Coches de segunda mano

Dossier técnico

Descripción

Anuncios de venta de coches en las **principales plataformas.**

Las características de este dataset son las siguientes:

- **Frecuencia de actualización**: diariamente
- **Volumen estimado**: 300.000 registros cada día
- **Histórico**: disponible desde noviembre de 2020

El dataset completo se puede adquirir en DataMarket, plataforma de referencia de datos externos en España.

Muestra

La muestra contiene unas **50.000 filas** las cuales se han seleccionado del día 15/01/2021. Se puede acceder a dicha muestra en siguiente enlace:

- https://datamarket.es/media/samples/coches-de-segunda-mano-sample.csv

Columnas

A continuación se muestran las columnas de las que consta el dataset:

nombre	***tipo***	***descripción***	***ejemplo***
color	*str*	*Color del vehículo.*	*Gris / Plata (GRIS)*
company	*str*	*Identificador del sitio web donde se ha listado el vehículo en venta.*	*9881BCDD5A0AD4733037B3FB25E69C3A*
country	*str*	*País donde se vende el vehículo.*	*España*
dealer	*str*	*Vendedor del vehículo. En el caso de vendedores particulares (no concesionarios), esta información está encriptada en el dataset para cumplir con la GDPR.*	*Autoplanet*
doors	*int*	*Número de puertas del vehículo.*	*5*
fuel	*str*	*Tipo de combustible del vehículo (diésel, gasolina, eléctrico, híbrido).*	*Híbrido*
insert_date	*datetime*	*Fecha de extracción de la información.*	*2020-11-24 00:00:00*
is_professional	*bool*	*Indica si el vendedor es profesional (un concesionario).*	*True*
kms	*int*	*Kilometraje del vehículo.*	*78742*
make	*str*	*Marca del coche.*	*LEXUS*

model	*str*	*Modelo del vehículo.*	*NX*
photos	*int*	*Número de fotografías del vehículo disponibles en el anuncio.*	*32*
power	*int*	*Potencia del vehículo.*	*197*
price	*int*	*Precio de venta del vehículo.*	*27900*
price_financed	*int*	*Precio si el coche está financiado.*	*27900*
province	*str*	*Provincia donde se vende el vehículo.*	*Madrid*
publish_date	*datetime*	*Fecha de publicación del anuncio.*	*2020-10-30 11:24:56*
shift	*str*	*Tipo de cambio (Automático/Manual).*	*Automático*
url	*str*	*Url del coche de segunda mano en venta.*	*2e6ffb51e1ddb2db51d3cf56a4406f6c*
version	*str*	*Versión del vehículo.*	*LEXUS NX 300h F Sport 4WD Navibox 5p.*
year	*int*	*Año de fabricación del vehículo.*	*2015*

Casos de uso

En esta sección se describen **cinco casos de uso reales** para este dataset. Estos casos de uso se estructuran en forma de cinco preguntas que se pueden responder haciendo uso del mismo.

De los 5 casos de uso, 4 tratan de comprender la realidad usando el dataset, mientras que el último es un modelo simple de Machine Learning que modela la realidad para predecir comportamientos futuros.

Los resultados se presentan desde el punto de vista del detalle del código en SQL o Python usado, como también en forma de visualización que resume los resultados obtenidos.

| ¿Cuál es el precio medio de venta por marca?

Descripción

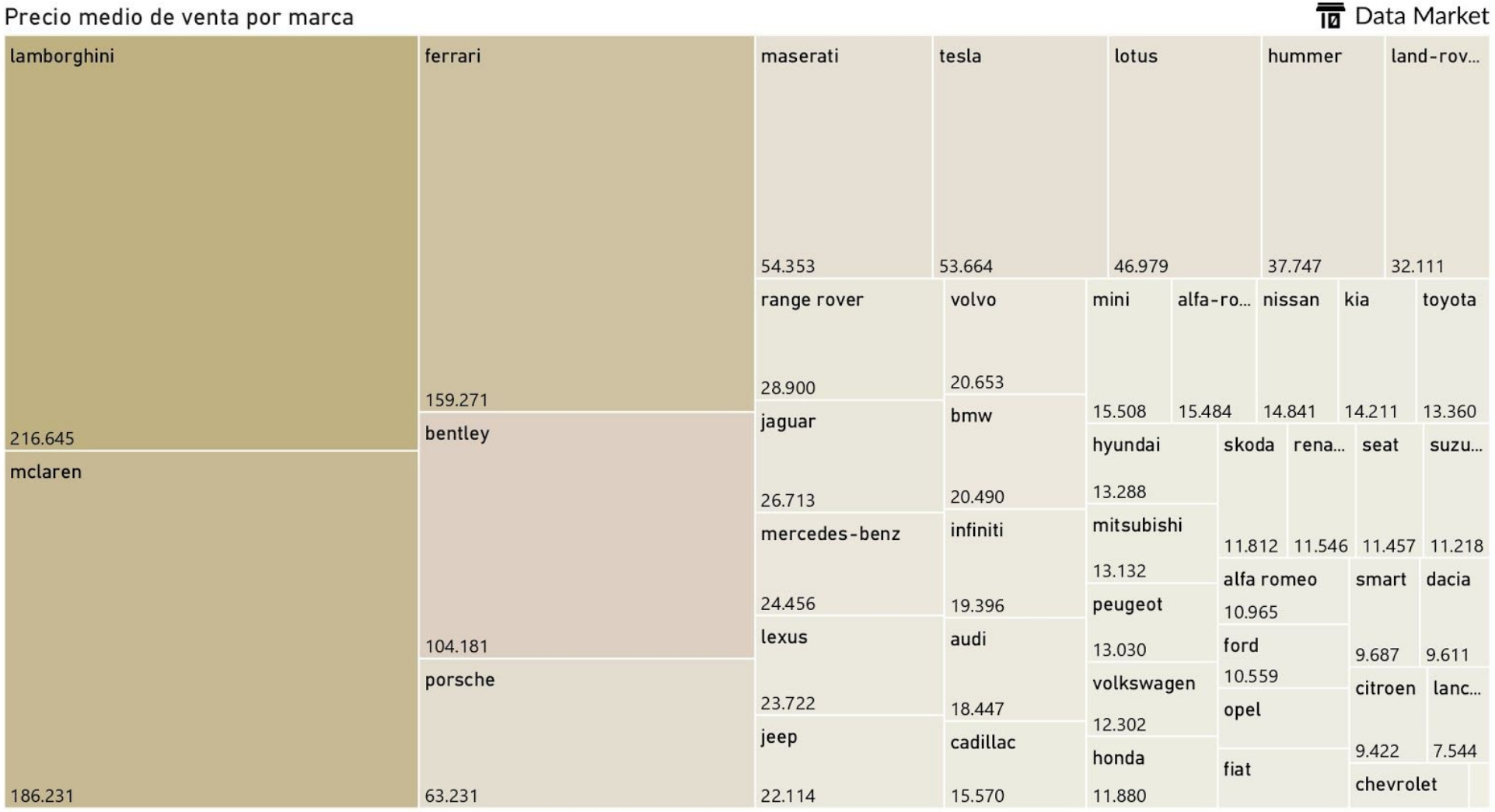
Calcular el precio medio de venta de todas las marcas de coches con la finalidad de conseguir una visión más clara de cuáles de ellas son las más caras y más baratas en el mercado de coches de segunda mano, kilómetro cero y seminuevos.

Solución

Consulta SQL

```sql
with last_appearance_date as (
select
    max(insert_date) as insert_date,
    url
from
    coches_de_segunda_mano
group by
    coches_de_segunda_mano.url ),
car_last_appearance as (
select
    coches_de_segunda_mano.*
from
    coches_de_segunda_mano
join last_appearance_date on
    last_appearance_date.url = coches_de_segunda_mano.url
    and coches_de_segunda_mano.insert_date = last_appearance_date.insert_date )
select
    cars_last_appearance.make,
    round(avg(cars_last_appearance.price), 2) as avg_price
from
    cars_last_appearance
group by
    make
```

Visualización



| Fecha media de fabricación de los vehículos por marca

Descripción

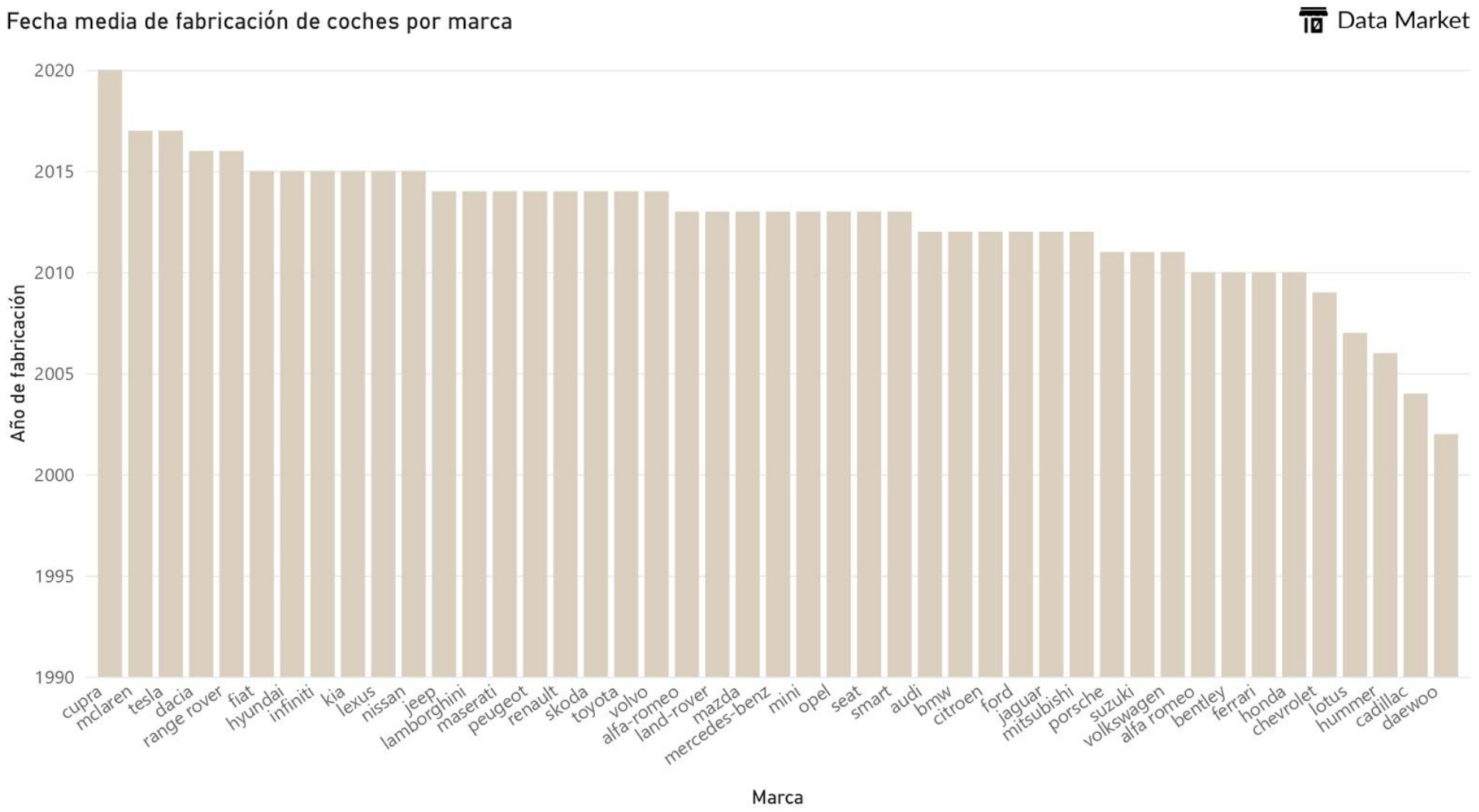
Se quiere ver la tendencia de los usuarios a lo largo del tiempo en cuanto a preferencia del tipo de marca de coches. Por ello, se hará una media de la fecha de fabricación de los vehículos por marca de coches.

Solución

Consulta SQL

```sql
with last_appearance_date as (
select
    max(insert_date) as date_last,
    url
from
    coches_de_segunda_mano
group by
    coches_de_segunda_mano.url ),
cars_last_appearance as(
select
    last_appearance_date.*,
    year,
    make
from
    last_appearance_date
join coches_de_segunda_mano on
    coches_de_segunda_mano.url = last_appearance_date.url
    and coches_de_segunda_mano.insert_date = last_appearance_date.date_last )
select
    make,
    round(avg(year), 0) as avg_year
from
    cars_last_appearance
group by
    make
```

Visualización



| Top 10 de modelos de coches en venta (marca + modelo)

Descripción

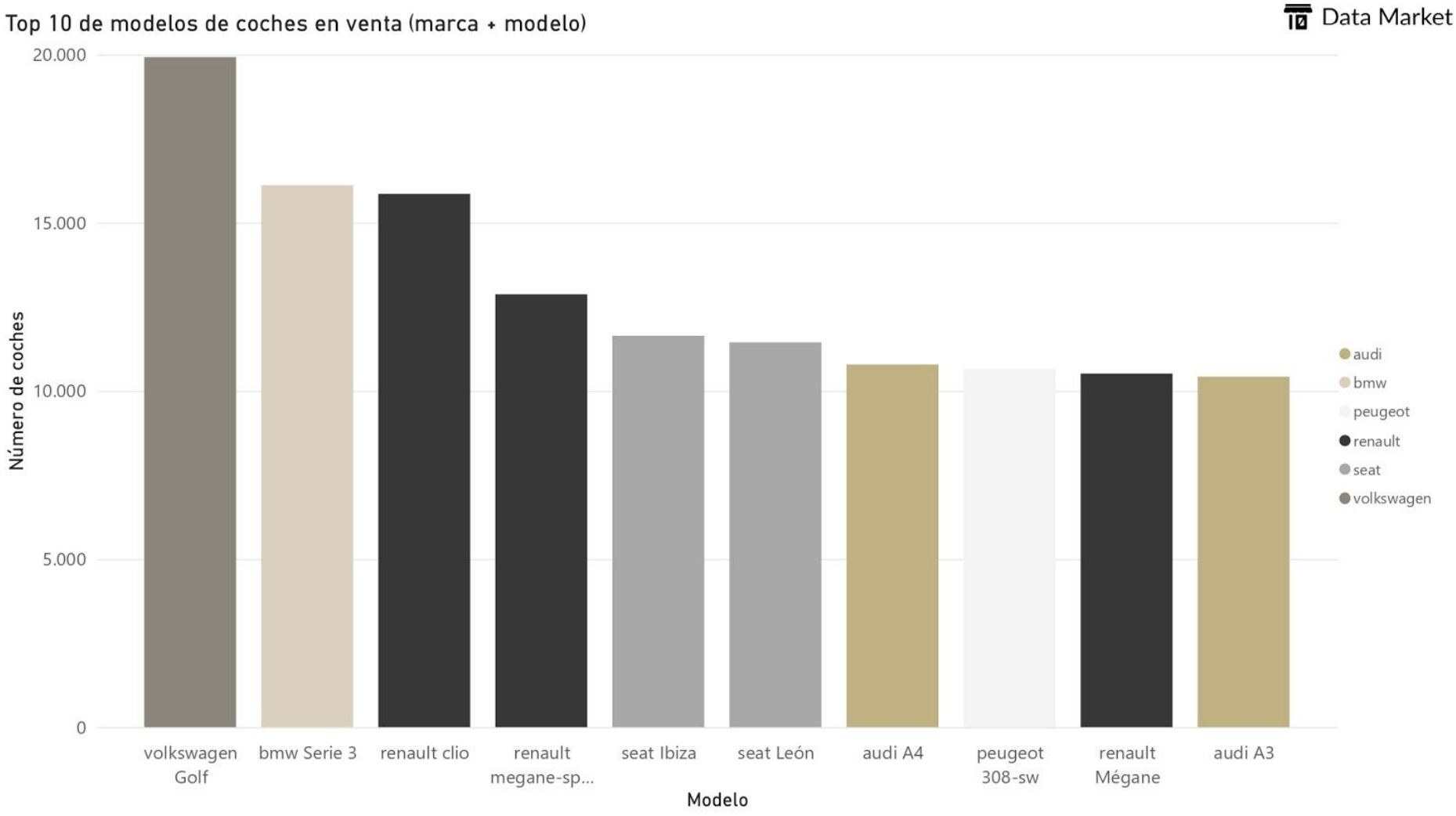
Selección de los modelos con mayor número de coches en venta en las plataformas de coches de segunda mano.

Solución

Consulta SQL

```sql
with last_appearance_date as (
select
    max(insert_date) as date_last,
    url
from
    coches_de_segunda_mano
group by
    url
    ),
cars_last_appearance as(
select
    coches_de_segunda_mano.*
from
    coches_de_segunda_mano
join last_appearance_date on
    last_appearance_date.date_last = coches_de_segunda_mano.insert_date
    and last_appearance_date.url = coches_de_segunda_mano.url
    )
select
    make,
    count(url) as n_cars
from
    cars_last_appearance
group by
    make
order by
    n_cars DESC
limit 10
```

Visualización



| Porcentaje de vehículos eléctricos (frente al total) en venta por provincias

Descripción

Obtener a nivel provincial la cantidad de coches eléctricos de segunda mano o seminuevos en venta, para medir la adopción del coche eléctrico en España.

Solución

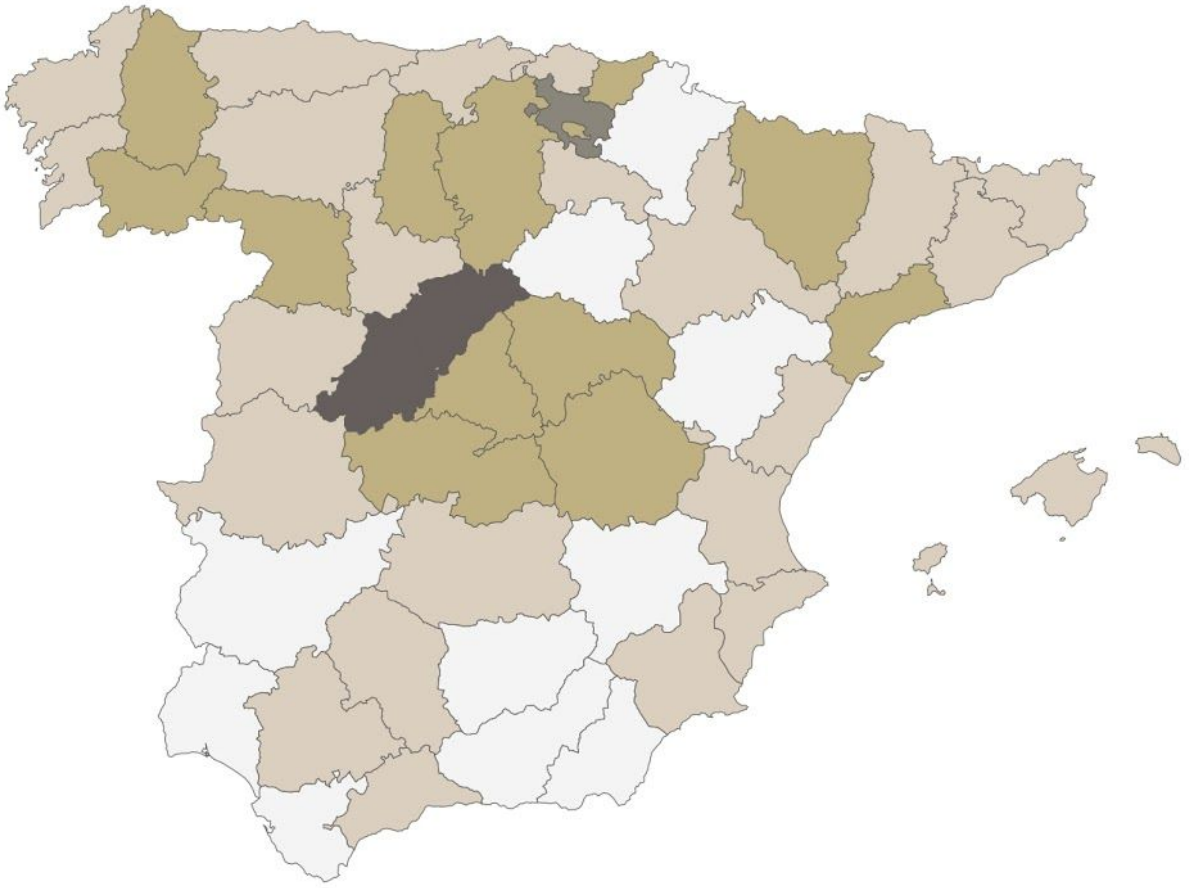
Consulta SQL

```sql
with last_appearance_date as (
select
    max(insert_date) as date_last,
    url
from
    coches_de_segunda_mano
group by
    url
    ),
cars_last_appearance as(
select
    coches_de_segunda_mano.*
from
    coches_de_segunda_mano
join last_appearance_date on
    last_appearance_date.date_last = coches_de_segunda_mano.insert_date
    and last_appearance_date.url = coches_de_segunda_mano.url
    ),
electric_cars as (
select
    province,
    count(url) as n_cars_electric
from
    cars_last_appearance
where
    country = 'Spain'
    and fuel = 'Eléctrico'
group by
    country,
    province
    ),
total_cars as (
select
    province,
    count(url) as n_cars_total
from
    cars_last_appearance
where
    country = 'Spain'
group by
    country,
    province
    )
select
    total_cars.province,
    n_cars_total,
    n_cars_electric,
    cast(electric_cars.n_cars_electric as float) / cast(total_cars.n_cars_total as float) * 100
as perc_electric_cars
from
    electric_cars
join total_cars on
    electric_cars.province = total_cars.province
```

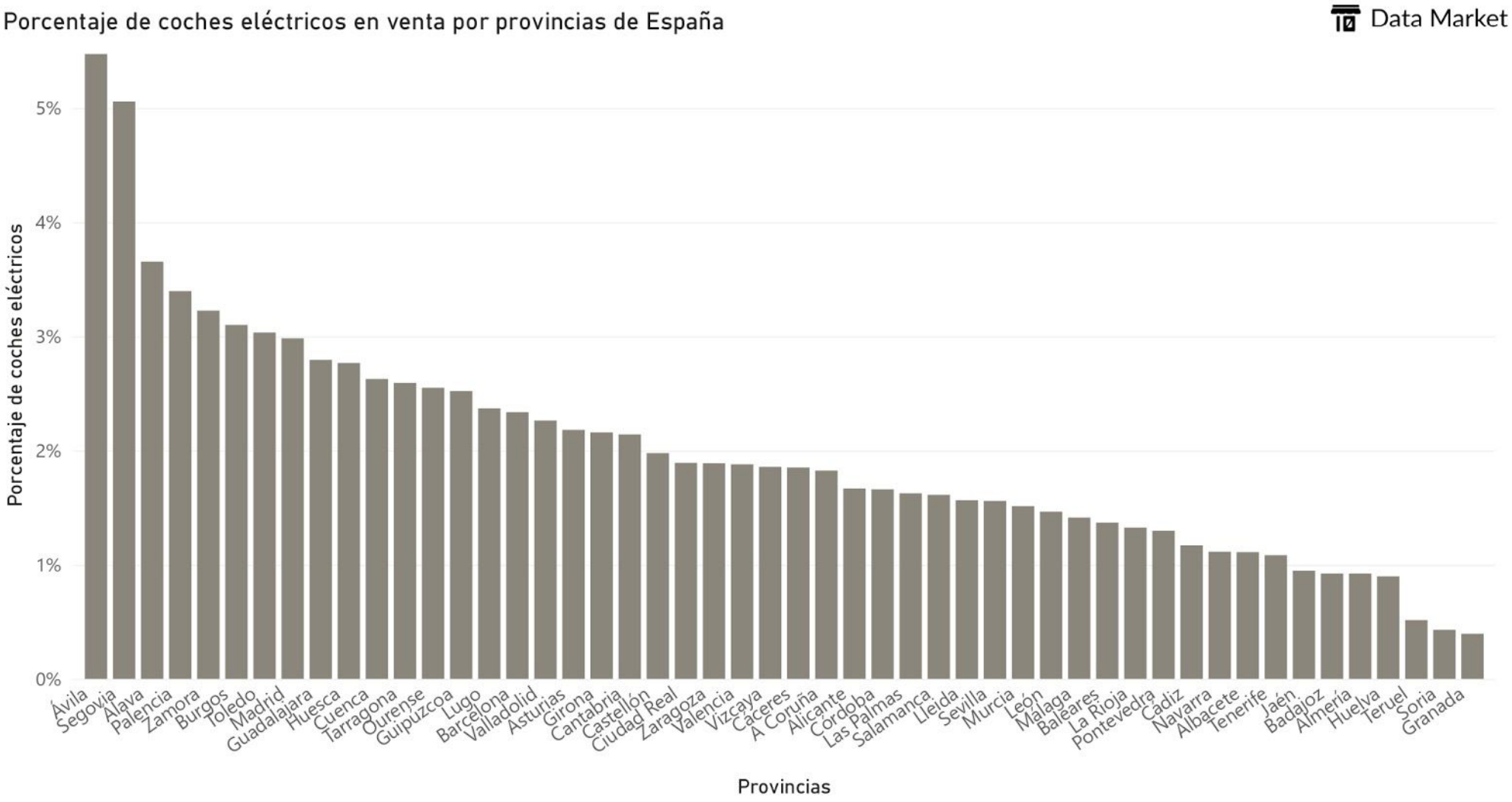
Visualización

Porcentaje de coches eléctricos en venta por provincias en España

Data Market



Porcentaje de coches eléctricos en venta por provincias de España



| Predicción del precio de un coche en función de sus características

Descripción

Determinar el precio de venta de un coche de segunda mano a largo plazo dependiendo de sus características.

Solución

Modelo predictivo simple

```
import pandas as pd
import numpy as np
from sklearn.pipeline import Pipeline
from sklearn.compose import ColumnTransformer
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler, OrdinalEncoder
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor

#Cargamos el dataset y elegimos las columnas que vamos a usar para predecir el precio de un coche.
df_coches = pd.read_csv("coches-de-segunda-mano-sample.csv")
df_coches_def = df_coches[['make','model','fuel', 'year', 'kms', 'power', 'doors', 'shift', 'color',
'is_professional', 'price']]
```

```
#Separamos las columnas en categóricas y númericas ya que se tratan diferentes los datos
NUM_FEATS = ['year', 'kms', 'power']
CAT_FEATS = ['make', 'model', 'fuel', 'shift',
             'color', 'doors', 'is_professional']
FEATS = NUM_FEATS + CAT_FEATS
TARGET = 'price'

#Preprocesamos los datos, rellenamos los nulos, escalamos las columnas numéricas y pasamos las
categóricas a números.
numeric_transformer = \
Pipeline(steps=[('imputer', SimpleImputer(strategy='median')),
                ('scaler', StandardScaler())])
categorical_transformer = \
Pipeline(steps=[('imputer', SimpleImputer(strategy='constant', fill_value="missing")),
                ('ordinal', OrdinalEncoder(handle_unknown='use_encoded_value',
unknown_value=-10000000))])
preprocessor = \
ColumnTransformer(transformers=[('num', numeric_transformer, NUM_FEATS),
                                ('cat', categorical_transformer, CAT_FEATS)])

#Hacemos un pipeline para el modelo donde cargamos el preprocesamiento y el modelo, en este caso un
RandomForest
model= Pipeline(steps=[('preprocessor', preprocessor),
                       ('regressor', RandomForestRegressor())])
```

```python
#Dividimos el dataset aleatoriamente en dos, una parte para entrenar y la otra para validar el
resultado.
coches_train, coches_test = train_test_split(df_coches_def, random_state=5)

#Entrenamos del modelo
model.fit(coches_train[FEATS], coches_train[TARGET]);

#Predecimos el precio
y_test = model.predict(coches_test[FEATS])
y_train = model.predict(coches_train[FEATS])

#Vemos el error con al métrica del error cuadrático medio.
print(f"Error cuadrático medio del test: {mean_squared_error(y_pred=y_test, y_true=coches_test[TARGET],
    squared=False).round(3)}€")
print(f"Error cuadrático medio del train: {mean_squared_error(y_pred=y_train,
y_true=coches_train[TARGET], squared=False).round(3)}€")

#Error cuadrático medio del test: 4491.633€
#Error cuadrático medio del train: 2051.871€
```

Visualización (error cuadrático medio cometido por marca)