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 [**Objone**].

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 [**Objtwo**]. 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 [**Objthree**].

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 [**Objfour**].

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 [**Objfive**].

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 [**Backone**]
- en la ciudad de Londres un conductor pierde de media 67 horas al año buscando aparcamiento [**Backtwo**]
- 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 [**Backthree**]

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 [**Backfour**]. En Los Ángeles este nivel de tráfico llega al 30 % [**Backfive**]. 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 [**Backsix**].

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 [**FDone**]. 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 [**FDtwo**].

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 [**FDthree**].

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* [**FDfour**], 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* [**FDfive**], 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 [**FDsix**] [**FDsixb**]. 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 [**FDsixc**] un dataset que contiene información meteorológica histórica desde 1948 hasta 2017 (**DATASET-4**: fichero *'seattleWeather_1948-2017.csv'*) [**FDseven**]. 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'* [**FDeight**]).

	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 [**FDnine**] (**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 [**FDten**]:

- 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 [**FDeleven**] (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 [**FDtwelve**]. 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 [**FDthirteen**] (**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

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 [**ETLtwo**] que contiene el campo *ELMNT-KEY* 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.

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.

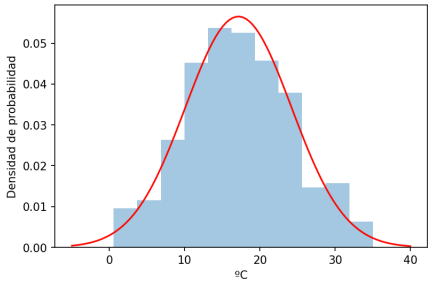

Figura 4.2: Distribución de temperaturas máximas

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 $\alpha = 0{,}05$.

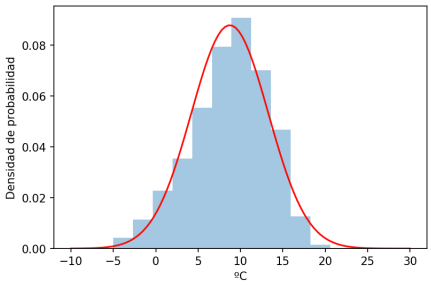

Figura 4.3: Distribución de temperaturas mínimas

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.

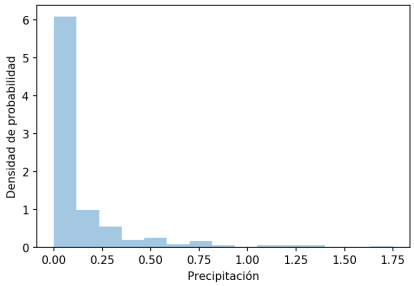

Figura 4.4: Distribución de precipitaciones

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.

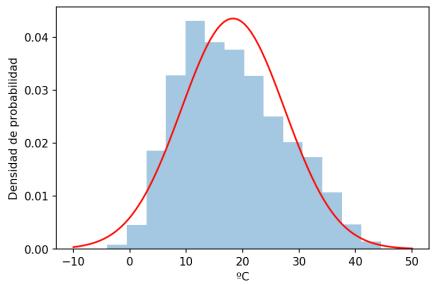

Figura 4.5: Distribución de temperatura media horaria del asfalto

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 ($\alpha = 0{,}05$).

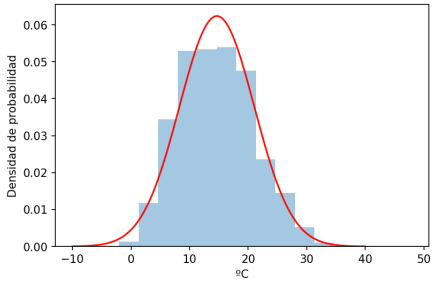

Figura 4.6: Distribución de temperatura media horaria ambiente

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 ($\alpha = 0{,}05$).

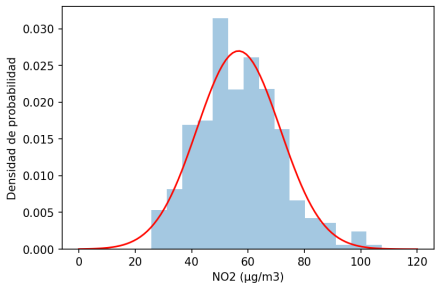

Figura 4.7: Distribución de cantidad de dióxido de nitrógeno

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 ($\alpha = 0{,}05$).

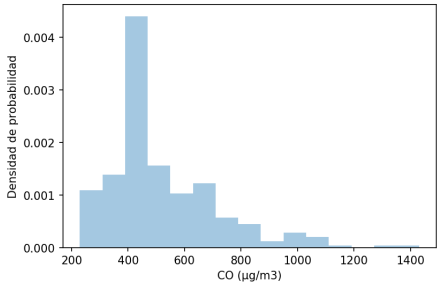

Figura 4.8: Distribución de cantidad de monóxido de carbono

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 ($\alpha = 0{,}05$).

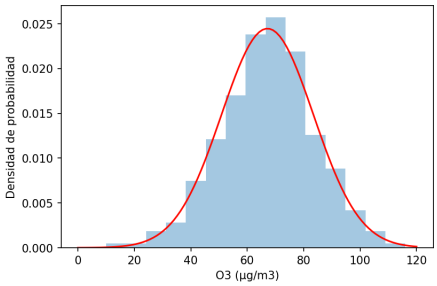

Figura 4.9: Distribución de cantidad de ozono

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 ($\alpha = 0{,}05$).

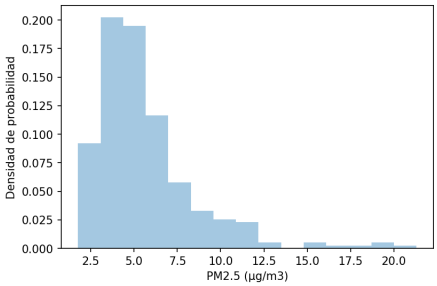

Figura 4.10: Distribución de cantidad de partículas en suspensión

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 %.

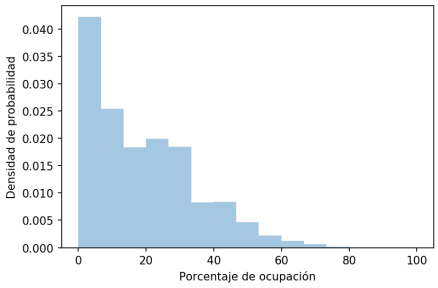

Figura 4.11: Distribución del porcentaje de ocupación medio de los parquímetros

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 (**PDTE poner referencia**), 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 (**PDTE poner referencia**), 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.

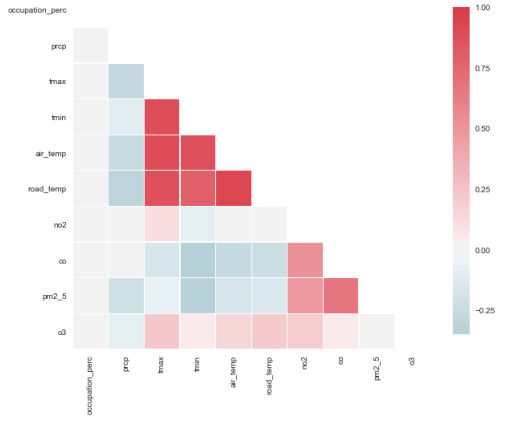

Figura 4.12: Matriz de correlación de las covariables

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).

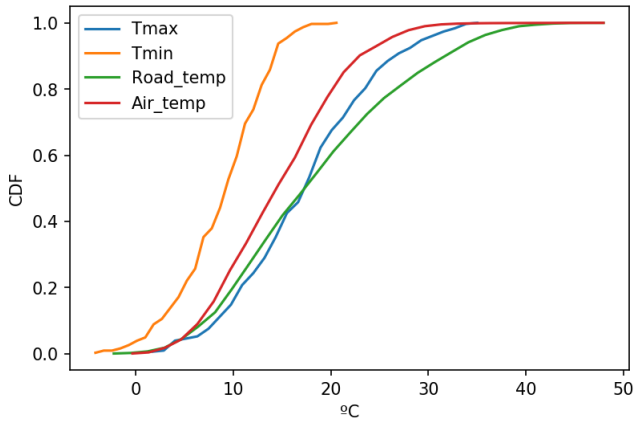

Figura 4.13: Funciones de distribución de las variables de temperaturas

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.

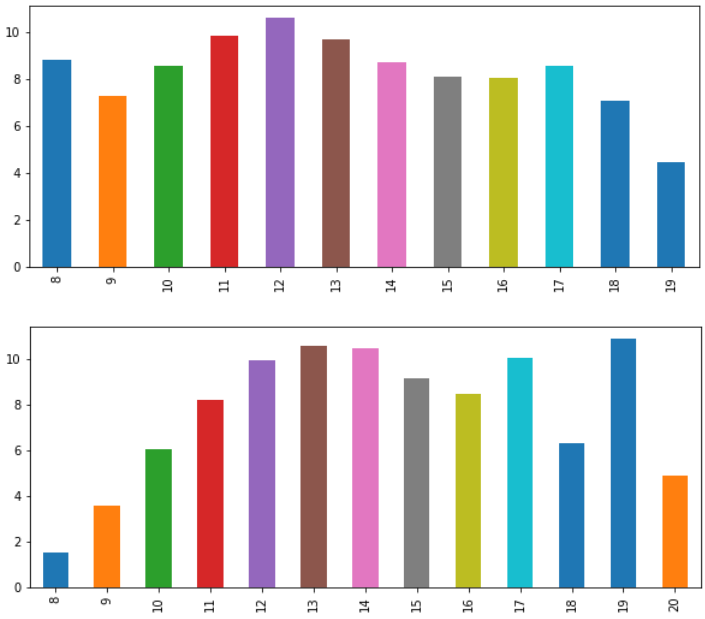

Figura 4.14: Distribución de la hora de inicio y de la hora de fin de las transacciones

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.

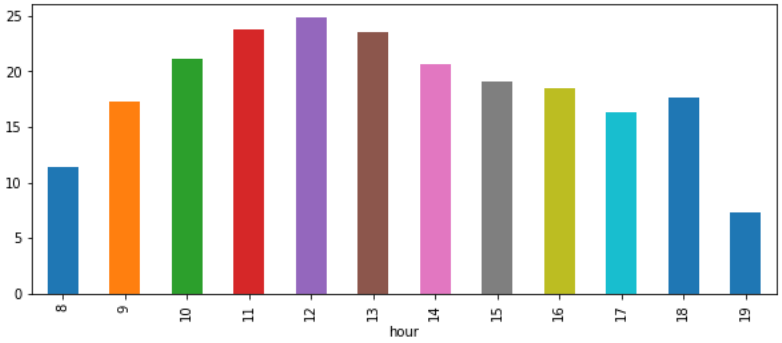

Figura 4.15: Distribución de la ocupación de los parquímetros en función de la hora del día

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.

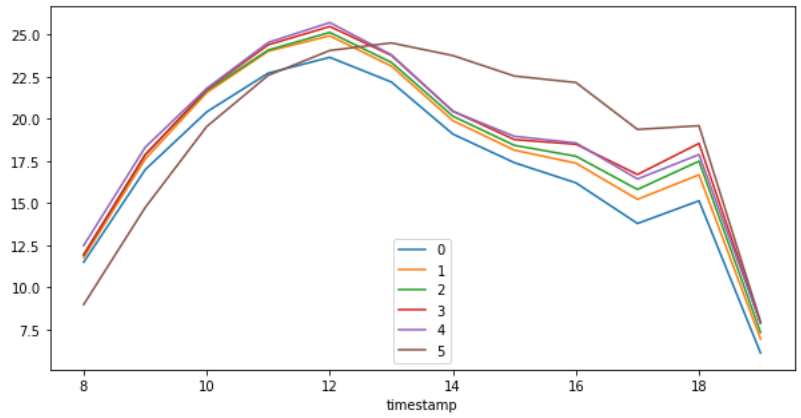

Figura 4.16: Distribución de la ocupación de los parquímetros en función del día de la semana

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.

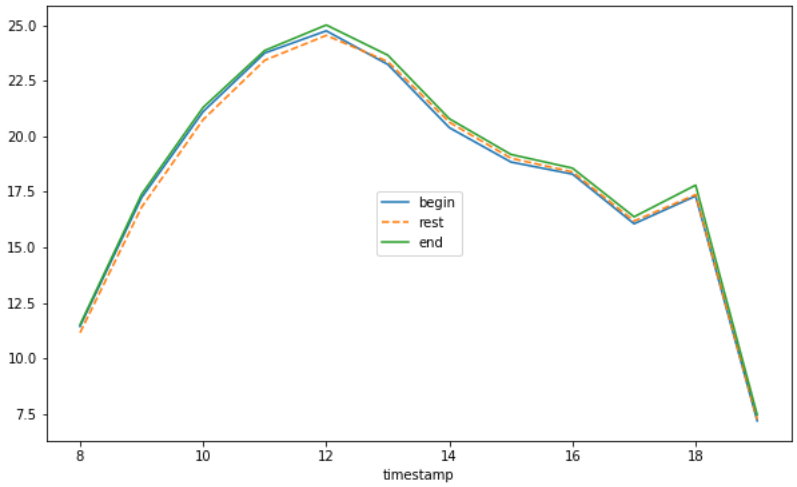

Figura 4.17: Distribución de la ocupación de los parquímetros en función del día del mes

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).

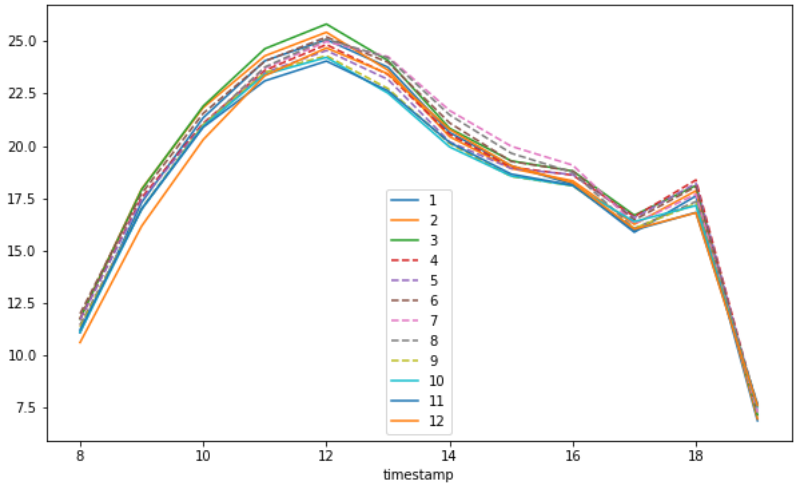

Figura 4.18: Distribución de la ocupación de los parquímetros en función del mes

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:

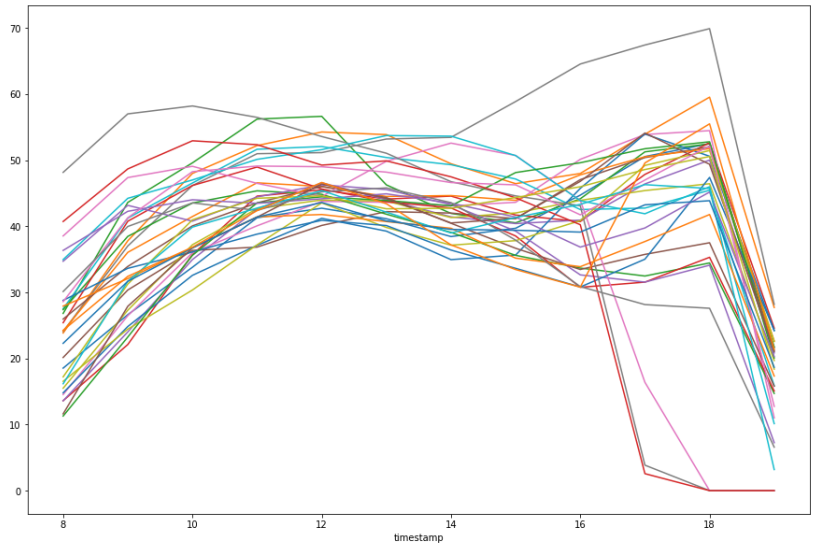

Figura 4.19: Distribución de la ocupación de los parquímetros con mayor porcentaje medio ($> 35\,\%$)

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

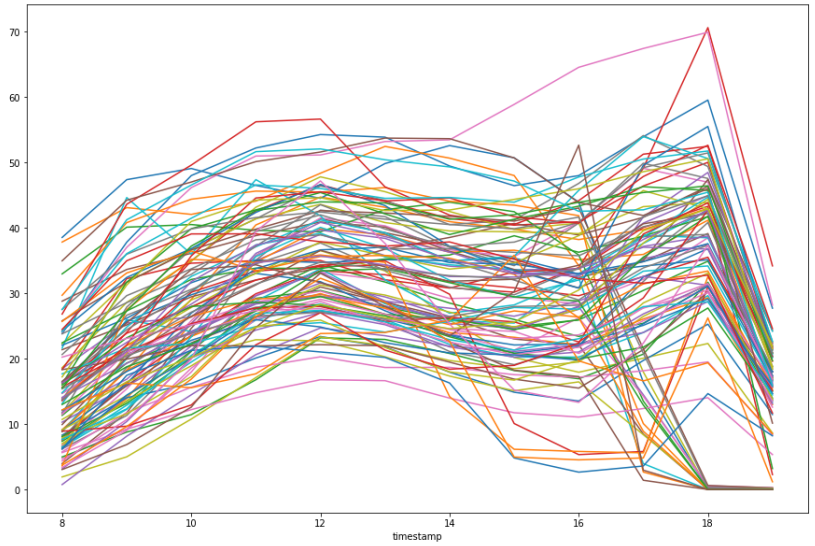

Figura 4.20: Distribución de la ocupación de los 100 parquímetros con más transacciones

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.

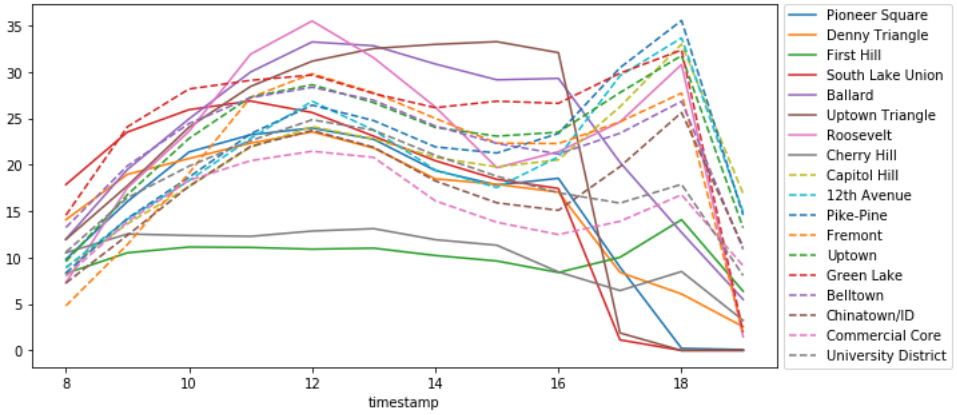

Figura 4.21: Distribución de la ocupación de los parquímetros según su distrito

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:

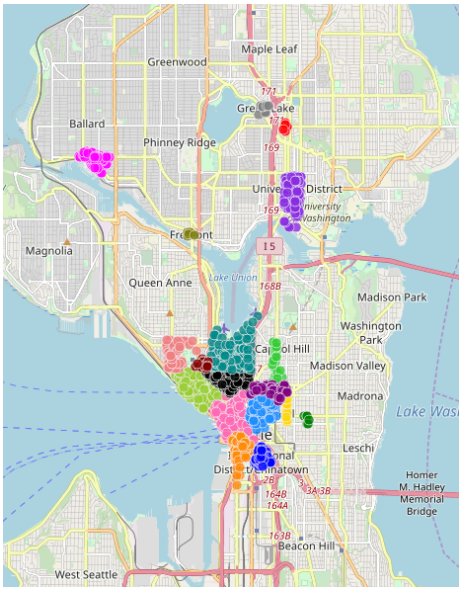

Figura 4.22: Mapa de los parquímetros por distritos

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.

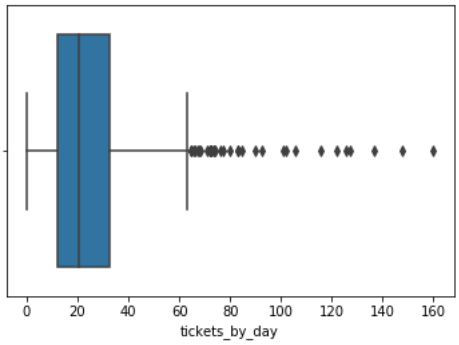

Figura 4.23: Diagrama de caja asociado al número medio de transacciones diarias de los parquímetros

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:

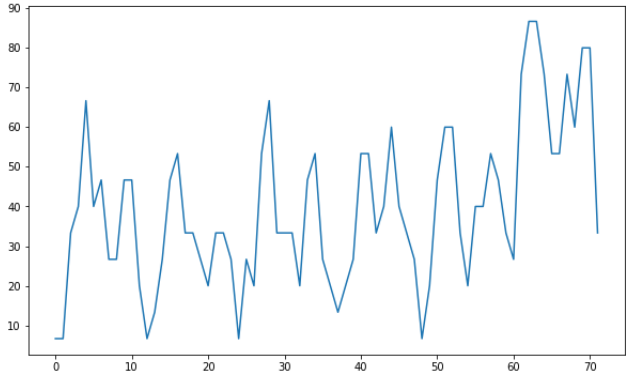

Figura 4.24: Porcentaje de ocupación del parquímetro 12289 durante la primera semana del año

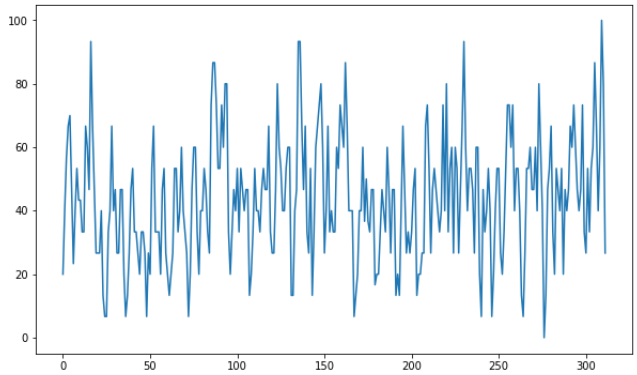

Figura 4.25: Porcentaje de ocupación del parquímetro 12289 durante el mes de Enero

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

Descripción y aplicación de modelos predictivos

5.1. Introducción

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

- 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.

Predicciones ex-ante versus post-ante en modelos de regresión de series temporales: Se distinguen dos tipos de predicción dependiendo de la información que se conoce a la hora de calcular la predicción:

- Ex-ante: utilizan sólo la información que se conoce con anterioridad. En modelos con regresores externos se requiere disponer de predicciones de los regresores.
- Ex-post: utilizan información de los predictores externos posteriores al momento de la predicción. En estos casos las predicciones no son válidas salvo para estudiar el comportamiento de los modelos de predicción en relación a los regresores.

Transformación logarítmica de la serie: 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 compararemos el rendimiento en la predicción de los modelos considerando la serie original y la serie transformada logarítmicamente.

5.2. Estacionalidad múltiple: modelos considerados

[**MODone**] [**MODtwo**] 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. Si acotamos el análisis al primer trimestre para que nuestros equipos informáticos puedan realizar los cálculos en plazos de tiempo razonables, tendríamos 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, se utiliza la clase *msts* de R que nos permite especificar todas las frecuencias que son relevantes e incluso admite frecuencias no enteras.

5.2.1. Modelo auto-arima

El modelo *auto-arima* ajusta la serie combinando valores de p, d y q y selecciona el modelo *ARIMA* que tiene el menor estadístico AIC, que sería el mejor modelo *ARIMA*. El estadístico que utilizamos es el AIC corregido. Hemos querido probar si la predicción mejoraba si establecíamos un lambda distinto al NULL que viene por defecto. Hemos probado lambda = "auto" para que se seleccione automáticamente una transformación BoxCox. Las transformaciones BoxCox 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 que menos ceros tiene de ocupación, el 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 sale por tanto perjudicado y ésta es la razón por la que hemos optado por dejar el valor por defecto y no proceder a ninguna transformación BoxCox en este punto. Seleccionamos seasonal=TRUE para ser coherentes con la elección del modelo multiseasonal de msts.

También hemos querido estudiar cómo se comporta un modelo que incluya *regresores* en el modelo *auto.arima*, por lo que los hemos definido en "xreg" y hemos seleccionado únicamente tres: la temperatura de la carretera, los días de la semana y la temperatura mínima. Esta selección no es aleatoria, y es que hemos querido seguir los resultados obtenidos en el apartado de análisis exploratorio de los datos (EDA). "Xreg" tiene que tener las mismas filas que la serie temporal, pero esto no es un obstáculo pues una vez que tenemos estos regresores seleccionados, los incluimos en la creación del modelo. Le aplicamos la función estrella "forecast" en la que nos encontramos con un problema, explicado en la documentación de la función; y es que cuando a un modelo creado con *arima* se le especifican *regresores*, la función forecast ignorará el periodo de predicción (h). Lo que nos pasa entonces es que forecast entiende que tiene que predecir tantos periodos como filas tenga el dataset, lo cual no nos serviría para nuestro objetivo de comparar los resultados que obtengamos del MAE con los resultados de otros modelos. Por ello, la solución a la que hemos llegado es que el periodo en que se tenían en cuenta los regresores para predecir sería precisamente el de la predicción, poniendo entonces los datos de los regresores de las 12 horas siguientes a nuestro train. Esta no es la solución óptima, pero era la única solución para poder incluir los regresores y que se nos respetara nuestro periodo de predicción. Tendremos este pequeño detalle en cuenta a la hora de valorar esta predicción.

5.2.2. Modelo de medias móviles

La base del modelo de *medias móviles (Moving-Average)* es que asume que las desviaciones actuales de la media dependen de las desviaciones pasadas. Consiste en la media de valores consecutivos en distintos periodos de tiempo. 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.

5.2.3. Modelo STL

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. *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

5.2.4. Modelo MSTL

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.

5.2.5. Modelos BATS y TBATS

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

- B: transformaciones Box-Cox (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.

5.2.6. Modelo de Holt-Winters

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.

5.2.7. Modelo de DSHW

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.

5.2.8. Modelo BSTS

[**MODthree**] [**MODfour**] [**MODfive**] [**MODsix**] [**MODseven**] El modelo *BSTS* de acuerdo a sus iniciales (Bayesian Structured Time Series) se puede explicar:

- Time series. Se usan principalmente para el pronóstico, aunque los modelos de espacio de estado, para los cuales BSTS es solo una extensión, se han usado tradicionalmente en los filtros activos de ingeniería, y actualmente se usan en una variedad de aplicaciones.

  Modelos de espacio de estado: 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.

- 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.

- Bayesiana. Significa que la implementación utilizada para esta propuesta tiene un enfoque bayesiano. Esto no significa que uno tenga que çonvertirse.[en] estadísticas bayesianas. 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 para expresar el conocimiento previo sobre la serie objetivo a través de los priors bayesianos (que pueden considerarse como hiperparámetros del modelo)

  En sus diferentes formas (modelos de espacio de estado, filtros de Kalman), los BSTS se han utilizado "tradicionalmente"(es decir, desde los años 60), 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 para predecir. Estos componentes capturarán diferentes aspectos de la serie, y se pueden agregar o eliminar según las necesidades, como en un juego de construcción. En la descomposición clásica, se clasifican aproximadamente como:

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

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

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

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

- Componentes trend, que capturan los aspectos de tendencia de la serie
- Componentes seasonal, que captan los aspectos periódicos de la serie
- Componentes regression, que capturan 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 describe en la descripción de la biblioteca BSTS. [**MODeight**]

- Breve descripción del funcionamiento de BSTS

BSTS realiza dos operaciones principales: filtrado y suavizado (filtering and smoothing).

- El filtrado proporciona una predicción one-step de la serie, dados todos los datos disponibles hasta el momento; es decir, en una serie mensual como el objetivo, proporciona la predicción para el próximo mes, dados todos los datos disponibles hasta el mes actual.
- 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; Las predicciones no serán muy útiles para los últimos periodos del horizonte de predicción, si es muy amplio. Es difícil definir un ancho óptimo del horizonte, dependiendo de las características de la serie y la complejidad del modelo.

- 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 el analista traducir el conocimiento del negocio sobre la serie y el dominio a un sigma prior, por ejemplo. Sin embargo, debería ser posible para un analista de negocios (quien quizás sepa qué es la variación, pero de todos modos no tiene idea de lo que es un sigma prior), para averiguar la información necesaria.

Continuemos con esta explicación para el componente de tendencia más simple disponible, el componente de nivel local (LocalLevel), que es un componente de tendencia. En cuanto a la documentación en el paquete R, tiene 2 priors:

- initial.state.prior: describe la distribución anterior del vector de estado inicial (en el momento 1)

- sigma.prior: describe la distribución anterior para la desviación estándar de la aleatoria caminar incrementos

Se puede omitir initial.state.prior, en particular si el analista no está seguro de cuál es el vector de estado. En la mayoría de las situaciones prácticas, el modelo calcula correctamente el valor inicial del vector de estado sin necesidad de agregar información previa.

¿Qué pasa con el sigma.prior? El componente a nivel local describe la tendencia de una manera bastante simple; se puede ver como un suavizado exponencial de los valores previos de la serie. Por lo tanto, el modelo predice el valor del siguiente paso (es decir, para la serie objetivo) como un suavizado de los valores pasados. Lo anterior se puede ver entonces como un parámetro para este suavizado: determina qué tan ajustado es el componente de tendencia que seguirá los últimos valores de la serie. Un valor alto de sigma indica que el componente seguirá la serie con firmeza, y un valor bajo indica lo contrario.

Cuando no se tiene claro los priors se puede hacer validación cruzada con las desventajas que esto conlleva (tiempo excesivo, soluciones buenas pero quizás no las mejores etc.)

- Aplicación de BSTS al TFM

BSTS aporta los siguientes componentes:

- Componentes trend
  - AddAr
  - AddAutoAr
  - AddLocalLevel
  - AddLocalLinearTrend
  - AddStudentLocalLinearTrend
  - AddGeneralizedLocalLinearTrend

  Uno puede pensar que un linear trend como una relación de regresión entre la serie y el tiempo. Para un cambio en el tiempo Δ, la respuesta debería cambiar $\beta_1 * \Delta_1$.
- Seasonal components. Los componentes estacionales capturan la periodicidad en los datos. Las opciones incluyen:
  - AddTrig
  - AddSeasonal
  - AddNamedHolidays
  - AddFixedDateHoliday
  - AddNthWeekdayInMonthHoliday
  - AddLastWeekdayInMonthHoliday

Se han seleccionado **AddLocalLevel**, **AddAr** , **AddLocalLinearTrend** para establecer comparativas aun sabiendo que algunos no son los más adecuados basándose en la respuesta del propio Scott: [**MODnine**]

"El modelo ***LocalLinearTrend*** *es muy flexible, pero esta flexibilidad puede aparecer como una varianza no deseada en los pronósticos a largo plazo. Hay algunos otros modelos de tendencias que contienen un poco más de estructura.* ***GeneralizedLocalLinearTrend*** *(perdón por el*

nombre no descriptivo) asume que el componente "slope"de la tendencia es un proceso AR1 en lugar de un random walk. Es mi opción por defecto si quiero pronosticar en el futuro. La mayoría de las variaciones de tus series de tiempo parecen provenir de la estacionalidad, por lo que puedes probar ***AddLocalLevel*** *o incluso* ***AddAr*** *en lugar de* ***AddLocalLinearTrend*** *"*

Además, por cada componente se realiza su aplicación sin regresores y con regresores y se establecen comparativas entre los diferentes métodos.

- Local linear trend state component

  Agrega un modelo de tendencia lineal local a una especificación de estado. El modelo de tendencia lineal local asume que tanto la media como la pendiente de la tendencia siguen random walk. 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.

- AR(p) state component

  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 retrasos del α. La matriz de transición de estado tiene ϕ en su primera fila, unos a lo largo de su primer subdiagonal y ceros en otros lugares. La matriz de varianza del estado tiene σ^2 en su esquina superior izquierda y es cero en otras partes. La matriz de observación tiene 1 en su primer elemento y es cero en caso contrario.

- Local level trend state component Agrega un modelo de nivel local a una especificación de estado. El modelo a nivel local asume que la tendencia es un random walk:

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

```r
AddLocalLevel_sin_regresores(ek)
AddLocalLevel_con_regresores(ek)
AddAr_sin_regresores(ek)
AddAr_con_regresores(ek)
AddLocalLinearTrend_sin_regresores(ek)
AddLocalLinearTrend_con_regresores(ek)

resultados_ek <- c()
resultados_AddLocalLevel_sin_regresores(ek)
resultados_AddLocalLevel_con_regresores(ek)
resultados_AddAr_sin_regresores(ek)
resultados_AddAr_con_regresores(ek)
resultados_AddLocalLinearTrend_sin_regresores(ek)
resultados_AddLocalLinearTrend_con_regresores(ek)

CompareBstsModels(list("Sin regresores" = modelo_AddLocalLevel_sin_regresores,
                       "Con regresores" = modelo_AddLocalLevel_con_regresores),
                  filename= paste("ek_AddLocalLevel_",ek,".pdf",sep=""),
                  main = paste("AddLocalLevel",ek))

CompareBstsModels(list("Sin regresores" = modelo_AddAr_sin_regresores,
                       "Con regresores" = modelo_AddAr_con_regresores),
                  filename= paste("ek_AddAr_",ek,".pdf",sep=""),
                  main = paste("AddAr",ek))

CompareBstsModels(list("Sin regresores" = modelo_AddLocalLinearTrend_sin_regresores,
                       "Con regresores" = modelo_AddLocalLinearTrend_con_regresores),
                  filename= paste("ek_AddLocalLinearTrend_",ek,".pdf",sep=""),
                  main = paste("AddLocalLinearTrend",ek))

CompareBstsModels(list("AddLocalLevel" = modelo_AddLocalLevel_sin_regresores,
                       "AddAr" = modelo_AddAr_sin_regresores,
                       "AddLocalLinearTrend" = modelo_AddLocalLinearTrend_sin_regresores),
                  filename= paste("ek_sin_regresores_",ek,".pdf",sep=""),
                  main = paste("Sin regresores",ek))

CompareBstsModels(list("AddLocalLevel" = modelo_AddLocalLevel_con_regresores,
                       "AddAr" = modelo_AddAr_con_regresores,
                       "AddLocalLinearTrend" = modelo_AddLocalLinearTrend_con_regresores),
                  filename= paste("ek_con_regresores_",ek,".pdf",sep=""),
                  main = paste("Con regresores",ek))
```

Figura 5.1: Ejemplo de código R donde se aplica por parquímetro los diferentes componentes y comparativas

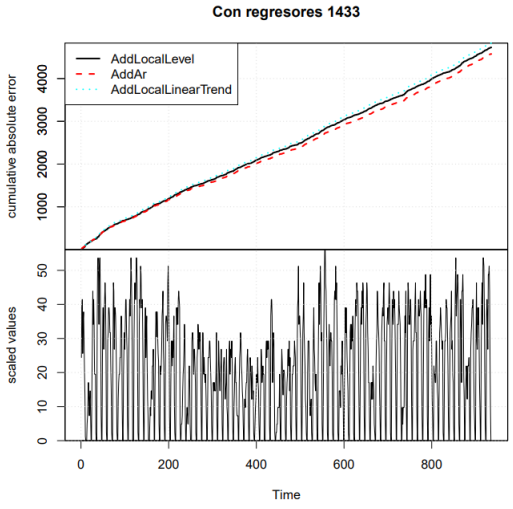

Figura 5.2: Ejemplo de comparativa de tres modelos con componentes distintos para un parquímetro