Predicting Project Approval

Abhinav Ganesh
Data Inspired Young Adults

Abstract (about 50 words)

DonorsChoose released a dataset with information on project applications sent to the organization and the status of whether or not it approved these projects. Using the decision tree and random forest models, I seek to use the variables total resource cost, number of previously posted projects, length and number of essays submitted; month, day, year, and hour of submission; length of the resource summary, length of the title of the proposal, teacher prefix, project grade category, project subject category, and school state to predict approval of a project. I measure success through project accuracy and f1 scores, evaluating a model as successful if it is able to predict accuracy better than a model that would only guess that a project is approved. Without balancing the data, no model was successful. After balancing data, both the optimized decision tree and random forest models were able to perform significantly better than this baseline. However, the accuracy of my models was still very close to the baseline score, and was poor relative to the maximum possible values of the accuracy and f1 scores. Future improvement may lie in investigating and analyzing the contents of the essays in the project application, as well as doing case studies of the relatively rare project applications that were not approved.

Introduction/Motivation

https://www.kaggle.com/c/donorschoose-application-screening

The organization DonorsChoose is a US-based nonprofit organization that creates a platform connecting teachers in need of donations with willing donors. Teachers' applications for donation funding are first screened by DonorsChoose before they are approved and posted on its website. DonorsChoose has recorded and released a dataset which includes information on the teacher's prefix (which could be used to identify gender), the state of the school, the time of submission of the proposal, the grade, subject categories, project subcategories, title, essays in response to 4 prompts, the resources the project needs, the number of previously posted applications by that teacher, and whether or not a project is approved. When a project submission is "approved," it gets the privilege to be posted on the DonorsChoose.org website.

Data science techniques can use prior information on projects and their approval in order to make a prediction on which future projects will likely be approved, enabling DonorsChoose to easily identify projects that need further review. An accurate predictive model will allow DonorsChoose to fund more projects faster.

Given my limited knowledge about Data Science algorithms when starting the project and lack of knowledge about Natural Language Processing, I sought to create a useful model without

analyzing the content of the essays in the project proposals. To maximize performance, I sought to use much of the other information provided. Thus, I asked the question:
Can I predict whether a project submitted to DonorsChoose is approved based on the number of previously posted projects, school state, essay lengths, number of essays, subject categories, and total resource cost?*

I hypothesized that I would be able to predict approval to a moderate degree based on said variables after doing exploratory analysis that suggested these variables might have a significant influence on project approval. Several of the quantitative variables appeared to have a trend related to approval: approval appeared to correlate positively with the length of the second essay, and teachers who were approved had posted more previous projects and generally had lesser total resource costs. I believed these trends were significant and would be useful in predicting approval.
Several challenges had to be overcome before a model could be used:
Separate Datasets: Information on the resources requested for a project and their prices was separated into a different dataset than that with the rest of the data for each project.
Outliers: Data that were outliers could be a cause for noise in the dataset which might distract a model from learning a trend.
Null/NaN Values: Data that holds null values could not be fed to a machine learning model.
Categorical Values: Categorical Data such as subject category, teacher prefix, and state held string values but only numerical values could be fed into most machine learning models.
Imbalance in Data: Approximately 85% of the projects in the dataset were approved and only approximately 15% were not approved. This large skew in the variable being predicted might lead to a decreased accuracy of a machine learning model.

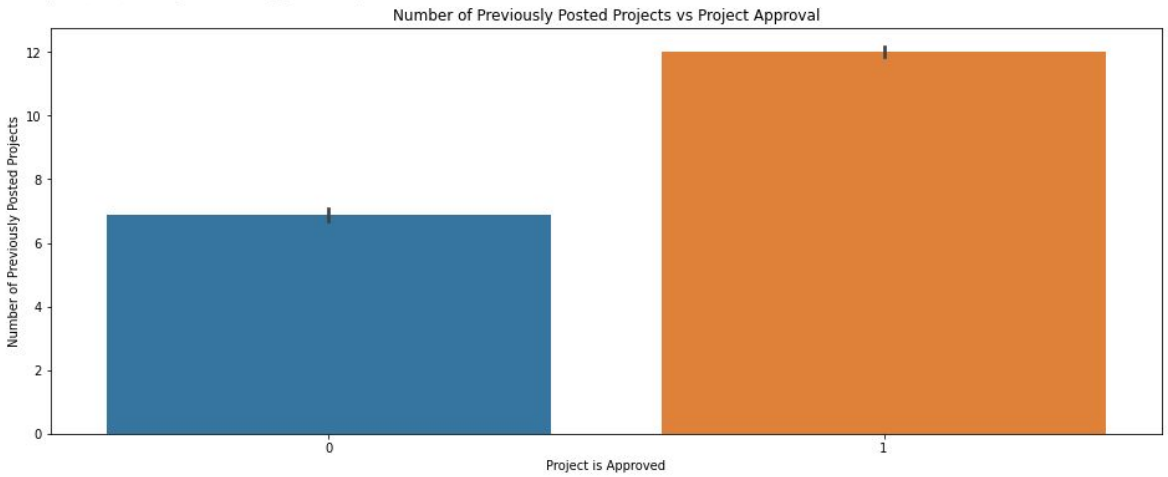
Previous Related Work

Outliers: In order to solve the noise created by outliers in the data a general approach is to simply remove outliers from the model altogether.
Null/NaN Values:There are several approaches to solving the issue of Null data using the pandas library. The simplest is to remove the data using the dropna function. One can also use the fillna function to replace null values with a default value. A third solution is to use other fields in the data to predict a potential realistic value for the Null data.
Categorical Values: For categorical values which are cardinal and can be ranked relative to each other, scikit learn has a label encoding method. This allows the user to convert non-numerical data directly to numbers based on each unique value. Noncardinal categorical variables instead must be one-hot encoded. This involves creating new columns based on all the unique values of a variable which can then take boolean values to represent the value of the variable. The scikit learn library includes a one-hot encoding method. The pandas library includes one hot encoding through the get_dummies method.
Imbalance in Data: The general approach to solving an imbalance in the dataset with a machine learning algorithm is to balance the data towards a more uniform distribution of the

frequencies of the different values of the variable involved. This is done by either looping underrepresented values or dropping/ignoring overrepresented values.

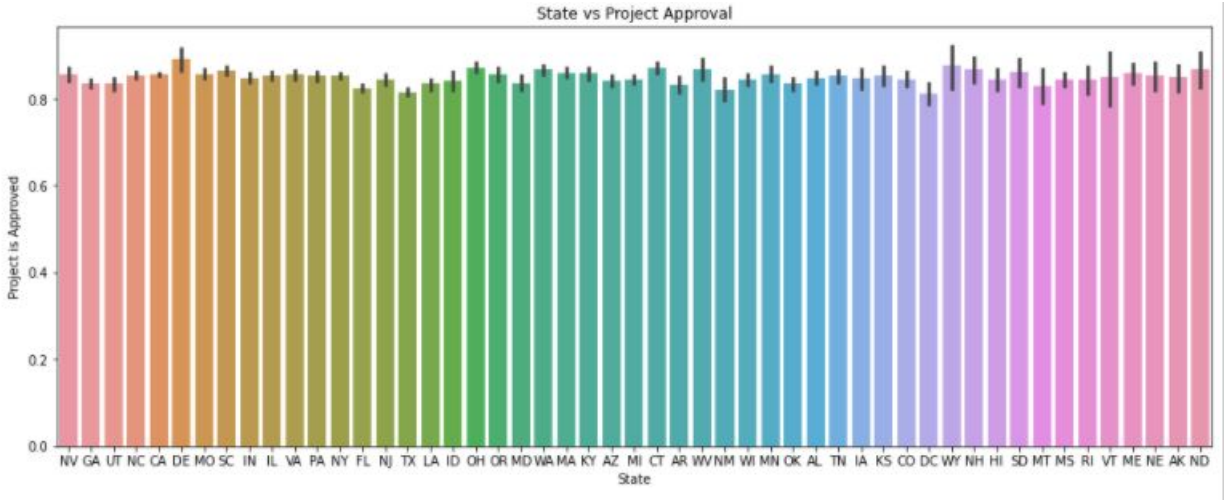
Method/Rationale/Approach

In order to solve the problem of predicting project approval, I intended to use the variables school state, essay length, number of essays, subject categories, and total resource cost as predictors. As mentioned earlier, I had no experience with natural language processing so I did not believe I would be able to create a useful analysis of text data from the essays. Thus, I decided to focus on other variables that appeared to have an influence on project approval based on my exploratory analysis*.
Justification for my usage of these variables are as follows:
Figure 1:



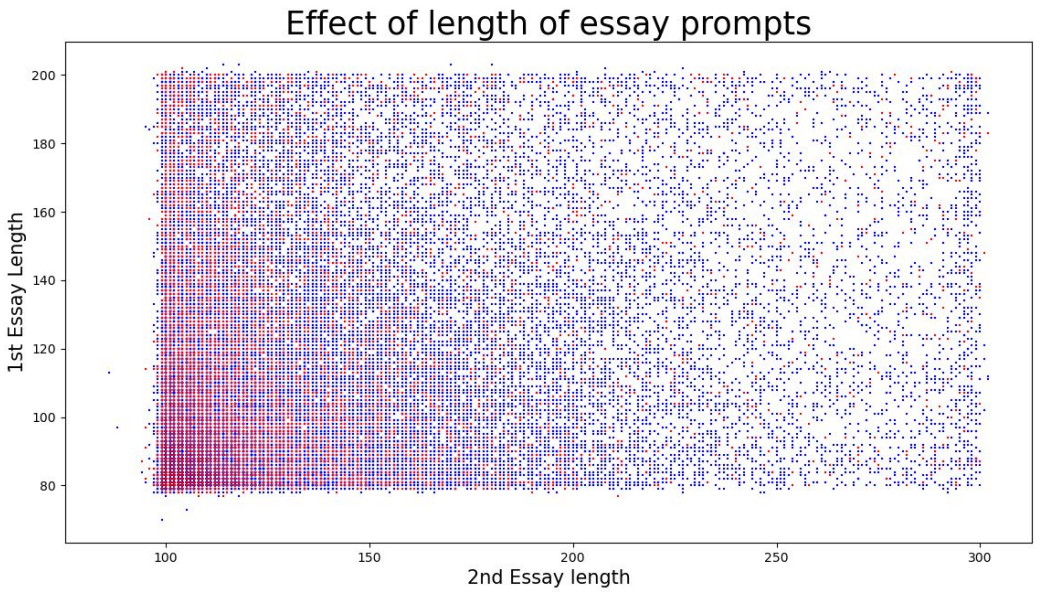
Projects that are approved are submitted by teachers who have posted a greater number of projects than projects which were not approved, on average. The error bars in each category do not overlap which suggests that there may be a statistically significant difference between the number of previously posted projects between the two categories.

Figure 2:



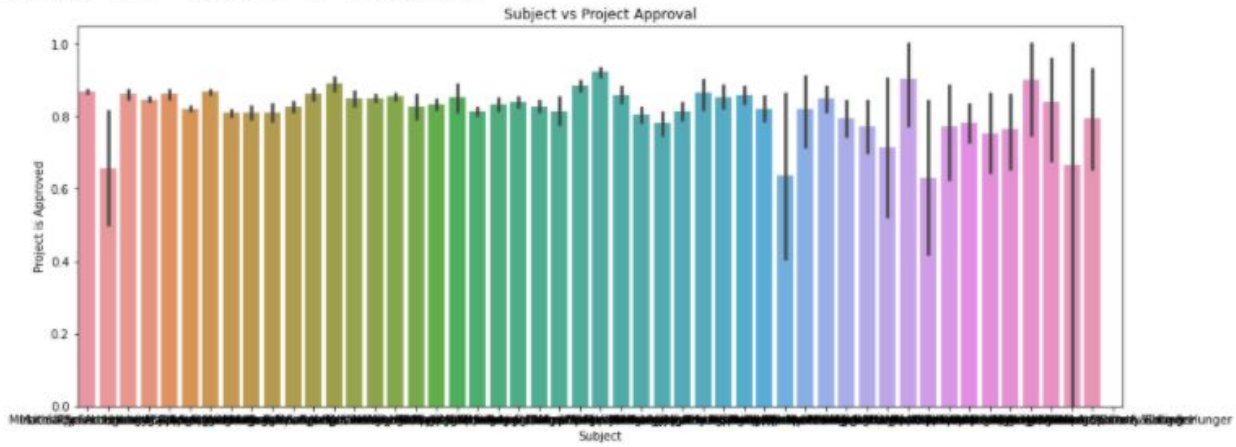
The difference project approval rating between some states may be statistically significant because the error bars do not overlap.

Figure 3:



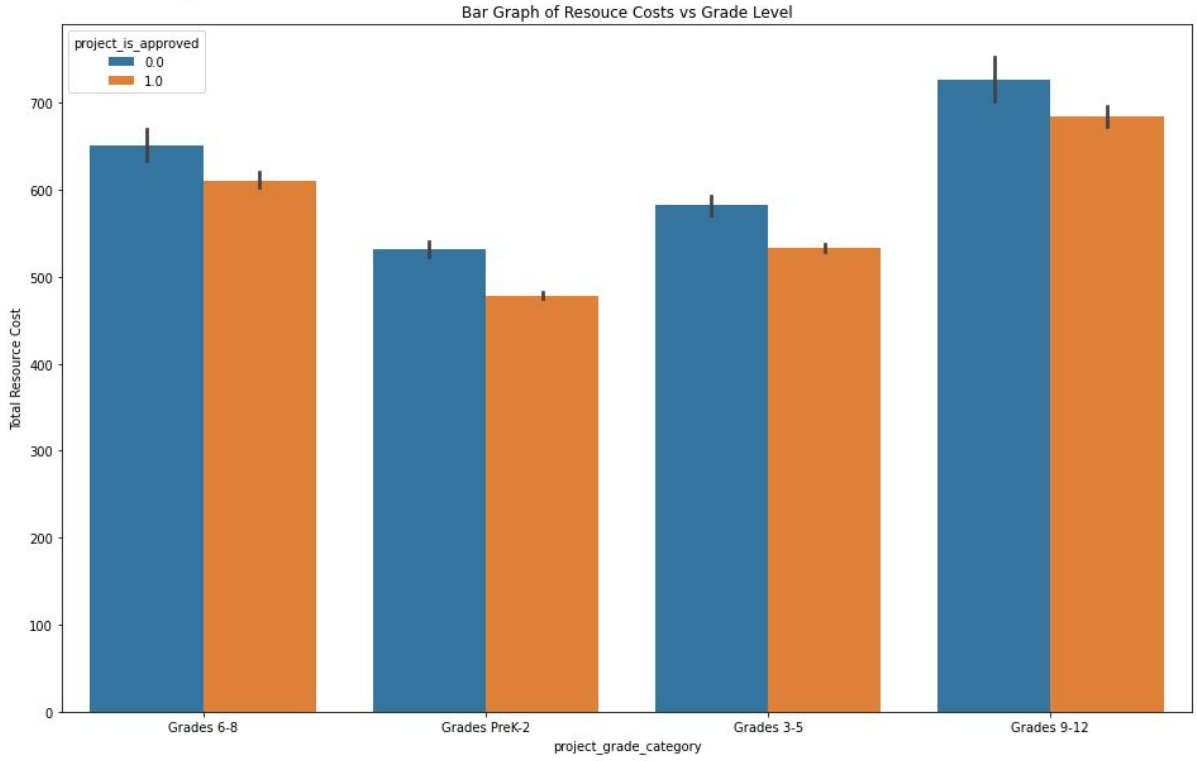
The ratio of projects that are approved to projects that are not approved seems to increase as the length of the second essay increases. The length of the second essay appears to have a significant influence on whether a project is not approved (red) or is approved (blue).

Figure 4:



The difference project approval rating between certain subjects may be statistically significant because the error bars do not all overlap.

Figure 5:



Total resource cost can be calculated by multiplying the price and quantity desired for every unique resource requested in a project and then summing these values together. Projects that are approved generally have lower total resource costs across all grade levels, and this is likely statistically significant because the error bars between the two categories do not overlap.

*After initial poor performance of my models, I found that including more variables as predictors would likely only help improve them. For this reason, I included the Month, Day, Year, and Hour

of submission of a project submission, as well as the length of resource summary and project title.

Online research showed that a train-test split ratio of 80% to 20% was in the optimal range and thus I used that split ratio for my project.

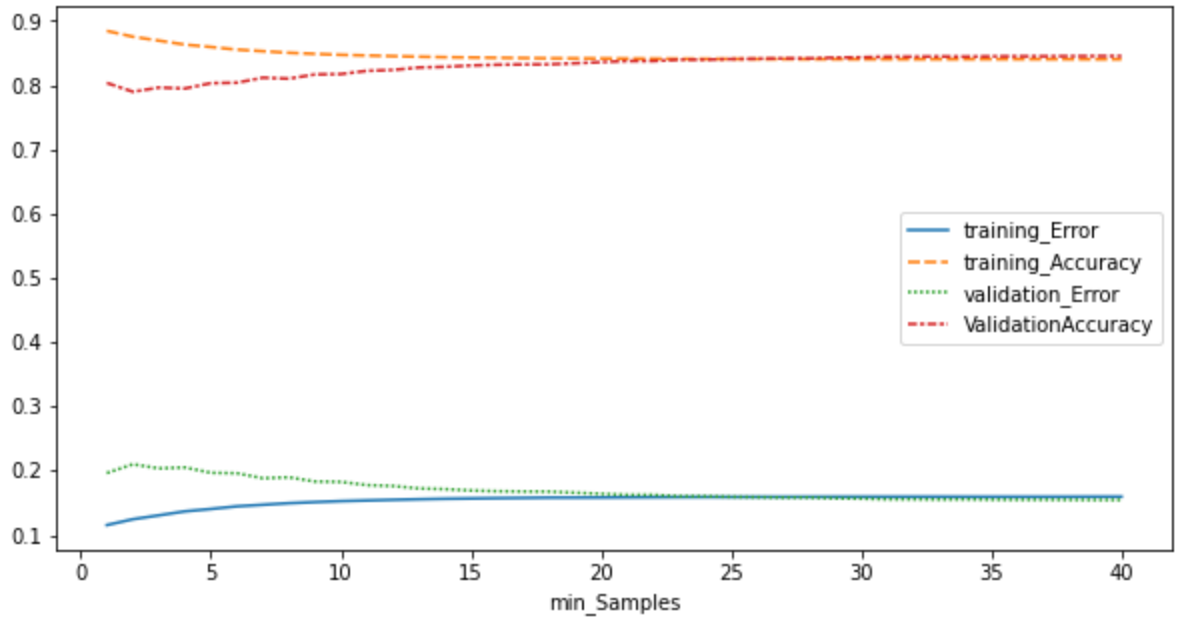
I planned to use scikit's decision tree and random forest algorithms and change hyperparameters such as maximum depth, minimum samples at leaf node, and number of trees (random forest only) to optimize them for best performance. Because the data set was not balanced in regards to project approval, I used both the accuracy score and f1 scores of these models to evaluate their success. With an unbalanced data set, the f1 score, which takes into account the false positives and false negatives that a model outputs rather than only looking about right and wrong predictions, is a better evaluator. However, after balancing the dataset, I turned more to accuracy score, which was a more simple to understand and intuitive measure of the success of a model, as accuracy score was a better evaluator with a balanced dataset. Furthermore, only if the model's predictive accuracy was significantly greater than the baseline approval rate in the testing set (which is the accuracy of a model that would only guess a project was approved), did I deem that a model's performance was successful. This can be found by conducting a statistical test for proportions.

Experiments

I attempted to solve the problem by optimizing the hyperparameters of the decision tree and random forest regressors provided on scikit learn such that the accuracies of my model on the validation set were the highest. In order to do so, I first had to grasp how these hyperparameters affected the accuracy of my model on the testing set.

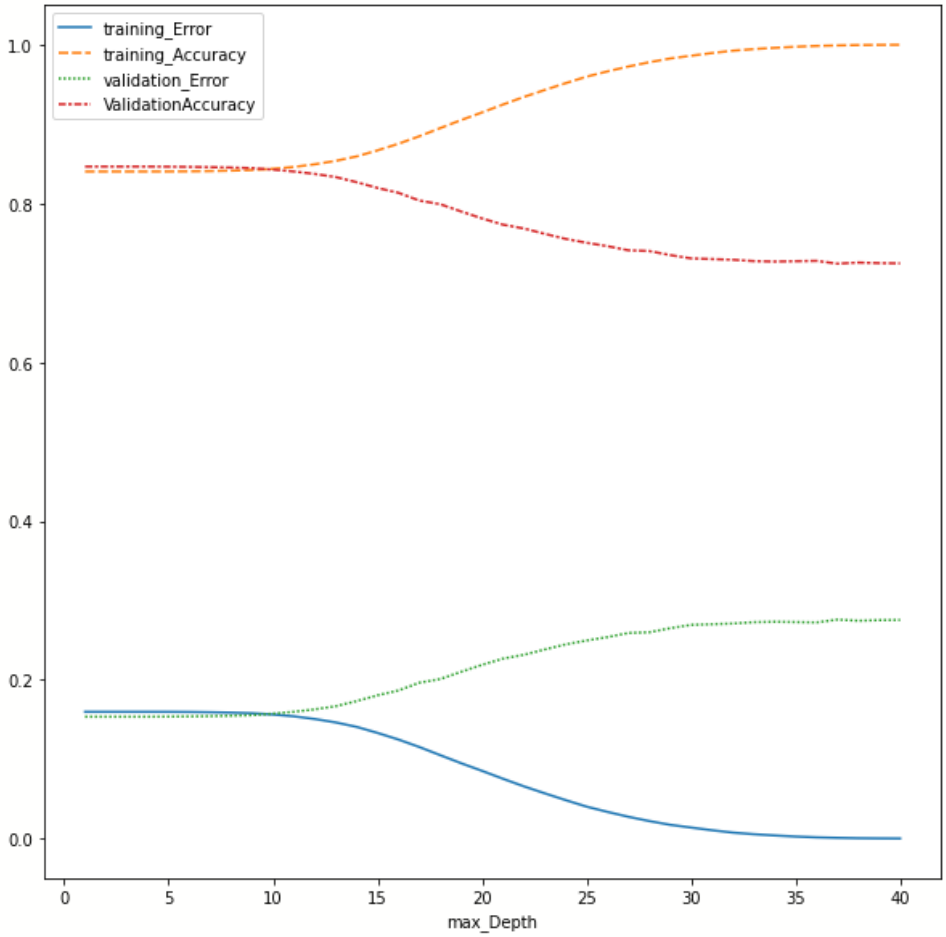
The "validation" set labeled in the following graphs was used as the testing set

Minimum Samples at a Leaf Node:



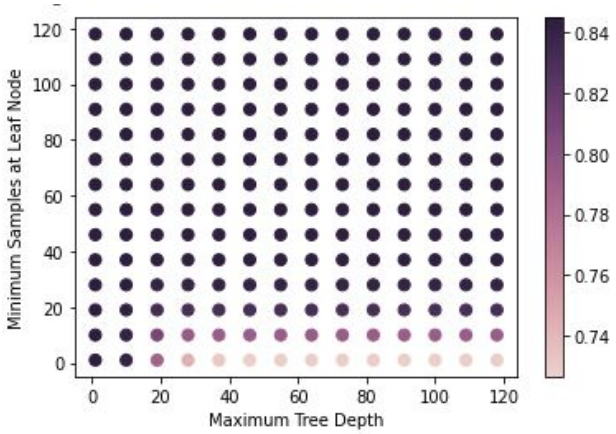
Increasing the number of minimum samples at a leaf node increased the accuracy of my model on the testing set.

Maximum Tree Depth:



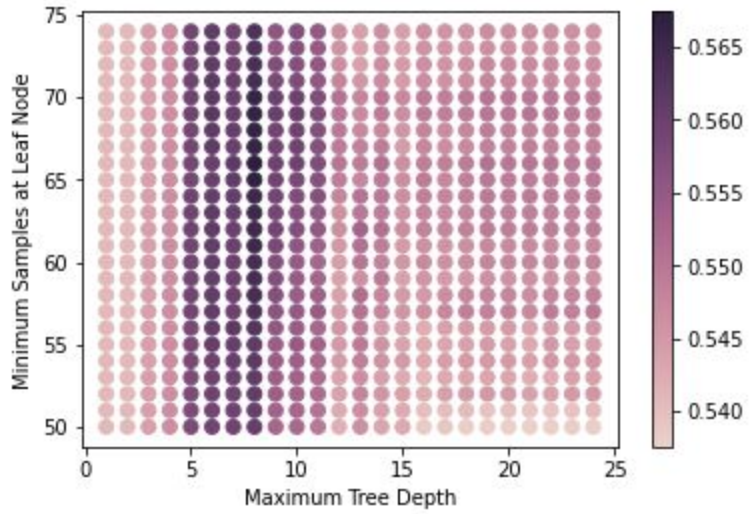
Increasing the maximum depth of my tree generally decreased the accuracy of my model on the testing set.

In order to completely optimize the model I needed to alter both variables at once. Thus, I made an algorithm that created a dataframe of decision trees with different hyperparameter values.



The overall approval rate in the testing set was 0.841936, which was also the greatest value recorded in this graph. This meant the best decision tree I could produce without balancing the data was only guessing that every project was approved. As mentioned before, I compared my approach with the accuracy score a model would get if it simply guessed all projects were approved. This meant that my model was not successful with the unbalanced dataset.

After balancing the dataset so that approximately 50% of projects in both the training and testing sets were approved and 50% were not, I repeated the process and found that I could obtain accuracy scores greater than the baseline approval rate.



There was more evidence of approaching a sweet spot, but the fact that the accuracy of the model remained near the approval rate of the dataset after balancing suggested poor predicting ability of the model.

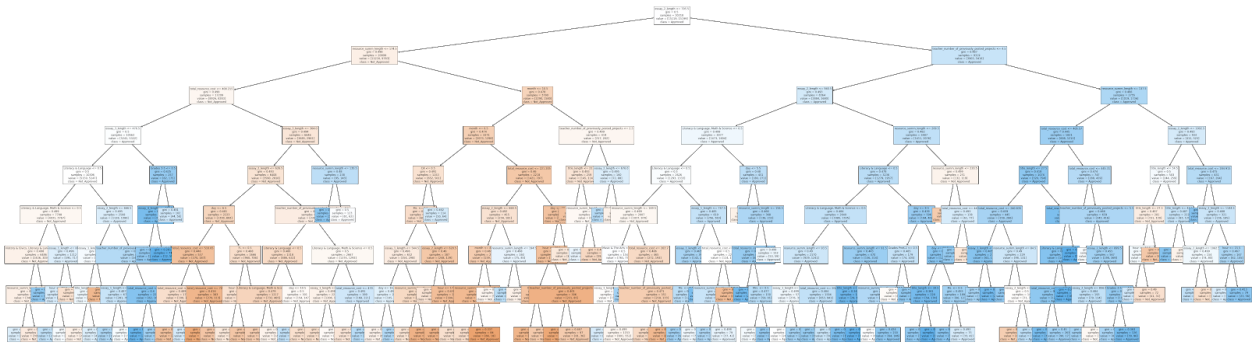
Lastly, by plotting decision trees, I evaluated the importance of my predictors. In the plotted decision trees, the length of the second essay appeared to be an important factor in predicting approval. Longer lengths of the second essay generally led to a prediction of project approval. A similar trend followed with the lengths of the other essays, which agreed with my initial hypothesis based on my exploratory analysis. Although my exploratory analysis suggested that the number of previously posted projects and total resource cost would be good predictors of approval, predictive analysis based on the decision tree plots showed a more unclear pattern.

Results and Discussions

Include plots and tables

Decision Tree:

The best performing decision tree had the following plot:



max_Depth	8.000000
min_Samples	66.000000
training_Error	0.413319
training_Accuracy	0.586681
testing_Error	0.432586
testing_Accuracy	0.567414
train_f1	0.543181
test_f1	0.521942

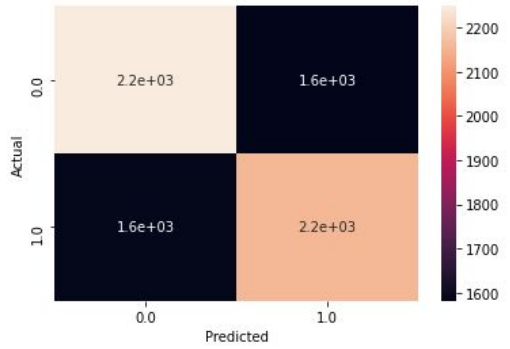
Performance was based on the accuracy score in the testing set as this was after the dataset was balanced.

The best performing random forest model was as follows:

max_depth	9.000000
max_features	16.000000
min_samples	31.000000
number_of_trees	11.000000
train_accuracy	0.598819
train_f1	0.599572
testing_accuracy	0.581794
test_f1	0.576995

Confusion Matrix:

2248	1582
1588	2162



To evaluate if there was a significant difference between the test accuracy of my model and that which would have been found by a model which only guessed projects were approved, I conducted a 2 proportion z test.
0.494723 of the projects were approved. There were 7480 projects total. 0.581794 of them were accurately guessed by my model.

2-Prop-z(x1 = 4351, n1 = 7480, x2 = 3700, n2 = 7480) with the alternate hypothesis that p1>p2 returns a p-value of 6.736*10^-27 which is less than 5%. I can thus reject the null hypothesis, which means that my model likely guesses significantly greater projects correctly than a baseline model which would only guess that all projects are approved.

Conclusions (not necessary to have an overarching conclusion, but if you have one, this is where you would include it)
The greatest accuracy score of any of my models on the balanced dataset was 0.581794, which was a significantly greater proportion than what could have been guessed by a model which only guessed projects were approved. In this way, I was able to successfully predict the approval of a project based on the predictor variables I chose. However, the success of my model was clearly limited with such low accuracy and f1 scores relative to their maximum possible values of 1. This means that while the variables I used were relevant predictors of project approval, they do not account for much of the variation in project accuracy. Seeing as I did not analyze the content of the essays there is definitely room for improvement, and they provide potential for creating new predictors. Furthermore, given that the proportion of projects that were not approved was much less than that for projects that were approved, it might be appropriate and useful to do case studies into certain projects that were not approved that were not necessarily outliers in any of the known data.

Link to my github for this project: https://github.com/Abhi-Gan/DonorsChoose-Models