

```
In [117]: import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
```

```
In [118]: titanic_data = pd.read_csv("train.csv")
titanic_data
```

```
Out[118]:
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500
...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500

891 rows × 12 columns



Just info about dataset


```
In [119]: len(titanic_data)
```

```
Out[119]: 891
```

```
In [120]: titanic_data.head()
```

Out[120]:


	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	I
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	I
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	I



```
In [121]: titanic_data.tail()
```

Out[121]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00	NaN
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	B42
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	NaN
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C14E
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN



In [122]: titanic\_data.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
```

In [123]: titanic\_data.dtypes

```
Out[123]: PassengerId      int64
Survived      int64
Pclass        int64
Name          object
Sex           object
Age           float64
SibSp         int64
Parch         int64
Ticket        object
Fare          float64
Cabin         object
Embarked      object
dtype: object
```

In [124]: titanic\_data.describe()

```
Out[124]:
```

	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

```
In [125]: titanic_data.corr()
```

```
Out[125]:
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
PassengerId	1.000000	-0.005007	-0.035144	0.036847	-0.057527	-0.001652	0.012658
Survived	-0.005007	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.257307
Pclass	-0.035144	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.549500
Age	0.036847	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.096067
SibSp	-0.057527	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.159651
Parch	-0.001652	0.081629	0.018443	-0.189119	0.414838	1.000000	0.216225
Fare	0.012658	0.257307	-0.549500	0.096067	0.159651	0.216225	1.000000

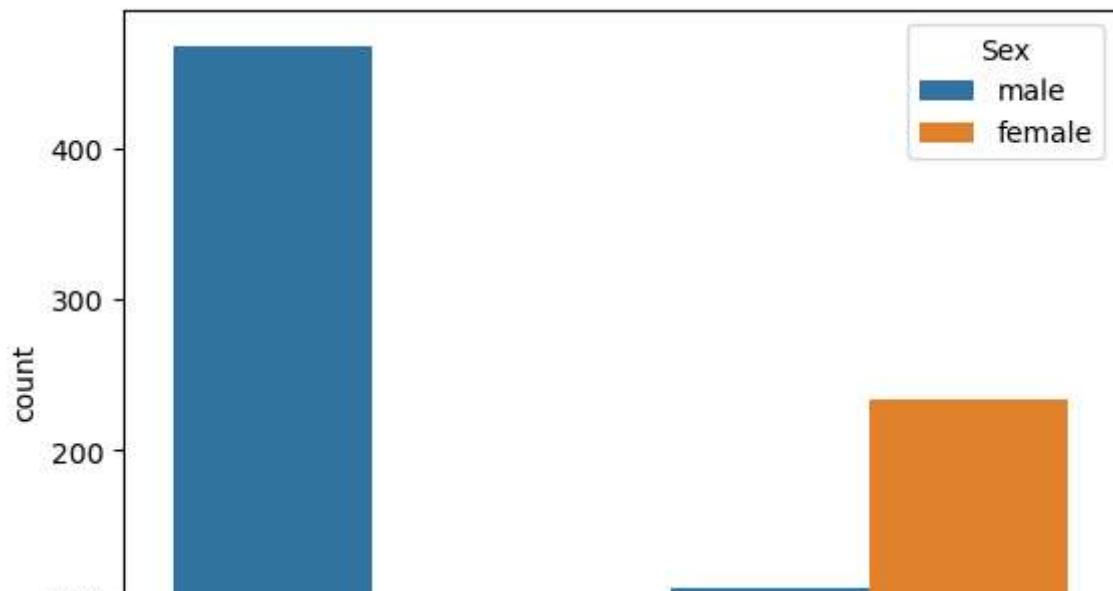
```
In [126]: titanic_data.index
```

```
Out[126]: RangeIndex(start=0, stop=891, step=1)
```

Graphing how many men and women survived

```
In [127]: sns.countplot(x = 'Survived', data = titanic_data, hue = 'Sex')
```

```
Out[127]: <AxesSubplot:xlabel='Survived', ylabel='count'>
```




```
In [128]: titanic_data.isna()
```

Out[128]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	False	False	False	False	False	False	False	False	False	False	True	
1	False	False	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	False	True	
3	False	False	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	False	True	
...	...	...	...	...	...	...	...	...	...	...	...	
886	False	False	False	False	False	False	False	False	False	False	True	
887	False	False	False	False	False	False	False	False	False	False	False	
888	False	False	False	False	False	True	False	False	False	False	True	
889	False	False	False	False	False	False	False	False	False	False	False	
890	False	False	False	False	False	False	False	False	False	False	True	

891 rows × 12 columns

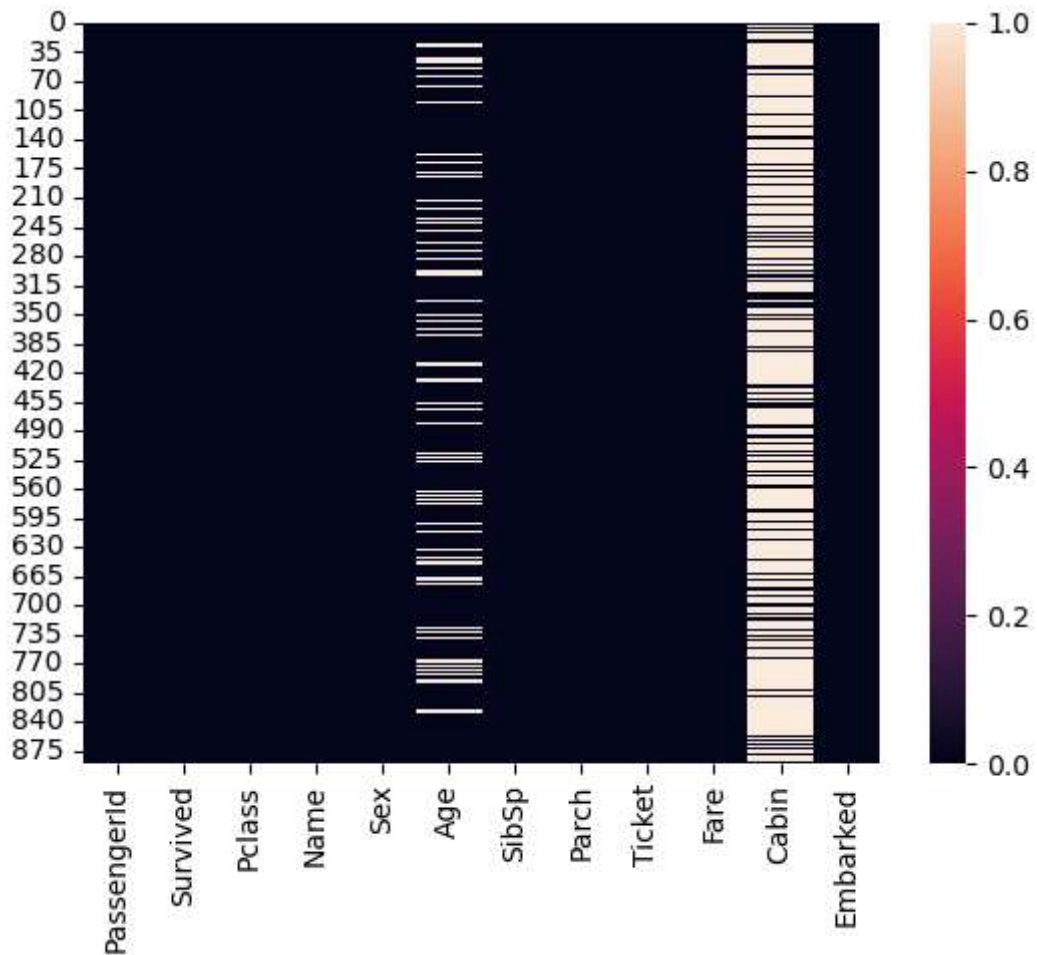


```
In [129]: titanic_data.isna().sum()
```

Out[129]: 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

```
In [130]: sns.heatmap(titanic_data.isna())
```

```
Out[130]: <AxesSubplot:>
```



```
In [131]: titanic_data['Age'].isna().sum()/len(titanic_data['Age'])*100
```

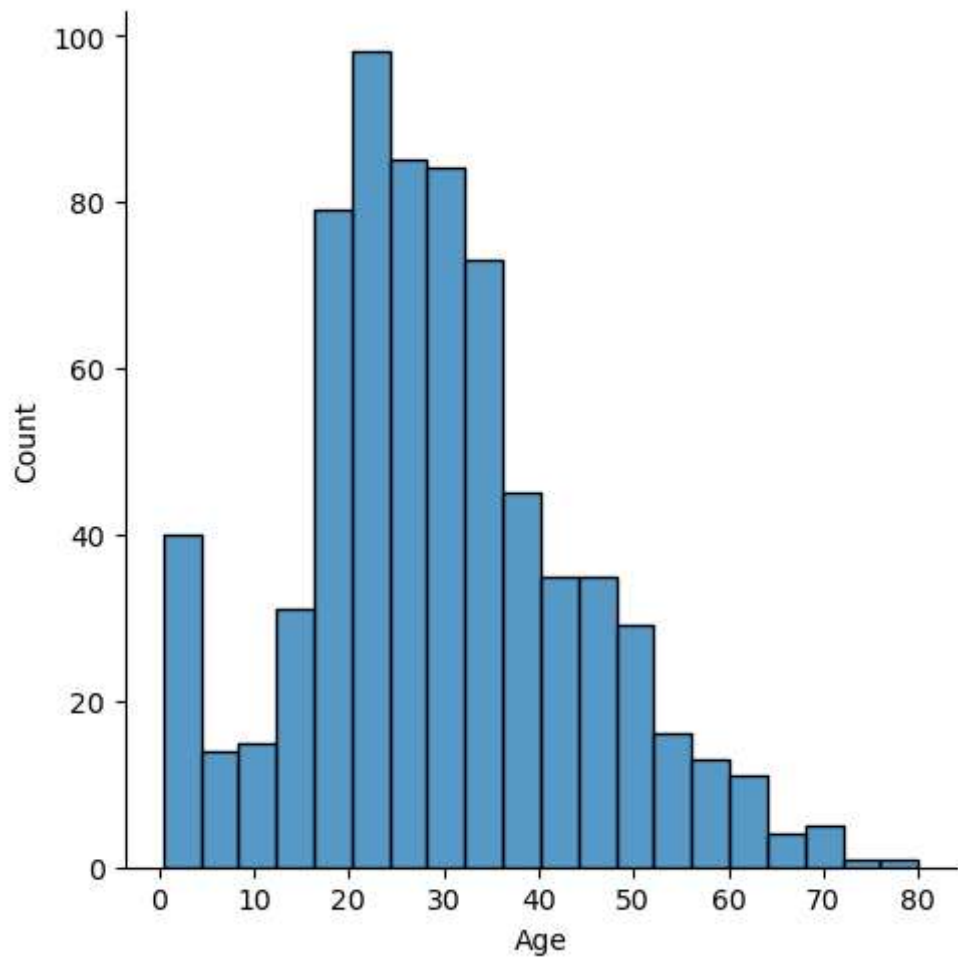
```
Out[131]: 19.865319865319865
```

```
In [132]: titanic_data['Cabin'].isna().sum()/len(titanic_data['Cabin'])*100
```

```
Out[132]: 77.10437710437711
```

```
In [133]: sns.displot(x = 'Age', data = titanic_data)
```

```
Out[133]: <seaborn.axisgrid.FacetGrid at 0x21679d060d0>
```



```
In [134]: titanic_data['Age'].mean()
```

```
Out[134]: 29.69911764705882
```

```
In [135]: titanic_data['Age'].fillna(titanic_data['Age'].mean(), inplace=True)
```

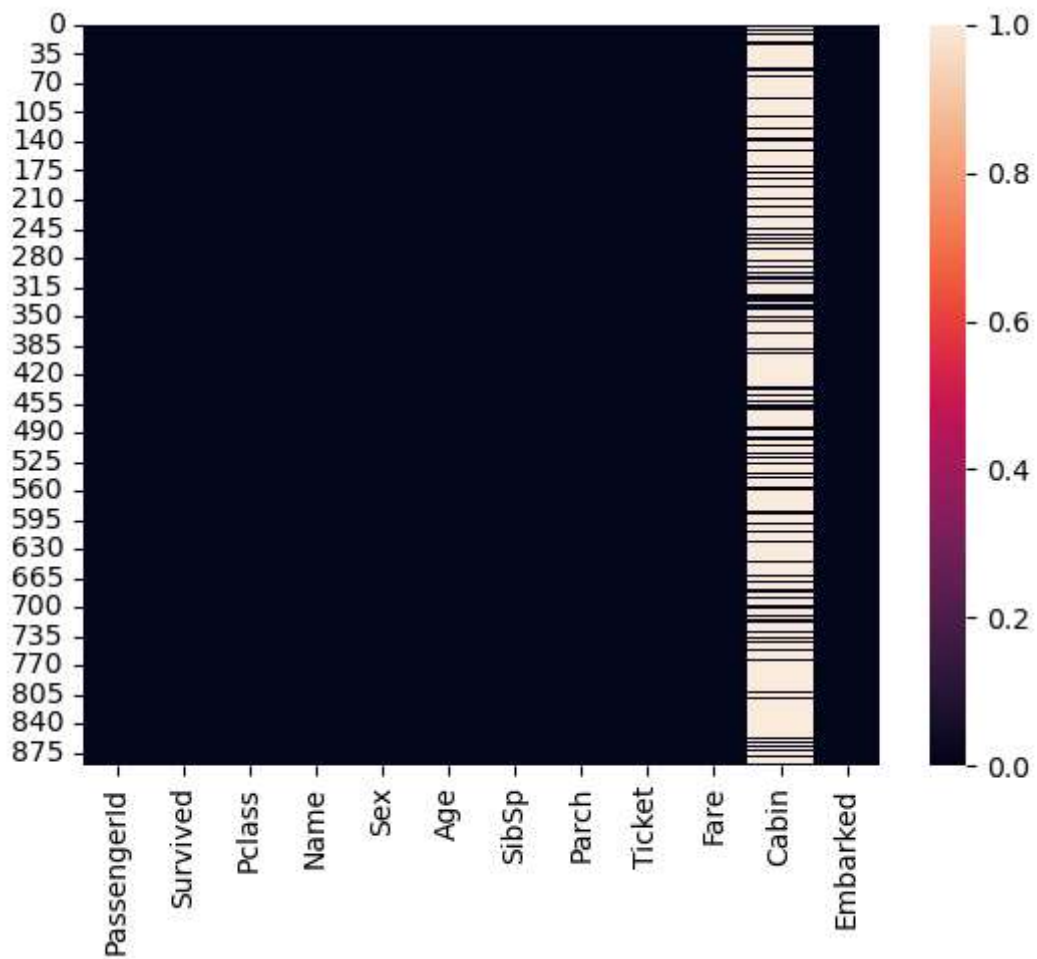
```
In [136]: titanic_data['Age'].isna().sum()
```

```
Out[136]: 0
```



```
In [137]: sns.heatmap(titanic_data.isna())
```

```
Out[137]: <AxesSubplot:>
```

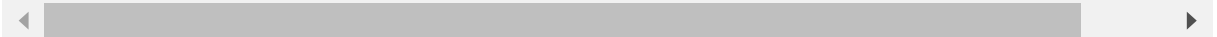


```
In [138]: titanic_data.drop('Cabin',axis=1,inplace=True)
```

In [139]: titanic\_data.head()

Out[139]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Er
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	



In [140]: titanic\_data.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 11 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          891 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  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(4)
memory usage: 76.7+ KB
```

```
In [141]: titanic_data.dtypes
```

```
Out[141]: PassengerId      int64
Survived      int64
Pclass      int64
Name      object
Sex      object
Age      float64
SibSp      int64
Parch      int64
Ticket      object
Fare      float64
Embarked      object
dtype: object
```

```
In [142]: gender = pd.get_dummies(titanic_data['Sex'],drop_first=True)
```

```
In [143]: titanic_data['Gender'] = gender
```

```
In [144]: titanic_data.head()
```

```
Out[144]:
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Er
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	

```
In [145]: titanic_data.drop(['Name', 'Sex', 'Ticket', 'Embarked'],axis=1,inplace=True)
```

```
In [146]: titanic_data.head()
```

```
Out[146]:
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare	Gender
0	1	0	3	22.0	1	0	7.2500	1
1	2	1	1	38.0	1	0	71.2833	0
2	3	1	3	26.0	0	0	7.9250	0
3	4	1	1	35.0	1	0	53.1000	0
4	5	0	3	35.0	0	0	8.0500	1

```
In [147]: x=titanic_data[['PassengerId', 'Pclass', 'Age', 'SibSp', 'Parch', 'Fare', 'Gender']]
y=titanic_data['Survived']
```

```
In [148]: x
```

```
Out[148]:
```

	PassengerId	Pclass	Age	SibSp	Parch	Fare	Gender
0	1	3	22.000000	1	0	7.2500	1
1	2	1	38.000000	1	0	71.2833	0
2	3	3	26.000000	0	0	7.9250	0
3	4	1	35.000000	1	0	53.1000	0
4	5	3	35.000000	0	0	8.0500	1
...	...	...	...	...	...	...	...
886	887	2	27.000000	0	0	13.0000	1
887	888	1	19.000000	0	0	30.0000	0
888	889	3	29.699118	1	2	23.4500	0
889	890	1	26.000000	0	0	30.0000	1
890	891	3	32.000000	0	0	7.7500	1

891 rows × 7 columns

```
In [149]: y
```

```
Out[149]:
```

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

Name: Survived, Length: 891, dtype: int64

```
In [150]: x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.33,random_
```

```
In [151]: lr=LogisticRegression()
```

```
In [152]: lr.fit(x_train,y_train)
```

```
C:\Users\Noah\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814: 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> (<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](https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression) ([https://scikit-learn.org/stable/modules/linear\\_model.html#logistic-regression](https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression))  
n\_iter\_i = \_check\_optimize\_result(

```
Out[152]: LogisticRegression()
```

```
In [153]: predict = lr.predict(x_test)
```

```
In [154]: from sklearn.metrics import confusion_matrix
```

```
In [155]: pd.DataFrame(confusion_matrix(y_test,predict), columns = ['Predicted No','Pred
```

```
Out[155]:
```

	Predicted No	Predicted Yes
--	--------------	---------------

Actual No	151	24
Actual Yes	38	82

```
In [157]: from sklearn.metrics import classification_report
```

```
In [158]: print(classification_report(y_test,predict))
```

	precision	recall	f1-score	support
0	0.80	0.86	0.83	175
1	0.77	0.68	0.73	120
accuracy			0.79	295
macro avg	0.79	0.77	0.78	295
weighted avg	0.79	0.79	0.79	295