las 10 y las 14h. Se aprecia que en los meses de verano hay mayor ocupación que a lo largo del resto del año, posiblemente debido a la influencia de temperaturas más suaves, mientras que en las horas de ocupación mayor también destacan los meses de Febrero y Marzo (Figura 4.18).

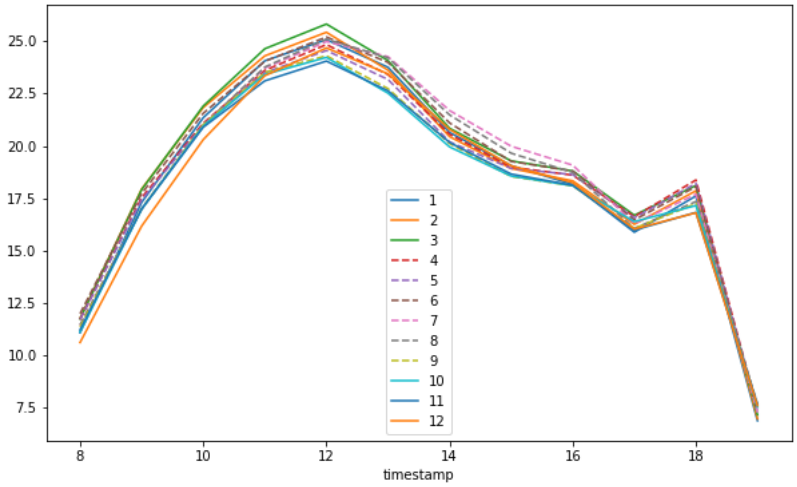

Figura 4.18: Distribución de la ocupación de los parquímetros en función del mes

A continuación observamos la variabilidad de la distribución de ocupación para aquellos parquímetros con mayores porcentajes de ocupación en media que se presentan en la Figura 4.19:

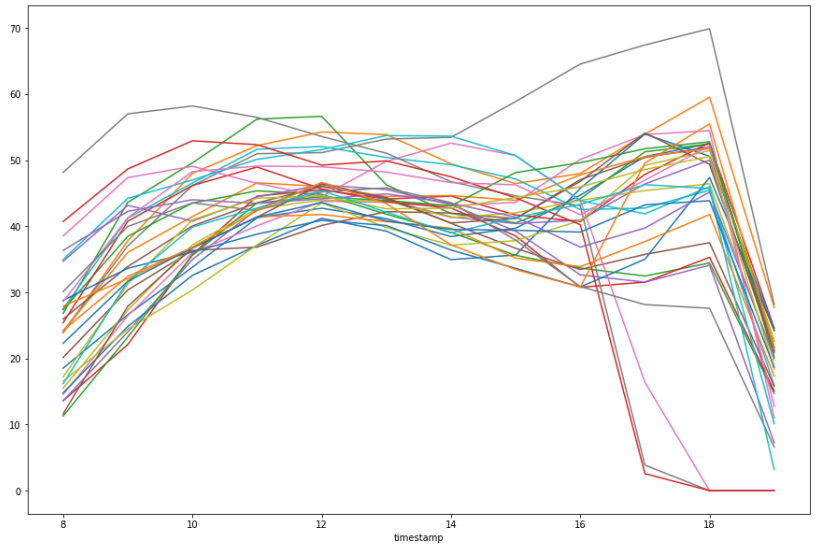

Figura 4.19: Distribución de la ocupación de los parquímetros con mayor porcentaje medio ($> 35\,\%$)

Y por último la distribución de ocupación para los 100 parquímetros con mayor número de transacciones:

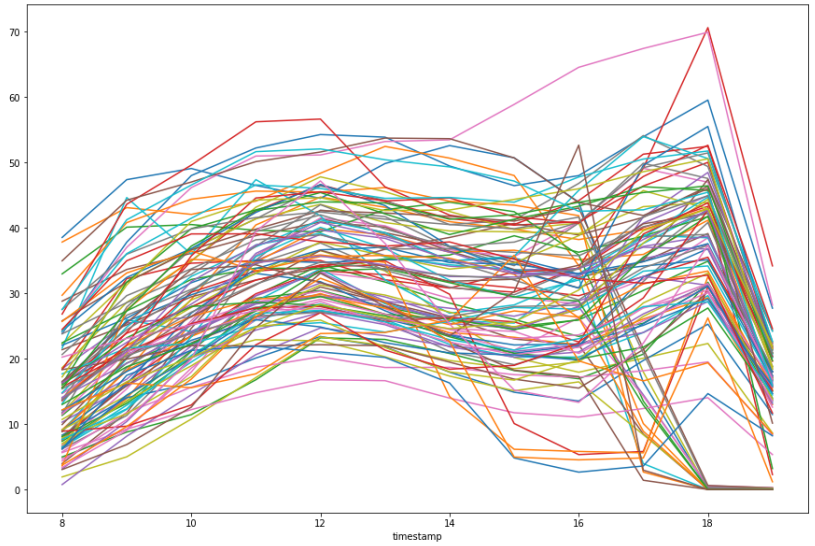

Figura 4.20: Distribución de la ocupación de los 100 parquímetros con más transacciones

4.2.2. Análisis espacial

En cuanto a la distribución espacial de la ocupación de los parquímetros teniendo en cuenta el distrito al que pertenecen, puede observarse en la Figura 4.21 que es muy heterogénea. Hay parquímetros con una tasa de ocupación elevada durante gran parte del día, con picos altos en el rango de 18-19h, y parquímetros que apenas se llenan durante todo el día.

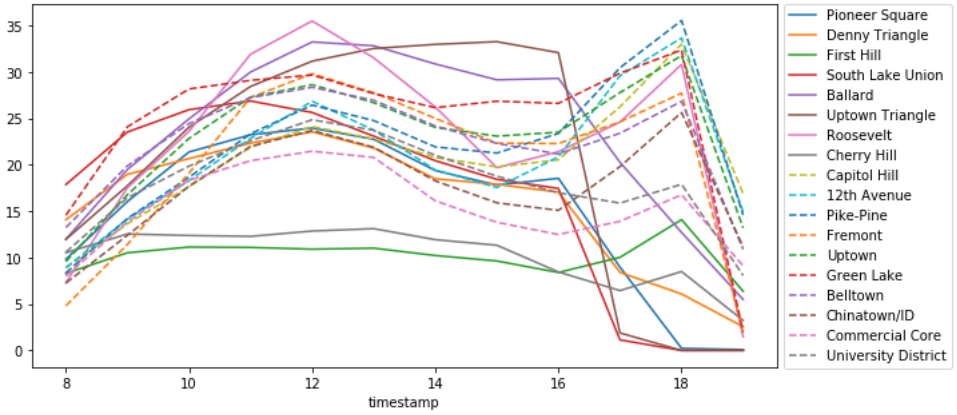

Figura 4.21: Distribución de la ocupación de los parquímetros según su distrito

En la Figura 4.22 puede observarse la ubicación de los parquímetros contenidos en la serie y agrupados por colores identificando los distintos distritos a los que pertenecen:

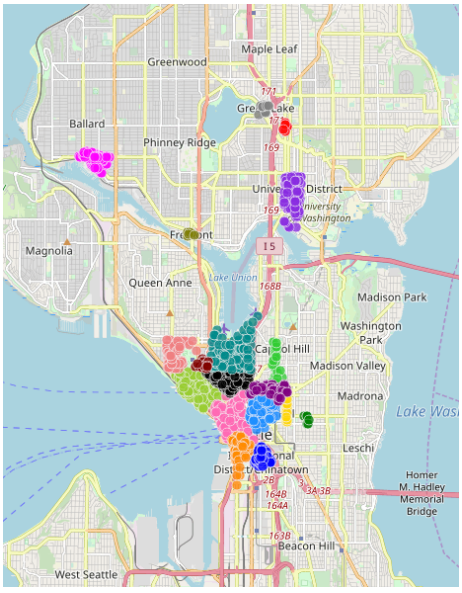

Figura 4.22: Mapa de los parquímetros por distritos

4.2.3. Transacciones diarias por parquímetro

En el año 2016 tenemos 304 días hábiles para el uso de los parquímetros (sin domingos y festivos). Calculamos el número medio de transacciones por día para los 1110 parquímetros existentes en la serie y observamos que sólo para 217 parquímetros hay transacciones todos esos días hábiles. En la Figura 4.23 vemos la distribución del número medio de transacciones por día. El 25 % de los parquímetros tiene en media menos de 1 transacción por hora, el 60 % de los parquímetros tiene en media menos de 2 transacciones por hora y sólo el 5 % de los parquímetros tiene más de 4 transacciones por hora. Tenemos por tanto parquímetros con alto número de transacciones que ven circular muchos vehículos por ellos durante el día junto a parquímetros que prácticamente no tienen movimiento. Disponer de muchas transacciones da información sobre el fenómeno bajo estudio, por lo que seleccionaremos parquímetros de ese tipo que nos permitan realizar buenas generalizaciones.

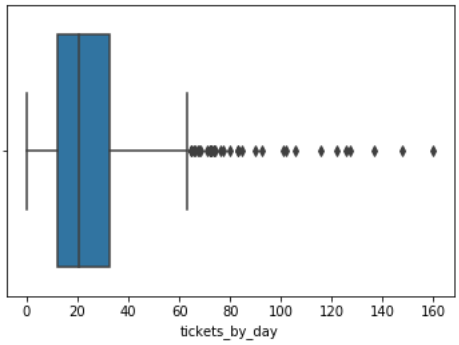

Figura 4.23: Diagrama de caja asociado al número medio de transacciones diarias de los parquímetros

Tomando el parquímetro con id 12289, que de los 30 parquímetros seleccionados para la evaluación de los modelos de predicción es el que tiene menor número de valores de ocupación igual a 0, en la Figura 4.24 se presenta la variación del porcentaje de ocupación en función del tiempo durante la primera semana del año y en la Figura 4.25 durante el primer mes:

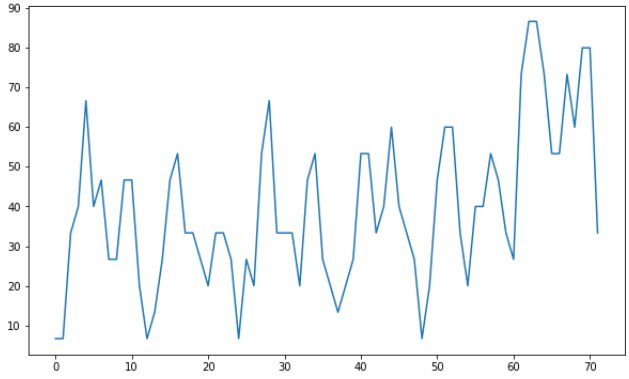

Figura 4.24: Porcentaje de ocupación del parquímetro 12289 durante la primera semana del año

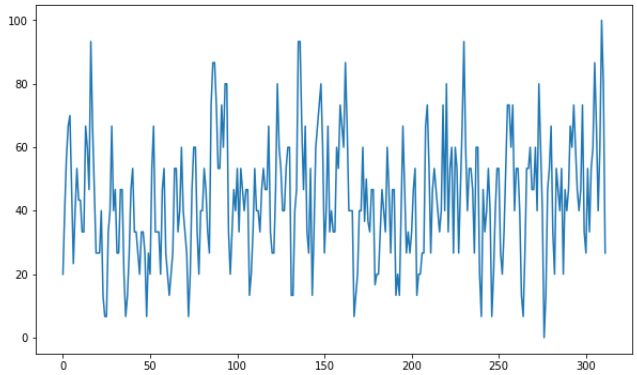

Figura 4.25: Porcentaje de ocupación del parquímetro 12289 durante el mes de Enero

Se observa que no hay una estacionalidad clara en las gráficas, por lo que a priori parece no es sencillo que los modelos consigan realizar una predicción muy exacta.

Capítulo 5

Selección de Variables

Este capítulo presenta un análisis referente a la elección de las mejores variables para alimentar el proceso de entrenamiento y predicción de los modelos de capítulos posteriores. Se llevan a cabo dos análisis independientes, desde dos puntos de vista diferentes. El primer análisis estudia, mediante procedimientos estadísticos, las correlaciones y asociaciones que existen entre las covariables (regresores exógenos) y el *target*. Para ello, se utilizan diferentes técnicas estadísticas. Por otro lado, se realiza también un análisis basado en *Machine Learning*, observando la importancia que las diferentes covariables tienen a la hora de ajustar un modelo estático.

5.1. Análisis estadístico de la importancia de las variables

A la hora de realizar una predicción mediante cualquier modelo predictivo, es de crucial importancia realizar una selección previa de las variables. Esto se debe a que el potencial predictivo de un conjunto de datos depende directamente de las distintas características que introduzca. Un dataset con pocas variables no podrá realizar predicciones complejas y precisas, mientras que un dataset con demasiadas variables tendrá dificultades para generalizar el resultado obtenido (sobreajuste). Normalmente, una combinación de las variables del dataset que deje fuera algunas de las características con menos importancia dará un resultado con la suficiente complejidad como para ser útil, y por otro lado será capaz de generalizar el resultado a muestras que no se encuentren en el conjunto de entrenamiento.

Esta primera sección del capítulo explora las relaciones que existen entre las distintas variables del dataset de parquímetros de Seattle, con el objetivo de ver, exclusivamente, cómo se relacionan los distintos regresores exógenos con la variable de predicción (porcentaje de ocupación de un parquímetro en concreto, a una hora determinada). En el capítulo de EDA ya se han estudiado algunos parámetros similares a lo que se muestra en este capítulo, si bien el objetivo aquí es definir un conjunto de variables para alimentar a los modelos predictivos de capítulos posteriores.

A lo largo de este capítulo se hace referencia en varias ocasiones a la **prueba U de Mann-Whitney**. Esta prueba, entre otras aplicaciones, se utiliza para establecer si hay asociación entre una variable continua (el porcentaje de ocupación en nuestro caso) y una variable binaria. La prueba U de Mann-Whitney es un test no paramétrico (no impone ninguna condición a la distribución de los datos), basado en suma de rangos. En ella, se calcula el estadístico U, definido como:

$$U = \text{mín}(U_1, U_2)$$

U_1 y U_2 se calculan como sigue:

$$U_i = R_i - \frac{n_i(n_i+1)}{2},\ i \in \{1,2\}\ ,$$

donde n_1 es el número de muestras que corresponden a uno de los dos valores de la variable binaria bajo estudio, mientras que n_2 es el número de muestras restantes. R_i es la suma de los rangos correspondiente a cada muestra.

5.1.1. Relevancia de los intervalos horarios

A continuación, se muestran los resultados de aplicar la prueba U a los 30 *element key* seleccionados para análisis. Para ello, se segmenta el dataset por horas, definiendo la hipótesis nula como sigue:

H_0 : encontrarse en el rango horario h_i tiene la misma distribución estadística que no encontrarse en él

Del notebook de Python *'SV_Analisis Estadistico.ipynb'* extraemos los resultados, que se muestran en las Figuras 5.1 - 5.6, e indican que, para un nivel de confianza del 99.5 %, la gran mayoría de los intervalos horarios son significativos, aunque la distribución concreta depende de cada *element_key*.

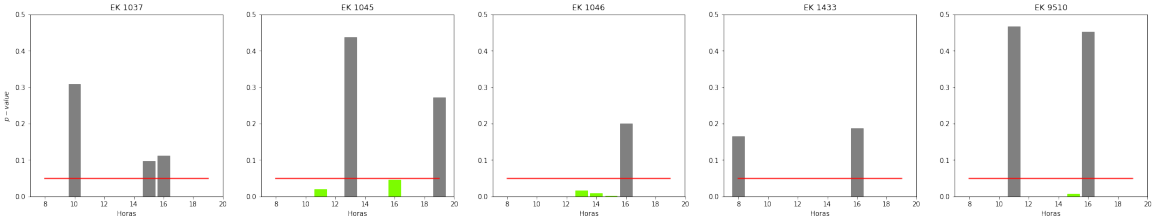

Figura 5.1: Relevancia horas top 30 EK (1 de 6)

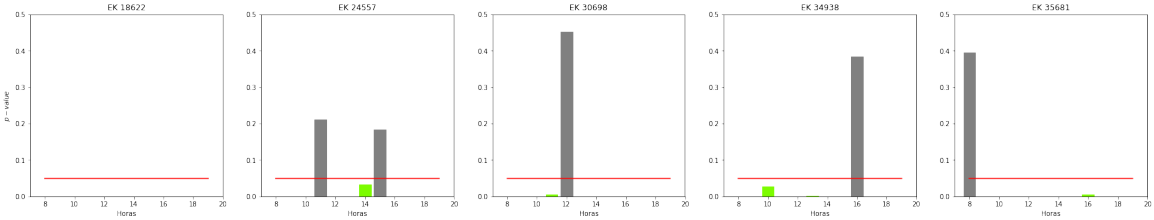

Figura 5.2: Relevancia horas top 30 EK (2 de 6)

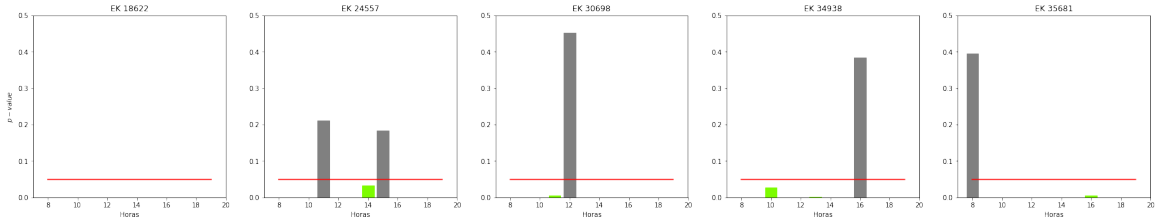

Figura 5.3: Relevancia horas top 30 EK (3 de 6)

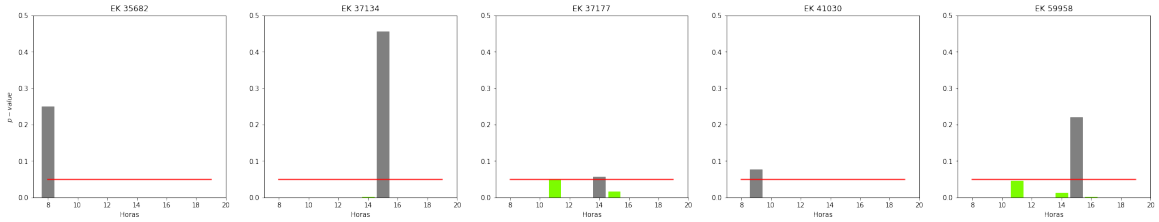

Figura 5.4: Relevancia horas top 30 EK (4 de 6)

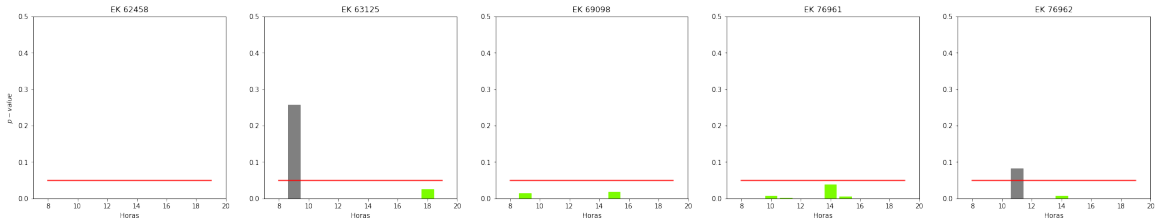

Figura 5.5: Relevancia horas top 30 EK (5 de 6)

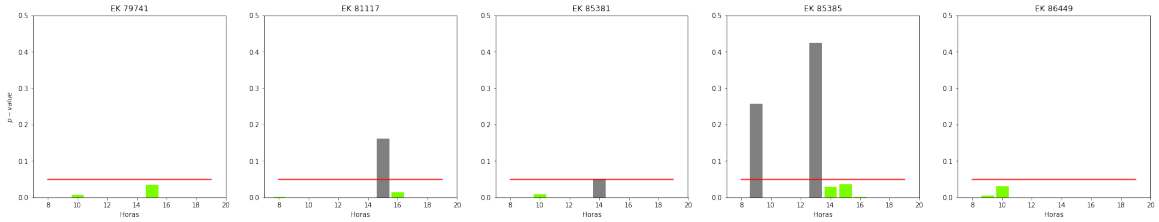

Figura 5.6: Relevancia horas top 30 EK (6 de 6)

5.1.2. Relevancia del día de la semana

En esta sección se lleva a cabo un análisis equivalente al de la sección anterior, pero estudiando la significación estadística del día de la semana en el que se realiza la predicción. Para ello, se utiliza la prueba U de Mann-Whitney, pues estamos comparando de nuevo una variable con varios niveles y una variable continua. En este caso, los intervalos corresponden con el día de la semana: Lunes (L), Martes (M), Miércoles (X), Jueves (J), Viernes (V) y Sábado (S). Los domingos quedan fuera

del análisis al ser días no operativos de los parquímetros. Los resultados calculados en el notebook de Python *'SV_Analisis Estadistico.ipynb'* se presentan en las Figuras 5.7 - 5.12. Del mismo modo que en la sección anterior, se presentan gráficos de barras donde se puede observar el nivel de significación asociado a cada día de la semana y a cada EK. El código de colores es idéntico al de la sección anterior.

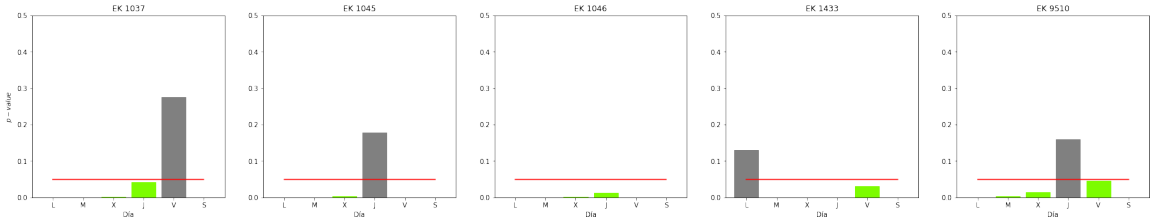

Figura 5.7: Relevancia weekday top 30 EK (1 de 6)

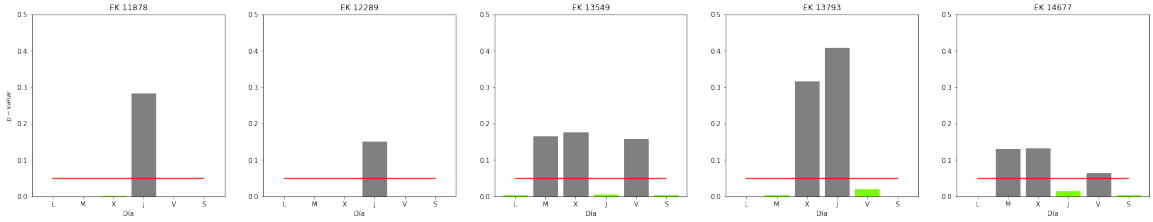

Figura 5.8: Relevancia weekday top 30 EK (2 de 6)

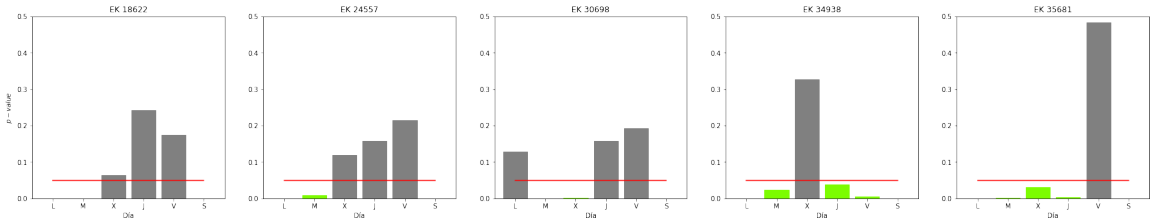

Figura 5.9: Relevancia weekday top 30 EK (3 de 6)

La relevancia del día de la semana es menor que la de la hora del día, pues hay más columnas en gris para el día de la semana. Aun así, los resultados muestran que el día de la semana es relevante para la predicción, aunque menos que la hora del día.

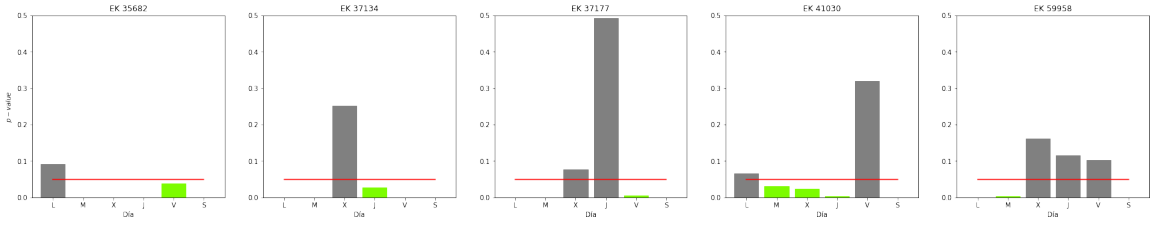

Figura 5.10: Relevancia weekday top 30 EK (4 de 6)

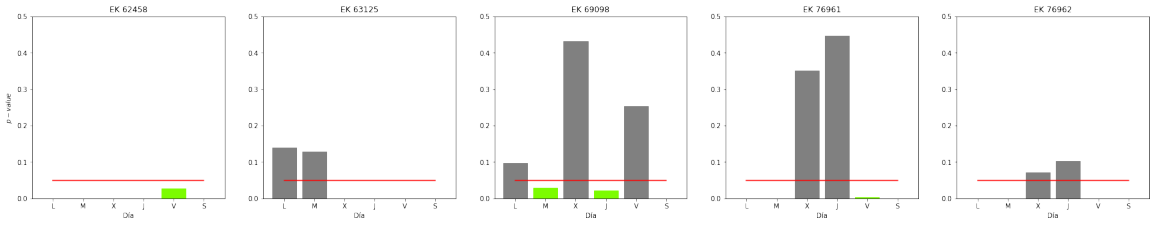

Figura 5.11: Relevancia weekday top 30 EK (5 de 6)

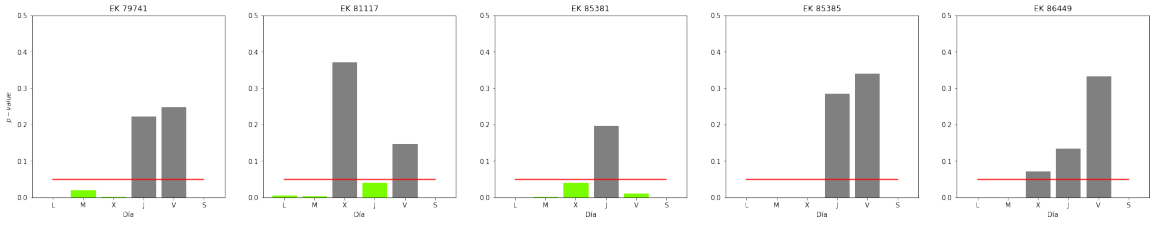

Figura 5.12: Relevancia weekday top 30 EK (6 de 6)

5.1.3. Regresores exógenos

Esta sección analiza, mediante varios métodos, la importancia de los regresores exógenos del dataset. Dado que hay distintos tipos de variables, realizaremos una prueba diferente para cada una de ellas. En concreto:

- Variables continuas: se compara una variable continua con el target (variable continua). Test: Correlación de Pearson
- Variables binarias: se compara una variable binaria con el target (variable continua). Test: U Mann-Whitney

Variables continuas

Son lecturas de sensores acerca de las condiciones atmosféricas y de temperatura. Se les aplica un test de correlación de Pearson con respecto al porcentaje de ocupación (target). Utilizaremos las siguientes reglas para calibrar el significado del coeficiente de correlación (ρ):

- $0{,}0 \leq |\,\rho\,| < 0{,}3$: correlación *débil*

- $0{,}3 \leq |\,\rho\,| < 0{,}7$: correlación *moderada*
- $0{,}7 \leq |\,\rho\,| \leq 1{,}0$: correlación *fuerte*

Si la correlación es moderada o fuerte, el signo de ρ nos indicará el sentido de la correlación (directa o inversa). En el notebook *'SV_Analisis Estadistico.ipynb'*, el nivel de significación se establece en $\alpha = 0{,}05$. En la Figura 5.13, se presenta un gráfico de barras con los resultados del cálculo de la correlación para un *element key* en concreto. Se marcan en gris las variables cuya significación asintótica es menor que el nivel de significación.

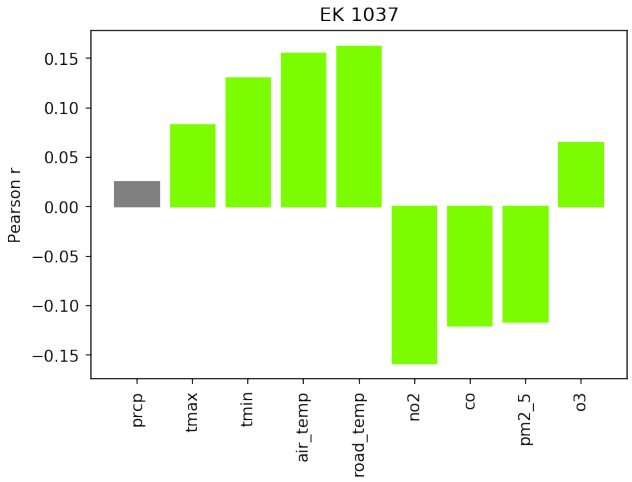

Figura 5.13: Regresores exógenos EK1037

En las Figuras 5.14 y 5.15 se muestra la distribución del coeficiente de correlación de Pearson de cada uno de los regresores exógenos con el porcentaje de ocupación, a lo largo del dataset formado por los 30 EK seleccionados. Además, se representa también su significación asintótica, para calibrar el nivel de significación que tiene el resultado. Como conclusión se establece que no existe una correlación demasiado elevada para ningún regresor exógeno. Esto se debe a la complejidad del fenómeno bajo estudio (distribución del flujo de personas que buscan sitio en un parquímetro). Solo existe un grado de significación compatible con $\alpha = 0{,}05$ para algunas de las variables, que son aquellas para las que, en la Figura 5.15 tienen un pico en la distribución cerca de 0.

5.1.4. Variables binarias

Este análisis únicamente es relevante si se tienen en cuenta varios *element keys*. Si solo se analiza un EK aislado, entonces todas estas variables son constantes y por tanto no aportan información. Por ello, para este cálculo, se consideran todos los EK. Tomando el dataset de *element keys* seleccionados, solo es distinta de cero la variable 'Punto de interés'. Además, esta variable obtiene un nivel de significación del orden de 10^{-53} bajo una prueba U de Mann-Whitney, lo que la convierte en una variable relevante, pero, tal y como se ha comentado, solo en el caso de que se consideren localizaciones geográficas diferentes.

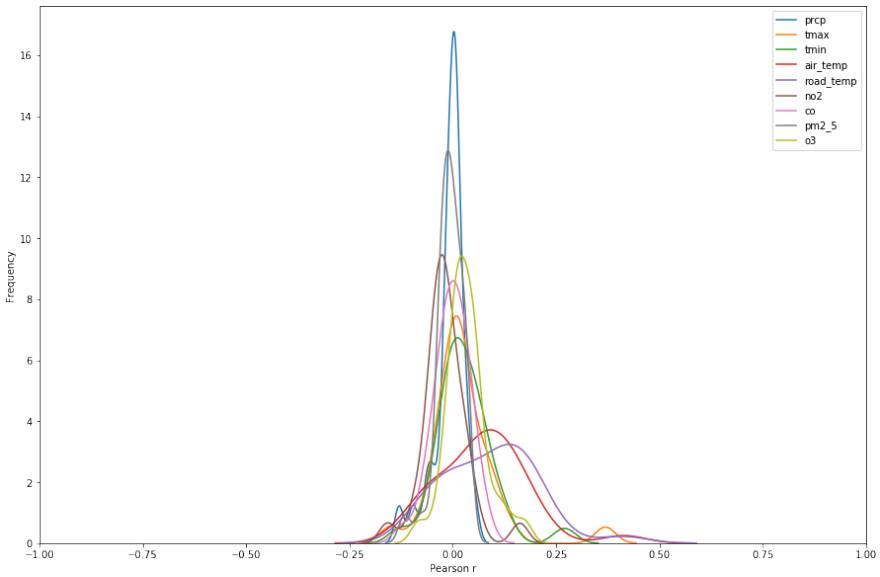

Figura 5.14: Distribución del coeficiente de correlación de Pearson en el top 30

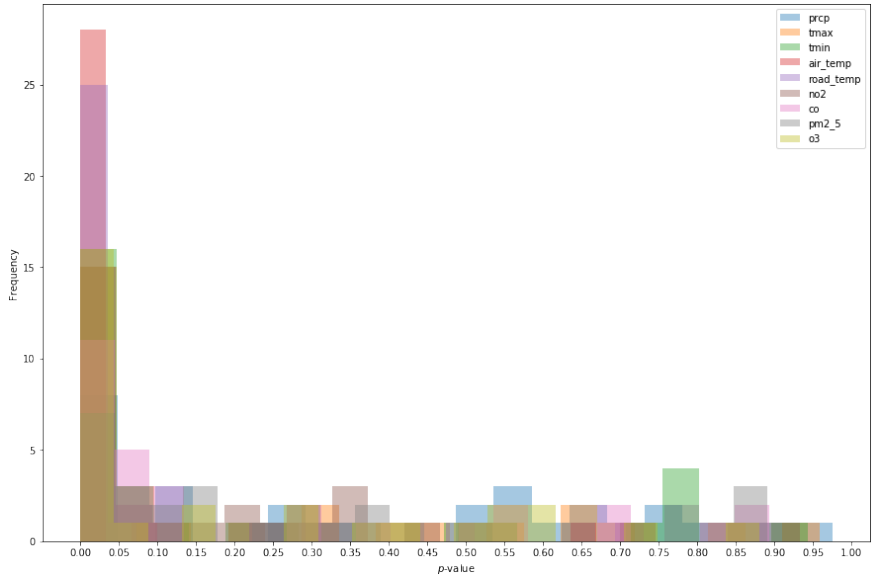

Figura 5.15: Significación asintótica de la correlación del top 30

5.2. Importancia de variables basada en métodos de Machine Learning

En esta sección se presenta un análisis de la importancia de las variables según métodos de Machine Learning. Para ello, se utiliza el concepto de **importancia de permutación** [29]. Este concepto consiste en evaluar la diferencia que existe en la métrica utilizada para el análisis (AUC, precisión, ...) cuando se elimina una de las variables del dataset. Tras llevar a cabo este análisis, se obtienen diferentes pesos para cada una de las variables, clasificándose en términos de su importancia. Debe tenerse cuidado con este método para evaluar la importancia pues, en primer lugar, necesita de un modelo entrenado que dé buenos resultados y, en segundo lugar, es vulnerable a la multicolinealidad.

Por otro lado, es de crucial relevancia mencionar que la interpretación de los resultados de este análisis debe realizarse con cuidado. Esto se debe a que la estructura de los datos bajo análisis es de tipo serie espacio-temporal. Es decir, cada registro está relacionado con el anterior. Por lo tanto, al entrenar los modelos de ML, y separarlos en entrenamiento y validación, se va a romper la estructura de serie de los datos, con lo que el rendimiento del modelo puede ser bajo. Con ello, los resultados obtenidos no van a ser generalizables en sentido estricto, pero sí se va a obtener una medida de cómo son de relevantes las variables, similar a una correlación, pero basándonos en modelos más complejos que el coeficiente de Pearson. Para evaluar la importancia de permutación, se ha utilizado la implementación de ELI5, que dispone de un *wrapper* para los modelos entrenados con *sklearn*. Se han entrenado dos modelos, basados en Random Forest y Gradient Boosting respectivamente.

5.2.1. Entrenamiento de los modelos

El entrenamiento de los modelos se ha realizado según los siguientes pasos:

1. Discretización de la variable target (porcentaje de ocupación) en 5 intervalos, para que encaje con los modelos de clasificación aceptados por la implementación de ELI5. El número de intervalos se ha obtenido de un compromiso entre rendimiento del modelo y capacidad de cómputo.

2. Discretización de la variable (latitud, longitud) en 25 cuadrantes.

3. Eliminación del identificador de cada EK.

4. Separación en entrenamiento y validación.

5. Fitting.

6. Análisis de resultados.

El modelo basado en Random Forest se ha entrenado con 75 estimadores, sin limitación en la profundidad de los nodos ni en su número de hojas. Del mismo modo, el modelo basado en Gradient Boosting utiliza también 75 árboles, sin penalizar la profundidad de los mismos. Aunque el hecho de no limitar la profundidad de los árboles puede llevar a sobreajustar el conjunto de datos, en este caso se ha realizado así para aumentar la capacidad predictiva de ambos modelos. Los resultados se evalúan mediante el ratio de verdaderos positivos (TPR) y el de falsos negativos (FPR), presentándolos en la curva ROC, y calculando el área bajo la curva (AUC), en la Figura

Modelo	AUC
Random Forest	0.57
Gradient Boosting	0.54

Tabla 5.1: AUC modelos Permutation Importance

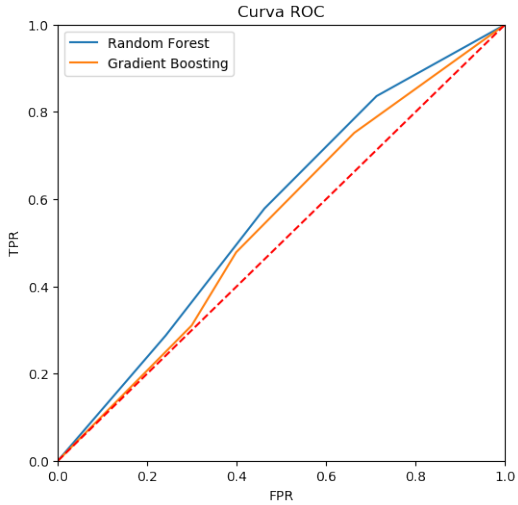

Figura 5.16: Validación de los modelos utilizados para la importancia de permutación

5.16, extraída del notebook *'SV_Analisis basado en metodos de Machine Learning.ipynb'*. Los dos modelos tienen, en concreto, el valor de AUC que se muestra en la Tabla 5.1.

Se observa que el modelo basado en Random Forest es algo superior al de Gradient Boosting, con los parámetros seleccionados para su entrenamiento. Sin embargo, los dos modelos tienen dificultades para superar el valor de 0.5, que correspondería a una clasificación aleatoria. Esto se debe a lo que se comentó previamente acerca de la estructura de serie de los datos.

5.2.2. Permutation Importance

Para concluir esta sección, se muestran los pesos de las variables más relevantes según los dos modelos estudiados. Estos resultados se presentan en la Figura 5.17, extraída del notebook *'SV_Analisis basado en metodos de Machine Learning.ipynb'*.

Weight	Feature
0.0412 ± 0.0027	road_temp
0.0200 ± 0.0013	air_temp
0.0179 ± 0.0007	poi
0.0175 ± 0.0006	longitude_0
0.0125 ± 0.0011	longitude_2
0.0107 ± 0.0007	latitude_5
0.0103 ± 0.0009	latitude_1
0.0095 ± 0.0016	longitude_1
0.0065 ± 0.0021	tmax
0.0050 ± 0.0004	latitude_0
0.0036 ± 0.0005	latitude_3
0.0022 ± 0.0016	longitude_5
0.0019 ± 0.0011	latitude_2
0.0013 ± 0.0013	latitude_4
0.0011 ± 0.0005	longitude_4
0.0003 ± 0.0013	longitude_3
0 ± 0.0000	event
0 ± 0.0000	soccer
-0.0001 ± 0.0001	basket
-0.0002 ± 0.0001	baseball

Random Forest

Weight	Feature
0.0656 ± 0.0025	road_temp
0.0337 ± 0.0024	tmax
0.0239 ± 0.0011	poi
0.0224 ± 0.0015	latitude_1
0.0192 ± 0.0009	longitude_0
0.0180 ± 0.0017	longitude_2
0.0167 ± 0.0014	longitude_1
0.0116 ± 0.0008	longitude_5
0.0114 ± 0.0007	latitude_0
0.0106 ± 0.0008	latitude_5
0.0062 ± 0.0006	latitude_4
0.0061 ± 0.0008	latitude_2
0.0047 ± 0.0009	tmin
0.0042 ± 0.0004	longitude_3
0.0036 ± 0.0006	air_temp
0.0030 ± 0.0012	latitude_3
0.0022 ± 0.0012	no2
0.0014 ± 0.0004	longitude_4
0.0012 ± 0.0004	o3
0.0009 ± 0.0004	prcp

Gradient Boosting

Figura 5.17: Pesos importancia de permutación

5.3. Conclusiones

Como conclusión, a partir de los análisis derivados de las secciones previas, se establece que tanto la hora del día como el día de la semana son relevantes para el análisis y en consecuencia deben formar parte del conjunto de datos de entrenamiento. Esto se ha deducido a partir de un análisis estadístico basado en la prueba U de Mann-Whitney.

Por otro lado, se ha presentado un estudio de los regresores exógenos del dataset. En primer lugar, se ha analizado cómo varía el coeficiente de correlación con la variable de respuesta (porcentaje de ocupación). Este estudio establece que no existen variables que estén fuertemente correlacionadas linealmente (coeficiente de Pearson). Sin embargo, algunas de ellas tienen un coeficiente de correlación mayor que otras: la temperatura de la carretera (*road_temp*) y la temperatura del aire (*air_temp*), que toman valores cercanos a 0.3 (positivo). Además, estas dos variables son significativas ($p < 0{,}05$). Por otro lado, aparecen como significativas la temperatura máxima y la mínima ($p < 0{,}05$). Sin embargo, la temperatura máxima y la mínima están correlacionadas (Capítulo de EDA), al igual que la temperatura del aire y la de la carretera. Por tanto, debe elegirse solo una de ellas, debido a que la otra puede deducirse a partir de la primera.

Asimismo, un análisis basado en métodos de Machine Learning (Permutation Importance de árboles de decisión, segmentando la variable de respuesta en intervalos, clasificando las muestras de entrada), establece que las variables que hemos mencionado anteriormente toman una parte importante en la predicción. Debe tenerse precaución con la inferencia de conclusiones a partir de la importancia de permutación, pues el modelo ajustado obtiene un AUC de entre 0.54 y 0.57, muy cerca de la clasificación aleatoria (0.5). Esto se debe a que estos métodos rompen la estructura de serie de los datos.

Por lo tanto, los variables seleccionadas para el análisis son:

1. Intervalos horarios de 1h de duración
2. Día de la semana
3. *road_temp*
4. *tmin*

En el caso de modelos que acepten distintos puntos geográficos como variable de entrada, la variable *'poi'* (punto de interés), también es significativa y por tanto se incluye en el análisis.

Capítulo 6

Descripción y aplicación de modelos predictivos

Que un hecho o cantidad sea predecible depende de varios factores principales [44]:

- cuántos datos hay disponibles
- cómo de bien entendemos los factores que contribuyen al hecho o cantidad
- si las predicciones pueden afectar al resultado de lo que se intenta predecir

Buenas predicciones son las que capturan los patrones y las relaciones que existen en los datos históricos pero no replican los eventos pasados que no volverán a ocurrir. Todos los entornos son cambiantes, pero un buen modelo de predicción capta el modo en que las cosas cambian, asumiendo habitualmente que el modo en que el entorno cambia continuará en el futuro.

Los métodos de predicción de series temporales más simples son aquellos que usan información sólo de la variable que se predice, sin tener en cuenta los factores externos que pueden afectar a su comportamiento. En aquellos métodos más avanzados que lo permiten, hemos considerado también la opción de incluir variables adicionales externas como regresores, seleccionando tres variables: la temperatura del asfalto y la temperatura mínima, porque son dos de las variables destacadas por el análisis de importancia comentado en el capítulo anterior, y añadimos el día de la semana como variable *dummie*. Y adicionalmente hemos considerado la transformación logarítmica de la serie para comparar el rendimiento en la predicción de los modelos. La transformación logarítmica es popular en análisis de series temporales porque estabiliza la varianza, aunque no por ello está asegurado que se mejoren las predicciones. En nuestro caso hemos comparado el rendimiento en la predicción de algunos de los modelos considerando la serie original y la serie transformada logarítmicamente.

Estacionalidad múltiple: Una serie temporal presenta efectos estacionales si el comportamiento de la serie es parecido en ciertos tramos de tiempo periódicos en el tiempo. Una serie con datos horarios como la nuestra suele tener típicamente 3 tipos de estacionalidad: diaria, semanal y anual. Hemos acotado el análisis al primer trimestre del año 2016 para que nuestros equipos informáticos puedan realizar los cálculos en plazos de tiempo razonables, por lo que nuestra serie tiene dos estacionalidades, diaria y semanal, con la particularidad de que el periodo diario está acotado a 12 horas y el periodo semanal a 6 días, por los horarios de funcionamiento de los parquímetros.

Para tratar con este tipo de series en los que hay varios tipos de estacionalidad, utilizamos la clase *msts* de R que nos permite especificar todas las frecuencias que son relevantes e incluso admite frecuencias no enteras.

En la Figura 6.1 se muestra para dos parquímetros distintos el desglose de su serie, acotada temporalmente al primer trimestre del año, con las dos estacionalidades mencionadas (diaria y semanal). Se observa en ambos casos una periodicidad dinámica y evolutiva en el comportamiento de las dos gráficas de estacionalidad y también en la de tendencia. Destacamos también la diferencia de comportamiento de la serie entre los dos parquímetros seleccionados al azar y que se puede intuir a partir de las gráficas que será complejo conseguir alcanzar unas buenas aproximaciones en las predicciones.

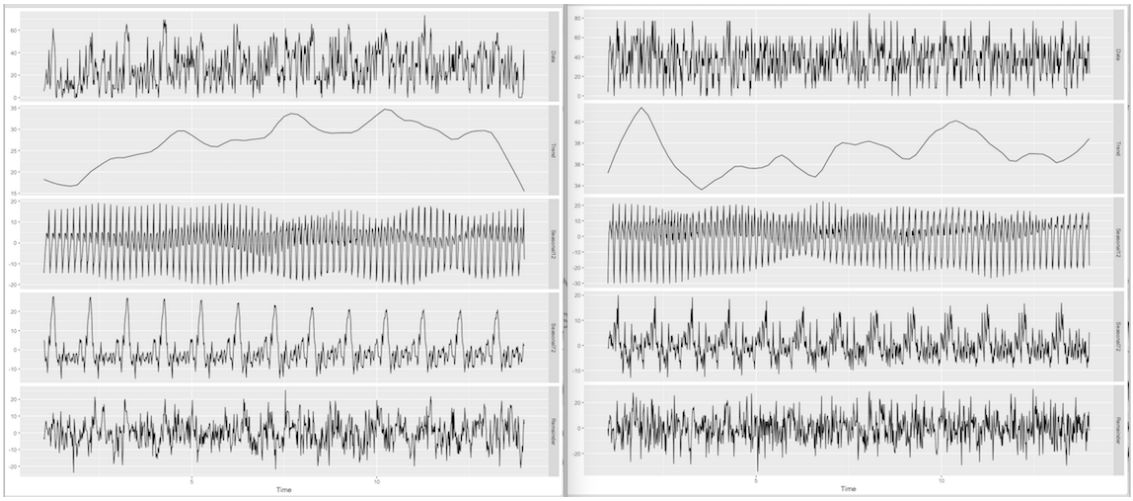

Figura 6.1: Doble estacionalidad de la serie para dos parquímetros distintos (ids 1037 y 37177)

6.1. Modelos considerados

6.1.1. Auto-arima

[30] El modelo *auto.arima* ajusta la serie combinando valores de p, d y q y selecciona el mejor modelo *ARIMA* que tiene el menor estadístico AIC corregido. Hemos comprobado si la predicción mejoraba estableciendo un lambda distinto al valor NULL por defecto, definiendo *lambda = "auto"* para que se seleccione automáticamente una transformación Box-Cox. Las transformaciones Box-Cox se basan en la siguiente función para $\lambda \neq 0$:

$$f_\lambda(x) = \frac{x^\lambda - 1}{\lambda}$$

Si $\lambda = 0$ entonces la función sería:

$$f_0(x) = log(x)$$

Para efectuar esta prueba hemos seleccionado el *element key* cuya serie tiene menos valores igual a 0 de porcentaje de ocupación, id 12289, y hemos acotado el análisis al primer trimestre. Los resultados del MAE obtenido son los siguientes:

Min.	7.919
1st Qu.	11.075
Median	13.819
Mean	17.521
3rd Qu.	17.450
Max.	52.333

En contraposición con el que obteníamos dejando lambda=NULL:

Min.	7.399
1st Qu.	10.884
Median	12.916
Mean	13.122
3rd Qu.	15.334
Max.	19.947

El MAE empeora por lo que optamos por dejar el valor por defecto y no proceder a ninguna transformación Box-Cox en el notebook *'MOD_Forecast.Rmd'*. Y seleccionamos *seasonal=TRUE* porque tenemos una serie multiestacional definida con *msts*.

El modelo *auto.arima* permite incluir variables externas como regresores, por lo que hemos utilizado las 3 variables antes mencionadas (temperatura del asfalto, temperatura mínima y día de la semana). Al aplicar la función predictiva *forecast* nos hemos encontrado con un problema explicado en la documentación de la función; cuando a un modelo creado con *arima* se le especifican regresores, la función *forecast* ignora el periodo de predicción (h=12 horas siguientes), por lo que consideramos la aplicación de un tipo de predicción conocida como *ex-post*, ya que utilizamos información de los predictores externos en instantes de tiempo posteriores al momento de la predicción. En estos casos las predicciones no son válidas por sí mismas pero si son útiles como en nuestro caso para estudiar el comportamiento de los modelos de predicción en relación a los regresores.

En el notebook *'MOD_Forecast.Rmd'*, además de comparar los modelos *auto.arima* con y sin regresores para la serie original, hemos considerado la transformación logarítmica de la serie también en ambos casos, por lo que tenemos un total de 4 modelos *auto.arima* por cada parquímetro.

6.1.2. Medias Móviles

La base del modelo de *medias móviles (Moving-Average)* es que asume que las desviaciones actuales de los valores medios aritméticos de la serie dependen de las desviaciones anteriores. El orden que se le fija a la media móvil determinará la desviación de la estimación y, cuanto más alto sea este valor, más homogénea será la desviación. Nosotros hemos fijado el orden en 3, lo que significa que para la predicción del periodo n+1, el modelo tendrá en cuenta los valores de n, los de n-1 y los de n-2, calculando la media de ellos.

Este modelo no admite regresores. Hemos aplicado el modelo sobre la serie original y sobre la serie transformada logarítmicamente (notebook *'MOD_Forecast.Rmd'*).

6.1.3. MSTL

STL es un modelo versátil y robusto para descomponer series temporales. *STL* es un acrónimo de *"Seasonal and Trend decomposition using Loess"*, siendo *Loess* un método de estimación de relaciones no lineales [44]. *STL* tiene varias ventajas sobre otros métodos clásicos de descomposición:

- maneja cualquier tipo de estacionalidad, no sólo mensual o trimestral
- la componente de estacionalidad puede variar con el tiempo
- es robusto frente a outliers (valores atípicos) para las estimaciones de tendencia y estacionalidad aunque estos outliers afectarán a la componente residual de la serie

Utilizamos el modelo *MSTL* que es una variación de *STL* para series con estacionalidad múltiple. Utilizando la función *mstl* de R (también específica para series con estacionalidad múltiple), sobre la serie construida con la función *msts*, obtenemos las gráficas de las dos estacionalidades indicadas (diaria y semanal), la tendencia (que se observa evolutiva) y el componente residual.

Este modelo no admite regresores. Hemos aplicado el modelo sobre la serie original y sobre la serie transformada logarítmicamente (notebook *'MOD_Forecast.Rmd'*).

6.1.4. BATS y TBATS

BATS es otro modelo cuyo acrónimo destaca las características más relevantes [44]:

- B: transformaciones Box-Cox, mencionadas anteriormente, que arreglan problemas de normalidad y heterocedasticidad, es decir, no homogeneidad de varianzas
- A: errores ARMA (modelo de medias móviles autoregresivos para la estimación del componente residual de la serie)
- T: Tendencia
- S: estacionalidad (Seasonality)

Y *TBATS* es un modelo lanzado como BATS en 2011 y añade regresores trigonométricos para modelar múltiples estacionalidades. Los modelos *BATS* y *TBATS* son métodos de descomposición de una serie temporal que permiten que sus múltiples estacionalidades se incorporen simultáneamente y que cambien lentamente con el tiempo. Cada componente de la serie se estima explícitamente y se mide estadísticamente. Después cada componente estimado se recombina para realizar la predicción final. Un par de inconvenientes de los modelos *BATS* y *TBATS* es su lentitud, especialmente con series largas, y que los intervalos de confianza para la predicción suelen ser demasiado amplios.

Estos dos modelos no admiten regresores. Los hemos aplicado sobre la serie original y sobre la serie transformada logarítmicamente (notebook *'MOD_Forecast.Rmd'*).

6.1.5. Holt-Winters y DSHW

El modelo de *Holt-Winters* computa el filtrado de *Holt-Winters* de una serie temporal dada, que se fundamenta en la estacionalidad. Este modelo tiene dos variaciones, por una parte la aditiva y por la otra la multiplicativa. Se prefiere la aditiva cuando las variaciones estacionales son en general constantes a lo largo de la serie, mientras que se escogerá la multiplicativa cuando las variaciones estacionales cambien proporcionalmente según el nivel de la serie. Con nuestra serie hemos comprobado que se obtienen mejores resultados cuando la configuración es aditiva.

DSHW (Double Seasonal Holt-Winters) es un método de 1960 y como su nombre indica es una variación del modelo *Holt-Winters* para series con doble estacionalidad. Estos modelos no admiten regresores. Los hemos aplicado sobre la serie original y sobre la serie transformada logarítmicamente (notebook *'MOD_Forecast.Rmd'*).

6.1.6. BSTS

[45] [31] [32] [33] [34] El modelo *BSTS* de acuerdo a sus iniciales (Bayesian Structured Time Series) se puede explicar como:

- Bayesiano: significa que la implementación tiene un enfoque bayesiano. De manera pragmática, tiene 2 consecuencias principales:
  1. todas las salidas del modelo vendrán en una distribución con un intervalo de certeza (y en realidad todos los parámetros dentro del modelo tendrán una distribución), y
  2. es posible expresar el conocimiento previo sobre la serie objetivo a través de los *priors* bayesianos (que pueden considerarse como hiperparámetros del modelo).
- Estructural: significa que BSTS proporciona un enfoque estructural para el modelado, en el que hay disponible un kit de componentes para capturar diferentes aspectos de la serie; la arquitectura del modelo puede incluir o excluir cualquiera de esos componentes.
- Time series: BSTS utiliza modelos de espacio de estado, que es una metodología de modelado en la que el sistema se describe como la combinación de un vector de estado y un vector de observación, ambas series de tiempo. La relación entre el estado y la observación está descrita por el modelo de espacio de estado; el objetivo es inferir las propiedades del estado, que está oculto, de las observaciones disponibles en el pasado. Las previsiones se producen a partir de los estados futuros estimados.

En sus diferentes formas (modelos de espacio de estado, filtros de Kalman), los modelos BSTS se han utilizado "tradicionalmente"(desde los años 60), y todavía están en uso ya que son muy adecuados para algunos escenarios. No están tan comúnmente cubiertos en los cursos y recursos en línea, tutoriales, etc., y son un poco más difíciles de usar en comparación con otros modelos de aprendizaje automático; sin embargo, hay una biblioteca de código abierto de muy alta calidad disponible en R, razonablemente directa en su uso y bien documentada.

Componentes estructurales

BSTS proporciona un kit de componentes que se pueden usar para modelar la serie. Estos componentes capturan diferentes aspectos de la serie, y se pueden agregar o eliminar según las

necesidades, como en un juego de construcción. En la descomposición clásica son aproximadamente:

$$y_t = \underbrace{\mu_t}_{trend} + \underbrace{\gamma_t}_{seasonal} + \underbrace{\beta^T x_t}_{regression} + \epsilon_t$$

$$\mu_t = \mu_{t-1} + \delta_{t-1} + u_t$$

$$\delta = \delta_{t-1} + v_t$$

$$\gamma_t = -\sum_{s=1}^{S-1} \gamma_{t-s} + w_t$$

- Componente *trend*, captura los aspectos de tendencia de la serie
- Componente *seasonal*, capta los aspectos periódicos de la serie
- Componente *regression*, captura la influencia de las variables explicativas (es decir, variables externas a la serie para predecir que proporcionan cierta información a la predicción)

El kit completo de componentes se encuentra en la descripción de la biblioteca BSTS [35].

Breve descripción del funcionamiento de BSTS

BSTS realiza dos operaciones principales: filtrado y suavizado (*filtering and smoothing*). El filtrado proporciona una predicción de la serie dados todos los datos disponibles hasta el momento y el suavizado corrige el estado del modelo cuando una nueva observación de la serie está disponible; es decir, el modelo compara la predicción con la observación y utiliza el error para corregir su propio estado. Al entrenar el modelo la librería revisa internamente todos los datos periodo a periodo, ejecutando sucesivas operaciones de filtrado y suavizado, hasta el último periodo de tiempo disponible.

Es importante tener en cuenta que estas operaciones de filtrado consecutivas se vuelven cada vez menos precisas para un horizonte creciente de la predicción. En nuestra serie evitamos este problema considerando un horizonte de predicción corto, las 12 horas del día siguiente.

Enfoque bayesiano:

En cada uno de los componentes de un modelo BSTS, es posible configurar *priors* Bayesianos que capturan información previa sobre la serie objetivo para predecir. A primera vista, no parece fácil para un data scientist traducir el conocimiento del negocio sobre la serie a la definición de los priors. Por ejemplo, el componente *LocalLevel* más simple disponible, que es un componente de tendencia, tiene 2 *priors*:

- *initial.state.prior*: describe la distribución del vector anterior al estado inicial. Esta variable se puede omitir. En la mayoría de las situaciones prácticas, el modelo calcula correctamente el valor inicial del vector de estado sin necesidad de agregar información previa.
- *sigma.prior*: se puede interpretar como un parámetro para el suavizado, que determina qué tan ajustado es el componente de tendencia que seguirá los últimos valores de la serie. Un valor alto de *sigma.prior* indica que el componente seguirá la serie con firmeza, y un valor bajo indica lo contrario.

Lo más recomendable es realizar validación cruzada para seleccionar los parámetros con las desventajas que esto conlleva (tiempo excesivo, soluciones buenas pero quizás no las mejores, etc.)

Modelos de BSTS elegidos

Hemos seleccionado tres componentes del modelo BSTS: *LocalLevel*, *Ar* y *LocalLinearTrend*, sabiendo que alguno no es el más adecuado tal y como comenta el diseñador de la librería [36]:

"El modelo LocalLinearTrend es muy flexible, pero esta flexibilidad puede aparecer como una varianza no deseada en los pronósticos a largo plazo. La mayoría de las variaciones de tus series de tiempo parecen provenir de la estacionalidad, por lo que puedes probar LocalLevel o incluso AddAr en lugar de LocalLinearTrend"

Además, por cada componente consideramos su aplicación sin regresores y con regresores, considerando las 3 variables externas antes mencionadas.

Componente LocalLinearTrend:

Agrega un modelo de tendencia lineal local a una especificación de estado. El modelo asume que tanto la media como la pendiente de la tendencia son *random walk*, siendo *random walk* una serie temporal donde la observación actual es igual a la observación previa con una variación aleatoria incremental o decremental, teniendo esa variación una distribución de ruido blanco (sin correlación estadística entre sus valores).

La ecuación para la media es:

$$\mu_{t+1} = \mu_t + \delta_t + rnorm(1, 0, sigma.level)$$

La pendiente es:

$$\delta_{t+1} = \delta_t + rnorm(1, 0, sigma.slope)$$

La distribución anterior se encuentra en el nivel de desviación estándar *sigma.level* y la pendiente de desviación estándar *sigma.slope*.

Componente AR

Agrega un componente *AR(p)* a una especificación de estado.

El modelo es:

$$\alpha_t = \phi_1 * \alpha_{t-1} + ... + \phi_p * \alpha_{t-p} + \epsilon_{t-1}$$

, con

$$\epsilon_{t-1} \sim N(0, \sigma^2)$$

El estado consiste en los últimos valores de α. La matriz de transición de estado tiene ϕ en su primera fila, valores igual a 1 a lo largo de su primer subdiagonal y valores igual a 0 en el resto. La matriz de varianza del estado tiene σ^2 en su esquina superior izquierda y es cero en el resto. La matriz de observación tiene el valor 1 en su primer elemento y es cero en el resto.

Componente LocalLevel

Agrega un modelo de nivel local a una especificación de estado. El modelo a nivel local asume que la tendencia es un random walk:

$$\alpha_{t+1} = \alpha_t + rnorm(1, 0, \sigma)$$

Una de las ventajas de la librería de BSTS es que nos permite obtener gráficas como la siguiente mostrando una comparativa del comportamiento del error acumulado para 3 componentes del modelo con un parquímetro concreto:

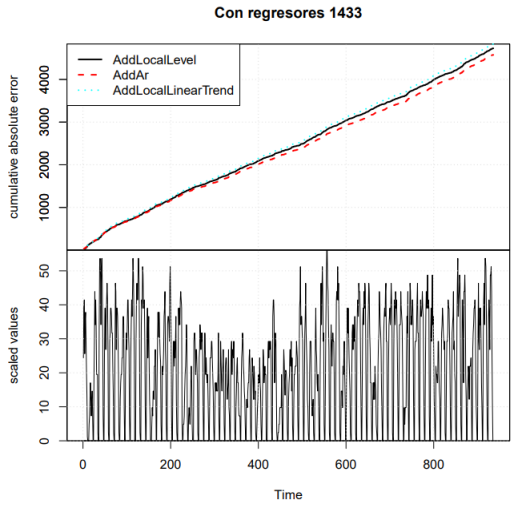

Figura 6.2: Ejemplo de comparativa de tres componentes BSTS para un parquímetro seleccionado

6.1.7. SpTimer

[37] [38]

Modelos espacio-temporales

SpTimer es una librería para análisis de experimentos que tienen una estructura espacio-temporal. Antes de empezar con spTimer, se describe muy brevemente cómo se modelan eventos espacio-temporales. De forma sencilla, un modelo espacio-temporal es aquel en el que se han recogido datos de forma espacial (coordenadas) y a intervalos regulares (temporal). Por ejemplo, en nuestro caso de los parquímetros, suponiendo la serie agregada, tendremos datos de ocupación ligados a una coordenada geográfica a intervalos regulares de tiempo. Entonces, la labor del modelo es más complicada que en otros tipos: debe encontrar correlaciones espaciales y correlaciones temporales.

SpTimer

SpTimer es una librería que construye modelos espacio-temporales basándose en modelos gaussianos bayesianos. Presenta tres modelos a partir de los cuales se realizan las predicciones:

- Bayesian Gaussian Process (GP)

- Bayesian Auto-Regressive Process (AR)
- Bayesian Gaussian Predictive Processes (GPP) based AR Model

Aunque son modelos complicados, es muy importante conocer cuáles son los parámetros que se van a ajustar y cuáles son sus características.

Características y parámetros de los modelos de spTimer:

- Constan de una parte lineal, parecida a una regresión. Estas componentes se suelen llamar *"covariates"* en la literatura, y suelen denotarse por X_{1t}. Cada una de esas componentes tiene asociado un parámetro, formándse así un vector de parámetros *regresivos*, β.
- Aparecen dos *componentes de error* diferentes: "nugget error", que se asocia con el término de error puro (ϵ), y errores aleatorios espacio-temporales (η). Matemáticamente, cada una de estas componentes se modela con una varianza (σ_ϵ^2 y σ_η^2).
- Hay dos parámetros más: ϕ, que controla el ratio con el que la correlación entre dos coordenadas decae conforme aumenta la distancia entre ellas, y ν que es el orden de la función de Bessel de segunda especie que se utiliza para calcular la correlación espacial.
- Son *bayesianos*, en el sentido de que parten de una distribución conocida (distribución a priori o prior) de los parámetros y calculan a partir de ella una distribución final (a posteriori, $\pi(\cdot \mid z)$).

En resumen, con cada uno de los tres modelos de spTimer, lo que se trata es de buscar los coeficientes:

- $\beta = (\beta_0, \beta_1, ..., \beta_p)^T$
- σ_ϵ^2
- σ_η^2
- ϕ
- υ

La búsqueda se hará para encontrar una aproximación que sea buena. La bondad de la aproximación se mide con el PMCC (Predictive Model Choice Criteria), que es un dato proporcionado por las funciones de la librería.

Descripción del dataset utilizado de ejemplo en la librería

Para estudiar una serie espacio temporal, necesitamos, al menos, varias coordenadas geográficas (2 en este caso: latitud y longitud) y una o varias escalas temporales. Un modelo con 2 escalas temporales, por ejemplo, sería: (año, semana). Para realizar un estudio previo, la librería utiliza el dataset de taxis de Nueva York (NYdata) que se encuentra dentro de la librería spTimer. A continuación se cargan y visualizan los datos:

s.index <int>	Longitude <dbl>	Latitude <dbl>	Year <int>	Month <int>	Day <int>	o8hrmax <dbl>	cMAXTMP <dbl>	WDSP <dbl>	RH <dbl>
1	-73.757	42.681	2006	7	1	53.88	27.85772	5.459953	2.766221
1	-73.757	42.681	2006	7	2	57.13	30.11563	8.211767	3.197750
1	-73.757	42.681	2006	7	3	72.00	30.00001	4.459581	3.225186
1	-73.757	42.681	2006	7	4	36.63	27.89656	3.692225	4.362334
1	-73.757	42.681	2006	7	5	42.63	25.65698	4.374314	3.950320
1	-73.757	42.681	2006	7	6	30.88	24.61968	4.178086	3.420533

Figura 6.3: Visualización de los datos de taxis de Nueva York

Además de las columnas principales (geográficas y temporales), el dataset presenta otras columnas de interés para la predicción. Por orden, son: concentración de ozono, temperatura máxima, velocidad del viento y humedad relativa. Como conclusión, un dataset para este tipo de problemas debe tener:

- Una o varias columnas geográficas. Normalmente suelen ser dos: latitud y longitud, aunque podrían ser otras.
- Una o varias columnas temporales. Se pueden definir varios niveles dentro de esta variable, para hacer referencia a modelos estacionales:
  - Año
  - Semana
  - Día
  - Hora

Estos dos bloques serían los principales. A ellos, se añadirían columnas que representen una característica o propiedad del fenómeno a representar, que ocurra en unas coordenadas geográficas determinadas y en un tiempo determinado. Estas columnas son las que deben agregarse en función de la granularidad que se defina, principalmente, en las columnas temporales.

Finalmente, se hace la predicción sobre una de estas columnas agregadas que hemos añadido.

Fitting

Para realizar el fitting, spTimer muestrea el dataframe con *muestreo de Gibbs* (muestreo utilizado para obtener muestras aleatorias de una distribución conjunta de dos o más variables aleatorias). Después, hace el fitting según los parámetros que le indiquemos. De entre todos los parámetros que hay, los más interesantes para empezar a arrancar el modelo son:

- Cuál es la parte regresiva del modelo (parámetros β). Se hace a través de un elemento "fórmula" de R.
- Los datos de origen, con todas las columnas (*data=*)
- Tipo de modelo: GP, AR o GPP. Es importante destacar que el modelo GPP requiere hacer operaciones adicionales con los datos. GP y AR son más directos. La diferencia entre GP y AR es que, si bien ambos parten de modelos gaussianos, AR incorpora un término de autorregresión (ρ) que GP no tiene.
- Coordenadas geográficas
- Coordenadas temporales

Después, se puede añadir complejidad al modelo con el resto de opciones. Por ejemplo, se pueden definir los priors de los parámetros. Por ejemplo, con el dataset de NY:

```
# Coordenadas temporales
time.data <- spT.time(t.series=60,segment=1)

# Distribuciones a priori de los parámetros
priors <- spT.priors(model="GP",inv.var.prior=Gamm(2,1),beta.prior=Norm(0,10^4))

# Valores iniciales de los parámetros del modelo
initials <- spT.initials(model="GP", sig2eps=0.01,sig2eta=0.5, beta=NULL, phi=0.001)

# Decaimiento espacial
spatial.decay <- spT.decay(distribution=Gamm(2,1), tuning=0.08)

model <- spT.Gibbs(formula=o8hrmax~CMAXTMP+WDSP+RH,
                   data=NYdata, model='GP',
                   coords=~Longitude+Latitude,
                   #time.data=time.data,
                   priors=priors,
                   initials=initials,
                   spatial.decay=spatial.decay)
summary(model)
```

Figura 6.4: Ejemplo de código con los datos de taxis de Nueva York

```
 Output: GP models
-------------------------------------------------------------
 Sampled: 5000 of 5000, 100.00%.
 Batch Acceptance Rate (phi): 30.59%
 Checking Parameters:
   phi: 0.0140, sig2eps: 2.5983, sig2eta: 116.7942
   beta[1]: -19.5990   beta[2]: 2.3843   beta[3]: 1.4166   beta[4]: -0.8966
-------------------------------------------------------------
##
# nBurn =  1000 , Iterations =  5000 .
# Overall Acceptance Rate (phi) =  30.58 %
##
##
# Elapsed time: 8.84 Sec.
##

# Model: GP
-----------------------------------------------------
Model: GP
Call: o8hrmax ~ CMAXTMP + WDSP + RH
Iterations: 5000
nBurn: 1000
Acceptance rate for phi (%): 30.58
-----------------------------------------------------
       Goodness.of.fit  Penalty     PMCC
values:        46290.96 174029.2 220320.2
-----------------------------------------------------
Computation time: 8.84  - Sec.
-----------------------------------------------------
Parameters:
                Mean   Median      SD  Low2.5p  Up97.5p
(Intercept) -17.4338 -17.4952  7.0529 -30.6919  -3.0863
CMAXTMP       2.3209   2.3296  0.2054   1.8768   2.7046
WDSP          1.6179   1.6182  0.3111   0.9863   2.2070
RH           -1.3938  -1.4523  0.8632  -2.9773   0.5237
sig2eps       2.3828   2.2891  0.7698   1.1837   4.1071
sig2eta     121.2027 113.3693 29.0799  89.2494 198.8153
phi           0.0124   0.0127  0.0024   0.0067   0.0165
-----------------------------------------------------
```

Figura 6.5: Resultado de ejecución de código con los datos de taxis de Nueva York

Parte predictiva

En la fase predictiva, spTimer utiliza la función *predict* que requiere dos argumentos:

- *Newcoords*: debe contener los nuevos puntos de coordenadas donde queremos predecir
- *Newdata*: que contiene los valores de las covariables

El tipo de argumento en la predicción puede ser "espacial" o "temporal". Si el valor es "espacial", solo se realizará la predicción espacial en el *newcoords* que deben ser diferentes de los sitios ajustados proporcionados por el argumento de *coords*. Cuando se especifica la opción "temporal", se realizará el pronóstico y, en este caso, los *newcoords* también pueden contener elementos de los sitios ajustados, en cuyo caso solo se utilizarán datos temporales. Esta función, realizará un pronóstico más allá del último punto de tiempo ajustado.

En cuanto a la medición de la predicción, la librería utiliza la función *spT.validation* que muestra las siguientes métricas:

- **MSE** (mean squared error): $\frac{1}{m}\sum_{i=1}^{m}(\widehat{z}-z_i)^2$
- **RMSE** (root mean squared error): $\sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\left(\frac{d_i-f_i}{\sigma_i}\right)^2}$
- **MAE** (mean absolute error): $\frac{1}{n}\sum_{t=1}^{n}|e_t|$
- **rBIAS** (relative bias): $\frac{1}{m\overline{z}}\sum_{i=1}^{m}(\widehat{z}-z_i)$
- **rMSEP** (relative mean separation): $\frac{\sum_{i=1}^{m}(\widehat{z}-z_i)^2}{\sum_{i=1}^{m}(\overline{z}_p-z_i)^2}$

Dataset parking en Seattle para modelos spTimer

En las pruebas que hemos realizado con los modelos spTimer para nuestro dataset, para poder hacer una comparativa hemos desechado la parte espacial del modelo y nos hemos centrado en la parte temporal. El dataset de prueba contiene 30 parquímetros de la ciudad de Seattle con mayor ocupación. En el notebook *'Modelo GP_ SpTimer.Rmd'* está el modelo para nuestro dataset quedaría de la siguiente manera:

```
set.seed(11)
post.gp <- spT.Gibbs(formula=occupation_perc ~Hour+tmin+road_temp+monday+tuesday+wednesday+thursday+friday,
                     data=DataFit,
                     model="GP",
                     coords=~longitude+latitude,
                     distance.method="geodetic:km",
                     scale.transform="SQRT",
                     tol.dist=0.001,
                     spatial.decay=spT.decay(distribution=Gamm(2,1),tuning=0.01))
print(post.gp)
```

Figura 6.6: Código spTimer para nuestros datos

Para nuestras pruebas, hemos probado con la combinación de diferentes modelos con los métodos de transformación para la variable de respuesta (*model* y *scale.transform*). De esta manera hemos probado los modelos GP y AR y los métodos de transformación de la variable respuesta SQRT y NONE, este último, el default del modelo. Destacar que hemos descartado el modelo GPP en nuestras pruebas. Este modelo trata de definir los efectos aleatorios $\eta(s_i,t)$ en un número menor, m, de ubicaciones, llamados *knots* y luego utiliza el método *kriging* para predecir esos efectos aleatorios en los datos y ubicaciones de predicción. En la predicción, hemos tomado la parte temporal del modelo para poder así hacer una comparativa con otros. En esta parte, spTimer incluye el plotting de librería forecast y definiendo el "site" que es la posición del *element_key* podemos presentar la predicción con gráficos como el siguiente:

6.2. Modelo descartado: HTS

HTS es una librería que trata las series temporales relacionadas con estructuras jerárquicas de los datos. Estas estructuras pueden ser dimensiones geográficas, de producto, de género, etc. y son frecuentes en datasets fruto de recopilación de datos. Estos datos, pueden tener una estructura jerárquica única, o no. Como ejemplo de estructura jerárquica, la documentación de la librería pone de ejemplo datos macroeconómicos donde la renta de un país se desagrega de la siguiente manera: $Y = C + G + I + X - M$, donde C es el consumo, G el gasto, I la inversión, X las exportaciones

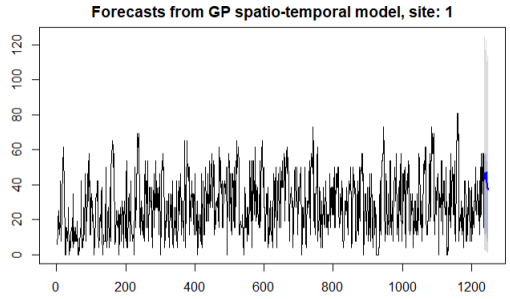

Figura 6.7: Representación de la predicción con spTimer

y M las importaciones. Para una estructura agrupada no jerárquica, el ejemplo indicado es la mortalidad de Australia desagregada por género y a su vez por estados. Pero esta jerarquía no es única, la podemos invertir. En la literatura estadística existen 3 enfoques en el pronóstico jerárquico de series temporales:

- "Top-down" o de arriba hacia abajo.
- "Bottom-up" o de abajo hacia arriba.
- "Middle-out" o de enfoque de aproximación de media salida.

Estos métodos, hacen pronósticos en los diferentes niveles y van agregando para obtener un pronóstico mayor. El modelo de HTS, implementa estos modelos tomando en cuenta la correlación entre las series en cada nivel. En el ejemplo práctico de HTS, se trata una serie temporal con la mortalidad infantil de Australia de 1933-2003 con 2 etiquetas, la de género (male or female) y por estados del país tomandose 8, lo que resulta 16 series en el nivel inferior (bottom). El resultado de la predicción por niveles para los siguientes 10 años estaría representado en estos gráficos:

Esta librería nos ha resultado de interés, pero habiendo hecho un estudio de ésta junto con las posibles combinaciones jerárquicas de las variables de nuestro dataset, hemos descartado su prueba e inclusión en la modelización.

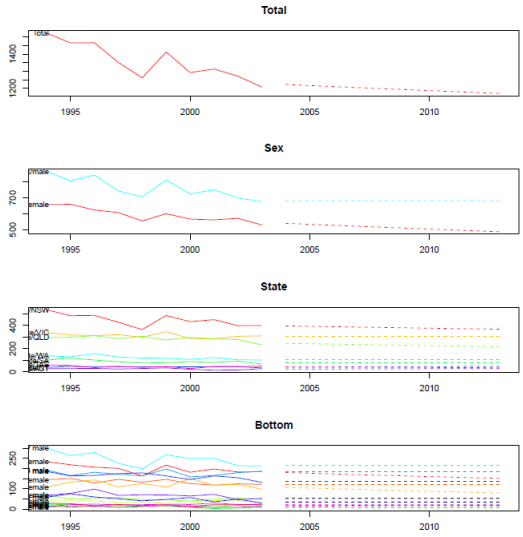

Figura 6.8: Predicciones de ejemplo con la librería HTS

6.3. Aplicación de los modelos

Para comparar todos los modelos mencionados con sus distintas variantes (sobre la serie original, o transformada logarítmicamente, o con regresores), utilizamos una técnica llamada *Day Forward-Chaining* [39].

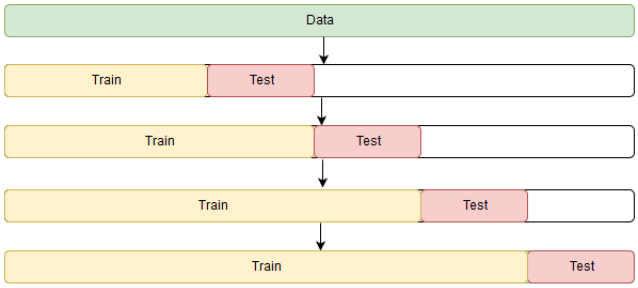

Figura 6.9: Técnica Day Forward-Chaining

Partimos de nuestra serie temporal acotada al primer trimestre del año con 936 observaciones, correspondientes a 78 días hábiles. Comenzamos considerando los 30 primeros días hábiles de la serie (30 días * 12 horas = 360 observaciones) como el primer conjunto de entrenamiento de los modelos, realizamos la primera predicción de las 12 horas del día siguiente, y calculamos el MAE (*Mean Absolute Error*) asociado. En el siguiente paso el conjunto de entrenamiento aumenta en 12 observaciones más de la serie y repetimos el proceso con la predicción de las 12 horas del día siguiente y cálculo del MAE asociado. De esta forma el conjunto de entrenamiento crece con cada

iteración hasta tener un tamaño final igual a 936 - 12 = 924 observaciones y se repite el proceso por última vez con la predicción de las últimas 12 horas del trimestre y cálculo del MAE asociado. Finalmente obtenemos un vector con un total de 48 valores de MAE distintos y calculamos su MAD (*Median Absolute Deviation)* y su Media winsorizada al 5 % superior para descartar posibles outliers. El valor de la Media winsorizada asociada a cada modelo y parquímetro será el parámetro que utilizaremos para comparar los modelos entre sí.

En el caso de los componentes del modelo BSTS se considera en cada iteración que las últimas 12 observaciones del conjunto de entrenamiento se utilizan para validación y selección de los mejores hiperparámetros del modelo. Y en el caso de spTimer de forma análoga para seleccionar el mejor submodelo entre GP y AR. En la Figura 6.10 se muestra el diagrama asociado.

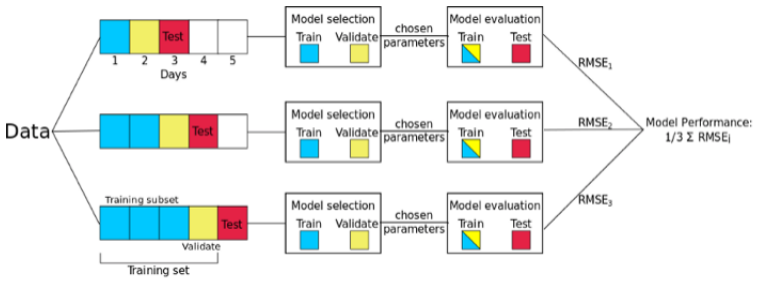

Figura 6.10: Técnica Day Forward-Chaining con validación (para BSTS y spTimer)

Y en la Figura 6.11 representamos para un parquímetro los 48 valores de MAE obtenidos para cada uno los tres mejores modelos, en color negro el mejor modelo, en color azul el segundo y en color verde el tercero y las líneas horizontales en esos tres colores son el valor de la media winsorizada asociada.

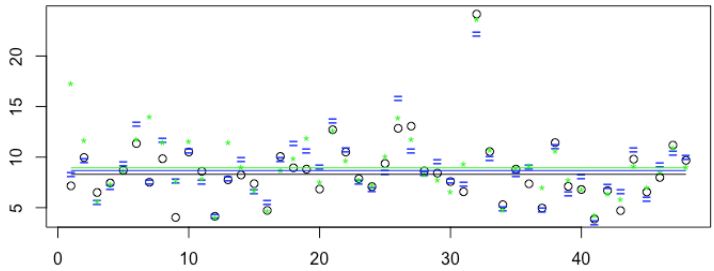

Figura 6.11: MAEs y media winsorizada de tres mejores modelos de un parquímetro seleccionado

Capítulo 7

Evaluación de los modelos predictivos

Por cada uno de los 30 parquímetros seleccionados para el análisis comparativo de los modelos de predicción hemos evaluado un total de 23 submodelos distintos:

- 4 variaciones del modelo Auto-arima (con y sin regresores; con y sin transformación logarítmica)
- 2 variaciones del modelo de Medias móviles (con y sin transformación logarítmica)
- 2 variaciones del modelo MSTL (con y sin transformación logarítmica)
- 4 variaciones de los modelos BATS y TBATS (cada uno con y sin transformación logarítmica)
- 4 variaciones de los modelos HW y DSHW (cada uno con y sin transformación logarítmica)
- 6 variaciones del modelo BSTS (3 componentes distintos, cada uno con y sin regresores)
- 1 modelo spTimer con elecciones internas del mejor submodelo entre AR o GP

Definimos el mejor modelo como aquel que tiene el menor valor de la media winsorizada al 5 % de los MAEs obtenidos en el proceso de validación nesteada comentado en el capítulo anterior.

En la Figura 7.1, extraída del notebook *'Resultados-fusion-analisis.ipynb'*, mostramos cuáles son los mejores modelos y sus resultados asociados para cada uno de los 30 parquímetros seleccionados. Los resultados están ordenados por la columna número de ceros, que cuenta los valores igual a 0 del porcentaje de ocupación, variable objetivo de la predicción y que hemos acotado la serie objetivo del análisis al primer trimestre del año para reducir el coste computacional. Además del predominio de las variaciones de los modelos BATS y TBATS ya mencionadas, vemos también que las transformaciones logarítmicas del modelo BATS y del modelo Holt-Winters son muy útiles para mejorar los resultados en aquellos parquímetros con mayor número de ceros en la variable objetivo de la predicción.

element_key	model	win_mean	mad	total_spaces	paid_parking_area	num_zeros
12289	tbats	11.245146	2.643076	15	Uptown	2
30698	bats	8.769529	2.862264	22	Commercial Core	3
24557	tbats	9.464568	2.882464	18	12th Avenue	7
76961	tbats	9.988782	2.819590	21	Ballard	15
79741	tbats	7.239283	1.857130	22	Pike-Pine	16
37177	LLSR	10.748729	2.474027	13	Pike-Pine	18
14677	tbats	8.184437	1.948148	18	Pike-Pine	19
11878	tbats	8.071804	1.891939	17	Pike-Pine	21
1037	tbats	9.286866	2.224751	26	Uptown	25
1046	tbats	8.403155	2.594322	28	Uptown	28
9510	tbats	6.488121	1.416762	32	University District	28
63125	bats	5.629876	1.745920	51	University District	35
34938	tbats	6.705976	1.673005	24	Uptown	37
37134	tbats	9.135864	1.996568	18	Pike-Pine	38
76962	tbats	8.566481	2.148615	20	Ballard	39
85381	tbats	10.236709	2.766473	17	Green Lake	43
1045	dshw	6.851390	1.962704	26	Uptown	44
85385	mstl	7.400131	1.661113	25	Green Lake	48
59958	tbats	8.893170	1.996802	18	Pike-Pine	57
35682	bats	7.164249	2.187295	33	Ballard	156
18622	bats-log	5.664481	1.850576	35	Ballard	158
35681	bats	7.392548	2.511494	31	Ballard	159
86449	bats-log	5.333876	1.903536	32	Ballard	159
1433	mstl	5.236297	1.597779	41	University District	161
41030	bats-log	6.139373	1.345629	37	Ballard	169
69098	ma	6.534231	2.160729	41	Pioneer Square	171
13549	hw-log	6.636055	1.661377	24	South Lake Union	199
13793	hw-log	8.300638	2.295736	17	South Lake Union	209
81117	bats-log	5.711554	1.265581	25	South Lake Union	220
62458	bats-log	5.167819	1.486762	26	Roosevelt	348

Figura 7.1: Resultados de los mejores modelos para los 30 parquímetros seleccionados

Se observa también que hay gran variabilidad entre los distintos parquímetros para los valores mínimos de la media obtenidos, desde 5.16 hasta 11.245, siendo curioso que los valores extremos mencionados aparezcan para los parquímetros con más ceros (menor media) y menos ceros (mayor media), ya que como se puede comprobar en la Figura 7.2 no hay una relación para el resto de los parquímetros, salvo que los valores están más acotados en rango para aquellos parquímetros con mayor número de ceros. Los valores del MAD (desviación absoluta media) están bastante acotados entre 1.26 y 2.88, indicando que hay homogeneidad en los resultados.

En la Figura 7.3 se muestra la distribución agrupada por familias de modelos. En más de un 76 % de los casos (23 sobre 30 parquímetros) los menores valores de MAE se han obtenido con la familia de modelos TBATS y BATS. Para el resto destacan los modelos HW y DSHW con un 10 % de los casos y los modelos MSTL (6 %, sólo 2 parquímetros). Los modelos Medias móviles y BSTS con componente LocalLevel sin regresores aparecen con 1 parquímetro cada uno.

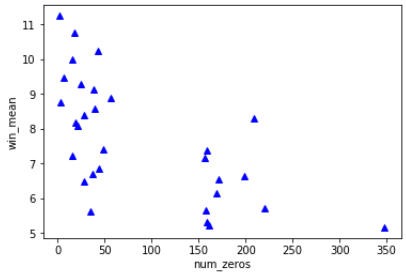

Figura 7.2: Relación entre la media mínima de los valores de MAE obtenidos y el número de ceros en la serie

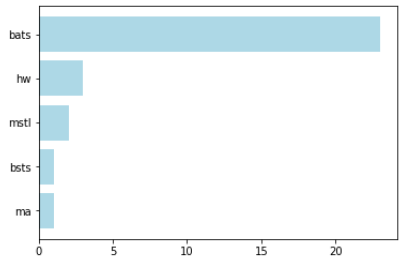

Figura 7.3: Distribución de los mejores modelos para los 30 parquímetros seleccionados

Por otra parte, comparando los valores de la media obtenida con el mejor modelo frente a la media obtenida con el segundo mejor modelo, vemos que para la mitad de los parquímetros los dos mejores modelos pertenecen a la misma familia, y que para tres parquímetros concretos apenas hay una diferencia de centésimas entre los valores de los dos mejores modelos, aunque en esos tres casos los dos mejores modelos pertenecen a familias distintas. Mostramos los resultados obtenidos para los 3 parquímetros en esa situación:

element_key	model	win_mean	mad	total_spaces	paid_parking_area	num_zeros
1045	dshw	6.851390	1.962704	26	Uptown	44
1045	bats	6.869928	1.849296	26	Uptown	44

Figura 7.4: Resultados de los dos mejores modelos para el parquímetro con id 1045

element_key	model	win_mean	mad	total_spaces	paid_parking_area	num_zeros
62458	bats-log	5.167819	1.486762	26	Roosevelt	348
62458	dshw	5.199932	1.201228	26	Roosevelt	348

Figura 7.5: Resultados de los dos mejores modelos para el parquímetro con id 62458

element_key	model	win_mean	mad	total_spaces	paid_parking_area	num_zeros
69098	ma	6.534231	2.160729	41	Pioneer Square	171
69098	ArSR	6.536458	1.783394	41	Pioneer Square	171

Figura 7.6: Resultados de los dos mejores modelos para el parquímetro con id 69098

Ampliando la evaluación a los 3 mejores modelos por cada parquímetro, obtenemos que se mantiene el predominio de la familia TBATS y BATS pero se reduce al 60 % de los casos, mientras que la familia Holt-Winters aparece destacada en un 22 % de los casos, un 12 % para la familia MSTL y un 5 % para los modelos BSTS.

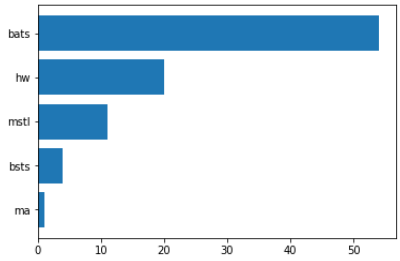

Figura 7.7: Distribución de los tres mejores modelos para los 30 parquímetros seleccionados

Y considerando los 3 mejores modelos encontramos algunas curiosidades:

- no hay ningún parquímetro que no tenga un modelo de la familia BATS y TBATS entre los 3 mejores
- hay 3 parquímetros en los que sus 3 mejores modelos son de la misma familia, BATS y TBATS (ids 18622, 30698 y 34938)
- hay 1 parquímetro donde sus 3 mejores modelos utilizan la serie transformada logarítmicamente (id 13793)

Recordamos que para cada parquímetro hemos analizado 23 modelos, que podemos ordenar en un ranking de 0 (mejor modelo) a 22 (peor modelo). Si calculamos la posición media de cada modelo en el listado ordenado de los mejores modelos por parquímetro, obtenemos los resultados que se muestran en la Figura 7.8 donde observamos que:

- el modelo TBATS tiene un mejor rendimiento que BATS
- DSHW también supera al modelo HW, algo esperable por la doble estacionalidad de nuestra serie
- los componentes sin regresores de BSTS tienen mejor rendimiento que los que utilizan regresores, y LocalLinearTrend se comporta peor que LocalLevel y Ar
- la transformación logarítmica no es recomendable para el modelo DSHW que pasa de estar en la cuarta posición a la última

modelo	ranking_medio
tbats	2.066667
bats	3.266667
mstl	4.200000
dshw	4.400000
tbats-log	5.166667
bats-log	5.400000
hw	5.766667
hw-log	6.566667
mstl-log	6.933333
LLSR	9.666667
ma	10.133333
ArSR	10.733333
auto-arima-regr	11.433333
auto-arima	12.133333
ma-log	13.100000
auto-arima-log	13.366667
auto-arima-regr-log	13.966667
LLTSR	15.500000
sptimer	17.700000
ArCR	19.633333
LLCR	19.800000
LLTCR	20.233333
dshw-log	21.833333

Figura 7.8: Posición media de cada modelo en el listado de mejores modelos por parquímetro

Capítulo 8

Conclusiones y casos de uso

8.1. Conclusiones

Para finalizar y una vez expuesto en este documento el trabajo desarrollado, llegamos a las siguientes conclusiones:

- Para una mayor comprensión y entendimiento del indicador del porcentaje de ocupación, se ha enriquecido el dataset de origen con datos externos, tales como: datos meteorológicos (precipitaciones, temperatura máxima y mínima, temperatura del ambiente y del asfalto); puntos de interés (eventos culturales y deportivos, así como la ubicación de sus monumentos, espacios culturales e instalaciones deportivas); y variables relacionadas con la calidad del aire (monóxido de carbono, ozono, etc.). En el análisis de correlaciones en el que hemos establecido el nivel de significación en el 5 %, no ha arrojado evidencias de que haya correlación entre las covariables incluidas en el estudio y el target.

- En la modelización, se han probado un total de 23 submodelos de las librerías Forecast, BSTS y spTimer. Hemos probado combinaciones con y sin transformaciones logarítmicas, incluyendo o no los regresores, dependiendo de la flexibilidad y particularidades de cada modelo. Para su evaluación, tomamos como mejor modelo el de menor valor de la media winsorizada al 5 % de los MAEs obtenidos en el proceso de validación nesteada. En su evaluación, podemos concluir que los modelos sencillos ganan a los modelos más complejos. Por otro lado, aunque hemos obtenido mejores resultados con los modelos de la librería Forecast, estos no tienen margen de mejora. Los modelos de las librerías BSTS y spTimer son ampliamente configurables, y con un análisis más exhaustivo de sus hiperparámetros, es muy posible que hubiéramos podido mejorar sus rendimientos.

- Este proyecto tiene como objetivo el estudio de los flujos de aparcamiento de Seattle, en concreto, se ha tomado como referencia el indicador del porcentaje de ocupación de los parquímetros situados en su área metropolitana. Entendemos que este estudio se puede extrapolar a otras ciudades del mundo, para la sostenibilidad de éstas y mayor bienestar de sus habitantes. Reseñar que la mitad de la población de nuestro planeta vive en ciudades, y según Naciones Unidas se prevé que en el 2030 este porcentaje se incremente al 60 %. Es por

esto, que nuestro estudio cobra mayor relevancia. Actualmente, la aplicación del Big Data para solucionar el problema de la saturación de zonas de aparcamiento está en pleno auge y desarrollo, pero todavía no ha llegado a su fase de madurez.

8.2. Casos de uso

Un negocio real que podría derivarse de este TFM sería la creación de una app que, utilizando el modelo predictivo TBATS de la librería *forecast* diera una predicción a los usuarios sobre el porcentaje de ocupación de una zona determinada. Por supuesto, somos conscientes que los datos obtenidos para este TFM son datos del pasado, y para poder crear una app que fuera rentable y de utilidad se tendrían que obtener y utilizar datos near-real-time o real-time.

Cabe señalar que nosotros hemos planteado a lo largo del TFM un estudio basado en la predicción de ocupación por horas del día siguiente. Esto podría simplificar la obtención de los datos, ya que los datos necesarios serían los del día anterior para predecir por horas la ocupación del día siguiente. Una primera posible mejora para un caso de negocio real sería la franja objetivo de predicción. En vez de agrupar por horas y dar un porcentaje de ocupación horaria, se podría dar una predicción por minutos, o al menos, por medias horas. Esto daría mucha más riqueza y utilidad a los usuarios.

Se podrían obtener los datos de fuentes muy diversas, pero lo menos complicado sería tener un acuerdo con la compañía responsable de los parquímetros, para asegurar la calidad de los datos. En Seattle lo han conseguido desde la app *PayByPhone* [40], que es una app desde la cual poder hacer los pagos de los parquímetros. Hasta tal punto ha llegado la satisfacción con esta app, que el Departamento de Transportes de Seattle (SDOT) ha ampliado su acuerdo con la misma, eliminando el sobrecoste de 35 céntimos que antes incluía a los usuarios de esta app en el pago de sus tickets [41]. Otra manera que permitiría la obtención de datos podría ser la instalación de sensores, lo cual daría al negocio cierta independencia del proveedor de los datos, aunque la contrapartida inevitable sería el alto coste de los mismos. Con los sensores se podrían tener datos muy actualizados, pero el coste de partida de las infraestructuras necesarias para el funcionamiento de la app sería bastante elevado.

También se podría aprovechar la información de los mismos usuarios, que compartirían con la app su geolocalización y la proveerían de datos. Éstos compartirían su localización y actualizarían el estado de las plazas libres. Sin embargo, ésta no podría ser la única fuente de datos, pues sin usuarios la app no lograría progresar.

Sería conveniente, antes incluso de negociar con la empresa de parquímetros, llevar a cabo un estudio de mercado analizando la demanda estimada que pudiera tener este servicio. Lo lógico sería centrarse en núcleos urbanos superpoblados donde el aparcamiento en la calle suponga una odisea para sus ciudadanos, y donde reine el aparcamiento regulado mediante parquímetros. Además habría que fijarse si estamos ante un nicho de mercado o, si por el contrario, ya existen competidores ofreciendo lo mismo o muy parecido. Por una parte, es positivo llegar los primeros a un mercado porque somos los pioneros y abrimos un servicio, creando una demanda que antes ni se planteaba; además de que disfrutaremos de momentos de tranquilidad solos ante los consumidores, aprovechando esa ventaja para fidelizarlos, por ejemplo. Por otra parte, es innegable que si somos los pioneros asumiremos un riesgo, el riesgo de que no se conozca nuestro producto y tengamos que esforzarnos mucho en darlo a conocer y hacer ver a los ciudadanos las ventajas de su uso; otros competidores siempre aportan riqueza a un negocio, procurando mejoras continuas y agilizando la

oferta de los servicios, por no mencionar que si hay ya competidores significa que existe el mercado y la necesidad ya está creada en los consumidores.

Bibliografía

[1] URL: https://www.ais-int.com/parking-smart-cities-ais-desarrolla-modelo-predictivo-aparcamiento-barcelona/.

[2] URL: https://www.esmartcity.es/2017/07/04/analisis-predictivo-aprendizaje-automatico-encontrar-aparcamiento-ciudad.

[3] URL: https://www.sport.es/es/motor/destacados/noticias/actualidad/movilidad/easypark-lanza-un-servicio-que-te-guia-hasta-el-aparcamiento-6792865.

[4] URL: https://www.opngo.com/es/page/aparcate-con-opngo.

[5] URL: https://www.telpark.com/.

[6] URL: http://www.expansion.com/economia-digital/innovacion/2017/02/11/589de1d7ca4741ac518b461e.html.

[7] URL: https://www.express.co.uk/life-style/cars/827333/Car-park-drivers-looking-for-spaces-cities-research.

[8] URL: http://inrix.com/press-releases/parking-pain-us/.

[9] URL: https://www.economist.com/leaders/2017/04/06/the-perilous-politics-of-parking.

[10] URL: https://apolitical.co/solution_article/los-angeles-cuts-downtown-congestion-smart-parking/.

[11] URL: https://eu.usatoday.com/story/money/2017/07/12/parking-pain-causes-financial-and-personal-strain/467637001/.

[12] URL: https://data.melbourne.vic.gov.au/Transport-Movement/On-street-Car-Parking-Sensor-Data-2016/dj7e-rdx9/data.

[13] URL: https://data.bathhacked.org/Government-and-Society/BANES-Historic-Car-Park-Occupancy/x29s-cczc.

[14] URL: https://github.com/rexthompson/DATA-512-Final-Project.

[15] URL: http://wwwqa.seattle.gov/Documents/Departments/SDOT/ParkingProgram/data/SeattlePaidTransactMetadata.pdf.

[16] URL: http://wwwqa.seattle.gov/Documents/Departments/SDOT/ParkingProgram/data/SeattlePaidBlockfaceMetadata.pdf.

[17] URL: http://gisrevprxy.seattle.gov/arcgis/rest/services/SDOT_EXT/DSG_datasharing/MapServer/54.

[18] URL: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=0ahUKEwjH1Lme2KvcAhXN3KQKHTOMBFkQFgg2MAI&url=https%3A%2F%2Fdata.seattle.gov%2Fapi%2Fassets%2FC57B1665-C730-4847-991C-98EB3F1C717B%3Fdownload%3Dtrue&usg=AOvVaw1-DrNeppTDAh4yBSN_nr4m.

[19] URL: https://www.kaggle.com.

[20] URL: https://www.kaggle.com/rtatman/did-it-rain-in-seattle-19482017.

[21] URL: https://data.seattle.gov/Transportation/Road-Weather-Information-Stations/egc4-d24i.

[22] URL: https://data.seattle.gov/Community/City-of-Seattle-Events/cprz-jsz8.

[23] URL: https://data.seattle.gov/browse?q=sport&sortBy=relevance.

[24] URL: https://data.seattle.gov/Community/Seattle-Cultural-Space-Inventory-Map/pknd-dutm.

[25] URL: https://www.epa.gov/outdoor-air-quality-data/download-daily-data.

[26] URL: http://www2.dmu.dk/atmosphericenvironment/expost/database/docs/ppm_conversion.pdf.

[27] URL: https://pypi.org/project/holidays/.

[28] URL: http://gisrevprxy.seattle.gov/arcgis/rest/services/SDOT_EXT/DSG_datasharing/MapServer/14.

[29] URL: https://www.kaggle.com/dansbecker/permutation-importance?utm_medium=email&utm_source=mailchimp&utm_campaign=ml4insights.

[30] URL: https://cran.r-project.org/web/packages/forecast/forecast.pdf.

[31] URL: http://people.ischool.berkeley.edu/~hal/Papers/2013/pred-present-with-bsts.pdf.

[32] URL: https://rstudio-pubs-static.s3.amazonaws.com/257314_131e2c97e7e249448ca32e555c9247c6.html.

[33] URL: https://rpubs.com/osazuwa/bsts/.

[34] URL: http://sisifospage.tech/docs/MineThatData_ForecastChallenge_whitepaper.pdf.

[35] URL: https://cran.r-project.org/web/packages/bsts/bsts.pdf.

[36] URL: https://stats.stackexchange.com/questions/209426/predictions-from-bsts-model-in-r-are-failing-completely.

[37] URL: https://cran.r-project.org/web/packages/spTimer/spTimer.pdf.

[38] URL: https://www.jstatsoft.org/article/view/v063i15/v63i15.pdf.

[39] URL: https://towardsdatascience.com/time-series-nested-cross-validation-76adba623eb9.

[40] URL: https://www.seattle.gov/transportation/projects-and-programs/programs/parking-program/paid-parking-information/pay-on-your-phone.

[41] URL: https://www.paybyphone.com/news/seattle-no-fee.

[42] G. Casella, L. Berger. *Statistical Inference*. 2002.

[43] G. Kanji. *100 statistical tests*. 2006.

[44] Rob J Hyndman & George Athanasopoulos. *Forecasting Principles and Practice*. 2014.

[45] Scott & Varian. *Predicting the Present with Bayesian Structural Time Series*. 2013.