

## ▼ Import

```
import pandas as pd

train_url = "http://s3.amazonaws.com/assets.datacamp.com/course/Kaggle/train.csv"
test_url = "http://s3.amazonaws.com/assets.datacamp.com/course/Kaggle/test.csv"

train = pd.read_csv(train_url)
test = pd.read_csv(test_url)
```

## ▼ Preprocess

```
print(train.head())
print("\n")
print(train.isnull().sum())
print("\n")
print(train.shape)

PassengerId  Survived  Pclass \
0            1        0     3
1            2        1     1
2            3        1     3
3            4        1     1
4            5        0     3

Name      Sex   Age SibSp \
0    Braund, Mr. Owen Harris   male  22.0      1
1  Cumings, Mrs. John Bradley (Florence Briggs Th... female  38.0      1
2       Heikkinen, Miss. Laina  female  26.0      0
3    Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35.0      1
4        Allen, Mr. William Henry   male  35.0      0

Parch      Ticket   Fare Cabin Embarked
0         0   A/5 21171  7.2500   NaN      S
1         0      PC 17599  71.2833  C85      C
2         0  STON/O2. 3101282  7.9250   NaN      S
3         0        113803  53.1000  C123      S
4         0        373450  8.0500   NaN      S

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

(891, 12)
```

```
# T8
train['Age'] = train['Age'].fillna(train['Age'].mode()[0])
train['Embarked'] = train['Embarked'].fillna(train['Embarked'].mode()[0])

test['Age'] = test['Age'].fillna(train['Age'].mode()[0])
test['Embarked'] = test['Embarked'].fillna(train['Embarked'].mode()[0])

print(f"Median Age = {train['Age'].median()}")
```

Median Age = 24.0

```
# T9
train.loc[train["Embarked"] == "S", "Embarked"] = 0
train.loc[train["Embarked"] == "C", "Embarked"] = 1
```

```

train.loc[train["Embarked"] == "Q", "Embarked"] = 2
train.loc[train["Sex"] == "male", "Sex"] = 0
train.loc[train["Sex"] == "female", "Sex"] = 1

test.loc[test["Embarked"] == "S", "Embarked"] = 0
test.loc[test["Embarked"] == "C", "Embarked"] = 1
test.loc[test["Embarked"] == "Q", "Embarked"] = 2

test.loc[test["Sex"] == "male", "Sex"] = 0
test.loc[test["Sex"] == "female", "Sex"] = 1

```

## Model

```

import numpy as np

X = np.array(train[["Pclass", "Sex", "Age", "Embarked"]].values, dtype=float)
y = np.array(train["Survived"].values, dtype=float)

m = len(y)
X_b = np.c_[np.ones((m, 1)), X]

def sigmoid(z):
    return 1 / (1 + np.exp(-z))

theta = np.zeros(X_b.shape[1])

learning_rate = 0.001
iterations = 100000

for i in range(iterations):
    z = np.dot(X_b, theta)
    h = sigmoid(z)

    gradient = np.dot(X_b.T, (h - y)) / m

    theta -= learning_rate * gradient

final_scores = sigmoid(np.dot(X_b, theta))

predictions = (final_scores >= 0.5).astype(int)

accuracy = np.mean(predictions == y)
print(f"Accuracy: {accuracy * 100:.2f}%")

```

Accuracy: 79.69%

## Predict

```

X_test = np.array(test[["Pclass", "Sex", "Age", "Embarked"]].values, dtype=float)
m_test = len(X_test)
X_test_b = np.c_[np.ones((m_test, 1)), X_test]

test_probabilities = sigmoid(np.dot(X_test_b, theta))
test_predictions = (test_probabilities >= 0.5).astype(int)

submission = pd.DataFrame({
    "PassengerId": test["PassengerId"],
    "Survived": test_predictions
})

submission.to_csv("submission.csv", index=False)

print(submission.head())

```

PassengerId	Survived	
0	892	0
1	893	0
2	894	0
3	895	0

4 896 1

```
theta_matrix = np.linalg.inv(X_b.T.dot(X_b)).dot(X_b.T).dot(y)

print("Theta (Gradient Descent):", theta)
print("Theta (Matrix Inversion):", theta_matrix)

weight_mse = np.mean((theta - theta_matrix) ** 2)
print(f"\nMSE of the weight difference: {weight_mse}")

Theta (Gradient Descent): [ 1.02648941 -0.94338079  2.53270066 -0.01848152  0.31728152]
Theta (Matrix Inversion): [ 0.76512686 -0.18828708  0.49299994 -0.00478358  0.04513561]

MSE of the weight difference: 0.9746213905435914
```