1 Regresion Lineal

- Relizar una regresion lineal con el dataset **Advertising.csv**
- la variable 'y' es la comuna 'Sales'
- Graficar y verificar dicha regresion y metrica (por ej R2)

1) Nuestro target es la columna 'Sales', cual de los tres productos de comunicacion posee mejor indice de correlacion lineal

1. TV
2. Radio
3. Newspaper

2 Regresion Lineal Multiple

2) Si en lugar de tomar una sola variable tomamos las dos de mejor correlacion, el MSE Aumenta o disminuye?

1. Aumenta
2. Dismuniye

3 Feature engineer

Para estos ejercicios se utilizara el archivo **ML_Cars_dataset.csv**

Descricion del dataset

- aspiration : Aspiration used in a car std Standard turbo Turbo

- enginelocation : Location of the engine on the car front Front rear Rear
- carwidth : Width of the car
- curbweight : Weight of the car
- enginetype : Type of engine on the car
- cylindernumber : Number of cylinders on the car
- stroke : Stroke of the car
- peakrpm : Peak RPM of the car
- price : Whether the car is considered to be expensive or cheap

- Remover registros duplicados

- Ver Valores nulos

3) El porcentaje de valores faltantes es mayor a 30%

1. Verdadero
2. Falso

4 Simple Imputer

- Imputar a los valores faltante del campo 'carwidth' con la mediana de toda la columna (explore bien esta columna)

4) Que estrategia utilizariamos en SimpleImputer

1. 'mean'
2. 'median'
3. 'most_frequent'

4.1 Detectando Outliers

Realizar lo mismo pero con la columna 'enginelocation'

5 Escalado de valores

Aplique StandarScaler en las siguientes columnas: peakrpm , carwidth , stroke y curbweight con sus respectivas distribuciones

5) Al Aplicar StandardScaler sobre cualquier campo , cambia realmente sus distribución.

1.Verdadero
2.Falso

Visualice sudataset

6) Usamos Standart scaler para...

1. que el modelo se ajuste mejor a los datos
2. no considerar los outliers
3. llegar a un score = 1

6 Feature encoding

Ahora nos dedicaremos a las variables categoricas, que son:

aspiration , enginelocation, enginetype y cylindernumber

- Realizar un OneHotEncoder sobre la columna 'aspiration y enginelocation' para obtener valores binarios

- Realizar un OneHotEncoder sobre la columna 'enginetype'

- Realizar un cambio del tipo de valor a la columna cylindernumber seguido de un MinMaxScaler

- Realizar un labelencoder a la columna **Price**

- Visualice el tamaño del dataset

7) Cuantas columnas quedaron en el dataset

1. 9
2. 6
3. 15

7 Correlacion de columnas

8)Metimos valores binarios a la columna **Price**, ahora queremos realizar una clasificación tomando esta última como variable objetivo.
Ves variables que estén minimamente correlacionada y sea conveniente quitar una de las dos? Cuáles?
1.carwidth y curbweight
2.ohcv y enginelocation
3.dohcv y l

- Elimina la variable que consideres adecuada

8 Clasificacion

- Realizar un modelo de Regresion logistica junto con una validacion cruzada
- De dicha validacion determine el score promedio (utilizamos el score por default)

9) En qué rango se encuentra el accuracy del modelo?

1. [0.7 , 0.75]
2. [0.55 , 0.65]
3. [0.85 , 0.9]
4. [0.95 , 1.0]

9 Metricas

In [1]:
```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

y_true = [0, 1, 0, 0, 1, 0, 1, 1, 0, 1]
y_pred = [0, 0, 0, 0, 1, 1, 1, 1, 1, 1]
```

10) Cual es el valor de la Exhaustividad (Recall)?

- 1. 0.5

- 2. 0.75
- 3. 0.8
- 4. 0.95

10 Ajustando Metricas

In [2]:
```
from sklearn.metrics import confusion_matrix,accuracy_score
y_test = [0, 1, 0, 0, 1, 0, 1, 1, 0, 1] # actual truths
preds = [0, 0, 0, 0, 1, 1, 1, 1, 1, 1] # predictions
```

11) Si nos enfocamos en la clase de los **1** ¿Cuál es la cantidad de verdaderos positivos?

1.4

2.3

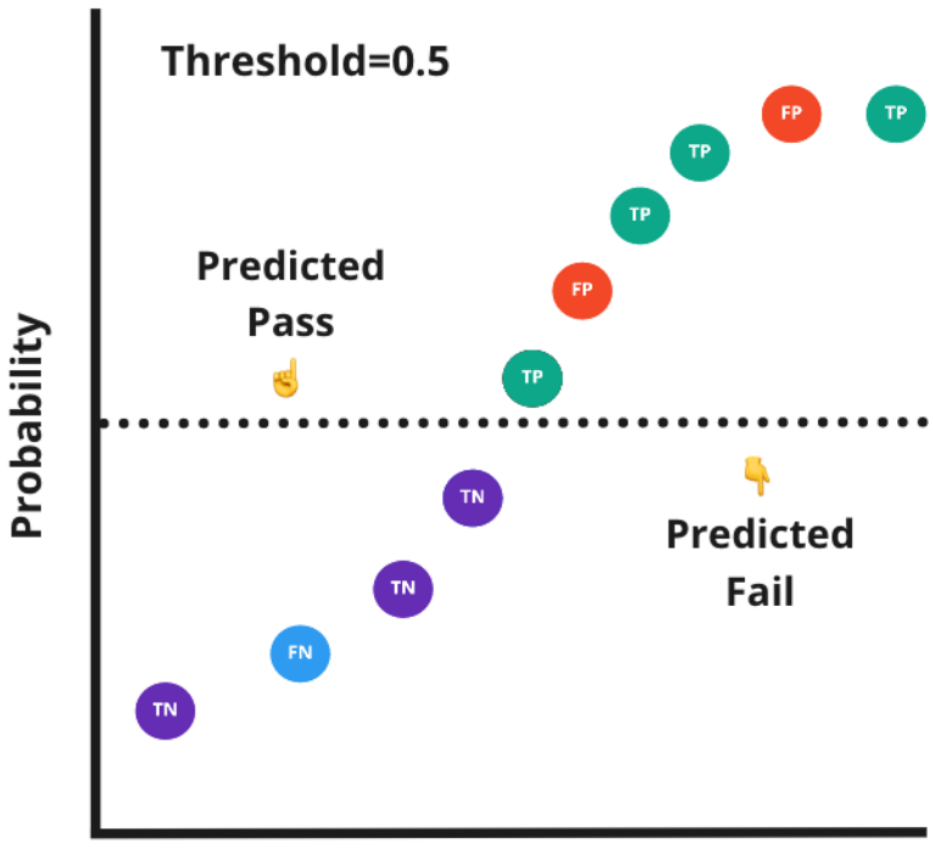
3.2

4.1

10.1 Ajuste de metricas

Tomando en cuenta un umbral (threshold) de 0.5 tenemos el siguiente comportamiento

Let's go back to our exam predictions...



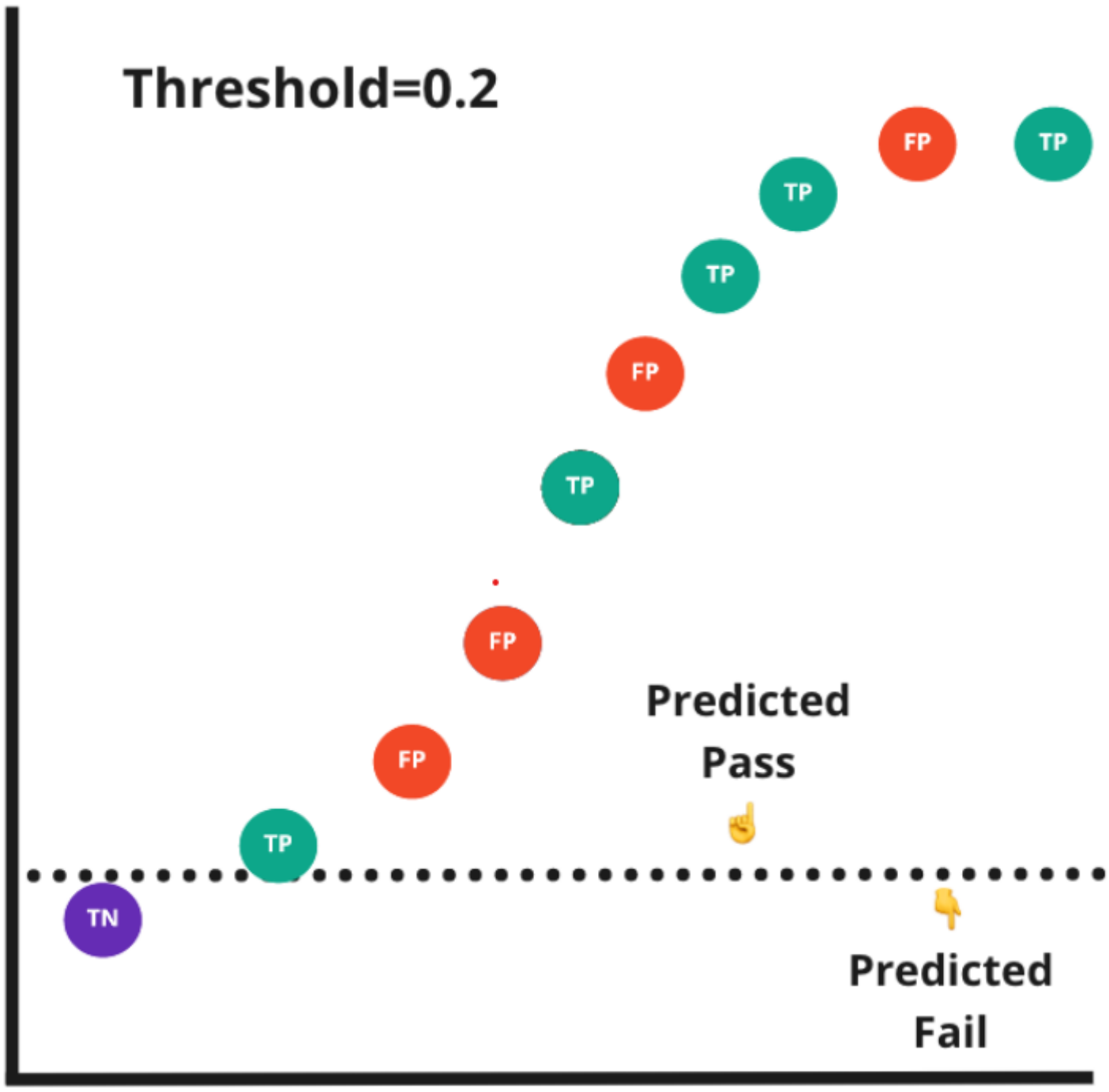
	Predicted ❌	Predicted ✅
Actual ❌	TN=3	FP=2
Actual ✅	FN=1	TP=4

$$recall = \frac{TP}{TP+FN} = \frac{4}{4+1} = 0.80$$

$$precision = \frac{TP}{TP+FP} = \frac{4}{4+2} = 0.67$$

$$F_1 = 2 \cdot \frac{precision \times recall}{precision+recall} = 2 \cdot \frac{0.67 \times 0.80}{0.67+0.80} = 0.73$$

- Ahora bien, tomemos la condicion de un umbral (threshold) es 0.2. Junto con su matriz de confusion.

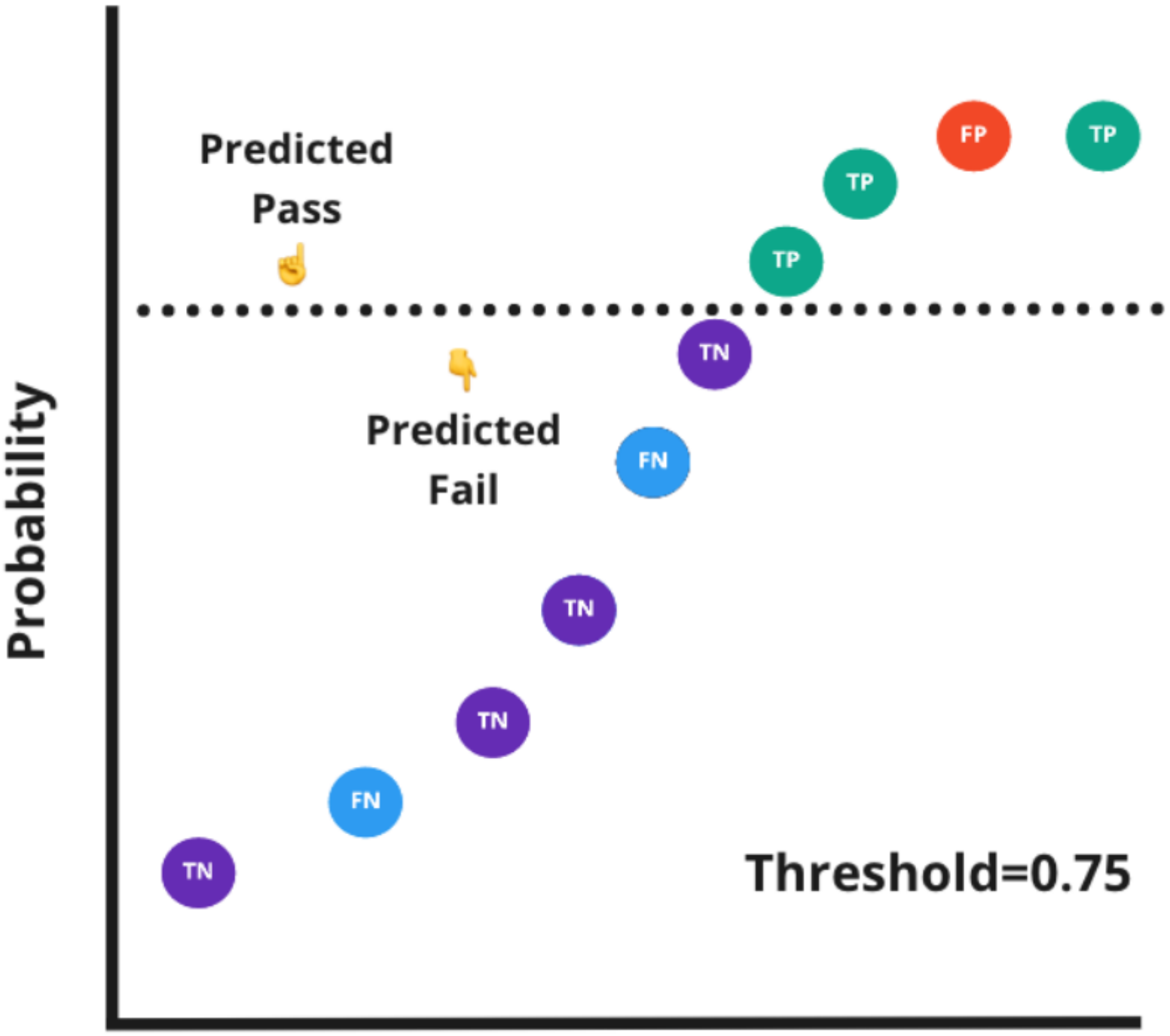


	Predicted ❌	Predicted ✅
Actual ❌	TN=1	FP=4
Actual ✅	FN=0	TP=5

12) La presición aumento?

1. Verdadero
2. Falso

- Ahora tomemos la condicion en el cual nuestro umbral (threshold) es de 0.75. Junto con su matriz de confusion



	Predicted ❌	Predicted ✅
Actual ❌	TN=4	FP=1
Actual ✅	FN=2	TP=3

13) La exhaustividad (recall) aumento?

1. Verdarero
2. Falso

11 KNN

- Segun los datos a unilizar determine el mejor numero de vecinos para realizar la regresion

In [39]:
```
df2 = pd.read_csv("./datasets/ML_Houses_clean.csv")
```

In [47]:
```
df2.head()
```

Out[47]:

	GrLivArea	BedroomAbvGr	KitchenAbvGr	OverallCond	CentralAir	SalePrice
0	0.380070	0.375	0.333333	0.500	1	208500
1	-0.312090	0.375	0.333333	0.875	1	181500
2	0.497489	0.375	0.333333	0.500	1	223500
3	0.390885	0.375	0.333333	0.500	1	140000
4	1.134029	0.500	0.333333	0.500	1	250000

In [56]:
```
from sklearn.neighbors import KNeighborsRegressor
from sklearn.model_selection import cross_validate
from sklearn.preprocessing import MinMaxScaler

knn_model = KNeighborsRegressor()

X1 = df2.drop(columns = ['SalePrice'])
y1 = df2.SalePrice

from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

X1_rescaled = scaler.fit_transform(X1)

cv_results = cross_validate(knn_model, X1_rescaled, y1, cv=5)
```

In [57]:
```
rescaled_score = cv_results['test_score'].mean()
rescaled_score
```

Out[57]:
```
0.649019431450802
```

14) Realiza una lista de score para cada valor de k entre 1 y 24 y plotear el resultado.
El mejor valor de K es
1.3
2.11
3.20

12 K-means

In [61]:
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.datasets import make_blobs
import warnings
warnings.filterwarnings('ignore')
```

In [62]:
```
random_state=42
X, y = make_blobs(n_samples=500, centers=4, random_state=random_state)
```

15) Teniendo en cuenta la metrica de la distancia media al centroide.
Cual seria el valor optimo de K?

1. 1
2. 2
3. 10
4. 4

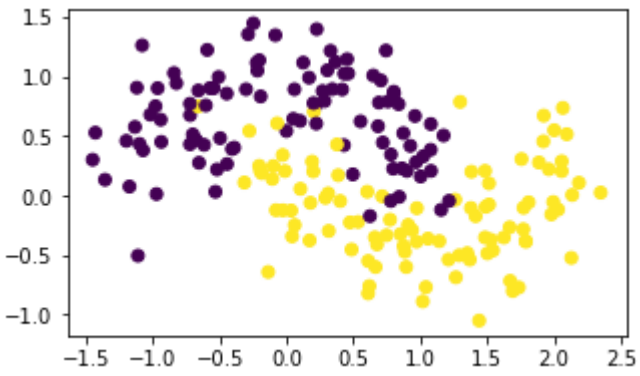
13 SVM + GridShearch

In [70]:
```
import numpy as np
import matplotlib.pyplot as plt

from matplotlib import rcParams
rcParams['figure.figsize'] = (5,3)
```

In [71]:
```
from sklearn.datasets import make_moons

n=200
X,y = make_moons(n_samples=n, noise=0.25, random_state=0)
plt.scatter(X[:,0], X[:,1], c=y);
```



In [4]:
```
from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV
from scipy import stats


# Hyperparameter search space
search_space = {
    'kernel': ['sigmoid', 'rbf'],
    'C': [0.01, 0.1, 1,10,100,1000],
    'gamma': [0,0.1,1,10,100],
    'coef0': [0,0.1,1],
}
```

16) Según GridSearch, cuáles son los mejores hiperparámetros:

1. {'C': 0.01, 'coef0': 0, 'gamma': 1, 'kernel': 'sigmoid'}
2. {'C': 1, 'coef0': 0, 'gamma': 1, 'kernel': 'rbf'}
3. {'C': 1, 'coef0': 0, 'gamma': 1, 'kernel': 'sigmoid'}
4. {'C': 0.1, 'coef0': 0, 'gamma': 1, 'kernel': 'rbf'}

14 Descenso de Gradiente

17) Cual es el objetivo concreto del algoritmo de descenso de gradiente

- 1. Ajustar la recta de regresion
- 2. Buscar un mínimo global mediante iteraciones en las cuales se va descendiendo en la función de costo
- 3. Obtener mejor score

15 Deep Learning

18) Donde son mas utilizadas las redes neuronales convolucionales (CNN)

- 1. En series temporales
- 2. En procesamiento de lenguaje natural
- 3. En procesamiento de imagenes

16 Pipelines

19) Realizar un pipeline con las siguientes caracteristicas:

- utilizar el dataset **data.csv**, en el el cual el y es la columna "target_5y"
- realizar un los siguientes **pipelines**:

- uno de preprocesamiento usando **MinMaxScales** y un **SimpleImputer** con estrategia promedio para los nulos
- Despues integrarlo todo con un pipeline en el cual este el modelo SVC
- Acto seguido realizar un **RandomizedSearchCV** con los siguientes atributos:

```
-'preprocessing__imputer__strategy': ['mean', 'median','most_frequent'],
 'model_svm__kernel' : ['linear', 'poly', 'rbf', 'sigmoid'],
 'model_svm__C': uniform(0.1,10)},
  cv=5,
  n_iter = 50,
  scoring="precision"
```

  - A este modelo lo llamaran yunned_pipe
  - realizan un .fit() con el tunned_pipe
- Por ultimo relizamos una validacion cruzada con un maximo de cv = 5 un scoring tipo "precision"

Cuanto es el resultado del score?

- 1. [0.4-0.6]
- 2. [0.9-1.0]
- 3. [0.7-0.8]

17 Modelo ensamblado

20) Cual de estos modelos de ensamble, no tienen un entrenamiento paralelizable

- 1. Bagging
- 2. Random Forest
- 3. Boosting