Predicting Passenger Transportation Using Random Forest Model

Fathi Al Adha Hylmi
22/492195/PA/21088

Kreshnayogi Dava Berliansyach
22/496686/PA/21352

***Abstract*—This paper explores the application of a Random Forest classifier to predict whether a passenger will be transported based on various features. We detail the dataset, preprocessing steps, model training, and evaluation, demonstrating the effectiveness of machine learning in making such predictions.**

I. Introduction

Transportation in space travel is a crucial component for ensuring the safety and efficiency of passenger journeys. Predicting whether a passenger will be transported is vital for planning and optimizing resources. This study aims to develop a machine learning model to accurately predict passenger transportation based on historical data. The ability to make accurate predictions can significantly enhance operational efficiency and passenger experience in the burgeoning field of space travel.

II. Dataset and Data Understanding

The dataset used in this study comprises various features about passengers, including demographics, travel details, and onboard spending. Each feature is described as follows:

- PassengerId: A unique Id for each passenger. Each Id takes the form gggg_pp where gggg indicates a group the passenger is traveling with and pp is their number within the group. People in a group are often family members, but not always.
- HomePlanet: The planet the passenger departed from, typically their planet of permanent residence.
- CryoSleep: Indicates whether the passenger elected to be put into suspended animation for the duration of the voyage. Passengers in cryosleep are confined to their cabins.
- Cabin: The cabin number where the passenger is staying. Takes the form deck/num/side, where side can be either P for Port or S for Starboard.
- Destination: The planet the passenger will be debarking to.
- Age: The age of the passenger.
- VIP: Whether the passenger has paid for special VIP service during the voyage.
- RoomService, FoodCourt, ShoppingMall, Spa, VRDeck: Amount the passenger has billed at each of the Spaceship Titanic's many luxury amenities.
- Name: The first and last names of the passenger.
- Transported: Whether the passenger was transported to another dimension. This is the target, the column you are trying to predict.

A. Data Preprocessing

The dataset required several preprocessing steps to ensure it was suitable for machine learning:

Handling Missing Values

Numerical columns with missing values were filled with the median, while categorical columns were filled with the mode to ensure completeness. This helps maintain data integrity and allows the model to handle missing values effectively.

```
from sklearn.impute import SimpleImputer
# Numerical imputer
numerical_imputer =
SimpleImputer(strategy='median')
X[numerical_cols] =
numerical_imputer.fit_transform(X[numerica
l_cols])
# Categorical imputer
categorical_imputer =
SimpleImputer(strategy='most_frequent')
X[categorical_cols] =
categorical_imputer.fit_transform(X[catego
rical_cols])
```

Converting Categorical Variables

Categorical features were transformed into numerical format using one-hot encoding, allowing the model to process them. One-hot encoding ensures that categorical data is represented in a format suitable for the model.

```
from sklearn.preprocessing import
OneHotEncoder
# One-hot encoding for categorical
variables
encoder =
OneHotEncoder(handle_unknown='ignore')
X_encoded =
pd.DataFrame(encoder.fit_transform(X[categ
orical_cols]).toarray())
X =
X.join(X_encoded).drop(columns=categorical
_cols)
```

Normalization

Numerical features were normalized using standard scaling to ensure uniformity across features with different scales. Normalization ensures that features contribute equally to the model.

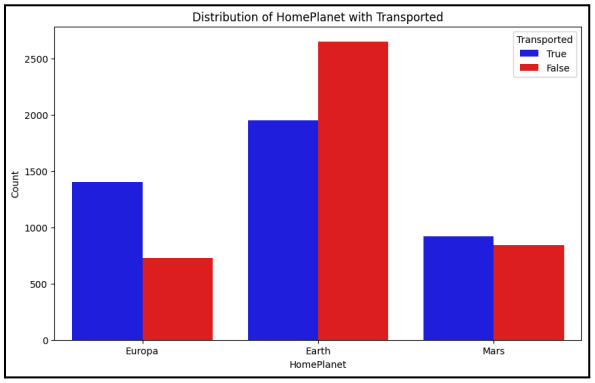
```
from sklearn.preprocessing import
StandardScaler
```

```
# Standard scaling for numerical features
scaler = StandardScaler()
X[numerical_cols] =
scaler.fit_transform(X[numerical_cols])
```

B. Data Analysis

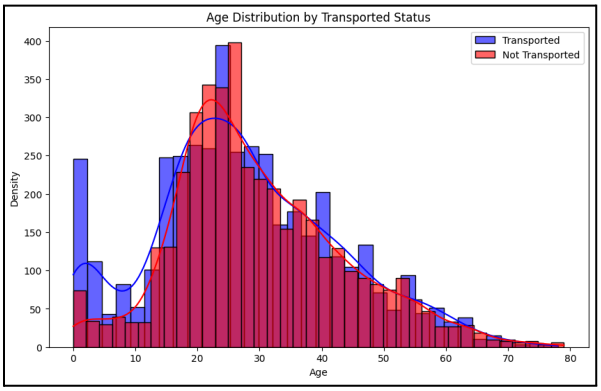
Exploratory data analysis (EDA) was conducted to understand the distribution and relationships of the features. It was observed that features such as HomePlanet, CryoSleep, and spending on services like RoomService, FoodCourt, etc., showed significant variance between transported and non-transported passengers. This analysis informed the feature selection and preprocessing steps.

HomePlanet Distribution



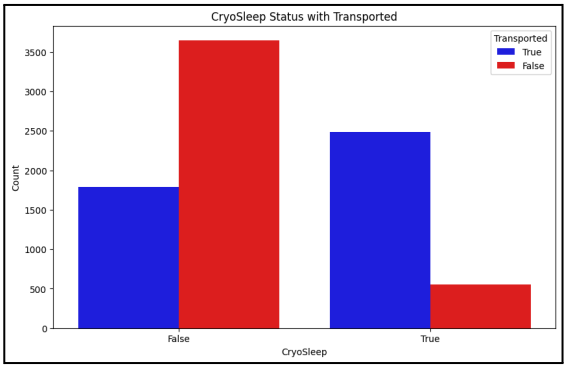
This visualization shows the distribution of passengers from different planets and their transportation status. It can be observed that Earth has the highest number of passengers, followed by Europa and Mars. The transportation status varies across these planets, with a higher proportion of transported passengers from Europa.

Age Distribution



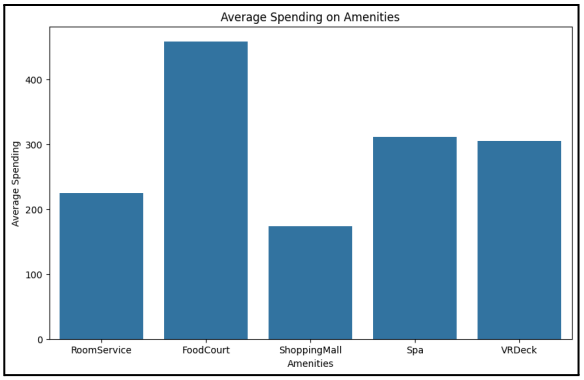
The age distribution indicates that younger passengers (age 25 and below) have a higher likelihood of being transported compared to older passengers. This could be due to various factors such as the type of travel or activities engaged in by different age groups.

CryoSleep Status



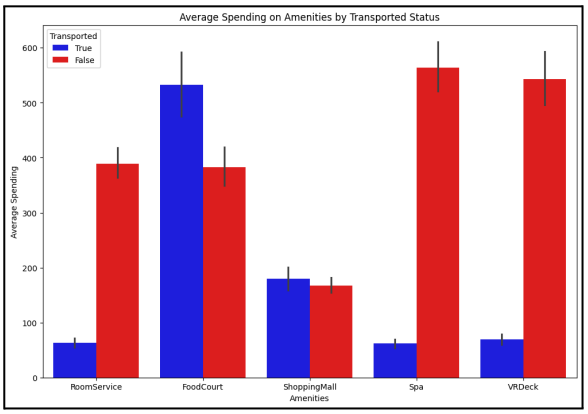
Passengers who opted for CryoSleep have a significantly higher chance of being transported. This suggests that CryoSleep could be a strong predictor for the transportation status.

Amenities Expenditure



The average spending on amenities such as RoomService, FoodCourt, ShoppingMall, Spa, and VRDeck shows variations among passengers. Higher spending might correlate with higher transportation rates, indicating a potential relationship between spending habits and transportation status.

Amenities Expenditure with Transport Status



The graph indicates the average spending on various amenities (RoomService, FoodCourt, ShoppingMall, Spa, and VRDeck) by transported status.

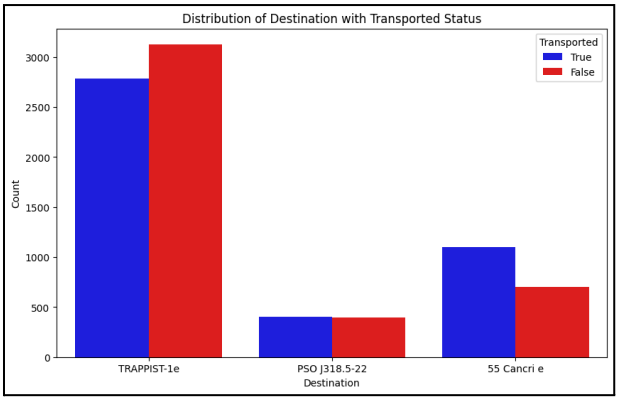
- **RoomService**: Passengers who were not transported spent significantly more on

RoomService compared to those who were transported.

- **FoodCourt**: Passengers who were transported spent considerably more on FoodCourt compared to those who were not transported.
- **ShoppingMall**: There is a slight difference, with transported passengers spending a bit more on average than those not transported.
- **Spa**: Passengers who were not transported spent significantly more on the Spa compared to those who were transported.
- **VRDeck**: Passengers who were not transported spent more on VRDeck than those who were transported.

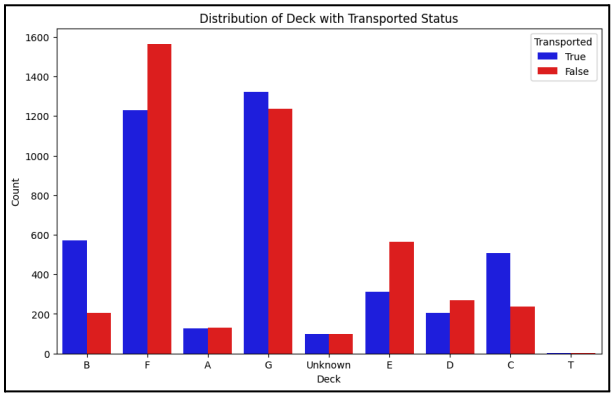
This suggests that spending habits on certain amenities may be a predictor of whether a passenger gets transported, with transported passengers generally spending more on FoodCourt and ShoppingMall, and less on RoomService, Spa, and VRDeck.

Destination Distribution



This visualization shows the distribution of passengers by their destination and transportation status. Certain destinations have higher rates of transported passengers, which could suggest that the destination is a significant factor in whether passengers get transported.

Cabin Distribution



The distribution of passengers by the deck of their cabin and transportation status can reveal if certain decks are associated with higher or lower transportation rates. This information could be useful for understanding the layout or conditions affecting transportation likelihood.

III. Machine Learning Model

The Random Forest classifier was selected due to its robustness and ability to handle both numerical and categorical data. This section details the process followed to train and evaluate the model.

A. Model Selection

Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes for classification tasks. It was chosen for its high accuracy, ability to handle overfitting, and effectiveness with mixed data types.

B. Training the Model

The dataset was split into training and testing sets, with 80% used for training and 20% for testing. A preprocessing pipeline was implemented to handle data transformation consistently across training and test datasets. The Random Forest model was then trained on the preprocessed training data.

```
from sklearn.model_selection import
train_test_split
from sklearn.ensemble import
RandomForestClassifier
from sklearn.pipeline import Pipeline
from sklearn.compose import
ColumnTransformer

# Split the data into training and testing
sets
X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.2,
random_state=42)

# Define the preprocessing steps
preprocessor = ColumnTransformer(
    transformers=[
        ('num', numerical_transformer,
numerical_cols),
        ('cat', categorical_transformer,
categorical_cols)
    ]
)

# Define the model pipeline
model = Pipeline(steps=[
    ('preprocessor', preprocessor),
    ('classifier',
RandomForestClassifier(random_state=42))
])

# Train the model
```

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

IV. Evaluation of the Machine Learning Algorithm

The model's performance was evaluated using various metrics, including accuracy, precision, recall, and F1-score on the test dataset. These metrics provide a comprehensive view of the model's predictive performance.

A. Results

The Random Forest model achieved an accuracy of 78.7%. Detailed evaluation metrics are as follows:

```
from sklearn.metrics import
classification_report, accuracy_score

# Make predictions
y_pred = model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test,
y_pred)

print(f'Accuracy: {accuracy}')
print(report)
```

False (Not Transported):

- Precision: 0.80
- Recall: 0.77
- F1-Score: 0.78

True (Transported):

- Precision: 0.78
- Recall: 0.81
- F1-Score: 0.79

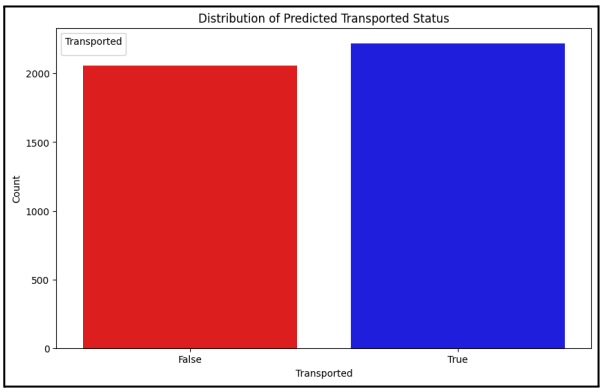
Overall:

- Accuracy: 0.79
- Macro Average Precision: 0.79
- Macro Average Recall: 0.79
- Macro Average F1-Score: 0.79

B. Discussion

The Random Forest model showed balanced performance across both classes, indicating its effectiveness in predicting transportation status. The model's precision and recall values suggest it is reliable in distinguishing between transported and non-transported passengers. The balanced F1-scores indicate the model's robustness and suitability for practical applications.

C. Final Submission



This visualization shows the distribution of predicted transported status for the test dataset. It provides an overview of how the model's predictions are distributed across the two classes (Transported and Not Transported). After submitting the predictions to the Kaggle competition, the prediction received a score of 0.78816 out of 1, indicating a fairly high accuracy in predicting the transportation status of passengers. This score aligns well with the model's F1 score of 0.79, demonstrating the consistency and reliability of the model's predictive performance.

V. Conclusion

This study demonstrates the feasibility and effectiveness of using a Random Forest classifier to predict passenger transportation in space travel. The model's robust performance suggests its potential for practical application in planning and resource optimization. Future work could explore hyperparameter tuning and feature engineering to further improve accuracy, as well as deploying the model in a real-world setting to validate its performance.

References

[1] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5-32, Oct. 2001, doi: 10.1023/A:1010933404324.

[2] A. Liaw and M. Wiener, "Classification and regression by randomForest," R News, vol. 2, no. 3, pp. 18-22, Dec. 2002. [Online]. Available: https://cran.r-project.org/doc/Rnews/Rnews_2002-3.pdf

[3] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, Oct. 2011. [Online]. Available: http://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf

[4] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, 3rd ed. Waltham, MA, USA: Elsevier, 2011.