Spaceship Titanic

Name Members: Shantanu Patil, Nishit Patel, Ramni Kotra
Date: 12/20/2022
Subject: General Business 656

Introduction:

Thinking about the future and future business issues that might occur, we chose the Spaceship Titanic problem. We know what happened to the Titanic ship and how people lost their lives and the shipping company had to compensate for it, along with their reputation being dragged down to the ground. Similarly, a month ago, the Spaceship Titanic, an intergalactic passenger liner, was launched. The ship left on its inaugural mission carrying emigrants from our solar system to three newly habitable exoplanets circling neighboring stars with approximately 13,000 passengers on board.

The unsuspecting Spaceship Titanic collided with a spacetime anomaly concealed beneath a dust cloud as it rounded Alpha Centauri on its way to its first destination—the scorching 55 Cancri E. Sadly, it experienced the same demise as its namesake. The ship was unharmed, but approximately half of the people were taken to a different dimension.

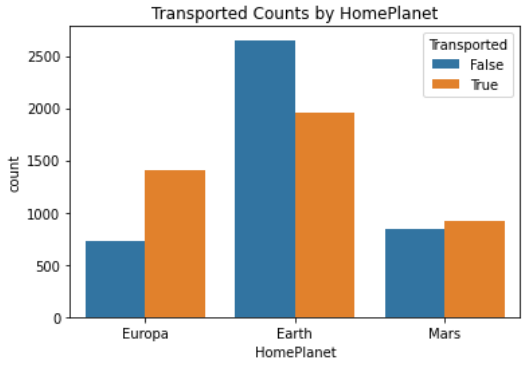
In order to prevent the same debacle from happening to the spaceship company we are using the data collected from the spaceship's broken computer system, we are identifying which passengers were carried by the anomaly to aid rescue teams and locate the missing passengers. This would not only help rescue people but also help the company by not having huge lawsuits filed against them and their reputation stay intact. However, to predict this we are first going to understand the data, do an Exploratory Data Analysis then clean the data to eliminate unnecessary variables. Then work on different prediction models to find the optimal solution.

Anomaly detection is growing in significance as a technique of data analysis and alerts. A value that deviates from the norm by a large enough margin to be considered an anomaly. The development of patterns is a necessary initial step in the detection process, followed by the identification of the units that violate those patterns. We can find this anomaly detection in fraud detection, intrusion detection, and many more. Talking fraud detection, credit card, bank account, and insurance fraud are all prevented with the use of graph-based anomaly detection (GBAD). With the use of behavioral biometrics, which can also detect irregularities in customer spending in real-time, ML systems also facilitate online banking fraud.

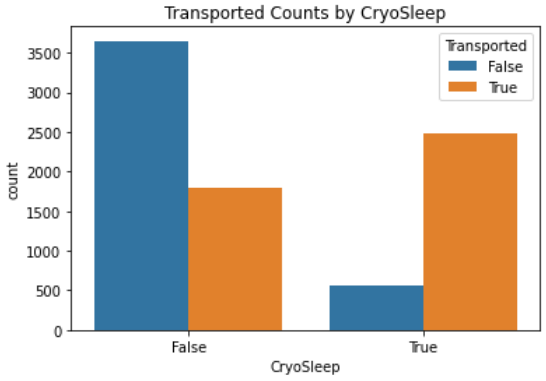
Understanding Data:

After reading the data, we first plotted all the graphs (EDA) for a better understanding of the data and analyzed the data using visual techniques. It would help us to discover trends, and patterns, or to check assumptions with the help of statistical summaries and graphical representations.

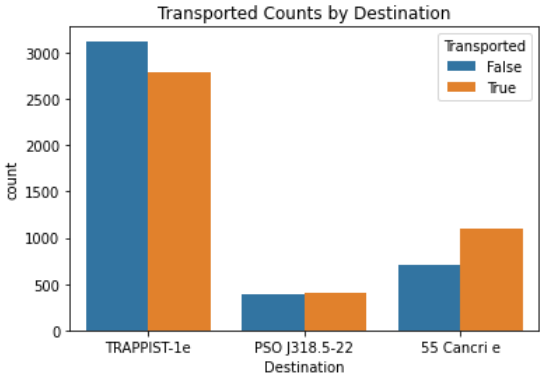
First, we can see the comparison of the number of passengers transported from their home planets like Europa, Earth, and Mars. Down we can see the graph of their count where True is passengers who have been transported and False is passengers who have not been transported yet.



After looking at the graph we can see that the Home planet is definitely correlated with the count of transported data.
Next, we see the number of passengers transported in Cryosleep: Passengers elected to be put into suspended animation for the duration of the voyage. Below we can find the graph that depicts the count of it. Here, we can see that even though more people have been transported in Cryosleep because of the crash it might be easier for people to be conscious and understand what's going on.

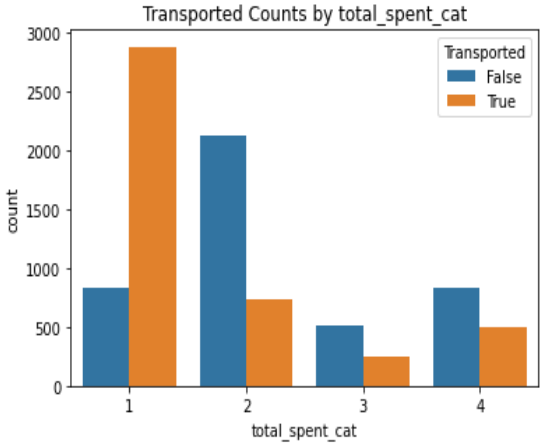


Moving on we have compared the destination of passengers that will be debarking to and found out that most of them wanted to be transported to Trappist-le. It helps us understand that most of the passengers preferred transporting to Trappist-le.



Next, we are adding a column called ‘Total Spent’ to the data frame which is the sum of the values in column ‘Room Service’ to ‘VR Deck’ which is basically the total money spent on amenities. We have

taken care of missing values when calculating the sum of it. This could be very useful for later but for

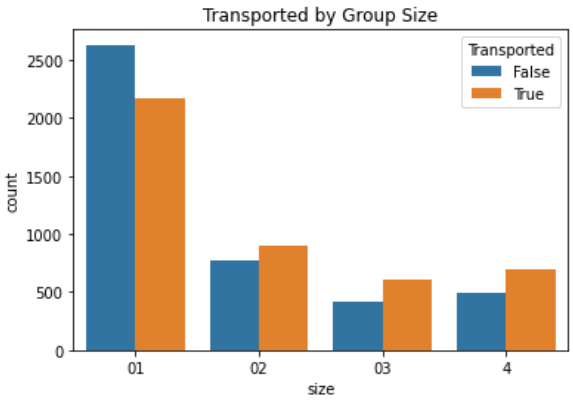


Name	Transported	total_spent	total_spent_cat
Maham Ofracculy	False	0.0	1
Juanna Vines	True	627.0	2
Altark Susent	False	10340.0	4
Solam Susent	False	5176.0	4
Willy Santantines	True	788.0	2
...	...	...	...
Gravior Noxnuther	False	8536.0	4
Kurta Mondalley	False	0.0	1
Fayey Connon	True	1873.0	3
Celeon Hontichre	False	4637.0	4
Propsh Hontichre	True	4700.0	4

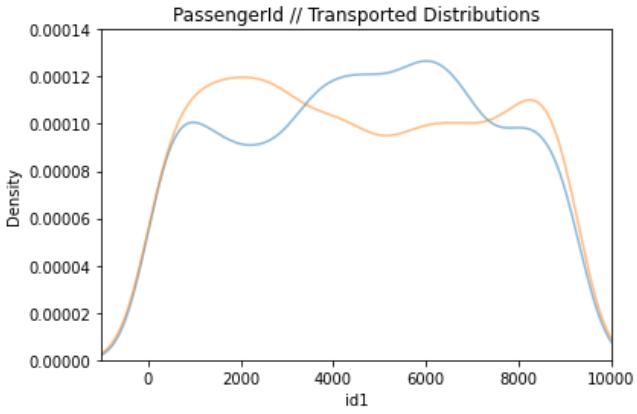
now, we have deleted the column.
Due to the equal proportions of Transported and the little number of TRUE observations, VIP doesn't seem to be very beneficial, therefore we excluded the graph of it.

Cleaning Data:

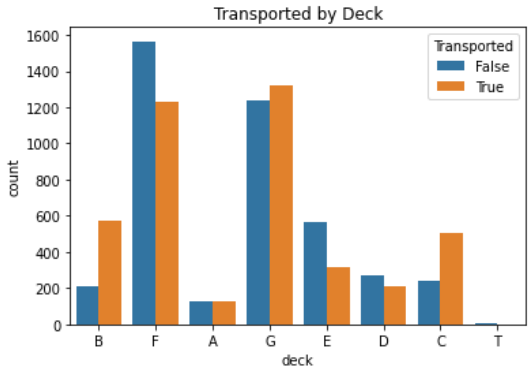
In order to improve machine learning model training we created a new data frame in order to store the newly manipulated data i.e split the Cabin and Passenger Id columns, calculating the group sizes. Because there are such small amounts of large groups, we decided to count groups of four or larger simply as four.

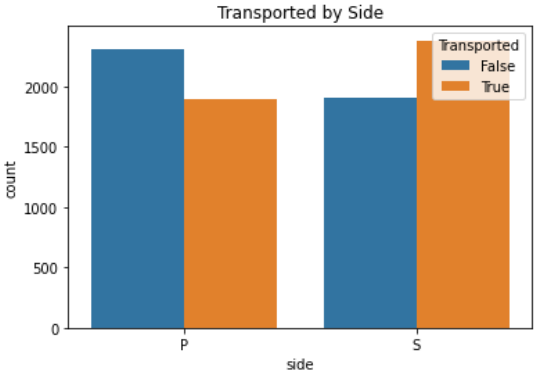


After creating a group size column we can see that singles were disproportionately likely to stay on the ship. Next, we will look at the density plot. According to PassengerId, it demonstrates that there are differences in the concentrations of transported and not transported passengers. We can see that below.



It is simple to add relevant data to the model by separating the deck and side from the cabin. The decks and sides are clearly heterogeneous and we can see that below.





We dropped ‘Name’, ‘num’, and ‘Cabin’ features as there is no point in keeping them as we cannot extract the gender from the data and not keep ‘num’ due to its variability.

Now, we assessed how much data is missing and found out that most of the columns are missing about 2% of the data. Furthermore, we saw the proportion of rows that are missing values and found out 24% of the records have missing data. In order to correct that we imputed the data which is replacing missing data with substituted values. We also thought of getting rid of the missing data but figured it is not a viable option.

So, first consider Cryosleep as we are aware that those using CryoSleep won't be able to make purchases at any of the facilities, hence, we got rid of those missing values.

Working on imputation based on the group, we can fill in some fields based on the group using the PassengerId data. When two individuals are traveling together, we presume that they are from the same HomePlanet, are headed to the same destination, and will probably be staying on the same deck.

```
HomePlanet      0.023122
Destination     0.020936
deck            0.022892
```

Here we conclude that 50% of the missing values have been corrected.

With the string "Mars," the missing values in the "HomePlanet" column are subject to further imputation. Next, we use the mode (most prevalent value) of the columns "deck," "side," "num," and "CryoSleep" to fill in any missing values. Lastly, the mode of the "Destination" and "VIP" columns could be used to fill in any empty information.

Depending on the kind of data in each column, we employed different filling techniques to replace any missing values. For instance, utilizing the median for numerical data like "num" and the mode for qualitative data like "deck" or "side."

Moving on we used the Linear Model with other features to predict the missing age values where we got 13.43. We figured that the id1, CryoSleep, id2, and side features weren't significant in our first model. The amenities don't seem to add much predictive power. So we tried another model without the insignificant features. With an RMSE of 13.43, the model with fewer features performs marginally better than the model with more characteristics. This is noticeably better than utilizing just the median values, which results in an RMSE of 14.73. The model will be retrained using all of the available data, and its predictions will then be used to fill in the missing values.

Modeling:

Logistic Regression: Here, we trained a logistic regression model on the data using the scikit-learn library, and then we used 5-fold cross-validation to assess how well it performed. We obtained **0.7914** after importing all required libraries, initializing the logistic regression model, running 5-fold cross-validation, and assessing test correctness. Cross-validation is the process of dividing the data into folds, training the model on various combinations of the data, and then testing it. It was used to assess the model's effectiveness and lower the chance of overfitting. The model is trained and tested five times in our code using five distinct combinations of folds as the training and test sets, which is known as five-fold cross-validation.

KNN: To train and fine-tune a k-nearest neighbors (KNN) classifier on our data and assess its performance on a test set, we utilized the scikit-learn library. To fine-tune the KNN model, we first constructed a grid of k values. The number of closest neighbors the model takes into account when making predictions is known as the k value. Our code then creates a grid search object utilizing the NeighborsClassifier class, the initialized KNN model, and the k value grid. The grid search object will use k-fold cross-validation to assess the model's effectiveness for each k value. The grid search object is then fitted to the training data. The KNN model is trained using each value of k on the grid, and the best model is chosen based on cross-validation results. Finally, it assesses the model using the test data and outputs the model's accuracy, which is **0.775**.

Random Forest: Here, we train a random forest classifier on the data using the scikit-learn library and assess its performance using k-fold cross-validation and a test set. Using the 'RandomForestClassifier' class, we first initialize the random forest classifier. Once the model has been fitted to the training set of data, k-fold cross-validation should be used to assess the model's performance. An array of scores, one per each fold, is returned by the cross_val_score function. The code outputs these results. Finally, re-fit the model to the training data and use the scoring technique to assess the model's performance on the test data. The model's accuracy on the test set might be printed as **0.822**.

GBM: Once more, we trained a gradient boosting classifier on the data using the scikit-learn module, and then used k-fold cross-validation and a test set to assess how well it performed. We set many hyperparameters for the model, including the number of estimators, the maximum depth of the trees, the learning rate, and the minimum amount of samples needed to be at a leaf node, when we initialized the gradient boosting classifier using the 'GradientBoostingClassifier' class. The model is then fitted to the training data set, and its performance is assessed using k-fold cross-validation, using the 'cross_val_score' function. An array of scores, one per each fold, is returned by the cross_val_score function. The model was then once more fitted to the training set of data, and its performance on the test set was assessed using the 'score' approach. On the test set, we were able to print the model's accuracy as **0.807**.

Conclusion:

After comparing all the modeling, we can see that Random forest has the highest accuracy. As a sort of machine learning method, random forests may be applied to both classification and regression applications. They are simple to use, precise, strong, and comprehensible. To do classification or regression, random forests build a large number of decision trees during our training phase, then output the class that represents the mean of the predictions made by each tree. Here, it gives the best results because it uses a bootstrap technique to pick the best combination of variables and bags them to give the highest accuracy. They can deal with missing values and noisy data and are resistant to overfitting. In general, random forests are a robust machine-learning technique that may be used for a variety of tasks.. Hence, we would go with Random forest to retrain the model with the entire training dataset. As a result, a lot of passengers could be rescued which gives a sigh of relief to the company.