Titanic - Machine Learning from Disaster

Machine Learning Project – Kaggle Titanic Competition

The Challenge

- The sinking of the Titanic is one of the most infamous shipwrecks in history.
- On April 15, 1912, during her maiden voyage, the widely considered “unsinkable” RMS Titanic sank after colliding with an iceberg. Unfortunately, there weren’t enough lifeboats for everyone onboard, resulting in the death of 1502 out of 2224 passengers and crew.
- While there was some element of luck involved in surviving, it seems some groups of people were more likely to survive than others.
- In this challenge, we ask you to build a predictive model that answers the question: “what sorts of people were more likely to survive?” using passenger data (i.e name, age, gender, socio-economic class, etc).

Dataset Overview (Kaggle Titanic Competition)

The dataset consists of **three CSV files**:

1. **train.csv**

- Contains data for **891 passengers**.

- Includes the **"Survived"** column:

    1 → Passenger survived

    0 → Passenger did not survive

- Used to **train** and understand survival patterns.

Dataset Overview (Kaggle Titanic Competition)

2. test.csv

- Contains data for 418 passengers.
- Does not include the "Survived" column.
- My task is to predict survival for these passengers.

3. gender_submission.csv

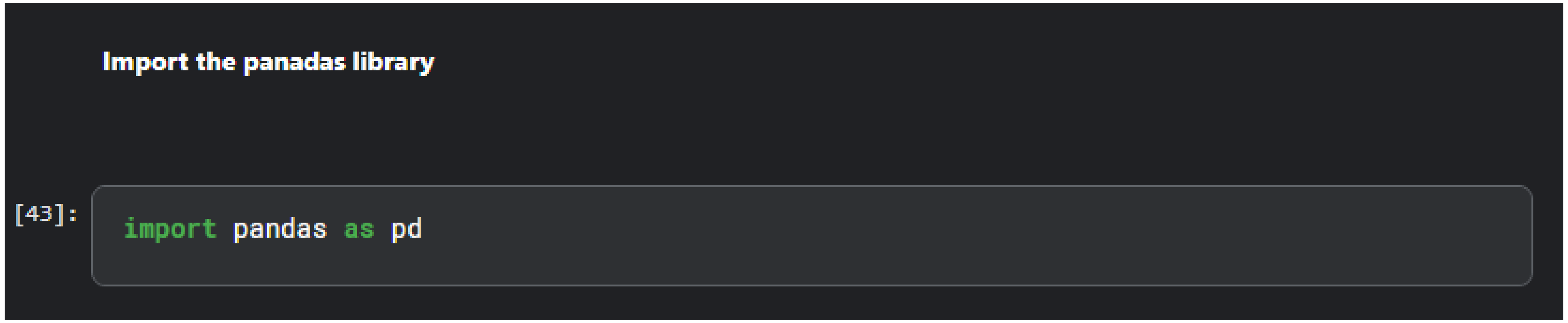
A sample submission file:

- Assumes all females survived and all males did not.

Shows the correct format for your submission.csv:

- Columns: PassengerId, Survived

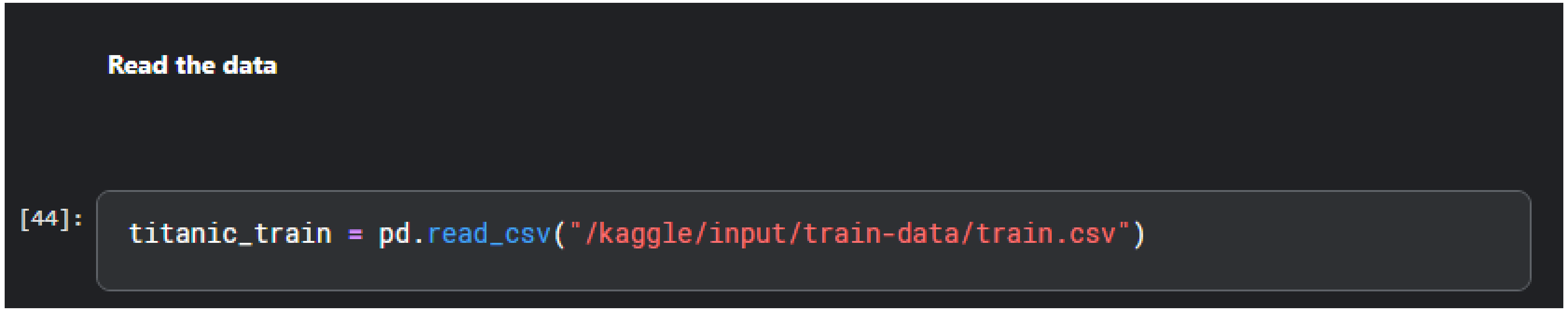
Data Preprocessing on training dataset



Import the panadas library

[43]:

```
import pandas as pd
```

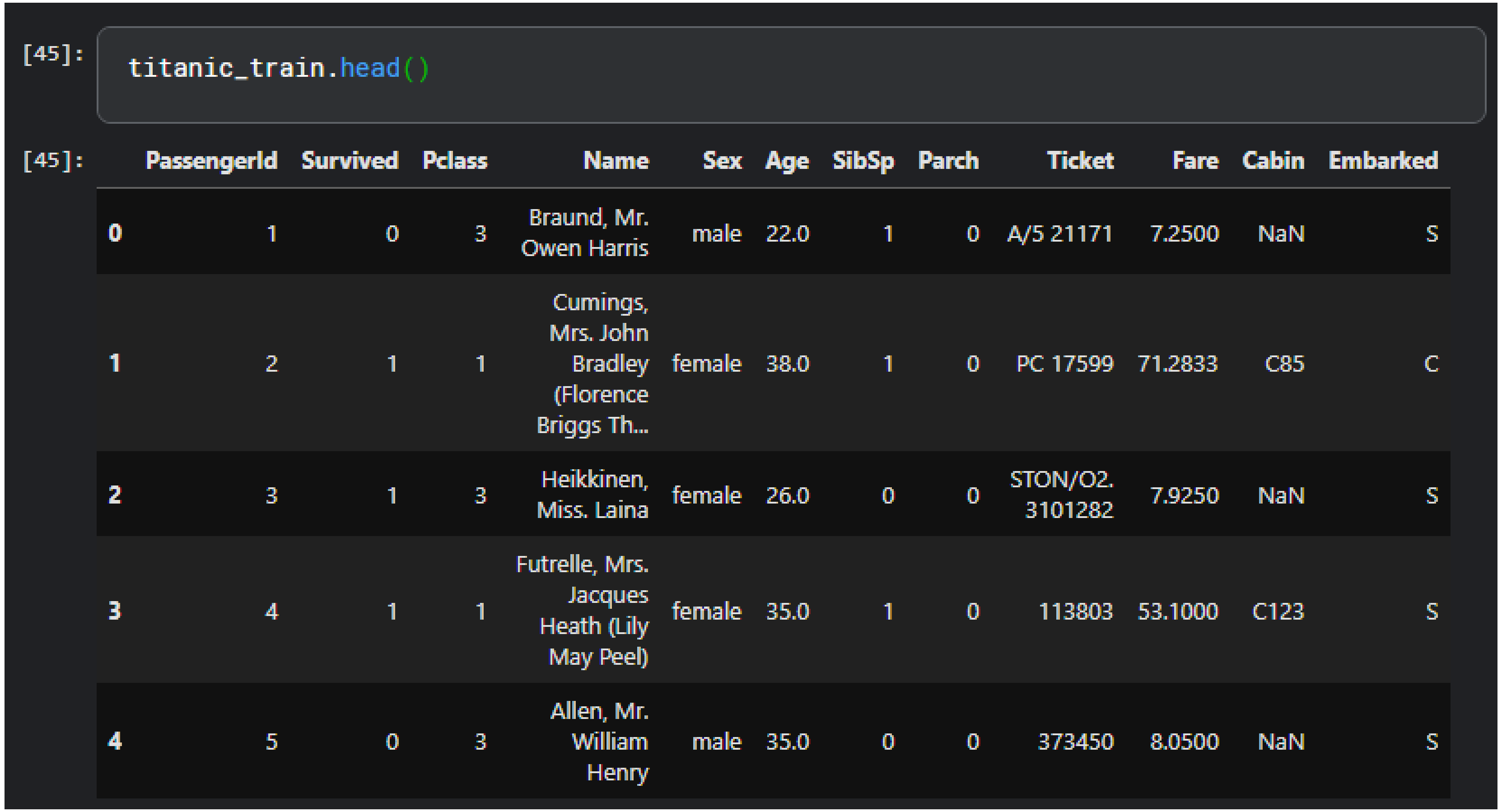


Read the data

[44]:

```
titanic_train = pd.read_csv("/kaggle/input/train-data/train.csv")
```

Data Preprocessing on training dataset



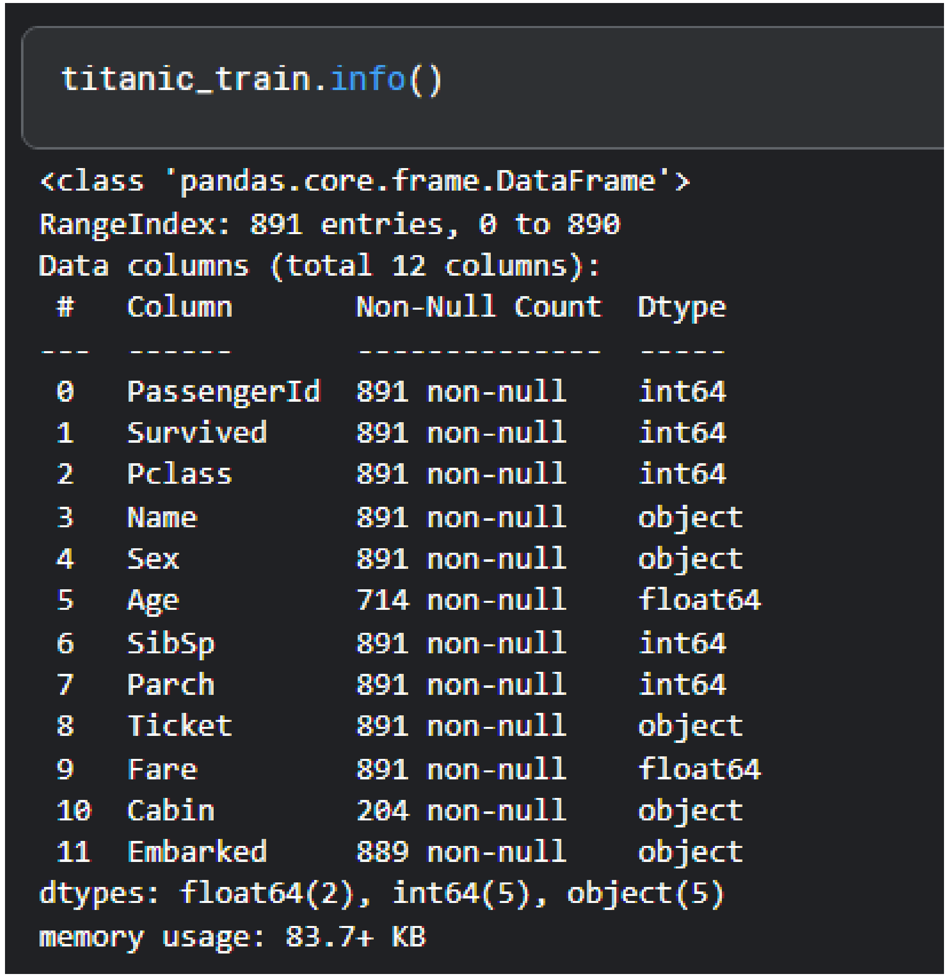
[45]:

```
titanic_train.head()
```

[45]:

	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

Data Preprocessing on training dataset

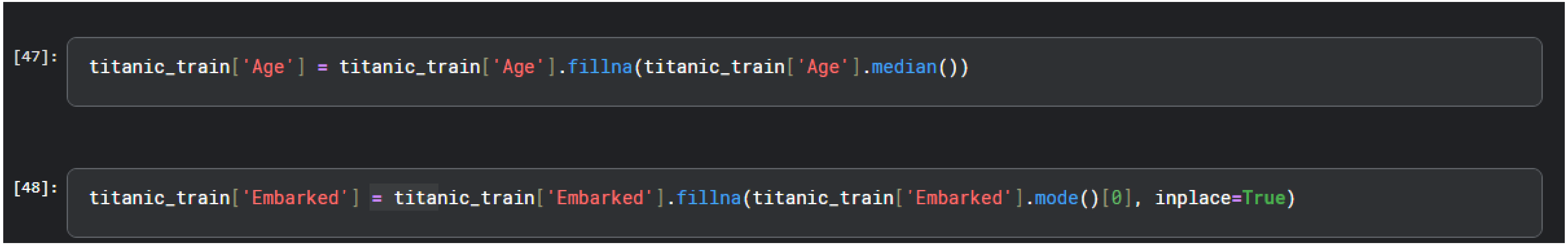


```
titanic_train.info()
```

```
<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
```

Data Preprocessing on training dataset

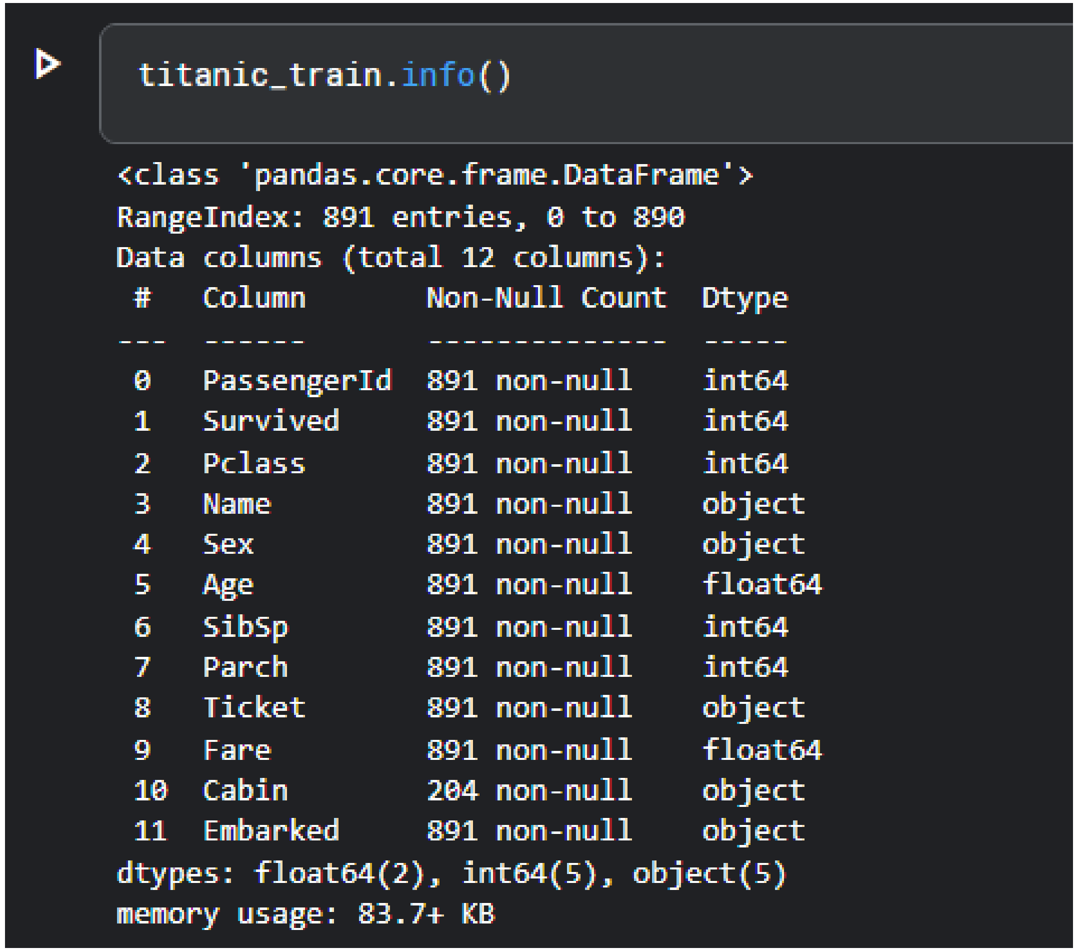
- **Fill the missing values**



```
[47]: titanic_train['Age'] = titanic_train['Age'].fillna(titanic_train['Age'].median())
```

```
[48]: titanic_train['Embarked'] = titanic_train['Embarked'].fillna(titanic_train['Embarked'].mode()[0], inplace=True)
```

Data Preprocessing on training dataset

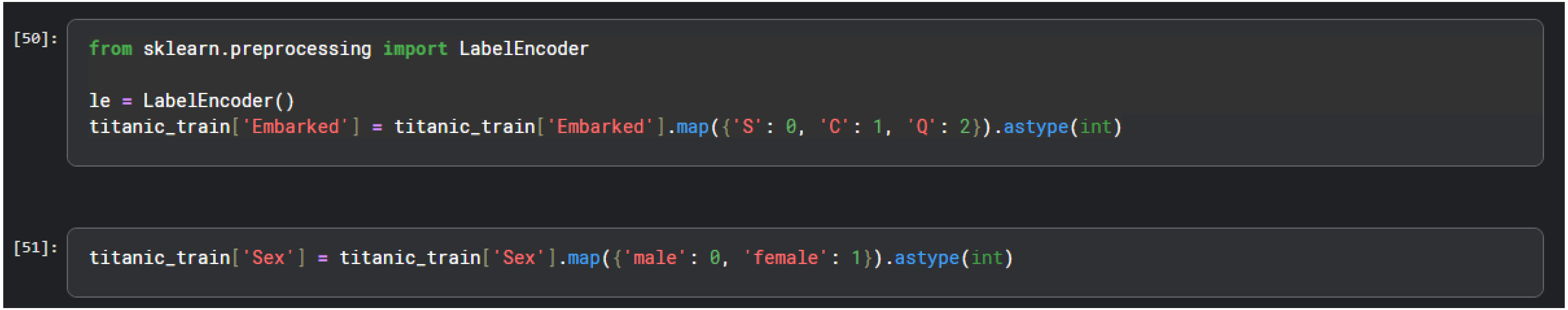


```
titanic_train.info()
```

```
<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          891 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     891 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

Data Preprocessing on training dataset

- **Encoding Categorical Variables**

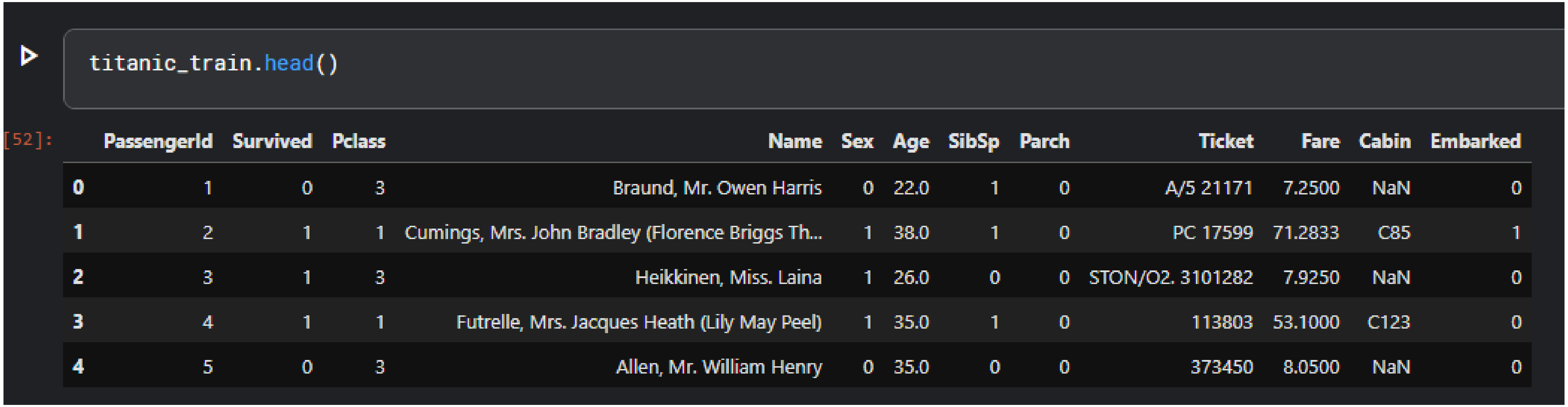


```
[50]: from sklearn.preprocessing import LabelEncoder

      le = LabelEncoder()
      titanic_train['Embarked'] = titanic_train['Embarked'].map({'S': 0, 'C': 1, 'Q': 2}).astype(int)
```

```
[51]: titanic_train['Sex'] = titanic_train['Sex'].map({'male': 0, 'female': 1}).astype(int)
```

Data Preprocessing on training dataset

- **See the data after encoding**



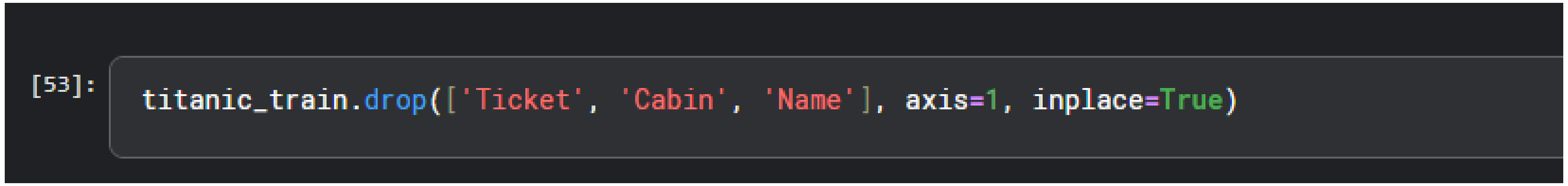
```
titanic_train.head()
```

[52]:

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

Data Preprocessing on training dataset

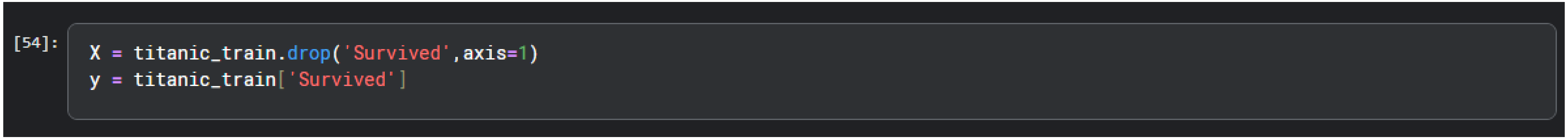
- ➢ **Drop the unnecessary columns**



```
[53]: titanic_train.drop(['Ticket', 'Cabin', 'Name'], axis=1, inplace=True)
```

Model Training using Random Forest

- **Feature and Target Separation**

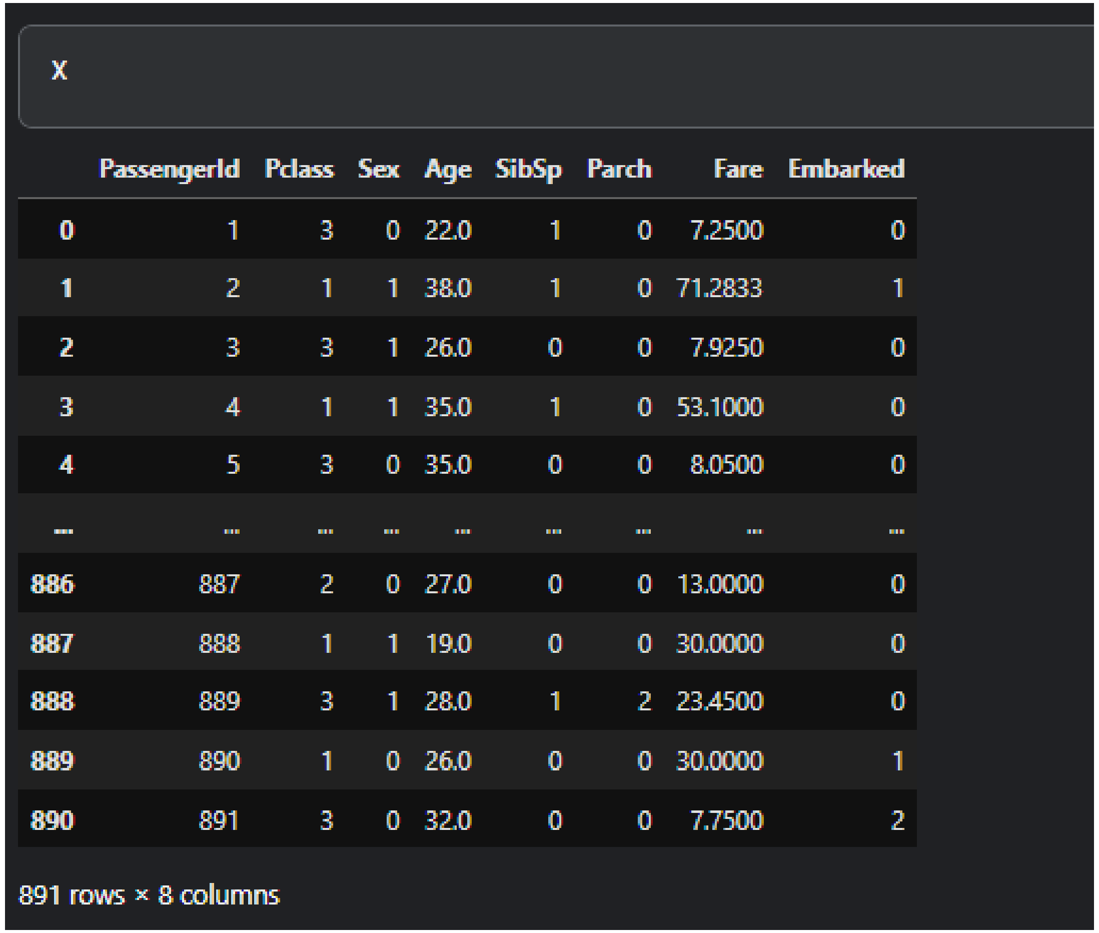


```
[54]:  X = titanic_train.drop('Survived',axis=1)
       y = titanic_train['Survived']
```

This separated the dataset into:

- X containing the input features (all columns except 'Survived')
- y containing the target variable 'Survived' for model training.

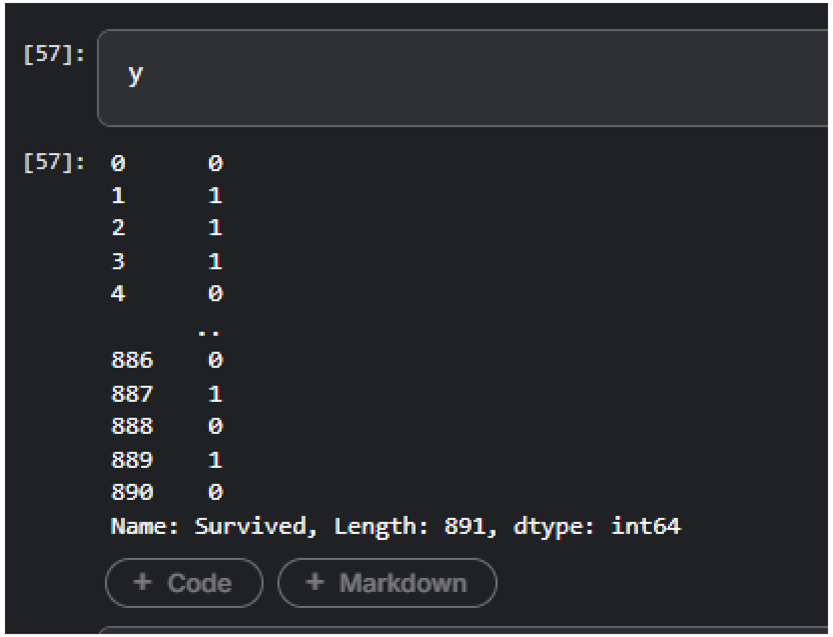
Model Training using Random Forest



```
X
```

	PassengerId	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	1	3	0	22.0	1	0	7.2500	0
1	2	1	1	38.0	1	0	71.2833	1
2	3	3	1	26.0	0	0	7.9250	0
3	4	1	1	35.0	1	0	53.1000	0
4	5	3	0	35.0	0	0	8.0500	0
...	...	...	...	...	...	...	...	...
886	887	2	0	27.0	0	0	13.0000	0
887	888	1	1	19.0	0	0	30.0000	0
888	889	3	1	28.0	1	2	23.4500	0
889	890	1	0	26.0	0	0	30.0000	1
890	891	3	0	32.0	0	0	7.7500	2

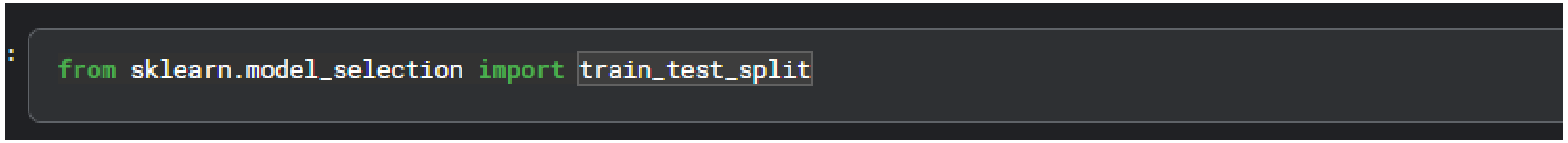
891 rows × 8 columns



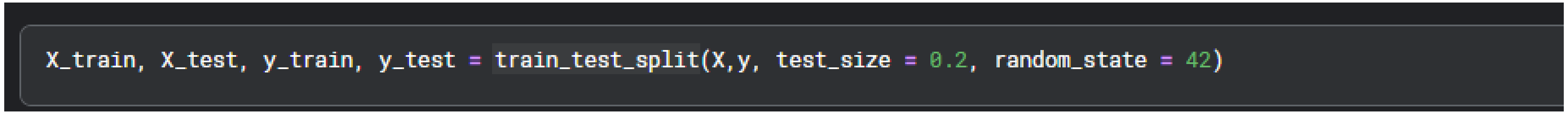
```
[57]:  y

[57]:  0      0
       1      1
       2      1
       3      1
       4      0
             ..
       886    0
       887    1
       888    0
       889    1
       890    0
       Name: Survived, Length: 891, dtype: int64
```

Model Training using Random Forest

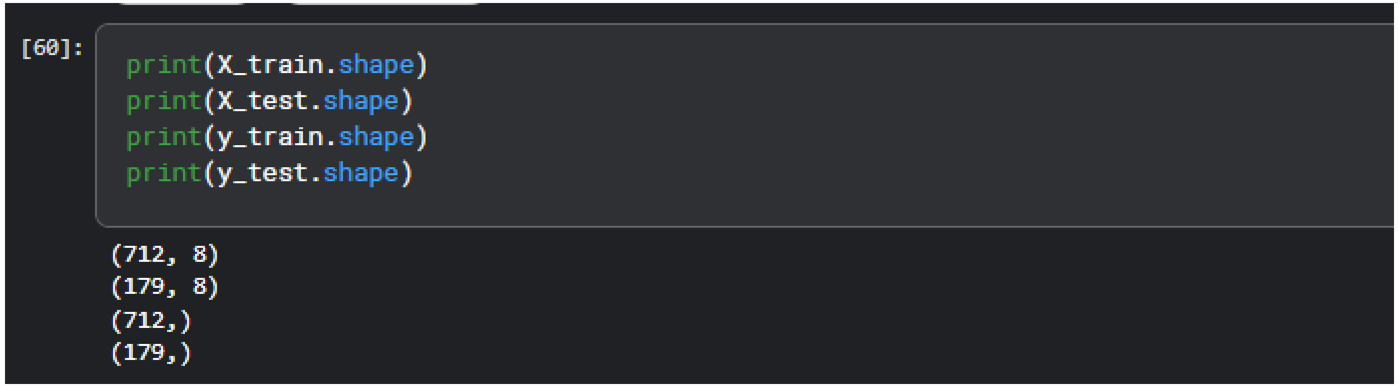
➢ **Splitting the data into training and testing**



```
from sklearn.model_selection import train_test_split
```



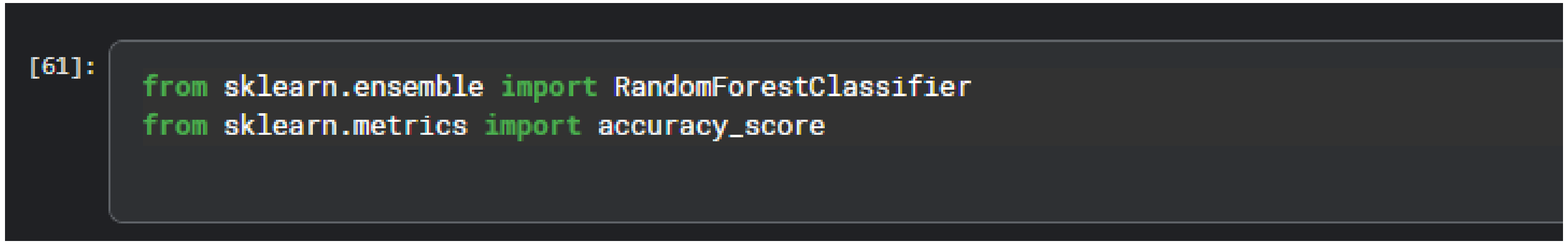
```
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.2, random_state = 42)
```



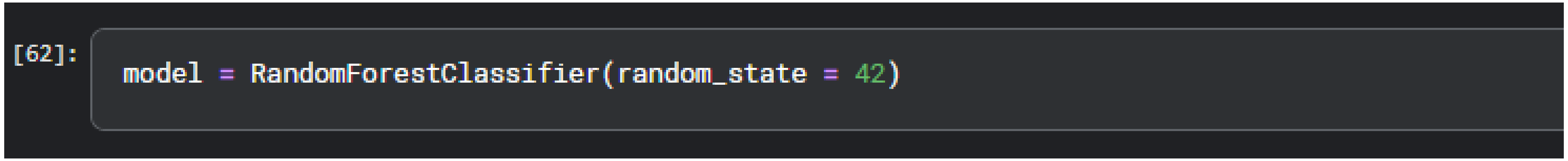
```
[60]: print(X_train.shape)
      print(X_test.shape)
      print(y_train.shape)
      print(y_test.shape)

(712, 8)
(179, 8)
(712,)
(179,)
```

Model Training using Random Forest

- **Import the Randomforest Model**



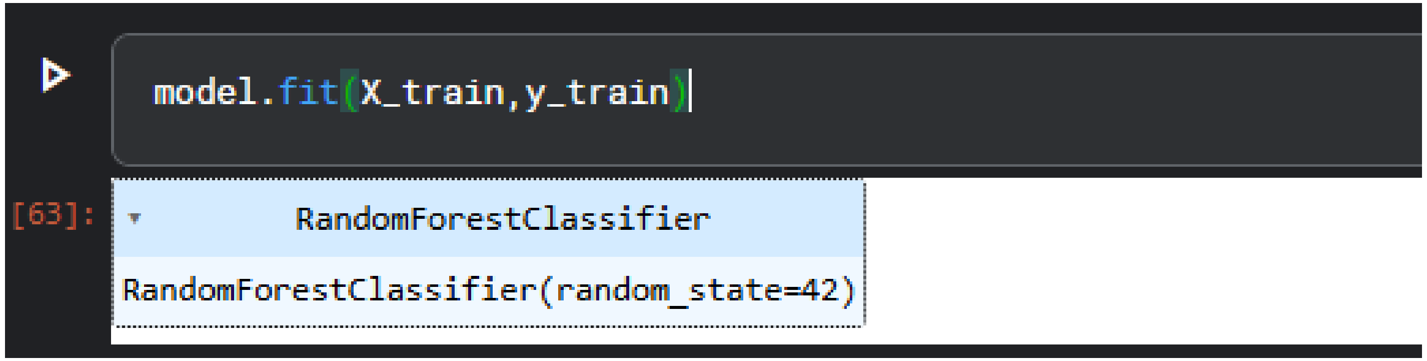
```
[61]:  from sklearn.ensemble import RandomForestClassifier
       from sklearn.metrics import accuracy_score
```



```
[62]:  model = RandomForestClassifier(random_state = 42)
```

Model Training using Random Forest

- **Train the model on training data**

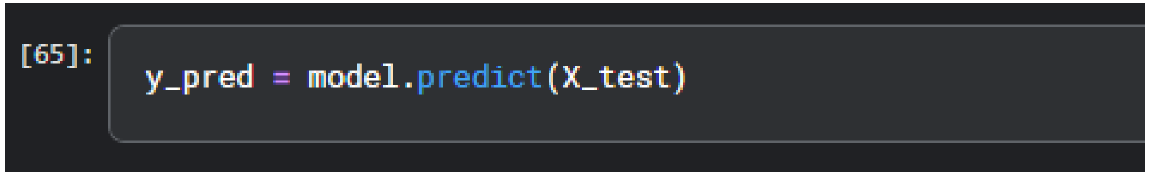


```
model.fit(X_train,y_train)
```

```
[63]:
▾ RandomForestClassifier
RandomForestClassifier(random_state=42)
```

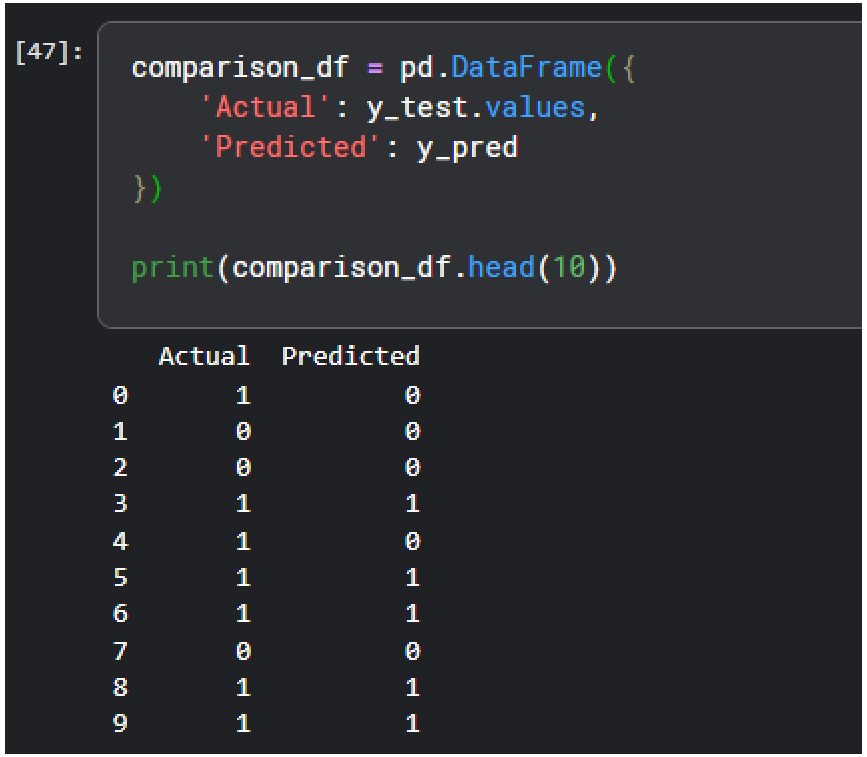
Model Training using Random Forest

- **Predictions of the model on testing data**



[65]:

```
y_pred = model.predict(X_test)
```



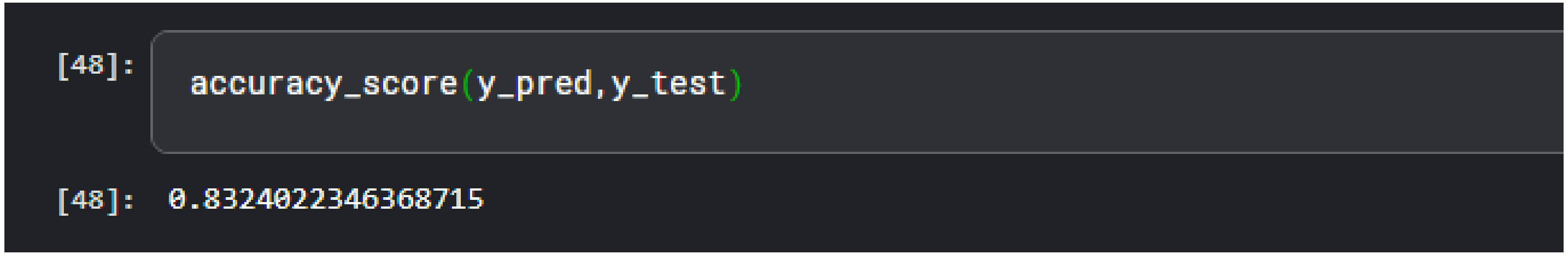
[47]:

```
comparison_df = pd.DataFrame({
    'Actual': y_test.values,
    'Predicted': y_pred
})

print(comparison_df.head(10))
```

```
   Actual  Predicted
0       1          0
1       0          0
2       0          0
3       1          1
4       1          0
5       1          1
6       1          1
7       0          0
8       1          1
9       1          1
```

Model Training using Random Forest

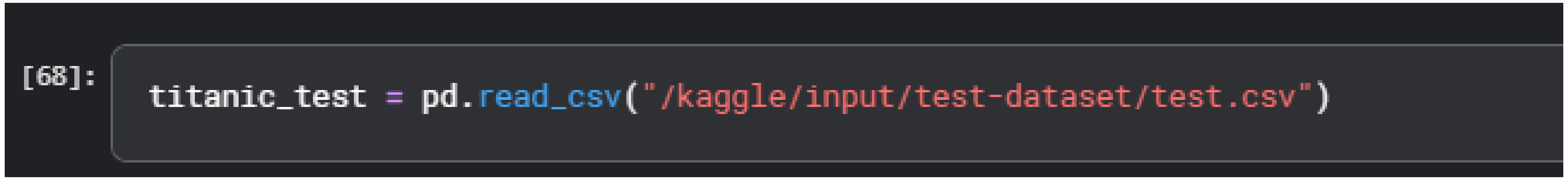
- **Accuracy score of this model**



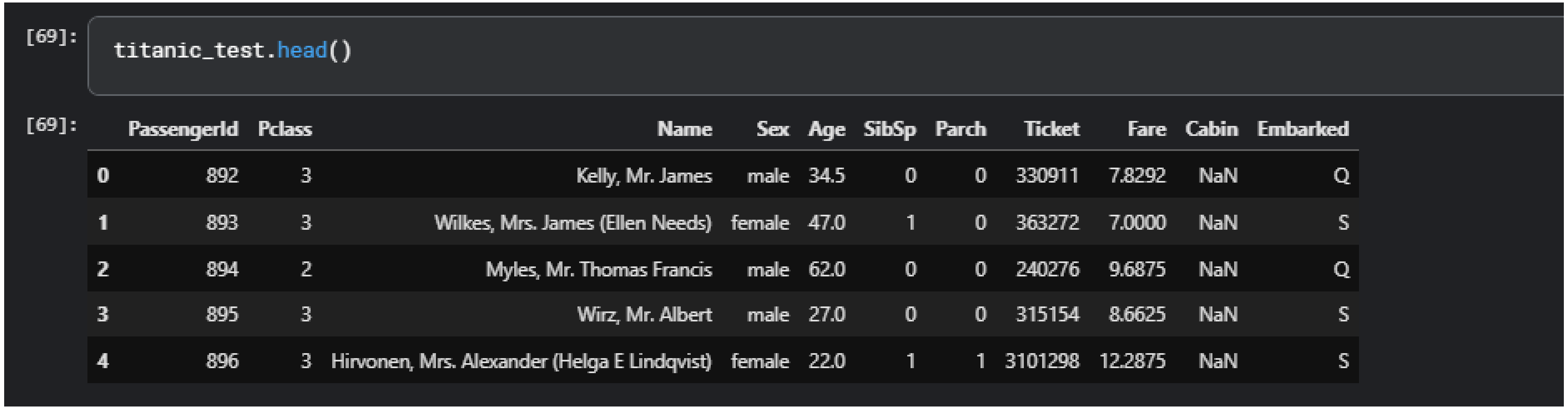
```
[48]:  accuracy_score(y_pred,y_test)

[48]:  0.8324022346368715
```

Data Preprocessing on testing data

- **Read the testing data**



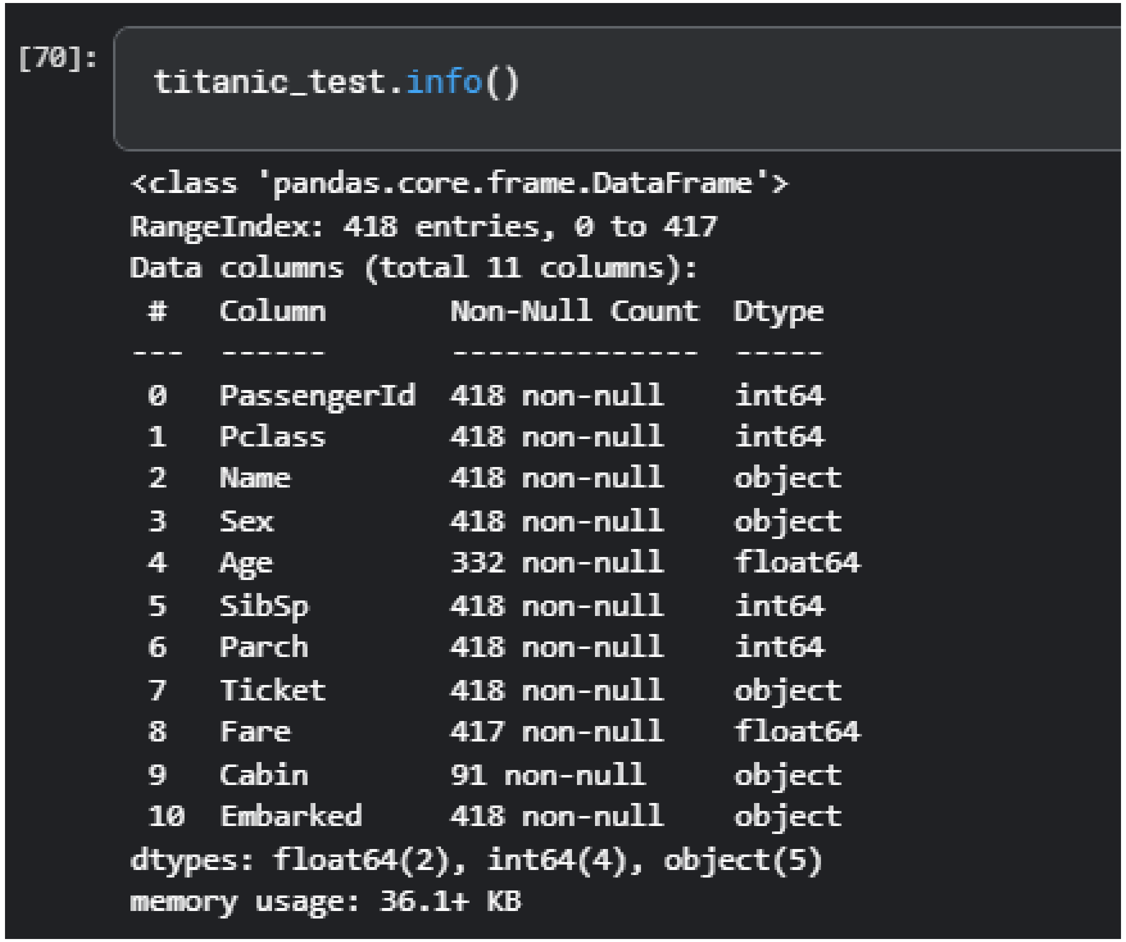
```
[68]:  titanic_test = pd.read_csv("/kaggle/input/test-dataset/test.csv")
```



```
[69]:  titanic_test.head()
```

```
[69]:
   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
```

Data Preprocessing on testing data

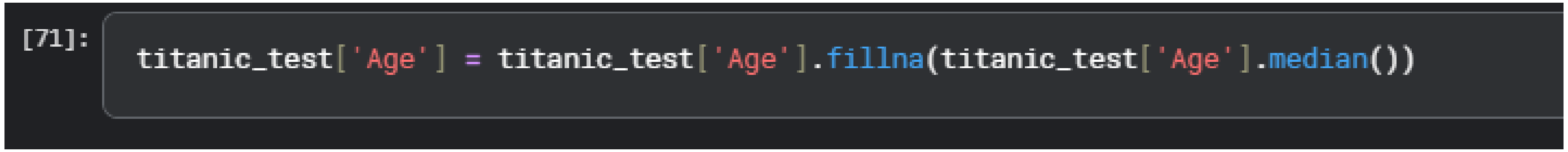


```
[70]: titanic_test.info()
```

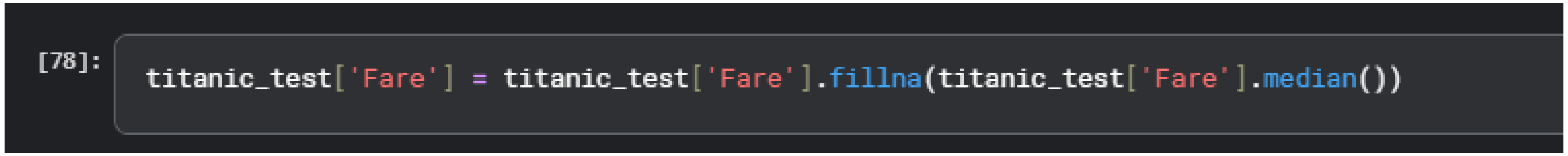
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  418 non-null    int64  
 1   Pclass       418 non-null    int64  
 2   Name         418 non-null    object 
 3   Sex          418 non-null    object 
 4   Age          332 non-null    float64
 5   SibSp        418 non-null    int64  
 6   Parch        418 non-null    int64  
 7   Ticket       418 non-null    object 
 8   Fare         417 non-null    float64
 9   Cabin        91 non-null     object 
 10  Embarked     418 non-null    object 
dtypes: float64(2), int64(4), object(5)
memory usage: 36.1+ KB
```

Data Preprocessing on testing data

- **Filling the missing values**



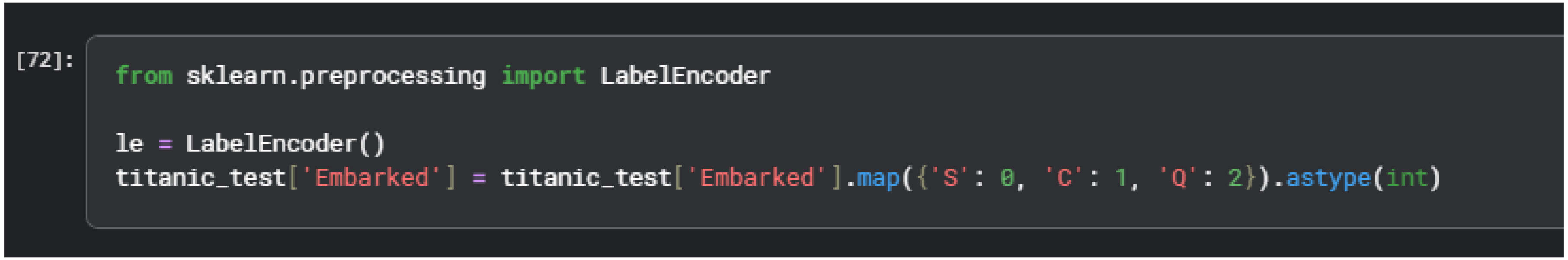
```
[71]:  titanic_test['Age'] = titanic_test['Age'].fillna(titanic_test['Age'].median())
```



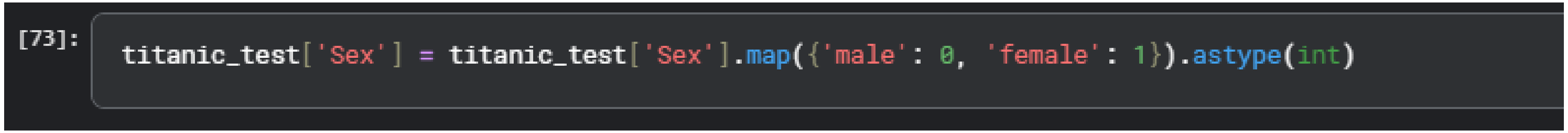
```
[78]:  titanic_test['Fare'] = titanic_test['Fare'].fillna(titanic_test['Fare'].median())
```

Data Preprocessing on testing data

- **Encoding Categorical Variables**



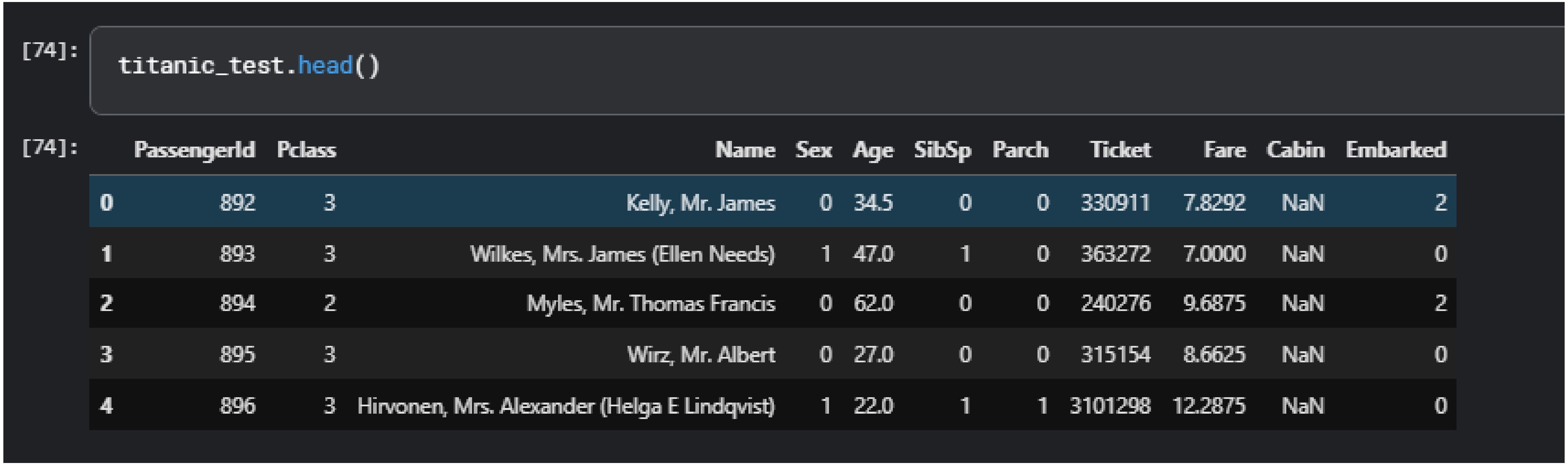
```
[72]: from sklearn.preprocessing import LabelEncoder

      le = LabelEncoder()
      titanic_test['Embarked'] = titanic_test['Embarked'].map({'S': 0, 'C': 1, 'Q': 2}).astype(int)
```



```
[73]: titanic_test['Sex'] = titanic_test['Sex'].map({'male': 0, 'female': 1}).astype(int)
```

Data Preprocessing on testing data

- **Preview of data after encoding**



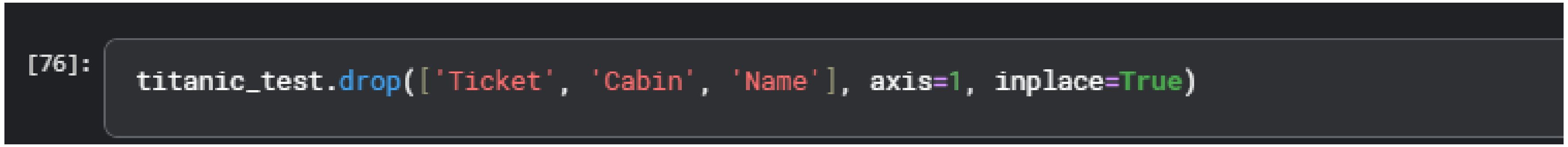
```
[74]: titanic_test.head()
```

[74]:

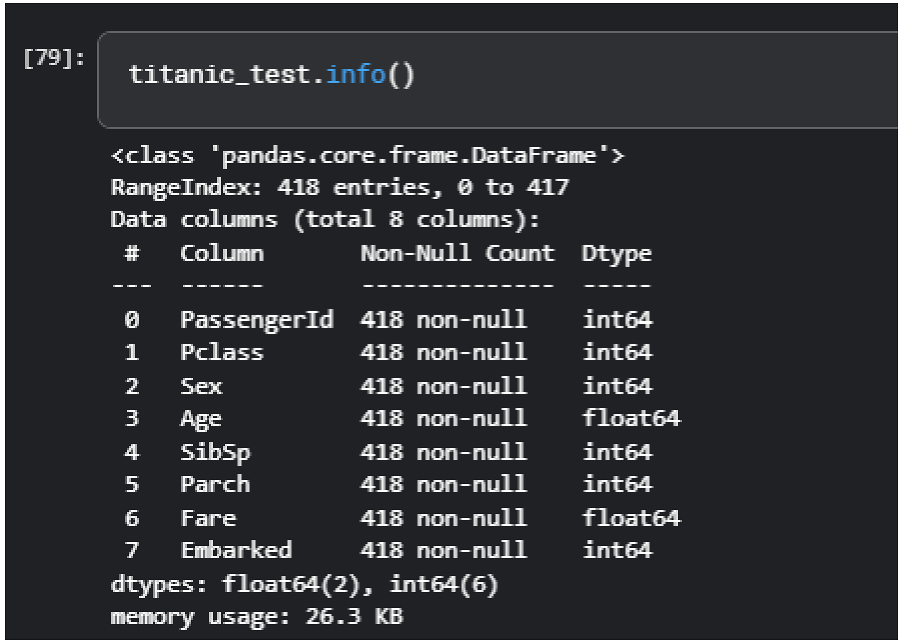
	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	3	Kelly, Mr. James	0	34.5	0	0	330911	7.8292	NaN	2
1	893	3	Wilkes, Mrs. James (Ellen Needs)	1	47.0	1	0	363272	7.0000	NaN	0
2	894	2	Myles, Mr. Thomas Francis	0	62.0	0	0	240276	9.6875	NaN	2
3	895	3	Wirz, Mr. Albert	0	27.0	0	0	315154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

Data Preprocessing on testing data

- **Removing the unnecessary columns**



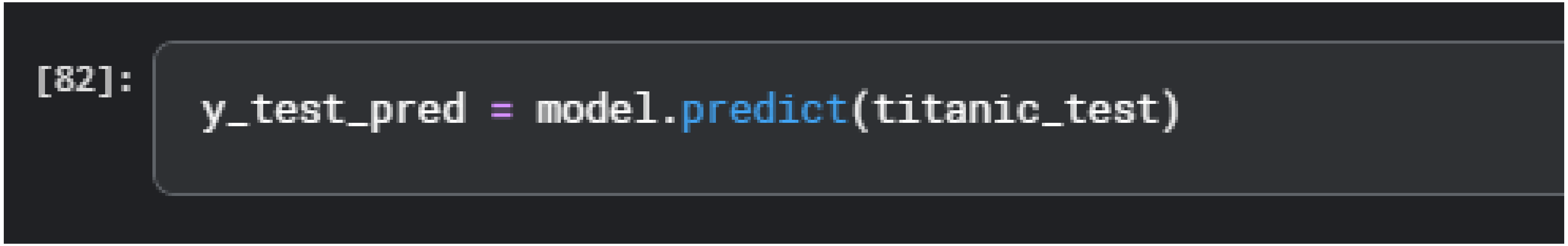
```
[76]: titanic_test.drop(['Ticket', 'Cabin', 'Name'], axis=1, inplace=True)
```



```
[79]: titanic_test.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  418 non-null    int64  
 1   Pclass       418 non-null    int64  
 2   Sex          418 non-null    int64  
 3   Age          418 non-null    float64
 4   SibSp        418 non-null    int64  
 5   Parch        418 non-null    int64  
 6   Fare         418 non-null    float64
 7   Embarked     418 non-null    int64  
dtypes: float64(2), int64(6)
memory usage: 26.3 KB
```

Predicting the testing data

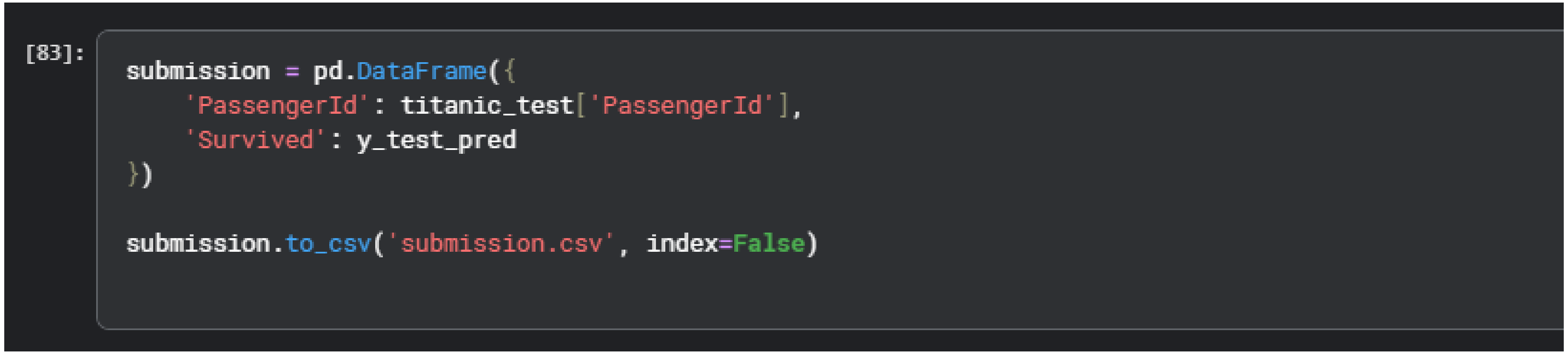
- **Predicting the testing data**



```python
y_test_pred = model.predict(titanic_test)
```

Predicting the testing data

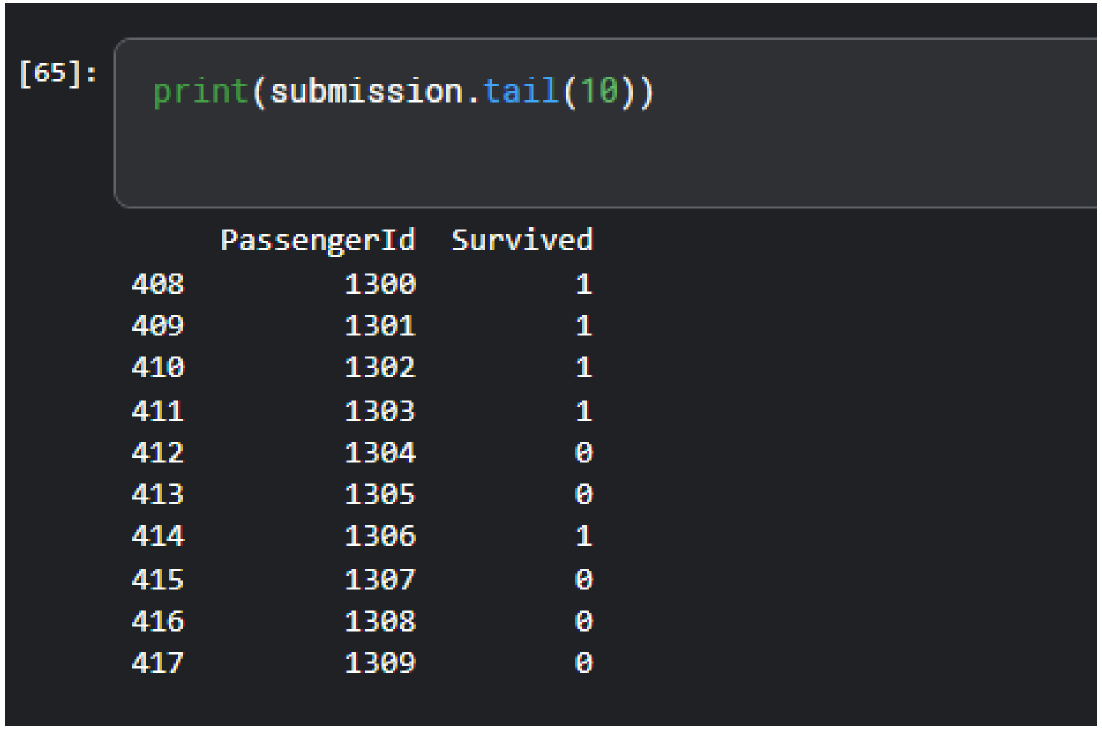
- **Downloading the submission file**



```
[83]: submission = pd.DataFrame({
          'PassengerId': titanic_test['PassengerId'],
          'Survived': y_test_pred
      })

      submission.to_csv('submission.csv', index=False)
```

Predicting the testing data

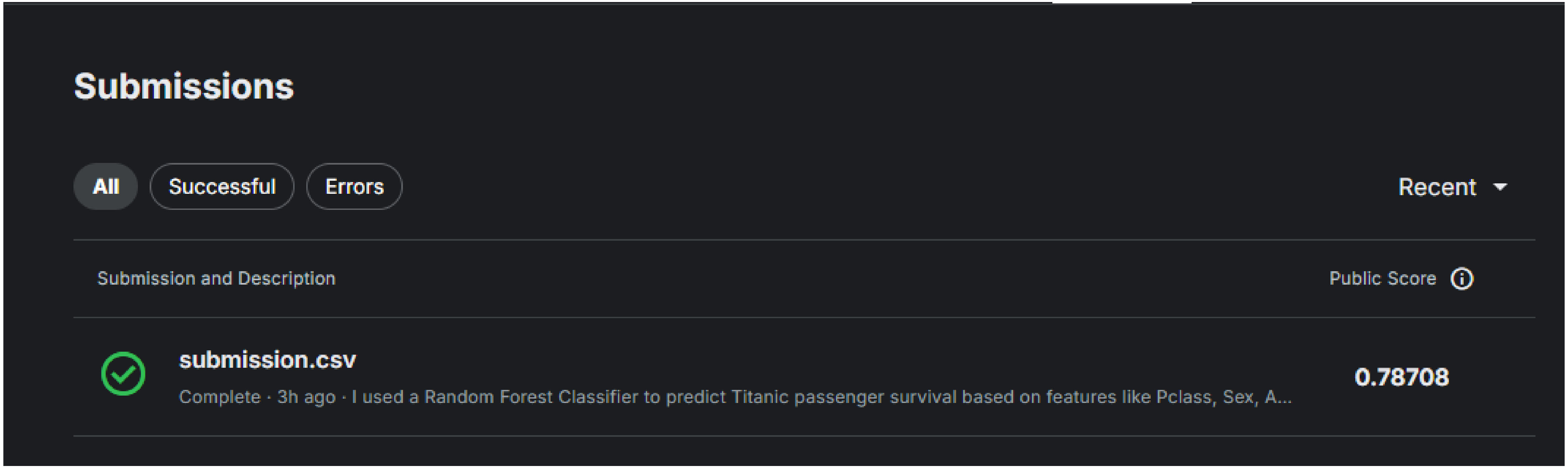
- **See the predictions of the model on testing data**



```
[65]: print(submission.tail(10))
```

```
     PassengerId  Survived
408         1300         1
409         1301         1
410         1302         1
411         1303         1
412         1304         0
413         1305         0
414         1306         1
415         1307         0
416         1308         0
417         1309         0
```

My submission score on the Kaggle leaderboard



Submissions

Submission and Description — Public Score ⓘ

✓ **submission.csv**

Complete · 3h ago · I used a Random Forest Classifier to predict Titanic passenger survival based on features like Pclass, Sex, A...

0.78708