import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

import seaborn as sns

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

df.head()

df.info()

df.shape

df.describe()

X = df[['Age', 'EstimatedSalary']]

Y = df['Purchased']

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

from sklearn.preprocessing import StandardScaler

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size = 0.2, random\_state = 0)

sc\_X = StandardScaler()

X\_train = sc\_X.fit\_transform(X\_train)

X\_test = sc\_X.transform(X\_test)

print(f'Train Dataset Size - X: {X\_train.shape}, Y: {Y\_train.shape}')

print(f'Test Dataset Size - X: {X\_test.shape}, Y: {Y\_test.shape}')

from sklearn.linear\_model import LogisticRegression

lm = LogisticRegression()

lm.fit(X\_train, Y\_train)

predictions = lm.predict(X\_test)

sns.regplot(x = X\_test[:, 1], y = predictions, scatter\_kws={'s':5});

plt.scatter(X\_test[:, 1], Y\_test, marker = '+');

plt.xlabel("User's Estimated Salary");

plt.ylabel('Ads Purchased');

plt.title('Regression Line Tracing');

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import classification\_report

cm = confusion\_matrix(Y\_test, predictions)

print(f'''Confusion matrix :\n

| Positive Prediction\t| Negative Prediction

---------------+------------------------+----------------------

Positive Class | True Positive (TP) {cm[0, 0]}\t| False Negative (FN) {cm[0, 1]}

---------------+------------------------+----------------------

Negative Class | False Positive (FP) {cm[1, 0]}\t| True Negative (TN) {cm[1, 1]}\n\n''')

cm = confusion\_matrix(Y\_test, predictions)

TP = cm[0, 0]

TN = cm[1, 1]

FP = cm[1, 0]

FN = cm[0, 1]

print(f"Accuracy : {(TP+TN)/(TP+FP+TN+FN)} ")

print(f'Error Rate: {(FP+FN)/(TP+TN+FN+FP)}')

print(f"True positive rate : {TP/(TP+FN)}")

print(f'True negative rate (Recall):{TN/(FP+TN)}')

print(f'Precision (Positive predictive value) : {TP/(TP+FP)}')

print(f'False Positive Rate : {FP/(TN+FP)}')

# Visualizing the Training set results

from matplotlib.colors import ListedColormap

X\_set, y\_set = X\_train, Y\_train

X1, X2 = np.meshgrid(np.arange(start = X\_set[:, 0].min() - 1, stop = X\_set[:, 0].max() + 1, step = 0.01),

np.arange(start = X\_set[:, 1].min() - 1, stop = X\_set[:, 1].max() + 1, step = 0.01))

plt.figure(figsize=(9, 7.5));

plt.contourf(X1, X2, lm.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),alpha = 0.6, cmap = ListedColormap(('red', 'green')));

plt.xlim(X1.min(), X1.max());

plt.ylim(X2.min(), X2.max());

for i, j in enumerate(np.unique(y\_set)):

plt.scatter(X\_set[y\_set == j, 0], X\_set[y\_set == j, 1],color = ListedColormap(('red', 'green'))(i), label = j);

plt.title('Logistic Regression (Training set)');

plt.xlabel('Age');

plt.ylabel('Estimated Salary');

plt.show();