3 To - Do - Task

Please Complete all the problem listed below.

3.1 Warming Up Exercises - Basic Inspection and Exploration:

Problem 1 - Data Read, Write and Inspect:

Complete all following Task:

- Dataset for the Task: "bank.csv"

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

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

df = pd.read_csv("/content/drive/MyDrive/Concept and technology of AI/bank.csv")
```

2. Check info of the DataFrame and identify following:

   (a) columns with dtypes=object

   (b) unique values of those columns.

   (c) check for the total number of null values in each column.


```
column = df.select_dtypes(include='object').columns
column
```

```
Index(['job', 'marital', 'education', 'default', 'housing', 'loan', 'contact',
       'month', 'poutcome', 'y'],
      dtype='object')
```

3. Drop all the columns with dtypes object and store in new DataFrame, also write the DataFrame in ".csv" with name "banknumericdata.csv"

```
for x in column:
    print(df[x].unique())
```

```
['management' 'technician' 'entrepreneur' 'blue-collar' 'unknown'
 'retired' 'admin.' 'services' 'self-employed' 'unemployed' 'housemaid'
 'student']
['married' 'single' 'divorced']
['tertiary' 'secondary' 'unknown' 'primary']
['no' 'yes']
['yes' 'no']
['no' 'yes']
['unknown' 'cellular' 'telephone']
['may' 'jun' 'jul' 'aug' 'oct' 'nov' 'dec' 'jan' 'feb' 'mar' 'apr' 'sep']
['unknown' 'failure' 'other' 'success']
['no' 'yes']
```

4. Read ”banknumericdata.csv” and Find the summary statistics.

```
km = df.copy()
df_numeric = km.drop(columns=column)
df_numeric.to_csv("banknumericdata.csv", index=False)
df_numeric.head()
```

	age	balance	day	duration	campaign	pdays	previous
0	58	2143	5	261	1	-1	0
1	44	29	5	151	1	-1	0
2	33	2	5	76	1	-1	0
3	47	1506	5	92	1	-1	0
4	33	1	5	198	1	-1	0

```
bnk = pd.read_csv("/content/drive/MyDrive/Concept and technology of AI/bank.csv");
bnk.describe()
```

	age	balance	day	duration	campaign	pd
count	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000
mean	40.936210	1362.272058	15.806419	258.163080	2.763841	40.197
std	10.618762	3044.765829	8.322476	257.527812	3.098021	100.128
min	18.000000	-8019.000000	1.000000	0.000000	1.000000	-1.000
25%	33.000000	72.000000	8.000000	103.000000	1.000000	-1.000
50%	39.000000	448.000000	16.000000	180.000000	2.000000	-1.000
75%	48.000000	1428.000000	21.000000	319.000000	3.000000	-1.000
max	95.000000	102127.000000	31.000000	4918.000000	63.000000	871.000

Problem 2 - Data Imputations:

Complete all the following Task:

- Dataset for the Task: "medical_student.csv"

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

```
med = pd.read_csv("/content/drive/MyDrive/Concept and technology of AI/ medical_st
med.head()
```

	Student ID	Age	Gender	Height	Weight	Blood Type	BMI	Temperature	Heart Rate
0	1.0	18.0	Female	161.777924	72.354947	O	27.645835	NaN	95.0
1	2.0	NaN	Male	152.069157	47.630941	B	NaN	98.714977	93.0
2	3.0	32.0	Female	182.537664	55.741083	A	16.729017	98.260293	76.0
3	NaN	30.0	Male	182.112867	63.332207	B	19.096042	98.839605	99.0
4	5.0	23.0	Female	NaN	46.234173	O	NaN	98.480008	95.0

2. Check info of the DataFrame and identify column with missing (null) values.

```
med.isnull().sum()
```

	0
Student ID	20000
Age	20000
Gender	20000
Height	20000
Weight	20000
Blood Type	20000
BMI	20000
Temperature	20000
Heart Rate	20000
Blood Pressure	20000
Cholesterol	20000
Diabetes	20000
Smoking	20000

dtype: int64

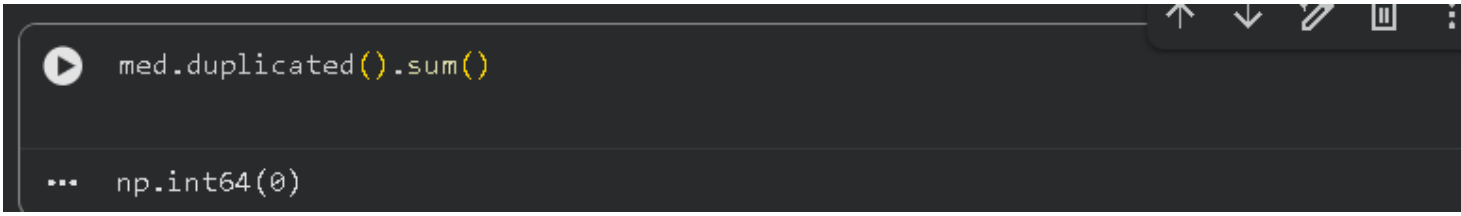
3. For the column with missing values fill the values using various techniques we discussed above. Try to explain why did you select the particular methods for particular column.

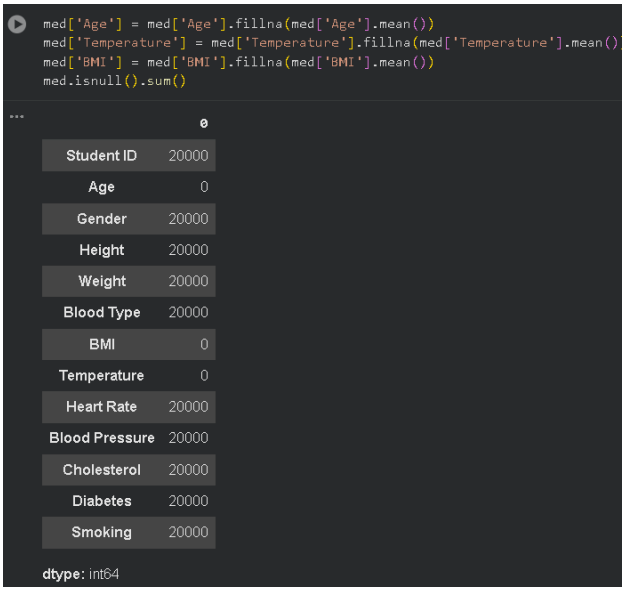
```
med.describe()
```

	Student ID	Age	Height	Weight	BMI	Tem
count	180000.000000	180000.000000	180000.000000	180000.000000	180000.000000	1800
mean	49974.042078	26.021561	174.947103	69.971585	23.338869	
std	28879.641657	4.890528	14.447560	17.322574	7.033554	
min	1.000000	18.000000	150.000041	40.000578	10.074837	
25%	24971.750000	22.000000	162.476110	54.969838	17.858396	
50%	49943.500000	26.000000	174.899914	69.979384	22.671401	
75%	74986.000000	30.000000	187.464417	84.980097	27.997487	
max	100000.000000	34.000000	199.998639	99.999907	44.355113	1

I see the data for age,temperature and BMI is mostly normal. The standard deviation is low and he mean and median is close to eachother. And the quartiles are evenly spaced.So I'll fill the age,temperature and BMI with the vlaue of mean.

4. Check for any duplicate values present in Dataset and do necessary to manage the duplicate items. {Hint: dataset.duplicated.sum()}





I see the data for height,weight,heart rate,blood pressure and cholestrol is skewed. The standard deviation is fairly high. And the quartiles aren't evenly spaced. So I'll fill the age,temperature and BMI with the vlaue of median. Also the diference between in minimum and maximun values are high which means there are outliners in the data.

3.2 Exercises - Data Cleaning and Transformations with "Titanic Dataset":

Dataset Used: "titanic.csv"

Problem - 1:

Create a DataFrame that is subsetted for the columns 'Name', 'Pclass', 'Sex', 'Age', 'Fare', and 'Survived'. Retain only those rows where 'Pclass' is equal to 1, representing first-class passengers. What is the mean, median, maximum value, and minimum value of the 'Fare' column?

```
tic = pd.read_csv("/content/drive/MyDrive/Concept and technology of AI/Titanic-Dat
tic.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	F
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1
4	5	0	3	Allen, Mr. William	male	35.0	0	0	373450	8.0

```
subset = tic[['Name','Pclass','Sex','Age','Fare','Survived']]
first_class = subset[subset['Pclass'] == 1]
print(first_class['Fare'].describe())
```

```
count    216.000000
mean      84.154687
std       78.380373
min        0.000000
25%       30.923950
50%       60.287500
75%       93.500000
max      512.329200
Name: Fare, dtype: float64
```

Problem - 2:

How many null values are contained in the 'Age' column in your subsetted DataFrame? Once you've found this out, drop them from your DataFrame.

```
tic1 = tic.copy()
tic1 = tic1.dropna(subset=['Age'])
tic1.isnull().sum()
```

```
...
                0
PassengerId     0
Survived        0
Pclass          0
Name            0
Sex             0
Age             0
SibSp           0
Parch           0
Ticket          0
Fare            0
Cabin         529
Embarked        2

dtype: int64
```

Problem - 3:

The 'Embarked' column in the Titanic dataset contains categorical data representing the ports of embarkation:

- 'C' for Cherbourg
- 'Q' for Queenstown
- 'S' for Southampton

Task:

1. Use one-hot encoding to convert the 'Embarked' column into separate binary columns ('Embarked C', 'Embarked Q', 'Embarked S').
2. Add these new columns to the original DataFrame.
3. Drop the original 'Embarked' column.
4. Print the first few rows of the modified DataFrame to verify the changes.

```
tic2 = tic.copy()
tic2.head()

tic2["C"] = np.where(tic["Embarked"] == "C", 1, 0)
tic2["Q"] = np.where(tic["Embarked"] == "Q", 1, 0)
tic2["S"] = np.where(tic["Embarked"] == "S", 1, 0)
tic2.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	C	Q	S
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S	0	0	1
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C	1	0	0
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S	0	0	1
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S	0	0	1
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S	0	0	1

```
tic2.drop(columns = ["Embarked"])
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	C	Q	S
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	0	0	1
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	1	0	0
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	0	0	1
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	0	0	1
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	0	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	0	0	1
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	0	0	1
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	0	0	1
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	1	0	0
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	0	1	0

891 rows × 14 columns

Problem - 4:

Compare the mean survival rates ('Survived') for the different groups in the 'Sex' column. Draw a visualization to show how the survival distributions vary by gender.

```
# Number of records
print(len(tic2))
# No.of survived people
print(tic2["Survived"].sum())

mean = tic2["Survived"].sum() / len(tic2)
print(mean)
```

```
891
342
0.3838383838383838
```

```
tic2["Age"].describe()
```

	Age
count	714.000000
mean	29.699118
std	14.526497
min	0.420000
25%	20.125000
50%	28.000000
75%	38.000000
max	80.000000

dtype: float64

```
name = ["'C' (Cherbourg)","'Q' (Queenstown)","'S' (Southampton)"]
value = [tic2["C"].sum(),tic2["Q"].sum(),tic2["S"].sum()]

plt.bar(name,value)
plt.xlabel("Port of Embark")
plt.ylabel("Number of passengers")
plt.title("Visualization")
plt.show()
```