Title: SRH-Spaceship-Titanic-ML

Student names: Deepshikha, Liu, Chen-Yu, Aleksandr Rykov, Francisco Saavedra

Introduction: what is the problem you are solving? Why does it matter?

The project aimed to predict passenger transport to another dimension using the fictional 'Spaceship-Titanic'(https://www.kaggle.com/competitions/spaceship-titanic/data) dataset from Kaggle, focusing on selecting the optimal model based on numerical and categorical variables. As a result, we were able to increase the accuracy of the model by increased to 0.8023.

Data + methods: what did you do?

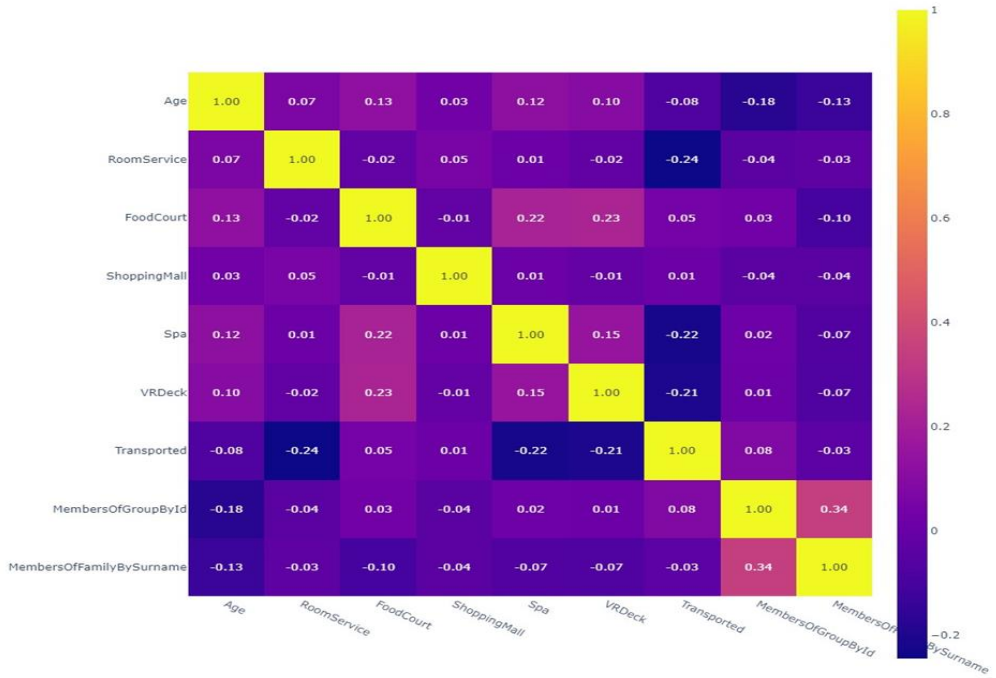
Data Extraction

In this phase, we created a new dataset with data segmentation, preparing for subsequent analysis.

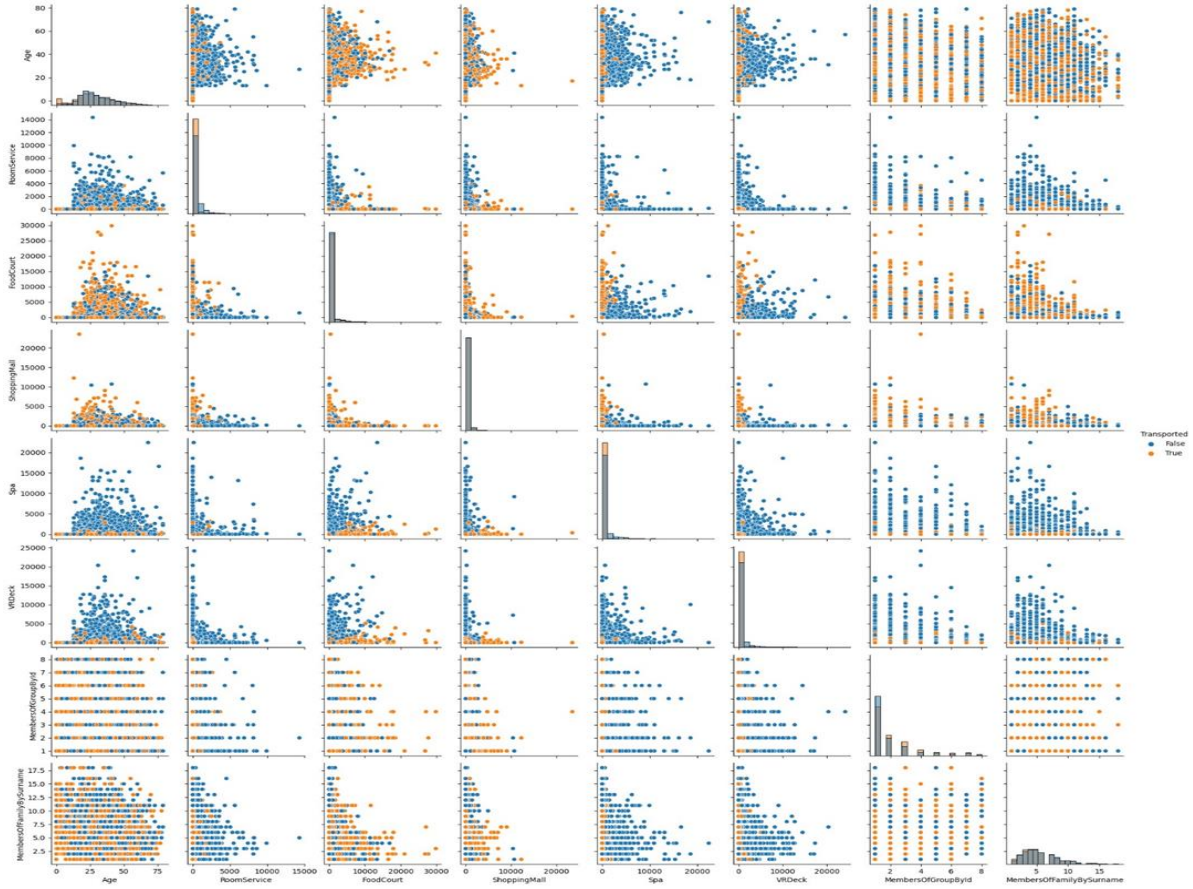
EDA

During the Exploratory Data Analysis (EDA) phase, we removed columns irrelevant to our goal and used box plots and scatter plots to identify outliers and analyze the relationship between numerical columns and the target column.

Heatmap to check the correlation between numerical columns



Scatterplot Matrix for all the numerical columns



Feature Engineering

In the feature engineering stage, we defined and handled outliers in numerical columns based on the rule of [mean - 5 standard deviations, mean + 5 standard deviations]. For columns without obvious outliers, null values were filled with the mean; for those with outliers, null values were filled with the median. Null values in categorical columns were marked as a new "Unknown" category, and obvious outliers in numerical columns were replaced with the mean plus or minus 5 standard errors. After applying one-hot encoding, the original encoded columns were removed, resulting in a dataset ready for training and testing.

Feature Selection

Utilizing permutation importance and greedy selection methods to identify crucial features that impact model performance. Permutation importance assesses the influence of feature order changes on accuracy, while greedy selection iteratively adjusts features to optimize model performance, ultimately determining the feature set for modeling.

Modeling

Employing a LightGBM classifier, the data is split into an 80% training-validation set and a 20% test set. Cross-validation is used to fine-tune the model and prevent overfitting, while the test set is reserved for the final evaluation of the model's generalization capabilities. The random_state parameter ensures experiment reproducibility.

Results: enumerate models you trained and most important scores. include 1-2 of your best plots.

In our machine learning project, we conducted a total of two modeling iterations using the LightGBM's LGBMClassifier. In the first iteration, the accuracy was below 0.8. Subsequently, based on the initial results, we optimized the hyperparameters and finalized the following configuration:

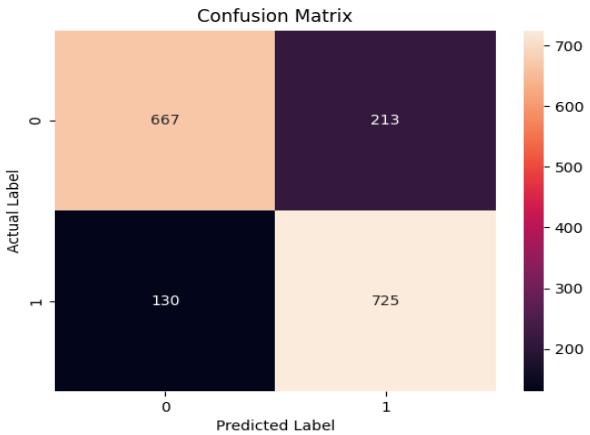
{'colsample_bytree': 0.9, 'max_depth': 10, 'n_estimators': 70, 'num_leaves': 11}

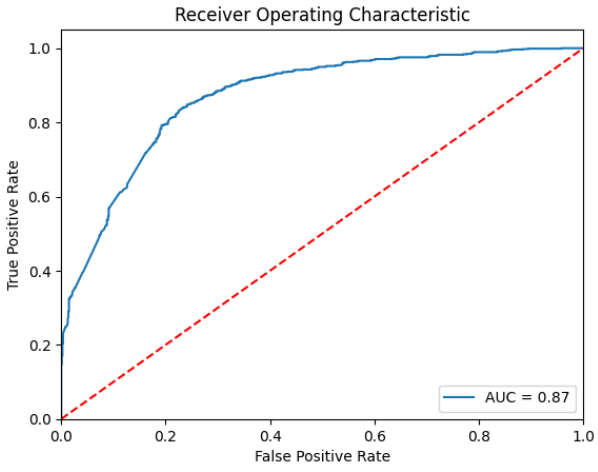
Re-training the model with this new set of hyperparameters led to better evaluation outcomes:

- Accuracy: 0.8023
- F1 Score: 0.8087
- Precision: 0.7729
- Recall: 0.8480

To ensure the reliability and comprehensiveness of the results, we further analyzed the model's performance using the following methods:

- **Confusion Matrix**: We delved deeper into the model's performance across different categories using the confusion matrix.
- **Dummy Model**: By comparing the F1 score (0.6602) of the dummy model, we confirmed that our model indeed surpassed the baseline performance.
- **ROC-AUC Curve**: An AUC value of 0.87 demonstrated the model's excellent capability in distinguishing between positive and negative classes.





Through this series of steps, not only did we optimize the model's performance via hyperparameter tuning, but we also validated the model's effectiveness and generalization ability through multi-faceted evaluations. This systematic approach provided us with a reliable framework for finding the optimal model in our machine learning project.

Short closing remark

Solving this problem involved exploring and understanding the dataset visually, handling missing values, and engineering features. It provided us with a practical scenario for feature selection and data preprocessing. Also, solving this problem allowed us to evaluate different machine learning models in terms of accuracy and other classification metrics. As a result, it helped us in understanding the strengths and limitations of various algorithms and discussed their accuracy and implementation as a team.