Name: Prakriti Khadka

Student Id : 2461868

To - Do - Task

Please Complete all the problem listed below.

3.1 Warm Up Exercises:

1. Sorting and Subsetting:

Complete all following Task:

• Dataset for the Task: "titanic.csv"

Following task is common for all the problem:

1. Load the provided dataset and import in pandas DataFrame.

2. Check info of the DataFrame and identify following:

Problem 1 - Sorting:

1. Create a DataFrame called fare that contains only the Fare column of the Titanic dataset. Print the head of the result.

2. Create a DataFrame called class age that contains only the Pclass and Age columns of the Titanic

dataset, in that order. Print the head of the result.

3. Create a DataFrame called survived gender that contains the Survived and Sex columns of the

Titanic dataset, in that order.Print the head of the result.

import pandas as pd

df = pd.read\_csv('/content/drive/MyDrive/data/Titanic-Dataset.csv')

data\_info = df.info()

print(data\_info)

fare = df[['Fare']]

print("\nHead of Fare DataFrame:")

print(fare.head())

class\_age = df[['Pclass', 'Age']]

print("\nHead of Class Age DataFrame:")

print(class\_age.head())

survived\_gender = df[['Survived', 'Sex']]

print("\nHead of Survived Gender DataFrame:")

print(survived\_gender.head())

Problem - 2 - Subsetting:

Complete all the following Task:

Subsetting Rows:

1. Filter the Titanic dataset for cases where the passenger’s fare is greater than 100, assigning it

to fare gt 100.View the printed result.

2. Filter the Titanic dataset for cases where the passenger’s class (Pclass) is 1, assigning it to

first class.View the printed result.

3. Filter the Titanic dataset for cases where the passenger’s age is less than 18 and the passenger

is female (Sex is "female"), assigning it to female under 18. View the printed result.

fare\_gt\_100 = df[df['Fare'] > 100]

print("\nPassengers with Fare > 100:")

print(fare\_gt\_100)

(Pclass) is 1

first\_class = df[df['Pclass'] == 1]

print("\nPassengers in First Class:")

print(first\_class)

female\_under\_18 = df[(df['Age'] < 18) & (df['Sex'] == 'female')]

print("\nFemale Passengers Under 18:")

print(female\_under\_18)

Subsetting Rows by Categorical variables:

1. Filter the Titanic dataset for passengers whose Embarked port is either "C" (Cherbourg) or "S"

(Southampton), assigning the result to embarked c or s. View the printed result.

2. Filter the Titanic dataset for passengers whose Pclass is in the list [1, 2] (indicating first

or second class), assigning the result to first second class.View the printed result.embarked\_c\_or\_s = df[df['Embarked'].isin(['C', 'S'])]

print("\nPassengers Embarked at C or S:")

print(embarked\_c\_or\_s)

first\_second\_class = df[df['Pclass'].isin([1, 2])]

print("\nPassengers in First or Second Class:")

print(first\_second\_class)

3.2 Exploratory Data Analysis Practice Exercise - 1.

Warning: Handle missing values in the Age column by filling them with the median age of

the dataset before performing the division.)

Answer the following questions from Dataset:

Which passenger had the highest fare paid relative to their age?

To answer the question perform following operations:

1. Add a column to the Titanic dataset, fare per year, containing the fare divided by the age of

the passenger(i.e., Fare/Age).

2. Subset rows where fare per year is higher than 5, assigning this to high fare age.

3. Sort high fare age by descending fare per year, assigning this to high fare age srt.

4. Select only the Name and fare per year columns of high fare age srt and save the result as result.

5. Look at the result.

df['Age'].fillna(df['Age'].median(), inplace=True)

data['fare\_per\_year'] = data['Fare'] / data['Age']

high\_fare\_age = data[data['fare\_per\_year'] > 5]

high\_fare\_age\_srt = high\_fare\_age.sort\_values(by='fare\_per\_year', ascending=False)

result = high\_fare\_age\_srt[['Name', 'fare\_per\_year']]

print("\nPassengers with Highest Fare Paid Relative to Age:")

print(result)

Which adult male passenger (age ≥ 18 and Sex is ’male’) paid the highest fare relative to their

class?

To answer the question perform following operations:

1. Add a column to the Titanic dataset, fare per class, containing the fare divided by the passenger

class i.e. Fare / Pclass.

2. Subset rows where the passenger is male (Sex is ”male”) and an adult (Age is greater than or equal

to 18), assigning this to adult males.

3. Sort adult males by descending fare per class, assigning this to adult males srt.

4. Select only the Name, Age, and fare per class columns of adult males sr and save the result

as result.

5. Look at the result.

df['fare\_per\_class'] = df['Fare'] / df['Pclass']

adult\_males = df[(df['Sex'] == 'male') & (df['Age'] >= 18)]

adult\_males\_srt = adult\_males.sort\_values(by='fare\_per\_class', ascending=False)

result\_male\_class = adult\_males\_srt[['Name', 'Age', 'fare\_per\_class']]

print("\nAdult Male Passengers with Highest Fare Paid Relative to Class:")

print(result\_male\_class)

3.3 Exploratory Data Analysis with Group-by Method Practice Exercise:

Based on the dataset Answer the following question:

What percent of the total fare revenue came from each passenger class?

To answer the question perform following operation:

1. Calculate the total Fare paid across all passengers in the Titanic dataset.

2. Subset for passengers in first class (Pclass is 1) and calculate their total fare.

3. Do the same for second class (Pclass is 2) and third class (Pclass is 3).

4. Combine the fare totals from first, second, and third classes into a list.

5. Divide the totals for each class by the overall total fare to get the proportion of fare revenue by class.

total\_fare = data['Fare'].sum()

first\_class\_fare = data[data['Pclass'] == 1]['Fare'].sum()

second\_class\_fare = data[data['Pclass'] == 2]['Fare'].sum()

third\_class\_fare = data[data['Pclass'] == 3]['Fare'].sum()

fare\_totals = [first\_class\_fare, second\_class\_fare, third\_class\_fare]

fare\_proportions = [fare / total\_fare for fare in fare\_totals]

print("\nTotal Fare Revenue by Class as Proportions:")

for idx, proportion in enumerate(fare\_proportions, start=1):

    print(f"Class {idx}: {proportion:.2%}")

Based on the dataset Answer the following question:

What percent of the total number of passengers on the Titanic belonged to each age group

(e.g., child, adult, senior)?

To answer the question perform following operation:

1. Create a new column, age group, that categorizes passengers into "child" (age < 18), "adult" (age

18{64), and "senior" (age 65 and above).

2. Calculate the total number of passengers on the Titanic.

3. Count the number of passengers in each age group.

4. Divide the count of each age group by the total number of passengers to get the proportion of passengers

in each age group.

5. Display the proportion as a percentage.

    conditions = [

    (df['Age'] < 18),

    (df['Age'] >= 18) & (df['Age'] < 65),

    (df['Age'] >= 65)

]

age\_groups = ['child', 'adult', 'senior']

df['age\_group'] = pd.cut(df['Age'], bins=[0, 18, 65, float('inf')], labels=age\_groups, right=False)

total\_passengers = len(df)

age\_group\_counts = df['age\_group'].value\_counts()

age\_group\_proportions = (age\_group\_counts / total\_passengers) \* 100

print("\nPercentage of Total Passengers by Age Group:")

for group, proportion in age\_group\_proportions.items():

    print(f"{group}: {proportion:.2f}%")