Spacetime Anomaly Passenger Recovery

Predicting Lost Passengers of the Spaceship Titanic

Project Overview

- Binary classification problem based on
- Kaggle's Spaceship Titanic dataset.
- Goal: Predict passengers transported to an alternate dimension.
- Models used: Logistic Regression, Decision Tree, Random Forest, AdaBoost, SVM, and XGBoost.

Dataset Overview

- 14 features across 8693 rows in the training dataset.
- Features include demographics, travel details, and spending behavior.
- Binary target variable: `Transported` (True/False).

Dataset Overview

- **PassengerId**: A unique identifier for each passenger (e.g., "0001_01" where "0001" is a group identifier and "01" is an individual within the group).
- **HomePlanet**: The planet the passenger departed from (e.g., Earth, Europa, Mars).
- **CryoSleep**: A boolean indicating whether the passenger opted for cryogenic sleep during the journey.
- **Cabin**: The cabin number, including deck and side of the ship (e.g., "B/45/P").
- **Destination**: The final destination of the passenger (e.g., TRAPPIST-1e, 55 Cancri e, PSO J318.5-22).
- **Age**: The passenger's age in years.
- **VIP**: A boolean indicating whether the passenger paid for special VIP services.

- **RoomService**: The amount billed for room service.
- **FoodCourt**: The amount billed for food court purchases.
- **ShoppingMall**: The amount billed for shopping mall purchases.
- **Spa**: The amount billed for spa services.
- **VRDeck**: The amount billed for virtual reality deck services.
- **Name**: The name of the passenger (e.g., "Doe, John").
- **Transported**: The target variable indicating whether the passenger was transported to the alternate dimension (True/False).

	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

Data Cleaning

Data info:

```
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
```

Data Cleaning

We have 3 type of columns:

- Numeric columns: Age, RoomService, FoodCourt, ShoppingMall, Spa, VRDeck
- Binary columns: CryoSleep, VIP, Transported
- Category columns: PassengerId, HomePlanet, Cabin, Destination, Name

Data Cleaning

Number of uniques for each category column

Column	Number of unique values
HomePlanet	3
PassengerId	8693
Cabin	6560
Destination	3

Columns PassengerId and Name have too many unique values. So I decided to drop them

Data Cleaning

Check for missing values

Column	Missing Values	Percentage
CryoSleep	217	2.496261
ShoppingMall	208	2.39273
VIP	203	2.335212
HomePlanet	201	2.312205
Name	200	2.300702
Cabin	199	2.289198
VRDeck	188	2.16266
FoodCourt	183	2.105142
Spa	183	2.105142
Destination	182	2.093639
RoomService	181	2.082135
Age	179	2.059128
PassengerId	0	0
Transported	0	0

Data Cleaning

- Columns that have missing values: HomePlanet, CryoSleep, Cabin, Destination, Age, VIP, RoomService, FoodCourt, ShoppingMall, Spa, VRDeck
- Handling Missing Values
  - Missing categorical and binary columns (e.g., HomePlanet, CryoSleep, Destination, VIP): Imputed with the most frequent value (mode).
  - Numerical columns (e.g., Age, RoomService, FoodCourt, ShoppingMall, Spa, VRDeck): Set their values to 0

Data Cleaning

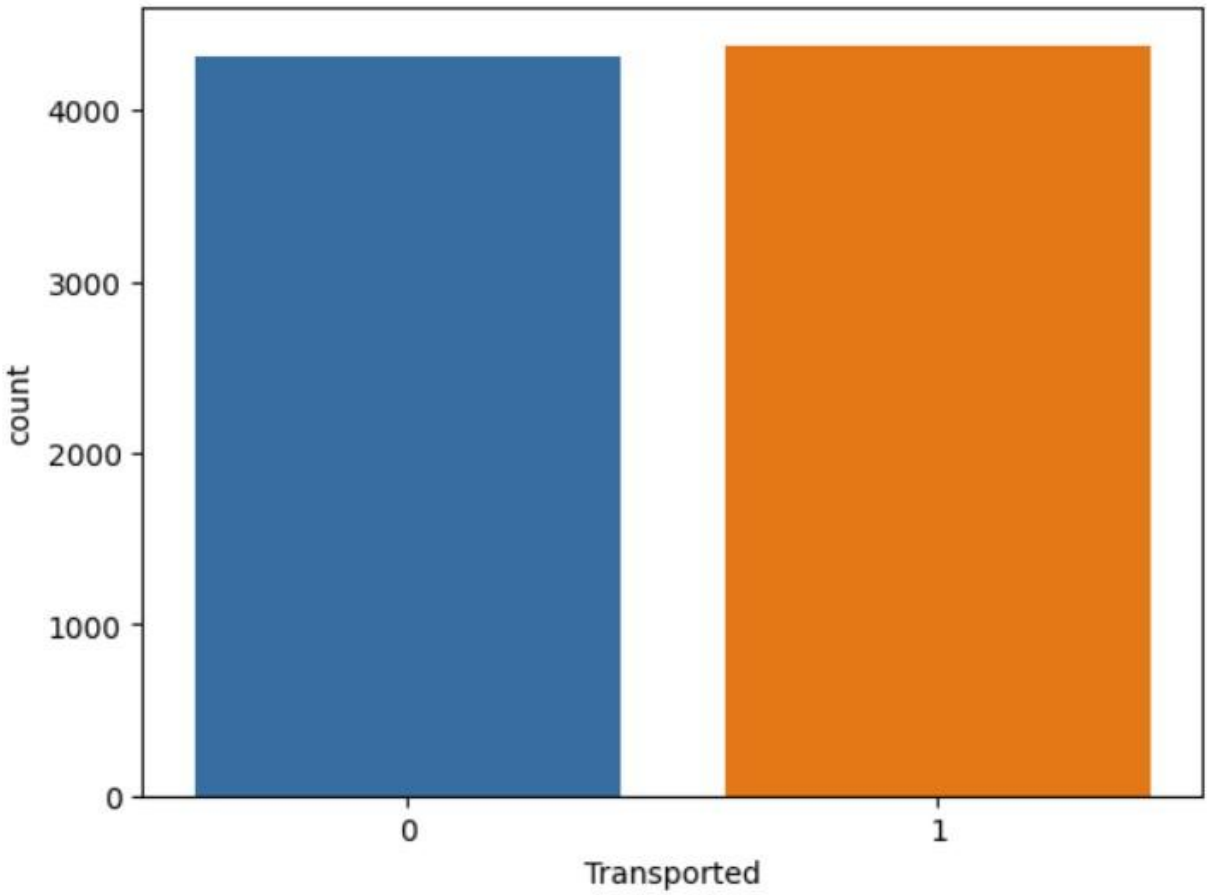
- Splitting and Standardizing the category columns
  - Split Cabin into 3 columns Deck, Number and Side using delimiter ‘/’
  - Apply One-Hot Encoding to these columns HomePlanet, Destination, Deck and Side

Data Cleaning

- Converted all features to numeric values.
  - Columns affected: VIP, Transported, CryoSleep
- Normalization
  - Scale all the features to range [0, 1] using MinMaxScaler from sklearn

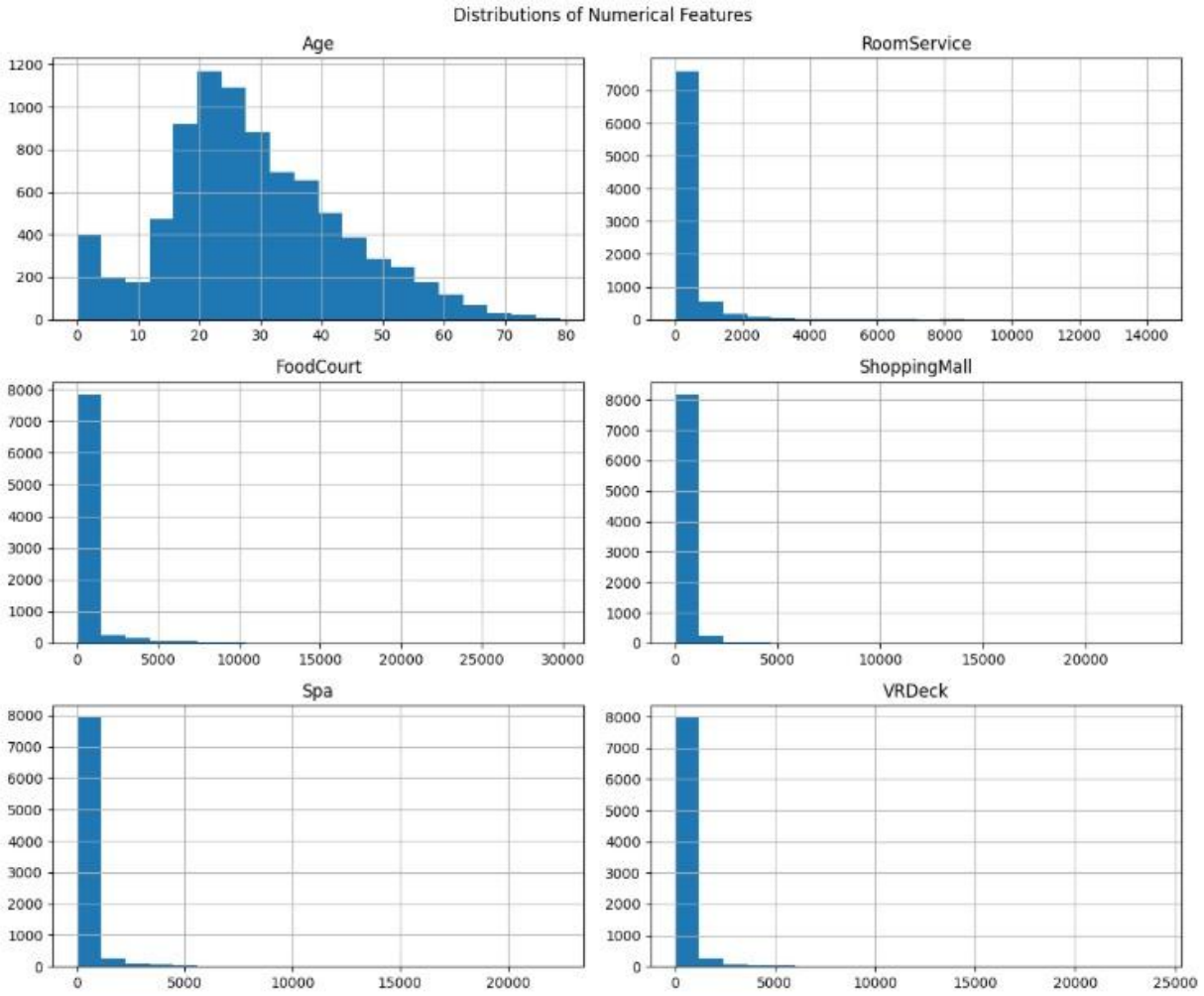
Exploratory Data Analysis (EDA)

- Target variable is balanced (~50% transported).



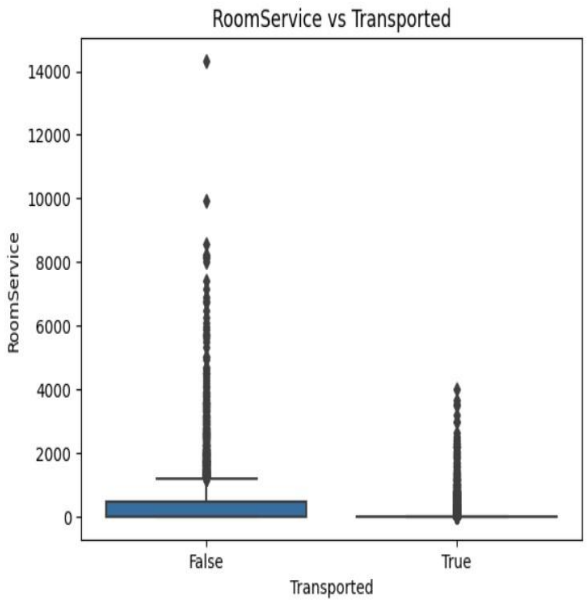
Exploratory Data Analysis (EDA)

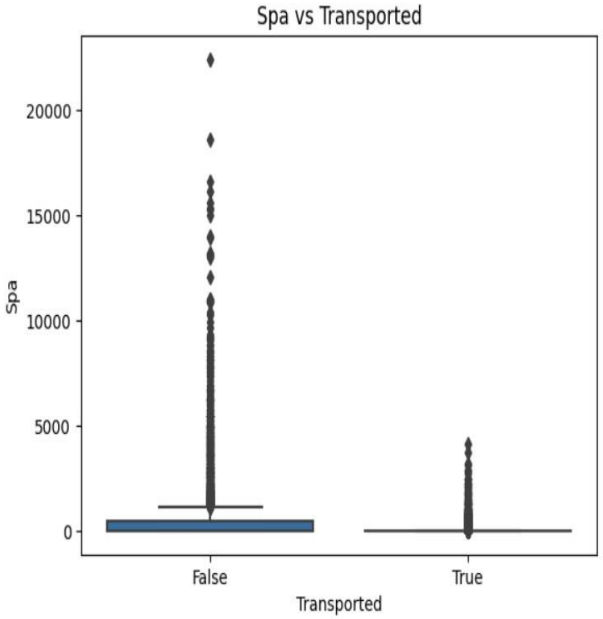
- Spending features are highly skewed with long tails.

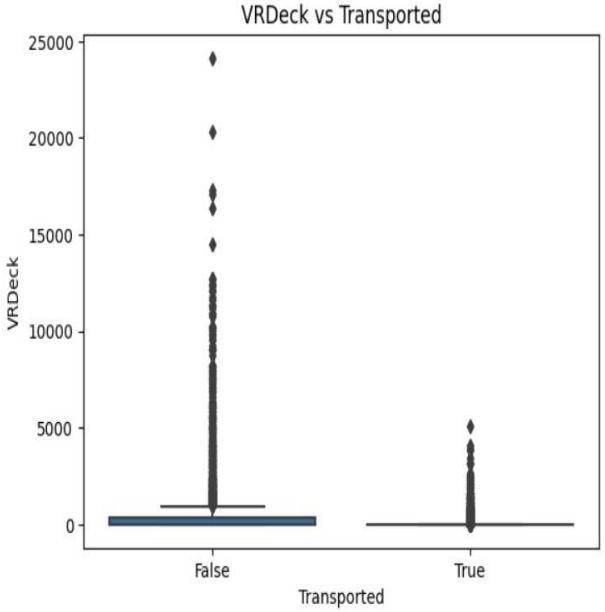


Exploratory Data Analysis (EDA)

- **Relationship between numerical features with Target Variable**
  - Passengers who spent high (> 5000) on services: `RoomService`, `Spa`, `VRDeck` were **less** likely to be transported.

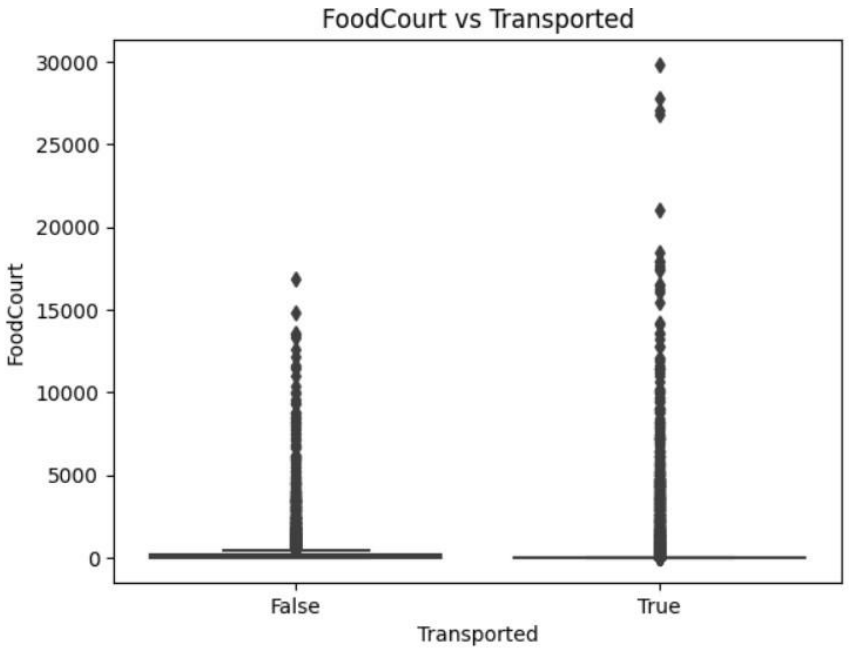


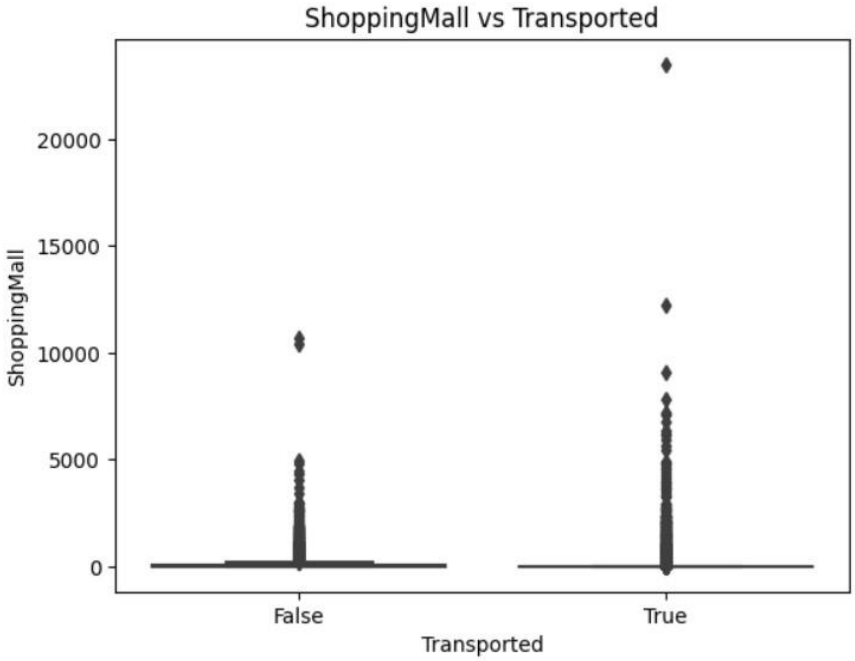




Exploratory Data Analysis (EDA)

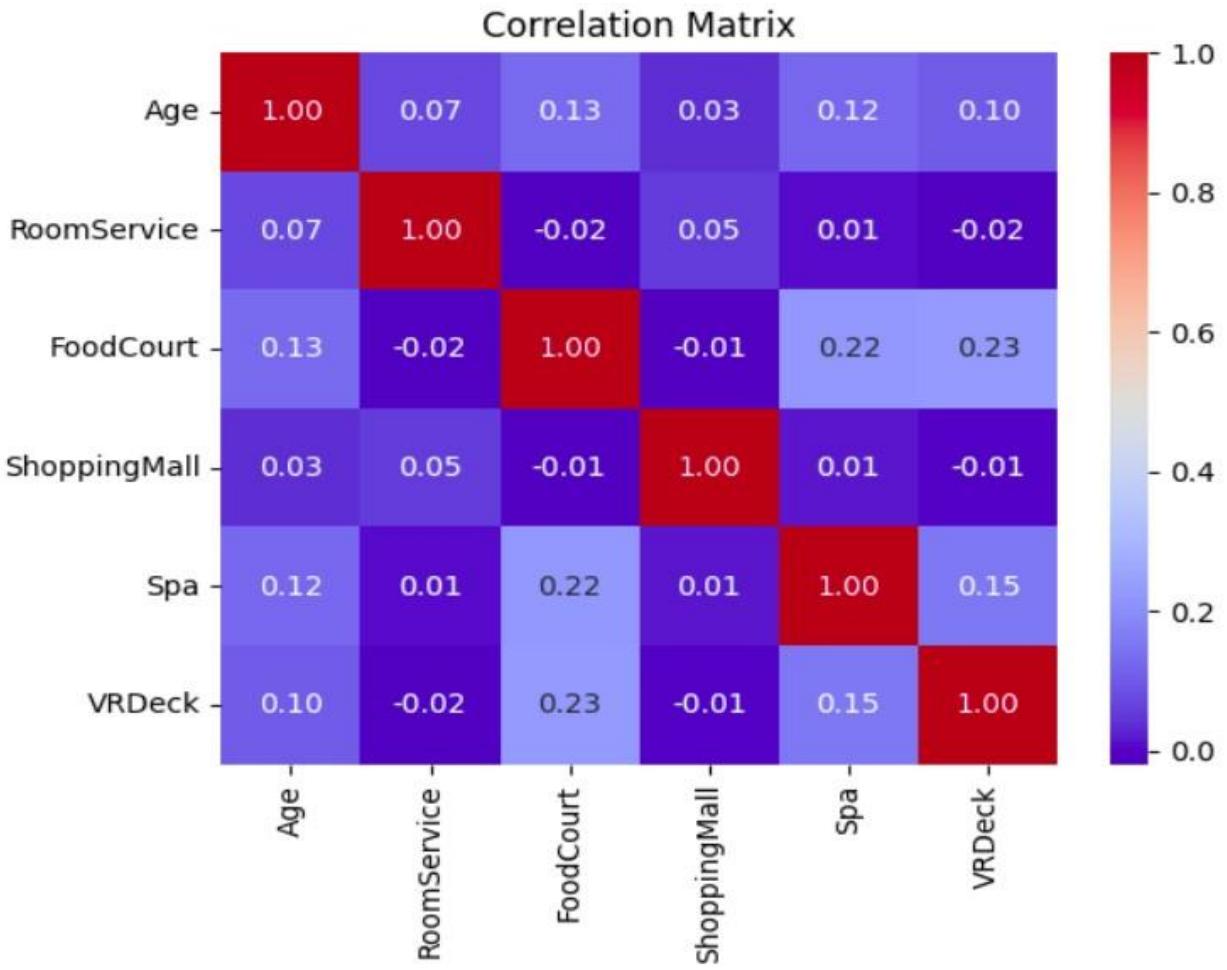
- **Relationship between numerical features with Target Variable**
  - Passengers who spent high on services: `FoodCourt > 20000`, `ShoppingMall > 14000` were **more** likely to be transported.





Exploratory Data Analysis (EDA)

- Minimal collinearity among features.



Model Selection and Optimization

- Models evaluated: Logistic Regression, Decision Tree, AdaBoost, Random Forest, SVM, and XGBoost.
- Used GridSearchCV for hyperparameter tuning.
- Techniques like regularization and depth constraints mitigated overfitting.

Model Performance

Model	Train Accuracy	Validation Accuracy	Test Accuracy
Logistic Regression	79.52%	78.26%	79.17%
Decision Tree	77.29%	76.19%	76.88%
AdaBoost	80.99%	80.16%	79.85%
Random Forest	81.00%	79.47%	80.08%
SVM	80.14%	78.95%	80.27%
XGBoost	81.19%	79.87%	80.01%

Key Insights and Recommendations

- **Best Model**: SVM slightly outperforms others on the test set, with Random Forest being a close second.
- **Trade-offs**: Random Forest provides more interpretability than SVM and might be preferred if explainability is crucial.
- **Next Steps**: Further improve models by feature engineering, especially for spending-related variables, and exploring advanced techniques like boosting variants or neural networks if computational resources allow.

Thank you