```
import pandas as pd
```

Double-click (or enter) to edit

```
test_data= pd.read_csv("/content/test.csv")
```

```
train_data= pd.read_csv("/content/train.csv")
```

```
train_data.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
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...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Next steps: Generate code with `train_data` New interactive sheet

```
test_data.head()
```

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

Next steps: Generate code with `test_data` New interactive sheet

```
train_data.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
```

```
train_data.isnull().sum()
```

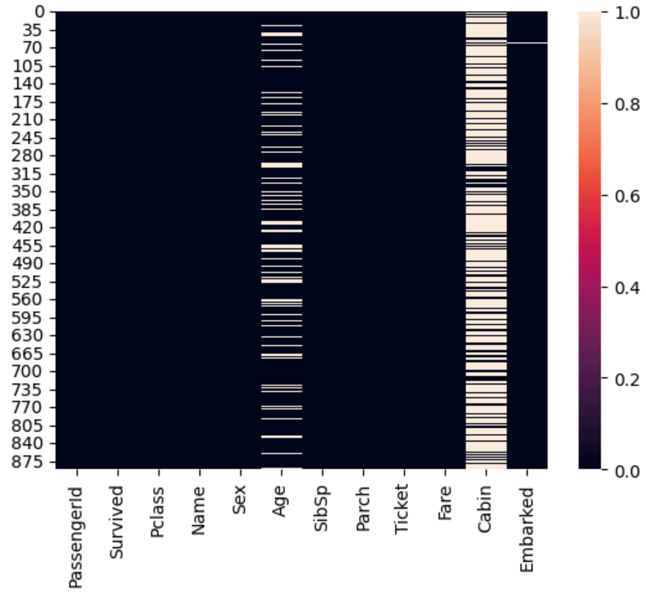
	0
PassengerId	0
Survived	0
Pclass	0
Name	0
Sex	0
Age	177
SibSp	0
Parch	0
Ticket	0
Fare	0
Cabin	687
Embarked	2

dtype: int64

```
import matplotlib.pyplot as plt
import seaborn as sns
```

```
sns.heatmap(train_data.isnull())
```

```
<Axes: >
```



```
train_data['Age'].fillna(train_data['Age'].mean(),inplace=True)
train_data['Embarked'].fillna(train_data['Embarked'].mode()[0],inplace=True)

test_data['Age'].fillna(test_data['Age'].mean(),inplace=True)
test_data['Fare'].fillna(test_data['Fare'].mean(),inplace=True)
```

```
/tmp/ipython-input-3382028760.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through c
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col]


  train_data['Age'].fillna(train_data['Age'].mean(),inplace=True)
/tmp/ipython-input-3382028760.py:2: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through c
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col]


  train_data['Embarked'].fillna(train_data['Embarked'].mode()[0],inplace=True)
/tmp/ipython-input-3382028760.py:4: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through c
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col]


  test_data['Age'].fillna(test_data['Age'].mean(),inplace=True)
/tmp/ipython-input-3382028760.py:5: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through c
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col]
```

```
test_data['Fare'].fillna(test_data['Fare'].mean(),inplace=True)
```

```
train_data.isnull().sum().sort_values(ascending=False)
```

	0
Cabin	687
PassengerId	0
Pclass	0
Survived	0
Name	0
Sex	0
SibSp	0
Age	0
Parch	0
Ticket	0
Fare	0
Embarked	0

dtype: int64

```
train_data=pd.get_dummies(train_data, columns=['Sex','Embarked'])
test_data=pd.get_dummies(test_data, columns=['Sex','Embarked'])
```

```
train_data.head()
```

	PassengerId	Survived	Pclass	Name	Age	SibSp	Parch	Ticket	Fare	Cabin	Sex_female	Sex_male	Embarked_C	Embar
0	1	0	3	Braund, Mr. Owen Harris	22.0	1	0	A/5 21171	7.2500	NaN	False	True	False	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	38.0	1	0	PC 17599	71.2833	C85	True	False	True	
2	3	1	3	Heikkinen, Miss. Laina	26.0	0	0	STON/O2. 3101282	7.9250	NaN	True	False	False	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	35.0	1	0	113803	53.1000	C123	True	False	False	
4	5	0	3	Allen, Mr. William Henry	35.0	0	0	373450	8.0500	NaN	False	True	False	

Next steps: Generate code with train_data New interactive sheet

```
X= train_data.drop(['Survived','Ticket', 'Name', 'Cabin', 'SibSp',  'Parch'],axis=1)
y=train_data['Survived']
X_test=test_data.drop(['Ticket', 'Name', 'Cabin' ],axis=1)
```

```
X.head()
```

	PassengerId	Pclass	Age	Fare	Sex_female	Sex_male	Embarked_C	Embarked_Q	Embarked_S
0	1	3	22.0	7.2500	False	True	False	False	True
1	2	1	38.0	71.2833	True	False	True	False	False
2	3	3	26.0	7.9250	True	False	False	False	True
3	4	1	35.0	53.1000	True	False	False	False	True
4	5	3	35.0	8.0500	False	True	False	False	True

Next steps: Generate code with X New interactive sheet

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,random_state=42)
```

```
from sklearn.ensemble import RandomForestClassifier
model=RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train,y_train)
```

```
▾     RandomForestClassifier     ⓘ ?
RandomForestClassifier(random_state=42)
```

```
from sklearn.metrics import accuracy_score
y_pred=model.predict(X_test)
accuracy=accuracy_score(y_test,y_pred)
accuracy
```

```
0.8324022346368715
```

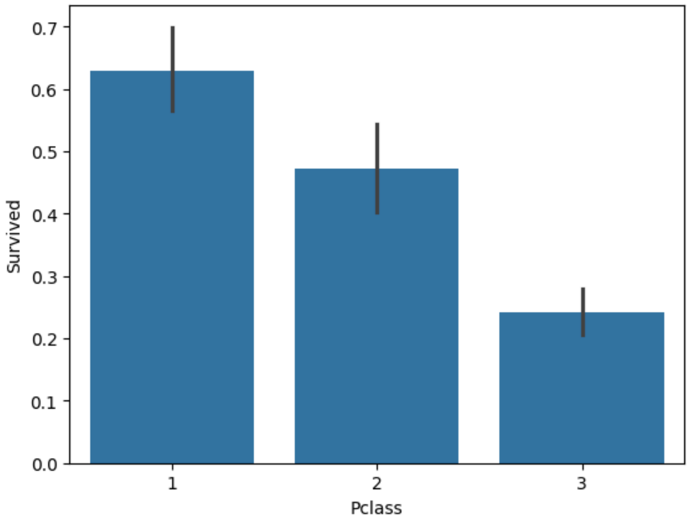
```
predictions = model.predict(test_data.drop(['Ticket', 'Name', 'Cabin', 'SibSp', 'Parch'], axis=1))
```

```
output = pd.DataFrame({'PassengerId': test_data.PassengerId, 'Survived': predictions})
output.to_csv('submission.csv', index=False)
print("Your submission was successfully saved!")
```

```
Your submission was successfully saved!
```

```
sns.barplot(x='Pclass', y='Survived', data=train_data)
```

```
<Axes: xlabel='Pclass', ylabel='Survived'>
```



Start coding or generate with AI.