Predicción del precio de venta de vehículos de 2ª mano en función de sus características

Icia Carro Barallobre, Karen Salazar Gutiérrez, Laura Llorente Sanz

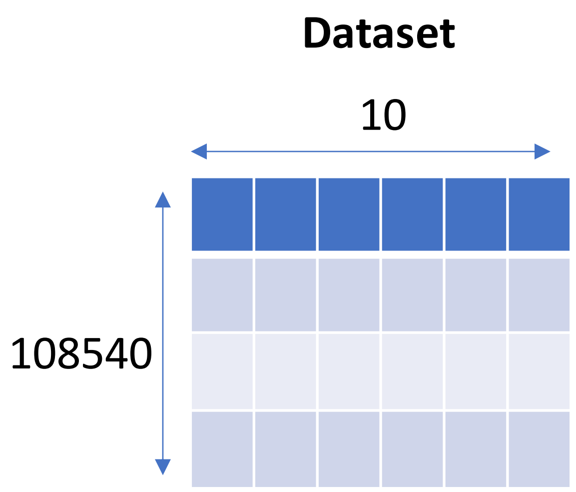
UPM FORMACIÓN

SAMSUNG

Preprocesamiento de los datos (I)

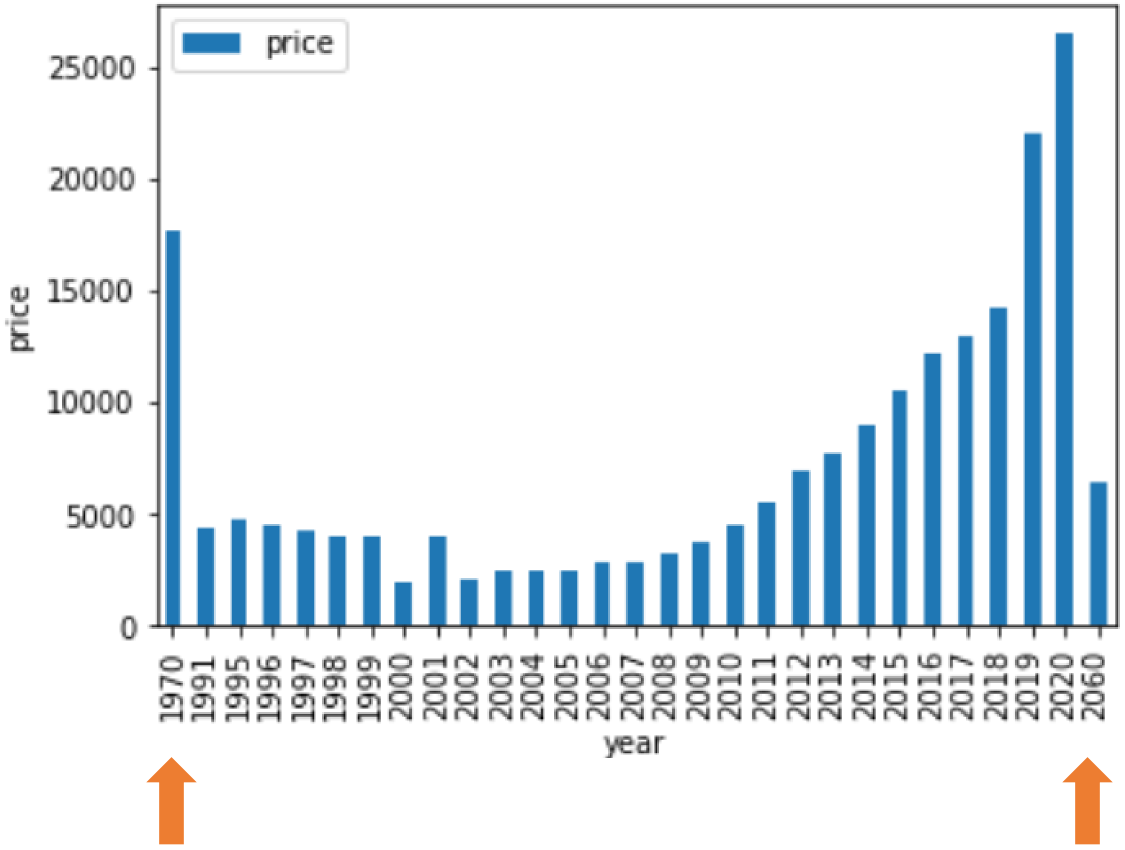
Nombre fichero	Núm coches x Features
Audi.csv	10668 x 9
Bmw.csv	10781 x 9
Cclass.csv	3899 x 7
Merc.csv	13119 x 9
Ford.csv	17965 x 9
Focus.csv	5454 x 7
Hyundi.csv	4860 x 9
skoda.csv	6267 x 9
toyota.csv	6738 x 9
vauxhall.csv	13632 x 9
vw.csv	15157 x 9

Brand	
Audi	
BMW	
Mercedes	17018 x 10
Ford	23419 x 10
Hyundi	
Skoda	
Toyota	
Vauxhall	
VW	



Preprocesamiento de los datos (II)

Valores atípicos

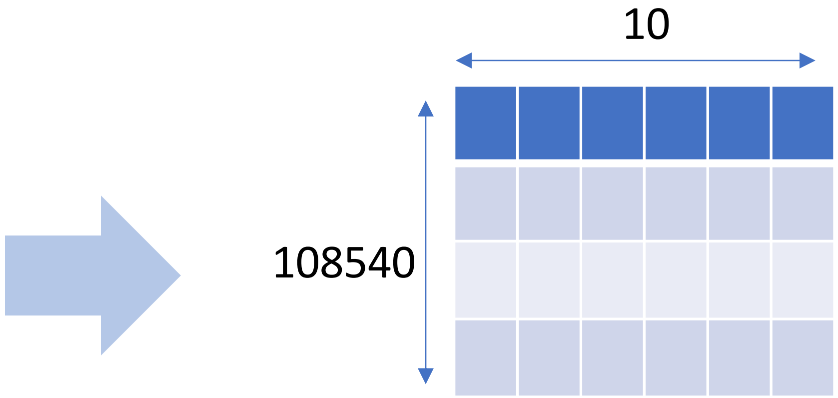


Ingeniería de características I

Variable Derivada:
Antigüedad

Model	Year	Price	Transmission	Mileage	fuelType	Tax	Mpg	engineSize	Brand	Old

Drop



Dataset

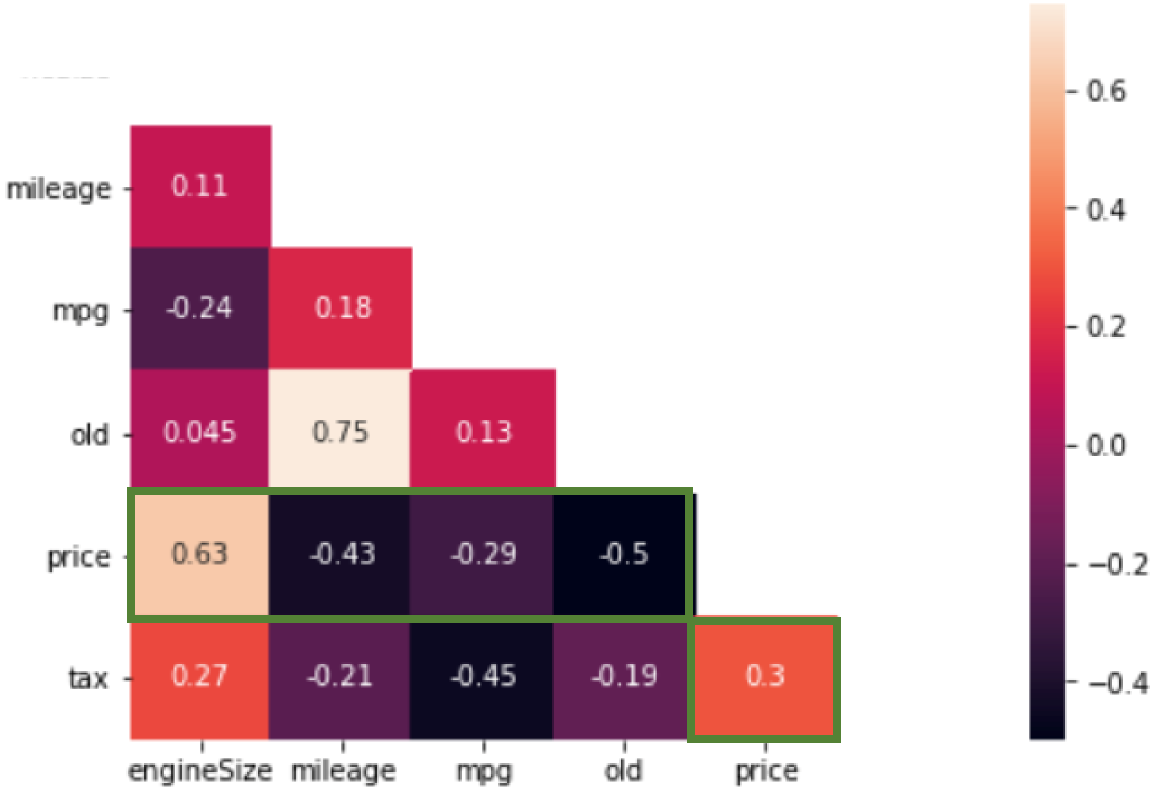
Exploración de la variable objetivo: Price

count	108540
mean	16890
std	9756
min	450
25%	10229
50%	14698
75%	20940
max	159999

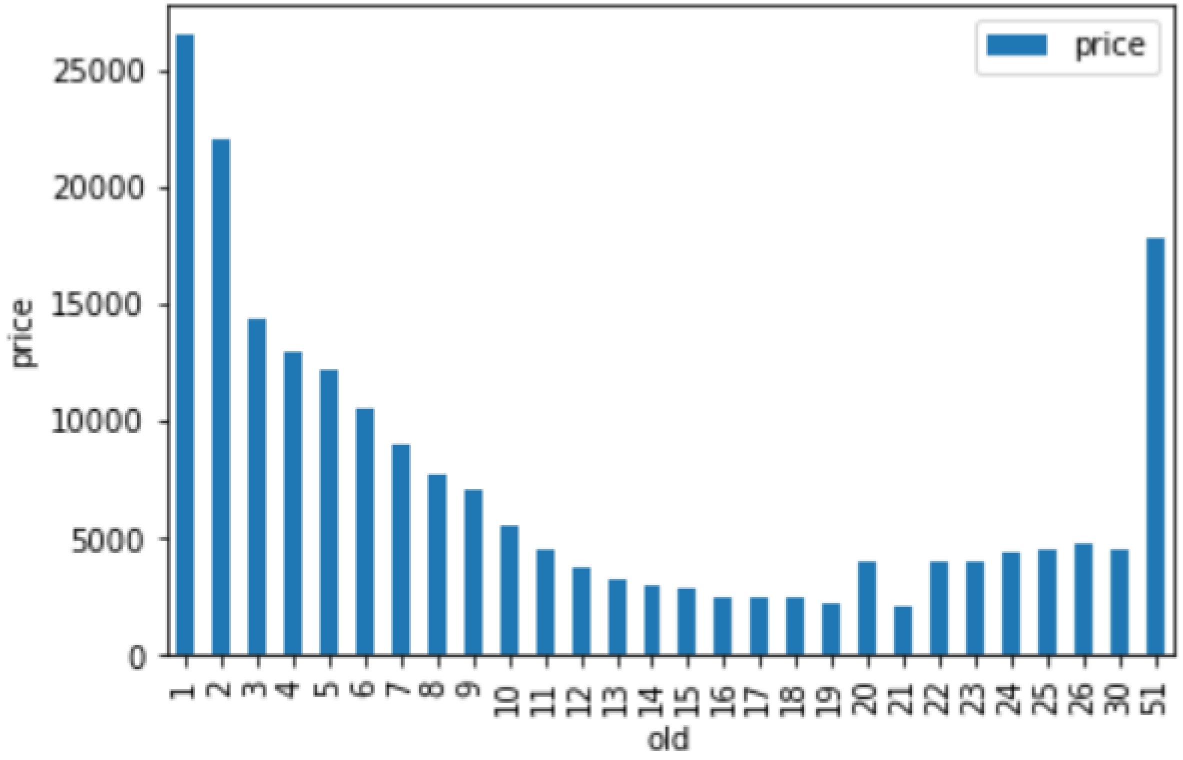
Matriz de correlación

(Variables No categóricas)

Correlaciones bastante altas

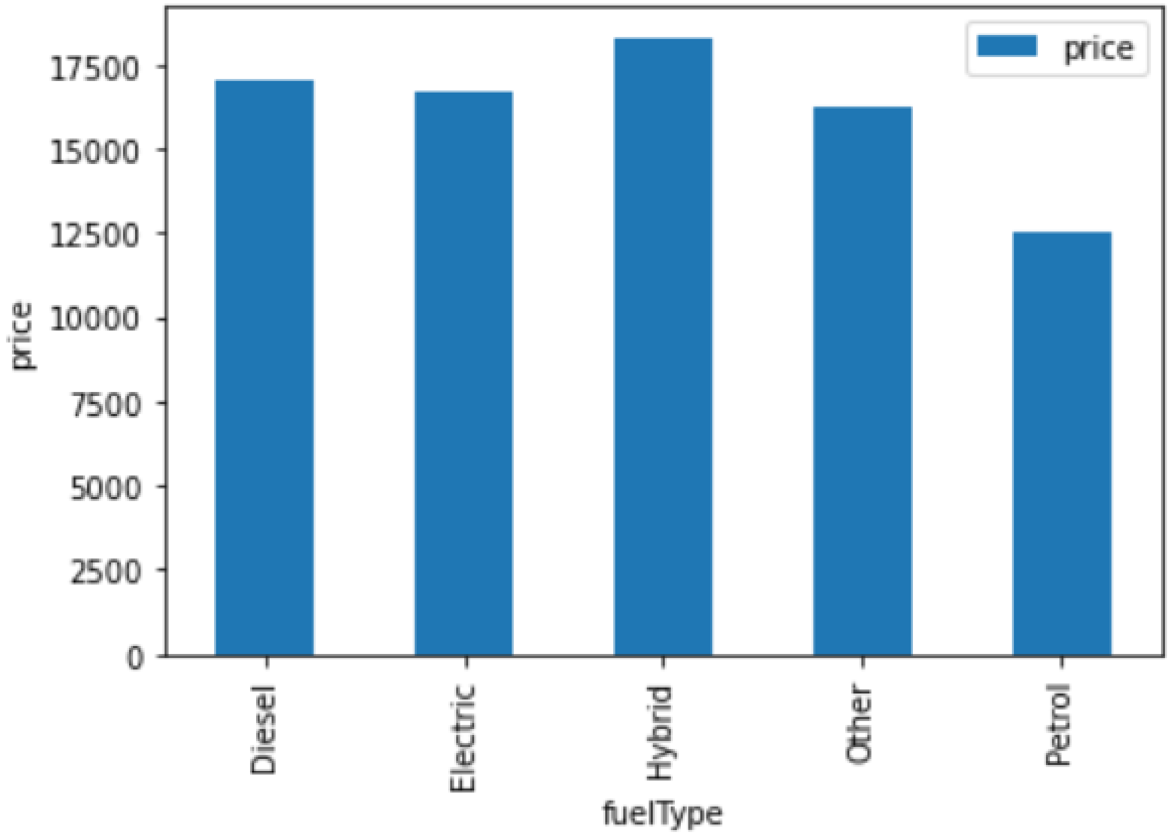


Comparación price - old



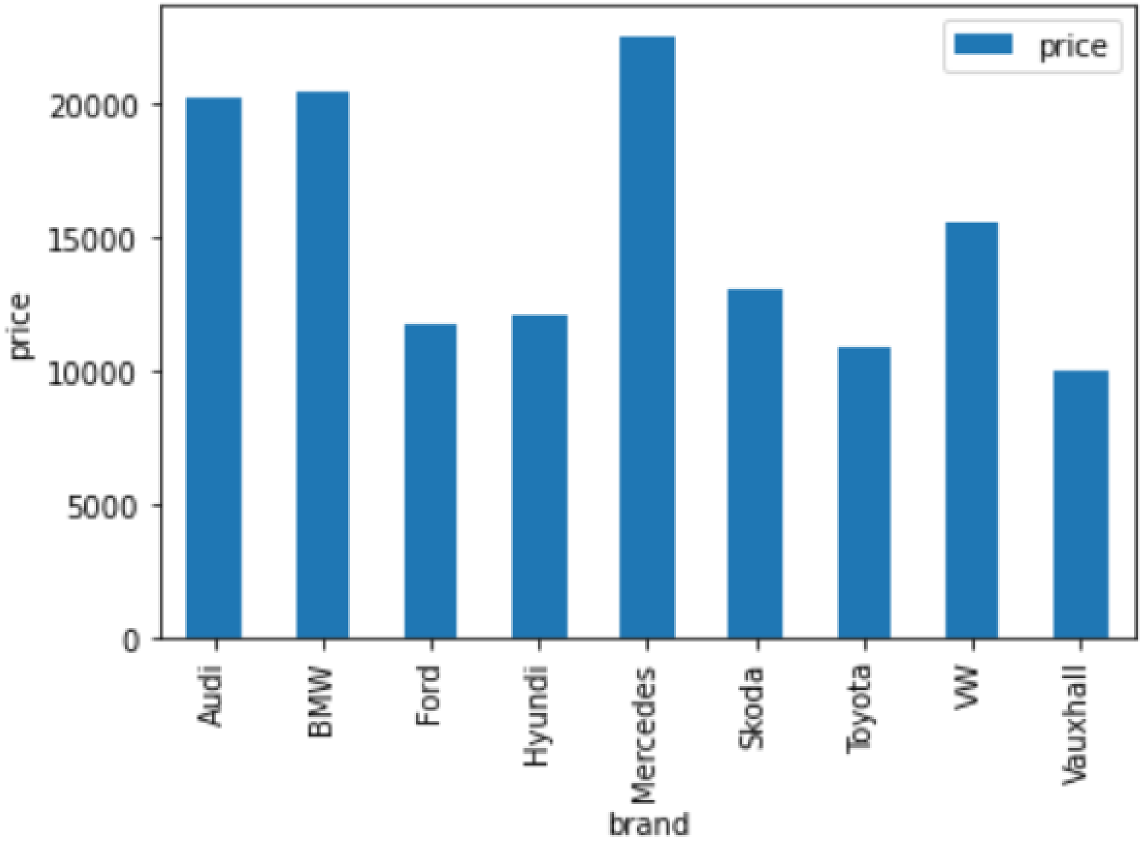
El precio del vehículo disminuye a medida que el coche es más antiguo

Comparación price - fuelType



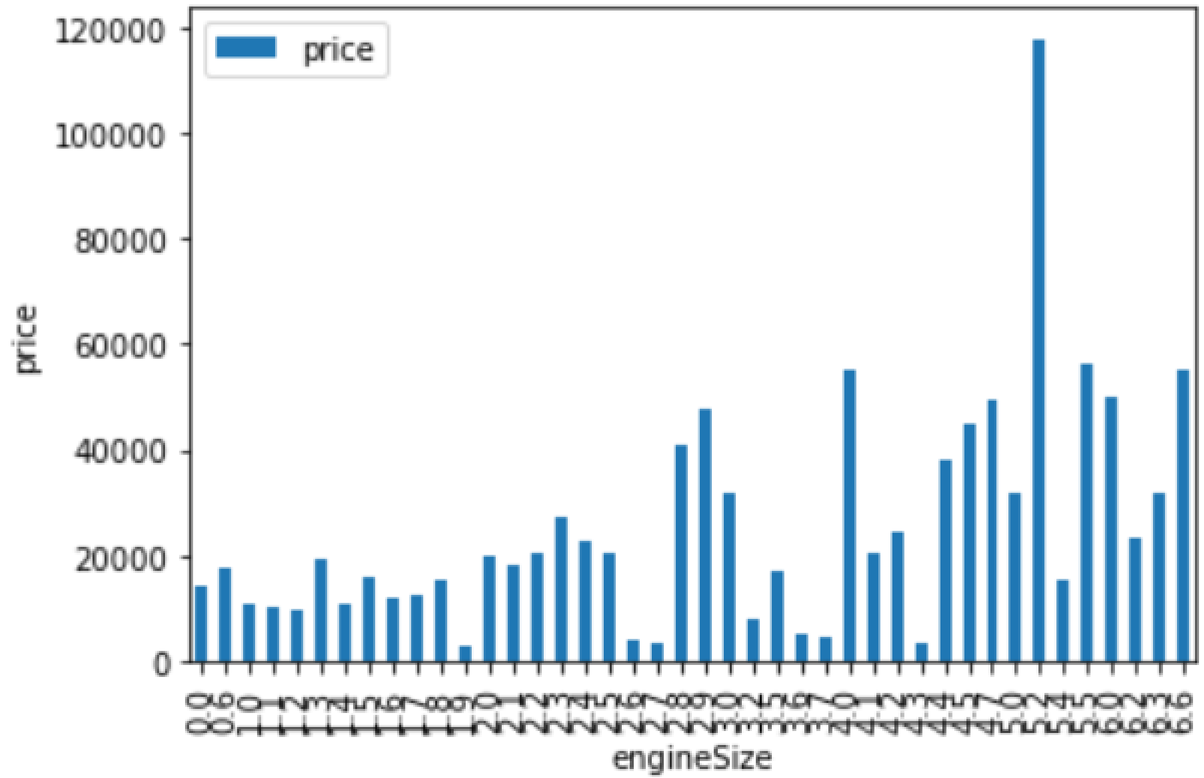
El tipo de fuel no influye significativamente en el precio del vehículo

Comparación price - brand



La marca del vehículo influye en el precio: Audi, BMW y Mercedes son las más caras

Comparación price - engineSize



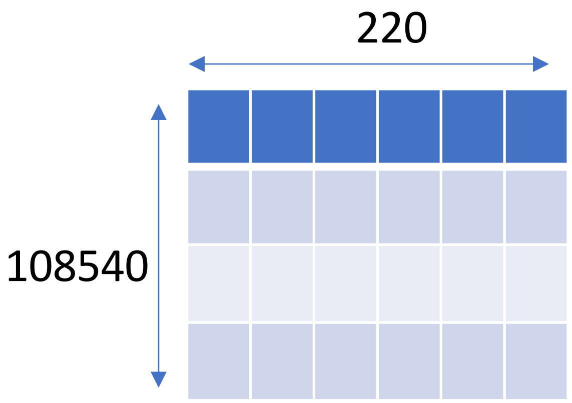
A mayor tamaño del motor del vehículo mayor precio

Ingeniería de características II

Variables continuas	Variables categóricas
• old • mileage • tax • mpg • engineSize	• model • transmission • fuelType • brand

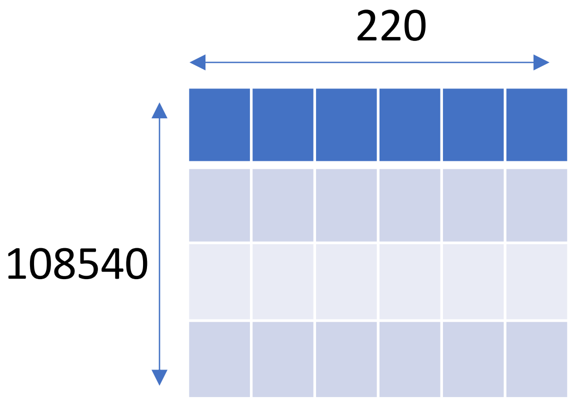
Scale --> media=0, desv=1
(asumimos normalidad)

One-Hot-Encoding



Dataset

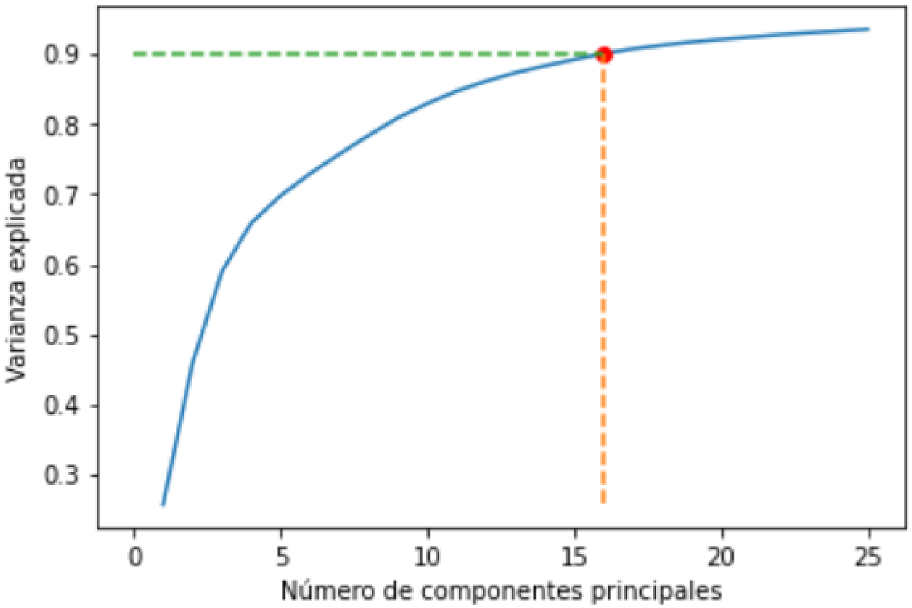
Reducción de dimensiones



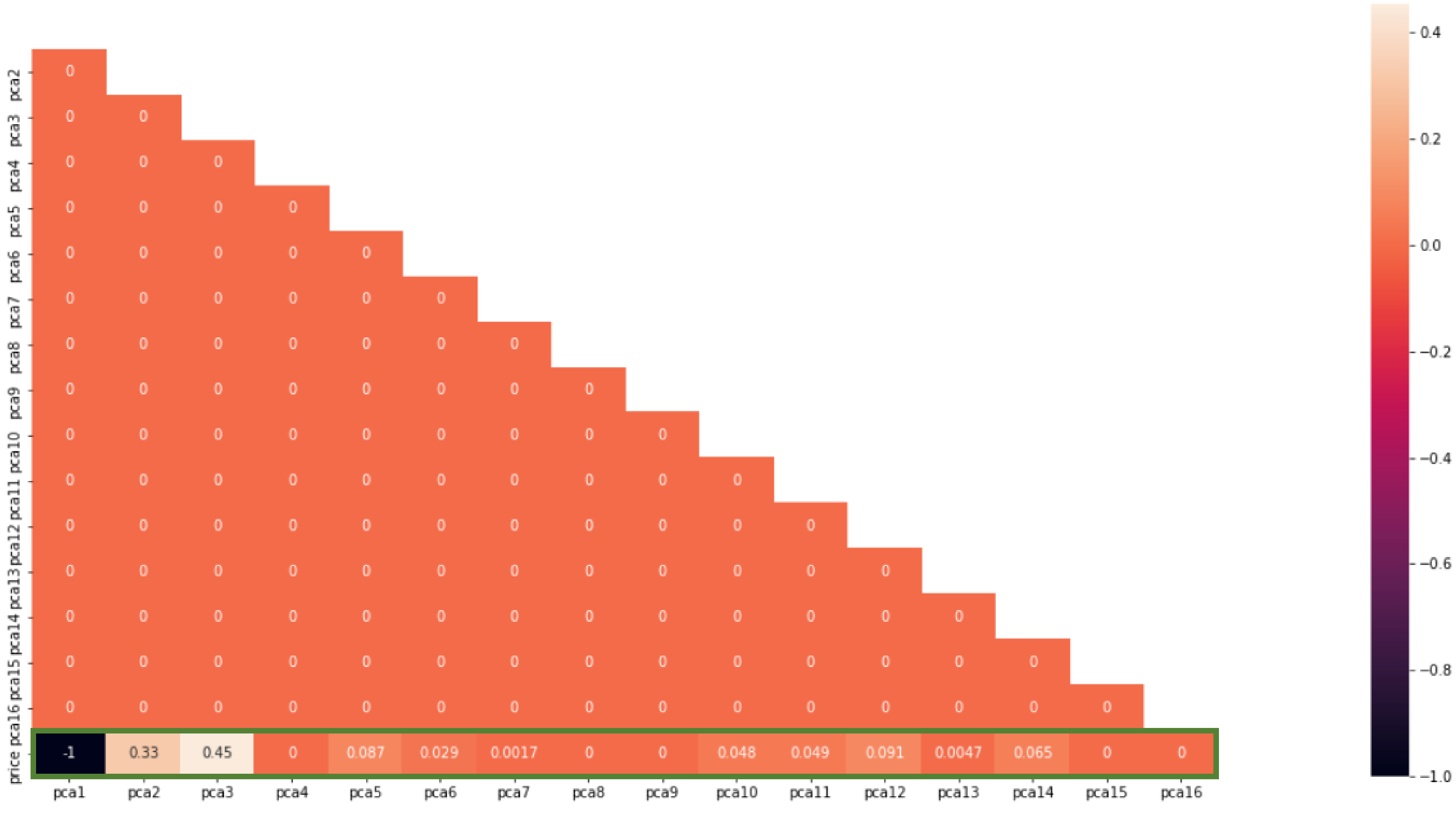
Dataset

Principal Component Analysis (PCA)

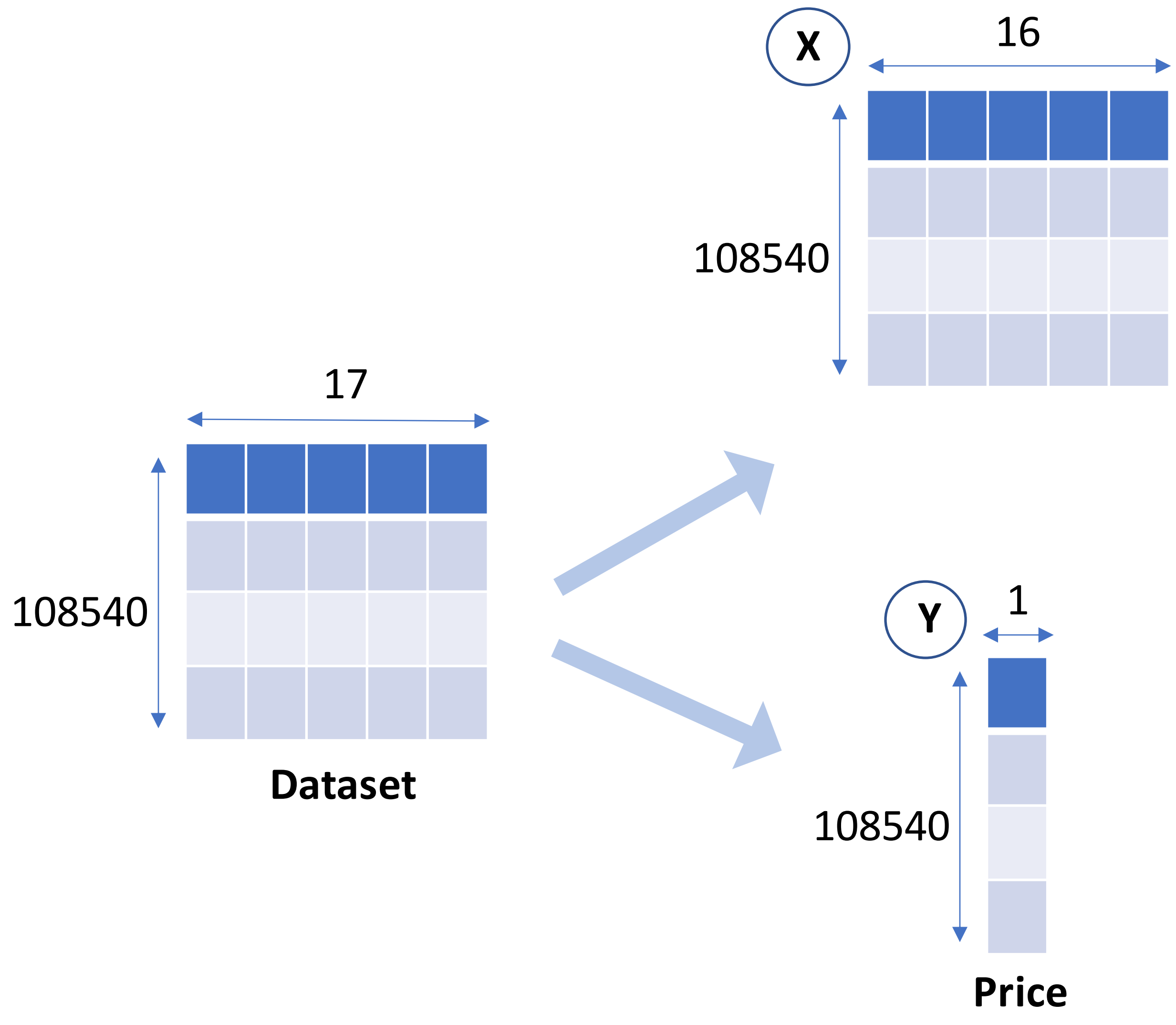
Con 16 componentes explicamos el 90% de las variables



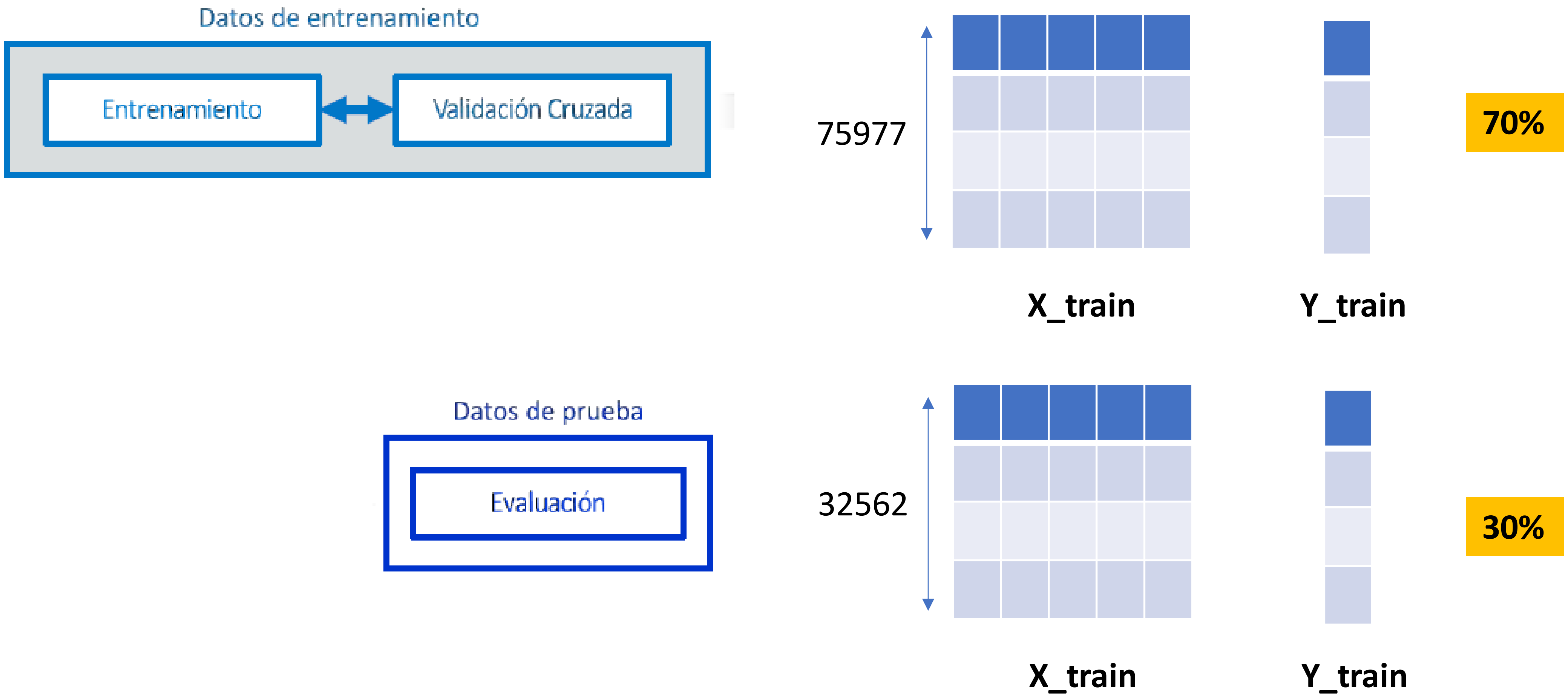
Matriz de correlación



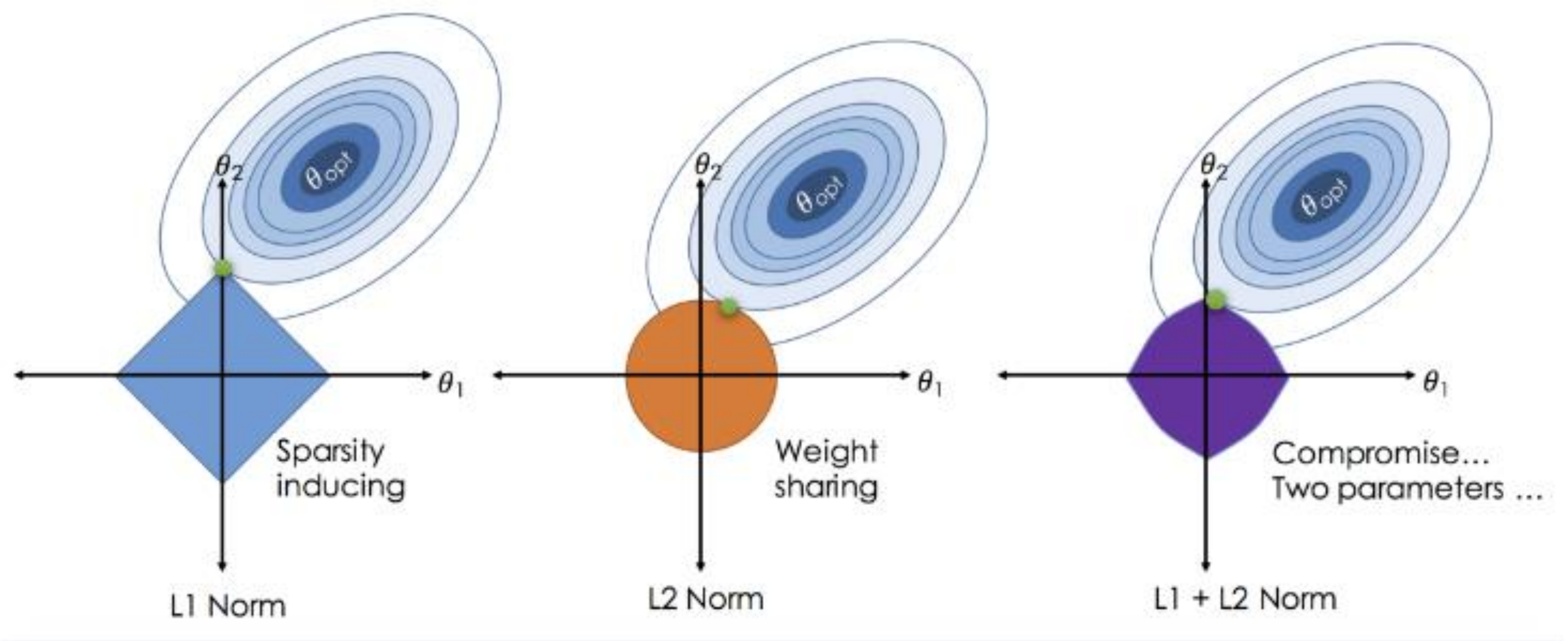
División del dataset

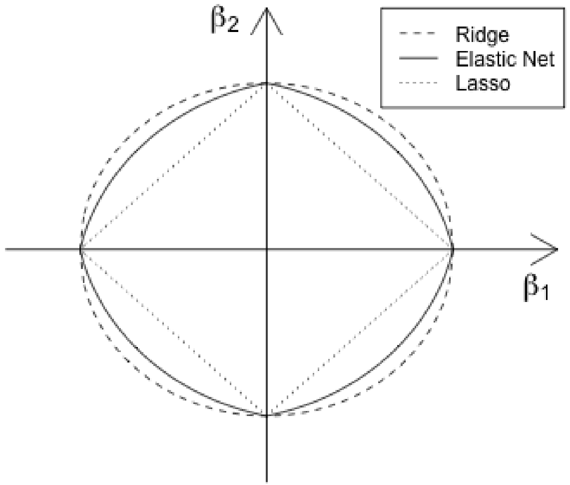


División del dataset



Regresión: ElasticNet





Regresión: ElasticNet

In [9]:
```
alpha = [0.001, 0.0001, 0.00001]
l1_ratio = [0.001, 0.0001, 0.00001, 0.000001]
parameters = {'alpha': alpha, 'l1_ratio': l1_ratio}
```

In [10]:
```
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import ElasticNet

gridCV = GridSearchCV(ElasticNet(), parameters, cv=5, n_jobs = -1)       # "n_jobs = -1" means "use all the CPU cores".
gridCV.fit(X_train, Y_train)
best_alpha = gridCV.best_params_['alpha']
best_l1_ratio = gridCV.best_params_['l1_ratio']
print("Best alpha : " + str(best_alpha))
print("Best l1_ratio : " + str(best_l1_ratio))
```

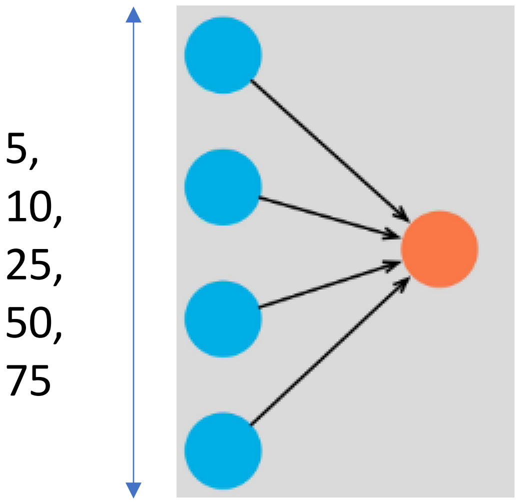
```
Best alpha : 0.0001
Best l1_ratio : 1e-06
```

In [11]:
```
elasticNet_best = ElasticNet(alpha=best_alpha, l1_ratio=best_l1_ratio, random_state=4815, fit_intercept=False)
elasticNet_best.fit(X_train,Y_train)
Y_pred = elasticNet_best.predict(X_test)
print( "Best RMSE : " + str(np.round(mean_squared_error(Y_test,Y_pred,squared=False, multioutput='raw_values'),3)))
```

```
Best RMSE : [17525.614]
```

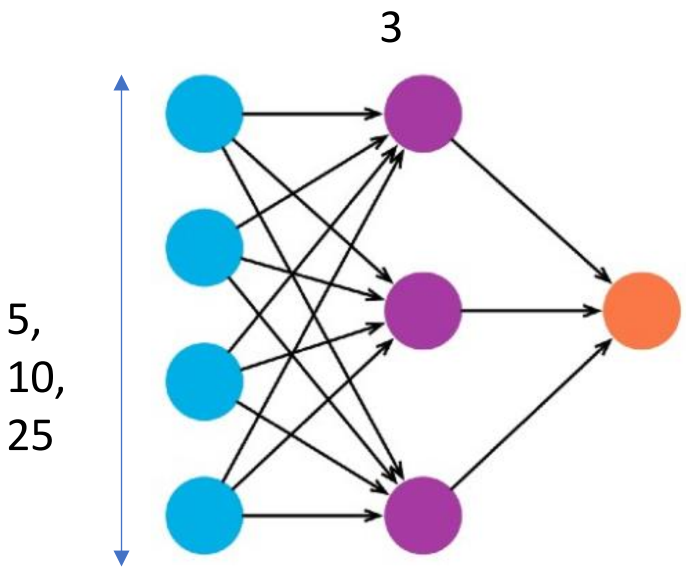
- ✓ Se puede probar con diferentes valores para alpha y l1_ratio.
- ✓ Mediante el mismo proceso se pueden obtener los mejores hiperparámetros para este modelo y conjunto de datos.
- ✓ En esta ocasión son l2 igual a 0,0001 y l1_ratio igual a 0,000001.

Regresión: Redes Neuronales Densas (I)



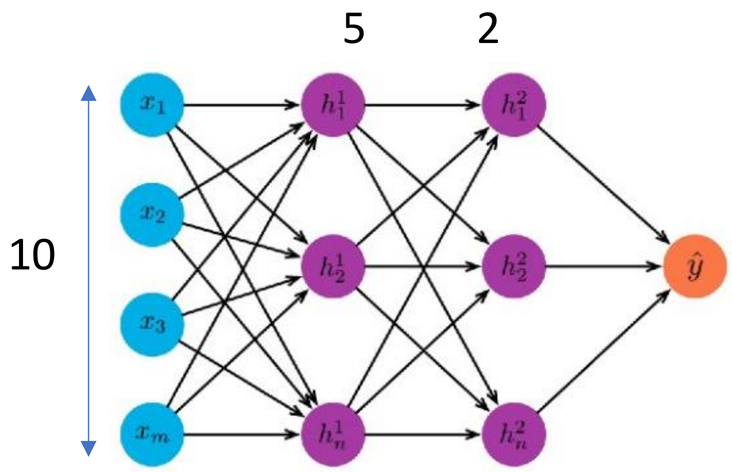
Input layer

RMSE = 4344, MAE = 2901
RMSE = 4420, MAE = 2717
RMSE = 4113, MAE = 2644
RMSE = 3966, MAE = 2500
RMSE = 3875, MAE = 2418



1 hidden layer

RMSE = 3911, MAE = 2476
RMSE = 3759, MAE = 2348
RMSE = 4113, MAE = 2644

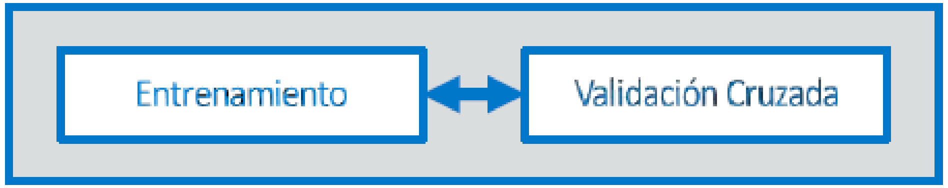


2 hidden layer

RMSE = 3486, MAE = 2172,

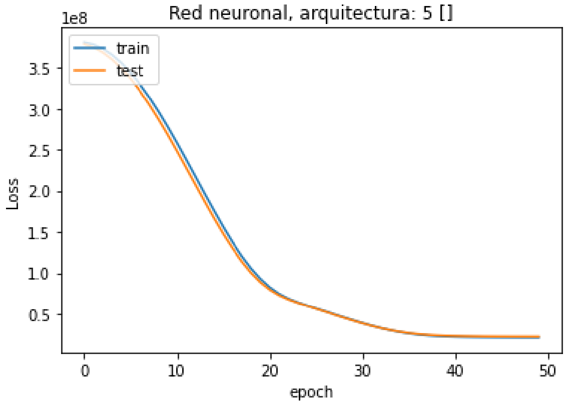
Regresión: Redes Neuronales Densas (II)

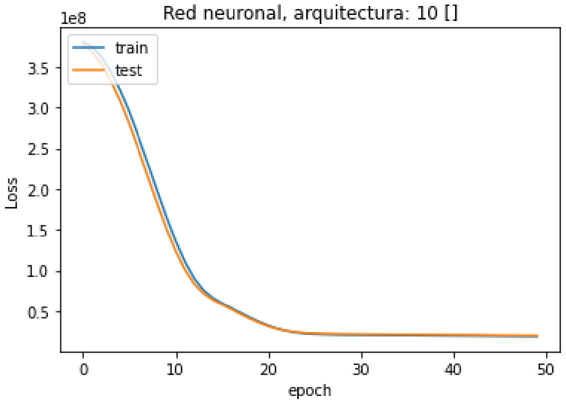
Datos de entrenamiento

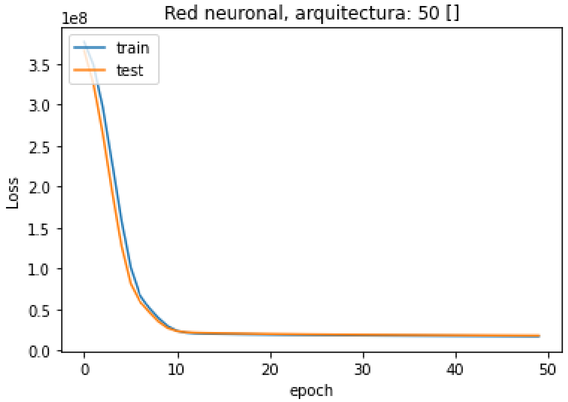


KFOLDS = 5

Datos de prueba







¿Alguna pregunta?