**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')

print("----------------Dataframe Info------------------")

print(df**.**info())

print("\n")

print("---------------Dataframe Descibe----------------")

print(df**.**describe())

print("\n")

print("---------------First 5 rows of Dataframe----------------")

print(df**.**head())

print("\n")

print("---------------Train Dataset-------------")

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 **=** 0.25, 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}')

print("----------------Linner Regression-------------------")

**from** sklearn.linear\_model **import** LogisticRegression

lm **=** LogisticRegression(random\_state**=**0, solver**=**'lbfgs')

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

predictions **=** lm**.**predict(X\_test)

plt**.**figure(figsize**=**(6, 6))

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')

plt**.**show()

print("-------------Confusion Matrix---------------")

**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 **=** classification\_report(Y\_test, predictions)

print('Classification report : \n', cm)

print("-----------------Visualizing Training set result----------------")

**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**.**legend()

plt**.**show()

print("\n")

print("-----------------Visualizing Test set result----------------")

**from** matplotlib.colors **import** ListedColormap

X\_set, y\_set **=** X\_test, Y\_test

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 (Test set)')

plt**.**xlabel('Age')

plt**.**ylabel('Estimated Salary')

plt**.**legend()

plt**.**show()

print("\n")