**Logistic Regression**

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

import seaborn as sns

from sklearn.metrics import classification\_report

from sklearn.metrics import confusion\_matrix,accuracy\_score

# Importing the dataset

dataset = pd.read\_csv('C:/Python\_3.8.0/Program/Social\_Network\_Ads.csv')

print(dataset.isnull().sum())

x = dataset.iloc[:, [2,3]].values

y = dataset.iloc[:, 4].values

print(x)

print(y)

# Splitting the dataset into the Training set and Test set

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

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size = 0.25, random\_state = 0)

k=x\_test

# Feature Scaling

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

x\_train = sc.fit\_transform(x\_train)

x\_test = sc.transform(x\_test)

# Fitting Logistic Regression to the Training set

from sklearn.linear\_model import LogisticRegression

classifier = LogisticRegression(random\_state = 0)

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

# Predicting the Test set results

y\_pred = classifier.predict(x\_test)

print(classification\_report(y\_test, y\_pred))

print(confusion\_matrix(y\_test, y\_pred))

print(accuracy\_score(y\_test,y\_pred))

sns.heatmap(pd.DataFrame(confusion\_matrix(y\_test,y\_pred)))

plt.show()

#-------------------------------------------------------

from matplotlib.colors import ListedColormap

x\_set, y\_set = x\_test, y\_test

aranged\_ages = np.arange(start = x\_set[:, 0].min(),

stop = x\_set[:, 0].max(), step = 0.01)

aranged\_salaries = np.arange(start = x\_set[:, 1].min(),

stop = x\_set[:, 1].max(), step = 0.01)

X1, X2 = np.meshgrid(aranged\_ages, aranged\_salaries)

plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(),

X2.ravel()]).T)

.reshape(X1.shape),

alpha = 0.5, cmap = ListedColormap(('orange', 'blue')))

#################################################################

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],

c = ListedColormap(('red', 'green'))(i), label = j)

plt.title('Logistic Regression (Test set)')

plt.xlabel('Age')

plt.ylabel('Salary')

plt.legend()

plt.show()