**Expt-4:**

**Program:**

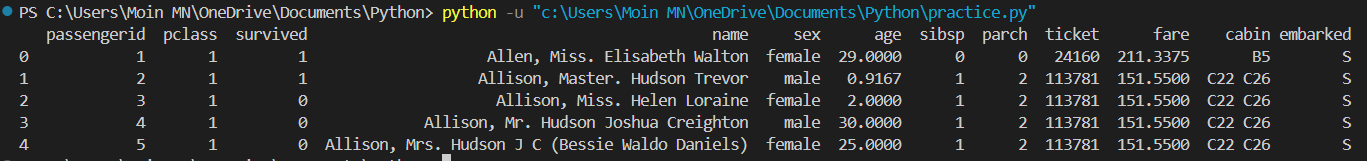
import pandas as pd

import matplotlib.pyplot as plt

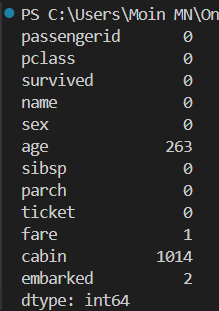
df = pd.read\_csv("./titanic.csv")

pd.read\_csv("./titanic.csv")

df.head()



df.isnull().sum()



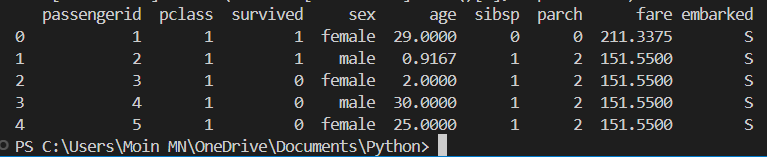
df['age'].fillna(value=df['age'].mean(), inplace=True)

df['fare'].fillna(value=df['fare'].mean(), inplace=True)

df['embarked'].fillna(value=df['embarked'].mode()[0], inplace=True)

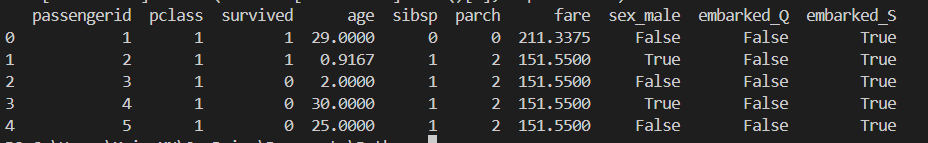
df.drop(labels = ['cabin', 'name', 'ticket'], axis= 1, inplace=True)

df.head()



df = pd.get\_dummies(df, columns=['sex', 'embarked'], drop\_first = True)

df.head()



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

X = df.drop('survived', axis=1)

y=df['survived']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

from sklearn.linear\_model import LogisticRegression

model = LogisticRegression (max\_iter=1000)

model.fit(X\_train, y\_train)



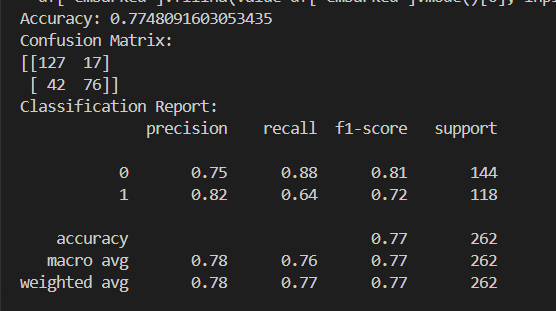
y\_pred = model.predict(X\_test)

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

print (f"Accuracy: {accuracy\_score (y\_test, y\_pred)}")

print (f"Confusion Matrix: \n{confusion\_matrix(y\_test, y\_pred)}")

print (f"Classification Report: \n{classification\_report (y\_test, y\_pred)}")



import tkinter as tk

from tkinter import ttk

from sklearn.metrics import classification\_report

import matplotlib.pyplot as plt

from matplotlib.backends.backend\_tkagg import FigureCanvasTkAgg

root = tk.Tk()

root.title("Logistic Regression Results By Moin MN 211P030")

report = classification\_report (y\_test, model.predict(X\_test), output\_dict=True)

report\_text = pd.DataFrame (report).transpose().round(2).to\_string()

text = tk. Text (root, height=10, width=60)

text.insert(tk. END, report\_text)

text.pack()

gender\_label = ttk. Label(root, text="Select Gender: ")

gender\_label.pack(pady=5)

gender = ttk.Combobox (root, values=["male", "female"])

gender.pack(pady=5)

pclass\_label = ttk. Label(root, text="Select Pclass:")

pclass\_label.pack(pady=5)

pclass = ttk.Combobox (root, values=[1, 2, 3])

pclass.pack(pady=5)

def show\_survivors():

  filtered = data[(data['sex'] == gender.get()) & (data['pclass'] == int(pclass.get())) & (data['survived'] == 1)]

  result\_text = f"Survivors: {len(filtered)}"

  result\_label.config(text=result\_text)

button = ttk.Button(root, text="Show Survivors", command=show\_survivors)

button.pack(pady=10)

result\_label = ttk. Label(root, text="")

result\_label.pack(pady=5)

root.mainloop()

