**Kickstarter Predictive Model**

*Uzair Masih Israrahmed*

## Executive Summary

Kickstarter.com is a platform for projects and potential investors. Each project has a goal amount that can be fulfilled by crowdfunding on the platform. Creators that need funding for their project can create rewards based on donation amounts to encourage more backers. The creators only have access to these funds if they hit their specified goal. There have been many projects that have hit their fundraising goal, but there are also projects that haven’t.

Knowing what makes a project “successful” (reaching/surpassing fundraising goal) and “unsuccessful” can be valuable information for future creators, investors, and Kickstarter. If we are able to accurately predict the success of projects on Kickstarter before going launch, we can discover correlations between past successful projects (and unsuccessful), and future creators can increase the likelihood of hitting their set goals. Potential investors can also be more convinced to back up a project if the likelihood of success is high. Hosting more successful projects on Kickstarter.com will improve the company’s credibility and attract more creators to crowdfund through Kickstarter. Predicting the success of projects can also allow Kickstarter to better manage projects that are very unlikely to succeed by either not allowing their project to launch or offering consulting to improve their chances of success.

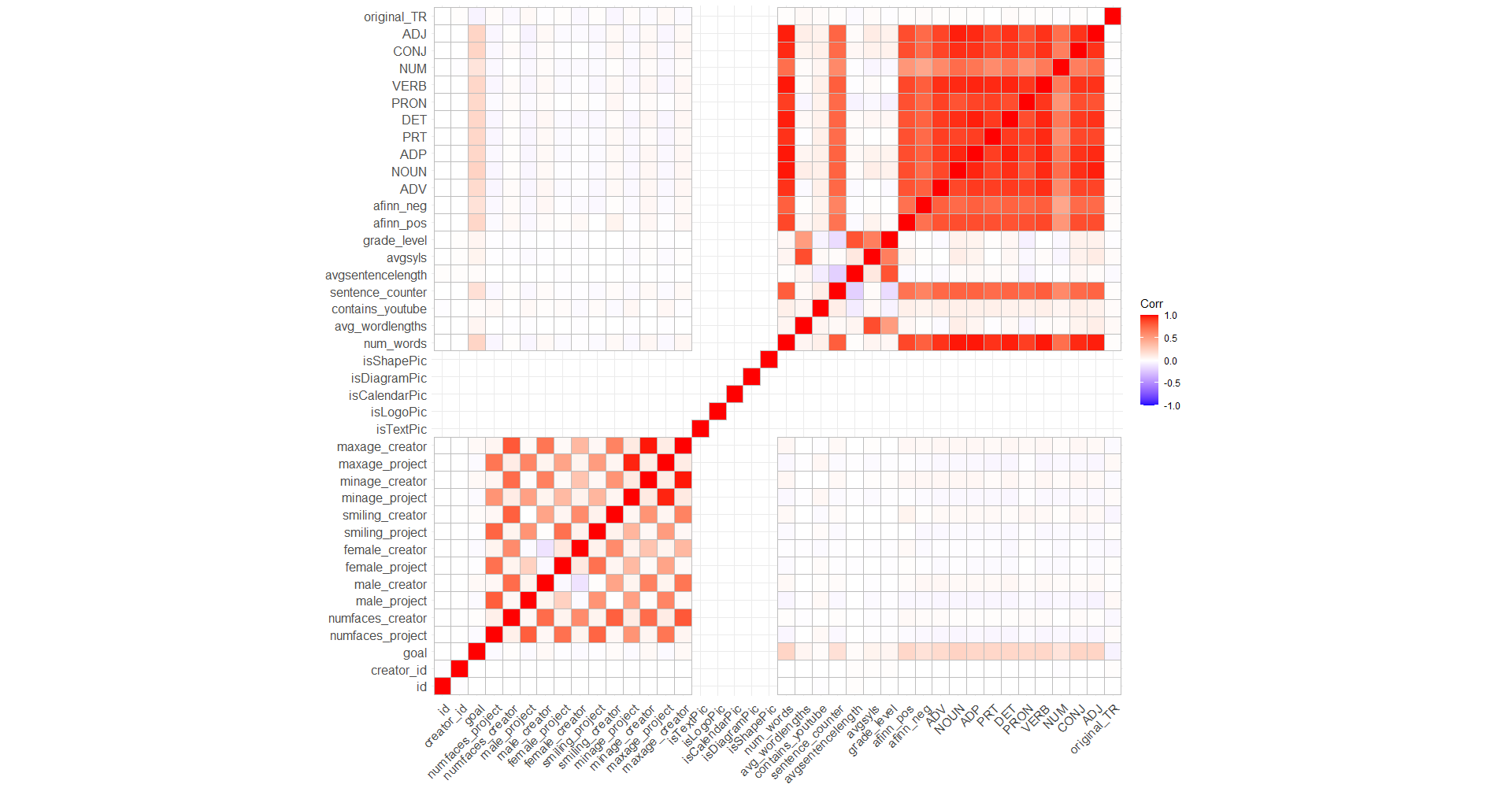
The data we are given is all information from project pages on Kickstarter.com. We utilized data on the project description, creator name, rewards, pictures, goal amount, launch date, etc. to build and train a model that will most accurately predict the success of future projects.

## 

## Model Description

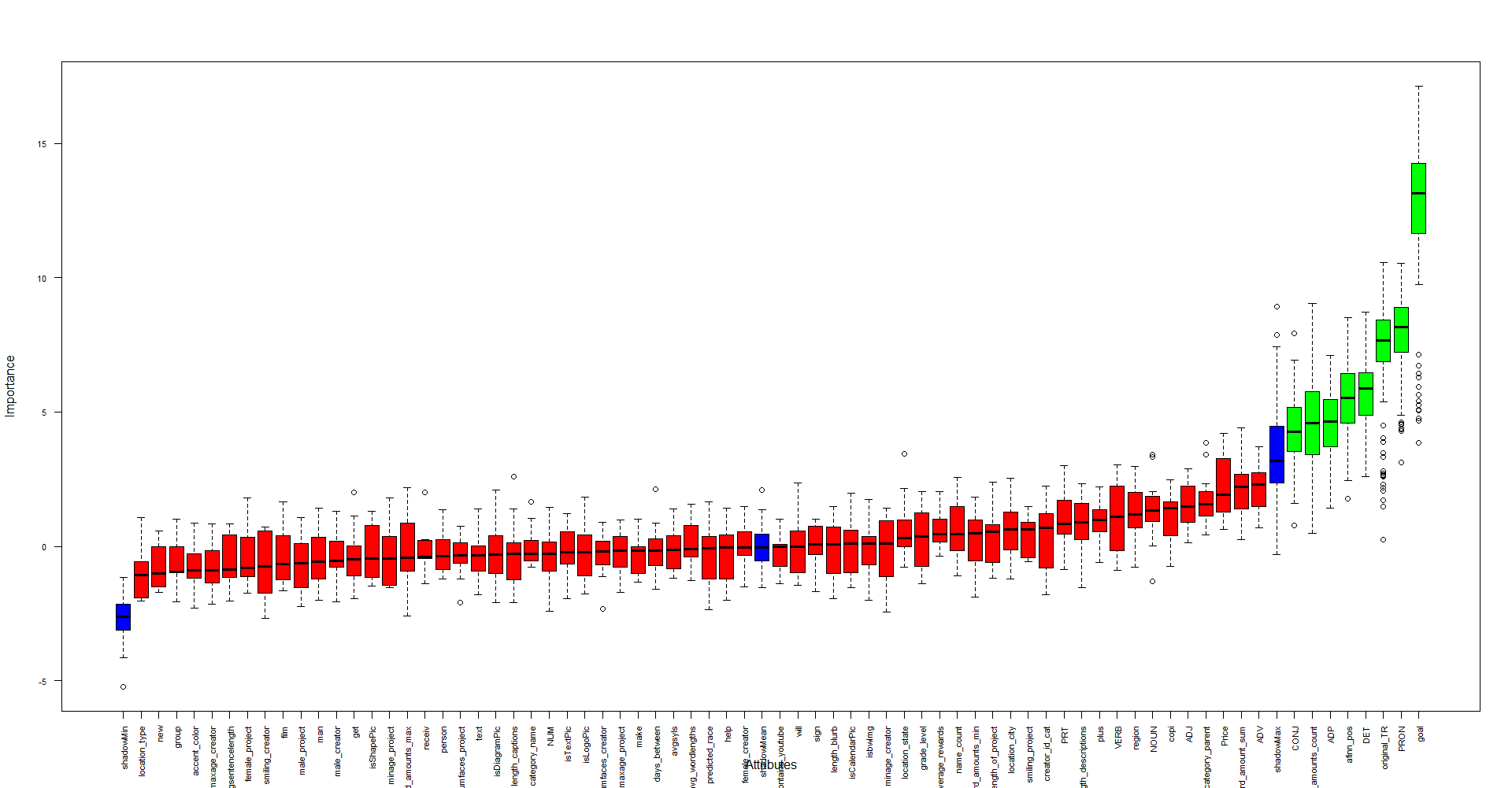
### Exploratory Analysis

To thoroughly explore all the instances we were given, we combined the provided training and test datasets. To create our training dataset, we combined the train x and train y into one dataset and binded the rest of the x variables from the test x dataset file. We also created a new column called original\_TR to be able to filter our original training data from the original test data. The original training set has an original\_TR value of 1, and the test set has a value of 0. Train\_success will include all data without big\_hit or backers\_count and filter out all instances with NAs in the success column. We found that some variables had NULL values, so in our data cleaning, we dealt with these NA’s.

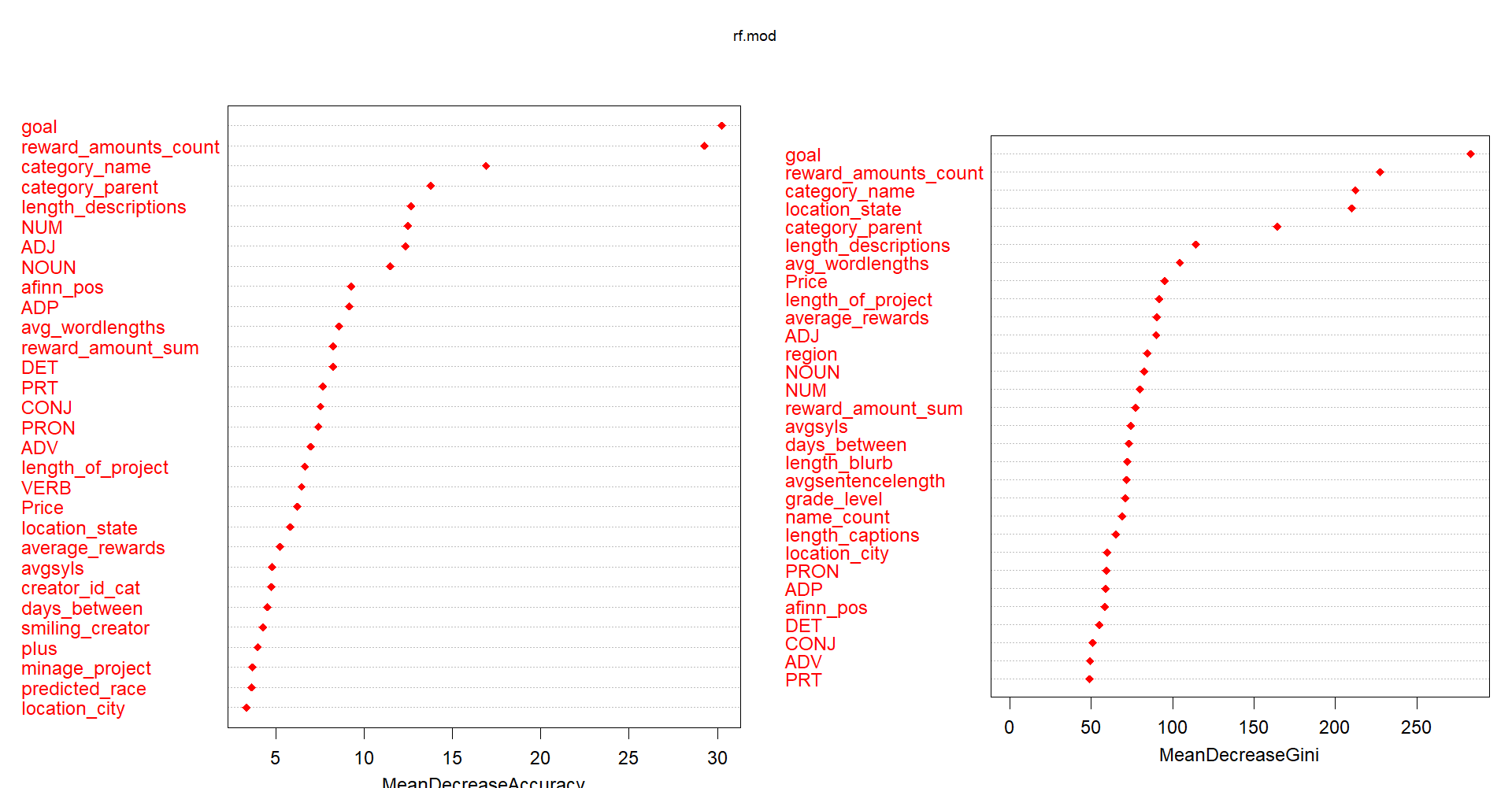
Our data exploration also showed us that some of the frequency of unique values in each categorical data column was relatively small, and our model would benefit from binning the less common values together in a new value called “Other.” Combining similar categories such as “journalism” and “photography” helps us with converting common values together into binned variables (factors). 

Lastly, we created a correlation matrix (Figure 1), variable importance plot (Figure 3), and feature selection plot (Figure 2). Based on the correlation matrix, we decided to remove num\_words, sentence\_counter, and afinn\_neg because these showed high correlation with other variables, and based on the variable importance plot, genre proved to not be important in predicting success.

*Figure 1. Correlation Matrix*



*Figure 2. Feature Selection Plot*

*Figure 3. Variable Importance Plot*

### Features Used in Our Model

### *Original Features*

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Feature**  (\* For Unused Features) | **Description** | **Feature Engineering** |
| 1 | creator\_id\_cat | Project creator’s ID bins | * Created from creator\_id as bins for frequency of creator ids and added labels for each bin * The labels are: “not popular” for <3 projects, “less popular for >2 and <5 projects, “okay popular” for >5 and <10 projects, “medium popular” for <40 and >10 projects, and “very popular” for >40 projects. |
| 2 | location\_state | Project location’s state | * Created from location\_slug as the location’s state * Converted any values that has less than or equal to 2090 count into “Other” |
| 3 | location\_city | Project location’s city | * Created from location\_slug as the location’s city * Converted any values that has less than or equal to 2000 count into “Other” |
| 4 | location\_type | Project location type | * Marked NA values as “Missing” * Converted the following values as “Other”: “Island“,”Zip”,”Miscellaneous”,”Estate”) |
| 5 | region | Project location region | * Converted to factor |
| 6 | category\_parent | Main project category | * Converted “journalism” and “photography” as “photography & journalism” |
| 7 | category\_name | Project subcategory | * Converted any values that has less than or equal to 2000 count into “Other” |
| 8 | isbwImg | Whether the main project picture is black and white or not | * Created from isbwImg1 where we marked NA values as “MISSING”. |
| 9 | color\_foreground\* | Foreground color of the main project picture | * Marked NA values as “Missing” * Converted the following values into “Other”: “Purple”, “Teal”,”Orange”,”Pink”, and ”None” |
| 10 | color\_background\* | Background color of the main project picture | * Marked NA values as “Missing” * Converted the following values into “Other”: “Purple”,”Pink”,”Teal”,”Orange”, “Red,” None”, and “Yellow” |
| 11 | accent\_color | Accent color of the main project picture | * Marked NA values as “Missing” * Converted value of “666666” as “black” |
| 12 | genre\* | First tag from tag\_names | * Created from tag\_names where we took the first tag |
| 13 | isTextPic | Whether the main project picture is text or not | * Marked NA values as 2 |
| 14 | isLogoPic | Whether the main project picture is a logo or not | * Marked NA values as 2 |
| 15 | isCalendarPic | Whether the main project picture is a calendar or not | * Marked NA values as 2 |
| 16 | isDiagramPic | Whether the main project picture is a diagram or not | * Marked NA values as 2 |
| 17 | isShapePic | Whether the main project picture is a shape or not | * Marked NA values as 2 |
| 18 | contains\_youtube | Whether the description contains a link to a YouTube video | * Convert the factor |
| 19 | length\_blurb | Length of blurb | * Created from blurb as the length of blurb |
| 20 | length\_captions | Length of captions | * Created from captions as the length of captions |
| 21 | length\_descriptions | Length of reward\_descriptions | * Created from reward\_description as the length of reward\_description |

***Features from Unstructured Text Variables***

* We wanted to know if certain frequent words in the unstructured text variables help projects to be more likely successful. (Performance improvement from these features can be seen in Table 3.)

|  |  |  |  |
| --- | --- | --- | --- |
| 22 | blurb TFIDF | TFIDF for blurb | * Tokenized and vectorized blurb to create TFIDF based on the top 5 most frequent words |
| 23 | reward\_description TFIDF | TFIDF for reward\_description | * Tokenized and vectorized reward\_description to create TFIDF based on the top 5 most frequent words |
| 24 | captions TFIDF | TFIDF for captions | * Tokenized and vectorized captions to create TFIDF based on the top 5 most frequent words |

### *Date/Time Features*

* We wanted to know if shorter projects are more likely to be successful.

|  |  |  |  |
| --- | --- | --- | --- |
| 25 | created\_at\_year\* | Project creation date’s year | * Created from created\_at as the year |
| 26 | created\_at\_month\* | Project creation date’s month | * Created from created\_at as the month |
| 27 | launched\_at\_year\* | Project launched date’s year | * Created from launched date as the year |
| 28 | launched\_at\_month\* | Project launched date’s month | * Created from launched date as the month |
| 29 | days\_between | The number of days between project created date and launched date | * Created by subtracting created\_at date from launched\_at date |
| 30 | length\_of\_project | The number of days between project launched date and deadline date | * Created by subtracting launched\_at date from deadline date |

### *Feature from External Data Set*

* We wanted to know if the overall economic performance has an impact on the likelihood of a project being successful. In other words, are projects more likely to be successful when the economic performance is good? (Performance improvement from these features can be seen in Table 3.)

|  |  |  |  |
| --- | --- | --- | --- |
| 31 | price |  | * Imported a dataset on [Dow Jones Industrial Average Historical](https://www.kaggle.com/datasets/mnassrib/dow-jones-industrial-average) data * Merged the Dow Jones data with Kick Starter data on date * Grouped the data by month and took the average of each month’s Dow Jones price and mapped it to each date. |

### *Reward Features*

* Do projects with higher reward amounts get funded more and more likely to be successful?

|  |  |  |  |
| --- | --- | --- | --- |
| 32 | reward\_amounts | Comma-delimited list of the donation amounts required to unlock project "rewards" | * Marked NA values as 0 |
| 33 | reward\_amounts\_count | Number of rewards | * Created from reward\_amounts where it counted the number of rewards |
| 34 | reward\_amounts\_max | Maximum reward amount | * Created from reward\_amounts where it represents the maximum reward amount |
| 35 | reward\_amounts\_min | Minimum reward amount | * Created from reward\_amounts where it represents the minimum reward amount |
| 36 | length\_descriptions | Reward description length | * Created from reward\_amounts as the reward description length |
| 37 | average\_rewards | Average number of rewards | * Created from rewards\_amounts as the average number of rewards |
| 38 | reward\_amount\_sum | Total number of rewards | * Created from rewards\_amounts as the total number of rewards |

***Predicted Race***

* We wanted to know if race plays a role in helping projects to be successful.

|  |  |  |  |
| --- | --- | --- | --- |
| 39 | predicted\_race | Creator’s predicted race | * Used predictrace library to predict the creator’s race based on their name |

***Sentiment Analysis***

* Do projects with higher sentiment value get funded more and more likely to be successful?

|  |  |  |  |
| --- | --- | --- | --- |
| 40 | sentiment\_list | Sentiment value of the project | * Utilized afinn dictionary to find sum of sentiment values of each word in text column |

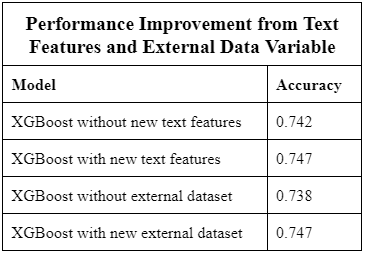
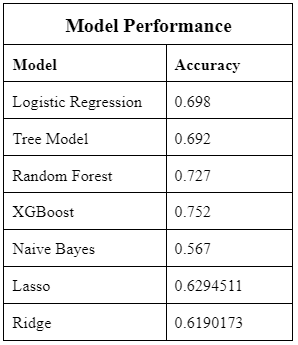
*Table 1. Variable Feature Engineering Table*

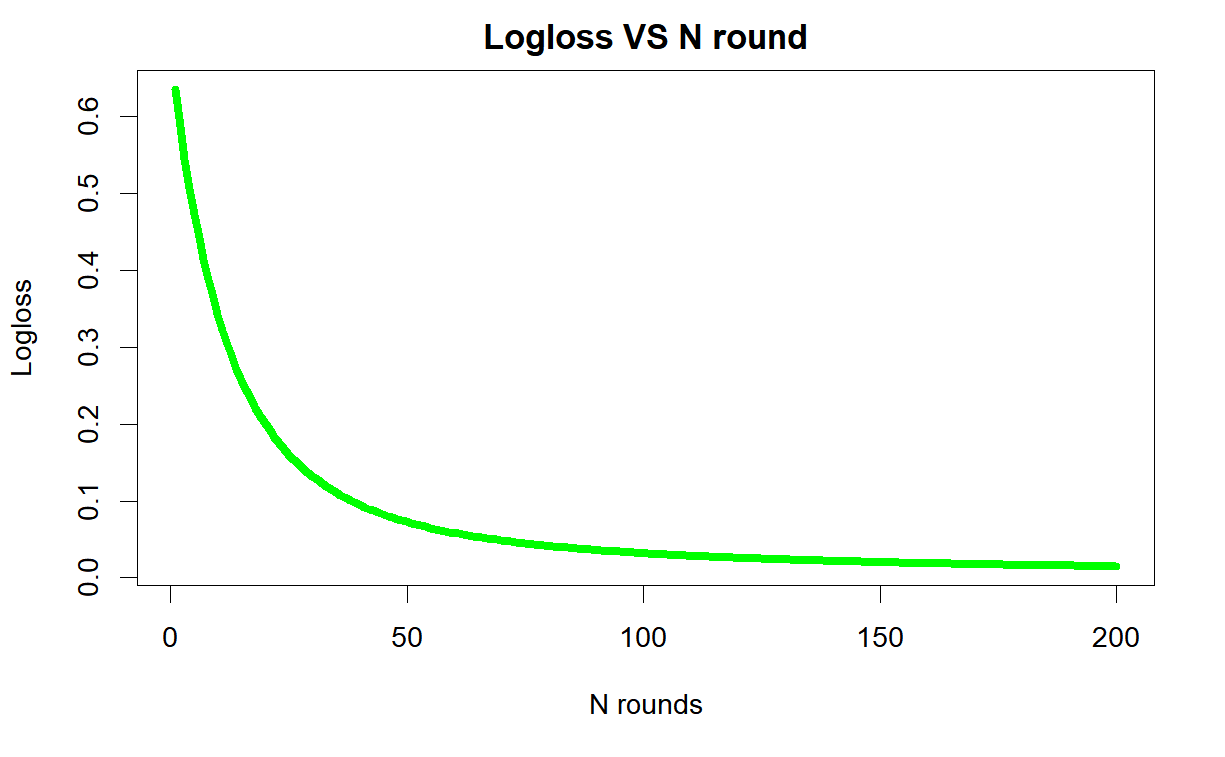
***Additional Variables Used***

The following variables are the variables that we used, but we didn’t perform any feature engineering. goal, numfaces\_project, numfaces\_creator, male\_project, male\_creator, female\_project, female\_creator, smiling\_project, smiling\_creator, minage\_project, minage\_creator, maxage\_project, maxage\_creator, accent\_color, avg\_wordlengths, avgsentencelength, avgsyls, grade\_level, affinn\_pos, ADV, NOUN, ADP, PRT, DET, NUM, CONJ, ADJ.

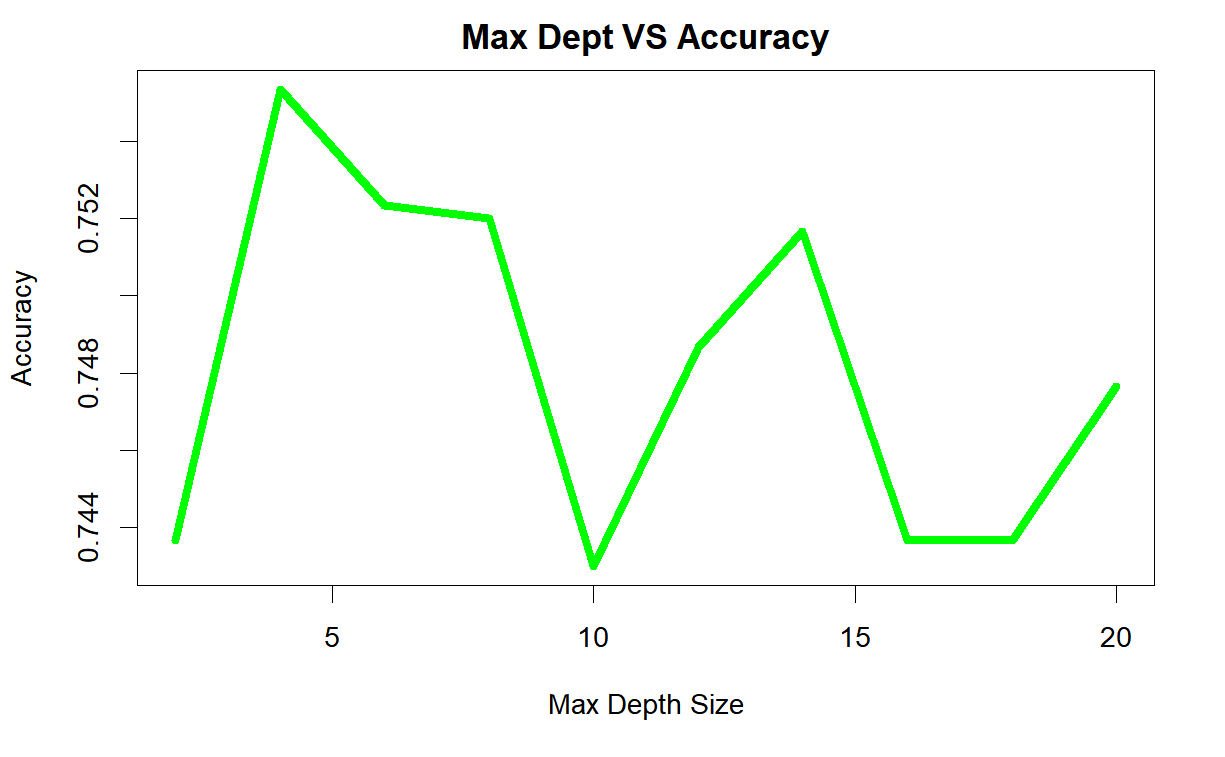
### Holdout Process & Modeling

We utilized two different kinds of holdout processes. When selecting our final model for predictions, we split our original training data set into 70% for training data and 30% for validation data. This split was used on every model to compare accuracies and choose the best-performing model, as seen in Table 2. We found that XGBoost had the highest accuracy based on our split of the training data set.   
 For our best model, XGBoost, we implemented cross-validation for hyperparameter tuning, specifically for the argument nrounds/number of iterations. This process included using the function xgb.cv with 5 folds which returns the best number of iteration (nrounds) for the XGBoost model based on the highest score (log of AUC).

Table 3 shows the performance improvement from features created from the unstructured text columns (blurb, caption, reward description) and external dataset (Dow Jones Industrial Average data). 



*Figure 4. Nrounds Accuracy Plot*



*Figure 5. Max Depth Accuracy Plot*

## Conclusion

### Takeaways

Our group did well on working together to brainstorm new features to include in our model and delegating the work to clean the data, create the new features, and build our predictive models. It was a challenge to coordinate tasks and conduct meetings on a regular basis as every member had a different schedule and availability. We were able to utilize each person's skill set to create a model with high accuracy. Some members were more versed in data cleaning, others more familiar with building the models or feature engineering.

One key challenge of the project was applying new concepts from class to our project. Since some feature engineering techniques were less familiar to us, it took longer to comprehend, write and implement the code. Additionally, the data set was very large, so we had to figure out how to test our code efficiently and adjust to different datasets between debugging and training our final model. Even though we were prepared to implement more models (like KNN), its lack of feasibility/time consumption led us to discard them.

If we were to do this project again, we would thoroughly comment on our code as we wrote it. Over the span of around a month, this project ended up requiring hundreds of lines of code, and it was easy to get lost in what we wrote before or the thought process behind it. This also helps your teammates to understand any changes to the code or for troubleshooting errors. However, one key change we would implement is to delegate the tasks more heavily and have a dedicated lead which would give a much needed structure to our project team. If we had more time to work on the project, we would further tune our hyperparameters and experiment with more feature engineering utlizing unsupervised learning methods. We also would’ve incorporated more than one external data set, such as geographic data or other crowdfunding websites, for additional information and features.

We got many important insights from this project. Most were affirmations of knowledge we had in theory, and some were new to us. Early on, we affirmed that methods like one-hot encoding (in our case for XGBoost) and feature split-binning would be an important part of this project. It helped us rescale the data and allowed representation of categorical data in a more expressive way, which in turn increases accuracy. It would not only help us prepare the proper input dataset which is compatible with our algorithm requirements but also help improve the performance of our models. Selecting the right hypertuning parameters also helped with increasing the accuracy of our models by minimizing the objective function over a dataset. We found some unique things in our models, such as how the length of blurb and caption columns is really useful. We also found that the description length of the captions and count of rewards also impacted our accuracy in a significant manner.

For students starting this project next year, we would tell them to regularly meet with your team (in-person would be quicker) and delegate work with specificity and urgency. Having teammates is extremely beneficial when your code isn’t working or brainstorming how to create a specific feature.

Our main business takeaways are that imperfect predictive information is better than just historical information, working as a team can create a better model, and each Kickstarter campaign has many different features that can contribute to the success of a project. Every approach in predictive analytics has an expected error. Improvements to the model are never-ending since a predictive model with 100% accuracy is impossible. However, we were able to derive new information from existing data that has the potential to help all stakeholders on Kickstarter. More money can be crowdfunded, and time can be saved with this new information. We also learned the value of working as a team. Different backgrounds and experiences allowed multiple perspectives on the problem, specifically during feature engineering. We were able to improve our model by coming up with new features to create every time we submitted our predictions. Through our developed model, knowing which feature variables contribute to the success of a campaign will help future users tailor their campaigns and reach their fundraising goals.

## 