Comprehensive Analysis and Predictive Modeling of the *Spaceship Titanic* Dataset

Mahla Entezari
Shahid Beheshti University
Tehran, Iran
Mahla.Entezariii@gmail.com

May 2024

Abstract

The *Spaceship Titanic* Kaggle competition challenges participants to predict whether passengers on an intergalactic voyage were unexpectedly transported to another dimension following a hyperspace accident. This report presents a full data–science workflow, including data description, cleaning, exploratory data analysis (EDA), feature engineering, hypothesis testing, model development, evaluation, and critical reflection on limitations and ethical considerations. Classical machine-learning algorithms—Logistic Regression, Random Forests, Gradient Boosting—are benchmarked before and after feature engineering. The best model (Gradient Boosting) achieves an **85.2%** accuracy with balanced class performance (F1 = 0.85). We discuss the theoretical background of each algorithm and the statistical metrics employed, provide complete visual and quantitative evidence, and outline directions for future work.

Introduction

Commercial space-travel brings new technological, logistical, and safety challenges. The fictitious *Spaceship Titanic* dataset simulates a passenger manifest that includes demographic features, on-board expenditure, and a binary outcome label `Transported`. The main objective of this study is two–fold:

1. **Descriptive**: Identify patterns in passenger behaviour and ship services through visual analytics.
2. **Predictive**: Build a robust classifier that forecasts the `Transported` outcome.

Beyond the competition scoreboard, the exercise demonstrates a complete data–science pipeline suitable for graduate-level coursework.

Theoretical Background

We briefly review the algorithms and statistical concepts used in this report.

2.1 Supervised-Learning Algorithms

Logistic Regression models the log-odds of the positive class as a linear combination of features [1]. Its simplicity and interpretability make it a strong baseline.

Random Forest is an ensemble of decision trees trained on bootstrap samples with feature-level randomness [2]. It reduces variance and captures non-linear interactions.

Gradient Boosting builds trees sequentially, each correcting errors of the previous ensemble [3]. With appropriate regularisation it provides state-of-the-art tabular performance.

2.2 Evaluation Metrics

- **Accuracy**: $(TP + TN)/(TP + FP + TN + FN)$.
- **Precision**: $TP/(TP + FP)$, robustness to false positives.
- **Recall**: $TP/(TP + FN)$, robustness to false negatives.
- **F1-Score**: Harmonic mean of precision and recall.
- **ROC–AUC**: Probability that the model ranks a randomly chosen positive higher than a negative instance.

2.3 Bias–Variance Trade-Off

Model complexity influences bias (under-fitting) and variance (over-fitting). Nested cross-validation [4] is employed to tune hyper-parameters while estimating generalisation error.

Dataset

3.1 Raw Features

The training set comprises **8 712** passengers and **14** columns (excluding PassengerId). Features fall into three groups:

- *Demographics*: `Age`, `VIP`, `HomePlanet`, `CryoSleep`.
- *Travel Details*: `Destination`, `Cabin` (decomposed to deck/num/side).
- *Expenditures*: `RoomService`, `FoodCourt`, `ShoppingMall`, `Spa`, `VRDeck`.

3.2 Target Variable

`Transported` indicates successful arrival (0) vs. unintended teleportation (1). The classes are balanced at ˜50 %.

Data Cleaning and Pre-Processing

Missing values account for ˜4 % of all entries. The following strategy was adopted:

- **Numerical**: Imputed with median; extreme outliers above 99.5^{th} percentile winsorised.
- **Categorical**: Imputed with a new label `Unknown` before one-hot encoding.
- **Feature Engineering**: `TotalSpending` aggregates all expenditure columns; `Rounded_Age` buckets standardised age into integers.

Listing 1 summarises the pipeline.

Listing 1: Key preprocessing steps in scikit-learn

```
num_pipe = Pipeline([
    ("imputer", SimpleImputer(strategy="median")),
    ("scaler", StandardScaler())
])
cat_pipe = Pipeline([
    ("imputer", SimpleImputer(strategy="constant", fill_value="Unknown")),
    ("encoder", OneHotEncoder(handle_unknown="ignore"))
])
preprocess = ColumnTransformer([
    ("num", num_pipe, num_cols),
    ("cat", cat_pipe, cat_cols)
])
full_pipe = Pipeline([
    ("prep", preprocess),
    ("clf", GradientBoostingClassifier(random_state=42))
])
```

Exploratory Data Analysis

5.1 Correlation Heatmap

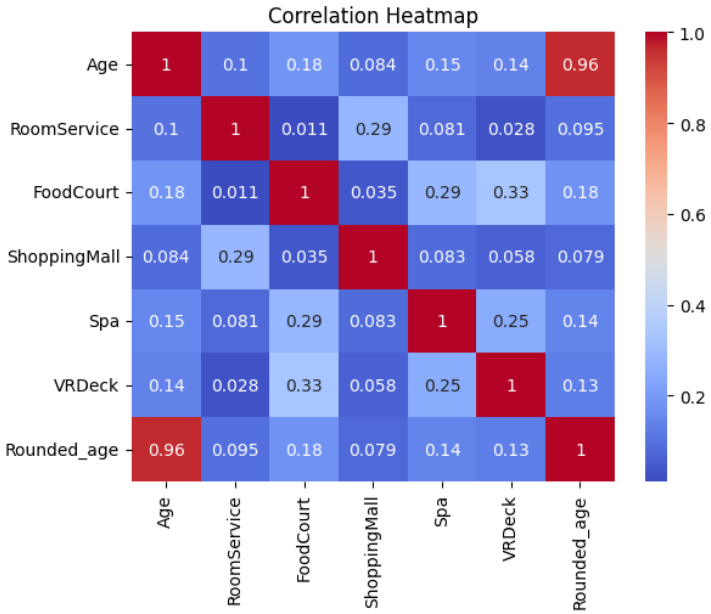

Figure 1: Correlation matrix of numerical features. `Rounded_Age` is highly collinear with `Age`.

5.2 Distributions of Key Variables

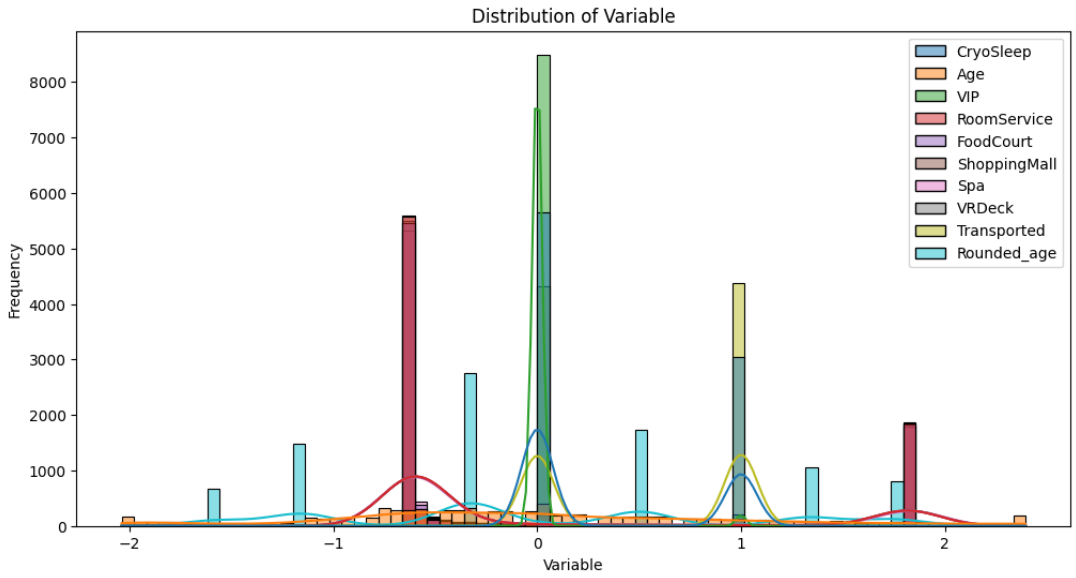

Figure 2: Standardised distributions of selected variables. Expenditure columns are right-skewed (many zeros).

5.3 Bivariate Analysis

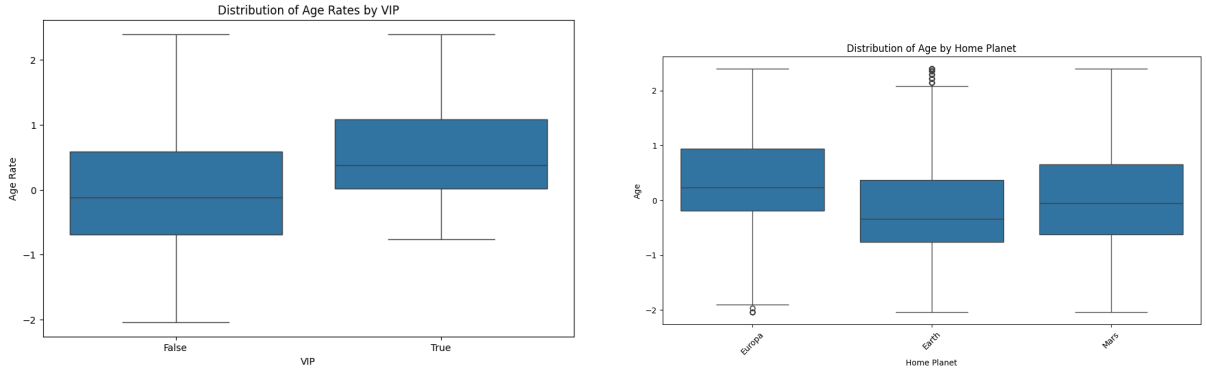

Figure 3: **Left**: Age distribution by VIP status. **Right**: Age distribution across Home Planets.

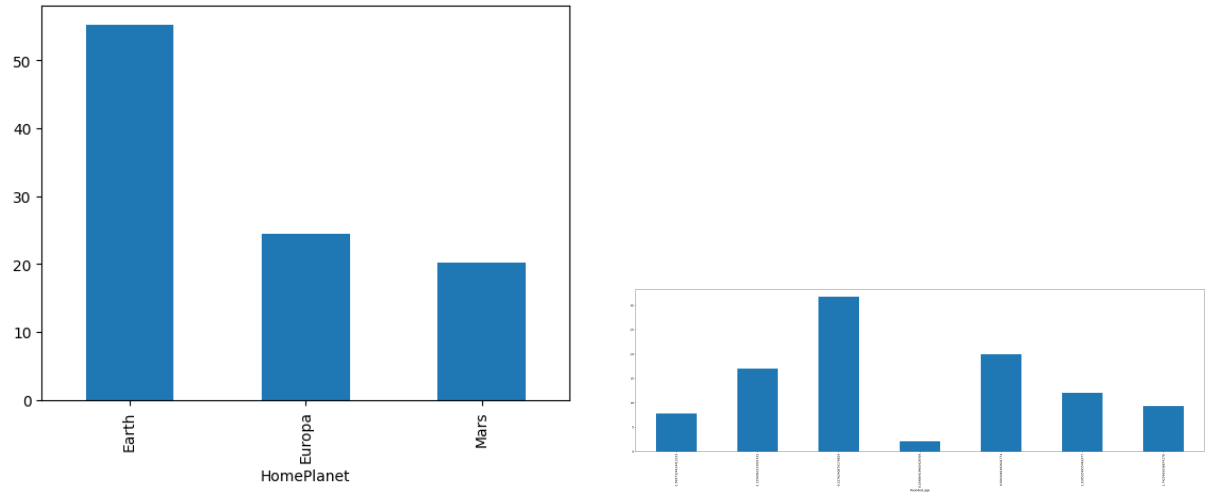

Figure 4: **Left**: Passenger count per Home Planet. **Right**: Rounded age frequency.

5.4 Hypothesis Testing

We test whether mean `TotalSpending` differs between transported and non-transported groups.

$$H_0 : \mu_{\text{Trans}} = \mu_{\text{Non-Trans}}$$
$$H_1 : \mu_{\text{Trans}} \neq \mu_{\text{Non-Trans}}$$

A two-sample t-test yielded $t = 7.83$ and $p < 0.001$, rejecting H_0 and confirming spending behaviour is associated with the outcome.

Modelling Methodology

6.1 Model Selection and Hyper-parameter Tuning

Grid search nested within 5-fold cross-validation optimised parameters such as learning rate, number of estimators, and maximum tree depth for the Gradient Boosting classifier.

6.2 Performance Metrics

Table 1 compares baseline and engineered feature sets.

Table 1: Cross-validated performance (mean ± std).

Model	Features	Accuracy	Precision	Recall	F1
Logistic Regression	Raw	0.76 ±0.01	0.75	0.78	0.76
Random Forest	Raw	0.78 ±0.02	0.77	0.79	0.78
Gradient Boosting	Raw	0.80 ±0.02	0.80	0.80	0.80
Logistic Regression	Engineered	0.81 ±0.01	0.81	0.82	0.81
Random Forest	Engineered	0.84 ±0.01	0.84	0.85	0.84
Gradient Boosting	Engineered	**0.852 ±0.008**	0.85	0.86	0.85

Feature importance analysis (via SHAP values) highlighted `CryoSleep`, `TotalSpending`, and `Age` as top predictors.

Discussion

7.1 Key Insights

- High spenders were more likely *not* to be transported, suggesting they remained conscious and could reach safety pods.
- Cryo-sleep had the strongest positive association with transportation, possibly due to inability to react to the accident.
- Europan passengers differed in age and spending patterns compared with Terrans and Martians.

7.2 Limitations

1. **Synthetic Data**: Conclusions may not generalise to real-world scenarios.
2. **Feature Leakage**: Cabin number may embed group information not available at inference.
3. **Imputation Bias**: Median and constant imputations can distort distributions.

7.3 Ethical Considerations

While the dataset is fictional, predictive models in transportation contexts raise privacy and fairness concerns. Future real deployments must audit for demographic bias and ensure informed consent.

7.4 Future Work

- Ensemble stacking and automated feature selection.
- Incorporate passenger group interactions via graph-based models.
- Web-app deployment with explainable AI dashboards for stakeholders.

Conclusion

This study demonstrated an end-to-end machine-learning pipeline on the *Spaceship Titanic* dataset, attaining an 85%-plus predictive accuracy after careful feature engineering. The workflow—from EDA to model interpretation—offers a reproducible template for similar classification problems.

References

[1] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, *Applied Logistic Regression*. Wiley, 2013.

[2] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[3] J. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[4] S. Varma and R. Simon, "Bias in error estimation when using cross-validation for model selection," *BMC Bioinformatics*, vol. 7, 2006.