Adam Erler
DSC 680
Project 2 Milestone 2
White Paper: Predictive Modeling for Spaceship Titanic

1. Business Problem:

The primary business challenge is to build a predictive model for the "transported" status of passengers aboard the Spaceship Titanic. This exercise aims to discern patterns and characteristics linked to the transportation outcome, while also examining how different features, including passenger class, age, and fare, can influence this result. Successful predictions can assist in creating hypothetical scenarios, offering insights by comparing with historical events, and deepening the understanding of passenger experiences.

2. Background/History:

Historically, the Titanic dataset has been a focal point for numerous predictive modeling activities due to its multifaceted data and the intriguing historical narrative it offers. With the Spaceship Titanic dataset, we're venturing into a realm where historical data meets science fiction, creating an opportunity to explore from a fresh, imaginative angle. This project is presented as part of a Kaggle competition, a renowned platform for data science learning and challenges, hosting myriad courses and contests for aspirants and professionals alike.

3. Data Explanation:

The initial approach to Exploratory Data Analysis (EDA) centered on identifying value distributions, the existence of null values, and spotting outliers. As observed in figure 1, some numerical columns exhibit outliers. Besides the passenger ID and transported values, every other column, be it numerical or categorical, has null values. These nulls seem to be randomly distributed, as deduced from figures 2 and 3.

To address these issues, I adopted a strategy that used mean and mode for imputation, ensuring the preservation of data quality. Subsequently, while names were omitted from the modeling process, titles extracted from them might shed light on societal or professional structures, potentially impacting transportation outcomes. Lastly, I adapted the data for the model by implementing one-hot encoding for categorical variables and normalizing to equalize the scale across features.

For a detailed breakdown of the data, refer to Appendix B.

4. Methods:

Gradient Boosting was the chosen technique for the final model. This ensemble method, which constructs trees in sequence with each tree aiming to rectify its predecessor's errors, was favored due to its adeptness at handling varied data types and its proven track record in diverse predictive challenges. To sharpen the model, hyperparameter tuning was executed, tweaking parameters such as tree depth and learning rate.

This model was settled upon after benchmarking its performance against XGBoost, LGB, and Random Forests. Notably, traditional Gradient Boosting outperformed the others post hyperparameter tuning. Performance metrics for LGB, Random Forests, and XGBoost are available in Appendix C.

5. Analysis:

The analysis commenced with a comprehensive EDA to unravel data patterns, distributions, and potential correlations. The subsequent phase involved establishing a base Gradient Boosting model, which was then fine-tuned through hyperparameter adjustments. The performance evaluation of the Gradient Boosting Model is illustrated in Appendix A, figures 4 and 5.

6. Conclusion:

The Gradient Boosting model demonstrated notable predictive prowess, emerging as a reliable instrument for forecasting transportation outcomes on the Spaceship Titanic. However, there's room for further refinement and research to perfect the model's predictive acumen.

7. Assumptions:

Data Representativeness: The dataset is a representative sample, ensuring the model's insights are both valid and applicable.
Feature Relevance: All features in the dataset, including derived attributes like titles, are deemed significant and influential in predicting transportation outcomes.

8. Limitations:

Despite its strength, the inherent complexity of Gradient Boosting may hamper straightforward interpretation, which could hinder its applicability in situations demanding clear interpretability.

9. Challenges:

Managing data, especially after one-hot encoding, presented challenges tied to model intricacy, interpretability, and computational demands. Running multiple models to gauge accuracy sometimes stretched the limits of my computer's computational capacity, exemplified by the hyper-tuning process faltering multiple times due to its exhaustive nature.

10. Future Uses/Additional Applications:

An ensemble of models might enhance performance in future iterations. Although Gradient Boosting was the standout performer individually, the margin of superiority over other models wasn't substantial.

13. Ethical Assessment:

I initiated this project using a base notebook from another Kaggle participant. To maintain transparency and fairness, I acknowledge that notebook (link) as a foundation for my ideas. While I revamped nearly every aspect, I also consulted Chat GPT to troubleshoot model performance issues. In real-world application, it's paramount to eliminate subjective values like names, ensuring predictions are unbiased. Given the model's potential role in real-time rescue missions, there's an ethical onus to uphold the highest standards of research and analysis.
Audience Questions:

Data Understanding:

What is the context and objective of gathering this dataset, and what do we aim to predict or infer? The context is a online competition to learn and practice problem solving skills in data science.

Feature Explanation:

Can you provide detailed descriptions and importance of each feature variable in the dataset? See the attached data dictionary.

Data Quality:

How were missing and outlier values dealt with during the data preprocessing stage, and why were those methods chosen? Outliers were normalized to prevent them from causing issues with the models, missing values were imputed.

Model Selection:

Why was the Gradient Boosting Classifier chosen as the model for this project, and were alternative models considered? Gradient Boosting was chosen since it had the best performance on the confusion matrix.

Model Performance:

How does the model perform in terms of various metrics (like accuracy, precision, recall, F1-score, etc.) and what do these metrics convey about the model's predictive capability?

Accuracy: The model correctly predicts the target variable for roughly 81%
81% of the validation set. The training and validation accuracies are close, suggesting that the model is not overfitting.

Precision: Precision tells us how many of the predicted positive instances are actually positive. For Class 0, 83% of the instances predicted as Class 0 are truly Class 0. Similarly, 79% of the instances predicted as Class 1 are truly Class 1.

Recall: Recall tells us how many of the actual positive instances are predicted correctly. For Class 0, the model identifies 77%of all actual Class 0 instances. For Class 1, the model identifies 84% of all actual Class 1 instances.

F1-Score: The F1-score is the harmonic mean of precision and recall. It's a good metric when the class distribution is unbalanced. Values closer to 1 are better, and both classes have F1-scores around 0.800.80, indicating a balanced performance between precision and recall for both classes.

Feature Engineering:

How were the new features, such as 'Title', engineered and what impact do they have on the model’s predictive performance? In the current state this a process that can be improved upon. It is hard to extract a features performance with present encoding method.

Class Imbalance:

Is there a class imbalance in the 'Transported' target variable, and if yes, how was this addressed during modeling? No, from the data transported was a fairly balanced variable.

Model Interpretability:

How can the predictions of the Gradient Boosting model be interpreted, especially given its ensemble nature? This is a place where there is room for improvement for the model. Direct interpretability from an ensemble is difficult and benefits from creating surrogate models to understand further.

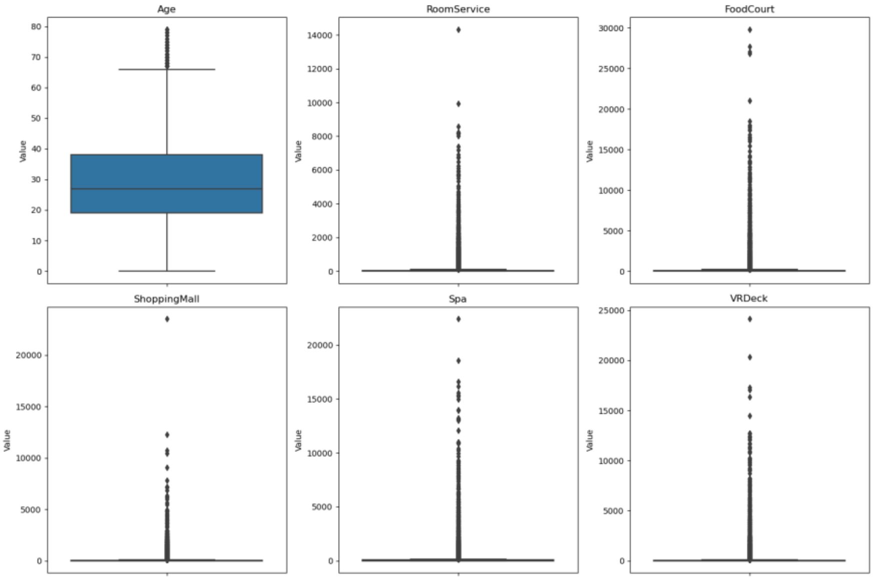
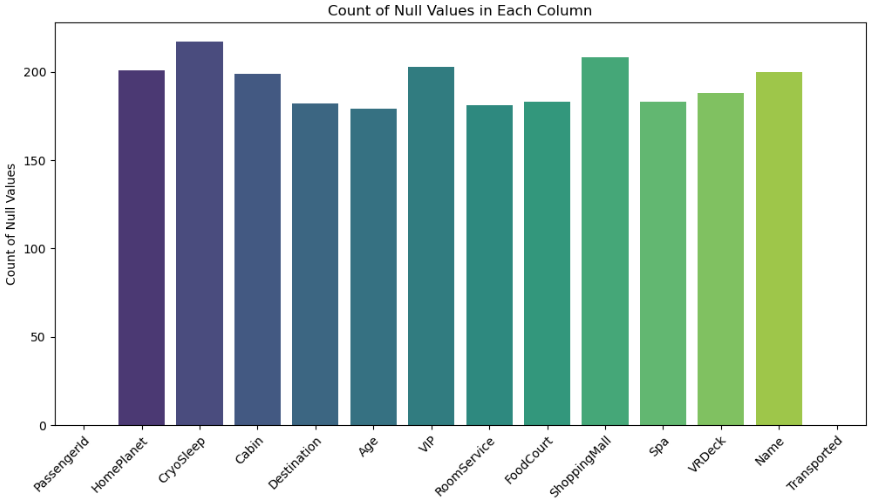
Model Validation:

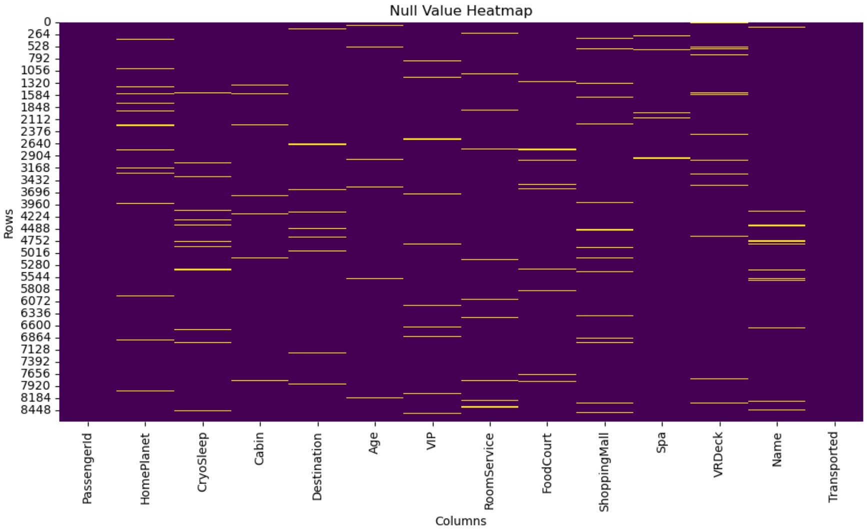
How was the model validated, and why was a particular split ratio chosen for the train-test split? 80/20 is a standard split to check for possible overfitting/underfitting.

Deployment and Usage:

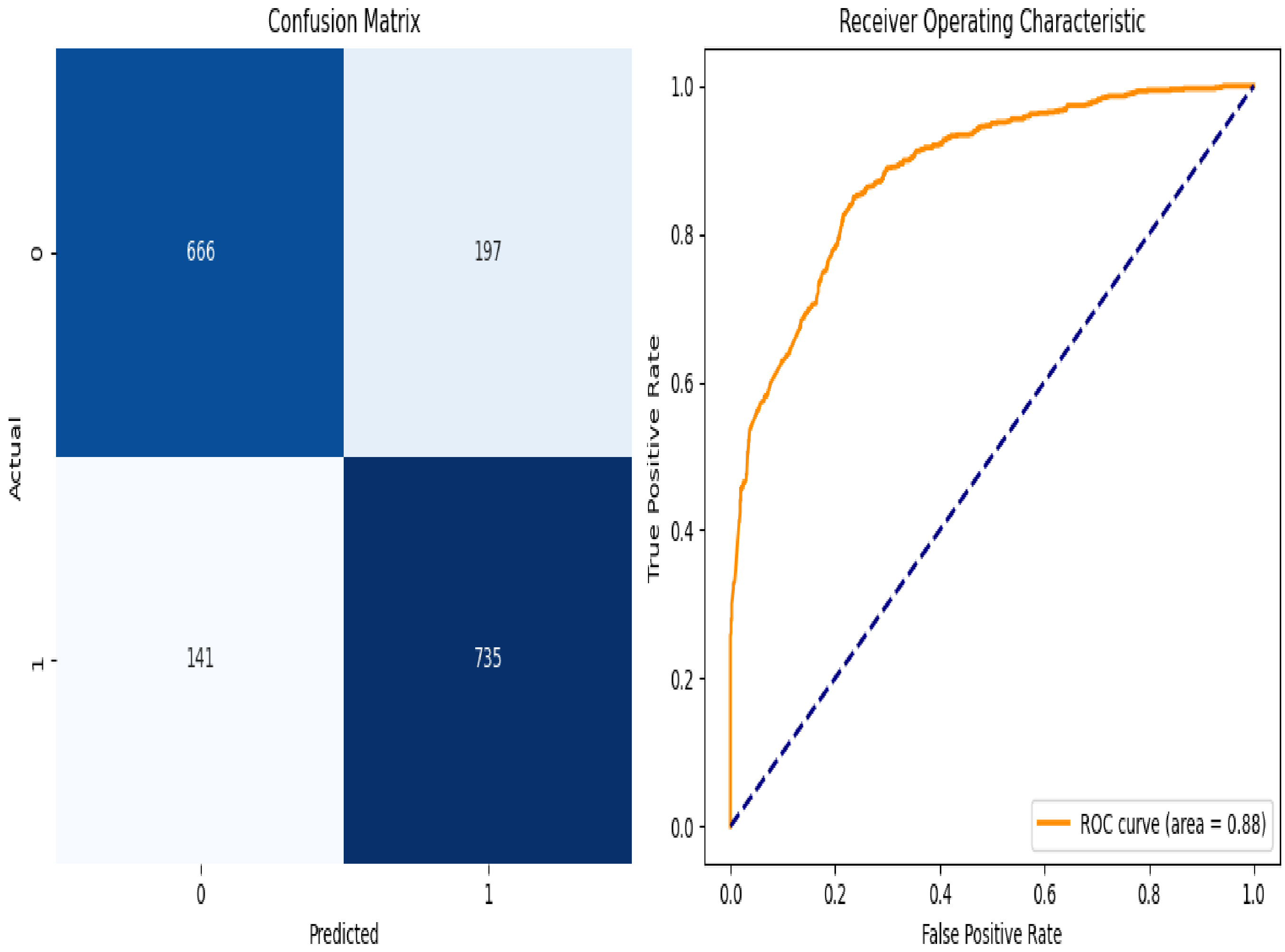
How can the developed model be deployed into a production environment, and what kind of system requirements or dependencies does it have? This model could be used as a case study to explore similar situations. But with this being a fictional situations such transpositions should be executed with caution.

Appendix A: Figures

Description	Figure
1: Box plots of the distribution of numerical values.	
2: Counts of nulls or missing values by column	

3:Heat map to visualize distribution of null or missing values.	

Figures 4 and 5: Gradient Boosting Performance.



Appendix B: Data Dictionary

Feature	Type	Description	Processing
PassengerId	Categorical/Nominal	Unique identifier for each passenger	None
HomePlanet	Categorical/Nominal	Home planet of the passenger	Missing values replaced with "Missing"
CryoSleep	Categorical/Boolean	Indicates if the passenger was in cryo sleep	Converted to string, missing values replaced with "Missing"
Cabin	Categorical/Nominal	Cabin and seat information of the passenger	Missing values replaced with "Missing"
Destination	Categorical/Nominal	Destination planet of the passenger	Missing values replaced with "Missing"
Age	Numerical/Continuous	Age of the passenger	Missing values imputed with median
VIP	Categorical/Boolean	Indicates if the passenger is a VIP	Converted to string, missing values replaced with "Missing"
RoomService	Numerical/Continuous	Metric related to room service	Missing values imputed with median
FoodCourt	Numerical/Continuous	Metric related to the food court	Missing values imputed with median
ShoppingMall	Numerical/Continuous	Metric related to the shopping mall	Missing values imputed with median
Spa	Numerical/Continuous	Metric related to the spa	Missing values imputed with

			median
VRDeck	Numerical/Continuous	Metric related to the VR deck	Missing values imputed with median
Name	Categorical/Nominal	Name of the passenger	Used to extract "Title", missing values replaced with "Missing"
Transported	Categorical/Boolean	Indicates if the passenger was transported	None
Title (Derived)	Categorical/Nominal	Title extracted from the name of the passenger	Extracted from "Name", missing values assigned "Unknown"

Appendix C: Code