EXPERIMENT – 3(A)

PANDAS LIBRARY – DATA PREPROCESSING

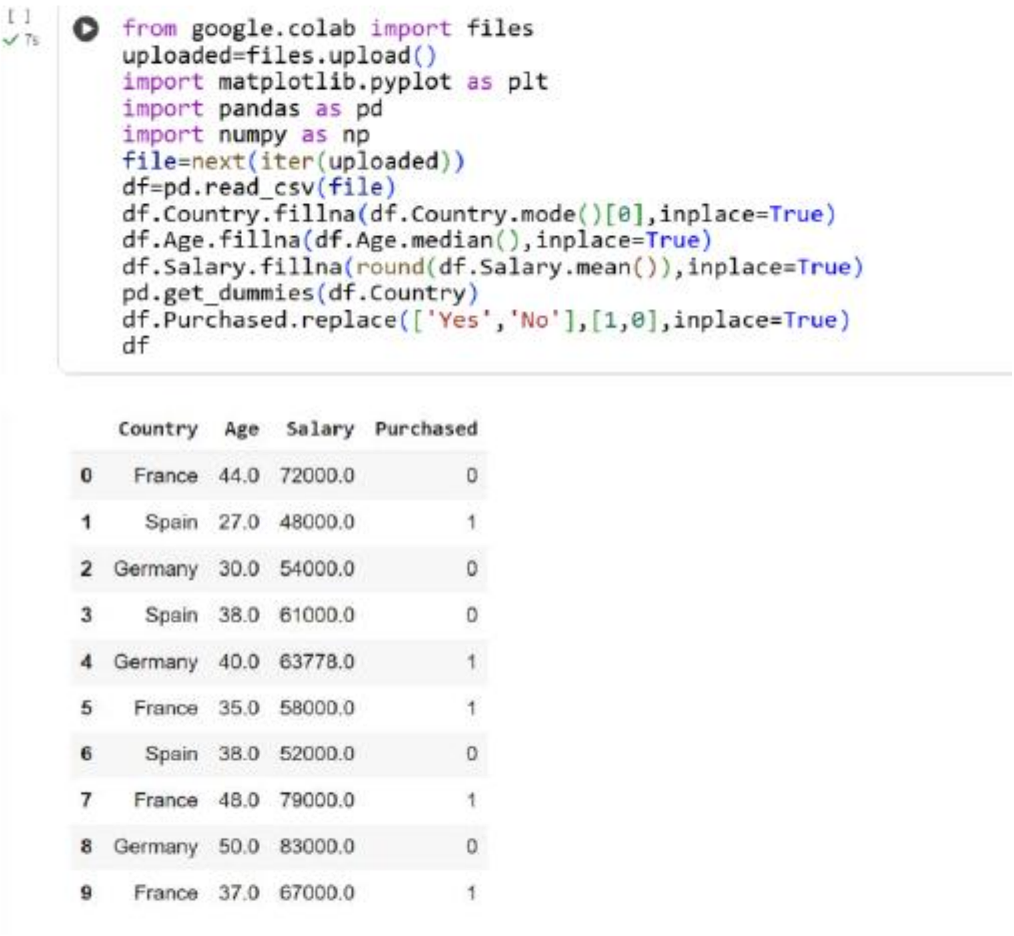
Aim:

To understand the importance of data preprocessing in data science

Procedure:

- Upload the given dataset(csv file) and read it
- Import necessities such as numpy ,pandas
- Now , process the data , find missing values and replace it with mean(),mode(),median()

Program:



```
from google.colab import files
uploaded=files.upload()
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
file=next(iter(uploaded))
df=pd.read_csv(file)
df.Country.fillna(df.Country.mode()[0],inplace=True)
df.Age.fillna(df.Age.median(),inplace=True)
df.Salary.fillna(round(df.Salary.mean()),inplace=True)
pd.get_dummies(df.Country)
df.Purchased.replace(['Yes','No'],[1,0],inplace=True)
df
```

	Country	Age	Salary	Purchased
0	France	44.0	72000.0	0
1	Spain	27.0	48000.0	1
2	Germany	30.0	54000.0	0
3	Spain	38.0	61000.0	0
4	Germany	40.0	63778.0	1
5	France	35.0	58000.0	1
6	Spain	38.0	52000.0	0
7	France	48.0	79000.0	1
8	Germany	50.0	83000.0	0
9	France	37.0	67000.0	1

Result:

Thus the python program to find missed values and replacing it was executed and verified

EXPERIMENT – 3(B)

PANDAS LIBRARY – HANDLING MISSING VALUES

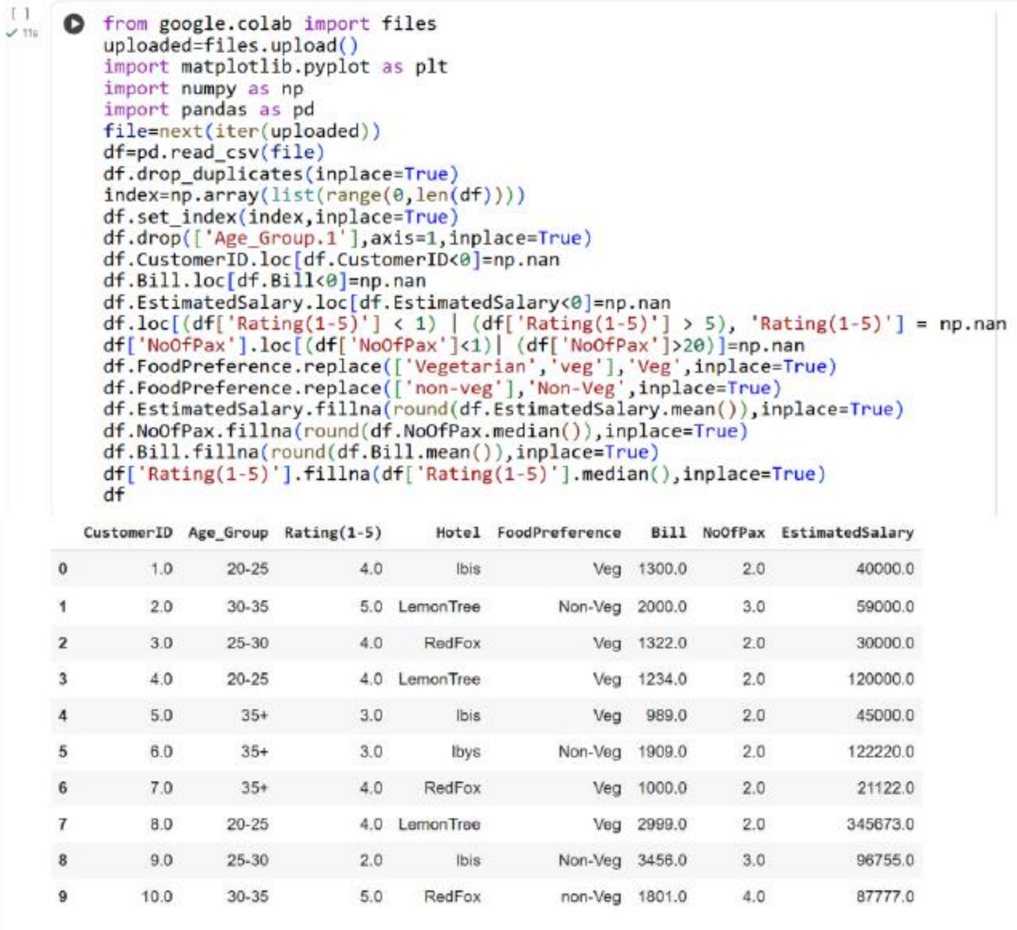
Aim:

To handle and analyze missing and inappropriate data in dataset

Procedure:

- Upload the csv file and read it
- Import necessities such as pandas , numpy
- Now check for missing and inappropriate data to replace with appropriate data

Program:



```
from google.colab import files
uploaded=files.upload()
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
file=next(iter(uploaded))
df=pd.read_csv(file)
df.drop_duplicates(inplace=True)
index=np.array(list(range(0,len(df))))
df.set_index(index,inplace=True)
df.drop(['Age_Group.1'],axis=1,inplace=True)
df.CustomerID.loc[df.CustomerID<0]=np.nan
df.Bill.loc[df.Bill<0]=np.nan
df.EstimatedSalary.loc[df.EstimatedSalary<0]=np.nan
df.loc[(df['Rating(1-5)'] < 1) | (df['Rating(1-5)'] > 5), 'Rating(1-5)'] = np.nan
df['NoOfPax'].loc[(df['NoOfPax']<1)| (df['NoOfPax']>20)]=np.nan
df.FoodPreference.replace(['Vegetarian','veg'],'Veg',inplace=True)
df.FoodPreference.replace(['non-veg'],'Non-Veg',inplace=True)
df.EstimatedSalary.fillna(round(df.EstimatedSalary.mean()),inplace=True)
df.NoOfPax.fillna(round(df.NoOfPax.median()),inplace=True)
df.Bill.fillna(round(df.Bill.mean()),inplace=True)
df['Rating(1-5)'].fillna(df['Rating(1-5)'].median(),inplace=True)
df
```

	CustomerID	Age_Group	Rating(1-5)	Hotel	FoodPreference	Bill	NoOfPax	EstimatedSalary
0	1.0	20-25	4.0	Ibis	Veg	1300.0	2.0	40000.0
1	2.0	30-35	5.0	LemonTree	Non-Veg	2000.0	3.0	59000.0
2	3.0	25-30	4.0	RedFox	Veg	1322.0	2.0	30000.0
3	4.0	20-25	4.0	LemonTree	Veg	1234.0	2.0	120000.0
4	5.0	35+	3.0	Ibis	Veg	989.0	2.0	45000.0
5	6.0	35+	3.0	Ibys	Non-Veg	1909.0	2.0	122220.0
6	7.0	35+	4.0	RedFox	Veg	1000.0	2.0	21122.0
7	8.0	20-25	4.0	LemonTree	Veg	2999.0	2.0	345673.0
8	9.0	25-30	2.0	Ibis	Non-Veg	3456.0	3.0	96755.0
9	10.0	30-35	5.0	RedFox	non-Veg	1801.0	4.0	87777.0

Result:

Thus we find missing and inappropriate data and replaced with appropriate ones

EXPERIMENT-3(C)

PANDAS LIBRARY – CREATE CSV FILE

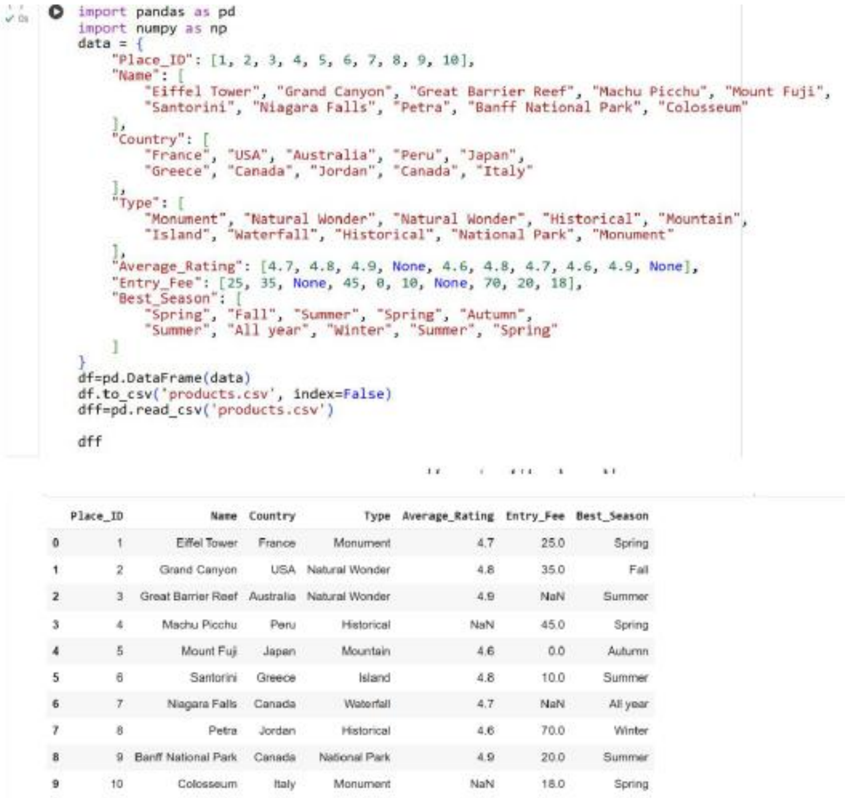
Aim:

To create a dataset and make it as csv file and handle inappropriate values

Procedure:

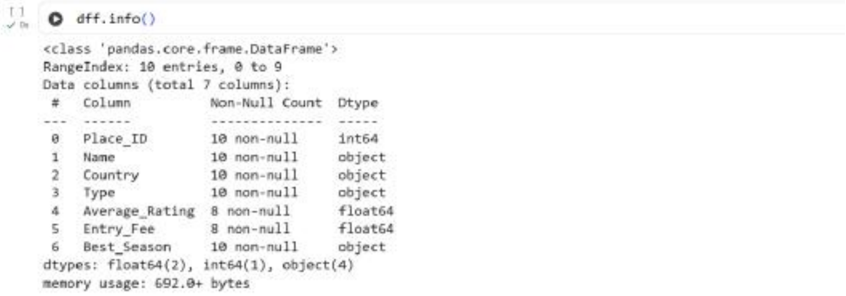
- Create a dataset
- Import pandas and numpy and with its help, create dataframe and change it to csv file
- Read the dataset and handle inappropriate and missing values

Program:



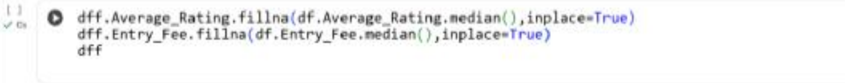
```
import pandas as pd
import numpy as np
data = {
    "Place_ID": [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
    "Name": [
        "Eiffel Tower", "Grand Canyon", "Great Barrier Reef", "Machu Picchu", "Mount Fuji",
        "Santorini", "Niagara Falls", "Petra", "Banff National Park", "Colosseum"
    ],
    "Country": [
        "France", "USA", "Australia", "Peru", "Japan",
        "Greece", "Canada", "Jordan", "Canada", "Italy"
    ],
    "Type": [
        "Monument", "Natural Wonder", "Natural Wonder", "Historical", "Mountain",
        "Island", "Waterfall", "Historical", "National Park", "Monument"
    ],
    "Average_Rating": [4.7, 4.8, 4.9, None, 4.6, 4.8, 4.7, 4.6, 4.9, None],
    "Entry_Fee": [25, 35, None, 45, 0, 10, None, 70, 20, 18],
    "Best_Season": [
        "Spring", "Fall", "Summer", "Spring", "Autumn",
        "Summer", "All year", "Winter", "Summer", "Spring"
    ]
}
df=pd.DataFrame(data)
df.to_csv('products.csv', index=False)
dff=pd.read_csv('products.csv')

dff
```

	Place_ID	Name	Country	Type	Average_Rating	Entry_Fee	Best_Season
0	1	Eiffel Tower	France	Monument	4.7	25.0	Spring
1	2	Grand Canyon	USA	Natural Wonder	4.8	35.0	Fall
2	3	Great Barrier Reef	Australia	Natural Wonder	4.9	NaN	Summer
3	4	Machu Picchu	Peru	Historical	NaN	45.0	Spring
4	5	Mount Fuji	Japan	Mountain	4.6	0.0	Autumn
5	6	Santorini	Greece	Island	4.8	10.0	Summer
6	7	Niagara Falls	Canada	Waterfall	4.7	NaN	All year
7	8	Petra	Jordan	Historical	4.6	70.0	Winter
8	9	Banff National Park	Canada	National Park	4.9	20.0	Summer
9	10	Colosseum	Italy	Monument	NaN	18.0	Spring

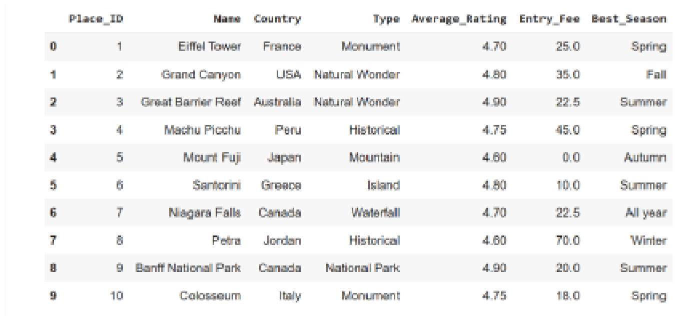


```
dff.info()
```

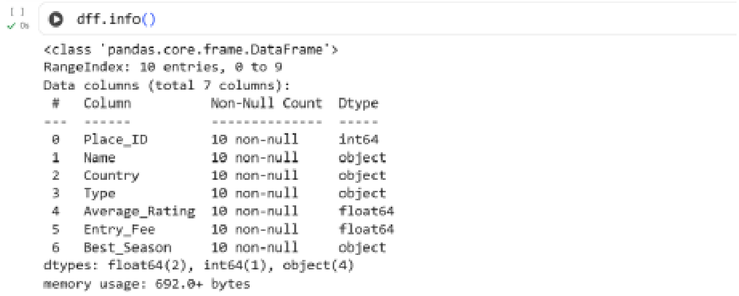
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 7 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Place_ID        10 non-null     int64  
 1   Name            10 non-null     object 
 2   Country         10 non-null     object 
 3   Type            10 non-null     object 
 4   Average_Rating  8 non-null      float64
 5   Entry_Fee       8 non-null      float64
 6   Best_Season     10 non-null     object 
dtypes: float64(2), int64(1), object(4)
memory usage: 692.0+ bytes
```



```
dff.Average_Rating.fillna(df.Average_Rating.median(),inplace=True)
dff.Entry_Fee.fillna(df.Entry_Fee.median(),inplace=True)
dff
```



	Place_ID	Name	Country	Type	Average_Rating	Entry_Fee	Best_Season
0	1	Eiffel Tower	France	Monument	4.70	25.0	Spring
1	2	Grand Canyon	USA	Natural Wonder	4.80	35.0	Fall
2	3	Great Barrier Reef	Australia	Natural Wonder	4.90	22.5	Summer
3	4	Machu Picchu	Peru	Historical	4.75	45.0	Spring
4	5	Mount Fuji	Japan	Mountain	4.60	0.0	Autumn
5	6	Santorini	Greece	Island	4.80	10.0	Summer
6	7	Niagara Falls	Canada	Waterfall	4.70	22.5	All year
7	8	Petra	Jordan	Historical	4.60	70.0	Winter
8	9	Banff National Park	Canada	National Park	4.90	20.0	Summer
9	10	Colosseum	Italy	Monument	4.75	18.0	Spring



```
dff.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 7 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Place_ID        10 non-null     int64
 1   Name            10 non-null     object
 2   Country         10 non-null     object
 3   Type            10 non-null     object
 4   Average_Rating  10 non-null     float64
 5   Entry_Fee       10 non-null     float64
 6   Best_Season     10 non-null     object
dtypes: float64(2), int64(1), object(4)
memory usage: 692.0+ bytes
```

Result:

Thus the python program to create own dataset and change it to csv file is executed and verified