Name - Kumar Rohit

PRN - 2107029002

Batch - AML (2021-2025)

Dataset - 1

```
# Importing the required libraries
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

# Setting the path for the dataset
train_df = pd.read_csv("/kaggle/input/titanic/train.csv")
train_df.shape
```

```
(891, 12)
```

```
train_df.head()
```

	passenger_id	survived	p_class	name	sex	age	s_sib_sp	p_arch_ch	ticket	fare	cabin	embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Till now I'm familiar

```
# Checking the data types of the columns
train_df.dtypes
```

```
passenger_id    int64
survived        int64
p_class         int64
name           object
sex            object
age           float64
s_sib_sp        int64
p_arch_ch       int64
ticket         object
fare          float64
cabin          object
embarked       object
dtype: object
```

```
# Checking the info of the dataset
train_df.info()
```

```
<class pandas.core.frame.DataFrame>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 passenger_id  891 non-null    int64
 survived      891 non-null    int64
 p_class       891 non-null    int64
 name          891 non-null    object
 sex           891 non-null    object
 age           714 non-null    float64
 s_sib_sp      891 non-null    int64
 p_arch_ch     891 non-null    int64
 ticket        891 non-null    object
 fare          891 non-null    float64
 cabin         204 non-null    object
 embarked      889 non-null    object
dtypes: float64(2), int64(7), object(3)
memory usage: 83.7+ KB
```

```
train_df.describe()
```

	passenger_id	survived	p_class	age	s_sib_sp	p_arch_ch	fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

```
# Checking the shape of the dataset
train_df.shape
```

```
(891, 12)
```

```
train_df.columns
```

```
Index(['passenger_id', 'survived', 'p_class', 'name', 'sex', 'age',
       's_sib_sp', 'p_arch_ch', 'ticket', 'fare', 'cabin', 'embarked',
       dtype: object
```

```
# Checking the number of unique values in each column
train_df.nunique()
```

```
passenger_id    891
survived          2
p_class           3
name            891
sex               2
age              88
s_sib_sp          7
p_arch_ch         7
ticket          681
fare            248
cabin           147
embarked          3
dtype: int64
```

```
# Checking the null values in each column
train_df.isnull()
```

	passenger_id	survived	p_class	name	sex	age	s_sib_sp	p_arch_ch	ticket	fare	cabin	embarked
0	False	False	False	False	False	False	False	False	False	False	True	False
1	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	True	False
3	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	True	False

```
# Checking the number of null values in each column
train_df.isnull().sum()
```

```
passenger_id    0
survived        0
p_class         0
name            0
sex             0
age           177
s_sib_sp        0
p_arch_ch       0
ticket          0
fare            0
cabin         687
embarked        2
dtype: int64
```

```
# Checking the unique values in the embarked column
train_df['embarked'].unique()
```

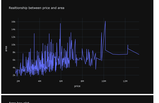
```
array['S', 'C', 'Q', nan]
```

```
# Checking the value counts in the embarked column
train_df['embarked'].value_counts()
```

```
embarked
S    644
C    168
Q     77
Name: count, dtype: int64
```

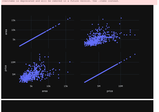
```
# Checking the correlation between the columns
train_df.corr()
```

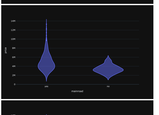
	passenger_id	survived	p_class	age	s_sib_sp	p_arch_ch	fare
passenger_id	1.000000	-0.005007	-0.035144	0.036847	-0.057527	-0.001652	0.012658
survived	-0.005007	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.257307
p_class	-0.035144	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.549500
age	0.036847	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.096067
s_sib_sp	-0.057527	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.159651
p_arch_ch	-0.001652	0.081629	0.018443	-0.189119	0.414838	1.000000	0.216225
fare	0.012658	0.257307	-0.549500	0.096067	0.159651	0.216225	1.000000

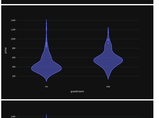


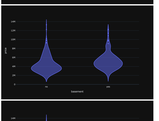
```
Numerical columns:['price', 'area', 'bedrooms', 'bathrooms', 'stories', 'mainroad', 'guestroom', 'basement', 'hotwaterheating', 'airconditioning', 'parking', 'prefarea', 'furnishingstatus']
Categorical columns:['mainroad', 'guestroom', 'basement', 'hotwaterheating', 'airconditioning', 'prefarea', 'furnishingstatus']
```

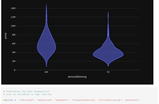
```
# Boxplot for each numerical column
numerical_cols = ['price', 'area']
for col in numerical_cols:
    plt.figure(figure=(8, 5))
    sns.boxplot(data=df, x=col)
    plt.title(f'Boxplot of {col}')
    plt.show()
```











```
# Calculate correlation matrix
correlation_matrix = df[numerical_cols].corr()
print("Correlation matrix:")
print(correlation_matrix)
```

```
Correlation matrix:
              price      area  bedrooms  bathrooms   stories  mainroad  guestroom  basement  hotwaterheating  airconditioning   parking  prefarea  furnishingstatus
price      1.000000  0.535997  0.366494   0.517545  0.420712  0.296894   0.255517  0.187057         0.093073         0.452954  0.384394  0.329777          0.304721
area       0.535997  1.000000  0.151858   0.193820  0.083796  0.088000   0.141090  0.047417         0.052167         0.205672  0.352980  0.375982          0.003026
bedrooms   0.366494  0.151858  1.000000   0.373930  0.408564  0.046049   0.080578  0.097670         0.046049         0.052510  0.139270  0.080549          0.305086
bathrooms  0.517545  0.193820  0.373930   1.000000  0.326165  0.036454   0.126039  0.176828         0.085754         0.038610  0.178806  0.064610          0.150404
stories    0.420712  0.083796  0.408564   0.326165  1.000000  0.045547   0.044099  0.093073         0.077544         0.083542  0.171562  0.044870          0.166698
mainroad   0.296894  0.088000  0.046049   0.036454  0.045547  1.000000   0.037466  0.052359         0.087799         0.092272  0.046049  0.078744          0.041544
guestroom  0.255517  0.141090  0.080578   0.126039  0.044099  0.037466   1.000000  0.364426         0.039265         0.071966  0.077586  0.099186          0.040480
basement   0.187057  0.047417  0.097670   0.176828  0.093073  0.052359   0.364426  1.000000         0.049067         0.117846  0.037350  0.071683          0.177394
hotwaterheating  0.093073  0.052167  0.046049  0.085754  0.077544  0.087799  0.039265  0.049067  1.000000  0.092865  0.093073  0.168379  0.129148
airconditioning  0.452954  0.205672  0.052510  0.038610  0.083542  0.092272  0.071966  0.117846  0.092865  1.000000  0.158403  0.322390  0.038610
parking    0.384394  0.352980  0.139270  0.178806  0.171562  0.046049  0.077586  0.037350  0.093073  0.158403  1.000000  0.234351  0.052510
prefarea   0.329777  0.375982  0.080549  0.064610  0.044870  0.078744  0.099186  0.071683  0.168379  0.322390  0.234351  1.000000  0.092865
furnishingstatus  0.304721  0.003026  0.305086  0.150404  0.166698  0.041544  0.040480  0.177394  0.129148  0.038610  0.052510  0.092865  1.000000
```

```
# Visualize correlation matrix
plt.figure(figure=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Matrix')
plt.show()
```

```
# Boxplot for each numerical column
numerical_cols = ['price', 'area']
for col in numerical_cols:
    plt.figure(figure=(8, 5))
    sns.boxplot(data=df, x=col)
    plt.title(f'Boxplot of {col}')
    plt.show()
```

```
# Boxplot for each numerical column
numerical_cols = ['price', 'area']
for col in numerical_cols:
    plt.figure(figure=(8, 5))
    sns.boxplot(data=df, x=col)
    plt.title(f'Boxplot of {col}')
    plt.show
```

```
# Boxplot for each numerical column
numerical_cols = ['price', 'area']
for col in numerical_cols:
    plt.figure(figure=(8, 5))
    sns.boxplot(data=df, x=col)
    plt.title(f'Boxplot of {col}')
    plt.show()
```