import pandas as pd

df = df = pd.read\_csv("/content/suv\_data.csv")

df

df.info()

df.describe()

import pandas as pd

import numpy as np

import seaborn as sns

from matplotlib import pyplot as plt

plt.figure(figsize=(25, 7))

sns.countplot(x='Gender', hue='Purchased', data=df)

plt.show()

plt.figure(figsize=(25, 7))

sns.countplot(x='EstimatedSalary', hue='Purchased', data=df)

plt.show()

plt.figure(figsize=(16, 14))

plt.subplot(2, 1, 1)

p = sns.barplot(x='Age', y='EstimatedSalary', hue='Purchased', data=df, estimator=np.sum)

p.set\_xticklabels(p.get\_xticklabels(),rotation=30)

plt.title('Total Salary')

df["Gender"] = df["Gender"].map({"Male":1, "Female":2})

df = df.drop('User ID', axis=1)

X = df.iloc[:, :3]

print(X)

Y = df['Purchased']

print(Y)

# Load libraries

from pandas.plotting import scatter\_matrix

from matplotlib import pyplot

#machine learning models

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

from sklearn.model\_selection import cross\_val\_score

from sklearn.model\_selection import StratifiedKFold

from sklearn.metrics import classification\_report

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import accuracy\_score

#Algorithm

from sklearn.linear\_model import LogisticRegression

from sklearn.tree import DecisionTreeClassifier

from sklearn.neighbors import KNeighborsClassifier

from sklearn.discriminant\_analysis import LinearDiscriminantAnalysis

from sklearn.naive\_bayes import GaussianNB

from sklearn.svm import SVC

LR = LogisticRegression(solver = 'liblinear')

kfold = StratifiedKFold(n\_splits=10, random\_state=1, shuffle=True)

# use of train\_test\_split

# Split-out training & validation dataset

X\_train, X\_validation, Y\_train, Y\_validation = train\_test\_split(X, Y, test\_size=0.20, random\_state=1)

#train the Model

cv\_results = cross\_val\_score(LR, X\_train, Y\_train, cv=kfold, scoring='accuracy')

print('%s: %f (%f)' % (LR, cv\_results.mean(), cv\_results.std()))

model = LR.fit(X\_train, Y\_train)

predictions = LR.predict(X\_validation)

print(predictions)

#Let's check the overall accuracy.

print(accuracy\_score(Y\_validation, predictions))