UNSUPERVISED MACHINE LEARNING

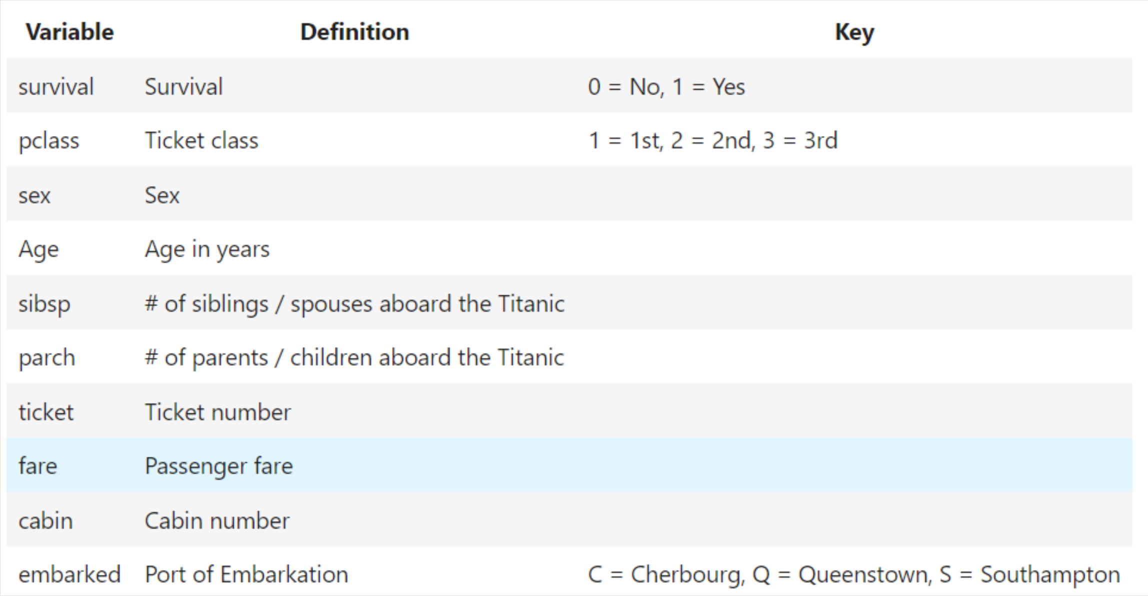
TING CHONG NA
28.05.2023

Data Description

TOPIC: Titanic Dataset

Objective: Determine if there is a relationship between survival and the different clusters

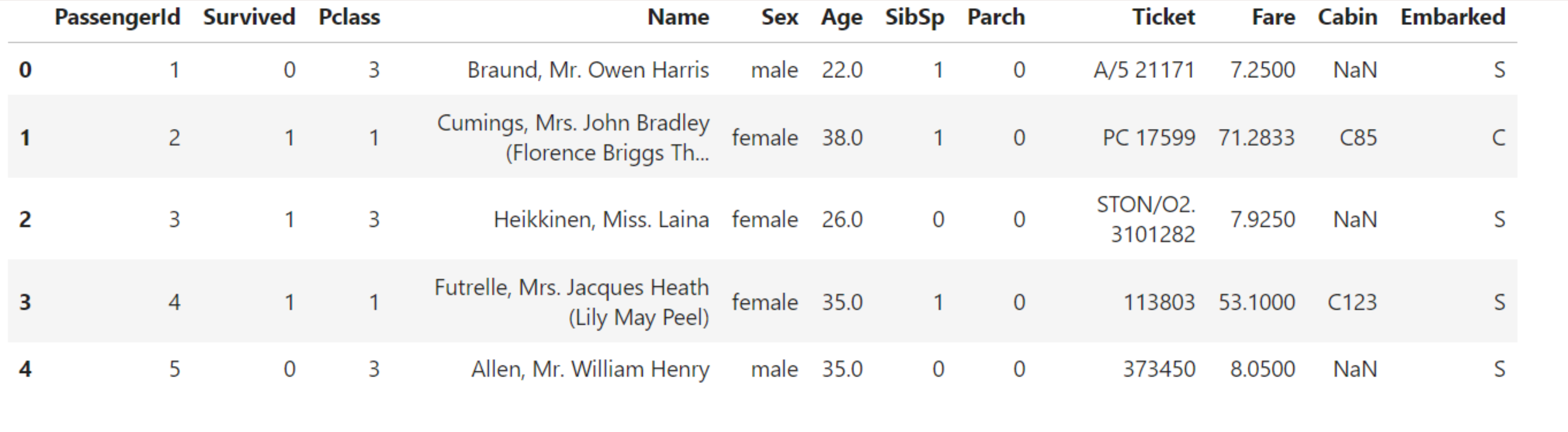
On April 15, 1912, the Titanic collided with an iceberg and sank. When the Titanic sank, it killed 1502 out of 2224 passengers and crew.



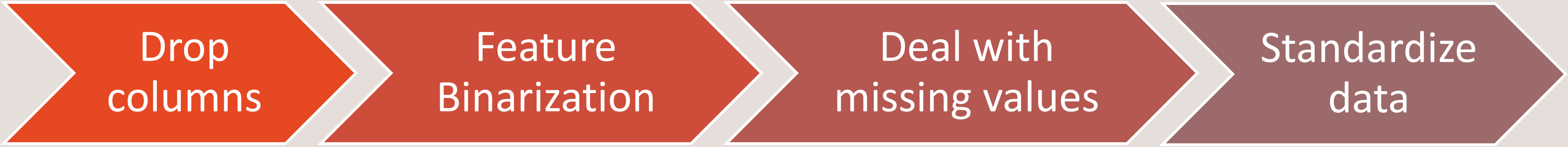
Variable	Definition	Key
survival	Survival	0 = No, 1 = Yes
pclass	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
sex	Sex	
Age	Age in years	
sibsp	# of siblings / spouses aboard the Titanic	
parch	# of parents / children aboard the Titanic	
ticket	Ticket number	
fare	Passenger fare	
cabin	Cabin number	
embarked	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

Data Preprocessing

Original Dataset



	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	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



Drop columns

Feature Binarization

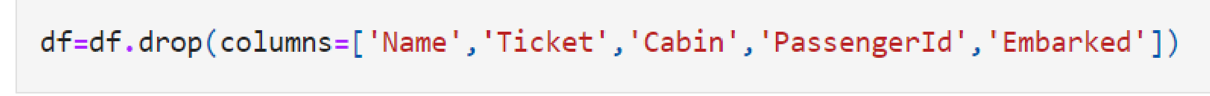
Deal with missing values

Standardize data

Data Preprocessing

DROP COLUMN

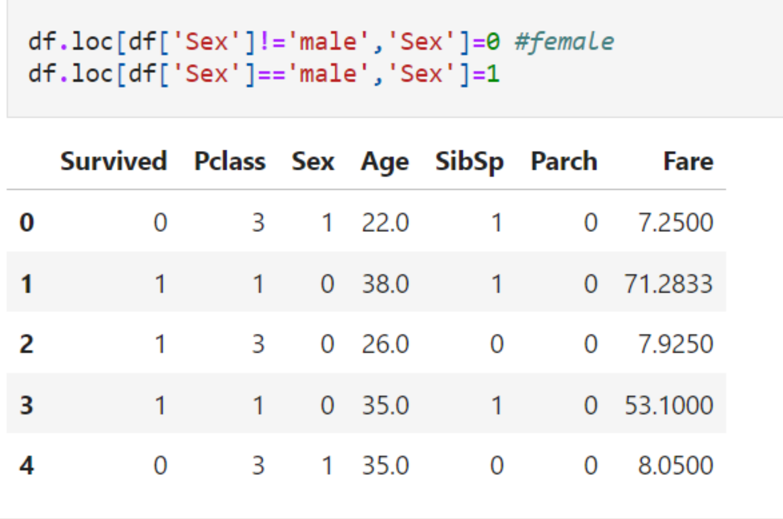
Eg: Name, Ticket, Cabin, PassengerID, Embarked



```
df=df.drop(columns=['Name','Ticket','Cabin','PassengerId','Embarked'])
```

FEATURE BINARIZATION

Assign "0" to "female" sex, and "1" to "male" sex,



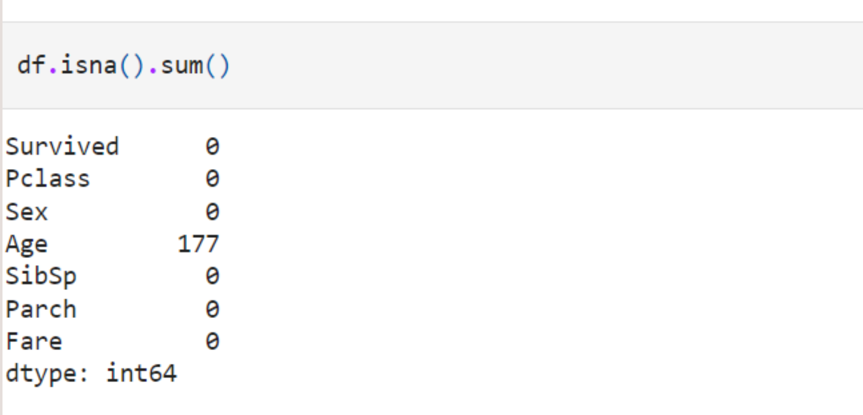
```
df.loc[df['Sex']!='male','Sex']=0 #female
df.loc[df['Sex']=='male','Sex']=1
```

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare
0	0	3	1	22.0	1	0	7.2500
1	1	1	0	38.0	1	0	71.2833
2	1	3	0	26.0	0	0	7.9250
3	1	1	0	35.0	1	0	53.1000
4	0	3	1	35.0	0	0	8.0500

Data Preprocessing

DEALING WITH MISSING VALUES

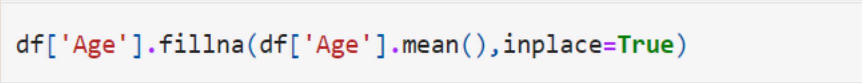
Check for missing values



```
df.isna().sum()
```

```
Survived      0
Pclass        0
Sex           0
Age         177
SibSp         0
Parch         0
Fare          0
dtype: int64
```

Replace the missing values in age, with the average age.

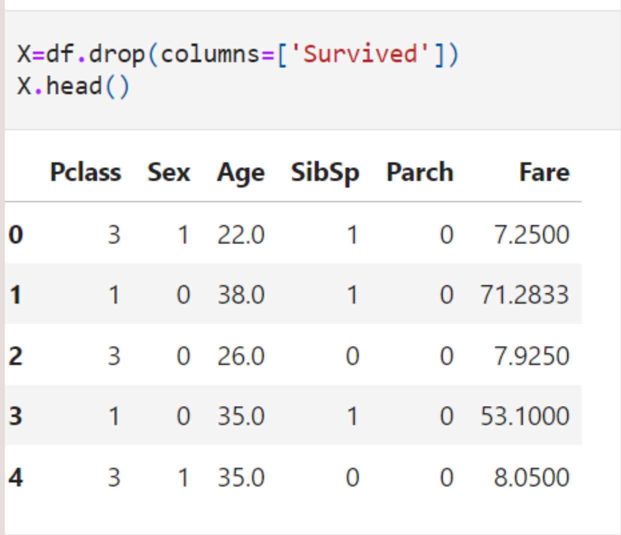


```
df['Age'].fillna(df['Age'].mean(),inplace=True)
```

Data Preprocessing

SANDARDIZE DATA X

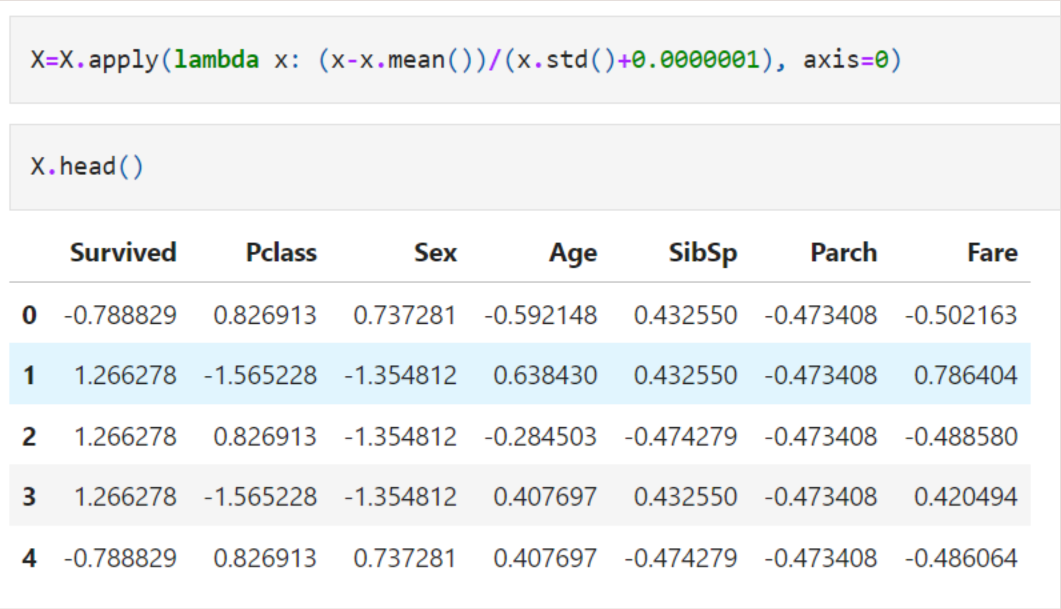
Assign the dataframe to X, for clustering, and drop our target,the Survival column.



```
X=df.drop(columns=['Survived'])
X.head()
```

	Pclass	Sex	Age	SibSp	Parch	Fare
0	3	1	22.0	1	0	7.2500
1	1	0	38.0	1	0	71.2833
2	3	0	26.0	0	0	7.9250
3	1	0	35.0	1	0	53.1000
4	3	1	35.0	0	0	8.0500

Standardize the data X



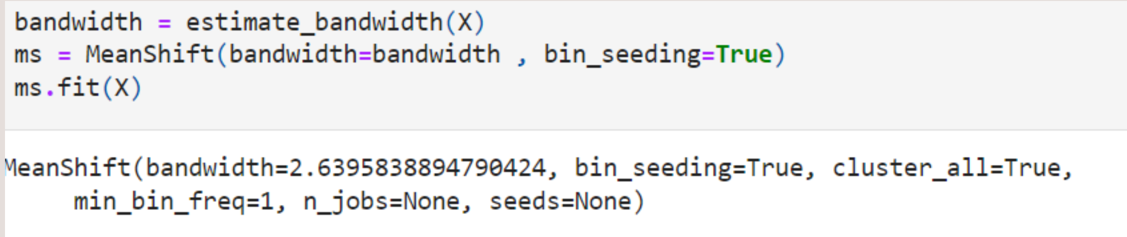
```
X=X.apply(lambda x: (x-x.mean())/(x.std()+0.0000001), axis=0)
```

```
X.head()
```

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare
0	-0.788829	0.826913	0.737281	-0.592148	0.432550	-0.473408	-0.502163
1	1.266278	-1.565228	-1.354812	0.638430	0.432550	-0.473408	0.786404
2	1.266278	0.826913	-1.354812	-0.284503	-0.474279	-0.473408	-0.488580
3	1.266278	-1.565228	-1.354812	0.407697	0.432550	-0.473408	0.420494
4	-0.788829	0.826913	0.737281	0.407697	-0.474279	-0.473408	-0.486064

Model: **Mean Shift**

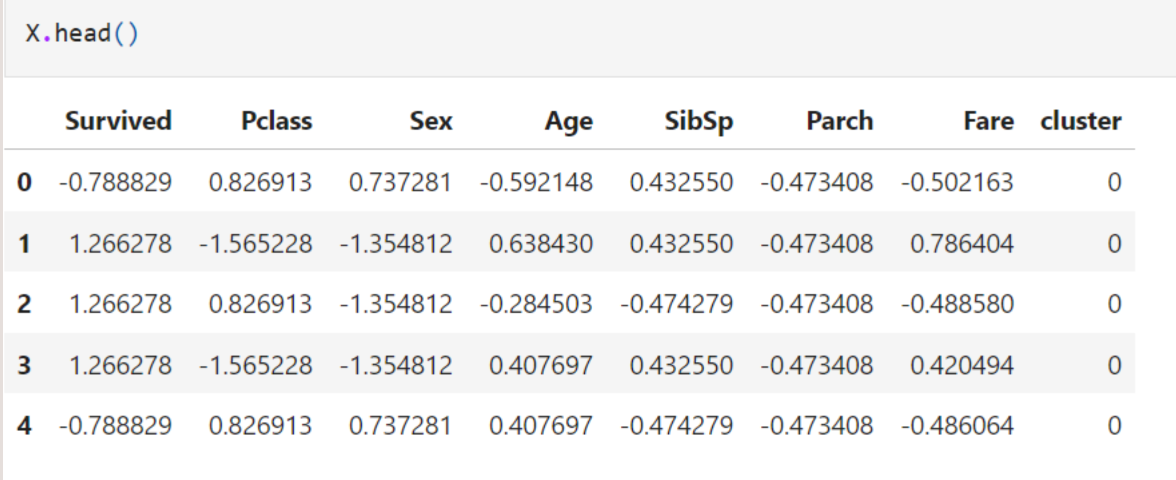
Apply the Mean-Shift algorithm to X



```
bandwidth = estimate_bandwidth(X)
ms = MeanShift(bandwidth=bandwidth , bin_seeding=True)
ms.fit(X)
```

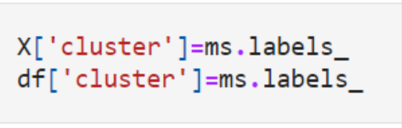
```
MeanShift(bandwidth=2.6395838894790424, bin_seeding=True, cluster_all=True,
     min_bin_freq=1, n_jobs=None, seeds=None)
```

Apply the clusters for analysis



```
X.head()
```

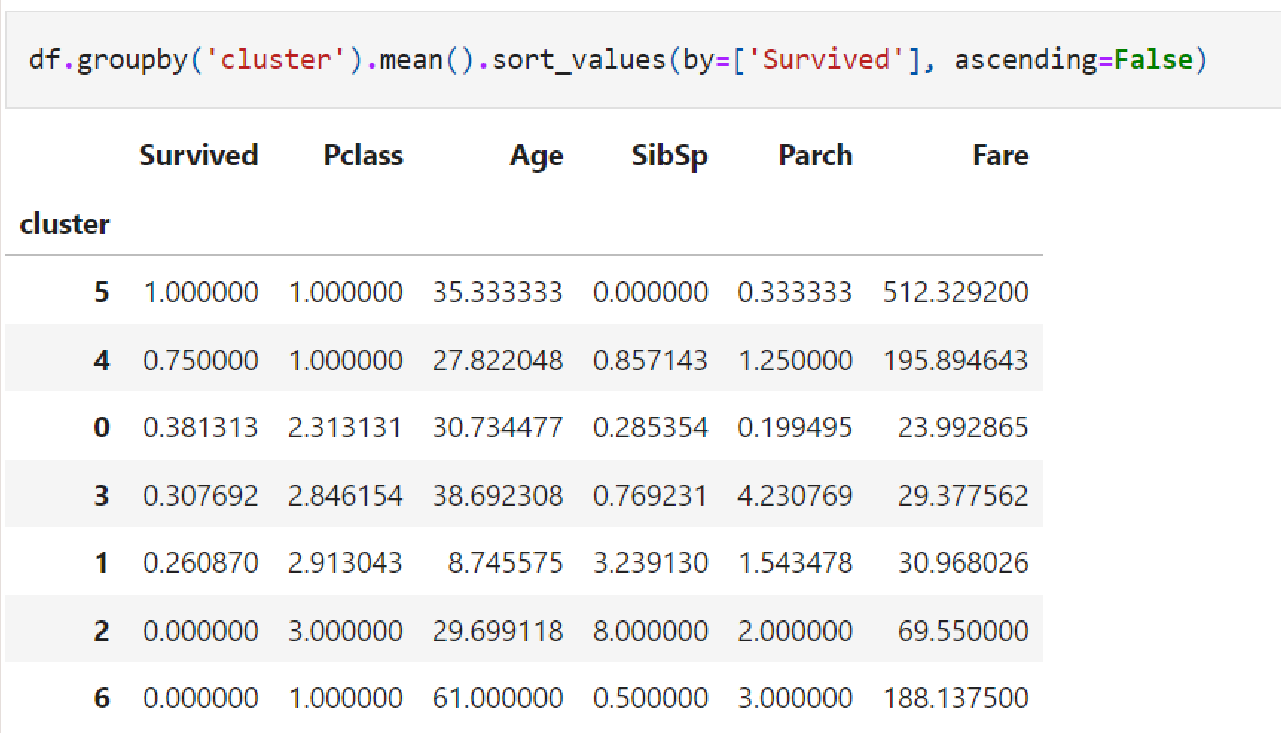
	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	cluster
0	-0.788829	0.826913	0.737281	-0.592148	0.432550	-0.473408	-0.502163	0
1	1.266278	-1.565228	-1.354812	0.638430	0.432550	-0.473408	0.786404	0
2	1.266278	0.826913	-1.354812	-0.284503	-0.474279	-0.473408	-0.488580	0
3	1.266278	-1.565228	-1.354812	0.407697	0.432550	-0.473408	0.420494	0
4	-0.788829	0.826913	0.737281	0.407697	-0.474279	-0.473408	-0.486064	0



```
X['cluster']=ms.labels_
df['cluster']=ms.labels_
```

Model: **Mean Shift**

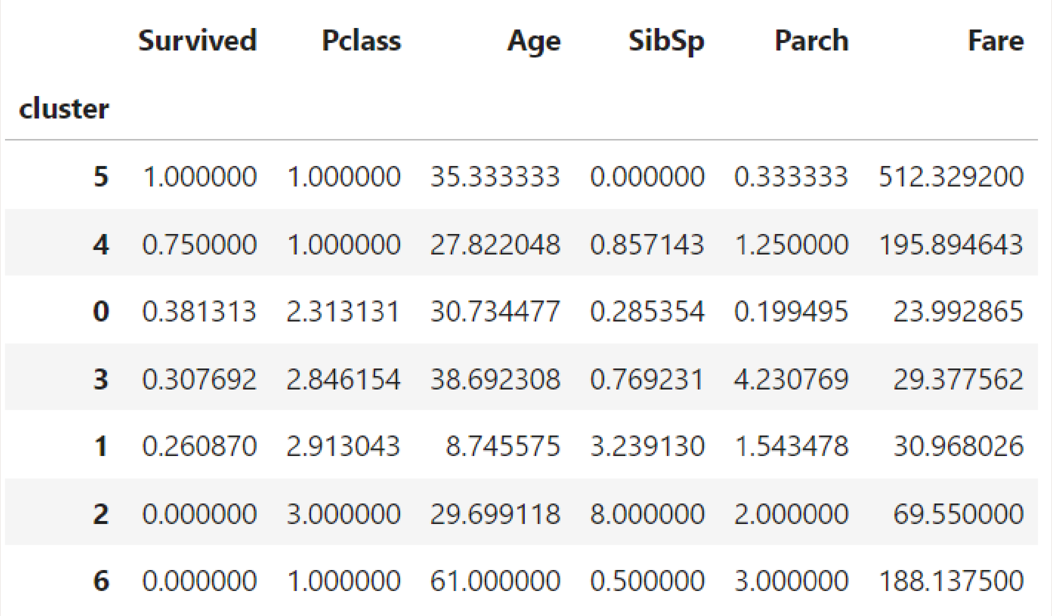
Group by clusters, to see that certain clusters have a larger chance of survival



```
df.groupby('cluster').mean().sort_values(by=['Survived'], ascending=False)
```

	Survived	Pclass	Age	SibSp	Parch	Fare
cluster						
5	1.000000	1.000000	35.333333	0.000000	0.333333	512.329200
4	0.750000	1.000000	27.822048	0.857143	1.250000	195.894643
0	0.381313	2.313131	30.734477	0.285354	0.199495	23.992865
3	0.307692	2.846154	38.692308	0.769231	4.230769	29.377562
1	0.260870	2.913043	8.745575	3.239130	1.543478	30.968026
2	0.000000	3.000000	29.699118	8.000000	2.000000	69.550000
6	0.000000	1.000000	61.000000	0.500000	3.000000	188.137500

Model: **Results**



	Survived	Pclass	Age	SibSp	Parch	Fare
cluster						
5	1.000000	1.000000	35.333333	0.000000	0.333333	512.329200
4	0.750000	1.000000	27.822048	0.857143	1.250000	195.894643
0	0.381313	2.313131	30.734477	0.285354	0.199495	23.992865
3	0.307692	2.846154	38.692308	0.769231	4.230769	29.377562
1	0.260870	2.913043	8.745575	3.239130	1.543478	30.968026
2	0.000000	3.000000	29.699118	8.000000	2.000000	69.550000
6	0.000000	1.000000	61.000000	0.500000	3.000000	188.137500

Cluster 5

100 % of survivors

- Average age of 35.3
- 1st class passengers
- Paid the highest fare (512.33 per ticket)

Cluster 6

0 % of survivors

- Average age of 61
- 1st class passengers
- Mid-range ticket fare

Conclusion

The highest odds for survival were held by the younger and richer groups of passengers.