**SUPPORT VECTOR MACHINE (CLASSIFICATION)**

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

df = pd.read\_csv('titanic.csv')

df

df.drop("Name", axis=1, inplace=True)

df

df.columns

df = df[['Pclass', 'Sex', 'Age', 'Siblings/Spouses Aboard', 'Parents/Children Aboard', 'Fare','Survived']]

df

x = df.iloc[:, :-1].values

y = df.iloc[:, -1].values

print(x)

print(y)

encoding Sex column

from sklearn.compose import ColumnTransformer

from sklearn.preprocessing import OneHotEncoder

ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [1])], remainder= 'passthrough')

x = np.array(ct.fit\_transform(x))

print(x)

splitting into training 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=0.2, random\_state=0)

print(x\_train)

print(y\_train)

print(x\_test)

print(y\_test)

Training the model on SVM

from sklearn.svm import SVC

classifier = SVC(kernel='rbf', random\_state=0)

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

Making a random prediction

print(x\_train[100])

print(y\_train[100])

classifier.predict([[0, 1, 2, 29, 0, 0, 10.5]])

Predicting the test set

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

print(np.concatenate((y\_test.reshape(len(y\_test),1), y\_pred.reshape(len(y\_test),1)),1))

making the confusion matrix

from sklearn.metrics import confusion\_matrix

cm = confusion\_matrix(y\_test, y\_pred)

cm

making the heatmap

import seaborn as sn

plt.figure(figsize=(5,4))

sn.heatmap(cm, annot=True)

plt.xlabel('Predicted value')

plt.ylabel('Truth or Actual Value')