

Titanic Survival Prediction Mo... Draft saved

Share

Save Version 1

File Edit View Run Settings Add-ons Help

+ X L Run All Code ▾

Draft Session (1h:27m) H D C P U R A M

⋮

```
[92]:  
import numpy as np  
import pandas as pd  
import seaborn as sns  
import matplotlib.pyplot as plt  
from sklearn.preprocessing import MinMaxScaler, LabelEncoder, StandardScaler  
from sklearn.linear_model import LogisticRegression  
from sklearn.model_selection import train_test_split  
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix  
from sklearn.neighbors import KNeighborsClassifier  
from sklearn.tree import DecisionTreeClassifier  
from sklearn.svm import SVC
```

```
[93]:  
df = pd.read_csv('/kaggle/input/titanic-dataset/Titanic-Dataset.csv')  
df.head()
```

	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

```
[94]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
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   object    
 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(5), object(5)  
memory usage: 83.7+ KB
```

```
[95]: df.isnull().sum()
```

```
[95]:  
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
```

```
[96]: df.drop(['Ticket', 'Cabin', 'Name'], axis=1)
```

```
[97]: df.dropna(inplace = True)
df.info()

<class 'pandas.core.frame.DataFrame'>
Index: 712 entries, 0 to 890
Data columns (total 11 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   PassengerId 712 non-null    int64  
 1   Survived     712 non-null    int64  
 2   Pclass       712 non-null    int64  
 3   Name         712 non-null    object  
 4   Sex          712 non-null    object  
 5   Age          712 non-null    float64
 6   SibSp        712 non-null    int64  
 7   Parch        712 non-null    int64  
 8   Ticket       712 non-null    object  
 9   Fare          712 non-null    float64
 10  Embarked     712 non-null    object  
dtypes: float64(2), int64(5), object(4)
memory usage: 66.8+ KB
```

```
[ ]:
```

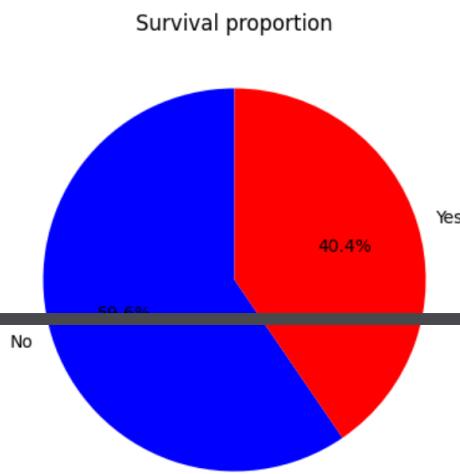
DEALING WITH SURVIVED COLUMN*

Type Markdown and LaTeX: α^2

```
[98]: survived_count = df['Survived'].value_counts()
survived_count

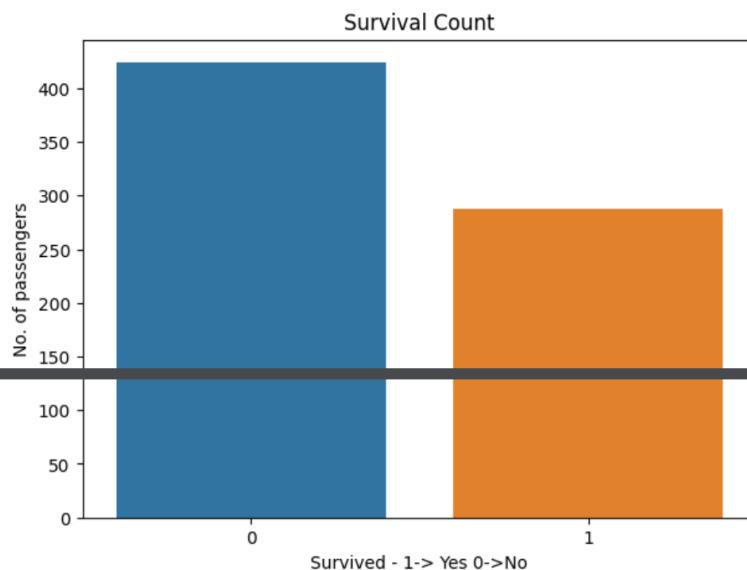
[98]: Survived
0    424
1    288
Name: count, dtype: int64
```

```
[99]: plt.figure(figsize = (7,5))
plt.pie(survived_count, labels = ["No", "Yes"], autopct = '%1.1f%%', colors = ['blue', 'red'], startangle = 90)
plt.title('Survival proportion')
plt.show()
```



```
[100]: plt.figure(figsize = (7,5))
sns.barplot(x = survived_count.index, y = survived_count.values)
plt.title('Survival Count')
```

```
plt.xlabel('Survived - 1-> Yes 0->No')
plt.ylabel('No. of passengers')
plt.show()
```



Type Markdown and LaTeX: α^2

The number of people who died was more than the survived count

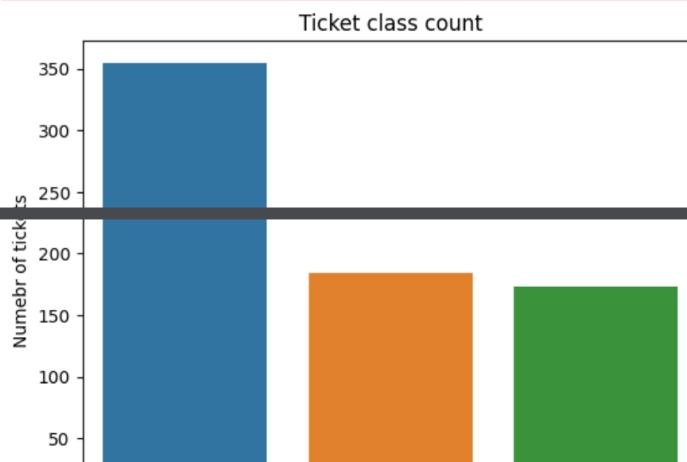
Pclass column

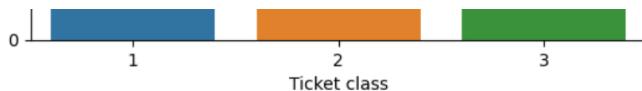
```
[101]: df['Pclass'].value_counts()
```

```
[101]: Pclass
3    355
1    184
2    173
Name: count, dtype: int64
```

```
[102]: plt.title('Ticket class count')
plt.ylabel('Number of tickets')
plt.xlabel('Ticket class')
plt.show();
```

/opt/conda/lib/python3.10/site-packages/seaborn/_oldcore.py:1765: FutureWarning: unique with argument that is not not a Series, Index, ExtensionArray, or np.ndarray is deprecated and will raise in a future version.
order = pd.unique(vector)





Type Markdown and LaTeX: α^2

Most of the tickets were First Class, followed by Second Class and Third class .

Type Markdown and LaTeX: α^2

MODEL

[103]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
Index: 712 entries, 0 to 890
Data columns (total 11 columns):
 #   Column      Non-Null Count  Dtype  
---  --  
0   PassengerId  712 non-null    int64  
1   Survived     712 non-null    int64  
2   Pclass       712 non-null    int64  
3   Name         712 non-null    object 
4   Sex          712 non-null    object 
5   Age          712 non-null    float64 
6   SibSp        712 non-null    int64  
7   Parch        712 non-null    int64  
8   Ticket       712 non-null    object 
9   Fare          712 non-null    float64 
10  Embarked     712 non-null    object 
dtypes: float64(2), int64(5), object(4)
memory usage: 66.8 KB
```

[104]: `df.corr(numeric_only = True)`

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
PassengerId	1.000000	0.029526	-0.035609	0.033681	-0.082704	-0.011672	0.009655
Survived	0.029526	1.000000	-0.356462	-0.082446	-0.015523	0.095265	0.266100
Pclass	-0.035609	-0.356462	1.000000	-0.365902	0.065187	0.023666	-0.552893
Age	0.033681	-0.082446	-0.365902	1.000000	-0.307351	-0.187896	0.093143
SibSp	-0.082704	-0.015523	0.065187	-0.307351	1.000000	0.383338	0.139860
Parch	-0.011672	0.095265	0.023666	-0.187896	0.383338	1.000000	0.206624
Fare	0.009655	0.266100	-0.552893	0.093143	0.139860	0.206624	1.000000

[105]: `df.drop(['PassengerId', 'Ticket', 'Name', 'Age'], axis = 1, inplace = True)`
`df.head()`

	Survived	Pclass	Sex	SibSp	Parch	Fare	Embarked
0	0	3	male	1	0	7.2500	S
1	1	1	female	1	0	71.2833	C
2	1	3	female	0	0	7.9250	S
3	1	1	female	1	0	53.1000	S
4	0	3	male	0	0	8.0500	S

[106]: `encoder = LabelEncoder()`
`df['Sex'] = encoder.fit_transform(df['Sex'])`
`df['Embarked'] = encoder.fit_transform(df['Embarked'])`
`df['Survived'] = encoder.fit_transform(df['Survived'])`

```
[107]:  
X = df.drop(columns = ['Survived'])  
y = df.Survived
```

X

```
[108...]  
Pclass  Sex  SibSp  Parch  Fare  Embarked  
0      3     1      1      0    7.2500      2  
1      1     0      1      0   71.2833      0  
2      3     0      0      0    7.9250      2  
3      1     0      1      0   53.1000      2  
4      3     1      0      0    8.0500      2  
...  
885     3     0      0      5   29.1250      1  
886     2     1      0      0   13.0000      2  
887     1     0      0      0   30.0000      2  
890     3     1      0      0    7.7500      1
```

712 rows × 6 columns

```
[109]:  
y
```

```
[109...]  
0      0  
1      1  
2      1  
3      1  
4      0  
..  
885     0  
886     0  
887     1  
889     1  
890     0  
Name: Survived, Length: 712, dtype: int64
```

```
[110]:  
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size = 0.5,shuffle = True,random_state = 10)
```

```
[112]:  
model = LogisticRegression(max_iter = 10000)  
model.fit(X_train,Y_train)
```

```
[112...]  
LogisticRegression  
LogisticRegression(max_iter=10000)
```

```
[113]:  
y_pred = model.predict(X_test)
```

```
[115]:  
model.score(X_train,y_train)*100
```

```
[115...]  
79.21348314606742
```

```
[116]: model.score(X_test,y_test)*100
```

```
[116... 78.37078651685393
```

```
[118]: rate = accuracy_score(y_test,y_pred)*100  
print(f'{int(rate)}%')
```

78%



+ Code

+ Markdown



es cookies from Google to deliver and enhance the quality of its services and to analyze traffic.

Learn more.

OK, Got it.