from keras.models import Sequential

from keras.layers import Dense

from keras.callbacks import EarlyStopping

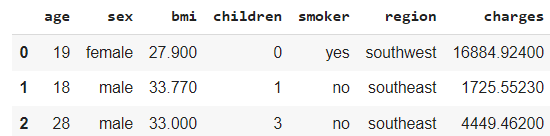
from sklearn.preprocessing import StandardScaler

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

import pandas as pd

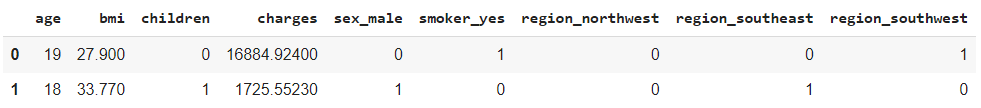
df = pd.read\_csv('insurance.csv')

df.head()



**df = pd.get\_dummies(df, columns=['sex','smoker','region'], drop\_first=True)**

**df.head()**



X\_cols = list(set(df.columns)-set(['charges']))

y\_col = ['charges']

X = df[X\_cols].values

y = df[y\_col].values

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,y)

sc\_x = StandardScaler().fit(X)

X\_train = sc\_x.transform(X\_train)

X\_test = sc\_x.transform(X\_test)

sc\_y = StandardScaler().fit(y)

y\_train = sc\_y.transform(y\_train)

y\_test = sc\_y.transform(y\_test)

model=Sequential([

         Dense(128,input\_dim=X\_train.shape[1], activation='relu'),

         Dense(32,activation='relu'),

         Dense(1)

  ])

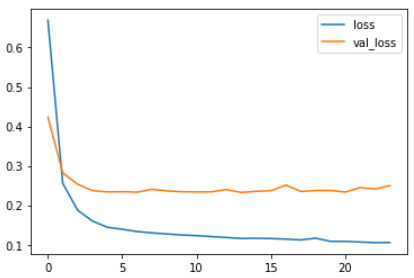
model.compile(optimizer='adam',loss='mse')

early=EarlyStopping(monitor='val\_loss', patience=10)

epocas=400

history=model.fit(X\_train,y\_train,epochs=epocas,validation\_split=0.25,callbacks=[early])

pd.DataFrame(history.history).plot();



y\_test y X\_test tienen shape (335, 8)

from sklearn.metrics import r2\_score

print("R2 ",r2\_score(y\_test,model.predict(X\_test)).round(3))

R2 0.837

MEJOR PREDICCION

PREDECIR….

model.predict(X\_test)

p0= X\_test[0]

p1= X\_test[1]

import numpy as np

arr = np.vstack([p0,p1])

model.predict(arr)

p1= X\_test[1]

q = model.predict( np.array( [p1,] )  )

q