Logo, company name

Description automatically generated

**ANL252**

**Python for Data Analytics**

**End-of-Course Assessment – July Semester 2021**

**Oon Ee Hai**

**PI No. Q2011729**

**Question 1a(i)**

**Python Program code**

# Importing the libraries for using NumPy and pandas

import numpy as np

import pandas as pd

# Use read\_csv to read csv file,indicate "." as missing values

ship = pd.read\_csv('C:\ship.csv', na\_values='.')

# ship dataframe

ship

**Screenshot of code**

Graphical user interface, text

Description automatically generated

**Screenshot of output**

Table

Description automatically generated with medium confidence

**Question 1a(ii)**

**Python Program code**

# Use rename to rename the coulums

ship = ship.rename(columns={"T": "types", "A": "c\_years", "P": "o\_periods", "MS": "s\_months", "Y": "incidents"})

# Renamed coLumns OF ship dataframe

ship

**Screenshot of code**



**Screenshot of output**Table

Description automatically generated

**Question 1a(iii)**

**Python Program code**

# Get average service months and the average number of incidents using mean & round to nearest integer

shipgroup = ship.groupby(['types', 'o\_periods'])[['s\_months', 'incidents']].mean().round()

# Grouped dataframe with required averages

shipgroup

**Screenshot of code**

A picture containing company name

Description automatically generated

**Screenshot of output**Table

Description automatically generated

**Question 1a(iv)**

**Python Program code**

# First we groupby the dataframe with 'types','o\_periods' columns

# Select 's\_months' or 'incidents' from the grouped dataframe to replace the null values have in 's\_months' or 'incidents'

# Using tranform function, get the mean for each row for respective 'types','o\_periods'

# Rounding to the nearest integer

ship['s\_months']= ship['s\_months'].fillna(ship.groupby(['types','o\_periods'])['s\_months'].transform('mean').round())

ship['incidents']= ship['incidents'].fillna(ship.groupby(['types','o\_periods'])['incidents'].transform('mean').round())

ship

**Screenshot of code**

Text

Description automatically generated

**Screenshot of output**Table

Description automatically generated

**Question 1a(v)**

**Python Program code**

# Select incidents column from ship dataframe and convert it to dataframe

Y = pd.DataFrame(ship['incidents'])

# Target variable Y

Y

**Screenshot of code**

A picture containing text

Description automatically generated

**Question 1b(i)**

**Python Program code**

# Convert datatype to categorical variables)

ship = ship.astype({'types': 'category','c\_years':'category','o\_periods':'category'})

ship.dtypes

**Screenshot of code**



**Screenshot of output**Text

Description automatically generated with low confidence

**Question 1b(ii)**

**Python Program code**

# Convert categorical variables into dummy variables

X = ship.copy()

X = pd.get\_dummies(X,columns = ['types', 'c\_years','o\_periods'])

X

**Screenshot of code**

A picture containing diagram

Description automatically generated

**Screenshot of output**

Table

Description automatically generated

**Question 1b(iii)**

**Python Program code**

# Apply log-transformation & round to 2 dp

log\_s\_months = round(np.log(ship['s\_months']),2)

# Adding 'log\_s\_months' column in both the dataframe "X"and "ship"

ship['log\_s\_months'] = log\_s\_months

X['log\_s\_months'] = log\_s\_months

ship

X

**Screenshot of code**

Text

Description automatically generated

**Screenshot of output  
  
ship output**Table

Description automatically generated

**X output**

Table

Description automatically generated

**Question 1c**

The train-test technique is not fitting as the Dataset is fairly small with only 39 samples. When dataset is spilt into train and test sets, there won't be enough data in the training dataset for the model to become familiar with a viable planning of contributions to yields. Likewise for the test set, there will not be enough date to determine the model presentation. The result will not be ideal if we were to spilt it. k-overlay cross-approval methodology would be an better substitute model assessment system if there is insufficient data.

**Question 1d**

**Python Program code**

# Saving dataframe to new .csv file

ship.to\_csv('ship\_prepared.csv')

from sqlalchemy import create\_engine

# create engine & datbase ship.db

engine = create\_engine('sqlite:///ship.db')

# Write records of DataFrame to a SQL database & Table name is ship\_table

ship.to\_sql('ship\_table', engine, if\_exists='append')

**Screenshot of code**

Graphical user interface, text

Description automatically generated

**Question 2a**

sklearn.linear\_model.PoissonRegressor is the scikit-learn module for Poisson regression. The fit() method parameters takes the training data as arguments and fit a generalized linear model which Trains or Fit a Poisson Regressor..The preditct()method predict Predict with Poisson Regressor using with input matrix X.

**Question 2b**

**Python Program code**

#Converting to numpy array

import numpy as np

X=X.to\_numpy()

Y=Y.to\_numpy()

from sklearn import linear\_model

model = linear\_model.PoissonRegressor()

model.fit(X, Y)

**Screenshot of code**

Text

Description automatically generated

**Question 2c**

**Python Program code**

Y\_pred=model.predict(X)

for y in Y:

if(y==0):

D.append(np.sign(y-np.exp(np.mean(Y\_pred))\*np.sqrt(-(y-np.exp(np.mean(Y\_pred))))))

else:

D.append(np.sign(y-np.exp(np.mean(Y\_pred))\*np.sqrt(2\*y\*np.log(y/np.exp(np.mean(Y\_pred)))-(y-np.exp(np.mean(Y\_pred))))))

print("Deviance is ",np.mean(D))

**Screenshot of code**

Text

Description automatically generated