```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you create a version using "Save & Run All"
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session
```

```
/kaggle/input/titanic/train.csv
/kaggle/input/titanic/test.csv
/kaggle/input/titanic/gender_submission.csv
```

```
df = pd.read_csv("/kaggle/input/titanic/train.csv")

df.head()
```

```
   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
```

```
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
```

```
df = df.drop(columns=["Cabin", "Ticket", "Name", "PassengerId"])
```

```
df['Age'] = df['Age'].fillna(df['Age'].mean())
```

```
X = df.drop(columns=['Survived'])
y = df['Survived']
y = y.to_numpy()
```

```
X.head()
```

```
   Pclass     Sex   Age  SibSp  Parch     Fare Embarked
0       3    male  22.0      1      0   7.2500        S
1       1  female  38.0      1      0  71.2833        C
2       3  female  26.0      0      0   7.9250        S
3       1  female  35.0      1      0  53.1000        S
4       3    male  35.0      0      0   8.0500        S
```

```
from sklearn.preprocessing import StandardScaler, LabelEncoder
```

```
X['Sex'] = LabelEncoder().fit_transform(X['Sex'])
X['Embarked'] = LabelEncoder().fit_transform(X['Embarked'])
#X['Age'] =
StandardScaler().fit_transform(X['Age'].to_numpy().reshape(-1,1))
#X['Fare'] =
StandardScaler().fit_transform(X['Fare'].to_numpy().reshape(-1,1))
X.head()
```

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	3	1	22.0	1	0	7.2500	2
1	1	0	38.0	1	0	71.2833	0
2	3	0	26.0	0	0	7.9250	2
3	1	0	35.0	1	0	53.1000	2
4	3	1	35.0	0	0	8.0500	2

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.33, random_state=42)

model = LogisticRegression()

model.fit(X_train, y_train)
```

```
LogisticRegression()
```

```
y_preds = model.predict(X_test)

from sklearn.metrics import accuracy_score

accuracy_score(y_test, y_preds)
```

```
0.8169491525423729
```

```
df2 = pd.read_csv("/kaggle/input/titanic/test.csv")

pid = df2['PassengerId']
df2 = df2.drop(columns=["Cabin", "Ticket", "Name", "PassengerId"])
df2['Embarked'] = df2['Embarked'].fillna('S')
df2['Fare'] = df2['Fare'].fillna(0.0)

df2.head()
```

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

```
from sklearn.preprocessing import LabelEncoder
df2['Sex'] = LabelEncoder().fit_transform(df2['Sex'])
```

```
df2['Age'] = df2['Age'].fillna(df2['Age'].mean())
df2['Embarked'] = LabelEncoder().fit_transform(df2['Embarked'])

df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Pclass    418 non-null    int64
 1   Sex       418 non-null    int64
 2   Age       418 non-null    float64
 3   SibSp     418 non-null    int64
 4   Parch     418 non-null    int64
 5   Fare      418 non-null    float64
 6   Embarked  418 non-null    int64
dtypes: float64(2), int64(5)
memory usage: 23.0 KB
```

```
preds = model.predict(df2)

output = pd.DataFrame({'PassengerId':pid.to_numpy(), 'Survived':
preds})

output.to_csv("/kaggle/working/submission.csv")
```