

**ANL252**

**Python for Data Analytics**

# **End-of-Course Assessment**

**July 2021 Presentation**

**Submitted by:**

|  |  |
| --- | --- |
| **Name** | **PI No.** |
| **Amir Hamzah Bin Abdul Khalid** | **B2110864** |

**Tutorial Group: ­­­­­­­­­­­­ T 09**

**Instructor’s Name: Dr. Munish Kumar**

**Submission Date: 13/09/2021**



File 1: Embedded JupyterLab File

1a)

i)

#read data and key in missing values as NaN

import pandas as pd

import numpy as np

ship = pd.read\_csv("ship.csv", na\_values = ".")

ship

ii)

#renaming of columns

ship.rename(columns={"T":"Types","A":"C\_Years","P":"O\_Periods","MS":"S\_Months","Y":"Incidents"}, inplace=True)

ship

iii)

#averaging the service months and incidents based on types and operating periods

shipgroup = ship.groupby(["Types", "O\_Periods"]).mean({"S\_Months":["Mean"], "Incidents":["Mean"]})

shipgroup.columns = ["C\_Years", "Average\_S\_Months", "Average\_Incidents"]

shipgroup = shipgroup.reset\_index()

print(shipgroup)

print (shipgroup.dtypes)

shipgroup["Average\_S\_Months"] = shipgroup["Average\_S\_Months"].round(0).astype(int)

shipgroup["Average\_Incidents"] = shipgroup["Average\_Incidents"].round(0).astype(int)

print(shipgroup)

iv)

#locating rows with missing values

print(ship[ship.isnull().any(axis=1)])

#replacing missing values with the specifed values calculated in iii

ship.iloc[6] = {'Types':1, 'C\_Years':4, 'O\_Periods':1, 'S\_Months':911, 'Incidents':3}

ship.iloc[14] = {'Types':2, 'C\_Years':4, 'O\_Periods':1, 'S\_Months':26852, 'Incidents':36}

ship.iloc[22] = {'Types':3, 'C\_Years':4, 'O\_Periods':1, 'S\_Months':914, 'Incidents':2}

ship.iloc[30] = {'Types':4, 'C\_Years':4, 'O\_Periods':1, 'S\_Months':296, 'Incidents':1}

ship.iloc[33] = {'Types':5, 'C\_Years':1, 'O\_Periods':2, 'S\_Months':1047, 'Incidents':7}

ship.iloc[38] = {'Types':5, 'C\_Years':4, 'O\_Periods':1, 'S\_Months':664, 'Incidents':7}

print(ship)

v)

# creating target variable Y which is the number of ship incident

Y = ship["Incidents"]

print(Y)

1b)

i)

#understanding the variable type in the dataframe

print (ship.dtypes)

#changing the variable in each column

ship["Types"] = ship["Types"].astype(str)

ship["C\_Years"] = ship["C\_Years"].astype(str)

ship["O\_Periods"] = ship["O\_Periods"].astype(str)

print (ship.dtypes)

print (ship)

ii)

#Converting categorical variables into dummy variables

X = pd.get\_dummies(ship[["Types", "C\_Years", "O\_Periods"]], drop\_first=False)

print(X)

X.to\_csv("X.csv", index=False)

iii)

#Performing Log-transformation on the S\_months column

ship["log\_S\_Months"] = np.log(ship["S\_Months"])

print(ship)

#Adding log\_S\_Months into X

log = ship["log\_S\_Months"]

print(log)

X = pd.read\_csv("X.csv")

X = pd.concat([X, log], axis=1, join="inner")

print(X)

#saving the file X

X.to\_csv("X.csv", index=False)

1c)

The data set contains how the training program will determine the type of ship, when it was constructed, its operating period, length of service and number the number of incidents the ship may be involved in. So, a user only has to input new ship data as its testing dataset and accurately determine the breakdown of their ship statistics in the testing set.

1d)

#saving the dataframe into a new file

ship\_prepared = ship

ship\_prepared.to\_csv("ship\_prepared.csv", index=False)

#putting ship\_prepared into SQL

import sqlite3

import pandas as pd

conn = sqlite3.connect("ship\_prepared.db")

cur = conn.cursor()

ship\_prepared = pd.read\_csv("ship\_prepared.csv")

ship\_prepared

ship\_prepared.to\_sql("ship\_prepared",conn, if\_exists="replace", index=False)

#Using cursor function

cur.execute ("SELECT \* FROM ship\_prepared;")

print(cur.fetchall());

2a)

The Poisson regression module allows us to calculate the number of times an event, in this case the number of incidents that will happen to a ship of a certain type, build, age and years of operation. The estimator to show the approximate point of the value from the array. Fit refers to the fit\_intercept whereby it specifies the bias or base constant to be added to the predicted linear function.

2b)

from sklearn import linear\_model

import matplotlib.pyplot as plt

import seaborn as seabornInstance

clf = linear\_model.PoissonRegressor(alpha=1,fit\_intercept=True,max\_iter=100,tol=1e-4,warm\_start=False,verbose=0)

X = X

Y = Y

print(clf.fit(X,Y, sample\_weight=None))

print(clf.get\_params(deep=True))

print(clf.predict(X))

print(clf.score(X,Y,sample\_weight=None))

print(clf.coef\_)

print(clf.intercept\_)

x = np.array(X["log\_S\_Months"])

y = np.array(Y)

plt.plot(x,y,'o')

m,b = np.polyfit(x,y,1)

plt.plot(x, m\*x + b)

plt.title("Occurence of Incidents vs Length of Service")

plt.xlabel("No. of Incidents")

plt.ylabel("log\_S\_Months")

Chart, scatter chart

Description automatically generated

Figure 1-1 :Poisson Regression Graph of Occurrence of Incidents vs Length of Service

2c)