spaceshipproject

May 21, 2024

LOADING DATA

[8]:
```
import pandas as pd
import numpy as np

# Load the data
train_df = pd.read_csv('/content/train.csv')
test_df = pd.read_csv('/content/test.csv')

# Display the first few rows of the training data
print(train_df.head())

# Display information about the training data
print(train_df.info())

# Check for missing values
print(train_df.isnull().sum())
```

```
  PassengerId HomePlanet CryoSleep  Cabin  Destination   Age    VIP  \
0     0001_01     Europa     False  B/0/P  TRAPPIST-1e  39.0  False
1     0002_01      Earth     False  F/0/S  TRAPPIST-1e  24.0  False
2     0003_01     Europa     False  A/0/S  TRAPPIST-1e  58.0   True
3     0003_02     Europa     False  A/0/S  TRAPPIST-1e  33.0  False
4     0004_01      Earth     False  F/1/S  TRAPPIST-1e  16.0  False

   RoomService  FoodCourt  ShoppingMall     Spa  VRDeck               Name  \
0          0.0        0.0           0.0     0.0     0.0    Maham Ofracculy
1        109.0        9.0          25.0   549.0    44.0       Juanna Vines
2         43.0     3576.0           0.0  6715.0    49.0      Altark Susent
3          0.0     1283.0         371.0  3329.0   193.0       Solam Susent
4        303.0       70.0         151.0   565.0     2.0  Willy Santantines

   Transported
0        False
1         True
2        False
3        False
4         True
```

```
<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
None
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
```

PREPROCESS DATA

[2]:
```python
from sklearn.preprocessing import LabelEncoder

# Fill missing values
train_df['Age'].fillna(train_df['Age'].median(), inplace=True)
train_df['RoomService'].fillna(0, inplace=True)
train_df['FoodCourt'].fillna(0, inplace=True)
train_df['ShoppingMall'].fillna(0, inplace=True)
train_df['Spa'].fillna(0, inplace=True)
```

```
train_df['VRDeck'].fillna(0, inplace=True)
train_df['CryoSleep'].fillna(False, inplace=True)
train_df['VIP'].fillna(False, inplace=True)
train_df['HomePlanet'].fillna('Unknown', inplace=True)
train_df['Destination'].fillna('Unknown', inplace=True)

# Encode categorical variables
label_encoders = {}
for column in ['HomePlanet', 'CryoSleep', 'Cabin', 'Destination', 'VIP']:
    le = LabelEncoder()
    train_df[column] = le.fit_transform(train_df[column].astype(str))
    label_encoders[column] = le

# Drop non-essential columns
train_df.drop(columns=['Name'], inplace=True)

# Display the cleaned data
print(train_df.head())
```

```
  PassengerId  HomePlanet  CryoSleep  Cabin  Destination   Age  VIP  \
0     0001_01           1          0    149            2  39.0    0
1     0002_01           0          0   2184            2  24.0    0
2     0003_01           1          0      1            2  58.0    1
3     0003_02           1          0      1            2  33.0    0
4     0004_01           0          0   2186            2  16.0    0

   RoomService  FoodCourt  ShoppingMall     Spa  VRDeck  Transported
0          0.0        0.0           0.0     0.0     0.0        False
1        109.0        9.0          25.0   549.0    44.0         True
2         43.0     3576.0           0.0  6715.0    49.0        False
3          0.0     1283.0         371.0  3329.0   193.0        False
4        303.0       70.0         151.0   565.0     2.0         True
```

MODEL BUILDING

[3]:
```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

# Split the data
X = train_df.drop(columns=['Transported', 'PassengerId'])
y = train_df['Transported']

X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

# Train the model
model = RandomForestClassifier(random_state=42)
```

```
model.fit(X_train, y_train)

# Validate the model
y_pred = model.predict(X_val)
print(f'Validation Accuracy: {accuracy_score(y_val, y_pred)}')
```

```
Validation Accuracy: 0.7791834387579069
```

PREDICTING

[9]:
```
# Fill missing values in test data
test_df['Age'].fillna(train_df['Age'].median(), inplace=True)
test_df['RoomService'].fillna(0, inplace=True)
test_df['FoodCourt'].fillna(0, inplace=True)
test_df['ShoppingMall'].fillna(0, inplace=True)
test_df['Spa'].fillna(0, inplace=True)
test_df['VRDeck'].fillna(0, inplace=True)
test_df['CryoSleep'].fillna(False, inplace=True)
test_df['VIP'].fillna(False, inplace=True)
test_df['HomePlanet'].fillna('Unknown', inplace=True)
test_df['Destination'].fillna('Unknown', inplace=True)

# Handle unseen labels for categorical variables
def handle_unseen_labels(column, le):
    test_labels = test_df[column].astype(str)
    known_labels = le.classes_
    unseen_mask = ~test_labels.isin(known_labels)
    test_labels[unseen_mask] = 'Unknown'
    le.classes_ = np.append(le.classes_, 'Unknown')
    return le.transform(test_labels)

for column in ['HomePlanet', 'CryoSleep', 'Cabin', 'Destination', 'VIP']:
    le = label_encoders[column]
    test_df[column] = handle_unseen_labels(column, le)

# Drop non-essential columns
test_df.drop(columns=['Name'], inplace=True)

# Make predictions
X_test = test_df.drop(columns=['PassengerId'])
predictions = model.predict(X_test)

# Format the predictions
submission = pd.DataFrame({
    'PassengerId': test_df['PassengerId'],
    'Transported': predictions
})
```

SUBMITING

[10]:
```
submission = pd.DataFrame({
    'PassengerId': test_df['PassengerId'],
    'Transported': predictions
})

# Save the submission file
submission.to_csv('submission.csv', index=False)
```