

```
In [2]: import pandas as pd
import numpy as np
```

```
In [3]: data = pd.read_csv("C:\\Users\\amang\\Downloads\\task2\\tested.csv")
```

```
In [4]: data.head()
```

```
Out[4]:
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN

```
In [5]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 12 columns):
#   Column          Non-Null Count  Dtype
---  -
0   PassengerId      418 non-null    int64
1   Survived         418 non-null    int64
2   Pclass           418 non-null    int64
3   Name             418 non-null    object
4   Sex              418 non-null    object
5   Age              332 non-null    float64
6   SibSp            418 non-null    int64
7   Parch            418 non-null    int64
8   Ticket           418 non-null    object
9   Fare             417 non-null    float64
10  Cabin            91 non-null     object
11  Embarked         418 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 39.3+ KB
```

In [6]: `data.isnull().sum()`

```
Out[6]: PassengerId      0
Survived      0
Pclass        0
Name          0
Sex           0
Age           86
SibSp         0
Parch         0
Ticket        0
Fare          1
Cabin        327
Embarked      0
dtype: int64
```

In [7]: `data = data.drop(["PassengerId", "Ticket", "Cabin", "Name"], axis=1)`
`data.head()`

```
Out[7]:
```

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	3	male	34.5	0	0	7.8292	Q
1	1	3	female	47.0	1	0	7.0000	S
2	0	2	male	62.0	0	0	9.6875	Q
3	0	3	male	27.0	0	0	8.6625	S
4	1	3	female	22.0	1	1	12.2875	S

In [8]: `data.describe()`

```
Out[8]:
```

	Survived	Pclass	Age	SibSp	Parch	Fare
count	418.000000	418.000000	332.000000	418.000000	418.000000	417.000000
mean	0.363636	2.265550	30.272590	0.447368	0.392344	35.627188
std	0.481622	0.841838	14.181209	0.896760	0.981429	55.907576
min	0.000000	1.000000	0.170000	0.000000	0.000000	0.000000
25%	0.000000	1.000000	21.000000	0.000000	0.000000	7.895800
50%	0.000000	3.000000	27.000000	0.000000	0.000000	14.454200
75%	1.000000	3.000000	39.000000	1.000000	0.000000	31.500000
max	1.000000	3.000000	76.000000	8.000000	9.000000	512.329200

```
In [9]: data.isnull().sum()
```

```
Out[9]: Survived      0
Pclass      0
Sex         0
Age        86
SibSp      0
Parch      0
Fare       1
Embarked    0
dtype: int64
```

```
In [10]: data = data.drop(["Fare"],axis=1)
```

```
In [23]: mean = data["Age"].mean()
data["Age"] = data["Age"].replace(np.nan, mean)
```

```
In [18]: print(data.dtypes)
```

```
Survived      int64
Pclass        int64
Sex           object
Age          float64
SibSp         int64
Parch         int64
Embarked      object
dtype: object
```

```
In [25]: data["Age"] = data["Age"].astype(int)
```

```
In [26]: data.head()
```

```
Out[26]:
```

	Survived	Pclass	Sex	Age	SibSp	Parch	Embarked
0	0	3	male	34	0	0	Q
1	1	3	female	47	1	0	S
2	0	2	male	62	0	0	Q
3	0	3	male	27	0	0	S
4	1	3	female	22	1	1	S

```
In [27]: data["Embarked"].unique()
```

```
Out[27]: array(['Q', 'S', 'C'], dtype=object)
```

```
In [28]: data = pd.get_dummies(data, columns=['Sex'], drop_first=True)
```

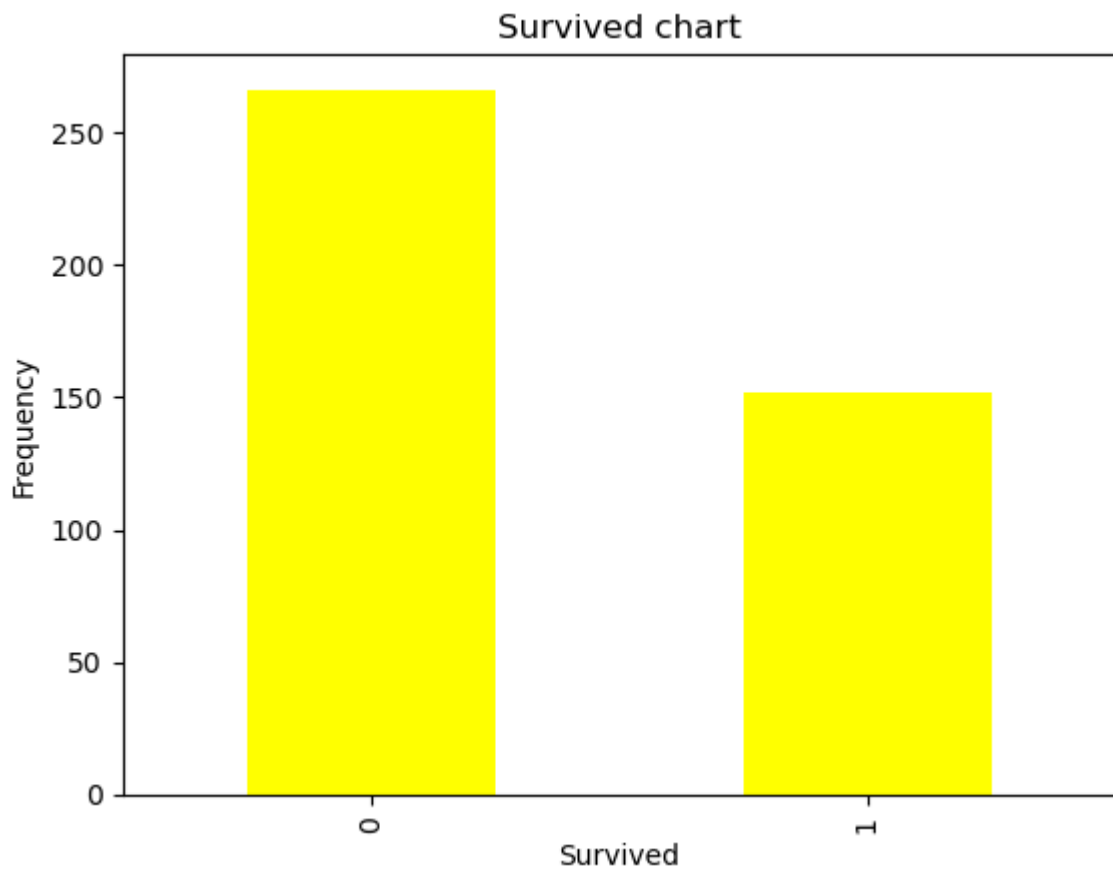
```
In [29]: data.head()
```

```
Out[29]:
```

	Survived	Pclass	Age	SibSp	Parch	Embarked	Sex_male
0	0	3	34	0	0	Q	1
1	1	3	47	1	0	S	0
2	0	2	62	0	0	Q	1
3	0	3	27	0	0	S	1
4	1	3	22	1	1	S	0

```
In [30]: import matplotlib.pyplot as plt
```

```
In [31]: data['Survived'].value_counts().plot(kind='bar',color='yellow')  
  
plt.xlabel("Survived")  
plt.ylabel("Frequency")  
plt.title("Survived chart")  
plt.show()
```



```
In [35]: from sklearn.model_selection import train_test_split
from sklearn.impute import SimpleImputer
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
```

```
In [40]: X = data.drop(["Survived", "Embarked"], axis=1)
y = data["Survived"]
```

```
In [41]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
In [42]: model = LogisticRegression()
```

```
In [43]: model.fit(X_train, y_train)
```

```
Out[43]: LogisticRegression()
```

```
In [45]: y_prediction = model.predict(X_test)
Accuracy = accuracy_score(y_test, y_prediction)
print("Accuracy:", Accuracy)
```

Accuracy: 1.0