1. There are totally **12 features** in the training set. They are **PassengerId, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin, Embarked**.
2. The features that are categorial are: **Survived, Pclass, Sex and Embarked.**
3. The features that are numerical are **PassengerID, Survived, Pclass, Age, SibSp, Parch and Fare**
4. The features that have mixed data type are **Ticket and Cabin**.
5. In training data, the features that contain blank, null or empty values are **Age (714 values), Cabin (204 values) and Embarked (889 values)**. Similarly, in the test data, the features that contain blank, null or empty values are **Age (332 values), Fare (417 values) and Cabin (91 values).**
6. The different data types in the training set are **float 64** (Age and Fare), **Object/String** (Name, Ticket, Cabin, Embarked) and **int 64** (PassengerId, Survived, Pclass, Sex, SibSp, and Parch)

**######################################################################################################**

**Output for questions from 1 through 6:**

**Output for the training set:**

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 int64

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(6), object(4)

**Output for the test set:**

RangeIndex: 418 entries, 0 to 417

Data columns (total 11 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 PassengerId 418 non-null int64

1 Pclass 418 non-null int64

2 Name 418 non-null object

3 Sex 418 non-null object

4 Age 332 non-null float64

5 SibSp 418 non-null int64

6 Parch 418 non-null int64

7 Ticket 418 non-null object

8 Fare 417 non-null float64

9 Cabin 91 non-null object

10 Embarked 418 non-null object

dtypes: float64(2), int64(4), object(5)

**Number of missing values in each feature of the training set:**

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

**Number of missing values in each feature of the testing set:**

PassengerId 0

Pclass 0

Name 0

Sex 0

Age 86

SibSp 0

Parch 0

Ticket 0

Fare 1

Cabin 327

Embarked 0

dtype: int64

############################################################################################

1. The following are the statistical data to understand the distribution of numerical feature values across the samples:

#####################################################################################################

**Output for question 7:**

A screen shot of a computer

Description automatically generated

PassengerId Survived Pclass Sex Age SibSp Parch Fare

count 891.000000 891.000000 891.000000 891.000000 714.000000 891.000000 891.000000 891.000000

mean 446.000000 0.383838 2.308642 0.647587 29.699118 0.523008 0.381594 32.204208

std 257.353842 0.486592 0.836071 0.477990 14.526497 1.102743 0.806057 49.693429

min 1.000000 0.000000 1.000000 0.000000 0.420000 0.000000 0.000000 0.000000

25% 223.500000 0.000000 2.000000 0.000000 20.125000 0.000000 0.000000 7.910400

50% 446.000000 0.000000 3.000000 1.000000 28.000000 0.000000 0.000000 14.454200

75% 668.500000 1.000000 3.000000 1.000000 38.000000 1.000000 0.000000 31.000000

max 891.000000 1.000000 3.000000 1.000000 80.000000 8.000000 6.000000 512.329200

###################################################################################################

1. First, the dtype of all the categorical features are converted to category and then the output is printed.

###################################################################################################

**Output of question 8:**

Survived Pclass Sex Embarked

count 891 891 891 889

unique 2 3 2 3

top 0 3 male S

freq 549 491 577 644

######################################################################################################

1. Yes, we should include this feature. When **Pclass = 1 there were 136 out of 216 passengers survived** which gives a surviving ratio of 0.62.

####################################################################################################

**Output for question 9:**

A picture containing object

Description automatically generated

Survived 0 1

Pclass

1 80 136

2 97 87

3 372 119

A screenshot of a cell phone

Description automatically generated

####################################################################################################

1. Yes, **women (233) are more likely to survive compared to men (189**).

###################################################################################################

**Output for question 10:**

Survived 0 1

Sex

0 81 233

1 468 109

A screenshot of a cell phone

Description automatically generated

#####################################################################################################

1. A) Yes, **infants under the age of 4 have high survival rate**.

B) Yes, **the oldest passenger survived**.

C) Yes, **a large number of people between the age 15 – 25 did not survive** (79 survived, 144 did not survive).

D) Yes, **we should consider age for our model training**.

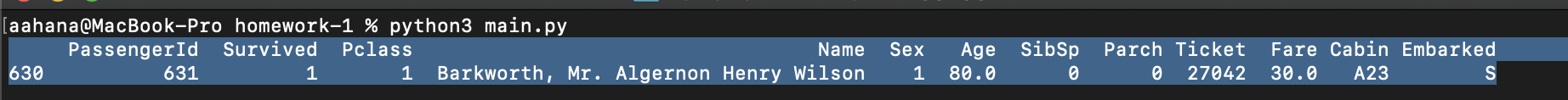
E) Yes, **it makes it easier to analyze the data set**.

#####################################################################################################

**Output for question 11:**

A screenshot of a cell phone

Description automatically generated



############################################################################################

1. A) Yes, **Pclass = 3 had most passengers, out of which many did not survive.**

B) Yes, **infant passengers (age <= 4) tend to survive**.

C) Yes, **most passengers in Pclass = 1 survive**.

D) Yes, **Pclass varies in terms of age distribution of passengers**.

E) Yes, **Pclass data set seems to provide vital information for our machine learning model**.

##############################################################################################

**Output for question 12:**

A picture containing screenshot

Description automatically generated

##################################################################################################

1. A) Yes, **those passengers that embarked to Southampton and Cherbourg with higher fare tickets seemed to have a higher survival rate as compared to those that traveled with lower fare tickets. Passengers that embarked to Queenstown seem to have very similar survival rate irrespective of the ticket fare.**

B) Yes, **we should consider banding fare feature**. The fare feature seems to group well in those intervals of probably 10 – 20..

#################################################################################################

**Output for question 13:**

A screenshot of a cell phone

Description automatically generated

####################################################################################################

1. There is a total of 891 ticket entries out of which about 681 are unique ticket values. Hence, **the duplication rate is 23.56% and remaining 76.43% of the ticket values are unique**. Duplication rate is calculated using the formula**, rate of duplicates = (total records - unique records)/total records.** However, the ticket feature does not seem to provide any useful information and thus we can drop this column.

####################################################################################################

**Output for question 14:**

Total number of ticket entries 891

Number of unique ticket values 681

Total ticket duplicate rate is 23.569023569023567 %

####################################################################################################

1. Yes, I would drop the Cabin feature. There are 891 entries in the train\_df data set and 418 features in the test\_df data set. The total number of entries for each feature in the combined data set is 1309 entries. Out of this, **the cabin feature has 1014 entries that are null valued**. Hence, we can go ahead and drop this feature.

####################################################################################################

**Output for Question 15:**

**Train\_df:**

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

**Test\_df:**

PassengerId 0

Pclass 0

Name 0

Sex 0

Age 86

SibSp 0

Parch 0

Ticket 0

Fare 1

Cabin 327

Embarked 0

dtype: int64

Out of 1309 entries in the cabin feature, the number of null values in Cabin feature are: 1014 entries.

##################################################################################################

1. All female entries in the Sex feature are assigned as 1 and all male entries in the Sex feature are assigned as 0.

##################################################################################################

**Output for question 16:**

0 0

1 1

2 1

3 1

4 0

..

886 0

887 1

888 1

889 0

890 0

Name: Sex, Length: 891, dtype: int64

###########################################################################################################

1. All Nan values in the Age feature are replaced using K-Nearest Neighbor algorithm, where k = 5.

###################################################################################################

**Output for question 17:**

**Some of the original and modified (using KNN algorithm) Age values are:**

22.0 22.0

38.0 38.0

26.0 26.0

35.0 35.0

35.0 35.0

nan 29.69911764705882

54.0 54.0

2.0 2.0

27.0 27.0

14.0 14.0

4.0 4.0

58.0 58.0

20.0 20.0

39.0 39.0

14.0 14.0

55.0 55.0

2.0 2.0

nan 29.69911764705882

31.0 31.0

nan 29.69911764705882

35.0 35.0

The number of Nan values in the Age column is: 0

###################################################################################################

1. The training data set has 2 missing values. The common frequency (mode) in the ‘Embarked’ feature is S. Hence, the two empty values are filled with S.

####################################################################################################

**Output for question 18:**

The mode for Embarked freature is: 0 S

dtype: object

The number of empty values in Embarked feature is: 2

After filling the empty values, the number of empty values in the Embarked feature is: 0

The following are the values in the Embarked feature: 0 S

1 C

2 S

3 S

4 S

..

886 S

887 S

888 S

889 C

890 Q

Name: Embarked, Length: 891, dtype: object

##########################################################################################

1. There is only one empty data in the Fare feature of the testing data set. This empty value is filled with the mode (most common frequency) of the fare feature. The mode is 7.75.

###################################################################################################

**Output for question 19**

In the testing dataset, The mode for Fare freature is: 0 7.75

dtype: float64

The number of empty values in Fare feature is: 1

After filling the empty values, the number of empty values in the Fare feature is: 0

The following are the values in the Fare feature: 0 7.8292

1 7.0000

2 9.6875

3 8.6625

4 12.2875

...

413 8.0500

414 108.9000

415 7.2500

416 8.0500

417 22.3583

Name: Fare, Length: 418, dtype: float64

##########################################################################################

1. The Fare feature has been binned and corresponding ordinal values (0, 1, 2 and 3) are assigned.

####################################################################################################

**Output for question 20:**

PassengerId Survived Pclass ... Cabin Embarked Fare\_bin

0 1 0 3 ... NaN S 0

1 2 1 1 ... C85 C 3

2 3 1 3 ... NaN S 1

3 4 1 1 ... C123 S 3

4 5 0 3 ... NaN S 1

[5 rows x 13 columns]

####################################################################################################

**GitHub link to the code:**

<https://github.com/monicabernard/CAP-5610_Machine-Learning.git>