


```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```


```
from google.colab import files
uploaded=files.upload()
```

 Choose Files Titanic-Dataset.csv



- **Titanic-Dataset.csv**(text/csv) - 61194 bytes, last modified: 2/25/2025 - 100% done

Saving Titanic-Dataset.csv to Titanic-Dataset (1).csv

```
import io
titanic_df = pd.read_csv((io.BytesIO(uploaded['Titanic-Dataset (1).csv'])))
titanic_df.head(5)
```



	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




Next steps:

[Generate code with titanic_df](#)



[View recommended plots](#)

[New interactive sheet](#)

```
titanic_df.describe()
```



	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200



```
titanic_df.shape
```

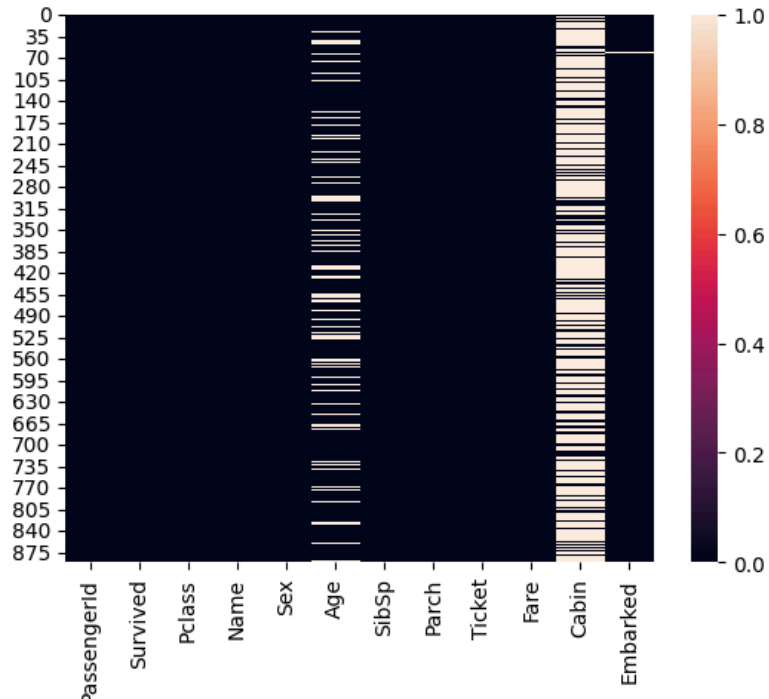
```
(891, 12)
```

```
titanic_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
```

```
sns.heatmap(titanic_df.isnull())
```

<Axes: >



```
ports = pd.get_dummies(titanic_df.Embarked, prefix='Embarked', dtype=int)
ports.head()
```

	Embarked_C	Embarked_Q	Embarked_S
0	0	0	1
1	1	0	0
2	0	0	1
3	0	0	1
4	0	0	1

Next steps:

[Generate code with ports](#)[View recommended plots](#)[New interactive sheet](#)

```
titanic_df = titanic_df.join(ports)
```

```
titanic_df.drop(['Embarked'],axis=1, inplace=True)
```

```
titanic_df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked_C	Embarked_Q	Embarked_S
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	0	0	1
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...)	female	38.0	1	0	PC 17599	71.2833	C85	1	0	0
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	0	0	1
3	4	1	1	Eutrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	0	0	1

Next steps: [Generate code with titanic_df](#) [View recommended plots](#) [New interactive sheet](#)

```
titanic_df.Sex= titanic_df.Sex.map({'male':0, 'female':1})
```

```
titanic_df['Sex']
```

	Sex
0	0
1	1
2	1
3	1
4	0
...	...
886	0
887	1
888	1
889	0
890	0

891 rows × 1 columns

```
Y = titanic_df.Survived.copy()
X = titanic_df.drop(['Survived'], axis = 1)
```

```
X.drop(['Cabin', 'Ticket', 'Name', 'PassengerId'], axis = 1, inplace=True)
```

```
X.isnull().sum()
```

```

0
Pclass    0
Sex        0
Age       177
SibSp      0
Parch      0
Fare       0
Embarked_C 0
Embarked_Q 0
Embarked_S 0

```

```
X.Age.fillna(X.Age.mean(),inplace=True)
```

<ipython-input-21-d9dd57ff4da2>:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method. The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation in-place.

```
X.Age.fillna(X.Age.mean(),inplace=True)
```

```
X.isnull().values.any()
```

```
False
```

```

from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test = train_test_split(X, Y, test_size = 0.2, random_state=7)

```

```

from sklearn.linear_model import LogisticRegression
model = LogisticRegression()

```

```
model.fit(X_train, Y_train)
```

```

/usr/local/lib/python3.11/dist-packages/sklearn/linear_model/_logistic.py:465: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

```

Increase the number of iterations (max_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
n_iter_i = _check_optimize_result(
```

▼ LogisticRegression ⓘ ?

```
LogisticRegression()
```

```
model.score(X_train,Y_train)
```

```
0.8089887640449438
```

```
Y_pred_test=model.predict(X_test)
```

```
Y_pred_test
```

```

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

```

```
Y_test
```



Survived	
725	0
861	0