Exploratory data analysis for machine learning

Dataset Introduction

The dataset used here is Titanic - Machine Learning from Disaster. This dataset is used for an example for classification problems.
The datasets consists of the following columns

Feature	Descriptions
survival	This feature tells us whether the passenger survived or not. This feature is gonna be used for predicting 0 = No; 1 = Yes
PassengerId	ID of the observation
pclass	Passenger Class (1 = 1st; 2 = 2nd; 3 = 3rd)
name	Name of the passenger
sex	Sex of the passenger
age	Age of the passenger
sibsp	Number of siblings or spouse traveling with the passenger
parch	Number of parents, children traveling with the passenger
ticket	Ticket number of the passenger
fare	Fare amount paid by the passenger
cabin	Cabin allocated for the passenger
embarked	Place of embarkment of the passenger (C = Cherbourg; Q = Queenstown; S = Southampton)

Total observation length = 891

Exploratory data analysis

Feature deletion

Before we start with the actual analysis we write a function to find features having unique value as much as observation length. With this function we get to know 2 features consists of unique value for each observations .Those are

1) Passenger ID
2) Name

As these features do not provide any useful information on predicting the survived column . We shall delete these features

Now using data.head() to check the remaining features. We notice that the Ticket feature looks a little suspicious . On doing some analysis we get to tis feature has a total of 681 unique values. As this feature is a categorical variable having a total 681 categories means an addition of 681 features to the model .So we shall find any order among the categories .As this feature only represents the ticket number of the passenger we are not able to find any order so we shall also delete this feature
So we deleted the passengerId, Name and Ticket feature to make our analysis accurate

Numerical Feature analysis

We have a total of 6 numerical features .They are

1) Survived
2) Pclass
3) Age
4) SibSp
5) Parch
6) Fare

We shall analysis each of these features one by one

Survived Feature

- Denotes whether a passenger survived or not
- Numerical column which has discrete values

Null values

We shall first check for any null values by using the isnull function.
The output was that this feature had 0 null values

```
data2['Survived'].isnull().sum()
```

```
0
```

Unique values

2 unique respectively

- 0 for died
- 1 for survived

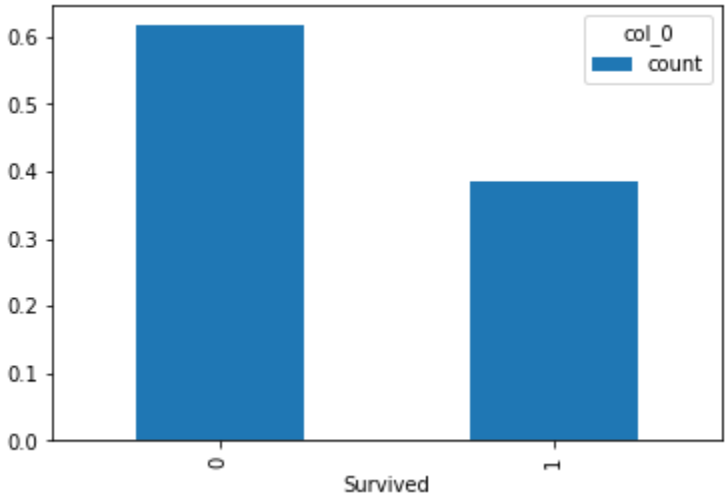
```
data2['Survived'].unique()
```

```
array([0, 1])
```

Distribution

We shall check the difference between survived and died passengers count

Survived	col_0 count
0	0.616162
1	0.383838



From the above 2 pictures we can conclude that 61% passenger died and only 38% survived

Pclass Feature

- Numerical feature with order discrete values
- Denotes the class of the passenger

Null Values

We shall first check for any null values by using the isnull function.
The output was that this feature had 0 null values

```
data2['Pclass'].isnull().sum()
```

```
0
```

Unique values

3 unique values respectively

- 1 for class 1
- 2 for class 2
- 3 for class 3

```
data2['Pclass'].unique()
```

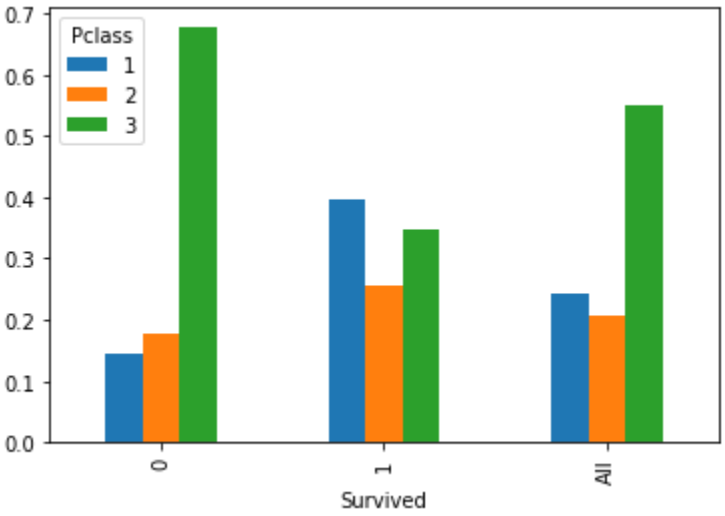
```
array([3, 1, 2])
```

Distribution

We shall check the distribution between classes

categories	counts	freqs
1	216	0.242424
2	184	0.206510
3	491	0.551066

Pclass	1	2	3	All
Survived				
0	0.089787	0.108866	0.417508	0.616162
1	0.152637	0.097643	0.133558	0.383838
All	0.242424	0.206510	0.551066	1.000000



We can check the majority of passenger who died were of class 3

We are gonna use ordinal encoding to encode this feature

Age Feature

- Continuous Variable
- Denotes the age of the passenger

Null Values

We shall first check for any null values by using the isnull function.
The output was that this feature had 177 null values
We know that age follows a normal distribution so we fill the null values with mean of the feature

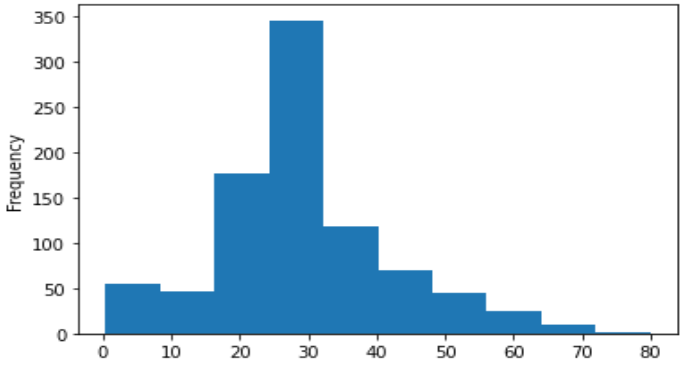
```
data3['Age'].isnull().sum()
```

```
177
```

Distribution

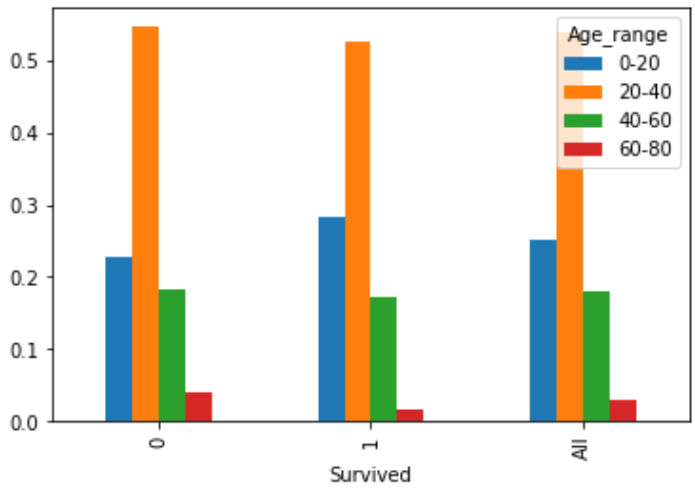
We shall check the distribution of the feature

We can see the the distribution is not normal so we may need to apply feature transformation



Now we shall see the distribution of survived passenger according age groups before replacing the null values

Age_range	0-20	20-40	40-60	60-80	All
Survived					
0	0.135854	0.324930	0.109244	0.023810	0.593838
1	0.114846	0.214286	0.070028	0.007003	0.406162
All	0.250700	0.539216	0.179272	0.030812	1.000000



We can see that the majority passenger in the age group 20-40 died the most and also survived the most

SibSp amd Parch feature

- SibSp denote the number of siblings or spouses the passenger is traveling with
- Parch denotes number of parents and children the passenger is traveling with
- Both take discrete values

Null Values

We shall first check for any null values by using the isnull function.
The output was that this feature had 0 null values

```
data3['SibSp'].isnull().sum()
```

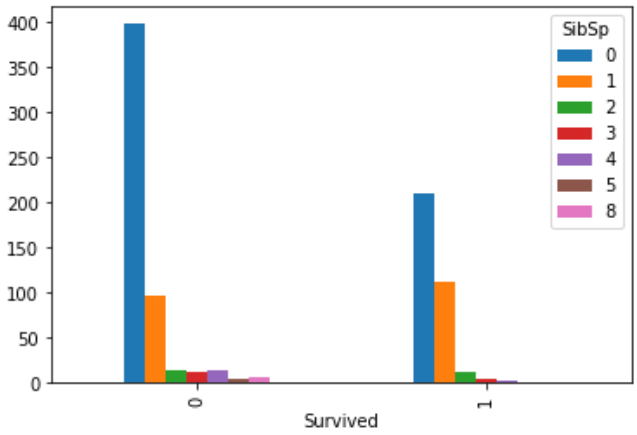
```
0
```

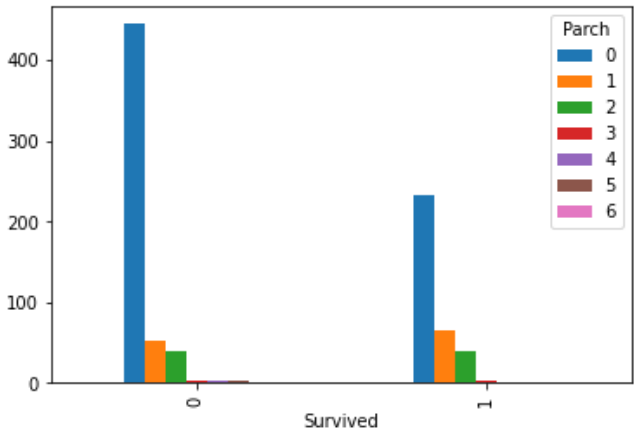
```
data3['Parch'].isnull().sum()
```

```
0
```

Distribution

Checking the passengers survived based on number of spouses and siblings and also number of parents and children





SibSp	0	1	2	3	4	5	8
Survived							
0	398	97	15	12	15	5	7
1	210	112	13	4	3	0	0

Parch	0	1	2	3	4	5	6
Survived							
0	445	53	40	2	4	4	1
1	233	65	40	3	0	1	0

There isn't a clear distinction between survived and having siblings or parents or children or spouses.We shall create a new feature called with family to find the distribution of having family and survived

With family

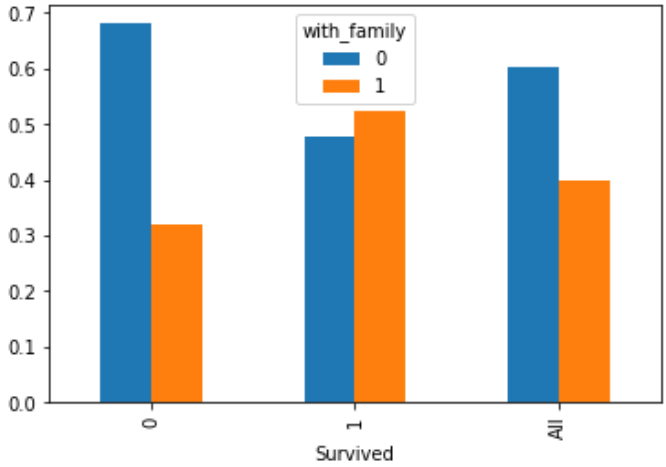
- denotes 1 if atleast one spouse, sibling, children, parent traveling with the passenger
- denotes 0 if has no family traveling with the passenger

with_family
1
1
0
1

Distribution

We shall see the distribution between with_family and survived

with_family	0	1	All
Survived			
0	0.419753	0.196409	0.616162
1	0.182941	0.200898	0.383838
All	0.602694	0.397306	1.000000



We can see that 60% passengers didn't have family and 39% had family with them and the passenger who didn't have a family traveling with them died the most

Fare column

- Amount of money in dollars paid by the passenger
- Continuous variable

Null value

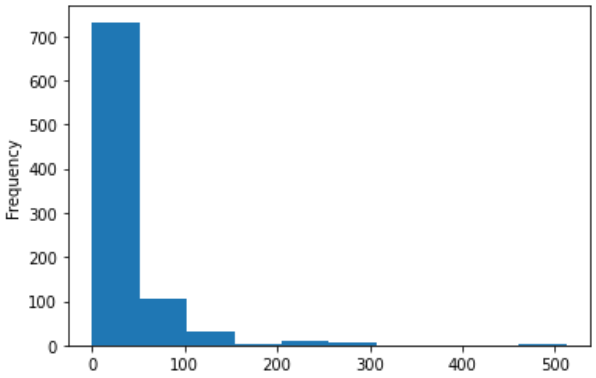
We shall first check for any null values by using the isnull function.
The output was that this feature had 0 null values

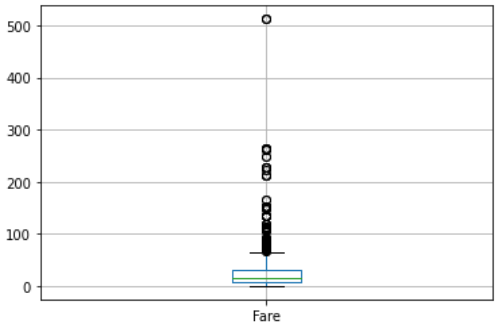
```
data3['Fare'].isnull().sum()
```

```
0
```

Distribution

We can see data is left skewed ,This is because price of ticket is constant depending on class and etc We can also notice some outliers





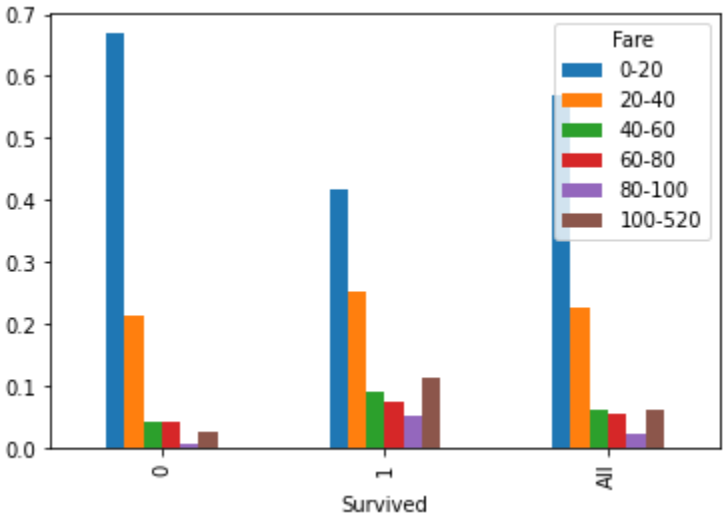
There are lot of outliers here.Let's try to understand why the outlier occurs

	Survived	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked	Pclass_1	Pclass_2	Pclass_3	with_family
27	0	male	19.000000	3	2	263.0000	C23 C25 C27	S	1	0	0	1
31	1	female	29.699118	1	0	146.5208	B78	C	1	0	0	1
88	1	female	23.000000	3	2	263.0000	C23 C25 C27	S	1	0	0	1
118	0	male	24.000000	0	1	247.5208	B58 B60	C	1	0	0	1
195	1	female	58.000000	0	0	146.5208	B80	C	1	0	0	0
215	1	female	31.000000	1	0	113.2750	D36	C	1	0	0	1
258	1	female	35.000000	0	0	512.3292	NaN	C	1	0	0	0
268	1	female	58.000000	0	1	153.4625	C125	S	1	0	0	1
269	1	female	35.000000	0	0	135.6333	C99	S	1	0	0	0
297	0	female	2.000000	1	2	151.5500	C22 C26	S	1	0	0	1

We can see that the passengers who paid the fare a lot wanted the class 1 of the passenger class because the pclass1 of these observations is all 1 .
So we won't remove the outliers as this provides some useful information.

Let's check the distribution of survived versus fare paid

Fare	0-20	20-40	40-60	60-80	80-100	100-520	All
Survived							
0	0.408676	0.130137	0.026256	0.026256	0.003425	0.015982	0.610731
1	0.162100	0.098174	0.035388	0.028539	0.020548	0.044521	0.389269
All	0.570776	0.228311	0.061644	0.054795	0.023973	0.060502	1.000000



We can see the passenger who paid less fair survived the less compared to who paid more

Categorical Feature Analysis

There are 3 Categorical Features. They are

1) Sex
2) Cabin
3) Embarked

We shall analysis each of these features one by one

Sex Feature

- Categorical variable
- Denotes the sex of the passenger

Null values

We shall first check for any null values by using the isnull function.
The output was that this feature had 0 null values

```
data4['Sex'].isnull().sum()
```

```
0
```

Unique values

There are 2 unique categories respectively
-Male
-Female

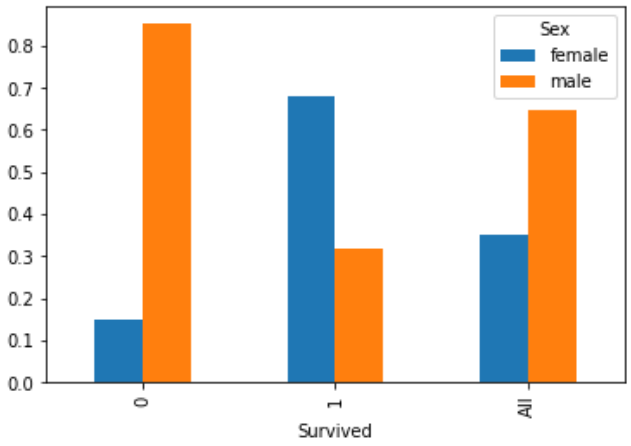
```
data4['Sex'].unique()
```

```
array(['male', 'female'], dtype=object)
```

Distribution

We shall check the distribution of Sex wrt survived

Sex	female	male	All
Survived			
0	0.090909	0.525253	0.616162
1	0.261504	0.122334	0.383838
All	0.352413	0.647587	1.000000



We can see that Male(52%) died more than Female(9%)

Now we will apply one hot encoding to encode this feature

Sex_female	Sex_male
0	1
1	0
1	0
1	0

Cabin Feature

- Categorical feature
- Feature denoting the cabin of the passengers

Null Values

We shall first check for any null values by using the isnull function.
The output was that this feature had 687 null values
It has a total 687 null value among 891 observations

```
data4['Cabin'].isnull().sum()
```

```
687
```

A normal solution should be to delete the feature for lack of information but cabin gives crucial information of the survival of passengers. Suppose the cabin A gets flooded first there is a high that the passengers in that cabin will die so instead of deleting the feature we will create a new category of that feature for null values called no cabin or n

Unique values

There are a lot of unique values but we can see some order if you observe the first character . So we shall only take the first character for the categories in this features

```
data4['Cabin'].unique()
```

```
array([nan, 'C85', 'C123', 'E46', 'G6', 'C103', 'D56', 'A6',
       'C23 C25 C27', 'B78', 'D33', 'B30', 'C52', 'B28', 'C83', 'F33',
       'F G73', 'E31', 'A5', 'D10 D12', 'D26', 'C110', 'B58 B60', 'E101',
       'F E69', 'D47', 'B86', 'F2', 'C2', 'E33', 'B19', 'A7', 'C49', 'F4',
       'A32', 'B4', 'B80', 'A31', 'D36', 'D15', 'C93', 'C78', 'D35',
       'C87', 'B77', 'E67', 'B94', 'C125', 'C99', 'C118', 'D7', 'A19',
       'B49', 'D', 'C22 C26', 'C106', 'C65', 'E36', 'C54',
       'B57 B59 B63 B66', 'C7', 'E34', 'C32', 'B18', 'C124', 'C91', 'E40',
       'T', 'C128', 'D37', 'B35', 'E50', 'C82', 'B96 B98', 'E10', 'E44',
       'A34', 'C104', 'C111', 'C92', 'E38', 'D21', 'E12', 'E63', 'A14',
       'B37', 'C30', 'D20', 'B79', 'E25', 'D46', 'B73', 'C95', 'B38',
       'B39', 'B22', 'C86', 'C70', 'A16', 'C101', 'C68', 'A10', 'E68',
       'B41', 'A20', 'D19', 'D50', 'D9', 'A23', 'B50', 'A26', 'D48',
       'E58', 'C126', 'B71', 'B51 B53 B55', 'D49', 'B5', 'B20', 'F G63',
       'C62 C64', 'E24', 'C90', 'C45', 'E8', 'B101', 'D45', 'C46', 'D30',
       'E121', 'D11', 'E77', 'F38', 'B3', 'D6', 'B82 B84', 'D17', 'A36',
       'B102', 'B69', 'E49', 'C47', 'D28', 'E17', 'A24', 'C50', 'B42',
       'C148'], dtype=object)
```

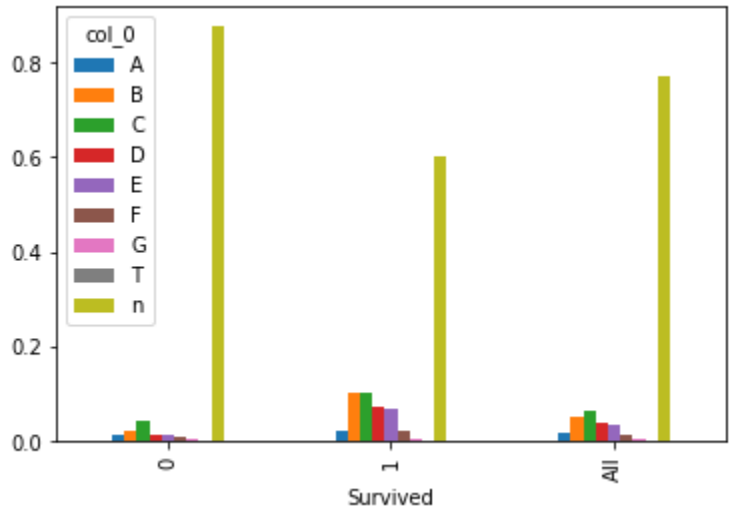
col_0 row_0	count
A	15
B	47
C	59
D	33
E	32
F	13
G	4
T	1
n	687

After string processing we can see that categories are reduced to 9. We can see the distribution of passengers per cabin

Distribution

Let's check the distribution wrt to survived

col_0	A	B	C	D	E	F	G	T	n	All
Survived										
0	0.008979	0.013468	0.026936	0.008979	0.008979	0.005612	0.002245	0.001122	0.539843	0.616162
1	0.007856	0.039282	0.039282	0.028058	0.026936	0.008979	0.002245	0.000000	0.231201	0.383838
All	0.016835	0.052750	0.066218	0.037037	0.035915	0.014590	0.004489	0.001122	0.771044	1.000000



Because the no cabin dominates the overall distribution we can't come to any conclusion

Embark column

- Categorical column
- Denotes the port of embarkment

Null values

We shall first check for any null values by using the isnull function.
The output was that this feature had 2 null values
We shall fill them after understanding the distribution of the categories in the features

```
data5['Embarked'].isnull().sum()
```

```
2
```

Unique values

There were 3 unique values respectively

- S(Southampton)
- Q(Queenstown)
- C(Cherbourg)

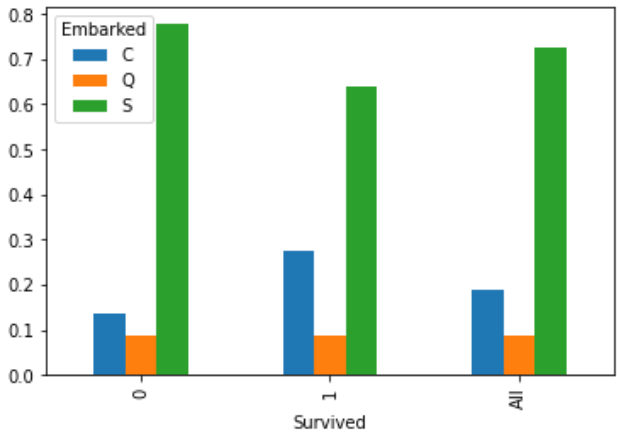
```
data5['Embarked'].unique()
```

```
array(['S', 'C', 'Q'], dtype=object)
```

Distribution

Let's check the distribution wrt survived

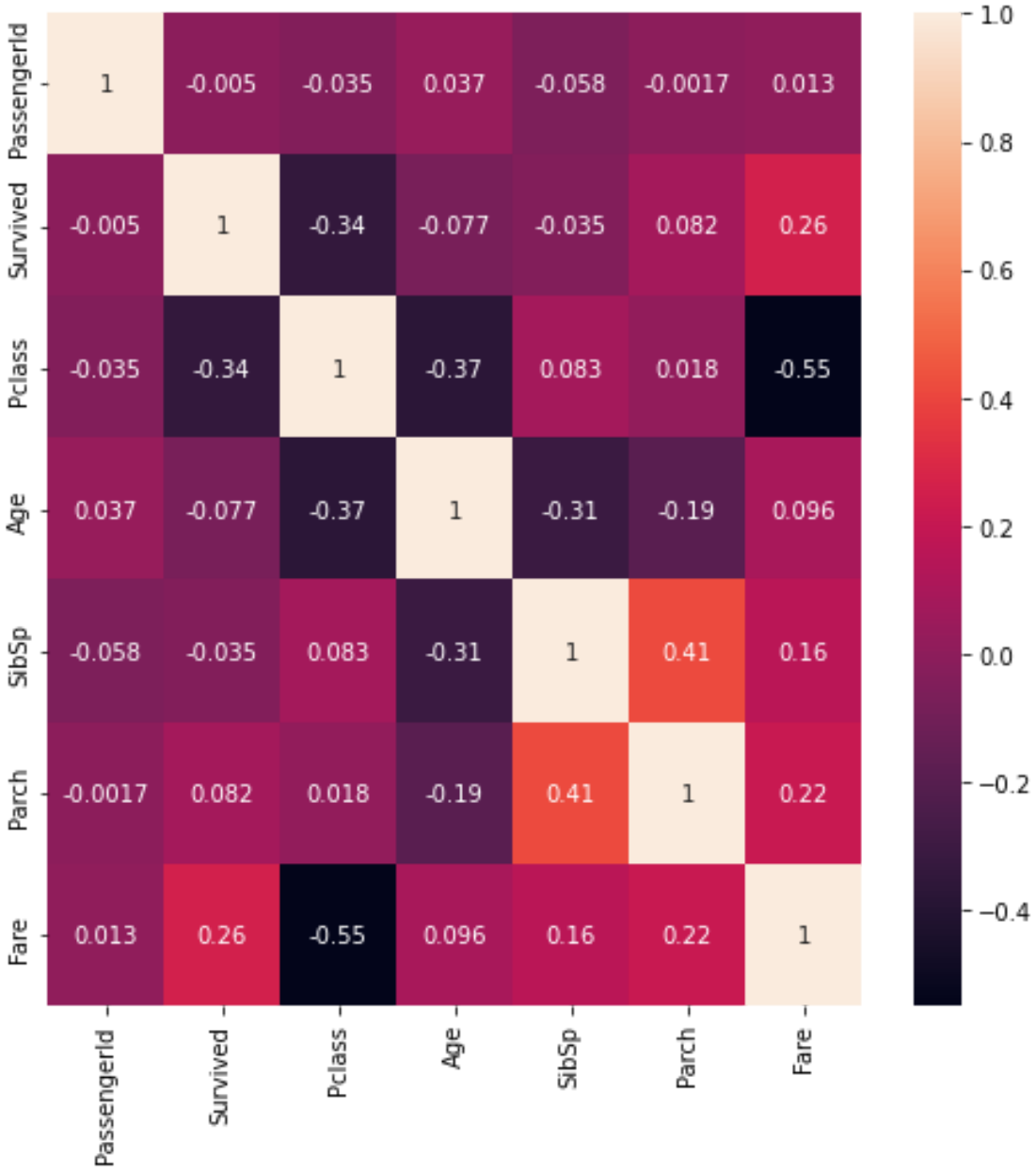
Embarked	C	Q	S	All
Survived				
0	0.084364	0.052868	0.480315	0.617548
1	0.104612	0.033746	0.244094	0.382452
All	0.188976	0.086614	0.724409	1.000000



We can see that 72 % of the passengers were
from Southampton so we would fill the nan values with Southampton

Interaction Terms

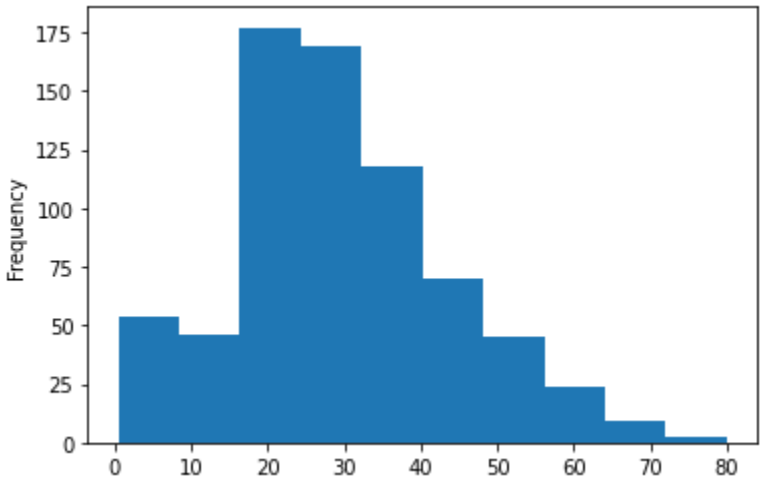
Before Calculating the interaction terms let's check the correlation between each features



We can see that all of the features are not strongly correlated . so we shall not add any interaction terms

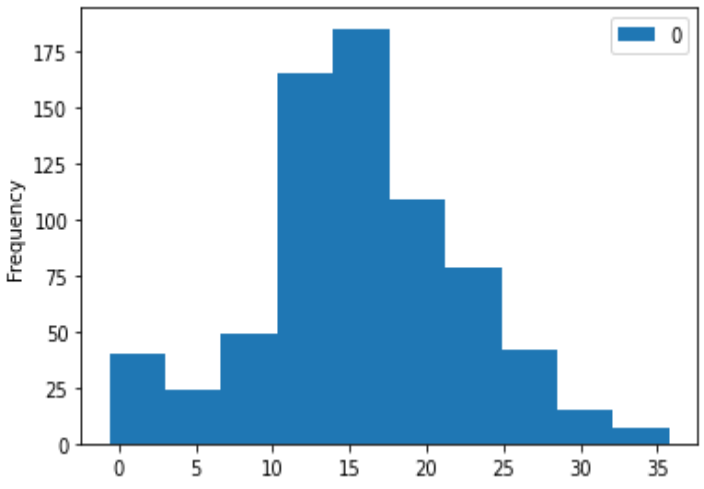
Hypothesis Testing

1)H0: Age is normally distributed
H1: Age is not normally distributed



Alpha = 0.05
Statistical test used : Normal test
P value obtained : 0.00011709599657350757
The p value is less than alpha so we reject the null hypothesis

Let's transform our age feature using box cox transformation. Distribution after transformation



Repeating the experiment

Alpha = 0.05
Statistical test used : Normal test
P value obtained : 0.5383710072962454

The p value obtained is greater than the alpha so we fail to reject the hypothesis

2)H0: male and female both have equal chance of survival
H1: Not equal chance of survival

Alpha :0.05
Statistical test used : Chi-square test
P value obtained: 1.1973570627755645e-58

P value less than alpha so we reject the null hypothesis

3)H0:there is no relationship between age and fare
H1: there is a relationship between age and fare

Alpha :0.05
Statistical test used : spearman r test
P value obtained: 0.47931972164440195

P value is greater than alpha so we fail to reject the null hypothesis

Conclusion

In this course we were able to Exploratory data analysis, plotting techniques, feature engineering, statistical analysis and was able to use those skills in this projectThe data of 891 observations and 12 features was converted to 891 observations with 23 features

Next Steps

More feature Engineering can be done for example adding the is child feature as we children and ladies were evacuated first and much more statistical test can be done with fair and survival or normal distribution check with normaltest for fare and many more