



AutoInsight

Predicción y Análisis
Inteligente de Vehículos

Víctor Angulo de Castro
Marcos de Castro Muñoz

¿Quiénes somos?

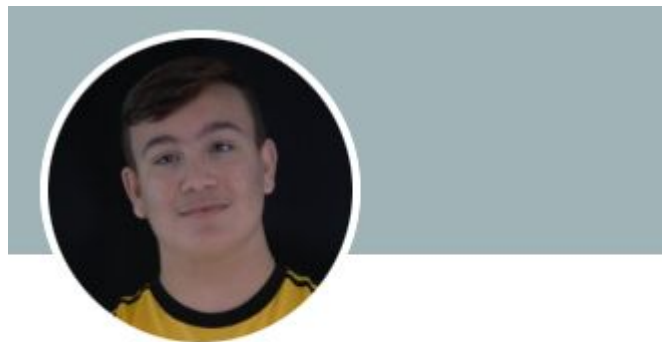


Víctor Angulo De Castro He/Him

[Añadir insignia de verificación](#)

Científico de Datos Junior

Madrid, Comunidad de Madrid, España · [Información](#)



Marcos De Castro Muñoz · 1er

Software Developer

España · [Información de contacto](#)

Scrapping de los datos

- Hemos obtenido un total de 3000 datos con las siguientes columnas:
 - Modelo y marca del coche
 - Precio
 - Kilómetros recorridos
 - Año de compra
 - Descripción del coche
 - Url de la primera imagen del anuncio



Milanuncios.com



TESLA - MODEL 3

Precio al contado	Precio financiado
23.990 €	23.980 €

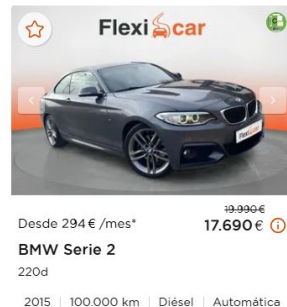
IVA Incluido

📍 Móstoles (Madrid)

164.000 kms • 2019 • Eléctrico

Tesla model 3 long range dual motor 351cv - iva deducible incluido en el precio - vehículo en perfecto estado, dos llave...

Flexicar.com



Desde 294 €/mes*

17.690 €

BMW Serie 2

220d

2015 | 100.000 km | Diésel | Automática

Transformación de los datos

- Nos quedamos con los 30 modelos de coches con más cantidad de resultados de ambos datasets (milanuncios y flexicar).
- Unimos los datasets en uno que contenga todos los datos.
- Eliminamos el símbolo “€” de la columna de precio.
- Eliminamos el texto de “km” de la columna de kilómetros.
- Cambiamos ambas columnas al tipo de dato “int64”.
- Usamos LabelEncoder() de SkLearn para pasar tanto los modelos de coches como los colores a números que puedan pasar por el modelo.
- Usamos getDummies() de Pandas para separar en 3 columnas booleanas la columna del estado del vehículo.

Predicción de colores en base a imágenes

- Nos descargamos un [dataset con imagenes de vehiculos](#) para entrenar una CNN (escaladas a 128x128), los colores/clases a predecir son: beige, negro, azul, marrón, dorado, verde, gris, naranja, rosa, morado, rojo, plateado, bronceado, blanco, amarillo.
- Definimos en la CNN que está compuesta por tres bloques convolucionales con funciones de activación ReLU (cada bloque reduce la resolución espacial mientras aumenta la profundidad de los canales), Después de la extracción de características, la red aplana la salida y pasa por un clasificador totalmente conectado que incluye una capa densa, activación ReLU, dropout para evitar sobreajuste y una capa final que proyecta al número de clases (colores).
- Obteniendo una precisión del 80 % en la clasificación del color del vehículo.

✓ Test Accuracy: 0.8072

Análisis y clasificación de descripciones

- Hemos usado el modelo [“all-MiniLM-L6-v2”](#) que importamos de la librería `sentence_transformers`.
- Para entrenar al modelo usamos principalmente descripciones de automóviles generadas con ChatGPT.
- El modelo devuelve 3 categorías diferentes:

		precision	recall	f1-score	support
○ Descripción buena					
○ Descripción neutra	bueno	0.94	1.00	0.97	48
	malo	1.00	0.98	0.99	53
○ Descripción mala	neutral	0.98	0.95	0.96	55
	accuracy			0.97	156

Modelo de predicciones

- Hemos probado 2 modelos diferentes para la predicción:
 - RandomForestRegressor de SkLearn
 - XGBoostRegressor
- Antes del `train_test_split()` hemos aleatorizado el orden de los datos para que no aparezcan en el orden en el que los sacamos.
- Para ambos modelos se ha hecho un Grid Search para elegir los mejores parámetros.
- Como métrica final hemos elegido Root Mean Squared Error (RMSE).

RMSE modelo
RandomForest

Best RMSE RF: 10638.413355031667

RMSE modelo
XGBoost

Best RMSE XGB: 10031.009090491852