Python Coding Assessment

Data Cleaning with Pandas

Dataset Used: Titanic Dataset from Kaggle
https://www.kaggle.com/datasets/yasserh/titanic-dataset

1) Loading Data in Pandas DataFrame and Printing rows of the Data

```
from google.colab import drive
import pandas as pd
drive.mount('/content/drive')
file_path = '/content/drive/My
Drive/case_study_dataset/Titanic-Dataset.csv'
df = pd.read_csv(file_path)
print('\n ==== First 5 rows in the Dataset ====')
print(df.head()) #prints first 5 rows
print('\n ==== Last 5 rows in the Dataset ====')
print(df.tail()) #print last 5 rows
```

Output

```
 ==== First 5 rows in the Dataset ====
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3

                                                Name     Sex   Age  SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0      1
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0      1
2                             Heikkinen, Miss. Laina  female  26.0      0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                           Allen, Mr. William Henry    male  35.0      0

   Parch            Ticket     Fare Cabin Embarked
0      0         A/5 21171   7.2500   NaN        S
1      0          PC 17599  71.2833   C85        C
2      0  STON/O2. 3101282   7.9250   NaN        S
3      0            113803  53.1000  C123        S
4      0            373450   8.0500   NaN        S
```

```
 ==== Last 5 rows in the Dataset ====
     PassengerId  Survived  Pclass                                      Name  \
886          887         0       2                     Montvila, Rev. Juozas
887          888         1       1              Graham, Miss. Margaret Edith
888          889         0       3  Johnston, Miss. Catherine Helen "Carrie"
889          890         1       1                     Behr, Mr. Karl Howell
890          891         0       3                       Dooley, Mr. Patrick

        Sex   Age  SibSp  Parch      Ticket   Fare Cabin Embarked
886    male  27.0      0      0      211536  13.00   NaN        S
887  female  19.0      0      0      112053  30.00   B42        S
888  female   NaN      1      2  W./C. 6607  23.45   NaN        S
889    male  26.0      0      0      111369  30.00  C148        C
890    male  32.0      0      0      370376   7.75   NaN        Q
```

2) Printing the column names of the DataFrame

```
print(df.columns)
```

Output

```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

3) Summary of Data Frame

```
print(df.info())
```

Output

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
None
```

4) Descriptive Statistical Measures of a DataFrame

```
print(df.describe())
```

Output

```
       PassengerId    Survived      Pclass         Age       SibSp  \
count   891.000000  891.000000  891.000000  714.000000  891.000000
mean    446.000000    0.383838    2.308642   29.699118    0.523008
std     257.353842    0.486592    0.836071   14.526497    1.102743
min       1.000000    0.000000    1.000000    0.420000    0.000000
25%     223.500000    0.000000    2.000000   20.125000    0.000000
50%     446.000000    0.000000    3.000000   28.000000    0.000000
75%     668.500000    1.000000    3.000000   38.000000    1.000000
max     891.000000    1.000000    3.000000   80.000000    8.000000

            Parch        Fare
count  891.000000  891.000000
mean     0.381594   32.204208
std      0.806057   49.693429
min      0.000000    0.000000
25%      0.000000    7.910400
50%      0.000000   14.454200
75%      0.000000   31.000000
max      6.000000  512.329200
```

5) Missing Data Handing

```
print(df.isnull().sum())
df_cleaned = df.dropna()
```

Output

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

After cleaning checking if there are any null values

```
print(df_cleaned.isnull().sum())
```

Output

```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Cabin          0
Embarked       0
dtype: int64
```

6) Sorting DataFrame values

```
df_sorted = df_cleaned.sort_values(by='PassengerId', ascending=False)
print(df_sorted.head())
```

Output

```
     PassengerId  Survived  Pclass  \
889          890         1       1
887          888         1       1
879          880         1       1
872          873         0       1
871          872         1       1

                                                 Name     Sex   Age  SibSp  \
889                              Behr, Mr. Karl Howell    male  26.0      0
887                        Graham, Miss. Margaret Edith  female  19.0      0
879     Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)  female  56.0      0
872                             Carlsson, Mr. Frans Olof    male  33.0      0
871  Beckwith, Mrs. Richard Leonard (Sallie Monypeny)  female  47.0      1

     Parch  Ticket     Fare        Cabin Embarked
889      0  111369  30.0000         C148        C
887      0  112053  30.0000          B42        S
879      1   11767  83.1583          C50        C
872      0     695   5.0000  B51 B53 B55        S
871      1   11751  52.5542          D35        S
```

7) Merge Data Frames

```
df1 = pd.read_csv(file_path)
df2 = pd.read_csv(file_path)
#merge data
df = pd.merge(df1, df2)
print(df)
```

Output

```
     PassengerId  Survived  Pclass  \
0              1         0       3
1              2         1       1
2              3         1       3
3              4         1       1
4              5         0       3
..           ...       ...     ...
886          887         0       2
887          888         1       1
888          889         0       3
889          890         1       1
890          891         0       3

                                                  Name     Sex   Age  SibSp  \
0                              Braund, Mr. Owen Harris    male  22.0      1
1    Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0      1
2                               Heikkinen, Miss. Laina  female  26.0      0
3         Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                             Allen, Mr. William Henry    male  35.0      0
..                                                 ...     ...   ...    ...
886                              Montvila, Rev. Juozas    male  27.0      0
887                       Graham, Miss. Margaret Edith  female  19.0      0
888           Johnston, Miss. Catherine Helen "Carrie"  female   NaN      1
889                              Behr, Mr. Karl Howell    male  26.0      0
890                                Dooley, Mr. Patrick    male  32.0      0

     Parch            Ticket     Fare Cabin Embarked
0        0         A/5 21171   7.2500   NaN        S
1        0          PC 17599  71.2833   C85        C
2        0  STON/O2. 3101282   7.9250   NaN        S
3        0            113803  53.1000  C123        S
4        0            373450   8.0500   NaN        S
..     ...               ...      ...   ...      ...
886      0            211536  13.0000   NaN        S
887      0            112053  30.0000   B42        S
888      2        W./C. 6607  23.4500   NaN        S
889      0            111369  30.0000  C148        C
890      0            370376   7.7500   NaN        Q
```

8) Apply Function

```
def status_upper(status):
  return status.upper()


df['Name'] = df['Name'].apply(status_upper)
print(df['Name'].head())
```

Output

```
0                                BRAUND, MR. OWEN HARRIS
1    CUMINGS, MRS. JOHN BRADLEY (FLORENCE BRIGGS TH...
2                                HEIKKINEN, MISS. LAINA
3          FUTRELLE, MRS. JACQUES HEATH (LILY MAY PEEL)
4                              ALLEN, MR. WILLIAM HENRY
Name: Name, dtype: object
```

Pandas Joins in Python

Inner Join

```
import pandas as pd


a = pd.DataFrame()
d = {'id': [1, 2, 10, 12, 13, 14, 15, 16],
     'val1': ['H', 'E', 'X', 'A', 'W', 'A', 'R','E']}
```

```
a = pd.DataFrame(d)
b = pd.DataFrame()

d = {'id': [1, 2, 9, 8],
     'val1': ['H', 'E', 'A', 'T']}
b = pd.DataFrame(d)
df = pd.merge(a, b, on='id', how='inner')
print(df)
```

Output

	id	val1_x	val1_y
0	1	H	H
1	2	E	E

Left Join

```
import pandas as pd

a = pd.DataFrame()
d = {'id': [1, 2, 10, 12, 13, 14, 15, 16],
     'val1': ['H', 'E', 'X', 'A', 'W', 'A', 'R','E']}

a = pd.DataFrame(d)
b = pd.DataFrame()

d = {'id': [1, 2, 9, 8],
     'val1': ['H', 'E', 'A', 'T']}
b = pd.DataFrame(d)
df = pd.merge(a, b, on='id', how='left')
print(df)
```

Output

	id	val1_x	val1_y
0	1	H	H
1	2	E	E
2	10	X	NaN
3	12	A	NaN
4	13	W	NaN
5	14	A	NaN
6	15	R	NaN
7	16	E	NaN

Right Outer Join

```
import pandas as pd

a = pd.DataFrame()
d = {'id': [1, 2, 10, 12, 13, 14, 15, 16],
     'val1': ['H', 'E', 'X', 'A', 'W', 'A', 'R','E']}

a = pd.DataFrame(d)
b = pd.DataFrame()

d = {'id': [1, 2, 9, 8],
     'val1': ['H', 'E', 'A', 'T']}
b = pd.DataFrame(d)
df = pd.merge(a, b, on='id', how='right')
print(df)
```

Output

	id	val1_x	val1_y
0	1	H	H
1	2	E	E
2	9	NaN	A
3	8	NaN	T

Full Outer Join

```
a = pd.DataFrame()
d = {'id': [1, 2, 10, 12, 13, 14, 15, 16],
     'val1': ['H', 'E', 'X', 'A', 'W', 'A', 'R','E']}

a = pd.DataFrame(d)
b = pd.DataFrame()

d = {'id': [1, 2, 9, 8],
     'val1': ['H', 'E', 'A', 'T']}
b = pd.DataFrame(d)
df = pd.merge(a, b, on='id', how='outer')
print(df)
```

Output

	id	val1_x	val1_y
0	1	H	H
1	2	E	E
2	8	NaN	T
3	9	NaN	A
4	10	X	NaN
5	12	A	NaN
6	13	W	NaN
7	14	A	NaN
8	15	R	NaN
9	16	E	NaN

Index Join

```
a = pd.DataFrame()
d = {'id': [1, 2, 10, 12, 13, 14, 15, 16],
     'val1': ['H', 'E', 'X', 'A', 'W', 'A', 'R','E']}

a = pd.DataFrame(d)
b = pd.DataFrame()

d = {'id': [1, 2, 9, 8],
     'val1': ['H', 'E', 'A', 'T']}
b = pd.DataFrame(d)
df = pd.merge(a, b, left_index=True, right_index=True)
print(df)
```

Output

	id_x	val1_x	id_y	val1_y
0	1	H	1	H
1	2	E	2	E
2	10	X	9	A
3	12	A	8	T