Logo

Description automatically generated

**ANL252**

**Python for Data Analytics**

**End of Course Assessment**

**January 2023 Presentation**

Name: Prabu S/O Ramasamy

T Group: T05

PI Number: Q2110662

Q1)

The following variables have missing data

1. Claim\_ID
2. Actual
3. Terms

Q2)

I have treated the missing data by using fillna and replacing the na values with blanks. I chose this method, instead of dropna, as there is a substantial number of rows in Actual that contain missing data, (1677). So removing those rows, might affect the results of the analysis, as the data population will be reduced.

Q3)

Three data prep that I undertook is as follows:-

i) I ensure that all the dates are in a standard format, so that any analysis using the dates will have increased reliability, versus when the dates are all in different formats. This was done by installing dateutil, and importing datetime.

ii) Sorted all the data by the created date, using the sort\_values function

iii) Created a new column named “Processing\_Time”, to find the difference between the created date and actual paid date.

Q4)

i)Pie Chart showing the number of Claims by the Region and the Payment Status.

We can see from here that the majority of the claims are from the LOC region, as opposed to the FVS region. We can assume that majority of the clients are from that region, or It can also point towards the population density of LOC region

Chart, pie chart

Description automatically generated

ii)

Chart, bar chart

Description automatically generated

This Bar Chart shows the amount of claim paid by the Type. As we can see O001 type has the highest claim

iii) Attempted to do bar chart which shows Processing Time by Type,

Chart

Description automatically generated

Question 5

#Question 1

##importing the required libaries

pip install python-dateutil

import sklearn as skl

import pandas as pd

import numpy as np

from datetime import datetime as dt

import matplotlib as mat

import seaborn as seabornInstance

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

from sklearn.linear\_model import LinearRegression

from sklearn import metrics

%matplotlib inline

#Assigning the source file to a variable called df, and defining null values and reading the data

df = pd.read\_csv("ECA.csv",na\_values = ["Unkn" , "???"])

print(df.head())

#finding the list of variables with missing values

df.isnull().sum(0)

#Question 2

#null value treatment, since there is substantial amount of missing data, I will replace missing data with 0

df1 = df.fillna("")

df1

#checking if there are still na values in the file

df1.isnull()

#to confirm that there is no missing data in the dataset

df1.isna().sum()

# Question3

#Data Transformation

#1) To ensure that all dates are a standard format, and either a float, string or any other readable value for visualization

#to see the data type of each column:

df1.dtypes

df1[['Planned','Actual','Created']].head(1)

#From above we can see that Planned is formatted in dd/m/yyyy, Actual is formatted as dd/mm/yyyy hh:mm,

#and created is formatted as yyyymmdd

#Standardising the date formats

df1['Planned'] = pd.to\_datetime(df1['Planned'],format ='%d/%m/%Y')

print(df1['Planned'].head(1))

df1['Actual'] = pd.to\_datetime(df1['Actual'],)

print(df1['Actual'].head(5))

df1['Created'] = pd.to\_datetime(df1['Created'],format = "%Y%m%d")

print(df1['Created'].head(5))

print(df1.head(5))

#Sorting data based on created date

df1.sort\_values(by=["Created"])

#find difference between created date and actual date (processing time)

df1["Processing\_Time"] = df1["Actual"] - df1["Created"]

df1

#Question 4

# A Pie chart showing number of claims by region, and payment status

import matplotlib.pyplot as plt

df1[["Region","Paid"]].value\_counts(normalize=True).plot.pie(autopct='%.1f %%')

#bar chart showing amount paid for different types of Claims

plt.barh(df1["Type"],df1["Amount"])

plt.xlabel("Type")

plt.ylabel("Amount")

plt.title("Amount Paid by Type of Claim")

#Bar Chart Showing Processing Time based on Type

plt.bar(df1["Type"],["Processing Time"])

plt.xlabel("Type")

plt.ylabel("Processing Time in days")

#question 5

#Getting the number of days there were processing delay

df1["Processing\_Delay"] = df1["Actual"] - df1["Planned"]

print(df1["Processing\_Delay"].head())

#changing the days delayed into integer

a=df1["Processing\_Delay"].astype(int)

print(a.head())

#attempt to plot the scatter graph for linear regression model

x = a

y = df1["Actual"].astype(int)

x,yy = df1["Actual"].astype(int)

x,y

df1.plot(x=x,y=y, style='o')

plt.title('Planned v/s Actual')

plt.xlabel('Planned')

plt.ylabel('Actual')

plt.show()

References:

pythonpython 4, & Anand S KumarAnand S Kumar 86.9k1818 gold badges183183 silver badges172172 bronze badges. (1962, September 1). *Normalize different date data into single format in Python*. Stack Overflow. Retrieved March 5, 2023, from <https://stackoverflow.com/questions/33051147/normalize-different-date-data-into-single-format-in-python>

Samuel, N. (2023, January 4). How to format dates in Python. Stack Abuse. Retrieved March 5, 2023, from <https://stackabuse.com/how-to-format-dates-in-python/>

GeeksforGeeks. (2020, October 5). *Convert the column type from string to datetime format in pandas dataframe*. GeeksforGeeks. Retrieved March 6, 2023, from <https://www.geeksforgeeks.org/convert-the-column-type-from-string-to-datetime-format-in-pandas-dataframe/>

*Drop duplicate rows in a DataFrame*. James LeDoux's Blog. (2019, June 1). Retrieved March 6, 2023, from <https://jamesrledoux.com/code/drop_duplicates#:~:text=To%20remove%20duplicates%20of%20only,equals%20either%20first%20or%20last%20>.

<https://stackoverflow.com/questions/66108871/how-to-create-a-pie-chart-from-a-text-column-data-only-using-pandas-dataframe-di>

<https://www.kdnuggets.com/2019/03/beginners-guide-linear-regression-python-scikit-learn.html>