**12.Logistic Regression**

**NAME : P JAYAPRADHA**

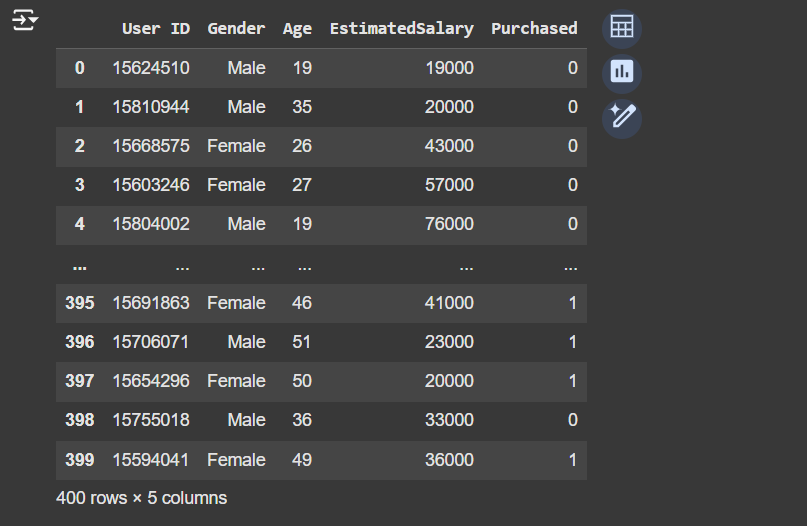
**ROLL NO : 230701130**

import numpy as np

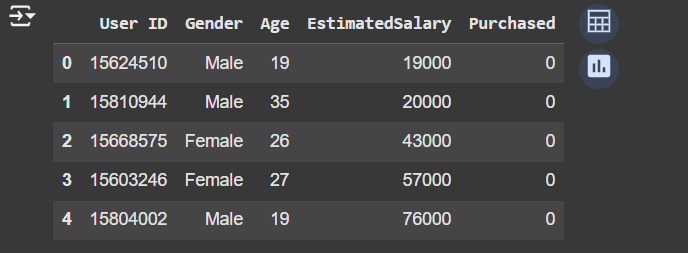
import pandas as pd

df=pd.read\_csv('Social\_Network\_Ads.csv')

df



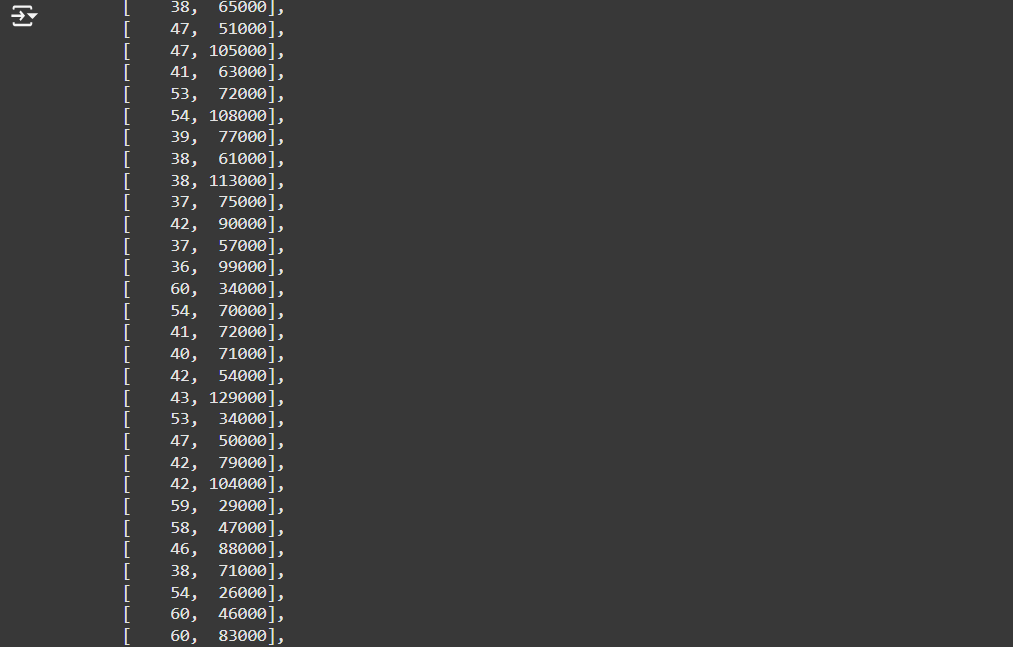
df.head()



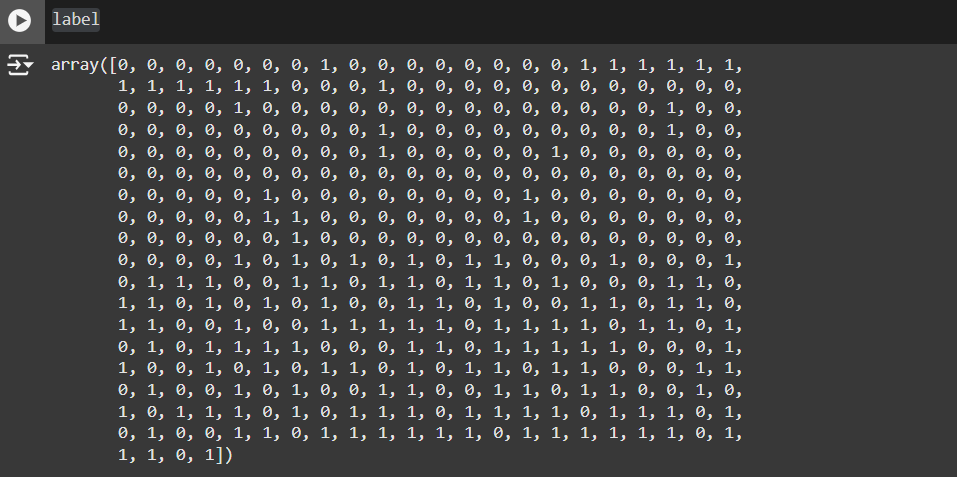
features=df.iloc[:,[2,3]].values

label=df.iloc[:,4].values

features



label



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

from sklearn.linear\_model import LogisticRegression

for i in range(1, 401):

    # Split the data into training and testing sets

    x\_train, x\_test, y\_train, y\_test = train\_test\_split(features, label, test\_size=0.2, random\_state=i)

    # Initialize the Logistic Regression model

    model = LogisticRegression()

    # Train the model

    model.fit(x\_train, y\_train)

    # Calculate the train and test scores

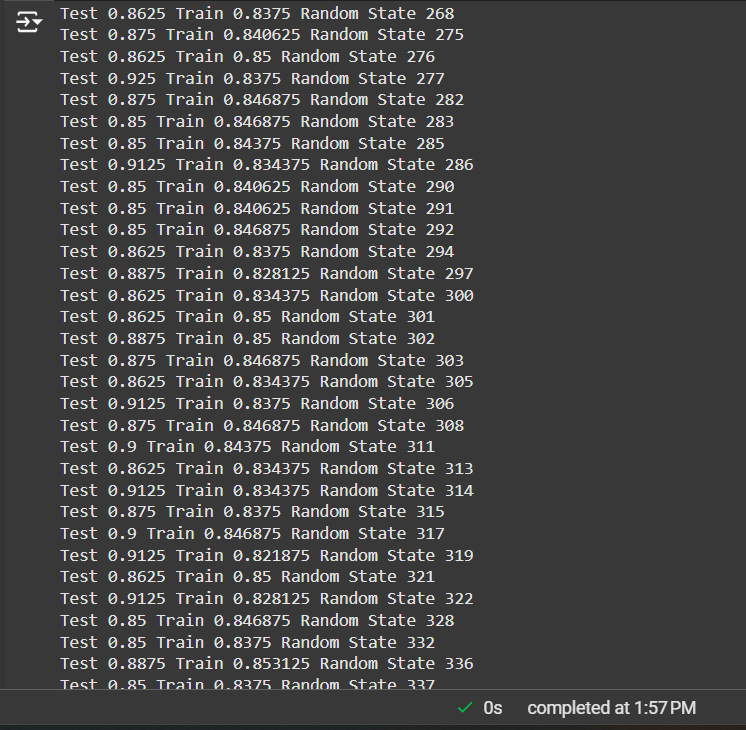
    train\_score = model.score(x\_train, y\_train)

    test\_score = model.score(x\_test, y\_test)

    # Print if test score is greater than train score

    if test\_score > train\_score:

        print("Test {} Train {} Random State {}".format(test\_score, train\_score, i))

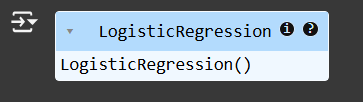


# Assuming features and label are defined earlier in your code

x\_train, x\_test, y\_train, y\_test = train\_test\_split(features, label, test\_size=0.2)

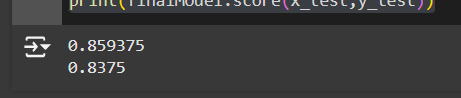
finalModel = LogisticRegression()

finalModel.fit(x\_train, y\_train)



print(finalModel.score(x\_train,y\_train))

print(finalModel.score(x\_test,y\_test))



from sklearn.metrics import classification\_report

print(classification\_report(label,finalModel.predict(features)))

