

Titanic Survival Prediction (Classification)

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [2]: df = pd.read_csv('titanic.csv')
```

```
In [3]: df.head()
```

```
Out[3]:
```

	PassengerId	Age	Fare	Sex	sibsp	zero	zero.1	zero.2	zero.3	zero.4	...	zero.12	zero.13	zero.14	Pclass	zero.15
0	1	22.0	7.2500	0	1	0	0	0	0	0	...	0	0	0	3	0
1	2	38.0	71.2833	1	1	0	0	0	0	0	...	0	0	0	1	0
2	3	26.0	7.9250	1	0	0	0	0	0	0	...	0	0	0	3	0
3	4	35.0	53.1000	1	1	0	0	0	0	0	...	0	0	0	1	0
4	5	35.0	8.0500	0	0	0	0	0	0	0	...	0	0	0	3	0

5 rows × 28 columns

```
In [4]: df.shape
```

```
Out[4]: (1309, 28)
```

```
In [5]: df.dropna()
```

```
Out[5]:
```

	PassengerId	Age	Fare	Sex	sibsp	zero	zero.1	zero.2	zero.3	zero.4	...	zero.12	zero.13	zero.14	Pclass	zero.15
0	1	22.0	7.2500	0	1	0	0	0	0	0	...	0	0	0	3	0
1	2	38.0	71.2833	1	1	0	0	0	0	0	...	0	0	0	1	0
2	3	26.0	7.9250	1	0	0	0	0	0	0	...	0	0	0	3	0
3	4	35.0	53.1000	1	1	0	0	0	0	0	...	0	0	0	1	0
4	5	35.0	8.0500	0	0	0	0	0	0	0	...	0	0	0	3	0
5	6	28.0	8.4583	0	0	0	0	0	0	0	...	0	0	0	3	0
6	7	54.0	51.8625	0	0	0	0	0	0	0	...	0	0	0	1	0
7	8	2.0	21.0750	0	3	0	0	0	0	0	...	0	0	0	3	0
8	9	27.0	11.1333	1	0	0	0	0	0	0	...	0	0	0	3	0
9	10	14.0	30.0708	1	1	0	0	0	0	0	...	0	0	0	2	0
10	11	4.0	16.7000	1	1	0	0	0	0	0	...	0	0	0	3	0
11	12	58.0	26.5500	1	0	0	0	0	0	0	...	0	0	0	1	0
12	13	20.0	8.0500	0	0	0	0	0	0	0	...	0	0	0	3	0
13	14	39.0	31.2750	0	1	0	0	0	0	0	...	0	0	0	3	0
14	15	14.0	7.8542	1	0	0	0	0	0	0	...	0	0	0	3	0
15	16	55.0	16.0000	1	0	0	0	0	0	0	...	0	0	0	2	0
16	17	2.0	29.1250	0	4	0	0	0	0	0	...	0	0	0	3	0
17	18	28.0	13.0000	0	0	0	0	0	0	0	...	0	0	0	2	0
18	19	31.0	18.0000	1	1	0	0	0	0	0	...	0	0	0	3	0
19	20	28.0	7.2250	1	0	0	0	0	0	0	...	0	0	0	3	0
20	21	35.0	26.0000	0	0	0	0	0	0	0	...	0	0	0	2	0
21	22	34.0	13.0000	0	0	0	0	0	0	0	...	0	0	0	2	0
22	23	15.0	8.0292	1	0	0	0	0	0	0	...	0	0	0	3	0
23	24	28.0	35.5000	0	0	0	0	0	0	0	...	0	0	0	1	0
24	25	8.0	21.0750	1	3	0	0	0	0	0	...	0	0	0	3	0
25	26	38.0	31.3875	1	1	0	0	0	0	0	...	0	0	0	3	0
26	27	28.0	7.2250	0	0	0	0	0	0	0	...	0	0	0	3	0
27	28	19.0	263.0000	0	3	0	0	0	0	0	...	0	0	0	1	0
28	29	28.0	7.8792	1	0	0	0	0	0	0	...	0	0	0	3	0
29	30	28.0	7.8958	0	0	0	0	0	0	0	...	0	0	0	3	0
...
1279	1280	21.0	7.7500	0	0	0	0	0	0	0	...	0	0	0	3	0
1280	1281	6.0	21.0750	0	3	0	0	0	0	0	...	0	0	0	3	0
1281	1282	23.0	93.5000	0	0	0	0	0	0	0	...	0	0	0	1	0
1282	1283	51.0	39.4000	1	0	0	0	0	0	0	...	0	0	0	1	0
1283	1284	13.0	20.2500	0	0	0	0	0	0	0	...	0	0	0	3	0
1284	1285	47.0	10.5000	0	0	0	0	0	0	0	...	0	0	0	2	0
1285	1286	29.0	22.0250	0	3	0	0	0	0	0	...	0	0	0	3	0
1286	1287	18.0	60.0000	1	1	0	0	0	0	0	...	0	0	0	1	0
1287	1288	24.0	7.2500	0	0	0	0	0	0	0	...	0	0	0	3	0
1288	1289	48.0	79.2000	1	1	0	0	0	0	0	...	0	0	0	1	0
1289	1290	22.0	7.7750	0	0	0	0	0	0	0	...	0	0	0	3	0
1290	1291	31.0	7.7333	0	0	0	0	0	0	0	...	0	0	0	3	0
1291	1292	30.0	164.8667	1	0	0	0	0	0	0	...	0	0	0	1	0
1292	1293	38.0	21.0000	0	1	0	0	0	0	0	...	0	0	0	2	0
1293	1294	22.0	59.4000	1	0	0	0	0	0	0	...	0	0	0	1	0
1294	1295	17.0	47.1000	0	0	0	0	0	0	0	...	0	0	0	1	0
1295	1296	43.0	27.7208	0	1	0	0	0	0	0	...	0	0	0	1	0
1296	1297	20.0	13.8625	0	0	0	0	0	0	0	...	0	0	0	2	0
1297	1298	23.0	10.5000	0	1	0	0	0	0	0	...	0	0	0	2	0
1298	1299	50.0	211.5000	0	1	0	0	0	0	0	...	0	0	0	1	0
1299	1300	28.0	7.7208	1	0	0	0	0	0	0	...	0	0	0	3	0
1300	1301	3.0	13.7750	1	1	0	0	0	0	0	...	0	0	0	3	0
1301	1302	28.0	7.7500	1	0	0	0	0	0	0	...	0	0	0	3	0
1302	1303	37.0	90.0000	1	1	0	0	0	0	0	...	0	0	0	1	0
1303	1304	28.0	7.7750	1	0	0	0	0	0	0	...	0	0	0	3	0
1304	1305	28.0	8.0500	0	0	0	0	0	0	0	...	0	0	0	3	0
1305	1306	39.0	108.9000	1	0	0	0	0	0	0	...	0	0	0	1	0
1306	1307	38.5	7.2500	0	0	0	0	0	0	0	...	0	0	0	3	0
1307	1308	28.0	8.0500	0	0	0	0	0	0	0	...	0	0	0	3	0
1308	1309	28.0	22.3583	0	1	0	0	0	0	0	...	0	0	0	3	0

1307 rows × 28 columns

```
In [6]: df = df.drop(['zero', 'zero.1', 'zero.2', 'zero.3', 'zero.4', 'zero.5', 'zero.6', 'zero.7', 'zero.8', 'zero.9', 'zero.10', 'zero.11', 'zero.1', 'zero.1', 'zero.1', 'zero.1', 'zero.1', 'zero.1', 'zero.1', 'zero.12', 'zero.13', 'zero.14', 'zero.15', 'zero.16', 'zero.17', 'zero.18'], axis = 1)
```

```
In [7]: df.describe()
```

```
Out[7]:
```

	PassengerId	Age	Fare	Sex	sibsp	Parch	Pclass	Embarked	Survived
count	1309.000000	1309.000000	1309.000000	1309.000000	1309.000000	1309.000000	1309.000000	1307.000000	1309.000
mean	655.000000	29.503186	33.281086	0.355997	0.498854	0.385027	2.294882	1.492731	0.261
std	378.02061	12.905241	51.741500	0.478997	1.041658	0.865560	0.837836	0.814626	0.438
min	1.000000	0.170000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000
25%	328.000000	22.000000	7.895800	0.000000	0.000000	0.000000	2.000000	1.000000	0.000
50%	655.000000	28.000000	14.454200	0.000000	0.000000	0.000000	3.000000	2.000000	0.000
75%	982.000000	35.000000	31.275000	1.000000	1.000000	0.000000	3.000000	2.000000	1.000
max	1309.000000	80.000000	512.329200	1.000000	8.000000	9.000000	3.000000	2.000000	1.000

```
In [8]: df.head(10)
```

```
Out[8]:
```

	PassengerId	Age	Fare	Sex	sibsp	Parch	Pclass	Embarked	Survived
0	1	22.0	7.2500	0	1	0	3	2.0	0
1	2	38.0	71.2833	1	1	0	1	0.0	1
2	3	26.0	7.9250	1	0	0	3	2.0	1
3	4	35.0	53.1000	1	1	0	1	2.0	1
4	5	35.0	8.0500	0	0	0	3	2.0	0
5	6	28.0	8.4583	0	0	0	3	1.0	0
6	7	54.0	51.8625	0	0	0	1	2.0	0
7	8	2.0	21.0750	0	3	1	3	2.0	0
8	9	27.0	11.1333	1	0	2	3	2.0	1
9	10	14.0	30.0708	1	1	0	2	0.0	1

Embarked : 0 = Cherbourg, 1 = Queenstown, 2 = Southampton

Pclass : 1 = 1st, 2 = 2nd, 3 = 3rd

Survived : 0 = No, 1 = Yes

sibsp : no. of siblings / spouses aboard the Titanic

Parch : no. of parents / children aboard the Titanic

```
In [9]: from sklearn.preprocessing import LabelEncoder
```

```
In [10]: X = df.iloc[:,1:-1]
X
```

```
Out[10]:
```

	Age	Fare	Sex	sibsp	Parch	Pclass	Embarked
0	22.0	7.2500	0	1	0	3	2.0
1	38.0	71.2833	1	1	0	1	0.0
2	26.0	7.9250	1	0	0	3	2.0
3	35.0	53.1000	1	1	0	1	2.0
4	35.0	8.0500	0	0	0	3	2.0
5	28.0	8.4583	0	0	0	3	1.0
6	54.0	51.8625	0	0	0	1	2.0
7	2.0	21.0750	0	3	1	3	2.0
8	27.0	11.1333	1	0	2	3	2.0
9	14.0	30.0708	1	1	0	2	0.0
10	4.0	16.7000	1	1	1	3	2.0
11	58.0	26.5500	1	0	0	1	2.0
12	20.0	8.0500	0	0	0	3	2.0
13	39.0	31.2750	0	1	5	3	2.0
14	14.0	7.8542	1	0	0	3	2.0
15	55.0	16.0000	1	0	0	2	2.0
16	2.0	29.1250	0	4	1	3	1.0
17	28.0	13.0000	0	0	0	2	2.0
18	31.0	18.0000	1	1	0	3	2.0
19	28.0	7.2250	1	0	0	3	0.0
20	35.0	26.000					