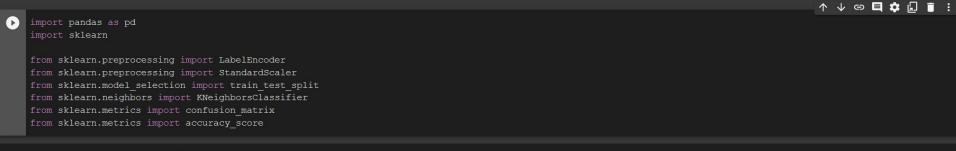
## Task-14

## GO\_STP\_5247

We have a dataset that contains multiple user's information through the social network who are interested in buying SUV Car or not.



[2] data=pd.read\_csv('/content/User\_Data.csv')

**3** 15603246 Female 27

[3] data.head() User ID Gender Age EstimatedSalary Purchased **0** 15624510 19000 1 15810944 20000 2 15668575 Female 43000 57000

Os completed at 5:25 PM

```
le = LabelEncoder()
    data['Gender'] = le.fit transform(data['Gender'])
    data.head()
        User ID Gender Age EstimatedSalary Purchased
     0 15624510
                                       19000
     1 15810944
                                       20000
     2 15668575
                                       43000
     3 15603246
                     0 27
                                       57000
     4 15804002
                                       76000
[6] data.drop(['User ID'],axis='columns',inplace=True)
[7] y=data['Purchased']
    X=data.drop(['Purchased'],axis='columns')
[8] scaler = StandardScaler()
    scaler.fit(X)
    X scaled=scaler.transform(X)
```

[9] print(X\_scaled)

```
[9] print(X scaled)
      [ 1.02020406 -0.15807423 -1.07893824]
[10] X train, X test, y train, y test = train test split(X scaled, y, test size=0.20, random state=42)
[11] knn=KNeighborsClassifier(n neighbors=5)
[12] knn.fit(X train,y train)
    KNeighborsClassifier(algorithm='auto', leaf size=30, metric='minkowski',
                          metric params=None, n jobs=None, n neighbors=5, p=2,
                          weights='uniform')
[15] y pred=knn.predict(X test)
    cm=confusion matrix(y test,y pred)
    print (cm)
    print("correct predicition", accuracy score(y test, y pred))
    print("wrong predicition", (1-accuracy score(y test, y pred)))
     [ 2 2611
    correct predicition 0.925
    wrong predicition 0.0749999999999996
```