import pandas as pd

df = pd.read\_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/housing/housing.data',header=None,sep='\s+')

df.columns=['CRIM','ZN','INDUS','CHAS','NOX','RM','AGE','DIS','RAD','TAX','PTRATIO','B','LSTAT','MEDV']

df.head()

RM e INDUS son dos variables con correlación, las tomaremos ambas33

from sklearn.preprocessing import StandardScaler

from sklearn.linear\_model import LinearRegression

#NO LE DAREMOS RESHAPE A RM CON INDUS

X = df[['RM','INDUS']].values

y = df['MEDV'].values.reshape(-1, 1)

sc\_x = StandardScaler()

sc\_y = StandardScaler()

X\_std = sc\_x.fit\_transform(X)

y\_std = sc\_y.fit\_transform(y)

slr = LinearRegression()

slr.fit(X\_std, y\_std)

PREDECIR

import numpy as np

rooms = 5

indus =3

v1 = np.array([rooms,indus]).reshape(1,-1)

precio = sc\_y.inverse\_transform(slr.predict(v1))

print("El precio de una casa con 5 habitaciones y 3 indus en Boston es de ", precio)

El precio de una casa con 5 habitaciones y 3 indus en Boston es de [[43.10323352]]

import matplotlib.pyplot as plt

from mpl\_toolkits.mplot3d import \*

import numpy as np

x1\_range = np.arange(df['RM'].min(),df['RM'].max())

x2\_range = np.arange(df['INDUS'].min(),df['INDUS'].max())

X1, X2 = np.meshgrid(x1\_range,x2\_range)

plano = pd.DataFrame({'RM':X1.ravel(), 'INDUS':X2.ravel()})

pred = slr.predict(plano).reshape(X1.shape)

pred = sc\_y.inverse\_transform(pred)

fig = plt.figure()

ax = fig.gca(projection='3d')

ax.plot\_surface(X1,X2,pred, alpha=0.4)

ax.scatter3D(df['RM'], df['INDUS'], df['MEDV'], color='R', marker='.')

#ROTARLO

ax.view\_init(elev=10, axim=5)

plt.show();

Chart, scatter chart

Description automatically generated

SOLO VER EL PLANO

import matplotlib.pyplot as plt

from mpl\_toolkits.mplot3d import \*

import numpy as np

# RANGOS

x1\_range = np.arange(df['RM'].min(), df['RM'].max())

x2\_range = np.arange(df['INDUS'].min(), df['INDUS'].max())

X1, X2 = np.meshgrid(x1\_range, x2\_range)

plano = pd.DataFrame({'RM':X1.ravel(), 'INDUS':X2.ravel()})

pred = slr.predict(plano).reshape(X1.shape)

pred = sc\_y.inverse\_transform(pred)

Z = pred

from matplotlib import cm

my\_col = cm.jet(Z/np.amax(Z))

fig , ax = plt.subplots(subplot\_kw={"projection":"3d"})

surf = ax.plot\_surface(X1,X2,pred, facecolors = my\_col)

fig.colorbar(surf);

