SUPERIOR UNIVERSITY

Name: Esha asif
Roll no:034
Section:BSAI 4A
Subject:Programming for AI
LAB TASK 2

Code:

```
import pandas as pd
import numpy as np
import pickle

from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier

train_df = pd.read_csv("train.csv")
test_df = pd.read_csv("test.csv")

print(f"train Dataset - Rows: {train_df.shape[0]}, Cols: {train_df.shape[1]}")
print(f"test Dataset - Rows: {test_df.shape[0]}, Cols: {test_df.shape[1]}")

import pandas as pd
df = pd.read_csv('train.csv')
df.head(5)
df.tail(5)
df.describe()
df.info()
print(df.count())
df.nunique()
print(df.isnull().sum())

label_encoder = LabelEncoder()
train_df['Spa'] = label_encoder.fit_transform(train_df['Spa'])

X = df.drop('Spa', axis=1)
y = df['Spa']

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

import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.impute import SimpleImputer

train_df = pd.read_csv("train.csv")
test_df = pd.read_csv("test.csv")


if y_train.isnull().any():
    y_train = y_train.fillna(y_train.mode()[0])
if y_train.dtype == 'object':
    label_encoder = LabelEncoder()
    y_train = label_encoder.fit_transform(y_train)

# Handle missing values in features (X_train and X_test)
imputer = SimpleImputer(strategy='most_frequent')
X_train_encoded = imputer.fit_transform(X_train)
X_test_encoded = imputer.transform(X_test)

# Convert the numpy arrays back to DataFrame to maintain column names
X_train_encoded = pd.DataFrame(X_train_encoded, columns=X_train.columns)
X_test_encoded = pd.DataFrame(X_test_encoded, columns=X_test.columns)

# Label encoding for categorical features in X_train_encoded and X_test_encoded
categorical_columns = X_train_encoded.select_dtypes(include=['object']).columns

# Combine both train and test categorical columns to fit the encoder
for col in categorical_columns:
    encoder = LabelEncoder()
    combined_data = pd.concat([X_train_encoded[col], X_test_encoded[col]], axis=0)
    encoder.fit(combined_data.astype(str))  # Fit the encoder on the combined data
    X_train_encoded[col] = encoder.transform(X_train_encoded[col].astype(str))
    X_test_encoded[col] = encoder.transform(X_test_encoded[col].astype(str))

model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train_encoded, y_train)

print("model trained")

test_pred = model.predict(X_test_encoded)
print(f"Length of test dataset: {len(test)}")
print(f"Length of PassengerId column: {len(test['PassengerId'])}")
print(f"Length of test_pred: {len(test_pred)}")



if len(test_pred) < len(test):
    missing = len(test) - len(test_pred)
    test_pred = np.concatenate([test_pred, [False] * missing])
test_pred = test_pred[:len(test)]

submission = pd.DataFrame({
    'PassengerId': test['PassengerId'],
    'Transported': test_pred.astype(bool)
})

submission.to_csv('submission.csv', index=False)
print("Submission file created successfully!")
print(submission.head())
```

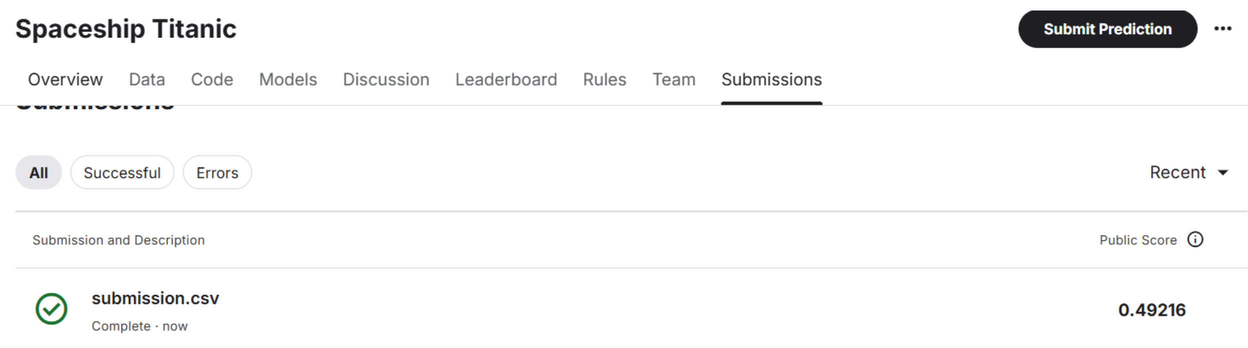
HOW AND WHY:

This is for a kaggle competition named as spaceship titanic our goal was to predict whether a passenger in a dataset was transported or not.

First of all i loaded the train and test data checked for its missing values and explored the dataset as u can see in the output below. Then I filled in any missing values and changed the categories into numbers.

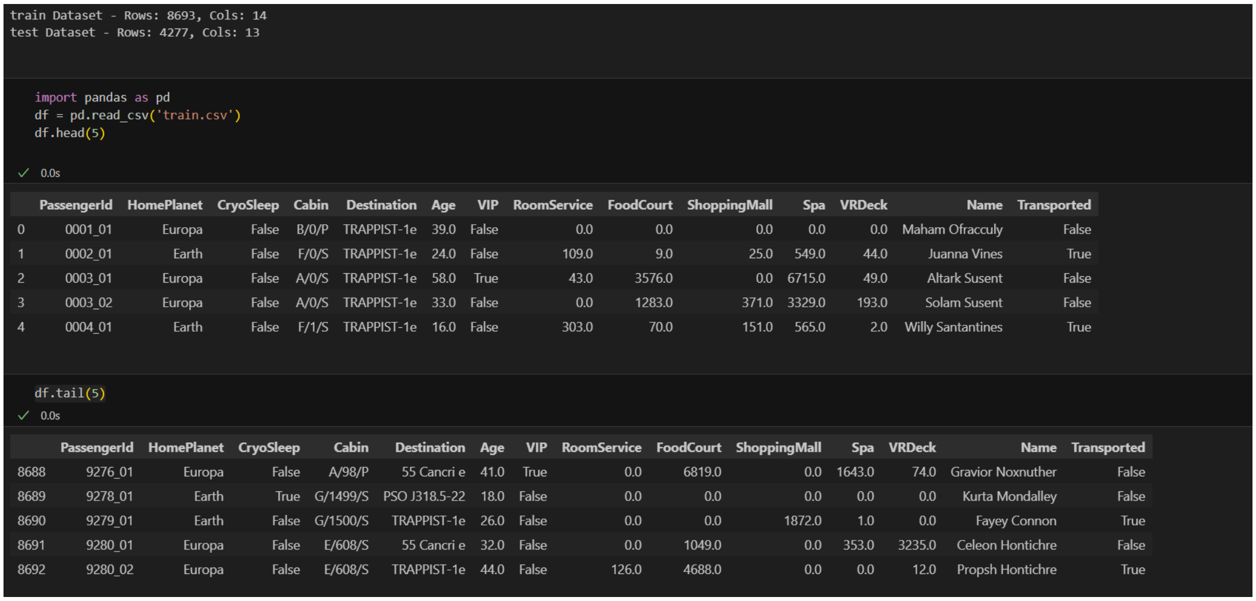
After that the dataset is split into training and testing using a machine learning model random forest classifier. This will train my model and in the end it will create a new file with new predictions and save it as submission.csv as u can see in the last few lines.

OUTPUT:



Spaceship Titanic

Submission and Description	Public Score
submission.csv Complete · now	0.49216



```
train Dataset - Rows: 8693, Cols: 14
test Dataset - Rows: 4277, Cols: 13
```

```python
import pandas as pd
df = pd.read_csv('train.csv')
df.head(5)
```

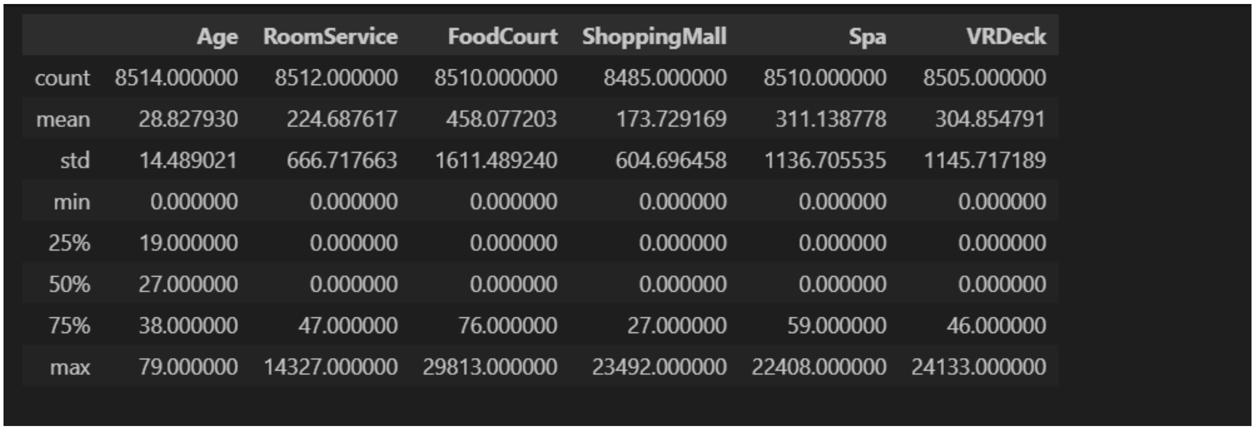
✓ 0.0s

	PassengerId	HomePlanet	CryoSleep	Cabin	Destination	Age	VIP	RoomService	FoodCourt	ShoppingMall	Spa	VRDeck	Name	Transported
0	0001_01	Europa	False	B/0/P	TRAPPIST-1e	39.0	False	0.0	0.0	0.0	0.0	0.0	Maham Ofracculy	False
1	0002_01	Earth	False	F/0/S	TRAPPIST-1e	24.0	False	109.0	9.0	25.0	549.0	44.0	Juanna Vines	True
2	0003_01	Europa	False	A/0/S	TRAPPIST-1e	58.0	True	43.0	3576.0	0.0	6715.0	49.0	Altark Susent	False
3	0003_02	Europa	False	A/0/S	TRAPPIST-1e	33.0	False	0.0	1283.0	371.0	3329.0	193.0	Solam Susent	False
4	0004_01	Earth	False	F/1/S	TRAPPIST-1e	16.0	False	303.0	70.0	151.0	565.0	2.0	Willy Santantines	True

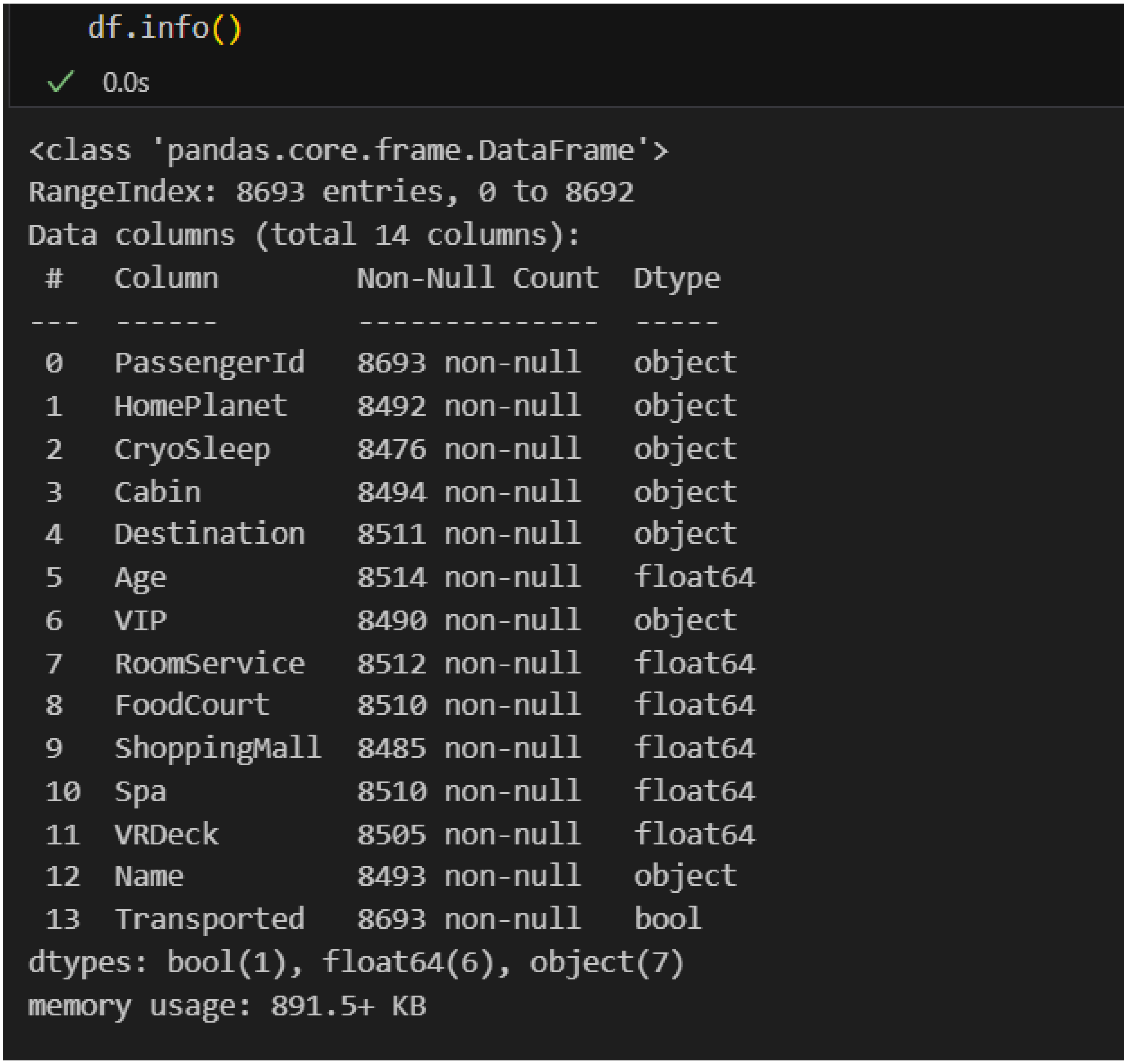
```python
df.tail(5)
```

✓ 0.0s

	PassengerId	HomePlanet	CryoSleep	Cabin	Destination	Age	VIP	RoomService	FoodCourt	ShoppingMall	Spa	VRDeck	Name	Transported
8688	9276_01	Europa	False	A/98/P	55 Cancri e	41.0	True	0.0	6819.0	0.0	1643.0	74.0	Gravior Noxnuther	False
8689	9278_01	Earth	True	G/1499/S	PSO J318.5-22	18.0	False	0.0	0.0	0.0	0.0	0.0	Kurta Mondalley	False
8690	9279_01	Earth	False	G/1500/S	TRAPPIST-1e	26.0	False	0.0	0.0	1872.0	1.0	0.0	Fayey Connon	True
8691	9280_01	Europa	False	E/608/S	55 Cancri e	32.0	False	0.0	1049.0	0.0	353.0	3235.0	Celeon Hontichre	False
8692	9280_02	Europa	False	E/608/S	TRAPPIST-1e	44.0	False	126.0	4688.0	0.0	0.0	12.0	Propsh Hontichre	True



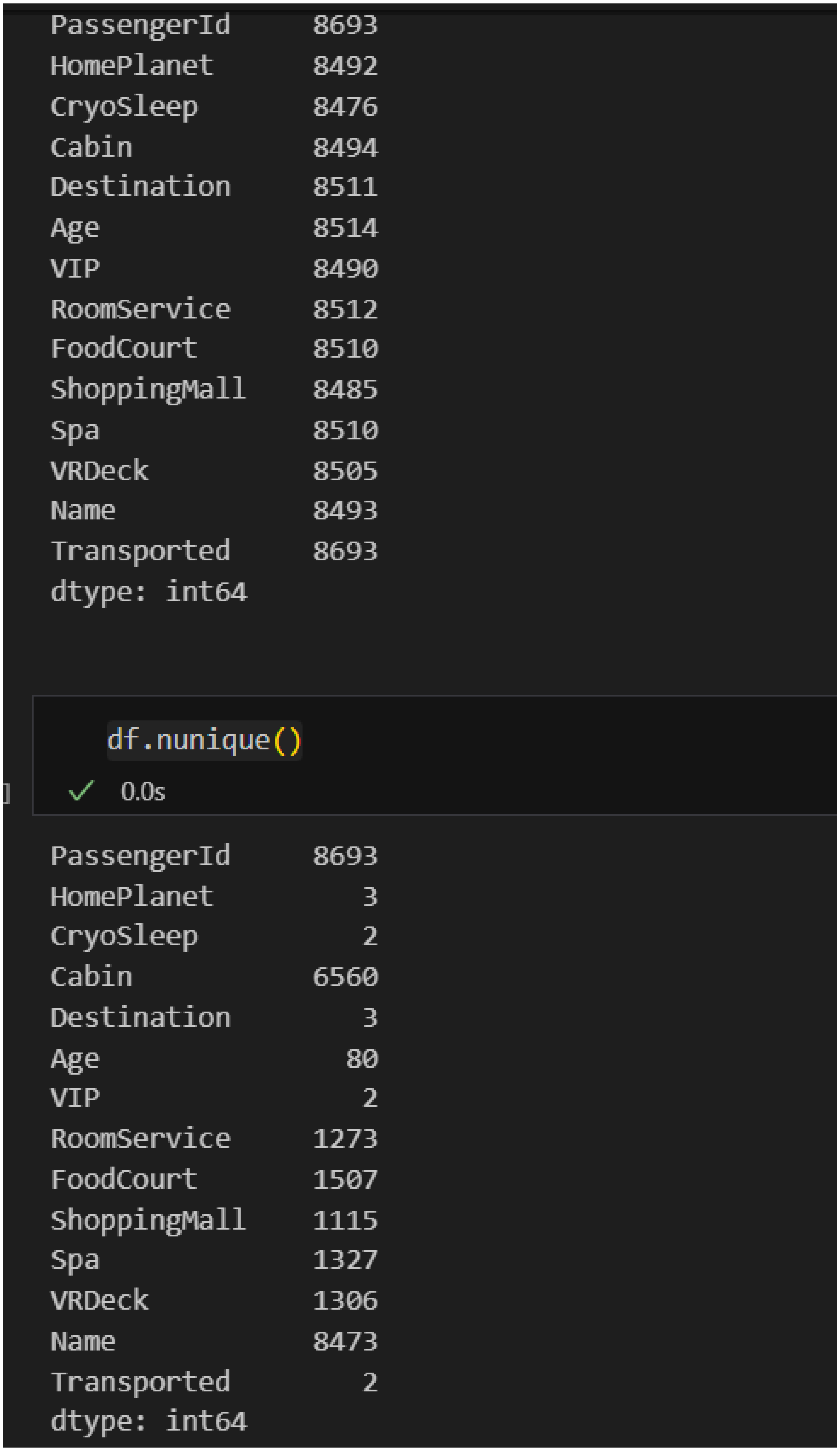
	Age	RoomService	FoodCourt	ShoppingMall	Spa	VRDeck
count	8514.000000	8512.000000	8510.000000	8485.000000	8510.000000	8505.000000
mean	28.827930	224.687617	458.077203	173.729169	311.138778	304.854791
std	14.489021	666.717663	1611.489240	604.696458	1136.705535	1145.717189
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	19.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	27.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	38.000000	47.000000	76.000000	27.000000	59.000000	46.000000
max	79.000000	14327.000000	29813.000000	23492.000000	22408.000000	24133.000000



```
df.info()
```

✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8693 entries, 0 to 8692
Data columns (total 14 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   PassengerId   8693 non-null   object 
 1   HomePlanet    8492 non-null   object 
 2   CryoSleep     8476 non-null   object 
 3   Cabin         8494 non-null   object 
 4   Destination   8511 non-null   object 
 5   Age           8514 non-null   float64
 6   VIP           8490 non-null   object 
 7   RoomService   8512 non-null   float64
 8   FoodCourt     8510 non-null   float64
 9   ShoppingMall  8485 non-null   float64
 10  Spa           8510 non-null   float64
 11  VRDeck        8505 non-null   float64
 12  Name          8493 non-null   object 
 13  Transported   8693 non-null   bool   
dtypes: bool(1), float64(6), object(7)
memory usage: 891.5+ KB
```

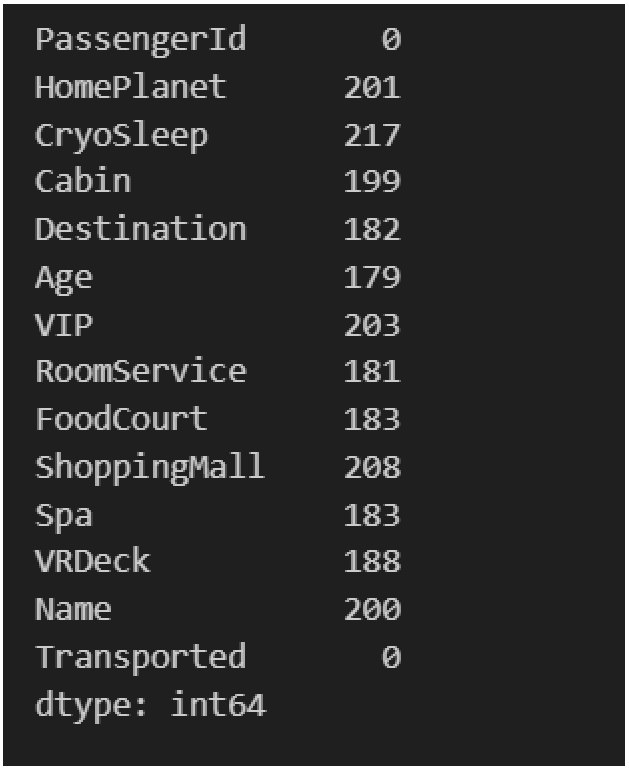


```
PassengerId     8693
HomePlanet      8492
CryoSleep       8476
Cabin           8494
Destination     8511
Age             8514
VIP             8490
RoomService     8512
FoodCourt       8510
ShoppingMall    8485
Spa             8510
VRDeck          8505
Name            8493
Transported     8693
dtype: int64
```

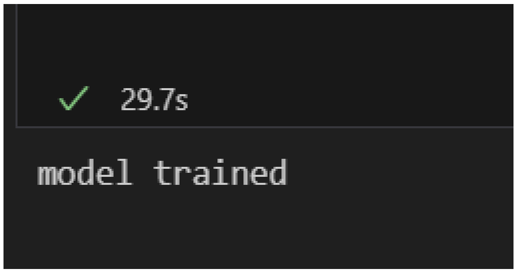
```
df.nunique()
```

✓ 0.0s

```
PassengerId     8693
HomePlanet         3
CryoSleep          2
Cabin           6560
Destination        3
Age               80
VIP                2
RoomService     1273
FoodCourt       1507
ShoppingMall    1115
Spa             1327
VRDeck          1306
Name            8473
Transported        2
dtype: int64
```

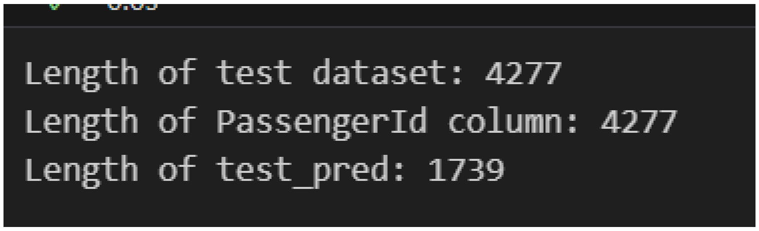


```
PassengerId       0
HomePlanet      201
CryoSleep       217
Cabin           199
Destination     182
Age             179
VIP             203
RoomService     181
FoodCourt       183
ShoppingMall    208
Spa             183
VRDeck          188
Name            200
Transported       0
dtype: int64
```



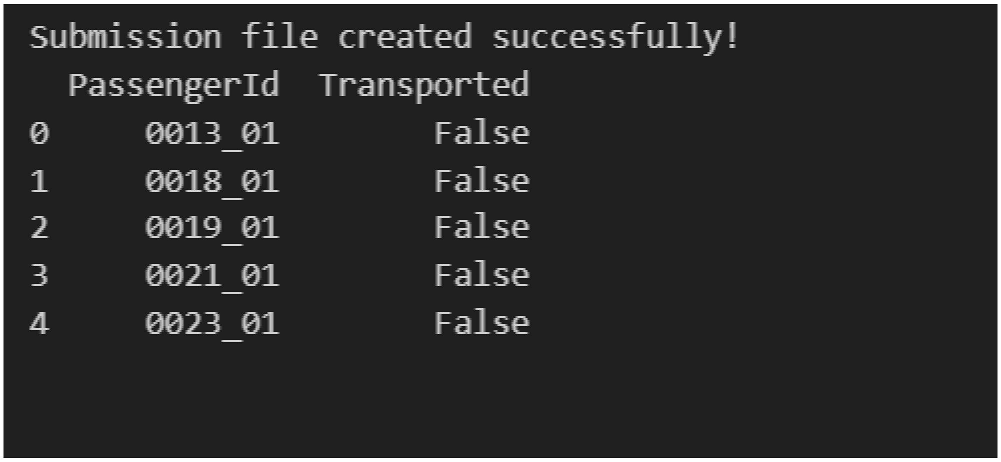
✓ 29.7s

```
model trained
```



✓ 0.0s

```
Length of test dataset: 4277
Length of PassengerId column: 4277
Length of test_pred: 1739
```



```
Submission file created successfully!
  PassengerId  Transported
0     0013_01        False
1     0018_01        False
2     0019_01        False
3     0021_01        False
4     0023_01        False
```