# Apunts codis- random forest- regressor

## 1. LLIBRERIES

**from sklearn.ensemble import RandomForestRegressor**

**from sklearn.model\_selection import train\_test\_split**

## 2. DEFINO X(FEATURE), Y (TARGET)

**features = df.drop(columns=['House\_Price'])**

**target = df['House\_Price']**

## 3. DIVIDO DATOS TRAIN Y TEST

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(features, target, test\_size=0.2, random\_state=29)**

## 4. Estandarización

#defino un nombre con la funcion de estandarización

**scaler = StandardScaler()**

**#importante la estandarizacion es para ayudar al modelo a que las variables tengan el mismo peso y no coja de referencia que por tener mas rango es mas importante**

**X\_train\_scaled = scaler.fit\_transform(X\_train)**

**X\_test\_scaled = scaler.transform(X\_test)**

## 5. Defino modelos y Parametros

**Importante el n\_estimators es el numero de arboles que quieres crear**

**rf\_model = RandomForestRegressor(n\_estimators=1000, random\_state=29)**

## 6. Train del modelo

**rf\_model.fit(X\_train\_scaled, y\_train)**

## 7. Predicción Modelo

**rf\_model\_pred = rf\_model.predict(X\_test\_scaled)**

**rf\_model\_pred**

## 8. Metricas regressión

## 9. importancia variables para las predicciones

Este bloque de código es para **visualizar la importancia de cada característica** (o variable de entrada) en el modelo de **Random Forest**. La importancia de características (feature\_importances\_) en un modelo de bosque aleatorio indica qué tanto cada variable contribuye a la precisión de las predicciones.

librerias: **import seaborn as sns**

**import matplotlib.pyplot as plt**

**importances = rf\_model.feature\_importances\_**

**feature\_names = X\_train.columns**

**rf\_model.feature\_importances\_ extrae las importancias de cada característica del modelo RandomForest.**

**feature\_names = X\_train.columns asigna los nombres de las columnas de X\_train a feature\_names para etiquetar las características en el gráfico.**

**plt.figure(figsize=(10,6))**

**sns.barplot(x=importances, y=feature\_names, palette = 'viridis')**

**plt.title('Feature Importance Plot (Random Forest)')**

**plt.xlabel('Importance')**

**plt.ylabel('Features')**

**plt.show()**

**plt.figure(figsize=(10,6)) establece el tamaño del gráfico.**

**sns.barplot(x=importances, y=feature\_names, palette='viridis') crea un gráfico de barras que muestra la importancia de cada característica, usando la paleta de colores viridis.**

**Los métodos plt.title, plt.xlabel, y plt.ylabel agregan títulos y etiquetas.**