**Final Project Report**

College of Professional Studies, Northeastern University

Fall 2022’B

ALY6040 Data Mining Applications

**Introduction**

We could immediately conjure up dark images of spies or hackers acquiring access to personal information when we hear the term "data mining." However, the truth is that data mining is important and helpful in our day-to-day lives. Through data mining, experts and academics can learn how to support humanitarian initiatives in many different countries. They can learn more about discrimination, climatic change, the spread of diseases, and other subjects. Without data mining, gathering the information needed to develop forecasts and handle global concerns would take months or even years. Worldwide, businesses use data mining for projects with a range of commercial applications and repercussions.

To put it simply, data mining is a technique used by companies to turn raw data into informative data. They use algorithms to find patterns in vast data sets and develop a better understanding of their clientele. It takes data from data sets, compares it, and assists the organization in making decisions. Long-term, this helps with strategy formulation, revenue growth, effective marketing, and other things.

Data mining and machine learning are commonly mistaken with data mining, but these concepts are all very distinct. Both data mining and machine learning use patterns and analytics, however whereas machine learning goes further to predict future events using the data, data mining focuses on finding patterns that already exist in the data. Data mining "rules" or trends aren't always clear. In many instances of machine learning, the computer is given a rule or variable to aid in understanding the data. In contrast to machine learning, which is intended to be begun by a person and then learn on its own, data mining also rely on human judgment and interaction. There are many similarities between data mining and machine learning, and data mining typically automates machine learning approaches.

Having stated that, we investigate several data mining approaches with a data set in this final project report and carry out a variety of tasks to help us gain insights and make predictions.

**Dataset Description**

This dataset was retrieved from Kaggle and is used in an ongoing Kaggle competition. The competition involves predicting whether a passenger was transported to an alternate dimension during the Spaceship Titanic's collision with the spacetime anomaly. The reason why this Dataset was chosen because as an aspiring Data Scientist, I felt that I have to learn, gain the required skills and practice a lot to get more experience. And I felt that participating in data science competitions to be one of the best approaches to help beginners in data science get more experience and finally apply for job opportunities. Below mentioned are the Files retrieved and Data Field Descriptions.

**Train set**

**Filename:** train.csv

**train.csv** - A total of 8700 passengers' personal records were collected, to be used as training data.

* **PassengerId** - One unique Id for every traveler. Each Id has the following format: gggg pp, where pp is the passenger's position inside the group and gggg denotes the group they are traveling with.  People in a group are often family members, but not always.
* **HomePlanet** - The planet from whence the traveler departed, usually their home planet.
* **CryoSleep -** Whether the traveler chose to be placed in suspended animation for the duration of the trip is indicated. Cryosleeping passengers are confined to their cabins.
* **Cabin** - The passenger's assigned cabin number. , where side can be either P for Port or S for Starboard, takes the form deck/num/side.
* **Destination** - The planet from which the passenger will disembark.
* **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 that the traveler has paid for each of the luxurious amenities on the Spaceship Titanic.
* **Name -** The first and last names of the passenger.
* **Transported -** Whether the traveller was taken to another dimension. The column we are attempting to forecast is the target, otherwise known as the target.

**Test set**

**Filename:** test.csv

**test.csv -** The personal information of the remaining 4300 passengers will be used as test data.

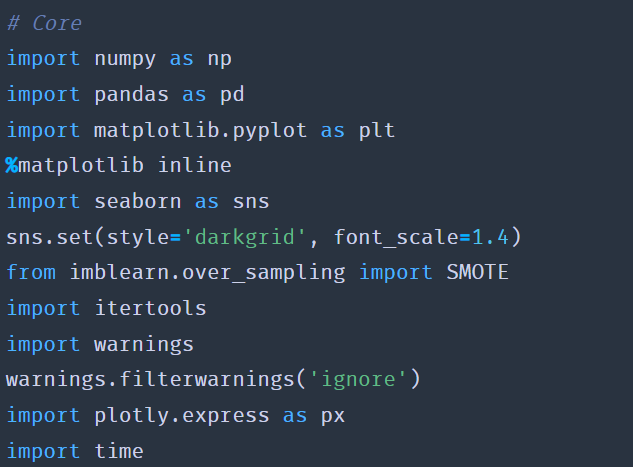
**Methodology**

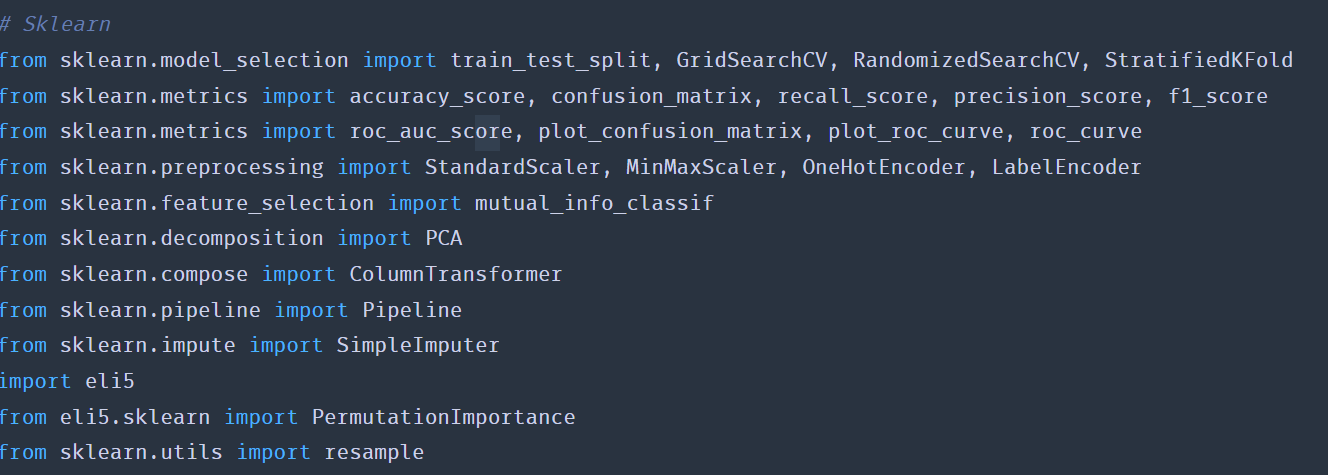
In this project report, we will cover the following:

* Exploratory Data Analysis
* Feature Engineering
* Data Cleaning
* Encoding, Scaling and Pre-processing
* Training Machine Learning Models
* Cross Validation and Ensemble Predictions

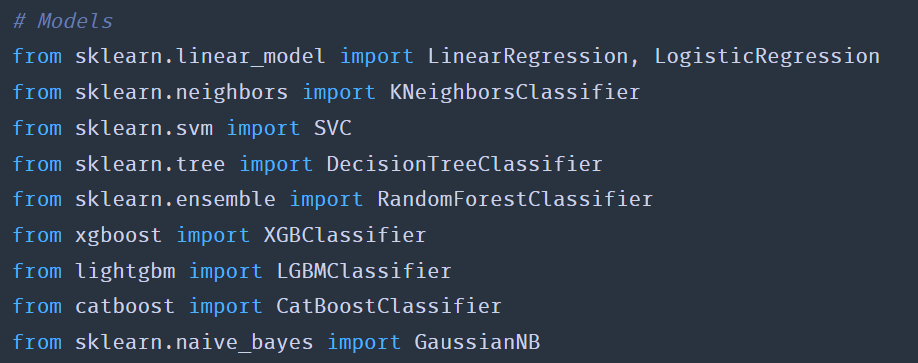
**Exploratory Data Analysis:**

Exploratory data analysis is one of the strategies used in data science nowadays that is most successful. The core of EDA is the transformation of the supplied data through the use of statistical modelling and visualization techniques. As always, we begin with importing the required packages, which is described in the two code snippets below.



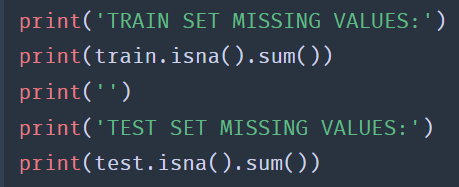


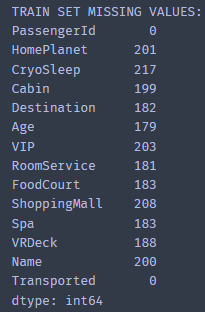
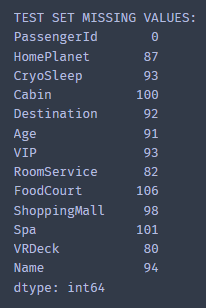
And we use the below listed packages for our models.



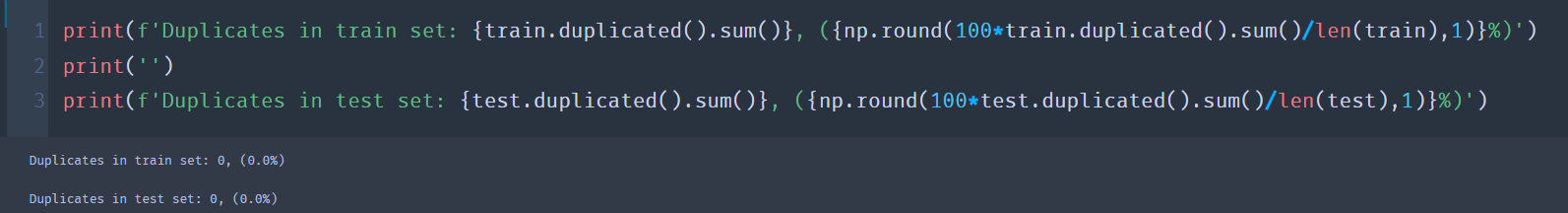
**Missing Values**

After loading the dataset, in the initial analysis, I attempted to find if there are any missing values and found that almost every feature has missing values. It is illustrated in the code snippet below.



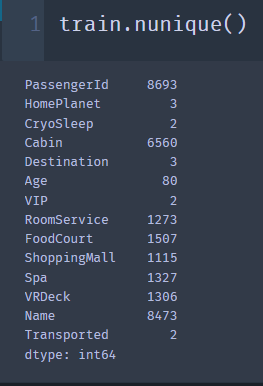
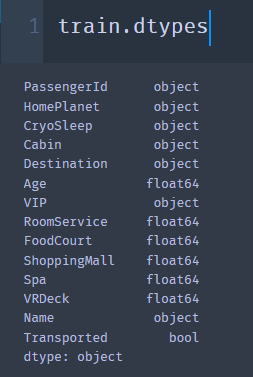
 

And upon further analysis, we found that there are no duplicates.

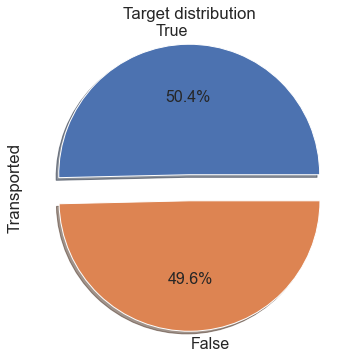


**Cardinality of features:**

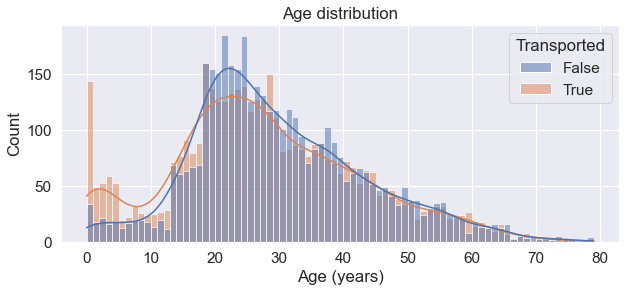
There are 6 continuous features, 4 categorical features (excluding the target) and 3 descriptive/qualitative features

It is required to transform the data to be numeric (int64 or float64) so that we can train machine learning models as they generally don't work on text. We move on to actually exploring our dataset to get some more insights, so we plotted our target distribution. It can be inferred from the below plot.



Since the target is highly balanced, so we luckily don't have to consider techniques like under/over-sampling. And so we moved on to plot a histogram based on age.

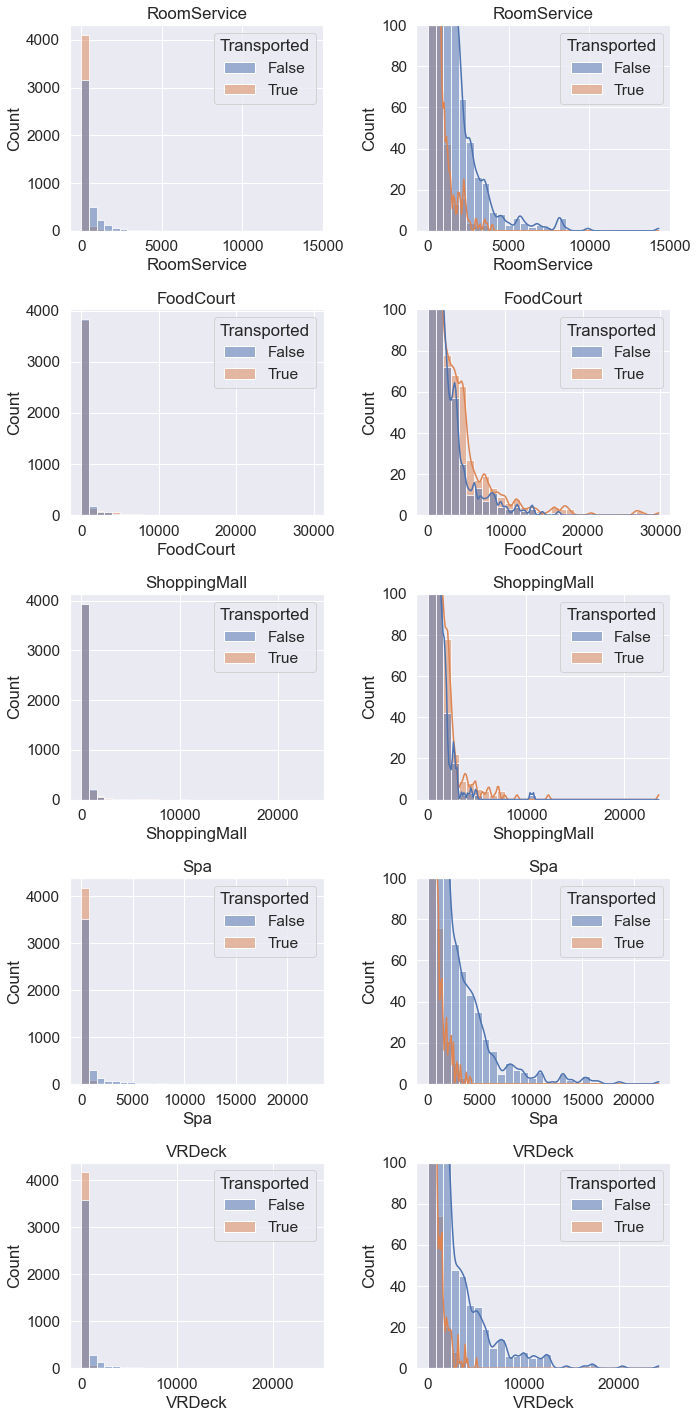


Insights from the above histogram:

* 0–18-year-olds were more likely to be transported than not.
* 18–25-year-olds were less likely to be transported than not.
* Over 25-year-olds were about equally likely to be transported than not.
* So, we now can create a new feature that indicates whether the passanger is a child, adolescent or adult.

We also played around with expenditure features such as Room service, Food Court,

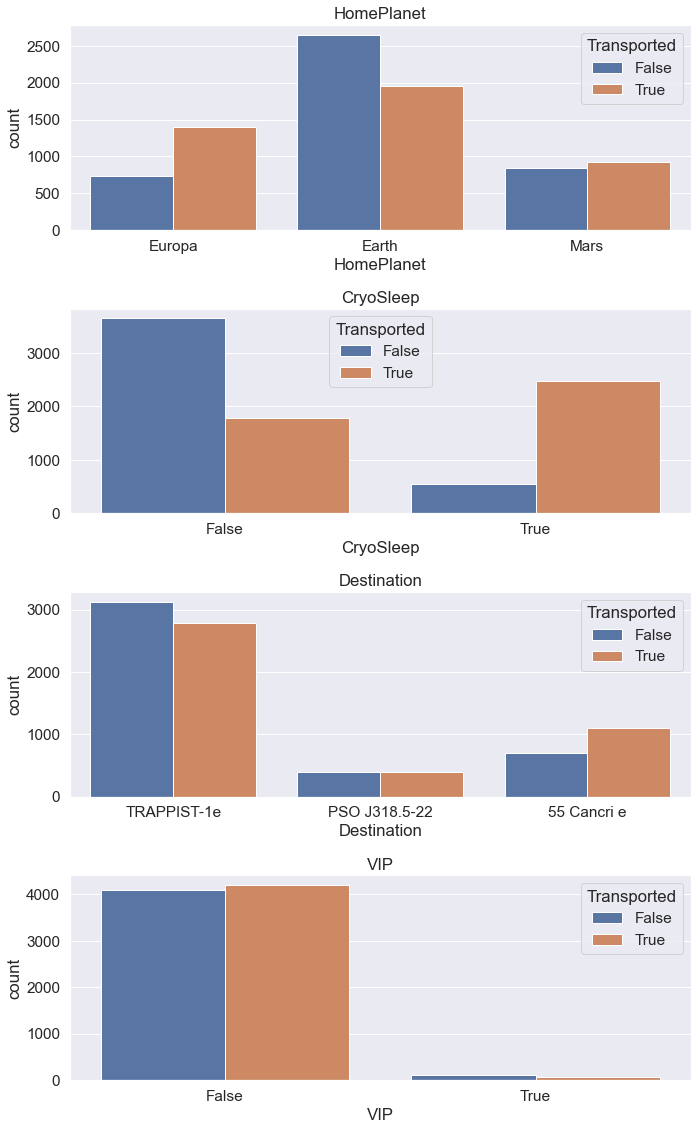
Shopping Mall, Spa, and the VR deck.



**Insights from Expenditure features:**

* Most people don't spend any money (as we can see on the left).
* The distribution of spending decays exponentially (as we can see on the right).
* There are a small number of outliers.
* People who were transported tended to spend less.
* RoomService, Spa and VRDeck have different distributions to FoodCourt and ShoppingMall - we can think of this as luxury vs essential amenities.
* We can create a new feature that tracks the total expenditure across all 5 amenities.
* We can also create a binary feature to indicate if the person has not spent anything. (i.e., total expenditure is 0). And finally take the log transform to reduce skew.

We also plotted all our categorical features as described in the below plots.

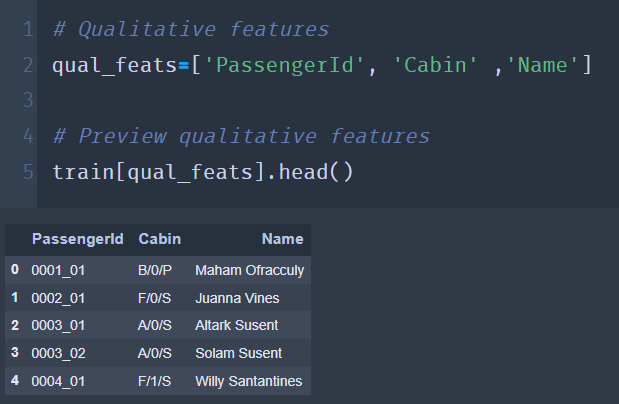


**Insights from above:**

* VIP does not appear to be a useful feature; the target split is more or less equal.
* CryoSleep appears the be a very useful feature in contrast.
* We can consider dropping the VIP column to prevent overfitting.

**Qualitative features:**

Our qualitative features are PassengerId, Cabin, and Name. we need to transform it into more useful features.



**Insights from Qualitative features:**

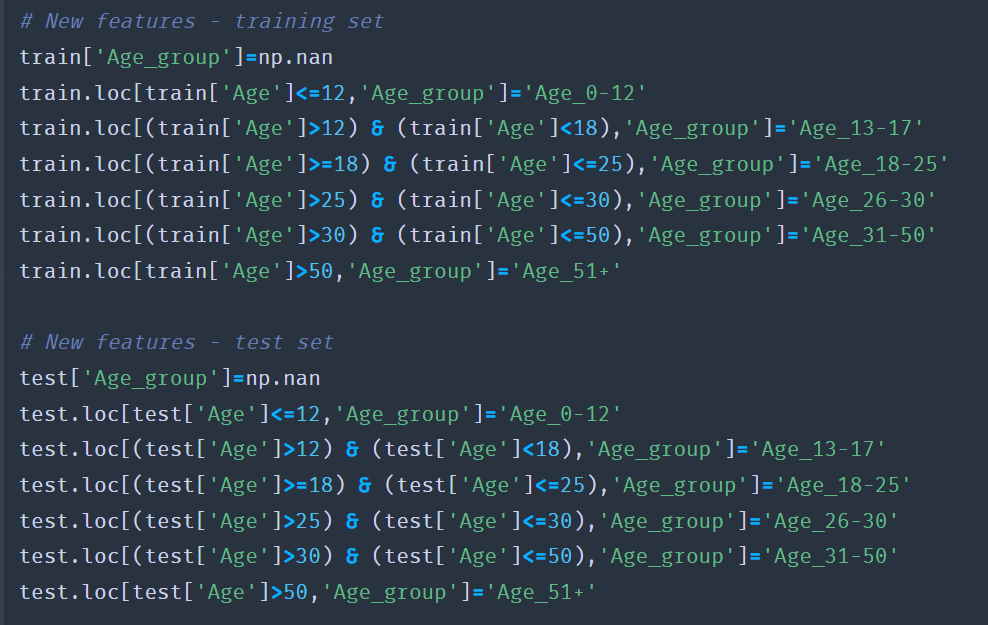
* PassengerId takes the form gggg\_pp where gggg indicates a group the passenger is travelling with and pp is their number within the group.
* Cabin takes the form deck/num/side, where side can be either P for Port or S for Starboard.
* We can extract the group and group size from the PassengerId feature.
* We can extract the deck, number and side from the cabin feature.
* We could extract the surname from the name feature to identify families.

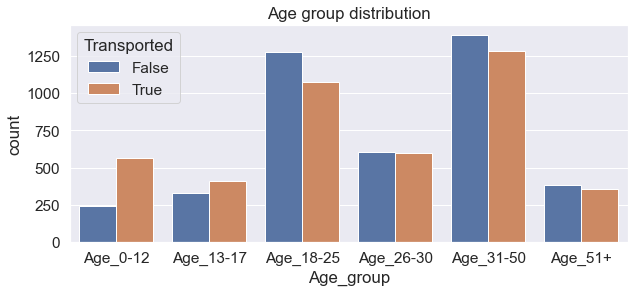
**Feature engineering:**

The philosophy to feature engineering is simple. Better features make better models. We highlight some of the tasks performed in our feature engineering below.

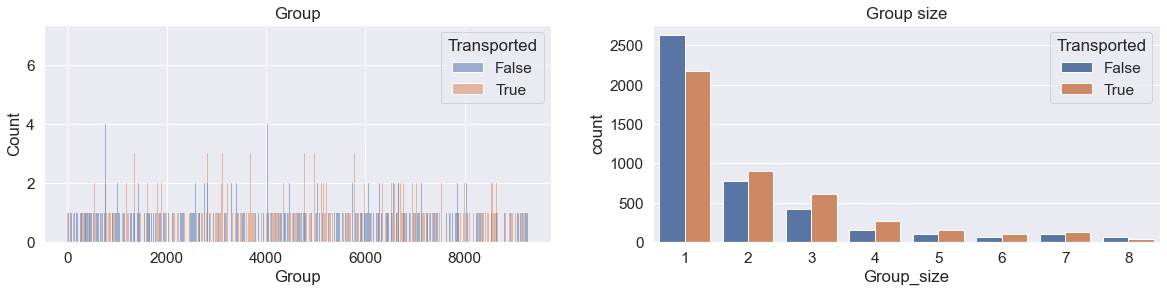
**Age status**

We create a new feature as age\_group in train set and in our test set. Its basically age feature into groups, which will be helpful for filling missing values like expenditure according to age. We divided it into Age between 13-17, 18-25, 26-30, 31-50, and 51 or above.



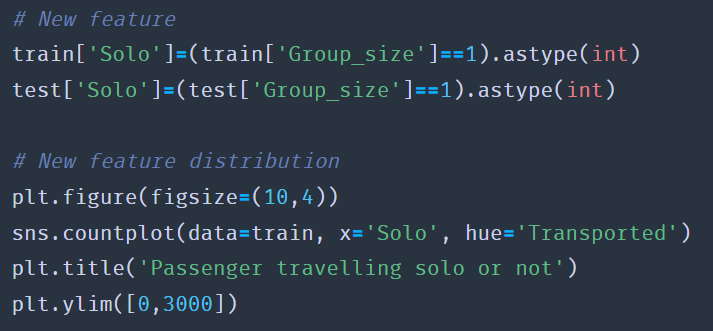


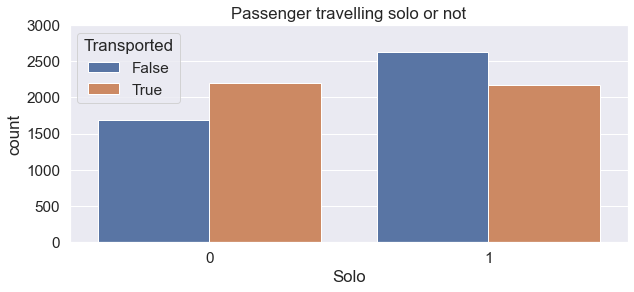
**Passenger group:**

****

From PassengerId, we derived the passenger group and group size. Because the Group feature has an excessively high cardinality (6217) and would drastically increase the number of dimensions using one-hot encoding, we are unable to include it in our models.

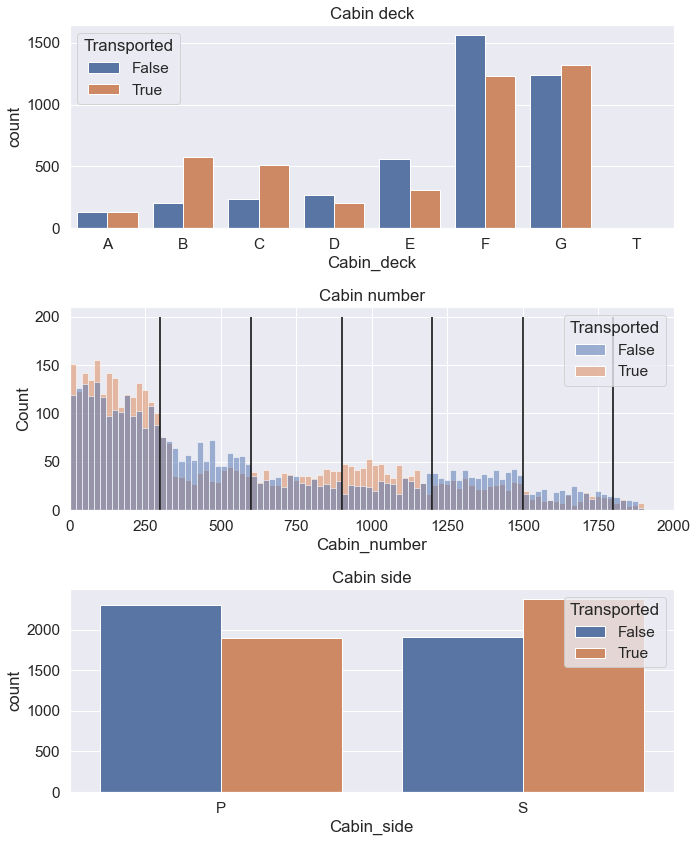
On the other side, the Group size feature ought to be handy. In fact, by adding a "Solo" column that tracks whether a person is traveling alone or not, we may reduce the feature even more. The right figure demonstrates that groups of one are less likely to be transferred than groups of more than one. And so, we created a new feature for travelling solo like mentioned below code snippet.





**Cabin location:**

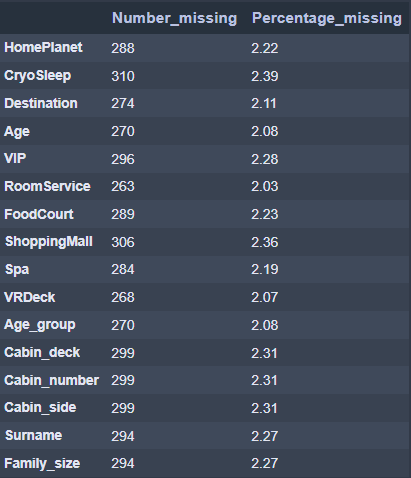
Here we extract deck, number and side from cabin feature.

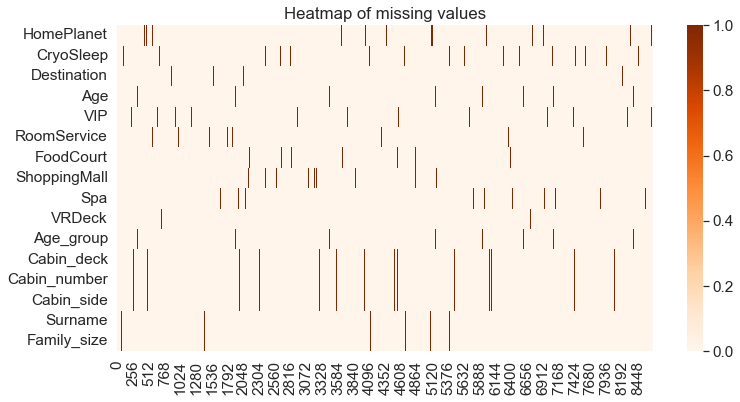


It seems as though Cabin number is divided into 300-cabin halves. As a result, we may condense this feature into a categorical one that identifies the chunk that each passenger belongs to. Additionally, the cabin deck "T," where there are just 5 samples, appears to be an oddity.

**Exploring missing values:**

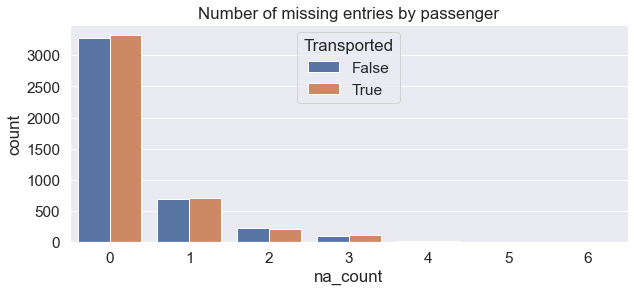
When exploring our missing values in depth, we found the following results.





Missing values make up about 2% of the data, which is a relatively small amount. For the most part, they don't seem to be happening at the same time except the features made from splitting Cabin and Name, however, I did inspect closer.

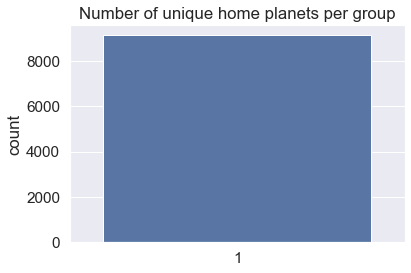
I began with creating a countplot of number of missing values by passenger as illustrated below.



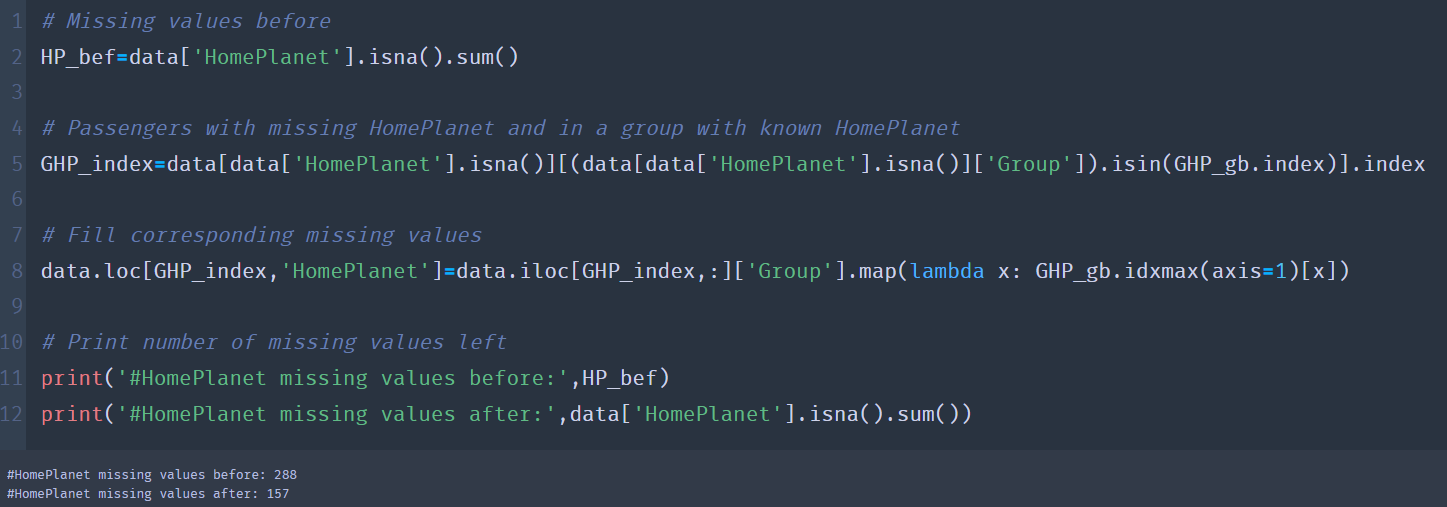
**Insights:**

* Missing values are independent of the target and for the most part is isolated.
* Even though only 2% of the data is missing, about 25% of all passengers have at least 1 missing value.
* PassengerId is the only (original) feature to not have any missing values.
* Since most of the missing values are isolated, it makes sense to try to fill these in as opposed to just dropping rows.
* If there is a relationship between PassengerId and other features we can fill missing values according to this column

Utilizing the median for continuous features and the mode for categorical features is the simplest technique to handle missing values. This will "sort of" work, but in order to maximize the accuracy of our models, we must search for patterns in the missing data. To achieve this, examine the combined distribution of traits, such as whether travelers from the same group tend to be related. There are undoubtedly numerous combinations, but just to highlight what we did, we will explore how we dealt with missing values on HomePlanet and Group. We initiated a joint distribution of Group and HomePlanet, and plotted all the unique values. It is illustrated in the below plot.



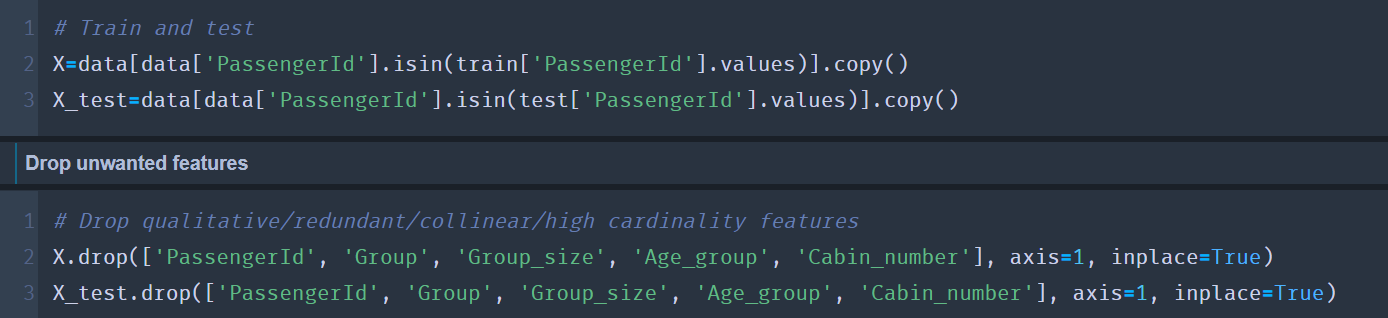
From the above we can understand that everyone in the same group comes from the same home planet. So, we can fill the missing HomePlanet values according to the group with at least the ones where the group size is bigger than 1. So, we filled the missing values in HomePlanet with values in Group, and managed to fill 131 values with 100% confidence.



We performed similar operations like above to deal with all the missing values for all the features such as Destination, Surname, cabin side, cabin deck, cabin number, VIP, Age, and CryoSleep. Now, let’s move on to pre-processing our data.

**Pre-processing:**

**Dropping unwanted values:**

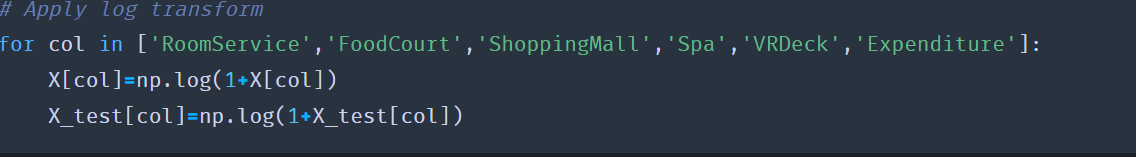


We dropped some of the redundant features like, 'PassengerId', 'Group', 'Group\_size', 'Age\_group' ,and 'Cabin\_number'.

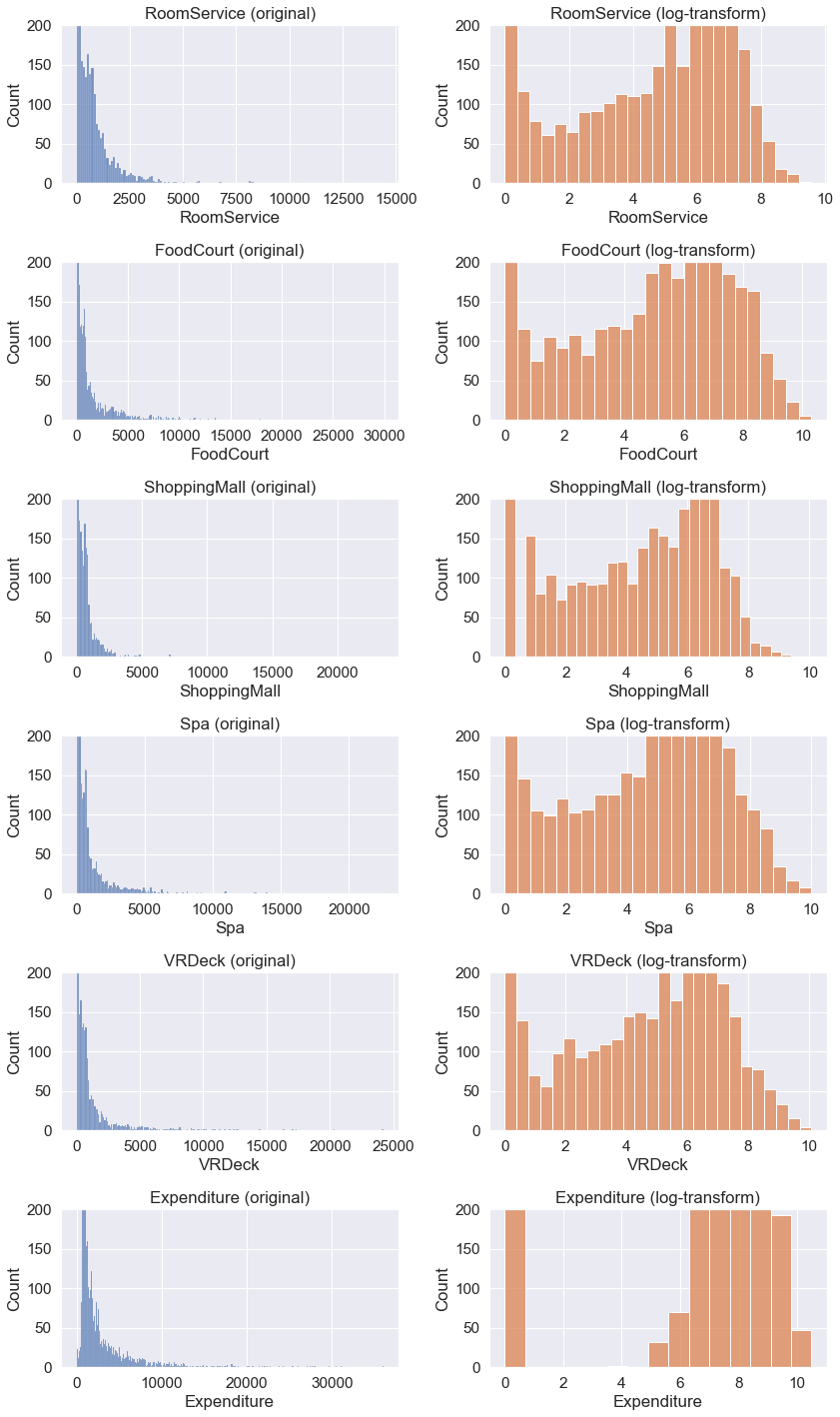
**Log transform**

It’s one of several methods that you can use to transform datasets to achieve linearity. This means it can help us obtain further insights into your data that may not be obvious at first. The log transformation can be used to make highly skewed distributions less skewed. This can be valuable both for making patterns in the data more. It can make it easier for algorithms to 'learn' the correct relationships. We will apply it to the expenditure features as these are heavily skewed by outliers. The results of our log transform are illustrated below.





The results of our log transform are illustrated below.



**Encoding and scaling:**

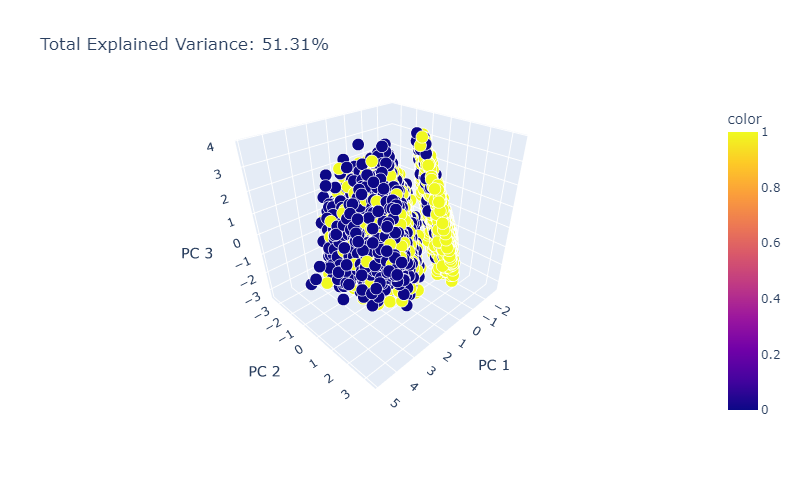
We use column transformers to identify numerical and categorical columns. We scaled numerical data to have a mean of 0 and variance of 1. And additionally, we performed one-hot encoding, a method of converting data to prepare it for an algorithm and get a better prediction. With one-hot, we convert each categorical value into a new categorical column and assign a binary value of 1 or 0 to those columns. Each integer value is represented as a binary vector. And finally, we combine and apply pre-processing. It is all illustrated below.



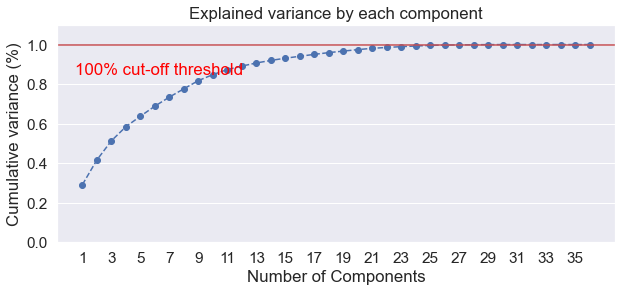
**Principal Component Analysis (PCA)**

A statistical method known as Principal Component Analysis (PCA) is used to reduce data without losing any of its features. Essentially, it does so without omitting a significant portion of the original information by describing the composition of variances and covariances through various linear combinations of the basic variables. In other words, the goal is to identify the particular set of orthogonal axes on which the data variance is highest. Its major goal is to solve the problem's multidimensionality. When dimensionality is reduced, it should be done so that the smallest amount of data is lost when higher dimensions are dropped.

Principal component analysis can also shed light on relationships between variables that are not immediately apparent. It aids in the analysis of the distribution of the observations and the identification of the distribution-related variables. Below is an illustration of transformed data in the PCA space. This gives a low dimensional representation of the data, which preserves local and global structure.



The sum of the variances of each individual primary component makes up the overall variance. The proportion of a principal component's variation to the total variance is known as the fraction of variance explained. Divide the sum of the variances of the primary components by the variance overall. We achieved a total variation of 51.31%, which is respectable because low variance suggests that data points are often comparable and do not deviate significantly from the mean. High variance denotes greater variability and wider deviations from the mean in data values. With the help of the variance calculator, you can calculate variance, standard deviation, sample size (n), mean, and sum of squares.



**Modelling:**

To briefly mention the algorithms I used,

**Logistic Regression:** This model fits a sigmoid-curve to the distribution of the target variable using Maximum Likelihood Estimation as opposed to linear regression, which utilizes Least Squares. When a binary output was provided for the data in question, the sigmoid/logistic curve is frequently utilized.

**K-Nearest Neighbours (KNN):** The metric employed is typically Euclidean distance, and KNN operates by choosing the majority class of the k-nearest neighbors. Although it is a straightforward and efficient technique, it can be sensitive to a number of variables, including the value of k, how the data was pre-processed, and the measure employed.

**Support Vector Machine (SVM):** SVM identifies the best hyperplane that seperates the data in the feature space. By examining which side of the hyperplane, the test point is located on, predictions are produced. Ordinary SVM makes the unfounded assumption that the data is always linearly separable. When this presumption fails, it is possible to transfer the data into a higher-dimensional space where it can be linearly separated using a kernel approach. SVM is a well-liked algorithm since it uses little computer resources and yields excellent results.

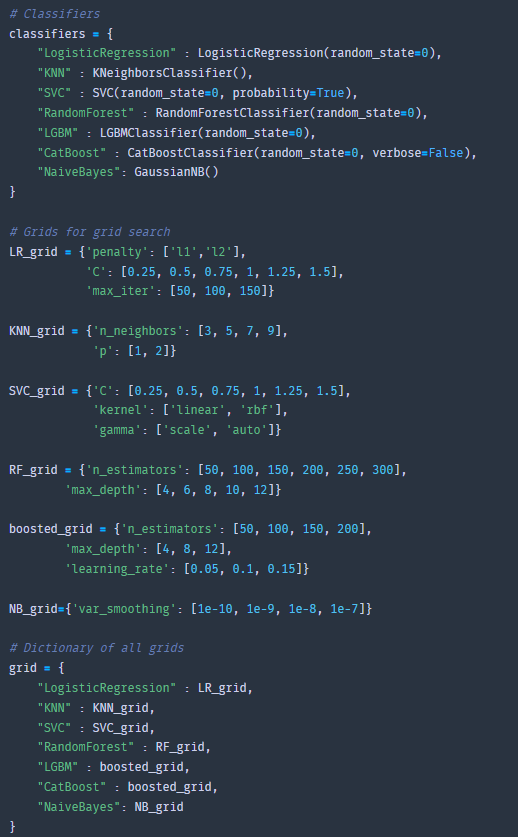
**Random Forest (RF):** Regression or classification issues can be solved using the trustworthy ensemble of decision trees known as Random Forest. Individual trees are constructed using bagging (i.e., the aggregation of bootstraps, which are nothing more than several train datasets produced through sampling with replacement) and divided using fewer features in this case. Because of the lower variance, the diversified forest of uncorrelated trees that results is more resistant to data change and maintains its prediction accuracy for new data. Both continuous and categorical data can be used successfully.

**Light Gradient Boosting Machine (LGBM):** XGBoost and LGBM both function roughly the same way, although LGBM uses a gentler boosting method. It usually gives similar results to XGBoost but is substantially faster.

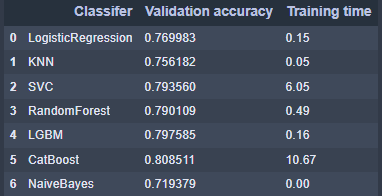
**Categorical Boosting (CatBoost):** Gradient boosted decision trees are the foundation of the open-source method CatBoost. It supports text, numerical, and category features. It is effective even with relatively tiny and heterogeneous data. Informally, it aims to combine the greatest aspects of LGBM and XGBoost.

**Naive Bayes (NB):** By applying the Bayes' Theorem, Naive Bayes gains the ability to classify data. It incorporates prior knowledge in accordance with Bayes' law to 'update' the likelihood of an event. Although the technique is relatively quick, it has the drawback of assuming independent input features, which is not necessarily the case.

**Model Implementations:**

****

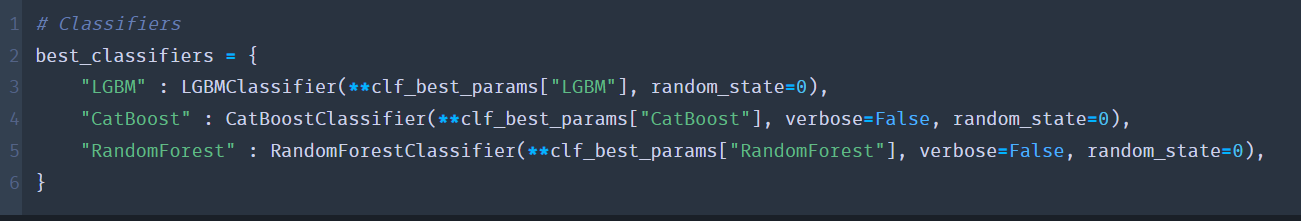
**Training and evaluating models:**

****

Overfitting is one of the key problems we want to avoid while training a machine learning model. This occurs when your model successfully fits the training data but fails to generalize and produce reliable predictions for untrained data. Data scientists utilize a method called cross-validation, where they divide their data into two parts—the training set and the validation set—to determine whether their model is overfitting. The validation set is solely used to assess the model's performance; the training set is utilized to train the model.

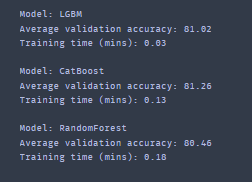
Metrics on the training set enable us evaluate how our model is progressing in terms of its training, but it's metrics on the validation set that let us obtain a gauge of the quality of your model - how well it's able to make new predictions based on data it hasn't seen before. With this in mind, loss and accuracy are measurements of loss and accuracy on the training set, whereas val loss and val acc are measures of loss and accuracy on the validation set. And with that stated, it was discovered in our experiment that CatBoost has the highest accuracy (0.808), followed by LGBM (0.797), which came in second, and Random Forest (0.79) in third.

This serves as motivation for us to move the top three performers to the final modeling step, where we can ultimately train our best model on the entire training set using cross validation and combining forecasts to produce the most accurate predictions.



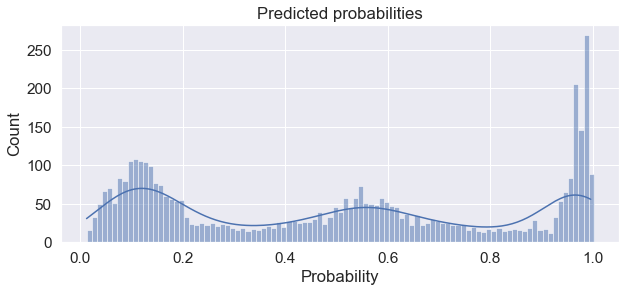
Predictions are ensembled together using soft voting. This averages the predicted probabilities to produce the most confident predictions as described in the below code snippet



.

After cross validation, and ensemble predictions, the final average validation accuracy for LGBM was 81.02, CatBoost 81.23, and 80.46 for Random Forest.

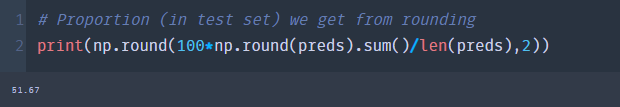
Moving further, we also looked into the distribution of predicted probabilities, as described in the plot below.



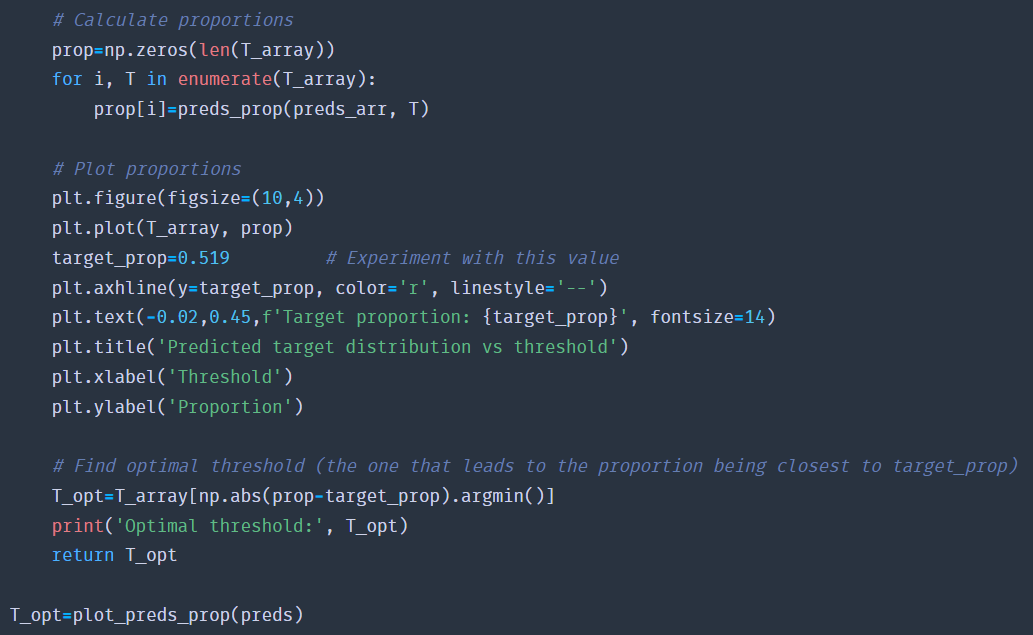
It's interesting to note that the models don't show much confidence in between; they are either very confident or quite unconfident.

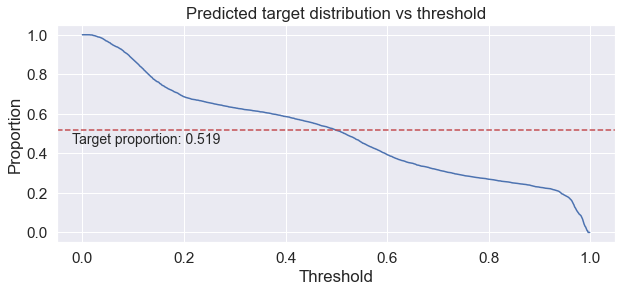
**Post-processing:**

Last but not least, we must assign each anticipated probability to one of the two classes (transported or not). Rounding each probability to the closest integer is the simplest method (0 for False or 1 for True). We can adjust the classification threshold to get a comparable proportion of transported/not transported in our predictions as in the train set, though, providing the train and test sets have similar distributions. Here we keep in mind that 50.4% of the passengers being transported were on the train set.

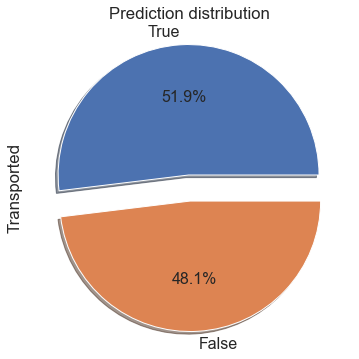


Since our models seem to (potentially) overestimate the number of transported passengers in the test set. Let's try to bring that proportion down a bit.





And finally, we read the sample submission once again to acquire the right format, and replace the column “Transported” with 0 to False and 1 to True. Our final prediction distribution is defined in the plot below.



**Conclusion:**

Businesses today run the danger of slipping behind the competition by taking only one shortcut. Data mining is now essential to the success of all firms, and for many of them, it presents amazing job opportunities. Businesses can benefit greatly from data mining, which gives them an advantage in fiercely competitive markets. Businesses can continually make more educated decisions thanks to data collection. Data mining is necessary to identify trends in customer behaviour or business requirements. For example, maintaining consistent statistics may help identify when inventory needs to be refilled or when more personnel is required to meet periods of consistently higher client activity.

In order to maintain effective corporate operations and make sure that money is distributed to the right needs, it's imperative to stay ahead of these decisions. By maintaining statistics, businesses can identify unsuccessful marketing initiatives, product lines, or sales. Despite the fact that a hypothetical dataset was utilized for this project, the methods used can be used to manipulate static or continuous data to aid in real-time decision-making.

**References**

Christiansen, L. (n.d.). *6 benefits to data mining in healthcare*. Zipreporting.com. Retrieved December 15, 2022, from <https://zipreporting.com/data-mining/data-mining-in-healthcare.html>

*Data science in 5 minutes: What is one hot encoding?* (n.d.). Educative: Interactive Courses for Software Developers. Retrieved December 14, 2022, from <https://www.educative.io/blog/one-hot-encoding>

David, D. (2021, October 14). *Why data science competitions are important & how to get started*. Hackernoon.com. <https://hackernoon.com/why-data-science-competitions-are-important-and-how-to-get-started>

Duprey, R. (1983). *Spaceship Titanic*. Dodd.

Htoon, K. S. (2020, February 29). *Log transformation: Purpose and interpretation - Kyaw saw htoon*. Medium. <https://medium.com/@kyawsawhtoon/log-transformation-purpose-and-interpretation-9444b4b049c9>

Jonathan. (2019). *What is data mining?* PowerKids Press.

*Log Transformations*. (n.d.). Onlinestatbook.com. Retrieved December 14, 2022, from <https://onlinestatbook.com/2/transformations/log.html>

Shmueli, G., Bruce, P. C., Gedeck, P., & Patel, N. R. (2019). *Data Mining for Business Analytics: Concepts, Techniques and Applications in Python*. Standards Information Network. <https://www.wgu.edu/blog/data-mining-business-analytics2005.html>