

Memoria: Transferencia de turismos

Estudio y modelaje de series temporales

Introducción

Como muchos otros servicios públicos, la Dirección General de Tráfico (DGT) pone a disposición de la ciudadanía información. En concreto, para este trabajo, es usar los microdatos que proporciona sobre las transacciones de vehículos.

La idea es, previamente, realizar un estudio sobre la información que se aporta la DGT para, con ella, realizar una previsión de transferencias de vehículos mediante un modelo de Machine Learning y generar una serie temporal con esos datos.

```
### Lines -> 1008504
### Merge with main datagramme
### Total lines -> 6376461
Processing export anual_trf_2021.csv.tar.gz
### Lines -> 1124603
### Merge with main datagramme
### Total lines -> 7501064
```

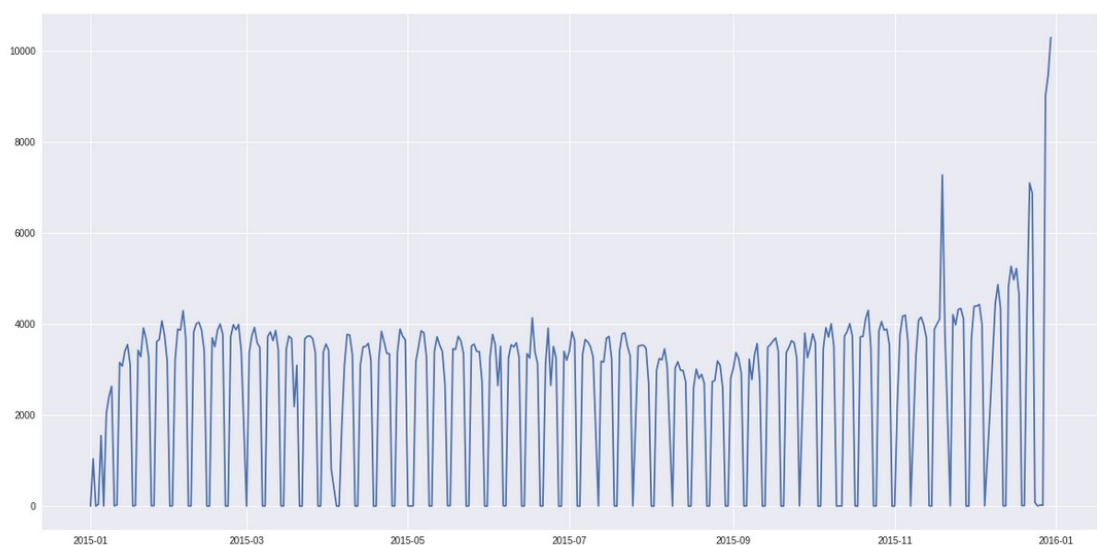
```
In [13]: df.sample(10)
```

```
Out[13]:
```

	FEC_MATRICULA	COD_CLASE_MAT	FEC_TRAMITACION	MARCA_ITV	MODELO_ITV	COD_PROCEDENCIA_ITV	COD_PROPULSION_ITV	CILINDRADA
691393	2017-04-28	0.0	2020-05-12	VOLKSWAGEN	TIGUAN	3.0	1	
2138528	2015-07-02	0.0	2016-01-20	VOLKSWAGEN	POLO	3.0	0.0	
1584299	2019-06-26	0.0	2019-07-17	OPEL	CORSA-E	3.0	6.0	
2554233	2002-06-26	0.0	2017-12-12	AUDI	A4	3.0	1	
957141	2008-04-29	0.0	2016-05-24	SEAT	LEON	0.0	1.0	
1346380	2001-08-09	0.0	2015-08-14	VOLKSWAGEN	GOLF	3.0	1.0	
2400588	2001-04-24	0.0	2020-12-18	VOLKSWAGEN	POLO 1.4 5V	0.0	0	
1284549	2017-03-27	0.0	2018-06-14	HYUNDAI	I20, I20 ACTIVE	1.0	0	
1480976	1967-04-18	0.0	2017-07-06	MERCEDES-BENZ	250 SE	1.0	0.0	
439455	1997-10-02	0.0	2016-02-01	VOLKSWAGEN	POLO 1.4 5V	0.0	0.0	

```
plt.plot(
    grouped[(grouped['FEC_TRAMITE'] >= '2015-01-01') & (grouped['FEC_TRAMITE'] < '2015-12-31')]['FEC_TRAMITE'],
    grouped[(grouped['FEC_TRAMITE'] >= '2015-01-01') & (grouped['FEC_TRAMITE'] < '2015-12-31')]['counts']
)
```

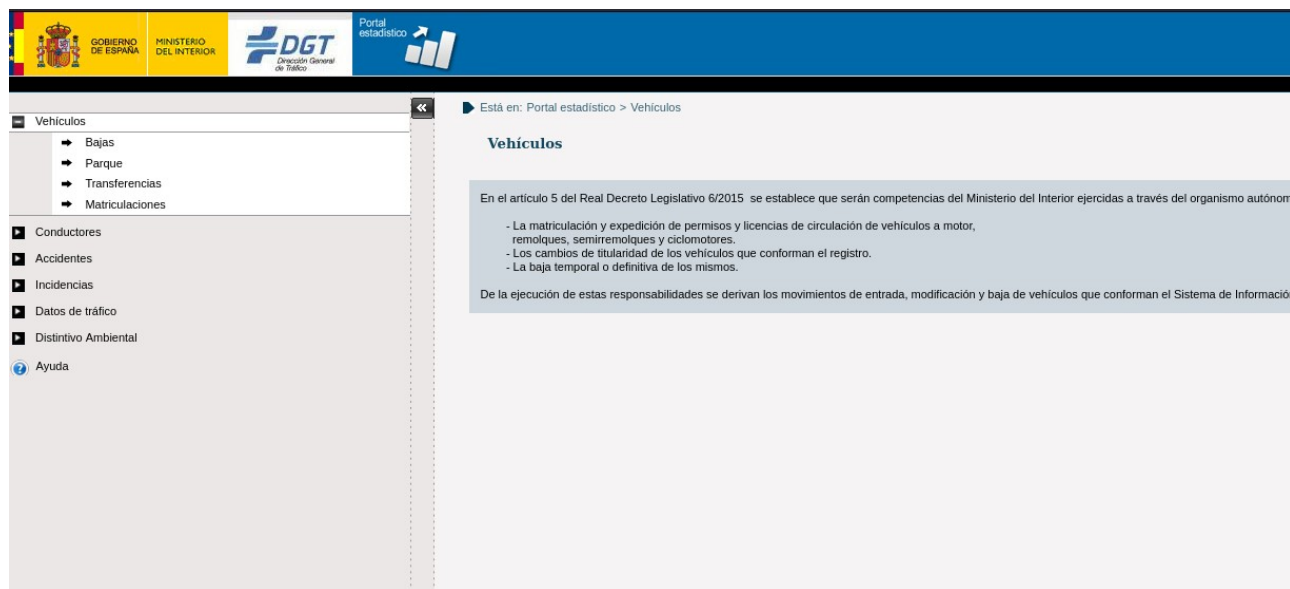
```
Out[18]: [<matplotlib.lines.Line2D at 0x7f8d3089b790>]
```



Origen de los datos

Los datos se obtienen del portal estadístico de la DGT

(https://sedeapl.dgt.gob.es/WEB_IEST_CONSULTA/). En la web se pueden encontrar información de transferencias desde diciembre del 2014. Por tener datos anuales completos, el estudio y la modelización se va a realizar de los datos obtenidos desde 2015 a 2021.



Los datos se deben descargar mes a mes, y se son almacenados en archivos de tipo “ancho fijo” y codificados en ISO-8859-1

1	38062006021032022CITROEN		C4 HDI	3VF7LC9HXC74501456	401	1560	11.47	1332	1777	5	0201		SASA20103202237008	ND
	B0037274SALAMANCA	66.20	5	N								1407	1777M1	
	LC900C													0 0
	01000													
	0			01032022										
2	230220220208032022MERCEDES-BENZ		A 250	3WDD1770471J077970	400	1991	13.28	0	2025	5	0201		ALMU2010320220473811042019UD	
	B0004102VICAR	165.00	5	170N										
	AG			1505	2025M1	AC	0EURO	6AG	01000					
	000000													
	272915591564H		N0000		01032022									
3	29122008001032022SUZUKI		GSXR 600	3J51GN7DA472105466	500	599	6.46	178	380	2	0201		M H 20103202228981	ND
	B0028106PARLA	92.00	2	N										

	00400											178	0L3e	0EURO III
	0			01032022										0 0
4	06042021091032022AUDI		Q3 SPORTBACK	3MAUZZZF30M1089892	401	1968	13.19	0	2145	5	0201		M H 20103202228721	NXAUD01NQP
	B0028121REDUEÑA	110.00	5	150N										
	AG			1055	2145M1	AB	0EURO	GAP	01000					
	000000													
	267715921597M		50000E13	28 29	01032022									
5	22032016001032022SKODA		FABIA	3TMBET6NJKGZ151127	401	1422	9.67	0	1620	5	0101		SSSS20103202220811	NDSKO0061A
	B0020069DONOSTIA	77.00	5	95N										
	A.S.			1165	1620M1	AB	0EURO	6W	01000					
	000000													
	245514631457M		0000		01032022									
6	08052009001032022VOLKSWAGEN		PASSAT	3WVWZZ3C29P068379	401	1968	13.19	1452	2090	5	0101		ZAZA20103202249600	ND
	B0049021BENAVENTE	103.00	5											

La organización de los datos sería la siguiente (se puede ver más sobre los campos en el pdf dentro de la ruta “data/dgt” del repositorio):

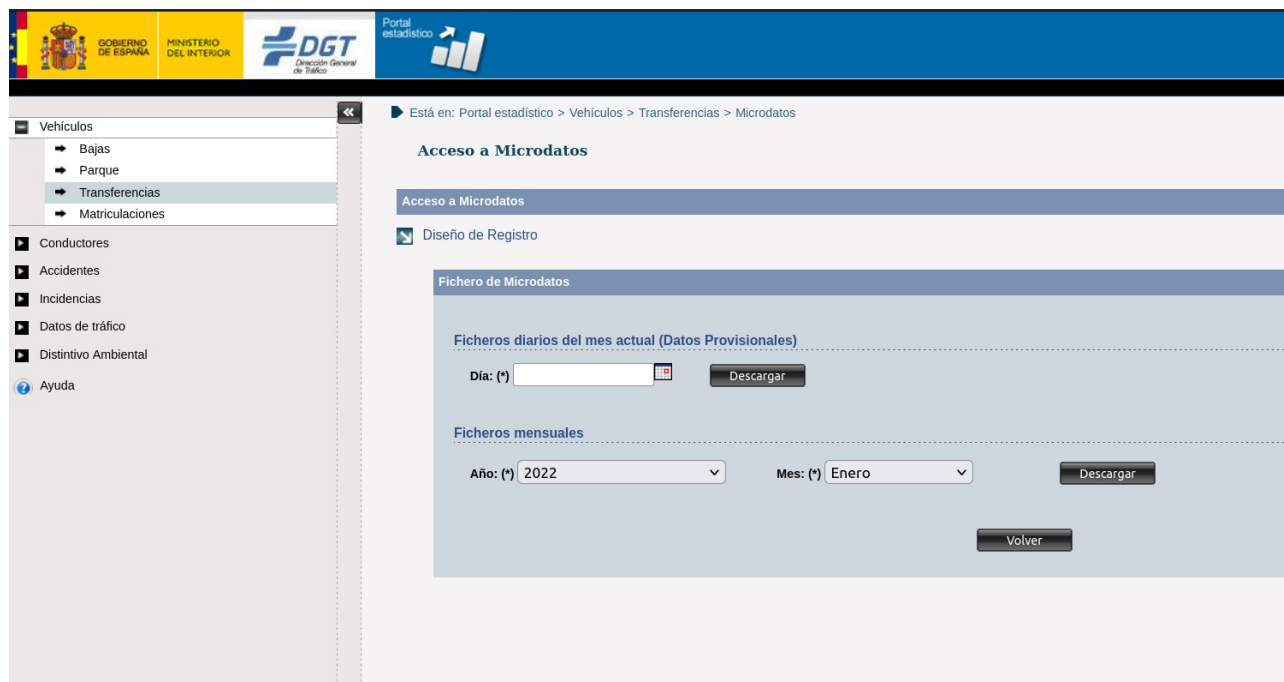
FEC_MATRICULA
COD_CLASE_MAT
FEC_TRAMITACION
MARCA_ITV
MODELO_ITV
COD_PROCEDENCIA_ITV
BASTIDOR_ITV
COD_TIPO
COD_PROPULSION_ITV
CILINDRADA_ITV
POTENCIA_ITV
TARA
PESO_MAX

NUM_PLAZAS
IND_PRECINTO
IND_EMBARGO
NUM_TRANSMISIONES
NUM_TITULARES
LOCALIDAD_VEHICULO
COD_PROVINCIA_VEH
COD_PROVINCIA_MAT
CLAVE_TRAMITE
FEC_TRAMITE
CODIGO_POSTAL
FEC_PRIM_MATRICULACION
IND_NUEVO_USADO
PERSONA_FISICA_JURIDICA
CODIGO_ITV
SERVICIO
COD_MUNICIPIO_INE_VEH
MUNICIPIO
KW_ITV
NUM_PLAZAS_MAX
CO2_ITV
RENTING
COD_TUTELA
COD_POSESION
IND_BAJA_DEF
IND_BAJA_TEMP
IND_SUSTRACCION
BAJA_TELEMATICA
TIPO_ITV
VARIANTE_ITV
VERSION_ITV
FABRICANTE_ITV
MASA_ORDEN_MARCHA_ITV
MASA_MÁXIMA_TECNICA_ADMISIBLE_ITV
CATEGORÍA_HOMOLOGACIÓN_EUROPEA_ITV
CARROCERIA
PLAZAS_PIE
NIVEL_EMISIONES_EURO_ITV
CONSUMO WH/KM_ITV
CLASIFICACIÓN_REGLAMENTO_VEHICULOS_ITV
CATEGORÍA_VEHÍCULO_ELÉCTRICO
AUTONOMÍA_VEHÍCULO_ELÉCTRICO
MARCA_VEHÍCULO_BASE
FABRICANTE_VEHÍCULO_BASE
TIPO_VEHÍCULO_BASE
VARIANTE_VEHÍCULO_BASE
VERSIÓN_VEHÍCULO_BASE
DISTANCIA_EJES_12
VIA_ANTERIOR_ITV
VIA_POSTERIOR_ITV
TIPO_ALIMENTACION_ITV
CONTRASEÑA_HOMOLOGACION_ITV
ECO_INNOVACION_ITV
REDUCCION_ECO_ITV
CODIGO_ECO_ITV
FEC_PROCESO

Obtención de los datos (Web Scraper)

Debido a que la obtención de los datos debe de realizarse mes a mes, se ha creado un web scraper el cual navega hasta la zona de descarga de la página de datos de la DGT y va recorriendo uno a uno los meses y los años, descargando cada uno de los archivos.

El volumen de datos no es muy grande, pero como demostración de conocimiento adquirido, se ha creado el web scraper para automatizar el proceso.



The screenshot displays the 'Portal estadístico' of the DGT (Dirección General de Tráfico). The breadcrumb trail indicates the path: 'Portal estadístico > Vehículos > Transferencias > Microdatos'. The main heading is 'Acceso a Microdatos'. Below this, there's a section for 'Fichero de Microdatos' with two sub-sections: 'Ficheros diarios del mes actual (Datos Provisionales)' and 'Ficheros mensuales'. The 'Ficheros diarios' section has a 'Día: (*)' dropdown and a 'Descargar' button. The 'Ficheros mensuales' section has 'Año: (*)' (set to 2022) and 'Mes: (*)' (set to Enero) dropdowns, a 'Descargar' button, and a 'Volver' button at the bottom.

Preprocesamiento de los datos

El objetivo de este preprocesamiento es, por una parte, y la principal, convertir el tipo de fichero de la DGT (ancho fijo) a un formato más amigable (CSV). Por otra parte, se realiza un cambio de codificación a UTF-8 de los archivos. Estos dos pasos facilitará el trabajo con los archivos.








Además, se recortarán los campos a una selección de ellos, eliminando aquellos que no aporten información sustancial, agilizando así el peso de la información. Los campos seleccionados son los siguientes:

FEC_MATRICULA
COD_CLASE_MAT
FEC_TRAMITACION
MARCA_ITV
MODELO_ITV
COD_PROCEDENCIA_ITV
COD_PROPULSION_ITV
CILINDRADA_ITV
POTENCIA_ITV

NUM_PLAZAS
NUM_TRANSMISIONES
NUM_TITULARES
LOCALIDAD_VEHICULO
COD_PROVINCIA_VEH
COD_PROVINCIA_MAT
CLAVE_TRAMITE
FEC_TRAMITE
CODIGO_POSTAL
FEC_PRIM_MATRICULACION
IND_NUEVO_USADO
PERSONA_FISICA_JURIDICA
COD_MUNICIPIO_INE_VEH
MUNICIPIO
KW_ITV
NUM_PLAZAS_MAX
CO2_ITV
RENTING
CATEGORÍA_HOMOLOGACIÓN_EUROPEA_ITV
NIVEL_EMISIONES_EURO_ITV
CONSUMO_WH/KM_ITV
CATEGORÍA_VEHÍCULO_ELÉCTRICO
AUTONOMÍA_VEHÍCULO_ELÉCTRICO

Se recorta la información a solo vehículos de tipo turismo, ya que el estudio se pensó en un primer momento para ese tipo de vehículos.

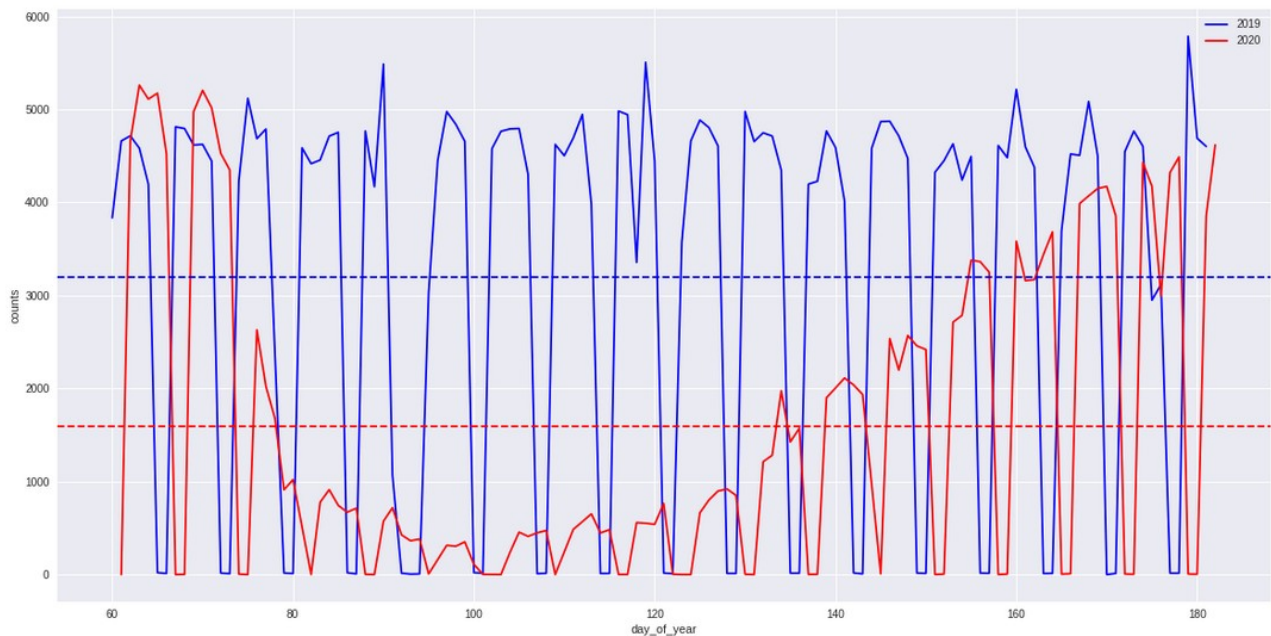
También para agilizar, ya que la DGT comprime los archivos en zip, el preprocesamiento almacenará los archivos comprimidos en formato “tar.gz”, siendo este un formato amigable para abrirse directamente desde los DataFrames de Pandas. Se empaquetarán año a año, en lugar de mes a mes.

Nombre	Tamaño
 export_anual_trf_2015.csv.tar.gz	72,9 MB
 export_anual_trf_2016.csv.tar.gz	80,9 MB
 export_anual_trf_2017.csv.tar.gz	89,4 MB
 export_anual_trf_2018.csv.tar.gz	96,0 MB
 export_anual_trf_2019.csv.tar.gz	98,4 MB
 export_anual_trf_2020.csv.tar.gz	84,3 MB
 export_anual_trf_2021.csv.tar.gz	96,3 MB

Estudio de los datos

En este punto, se cargan los datos en un DataFrame y se procederá al estudio de cada uno de los campos, los valores que pueden tomar y la relación que se establece entre los mismos.

Como base de entre todos los campos del dataset, y encarrilando el trabajo a la serie temporal, el campo que se analiza primero es el que corresponde a la fecha de tramitación. Este campo indica la fecha en la que se realiza la transferencia del vehículo.



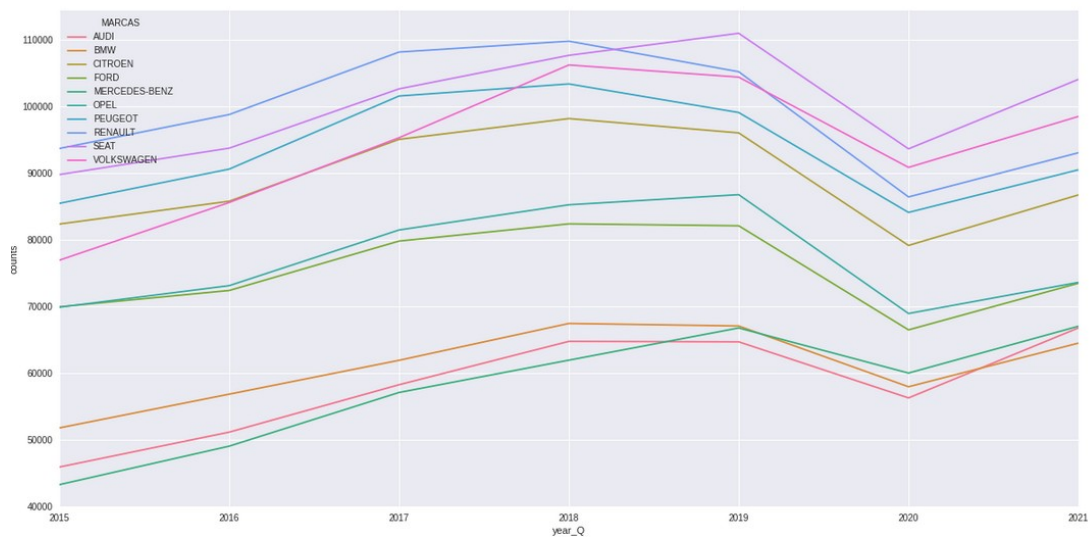
La fecha de tramite será el campo clave para comparar con el resto de columnas del dataset.

El estudio del campo muestra una clara disposición a trabajar con series temporales, ya que se puede apreciar una clara frecuencia en su comportamiento. Por una parte, semanalmente, viendo una bajada considerable cada fin de semana (que será clave al implementar la serie)

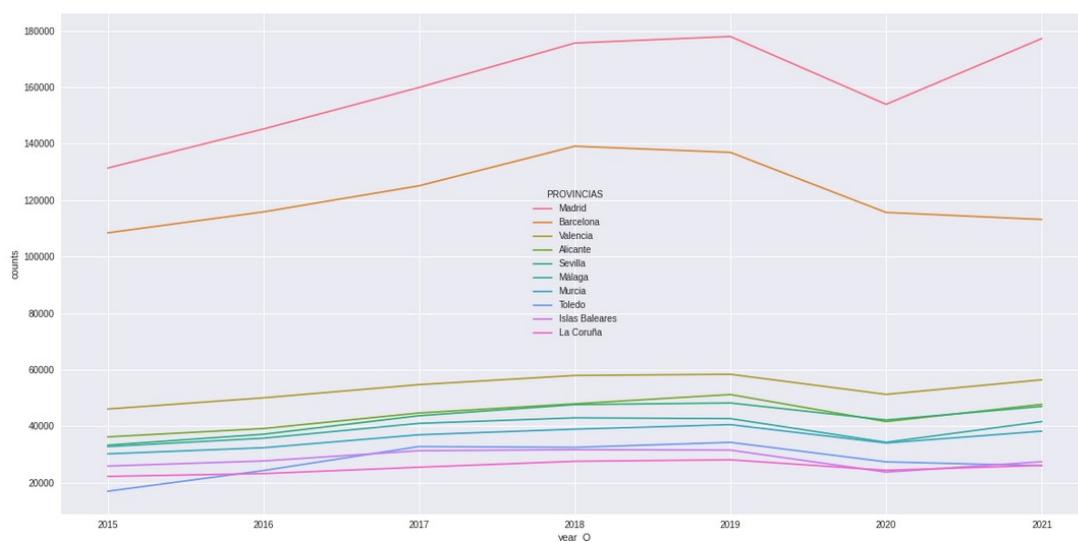


	day_of_week	counts
0	Friday	1401854
1	Monday	1458416
2	Saturday	3045
3	Sunday	1924
4	Thursday	1537891
5	Tuesday	1540341
6	Wednesday	1557593

El siguiente campo en importancia es la marca del vehículo. Se obtiene un top de las marcas que más aparecen y se estudia la evolución año a año de las transacciones.



Junto a la marca, otro de los campos determinantes es la provincia de procedencia del vehículo. Se aprecia un volumen mayor en ciudades como Madrid y Barcelona, separándose del resto de ciudades por un volumen de transacciones considerable.

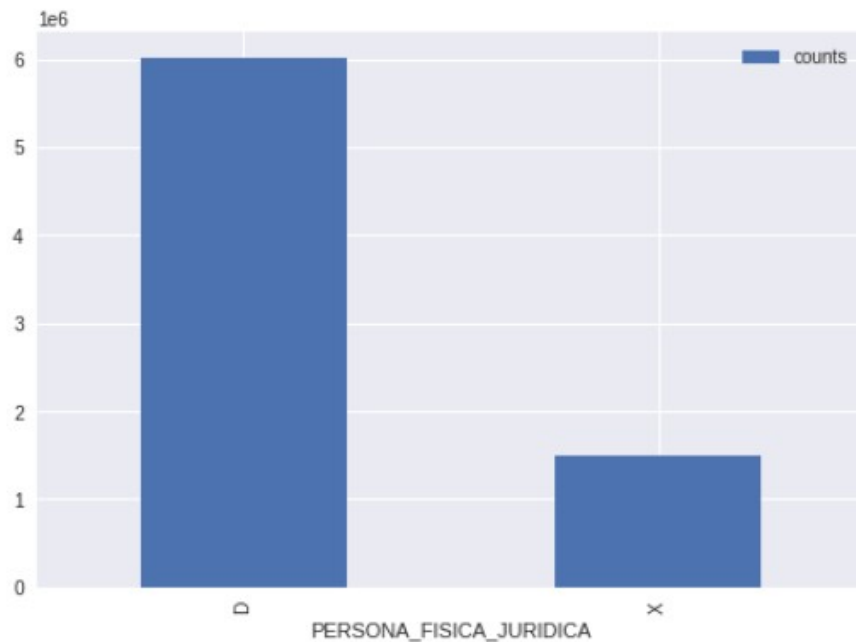


Luego, la importancia de los campos decrece en importancia considerablemente, siendo menos relevantes en cuanto a predictores.

Numerando uno a uno, serían:

- Diferenciación entre persona física y jurídica:

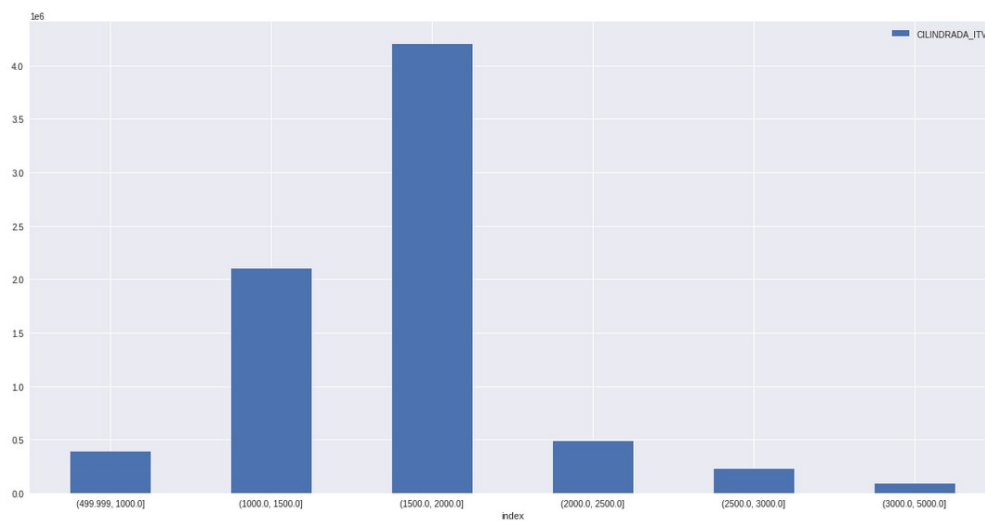
El propietario del vehículo luego de la transacción es un ente jurídico (por ejemplo, siendo lo más probable, una empresa) o una persona.

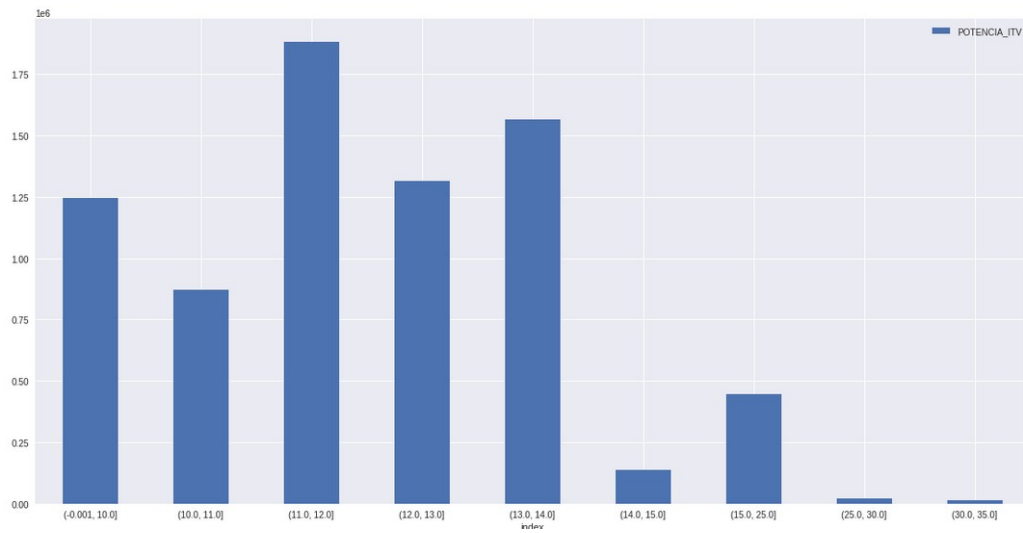


Tomando como categoría este campo, se puede comprobar que la predilección por una marca u otra de vehículo cambia.

- Cilindrada y potencia:

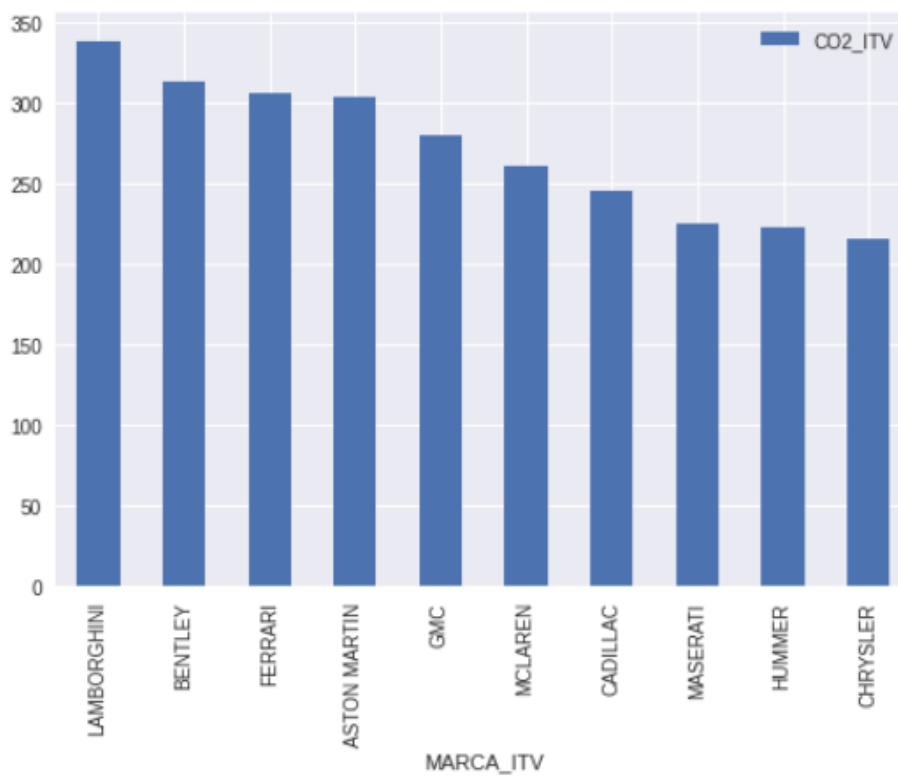
Son campos los cuales son determinantes en el modelo de vehículo, siendo características principales de los mismos.





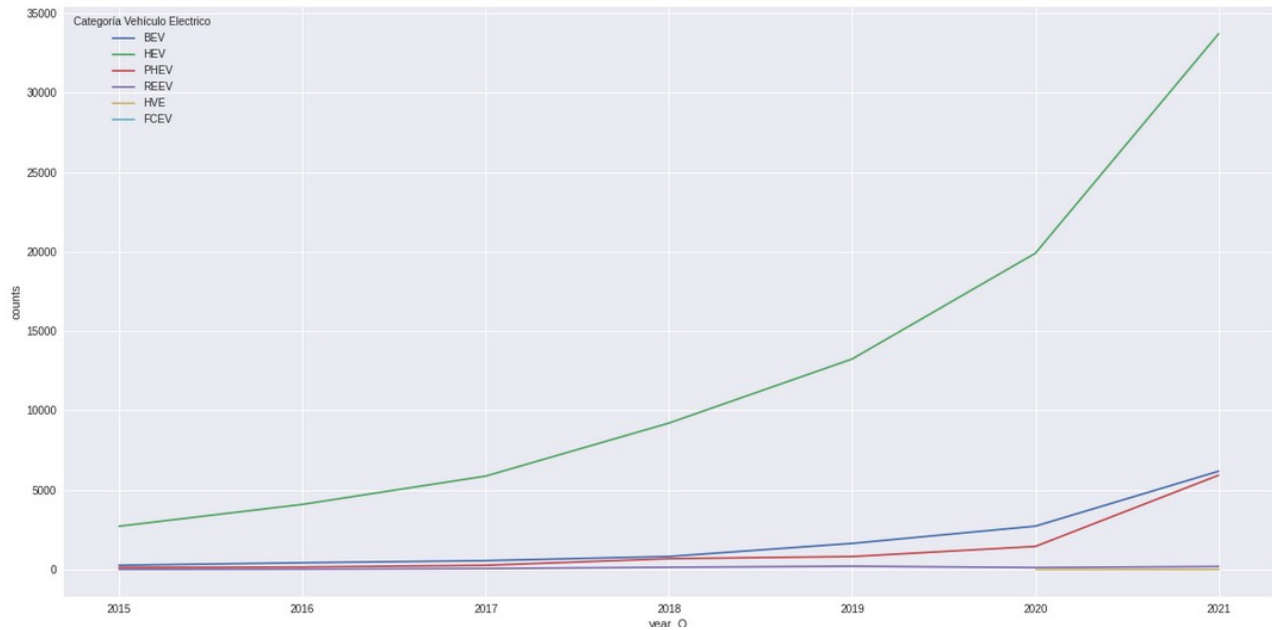
- Emisiones CO2:

Las emisiones contaminantes del vehículo. Tiene una relación directa con las marcas de turismos, quedando claro que los vehículos de marcas americanas o de lujo tienen emisiones más altas.



- Categoría de vehículo eléctrico:

En este campo se puede apreciar una tendencia a las transacciones de vehículos eléctricos híbridos.



Implementación de la serie temporal

El objetivo de este punto es crear una serie temporal en la que se calcule el número de traspasos de turismos. Primero se realiza la carga del dataset en un dataframe con una serie de funciones generadas en el estudio de los datos, el punto previo.

Luego de la carga del dataframe, se realiza una agrupación de los datos por la fecha de tramitación, y se realiza un conteo de los vehículos transaccionados cada día.

	FEC_TRAMITE	count
0	2015-01-01	1
1	2015-01-02	513
2	2015-01-03	3
3	2015-01-04	19
4	2015-01-05	779
...
2293	2021-12-27	3694
2294	2021-12-28	3139
2295	2021-12-29	3701
2296	2021-12-30	3714
2297	2021-12-31	11

Se comprueba los días que no tienen datos (son discontinuidades en la serie), por lo que se tienen que rellenar con información. Se identifican los días que son y se añaden al dataset, indicando previamente un valor inicial de 0 tramitaciones. Luego, comprobando se es un día perteneciente al fin de semana o no, se le asigna un valor u otro (las medias de ese tipo de días), ajustandose a la temporalidad.

	FEC_TRAMITE	date	count
0	2015-01-06	2015-01-06	0
1	2015-02-07	2015-02-07	0
2	2015-02-28	2015-02-28	0
3	2015-03-07	2015-03-07	0
4	2015-03-28	2015-03-28	0
...
254	2020-05-02	2020-05-02	0
255	2020-05-10	2020-05-10	0
256	2020-06-06	2020-06-06	0
257	2020-06-21	2020-06-21	0
258	2020-07-25	2020-07-25	0

Otro paso previo a la modelización, es indicar la temporalidad a la serie. Se añaden unas ondas para representar el ciclo semanal, mensual y anual del paso del tiempo

```
x = np.arange(len(result_dgt))

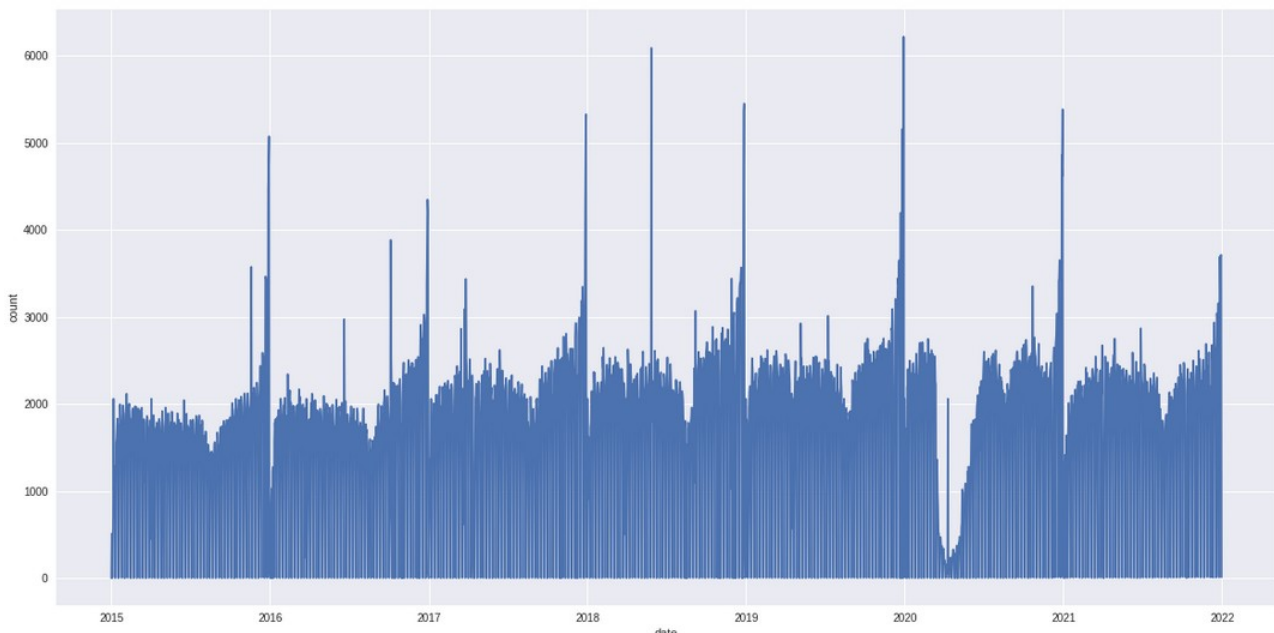
result_dgt["s_period_week"] = np.sin(2*np.pi*x/7)
result_dgt["s_period_month"] = np.sin(2*np.pi*x/30.5)
result_dgt["s_period_year"] = np.sin(2*np.pi*x/365)

result_dgt["c_period_week"] = np.cos(2*np.pi*x/7)
result_dgt["c_period_month"] = np.cos(2*np.pi*x/30.5)
result_dgt["c_period_year"] = np.cos(2*np.pi*x/365)

result_dgt.sort_values('date', ascending=True)
```

	count	date	is_weekend	s_period_week	s_period_month	s_period_year	c_period_week	c_period_month	c_period_year
FEC_TRAMITE									
2015-01-01	1	2015-01-01	0.0	0.000000e+00	0.000000	0.000000	1.000000	1.000000	1.000000
2015-01-02	513	2015-01-02	0.0	7.818315e-01	0.204552	0.017213	0.623490	0.978856	0.999852
2015-01-03	3	2015-01-03	1.0	9.749279e-01	0.400454	0.034422	-0.222521	0.916317	0.999407
2015-01-04	19	2015-01-04	1.0	4.338837e-01	0.579421	0.051620	-0.900969	0.815028	0.998667
2015-01-05	779	2015-01-05	0.0	-4.338837e-01	0.733885	0.068802	-0.900969	0.679273	0.997630
...
2021-12-27	3694	2021-12-27	0.0	-4.338837e-01	0.905702	0.979614	-0.900969	0.423914	-0.200891
2021-12-28	3139	2021-12-28	0.0	-9.749279e-01	0.973264	0.976011	-0.222521	0.229688	-0.217723
2021-12-29	3701	2021-12-29	0.0	-7.818315e-01	0.999668	0.972118	0.623490	0.025748	-0.234491
2021-12-30	3714	2021-12-30	0.0	-2.508671e-13	0.983798	0.967938	1.000000	-0.179281	-0.251190
2021-12-31	11	2021-12-31	0.0	7.818315e-01	0.926324	0.963471	0.623490	-0.376728	-0.267814

Con ello tenemos la secuencialidad del tiempo completa:



Separamos los predictores del resultado a modelizar, así como los datos de entrenamiento de los de test.

```
train_X = train_X[train_X['date'] < '2021-09-01']
test_X = test_X[test_X['date'] >= '2021-09-01']

train_y = train_y[train_y['date'] < '2021-09-01']
test_y = test_y[test_y['date'] >= '2021-09-01']
```

Y, luego de ello, eliminamos la columna de fecha

```
X = X[['is_weekend', 's_period_week', 's_period_week', 's_period_year', 'c_period_week', 'c_period_week', 'c_period_y',
y = y['count']

train_X = train_X[['is_weekend', 's_period_week', 's_period_week', 's_period_year', 'c_period_week', 'c_period_week', 'c_period_y',
test_X = test_X[['is_weekend', 's_period_week', 's_period_week', 's_period_year', 'c_period_week', 'c_period_week', 'c_period_y',

train_y = train_y['count']
test_y = test_y['count']
```

Es el momento de empezar a modelizar. Para ello se usará un modelo de tipo RandomForestReggresion. Luego de realizar una serie de pruebas, se encuentran los metadatos más óptimos y se realiza una comparación de la predicción con los datos reales.

Conclusiones

El modelo consigue predecir el descenso de transacciones los fines de semana. Dependiendo de las iteraciones, consigue o no el aumento al final de año (esto puede ser debido al fraccionamiento de los datos, ya que a cada generación se trae un porcentaje bajo de los mismo, pero a un volumen mayor el kernel se satura)