

```
import pandas as pd
import numpy as np

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

```
df = pd.read_csv("train.csv")
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	grid icon
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th... Heikkinen, Miss. Laina	female	38.0	1	0	PC 17599	71.2833	C85	C	
2	3	1	3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S	
3	4	1	1		female	35.0	1	0	113803	53.1000	C123	S	

Next steps: [Generate code with df](#) [New interactive sheet](#)

```
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
```

```
features = ['Pclass', 'Sex', 'Age', 'Fare', 'Embarked']
X = df[features]
y = df['Survived']
```

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

```
/tmp/ipython-input-1041008765.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view)  
`X['Age'] = X['Age'].fillna(X['Age'].mean())`

```
X['Embarked'] = X['Embarked'].fillna(X['Embarked'].mode()[0])
```

```
/tmp/ipython-input-1752051254.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view)  
`X['Embarked'] = X['Embarked'].fillna(X['Embarked'].mode()[0])`

```
X = pd.get_dummies(X, drop_first=True)
X.head()
```

	Pclass	Age	Fare	Sex_male	Embarked_Q	Embarked_S	grid
0	3	22.0	7.2500	True	False	True	
1	1	38.0	71.2833	False	False	False	
2	3	26.0	7.9250	False	False	True	
3	1	35.0	53.1000	False	False	True	
4	3	35.0	8.0500	True	False	True	

Next steps: [Generate code with X](#) [New interactive sheet](#)

```
def train_and_evaluate(test_size):
    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=test_size, random_state=42
    )

    model = LogisticRegression(max_iter=1000)
    model.fit(X_train, y_train)

    y_pred = model.predict(X_test)

    print(f"\nTrain/Test Split: {int((1-test_size)*100)}/{int(test_size*100)}")
    print("Accuracy:", accuracy_score(y_test, y_pred))
    print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
    print("Classification Report:\n", classification_report(y_test, y_pred))
```

```
train_and_evaluate(0.30) # 70/30
train_and_evaluate(0.20) # 80/20
train_and_evaluate(0.10) # 90/10
```

Train/Test Split: 70/30  
Accuracy: 0.7985074626865671  
Confusion Matrix:  
[[133 24]  
 [ 30 81]]  
Classification Report:  

	precision	recall	f1-score	support
0	0.82	0.85	0.83	157
1	0.77	0.73	0.75	111
accuracy			0.80	268
macro avg	0.79	0.79	0.79	268
weighted avg	0.80	0.80	0.80	268

Train/Test Split: 80/20  
Accuracy: 0.7988826815642458  
Confusion Matrix:  
[[88 17]  
 [19 55]]  
Classification Report:  

	precision	recall	f1-score	support
0	0.82	0.84	0.83	105
1	0.76	0.74	0.75	74
accuracy			0.80	179
macro avg	0.79	0.79	0.79	179
weighted avg	0.80	0.80	0.80	179

Train/Test Split: 90/10  
Accuracy: 0.8222222222222222  
Confusion Matrix:  
[[44 10]  
 [ 6 30]]  
Classification Report:  

	precision	recall	f1-score	support
0	0.88	0.81	0.85	54
1	0.75	0.83	0.79	36
accuracy			0.82	90
macro avg	0.81	0.82	0.82	90
weighted avg	0.83	0.82	0.82	90

```
!unzip "archieve(9).zip"
```

```
unzip:  cannot find or open archieve(9).zip, archieve(9).zip.zip or archieve(9).zip.ZIP.
```

```
!mv "archieve(9).zip" archive.zip
```

```
mv: cannot stat 'archieve(9).zip': No such file or directory
```