

```
In [1]: # Welcome to my first submission on Kaggle
```

TITANIC DATA ANALYSIS

```
In [2]: # Collection of Data
        ## we start by importing all necessary modules
```

```
In [3]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import math
```

```
In [4]: train = pd.read_csv('train.csv')
train.head(5)
```

Out[4]:

	traveller_ID	ticket_class	name	sex	age	Siblings_spouses	Parchil	TickNum	f
0	1214	3	Smiljanic, Mr. Mile	male	NaN	0	0	315037	8.6
1	677	3	Bostandyeff, Mr. Guentcho	male	26.0	0	0	349224	7.8
2	534	2	Phillips, Miss. Kate Florence ("Mrs Kate Louis...	female	19.0	0	0	250655	26.0
3	1174	3	Sage, Miss. Dorothy Edith "Dolly"	female	NaN	8	2	CA. 2343	69.5
4	864	3	Henriksson, Miss. Jenny Lovisa	female	28.0	0	0	347086	7.7

```
In [5]: test = pd.read_csv('test.csv')
test.head()
```

Out[5]:

	traveller_ID	ticket_class	name	sex	age	Siblings_spouses	Parchil	TickNum	f
0	1148	3.0	Rintamaki, Mr. Matti	male	35.0	0.0	0.0	STON/O 2. 3101273	7.1:
1	1049	3.0	Nakid, Mr. Sahid	male	20.0	1.0	1.0	2653	15.7:
2	982	3.0	Lyntakoff, Mr. Stanko	male	NaN	0.0	0.0	349235	7.8:
3	808	3.0	Ford, Mr. Arthur	male	NaN	0.0	0.0	A/5 1478	8.0:
4	1195	3.0	Shaughnessy, Mr. Patrick	male	NaN	0.0	0.0	370374	7.7:

```
In [6]: sample_submission = pd.read_csv('sample_submission.csv')
sample_submission.head(5)
```

Out[6]:

	traveller_ID	Survived
0	1148	0
1	1049	1
2	982	0
3	808	1
4	1195	0

```
In [7]: ## To get total number of passengers on board

print('Number of passengers on board is, ' + str(len(train.index)))
```

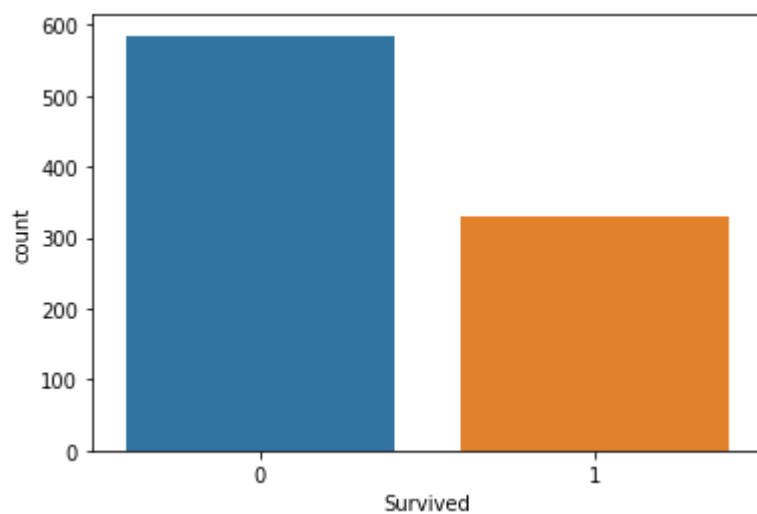
Number of passengers on board is, 916

Analysing of Data

```
In [8]: # number of those who survived
```

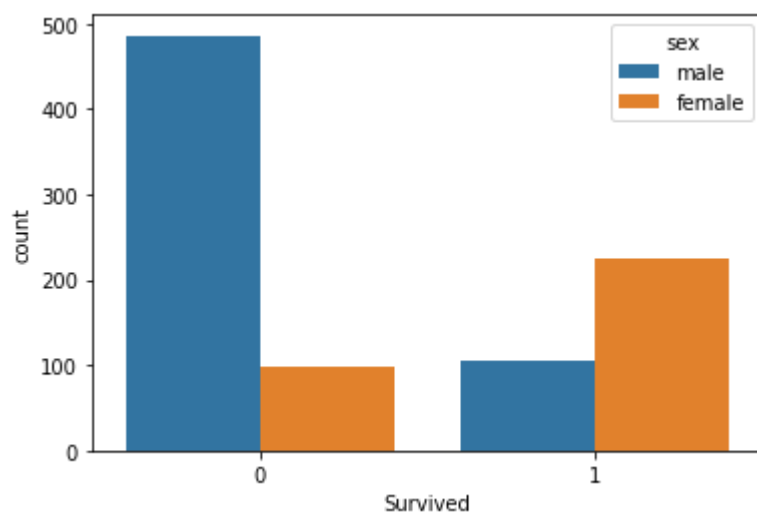
```
In [9]: sns.countplot( x="Survived", data=train)
```

```
Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0xadfae10>
```



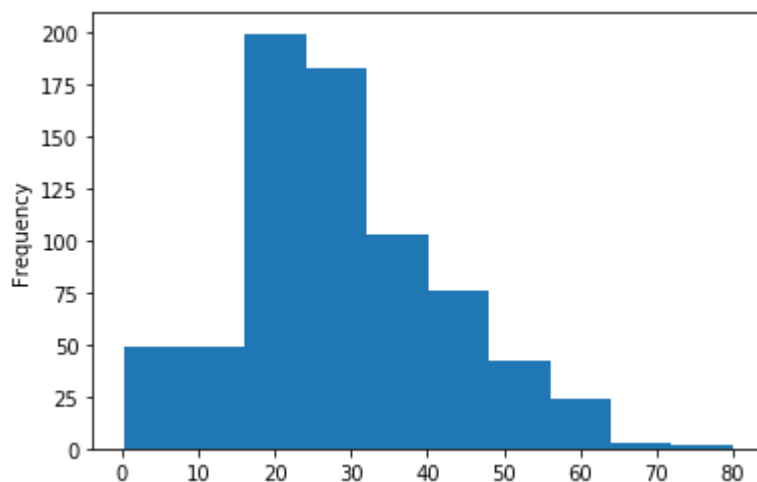
```
In [10]: # number of survivals who were males, and females  
sns.countplot(x="Survived", hue="sex", data=train)
```

```
Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0xaebae48>
```



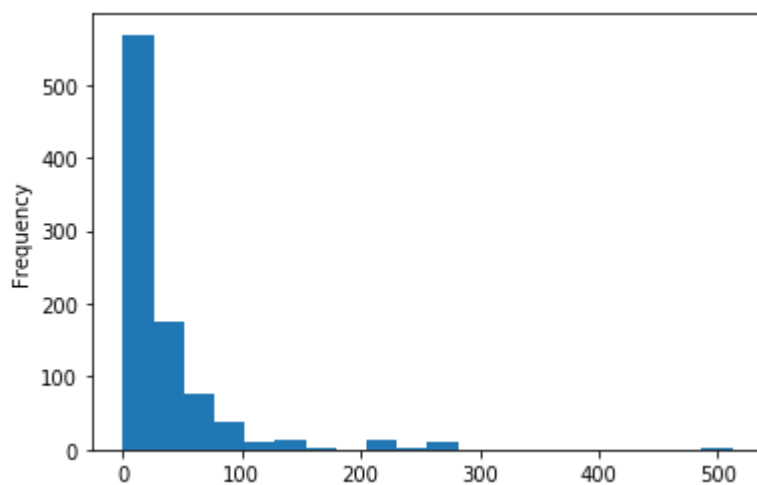
```
In [11]: #visualise age distribution  
train["age"].plot.hist()
```

Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0xaf4a4a8>



```
In [12]: # Visualise fare of travellers  
train["fare"].plot.hist(bins=20, )
```

Out[12]: <matplotlib.axes._subplots.AxesSubplot at 0xaff60f0>



In [13]: *#To obtain remaining information*
train.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 916 entries, 0 to 915
Data columns (total 13 columns):
traveller_ID      916 non-null int64
ticket_class      916 non-null int64
name              916 non-null object
sex               916 non-null object
age               729 non-null float64
Siblings_spouses  916 non-null int64
Parchil           916 non-null int64
TickNum           916 non-null object
fare              915 non-null float64
cabin            204 non-null object
embarked          915 non-null object
MedBoat           327 non-null object
Survived          916 non-null int64
dtypes: float64(2), int64(5), object(6)
memory usage: 93.1+ KB
```

In [14]: train.shape, test.shape

Out[14]: ((916, 13), (393, 12))

In [15]: train.describe()

Out[15]:

	traveller_ID	ticket_class	age	Siblings_spouses	Parchil	fare	Survived
count	916.000000	916.000000	729.000000	916.000000	916.000000	915.000000	916.000000
mean	656.248908	2.305677	29.102309	0.539301	0.386463	33.709221	0.361354
std	381.176191	0.841811	13.866954	1.082188	0.893933	52.840656	0.480294
min	0.000000	1.000000	0.166700	0.000000	0.000000	0.000000	0.000000
25%	328.250000	2.000000	21.000000	0.000000	0.000000	7.925000	0.000000
50%	670.000000	3.000000	28.000000	0.000000	0.000000	14.500000	0.000000
75%	974.250000	3.000000	37.000000	1.000000	0.000000	31.275000	1.000000
max	1308.000000	3.000000	80.000000	8.000000	9.000000	512.329200	1.000000

DATA CLEANING

In [16]: *#we proceed to cleaning of data by removal of categorical datas and replacing of missing values*

```
In [17]: train.isnull().head()
```

```
Out[17]:
```

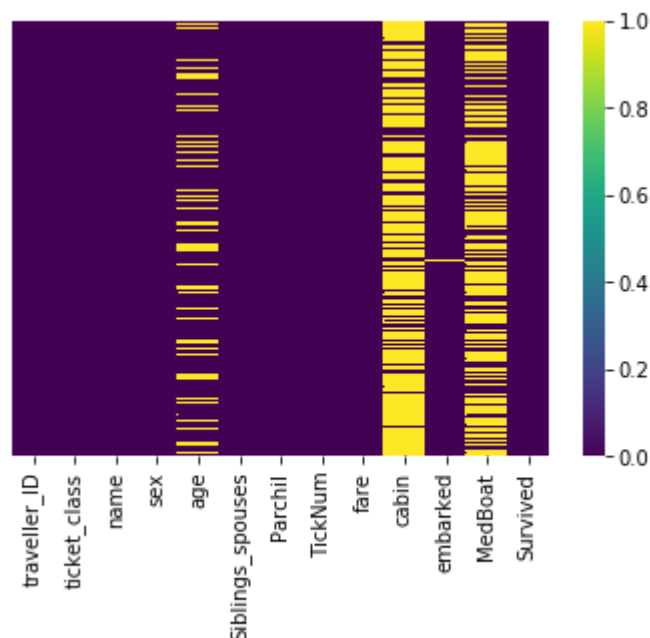
	traveller_ID	ticket_class	name	sex	age	Siblings_spouses	Parchil	TickNum	fare	cab
0	False	False	False	False	True	False	False	False	False	Tr
1	False	False	False	False	False	False	False	False	False	Tr
2	False	False	False	False	False	False	False	False	False	Tr
3	False	False	False	False	True	False	False	False	False	Tr
4	False	False	False	False	False	False	False	False	False	Tr

```
In [18]: train.isnull().sum()
```

```
Out[18]: traveller_ID      0
ticket_class      0
name              0
sex              0
age             187
Siblings_spouses  0
Parchil          0
TickNum          0
fare             1
cabin           712
embarked         1
MedBoat         589
Survived         0
dtype: int64
```

```
In [19]: sns.heatmap(train.isnull(), yticklabels = False , cmap='viridis' )
```

```
Out[19]: <matplotlib.axes._subplots.AxesSubplot at 0xb0c7828>
```



```
In [20]: def impute_age(cols):
          age = cols[0]
          ticket_class = cols[1]

          if pd.isnull(age):

              if ticket_class == 1:
                  return 37

              elif ticket_class == 2:
                  return 28

              else:
                  return 21

          else:
              return age
```

```
In [21]: train['age'] = train[['age', 'ticket_class']].apply(impute_age,axis=1 )
```

```
In [22]: train.embarked.fillna('S', inplace=True)
```

```
In [23]: train.cabin.fillna(0, inplace=True)
```

```
In [24]: train.fare.fillna(0, inplace=True)
```

```
In [25]: train.MedBoat.fillna(0, inplace=True)
```

```
In [26]: train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 916 entries, 0 to 915
Data columns (total 13 columns):
traveller_ID      916 non-null int64
ticket_class      916 non-null int64
name              916 non-null object
sex               916 non-null object
age               916 non-null float64
Siblings_spouses  916 non-null int64
Parchil           916 non-null int64
TickNum           916 non-null object
fare              916 non-null float64
cabin             916 non-null object
embarked          916 non-null object
MedBoat           916 non-null object
Survived          916 non-null int64
dtypes: float64(2), int64(5), object(6)
memory usage: 93.1+ KB
```

```
In [27]: train['cabin'].value_counts()
```



```

Out[27]: 0 712
          B57 B59 B63 B66 5
          G6 4
          C23 C25 C27 4
          C22 C26 4
          C101 3
          B58 B60 3
          F2 3
          F33 3
          A34 3
          E33 2
          C89 2
          D 2
          D21 2
          C92 2
          C65 2
          B5 2
          C6 2
          C54 2
          D17 2
          D33 2
          D35 2
          C85 2
          E50 2
          B49 2
          B20 2
          E46 2
          C123 2
          C83 2
          D36 2
          ...
          B94 1
          C45 1
          B10 1
          B38 1
          C86 1
          D22 1
          E52 1
          C106 1
          D37 1
          B61 1
          E34 1
          E38 1
          B101 1
          D46 1
          A26 1
          B79 1
          A23 1
          F E69 1
          B41 1
          C110 1
          C126 1
          D47 1
          E36 1
          A7 1
          C31 1
          C97 1

```

F G73	1
E10	1
E60	1
C99	1

Name: cabin, Length: 147, dtype: int64

```
In [28]: train['cabin'].value_counts()
```

```

Out[28]: 0 712
          B57 B59 B63 B66 5
          G6 4
          C23 C25 C27 4
          C22 C26 4
          C101 3
          B58 B60 3
          F2 3
          F33 3
          A34 3
          E33 2
          C89 2
          D 2
          D21 2
          C92 2
          C65 2
          B5 2
          C6 2
          C54 2
          D17 2
          D33 2
          D35 2
          C85 2
          E50 2
          B49 2
          B20 2
          E46 2
          C123 2
          C83 2
          D36 2
          ...
          B94 1
          C45 1
          B10 1
          B38 1
          C86 1
          D22 1
          E52 1
          C106 1
          D37 1
          B61 1
          E34 1
          E38 1
          B101 1
          D46 1
          A26 1
          B79 1
          A23 1
          F E69 1
          B41 1
          C110 1
          C126 1
          D47 1
          E36 1
          A7 1
          C31 1
          C97 1

```

```
F G73      1
E10        1
E60        1
C99        1
Name: cabin, Length: 147, dtype: int64
```

```
In [29]: train['embarked'].value_counts()
```

```
Out[29]: S      639
         C      187
         Q       90
         Name: embarked, dtype: int64
```

```
In [30]: train['TickNum'].value_counts()
```

```

Out[30]: CA. 2343          9
         PC 17608         7
         347077          6
         3101295         6
         CA 2144          5
         W./C. 6608      5
         113781          5
         349909          5
         S.O.C. 14879    5
         4133            5
         1601            5
         347082          5
         382652          5
         C.A. 33112      4
         19950           4
         24160           4
         347088          4
         2650            3
         371110          3
         2666            3
         W./C. 6607      3
         26360           3
         220845          3
         347080          3
         C.A. 37671      3
         PC 17572        3
         PC 17757        3
         110152          3
         2668            3
         12749           3
         ..
         7598            1
         315097          1
         350054          1
         345770          1
         350050          1
         315085          1
         349243          1
         2691            1
         7540            1
         112379          1
         237668          1
         384461          1
         7548            1
         SOTON/OQ 3101316 1
         17764           1
         335677          1
         368573          1
         237735          1
         315083          1
         S.C./A.4. 23567  1
         349910          1
         233639          1
         113807          1
         2223            1
         28133           1
         SCO/W 1585       1

```

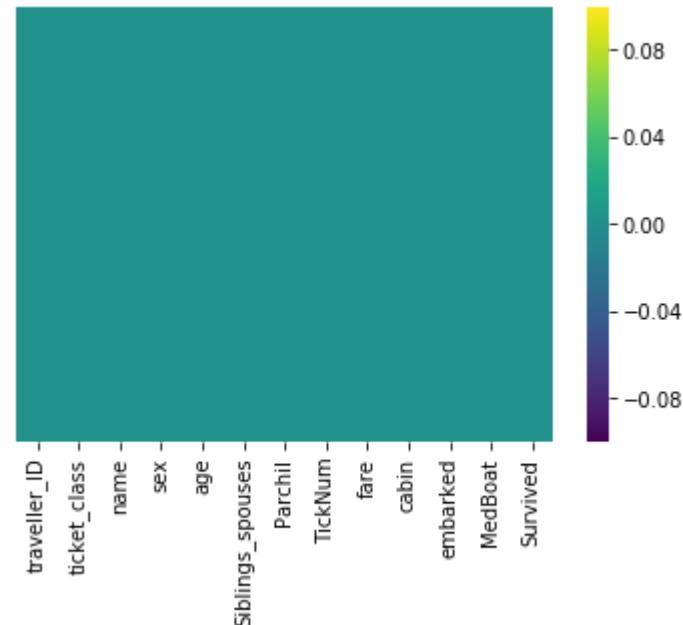
```

S.P. 3464      1
SC/AH Basle 541 1
244252        1
370377        1
Name: TickNum, Length: 694, dtype: int64

```

```
In [31]: sns.heatmap(train.isnull(), yticklabels = False , cmap='viridis' )
```

```
Out[31]: <matplotlib.axes._subplots.AxesSubplot at 0xb3dc550>
```



```
In [32]: train.head()
```

```
Out[32]:
```

	traveller_ID	ticket_class	name	sex	age	Siblings_spouses	Parchil	TickNum	f
0	1214	3	Smiljanic, Mr. Mile	male	21.0	0	0	315037	8.6i
1	677	3	Bostandyeff, Mr. Guentcho	male	26.0	0	0	349224	7.8i
2	534	2	Phillips, Miss. Kate Florence ("Mrs Kate Louis...	female	19.0	0	0	250655	26.0i
3	1174	3	Sage, Miss. Dorothy Edith "Dolly"	female	21.0	8	2	CA. 2343	69.5i
4	864	3	Henriksson, Miss. Jenny Lovisa	female	28.0	0	0	347086	7.7i


```
In [33]: train['embarked'].value_counts()
```

```
Out[33]: S      639
         C      187
         Q       90
         Name: embarked, dtype: int64
```

```
In [34]: train['sex'].value_counts()
```

```
Out[34]: male      592
         female    324
         Name: sex, dtype: int64
```

```
In [35]: Sex = pd.get_dummies(train['sex'], drop_first=True)
         Sex.head()
```

```
Out[35]:
```

	male
0	1
1	1
2	0
3	0
4	0

```
In [36]: Embarked = pd.get_dummies(train['embarked'], drop_first=True)
         Embarked.head()
```

```
Out[36]:
```

	Q	S
0	0	1
1	0	1
2	0	1
3	0	1
4	0	1

```
In [37]: Ticket_class = pd.get_dummies(train['ticket_class'], drop_first=True)
         Ticket_class.head()
```

```
Out[37]:
```

	2	3
0	0	1
1	0	1
2	1	0
3	0	1
4	0	1

In [38]: train.head()

Out[38]:

	traveller_ID	ticket_class	name	sex	age	Siblings_spouses	Parchil	TickNum	f
0	1214	3	Smiljanic, Mr. Mile	male	21.0	0	0	315037	8.6i
1	677	3	Bostandyeff, Mr. Guentcho	male	26.0	0	0	349224	7.8i
2	534	2	Phillips, Miss. Kate Florence ("Mrs Kate Louis...	female	19.0	0	0	250655	26.0i
3	1174	3	Sage, Miss. Dorothy Edith "Dolly"	female	21.0	8	2	CA. 2343	69.5i
4	864	3	Henriksson, Miss. Jenny Lovisa	female	28.0	0	0	347086	7.7i

In [39]: train_drop = train.drop(['name', 'traveller_ID', 'ticket_class', 'sex', 'TickNum', 'cabin', 'embarked'], axis=1, inplace=True)

In [40]: train.head()

Out[40]:

	age	Siblings_spouses	Parchil	fare	MedBoat	Survived
0	21.0	0	0	8.6625	0	0
1	26.0	0	0	7.8958	0	0
2	19.0	0	0	26.0000	11	1
3	21.0	8	2	69.5500	0	0
4	28.0	0	0	7.7750	0	0

In [41]: train_new = pd.concat([train, Sex, Embarked, Ticket_class,], axis=1,)

In [42]: train_new.head()

Out[42]:

	age	Siblings_spouses	Parchil	fare	MedBoat	Survived	male	Q	S	2	3
0	21.0	0	0	8.6625	0	0	1	0	1	0	1
1	26.0	0	0	7.8958	0	0	1	0	1	0	1
2	19.0	0	0	26.0000	11	1	0	0	1	1	0
3	21.0	8	2	69.5500	0	0	0	0	1	0	1
4	28.0	0	0	7.7750	0	0	0	0	1	0	1

Testing and Training of Data

```
In [43]: X = train_new.drop(['Survived'], axis=1 )  
y = train_new['Survived']
```

```
In [44]: # for i in X['MedBoat']:  
#         if type(i) == 'str':  
X['MedBoat'].replace(['A','B','C','D'], 0, inplace=True)  
  
X['MedBoat'].replace(['13 15', '8 10', '15 16', '5 7'], [14, 9, 15, 6], inplace=True)  
  
# X['MedBoat'] = X['MedBoat'].map({})
```

```
In [45]: X['MedBoat'].value_counts()
```

```
Out[45]: 0      642  
13      30  
15      25  
4       22  
10      21  
11      20  
5       20  
3       18  
14      17  
9       16  
16      14  
7       13  
8       13  
6       12  
2       12  
12      11  
1        5  
14        2  
15        1  
9         1  
6         1  
Name: MedBoat, dtype: int64
```

```
In [46]: y.head()
```

```
Out[46]: 0      0  
1      0  
2      1  
3      0  
4      0  
Name: Survived, dtype: int64
```

```
In [47]: SEED = 1000
```

```
In [48]: from sklearn.model_selection import train_test_split
```

```
In [49]: X_train, X_test, y_train, y_test = train_test_split(X,
                                                         y, test_size=0.30,
                                                         random_state=SEED)
```

```
In [50]: from catboost import CatBoostClassifier
```

```
In [51]: X_train.head()
```

Out[51]:

	age	Siblings_spouses	Parchil	fare	MedBoat	male	Q	S	2	3
520	24.0	1	0	82.2667	7	1	0	1	0	0
198	21.0	0	0	8.1125	13	1	0	1	0	1
554	25.0	1	0	17.8000	0	1	0	1	0	1
300	42.0	0	0	13.0000	0	1	0	1	1	0
863	31.0	1	0	57.0000	3	1	0	1	0	0

```
In [52]: %%time

params = {'loss_function': 'Logloss', # objective function
          'eval_metric': 'AUC', # metric
          'verbose': 1000, # output to stdout info about training process every 200 iterations
          'random_seed': SEED
        }
cbc_1 = CatBoostClassifier(**params)
cbc_1.fit(X_train, y_train, # data to train on (required parameters, unless we provide X as a pool object, will be shown below)
          eval_set=(X_test, y_test), # data to validate on
          use_best_model=True, # True if we don't want to save trees created after iteration with the best validation score
          plot=True # True for visualization of the training process (it is not shown in a published kernel - try executing this code)
        );
```

Learning rate set to 0.028392

```
0:      test: 0.9614032 best: 0.9614032 (0)      total: 49.6ms    remaining: 4
9.5s
999:    test: 0.9758446 best: 0.9779488 (56)    total: 2.44s    remaining: 0u
s
```

```
bestTest = 0.9779488066
bestIteration = 56
```

```
Shrink model to first 57 iterations.
Wall time: 3.06 s
```

Out[52]: <catboost.core.CatBoostClassifier at 0xae03400>

```
In [53]: predictions = cbc_1.predict(X_test)
```

```
In [54]: from sklearn.metrics import classification_report, confusion_matrix
```

```
In [55]: print(classification_report(y_test,predictions))
```

	precision	recall	f1-score	support
0	0.95	0.99	0.97	177
1	0.99	0.90	0.94	98
accuracy			0.96	275
macro avg	0.97	0.95	0.96	275
weighted avg	0.96	0.96	0.96	275

```
In [56]: test.head()
```

Out[56]:

	traveller_ID	ticket_class	name	sex	age	Siblings_spouses	Parchil	TickNum	f
0	1148	3.0	Rintamaki, Mr. Matti	male	35.0	0.0	0.0	STON/O 2. 3101273	7.1:
1	1049	3.0	Nakid, Mr. Sahid	male	20.0	1.0	1.0	2653	15.7:
2	982	3.0	Lyntakoff, Mr. Stanko	male	NaN	0.0	0.0	349235	7.8:
3	808	3.0	Ford, Mr. Arthur	male	NaN	0.0	0.0	A/5 1478	8.0:
4	1195	3.0	Shaughnessy, Mr. Patrick	male	NaN	0.0	0.0	370374	7.7:

```
In [57]: def impute_age(cols):
    age = cols[0]
    ticket_class = cols[1]

    if pd.isnull(age):

        if ticket_class == 1:
            return 37

        elif ticket_class == 2:
            return 28

        else:
            return 21

    else:
        return age
```

```
In [58]: test['age'] = test[['age', 'ticket_class']].apply(impute_age,axis=1 )
```

```
In [59]: test.embarked.fillna('S', inplace=True)
```

```
In [60]: test.fare.fillna(0, inplace=True)
```

```
In [61]: test.MedBoat.fillna(0, inplace=True)
```

```
In [62]: # for i in X['MedBoat']:
#         if type(i) == 'str':
test['MedBoat'].replace(['A','B','C','D', 'C D'], 0, inplace=True)

test['MedBoat'].replace(['13 15 B', '5 7', '5 9'], [1, 1, 1, ], inplace=True)

# X['MedBoat'] = X['MedBoat'].map({})
```

```
In [63]: test['MedBoat'].value_counts()
```

```
Out[63]: 0      261
14      16
15      12
7       10
8       10
4        9
16       9
9        9
13       9
12       8
6        8
3        8
10       8
5        7
11       5
1        3
2        1
Name: MedBoat, dtype: int64
```

```
In [64]: test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 393 entries, 0 to 392
Data columns (total 12 columns):
traveller_ID      393 non-null int64
ticket_class      393 non-null float64
name              393 non-null object
sex               393 non-null object
age              393 non-null float64
Siblings_spouses 393 non-null float64
Parchil           393 non-null float64
TickNum           393 non-null object
fare              393 non-null float64
cabin             91 non-null object
embarked          393 non-null object
MedBoat           393 non-null object
dtypes: float64(5), int64(1), object(6)
memory usage: 36.9+ KB
```

```
In [65]: Ticket_class2 = pd.get_dummies(test['ticket_class'], drop_first=True)
Ticket_class2.head()
```

```
Out[65]:
```

	2.0	3.0
0	0	1
1	0	1
2	0	1
3	0	1
4	0	1

```
In [66]: Sex2 = pd.get_dummies(test['sex'], drop_first=True)
Sex2.head()
```

```
Out[66]:
```

	male
0	1
1	1
2	1
3	1
4	1

```
In [67]: Embarked2 = pd.get_dummies(test['embarked'], drop_first=True)
Embarked2.head()
```

```
Out[67]:
```

	Q	S
0	0	1
1	0	0
2	0	1
3	0	1
4	1	0

```
In [68]: test_drop = test.drop(['name', 'traveller_ID', 'ticket_class', 'sex', 'TickNum',
, 'cabin', 'embarked'], axis=1, inplace=True)
```

```
In [69]: test_drop
```

```
In [70]: test_new = pd.concat([test, Sex2, Embarked2, Ticket_class2, ], axis=1, )
```

In [71]: `test_new.head(50)`

Out[71]:

	age	Siblings_spouses	Parchil	fare	MedBoat	male	Q	S	2.0	3.0
0	35.0000	0.0	0.0	7.1250	0	1	0	1	0	1
1	20.0000	1.0	1.0	15.7417	0	1	0	0	0	1
2	21.0000	0.0	0.0	7.8958	0	1	0	1	0	1
3	21.0000	0.0	0.0	8.0500	0	1	0	1	0	1
4	21.0000	0.0	0.0	7.7500	0	1	1	0	0	1
5	45.0000	0.0	0.0	26.5500	9	1	0	1	0	0
6	25.0000	0.0	0.0	7.9250	0	1	0	1	0	1
7	31.0000	0.0	0.0	13.0000	9	1	0	1	1	0
8	34.5000	0.0	0.0	7.8292	0	1	1	0	0	1
9	33.0000	1.0	0.0	53.1000	5	0	0	1	0	0
10	21.0000	8.0	2.0	69.5500	0	1	0	1	0	1
11	21.0000	2.0	0.0	21.6792	0	1	0	0	0	1
12	14.0000	1.0	0.0	11.2417	0	0	0	0	0	1
13	21.0000	0.0	0.0	7.8958	0	1	0	1	0	1
14	33.0000	0.0	0.0	7.8958	0	1	0	1	0	1
15	21.0000	0.0	0.0	7.7500	0	0	1	0	0	1
16	21.0000	0.0	0.0	7.2292	0	1	0	0	0	1
17	26.0000	0.0	0.0	7.8542	13	0	0	1	0	1
18	55.0000	0.0	0.0	135.6333	8	0	0	0	0	0
19	26.0000	1.0	2.0	20.5750	0	1	0	1	0	1
20	22.0000	0.0	0.0	9.8375	0	0	0	1	0	1
21	33.0000	0.0	0.0	5.0000	0	1	0	1	0	0
22	21.0000	0.0	0.0	7.8542	0	1	0	1	0	1
23	35.0000	0.0	0.0	26.2875	5	1	0	1	0	0
24	22.0000	0.0	0.0	7.2250	6	1	0	0	0	1
25	9.0000	1.0	1.0	15.2458	0	0	0	0	0	1
26	44.0000	0.0	0.0	13.0000	0	1	0	1	1	0
27	45.0000	0.0	2.0	30.0000	12	0	0	1	1	0
28	24.0000	0.0	0.0	7.7500	15	0	1	0	0	1
29	58.0000	0.0	0.0	146.5208	0	0	0	0	0	0
30	30.0000	0.0	0.0	12.7375	14	1	0	0	1	0
31	0.6667	1.0	1.0	14.5000	4	1	0	1	1	0
32	21.0000	2.0	0.0	23.2500	16	0	1	0	0	1
33	47.0000	1.0	0.0	61.1750	4	0	0	1	0	0
34	17.0000	0.0	0.0	73.5000	0	1	0	1	1	0

	age	Siblings_spouses	Parchil	fare	MedBoat	male	Q	S	2.0	3.0
35	28.0000	0.0	0.0	9.5000	0	1	0	1	0	1
36	37.0000	1.0	0.0	90.0000	14	0	1	0	0	0
37	36.0000	0.0	0.0	13.0000	0	1	0	1	1	0
38	39.0000	1.0	1.0	83.1583	14	0	0	0	0	0
39	37.0000	0.0	0.0	27.7208	0	1	0	0	0	0
40	32.0000	0.0	0.0	13.0000	9	0	0	1	1	0
41	19.0000	0.0	0.0	8.0500	0	1	0	1	0	1
42	42.0000	0.0	0.0	13.0000	0	0	0	1	1	0
43	26.0000	2.0	0.0	8.6625	0	1	0	1	0	1
44	16.0000	0.0	0.0	9.2167	0	1	0	1	0	1
45	21.0000	0.0	0.0	7.5500	0	1	0	1	0	1
46	32.5000	0.0	0.0	9.5000	0	1	0	1	0	1
47	18.0000	0.0	0.0	8.0500	0	0	0	1	0	1
48	27.0000	0.0	0.0	10.5000	16	0	0	1	1	0
49	22.0000	0.0	0.0	7.7500	13	0	0	1	0	1

In [72]: test_new.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 393 entries, 0 to 392
Data columns (total 10 columns):
age                393 non-null float64
Siblings_spouses   393 non-null float64
Parchil            393 non-null float64
fare               393 non-null float64
MedBoat            393 non-null object
male               393 non-null uint8
Q                  393 non-null uint8
S                  393 non-null uint8
2.0                393 non-null uint8
3.0                393 non-null uint8
dtypes: float64(4), object(1), uint8(5)
memory usage: 17.3+ KB
```

In [76]: test_new.columns

```
Out[76]: Index([      'age', 'Siblings_spouses',      'Parchil',
      'fare',      'MedBoat',      'male',
      'Q',      'S',      2.0,
      3.0],
      dtype='object')
```

In [77]: test_new.columns = ['age', 'Siblings_spouses', 'Parchil', 'fare', 'male', 'MedBoat', 'Q', 'S', 2, 3]

In [78]: `test_new.head()`

Out[78]:

	age	Siblings_spouses	Parchil	fare	male	MedBoat	Q	S	2	3
0	35.0	0.0	0.0	7.1250	0	1	0	1	0	1
1	20.0	1.0	1.0	15.7417	0	1	0	0	0	1
2	21.0	0.0	0.0	7.8958	0	1	0	1	0	1
3	21.0	0.0	0.0	8.0500	0	1	0	1	0	1
4	21.0	0.0	0.0	7.7500	0	1	1	0	0	1

In [79]: `preeditcs = cbc_1.predict(test_new)`

In [80]: `ts = pd.read_csv('test.csv')`
`ts.head()`

Out[80]:

	traveller_ID	ticket_class	name	sex	age	Siblings_spouses	Parchil	TickNum	f
0	1148	3.0	Rintamaki, Mr. Matti	male	35.0	0.0	0.0	STON/O 2. 3101273	7.1:
1	1049	3.0	Nakid, Mr. Sahid	male	20.0	1.0	1.0	2653	15.7:
2	982	3.0	Lyntakoff, Mr. Stanko	male	NaN	0.0	0.0	349235	7.8:
3	808	3.0	Ford, Mr. Arthur	male	NaN	0.0	0.0	A/5 1478	8.0:
4	1195	3.0	Shaughnessy, Mr. Patrick	male	NaN	0.0	0.0	370374	7.7:

In [81]: `predictDF = pd.DataFrame(preeditcs, columns=['Survived'])`
`submit = pd.concat([ts['traveller_ID'], predictDF], axis=1)`
`submit.head()`

`submit.to_csv('Fourth_ksubmission.csv', index=False)`