Predicting the Approval of a School Funding Proposal Using Derived Text-Analysis Variables & Random Forest Prediction Models

Sridevi Pulugurtha
Data Inspired Young Analysts Lead by Dr. Suma Bhat

Abstract

This research presents the investigation of data analysis techniques and approaches used to solve a DonorsChoose.org (Non-Profit Educational Funding Organization) statistical problem from Kaggle. The ultimate goal was to predict the approval of a given school-funding proposal via a machine-learning algorithm only using text-analysis data. This was achieved by only using the 2 essay prompts that were submitted for the funding proposal and generating new data with sentimental and grammatical text analysis.

The novelty of this project lies in the development of the prediction model based on variables that were potentially more definitive than other given fields such as teacher experience, relevant subject area, and time of submission. The prediction model consisted of a classification random forest, consisting of decision trees optimized with hyperparameter tuning addressing sample sizes, variable relevance, and information gain. The performance of this prediction model clearly demonstrates that text analysis is a valid approach to determining proposal approval, and proves to be a worthy competitor to various other types of data analyses and models. Text analysis has been used in a variety of fields from analyzing political influence in social media, all the way to synthesizing scientific literature. For its versatility and unique insight, text analysis holds a promising future in the fields of data analysis in social-sciences and literary studies on large scales.

Introduction

The DonorsChoose.org Kaggle problem consists of a large dataset of samples, each one being an individual proposal for school funding from teachers all over the nation. DonorsChoose.org is a non-profit organization that provides funds for educational purposes involving grades K-12. Each individual sample in the given data set consists of many different variables such as the state of origin, time and date of proposal submission, grade and subject categories, prompt essay submissions and most importantly a binary number indicating the approval of the sample. Using the given data, a machine-learning model is supposed to be built and used to predict the approval of a given test sample or data set. The variance of the solution to this particular problem lies in the different approaches and data analytical techniques used to analyze the given data and develop a model accordingly.

By developing an efficient machine learning model that can predict The approval of a given project, DonorsChoose.org Will be able to fund American classrooms in a more streamlined fashion. The development of models will also help DonorsChoose.org assess the needs of the

organization, uncover new insight from the available data and build an effective system that can fund as many classrooms as possible which is the ultimate goal of the organization. As of before, Donors choose.org used a meticulous, 100% hands-on process to personally evaluate each and every project proposal. Although this is somewhat effective, the proposal evaluation process can be much more efficient with the help of a machine learning model that recognizes the optimal characteristics and traits of an investable classroom project or funding proposal.

There are 3 challenging attributes to this project, which are described in the Kaggle problem parameters:

Performance: The prediction model should be accurate and be as effective as possible when determining acceptable projects and motivating donations. The ultimate goal is to come as close to accurately predicting the approval of a project; this is to be able to provide donations and approval insight as productively as possible. This challenge, or rather parameter can be achieved by modeling based on the data with relevant features and variables and developing the model itself with appropriate definitive parameters. Later, testing, validating and balancing the model will also play a key role in tuning and optimizing it.

Adaptable: Models should be implementable and realistic. The insight and results they provide should not only make numerical or theoretical sense, but also common sense. The results and tendencies of the model should match with those of a human evaluator and be consistent. Also, in a real-life application such as this, the prediction model should be able to be quickly implemented and planted in the organization's interface. The model should be compatible both technical and content-wise with the organization's interface.

Intelligible: Finally, a good model should be able to be changed as deemed necessary. This means the code and model development should be legible and interpretable. This can be achieved by making the code legible and by commenting on all appropriate parts, explaining the purpose. The code should also be fit to adapt and change.

The core of the hypotheses of this project adheres to the unique approach to developing a model. The goal is to develop a model solely based on text-analysis variables and indexes from the 2 prompt essays included in the original data.

Can an effective prediction model be built solely on text-analysis (content/project-related) variables and indexes derived from a fraction of the original data? In other terms, are newly derived, content-related features as effective as original, non-content-related features when developing a prediction model for project approval?

Previous Related Work

The data and parameters of this project were found on Kaggle, a large community of data science and machine learning practitioners. Here, along with providing educational tutorials and notebooks on data science, scientists and engineers help develop algorithms for various

organizations. As for the algorithm itself, DonorsChoose.org has already initiated its algorithm competition and has selected the best algorithm from all the participants as of 2018. However, the data and problems are still available to work with and download, as it is still a useful learning opportunity. The Kaggle problem provides legitimate data and problems, allowing for experimentation and creativity while working in a real environment with real statistics and parameters. It is important to note that even though the developed algorithm was not fully implemented, it was still tested; More so, the validity of the data analytical approach and solution's usefulness still remains.

As for this particular project, it was done in coalesce with other high school students who were solving the same problem with their own analysis and approach. Many inspirations and conclusions were drawn from each other's exploratory analysis and ideas as well. Whereas the specific approach to solving the problem differed, the overall ideas and methods were mirrored.

In more technical terms, the text analysis approach used to build this algorithm has been worked with extensively in the past. Multiple NLP (Natural Language Processing) libraries exist and are commonly used to perform various types of text analysis. Some are NLTK, TF-IDF and SpaCy, both of which include text analysis functions in Python, such as lemmatization, tokenization, word analysis and even sentiment analysis.

Rationale and Approach

The ultimate goal for this project was to develop a supervised (defined) machine learning algorithm that could predict concrete values for whether a given project is approved or not.

The first step taken to achieve this is to do some exploratory analysis on the provided data. Using Seaborn and Matplotlib, along with Pandas, the data (in CSV format) was downloaded and graphed. Different variables were graphed alongside each other and the target variable, the approval of the project. While certain vague trends were seen, nothing was especially definitive or certainly indicative of project approval. This seemed fathomable when taking into consideration the nature of the used variables. Fields such as subject category, grade category, state of origin, and the teacher's past experience with DonorsChoose.org did not seem to be variables that directly affected the quality and nature of the project itself. Under the assumption that DonorsChoose.org was not being selective of project acceptance based on uncontrollable factors and necessity. Uncontrollable factors include associated grade level, state of origin and even subject category as many teachers are obligated to work in and under certain conditions. These, under the assumption that DonorsChoose.org is not targeting or marketing themselves to a specific group of teachers/classrooms, may not be definitive features for acceptance. Along with uncontrollable factors, factors of mere necessity might not be valid variables for acceptance either. The product, along with its quantity and purpose should not be definitive factors of acceptance as they were supposed to already be optimized by the teacher based on the needs of his/her classroom.

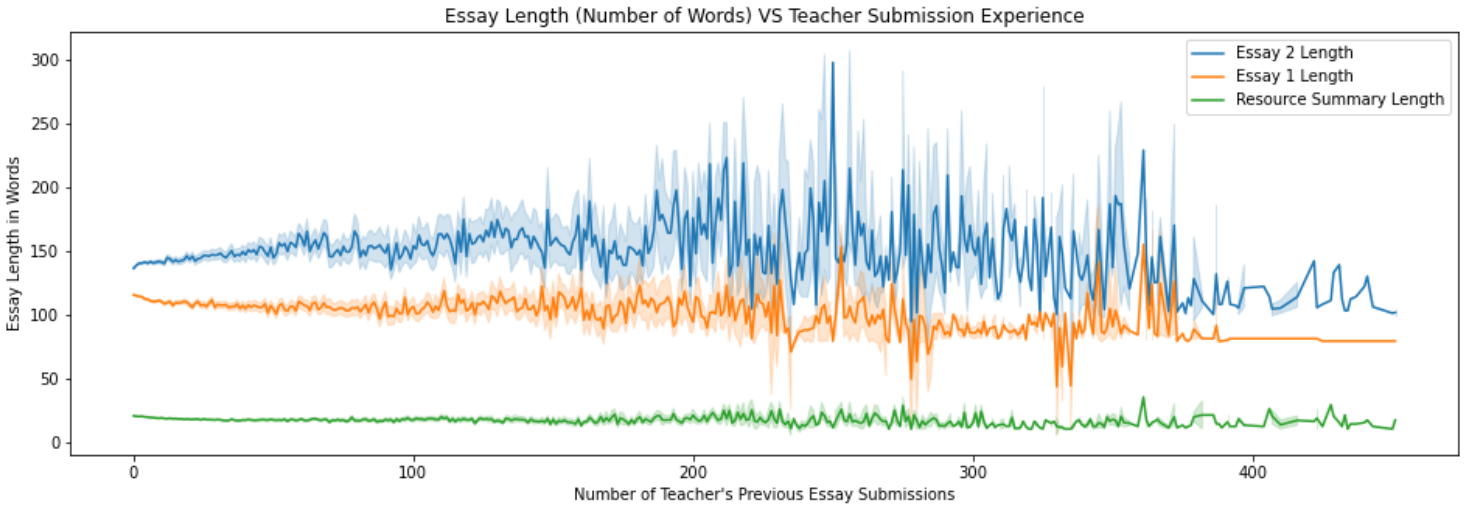
These ideas, as valid as they may or may not be, triggered the thought of what variables are truly definitive from a non-data-analytical, common-sense perspective. Teachers, or rather any applicants of any kind tend to enter general statistics while counting on the fact that they can explain further in the essay prompts. This is known from personal experience. It can also be determined from an evaluator's perspective that the essay prompt takes the most amount of time to go through when looking at a project proposal. Taking these two statements into consideration, The hypothesis was made that the quality of the essays, more in terms of grammar and emotion rather than actual content may have a considerable effect on the approval of a project.

The execution and testing of this hypothesis was carried out with the usage of text analysis variables such as sentiment indexes, dominant emotions, verb indexes, descriptive indexes, and personal pronoun indexes. These variables were calculated using the NLTK library in Python and were used to detect trends and develop models that could ultimately predict the approval of a project.

The success of this hypothesis would be determined by the conclusions made on the detected trends using graphical and exploratory analysis along with the performance of the prediction model developed with the variables.

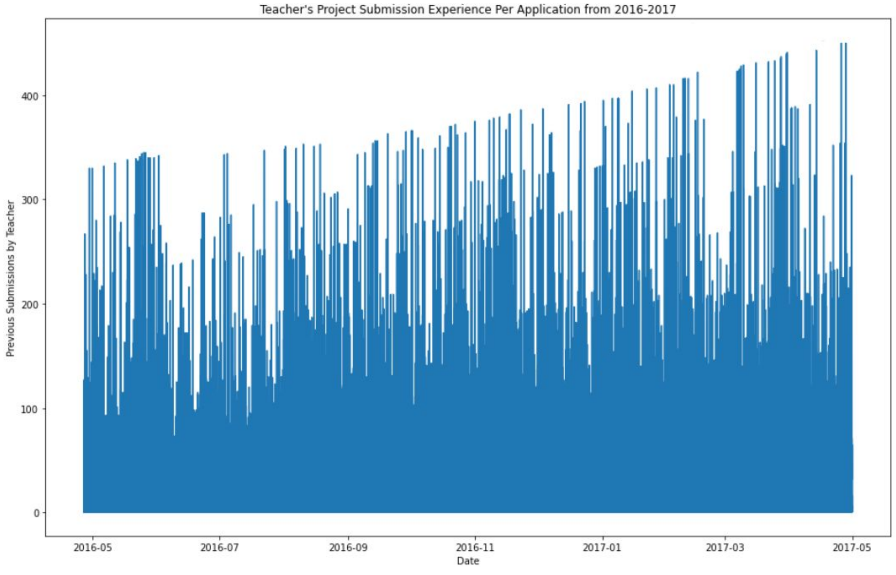
Exploratory Analysis – Unsupervised Machine Learning

Prior to developing the predictive models, exploratory analysis was performed on both the original data and the new, derived data. This initial stage of the project consisted of unsupervised machine learning, where the variables were explored and analyzed for trends. Using matplotlib and Seaborn, some basic exploratory analysis was done on the variables to gather bearings on general trends and behavior.

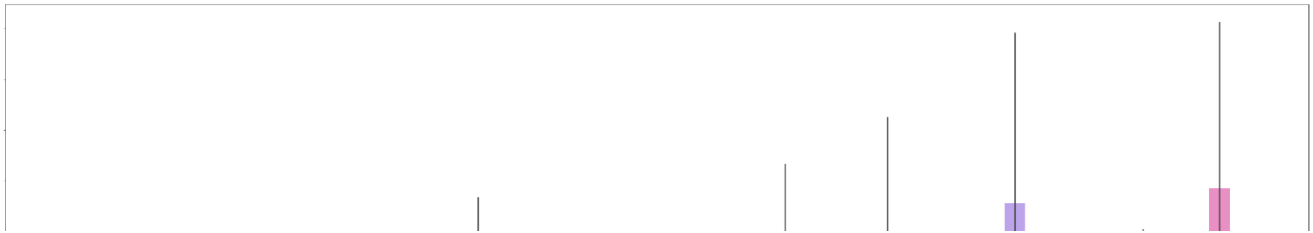


These linear plots depict the lengths of summary and essay prompts which were submitted in a project proposal in relation to a particular teacher's previous experience with DonorsChoose.org. It is clear that there are a few noticeable spikes in the middle of the graph potentially indicating that teachers write the most in their proposals around this point (225-275 Previous Projects) in their Donors Choose experience. The fact that dynamics are seen mainly in Essay 2 makes sense, as Essay 1 and the summary are not very impactful as Essay 2 in terms of defining the project itself. This particular trend is seen throughout the data, as Essay 2 is a large determinant in the acceptance of a particular proposal. The information that can be extracted from this particular graph is the fact that teachers, as they gain experience with this project, go through certain phases. They start out with relatively shorter pieces of writing, tend to increase and then decrease as their experience grows. However, this graph does have some inaccuracies as not all essays can be treated the same with respect to their appropriate teachers. Different teachers have different subcategories, categories, grade groups, and levels of passion and dedication towards their particular projects. All of these features are not addressed in this graph. However, although there are multiple minor factors that could affect this particular conclusion and that the graph is pretty sporadic, it can be decided from this particular graph that there is a "confidence" phase, and a lapse phase in which the length of essays somewhat taper out.

This line chart shows the submission date of teachers based on the number of previous submissions made with DonorsChoose.org. This line chart shows a pretty clear trend pointing to how more experienced teachers with higher submission numbers tend to turn in their project proposals at later dates. Although there are many other factors that go into submission dates and timings such as the personality of the teachers themselves, and whether they procrastinate, a pretty clear best fit line shows positive correlation between the submission date and the submission experience of a particular teacher. This graph, and the graph before both use information about the teachers' previous project submission factor. The purpose of this is to be able to find out how teachers optimize their chances of being accepted, and what they have learned by doing the funding project multiple times. Although the method seems effective, it is dependent on the fact that teachers with experience do get higher acceptance rates, but that has not been determined yet in the exploratory analysis.

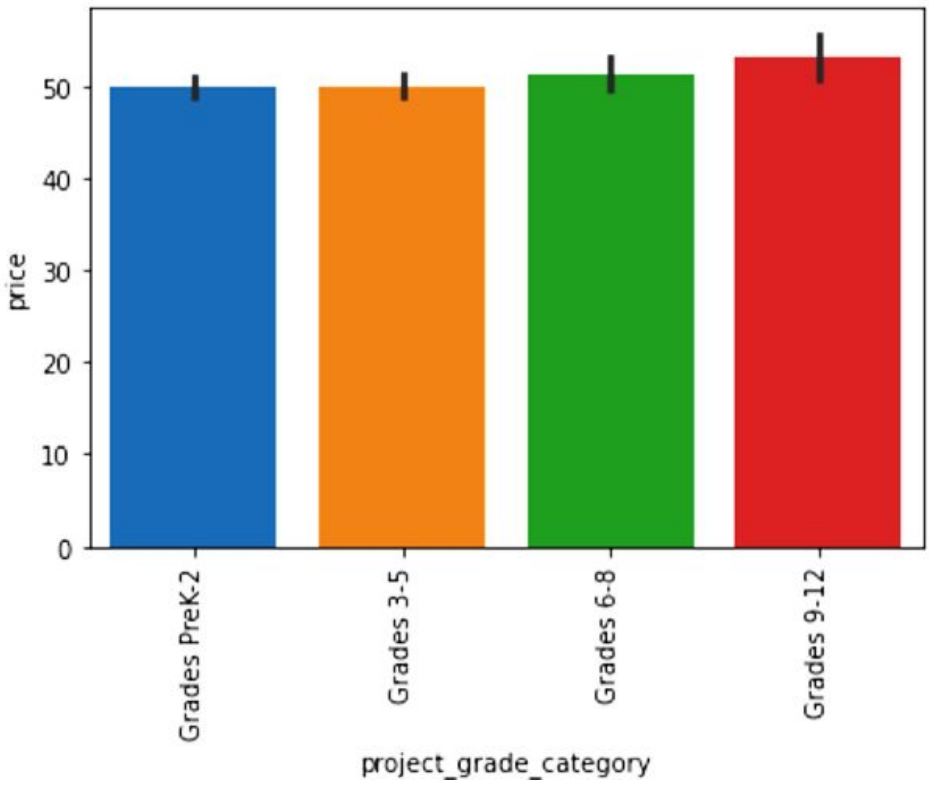


Price of Proposal Projects Based on Subject Category



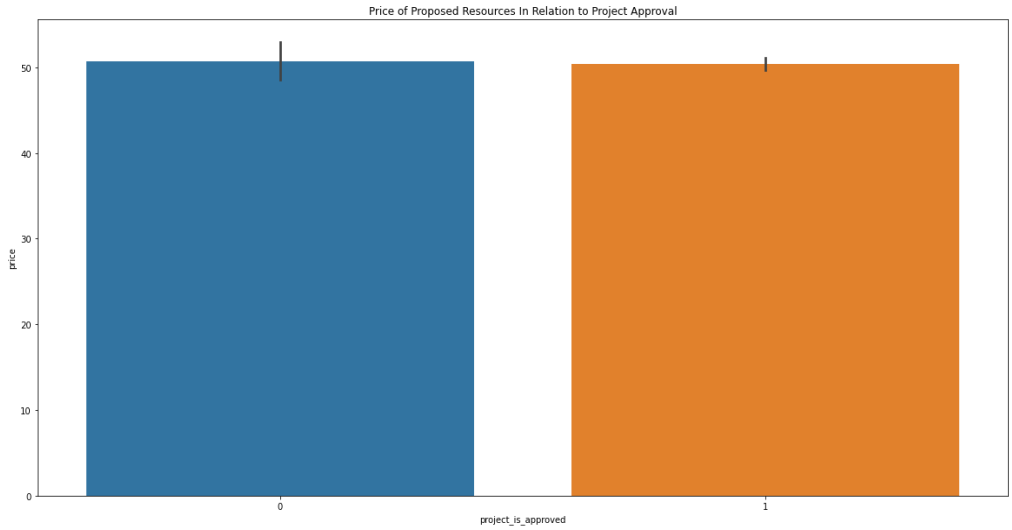
The above bar graph shows the price of the proposal projects based on their subject category. Although this information is not particularly relevant when trying to develop machine learning algorithms for determining the predictability of the acceptance of a proposal, the information can still be used to understand the nature of the proposals themselves. This data provides some rather surprising results, as the highest or rather most expensive projects are in the music and arts, applied learning category. Next however, a more appropriate runner up, is the literacy and language, and the care and hunger category. Then, in third place is health and sports, with applied learning. It is easy to assume that music and arts, with applied learning, has the highest price tag, and therefore is a more prioritized field when it comes to accepting proposals, but in reality, more primitive education oriented projects are of higher priority.

Project Grade Category vs Project Price



The bar graph above depicts the prices of the projects based on the grade category they are in. Similar to the previous graph, this one does not exactly yield any direct information about the characteristics of approved projects, but rather helps give an idea about the nature of the projects themselves. In this particular graph, one can see that all of the bars level out quite decently to some extent. However, the error bars show different implications. The high school level has a larger error bar; this makes sense as high school includes a larger variety of school subjects,

which all have a wide range of budgets as shown in the previous graph. The earlier grade groups have smaller error bars, implying that they work with more narrow subject areas and budgets.



This bar graph demonstrates the relationship between project price and approval. The graph shows that approval/disapproval is quite consistent as both bars are very level. It seems as if the evaluation system has no bias against pricier projects or simpler, lower cost ones. But the error bars have different implications. The error bar of the approval (1) is narrower than the former. This reveals that there is some selectivity about the cost of the project. The more narrow, centered error bar on the approval bar shows that "average" cost level projects are more commonly approved than projects with relatively lower or higher budgets. This graph is useful as it does reveal information about selection preferences regarding the cost of a project. However, it does make sense that evaluators would choose average-cost-level projects more often over others. A well-thought-out budget plan would be cost effective, without being too stingy, resulting in a medium level cost. Cheap/expensive projects are not exactly ideal, at least in the evaluator's terms in which they are trying to find a well-thought and intended project. This graph also contributes to the questionability of definitive certain data features in relation to the approval of projects.

Following the exploratory analysis of the original data, new text analysis variables were developed. After concluding the validity and relevance of the given data, new text-oriented variables were coded, and a new dataset was made.

```
def verb_eval(text):
    lower_case = text.lower()
    cleaned_text = lower_case.translate(str.maketrans('', '', string.punctuation))

    for char in string.punctuation:
        cleaned_text = cleaned_text.replace(char, ' ')

    tokenized_words = word_tokenize(cleaned_text, "english")
    tags = nltk.pos_tag(tokenized_words)
    counts = Counter(tag for word, tag in tagged)
    total = sum(counts.values())
    dict((word, float(count) / total) for word, count in counts.items())
    return dict.get('VB') + dict.get('VBD') + dict.get('VBN') + dict.get('VBP') + dict.get('VBZ')
```

```
def comp_desc_eval(text):
    lower_case = text.lower()
    cleaned_text = lower_case.translate(str.maketrans('', '', string.punctuation))

    for char in string.punctuation:
        cleaned_text = cleaned_text.replace(char, ' ')

    tokenized_words = word_tokenize(cleaned_text, "english")
    tags = nltk.pos_tag(tokenized_words)
    counts = Counter(tag for word, tag in tagged)
    total = sum(counts.values())
    dict((word, float(count) / total) for word, count in counts.items())
    return dict.get('JJR') + dict.get('RBR') + dict.get('JJS') + dict.get('RBS')

def personal_eval(text):
    lower_case = text.lower()
    cleaned_text = lower_case.translate
    tokenized_words = word_tokenize(cleaned_text, "english")
    tags = nltk.pos_tag(tokenized_words)
    counts = Counter(tag for word, tag in tagged)
    total = sum(counts.values())
    dict((word, float(count) / total) for word, count in counts.items())
    return dict.get('PRP') + dict.get('PRP$')
```

```
def prepare_text(text):
    lower_case = text.lower()
    cleaned_text = lower_case.translate(str.maketrans('', '', string.punctuation))

    for char in string.punctuation:
        cleaned_text = cleaned_text.replace(char, ' ')

    # Using word_tokenize because it's faster than split()
    tokenized_words = word_tokenize(cleaned_text, "english")

    # Removing Stop Words
    final_words = []
    for word in tokenized_words:...

    # Lemmatization - From plural to single + Base form of a word (example better-> good)
    lemma_words = []
    for word in final_words:
        word = WordNetLemmatizer().lemmatize(word)
        lemma_words.append(word)
    return str(lemma_words)
```

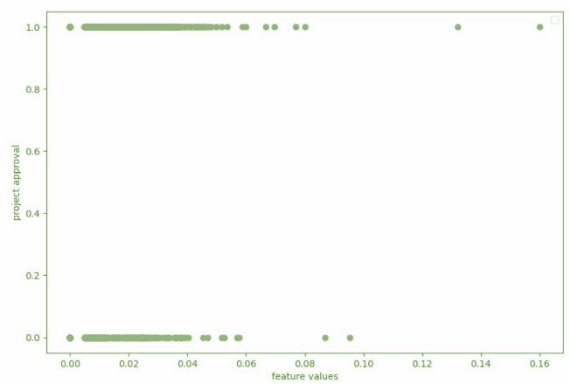
```
def dom_emotion(text):
    te = prepare_text(text)
    emotion_list = []
    with open('emotions.txt', 'r') as file:
        for line in file:
            clear_line = line.replace("\n", '').replace(",", '').replace("'", '').strip()
            word, emotion = clear_line.split(':')
            if word in te:
                emotion_list.append(emotion)
    d = Counter(emotion_list)
    return d.most_common(1)[0][0] if d else None

def sentiment_analyse(text):
    score = SentimentIntensityAnalyzer().polarity_scores(text)
    if score['neg'] > score['pos']:
        return -score['neg']
    elif score['neg'] < score['pos']:
        return score['pos']
    else:
        return score['pos']
```

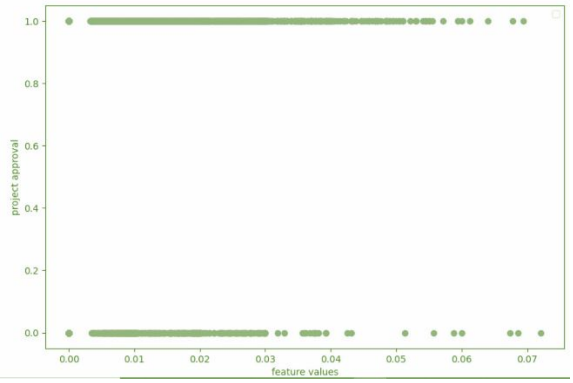
The features that will be used to ultimately determine the target variable are the sentiment analysis index (double), dominant emotion (string), verb evaluation index (double), descriptive word evaluation index (double) and personal/possessive word evaluation index. (double). These variables are extremely relevant to the prompt essay evaluations, which is an important and impactful part of a project evaluation. The content of the essay prompts is a large factor in terms of determining the approval of a project. Therefore the content related features of the essay prompts will be evaluated with these new features using the NLTK Library as shown above.

The sentiment index is an indicative feature that shows the ultimate attitude or overall sentiment of a given essay. it returns either a positive or negative double that indicates whether the overall sentiment is positive or negative, and how intensely positive or negative it is. The dominant emotion feature gives out an emotion detected via a very long list of words and their associated emotions. Using a counter, the most popular emotion exhibited in the essay is returned for the dominant emotion feature. Both of these variables are extremely important as they indicate attitude, sentiment, and emotion, which are very definitive features especially in a charity-oriented situation that involves humanity and need. The parts-of-speech-oriented variables are also important as they give different kinds of indicators about the type of essay. A high verb index can indicate an action-oriented essay that may imply that the teacher is confident and knows exactly what he or she intends to do. A large presence of verbs indicates that a particular teacher has maybe done a lot in the past and may have an agenda for the future as well, indicating a confident teacher with a definitive need. The descriptive word index is an indicator of passion or expressiveness in the essay. adjectives are used to further express and describe their associated nouns, and adverbs are used to further express and describe their associated verbs. This is useful as a high descriptive word evaluation index can imply a very descriptive essay, whose word choice is associated with passion. Finally the possessive/personal word evaluation index is an indication of how intense the teacher's connection with the project is. By using a lot of apostrophes, a personal or possessive connection to the objective/project can be established.

Essay 1 – Descriptive Index



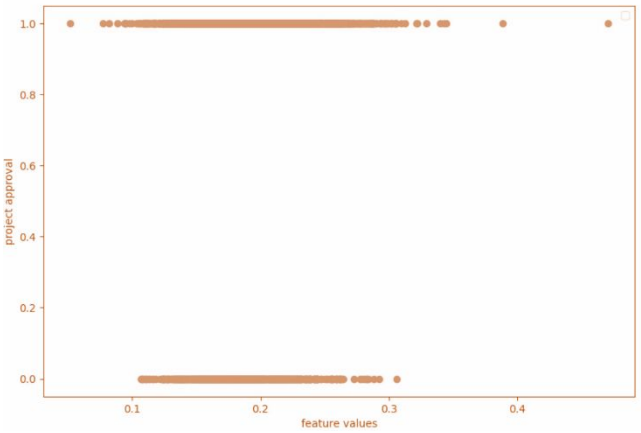
Essay 2 – Descriptive Index



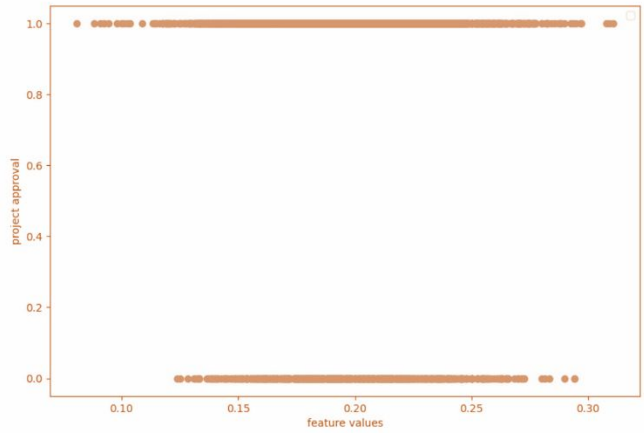
The graphs shown above depict the description index and their relation to the approval of a project for both Essay 1 and Essay 2. it can be seen clearly that Essay 2 seems to have more elaborate language as the variance in terms of the description index is a lot larger and there are a

lot more outliers which indicates that many applicants took the opportunity to expand upon their ideas more eloquently in the second essay. as grammar and vocabulary are very subjective topics in writing, the trends although clearly visible are somewhat vague. Essay 1 and 2 which have higher amounts of approval with the projects that have relatively higher descriptive indexes. This is clearly seen as there are many crowded dots at the project approval value one better further down the X-Axis. These indexes are indicators of descriptive language and expressiveness which can translate into passion and excitement pertaining to the project of interest. The correlation here indicates that the amount of passion expressed has a positive relation with the approval of a project.

Essay 1 – Verb Index

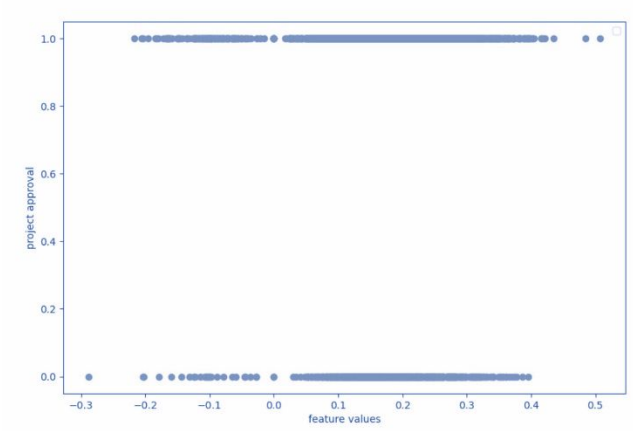


Essay 2 – Verb Index

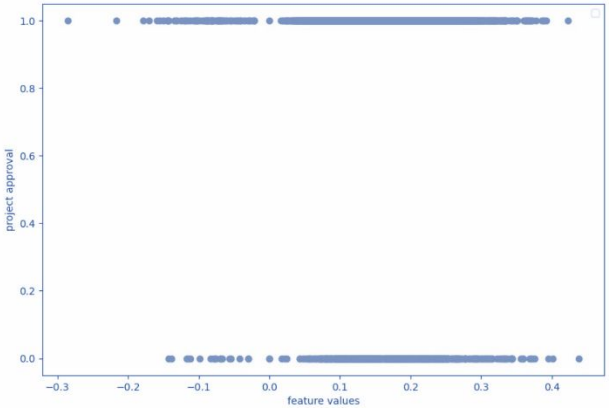


The graphs shown above depict the verb indexes in relation to the approval of a project. The verb index is an indication of how many verbs are detected in the content of an essay. The trends seen in these graphs are slightly more complex As the non-median verb index values, or rather a larger range of verb index value essays seem to have more project approval then the verb index values considered average. Even though this is a strange trend, and defies the original hypothesis that more verbs would mean a more action oriented essay, implying a more appealing project, these trends can still be used in a predictive model as a larger range of verb indexes seem to have a positive project approval correlation than a specific, smaller range.

Essay 1 – Sentiment Index



Essay 2 – Sentiment Index

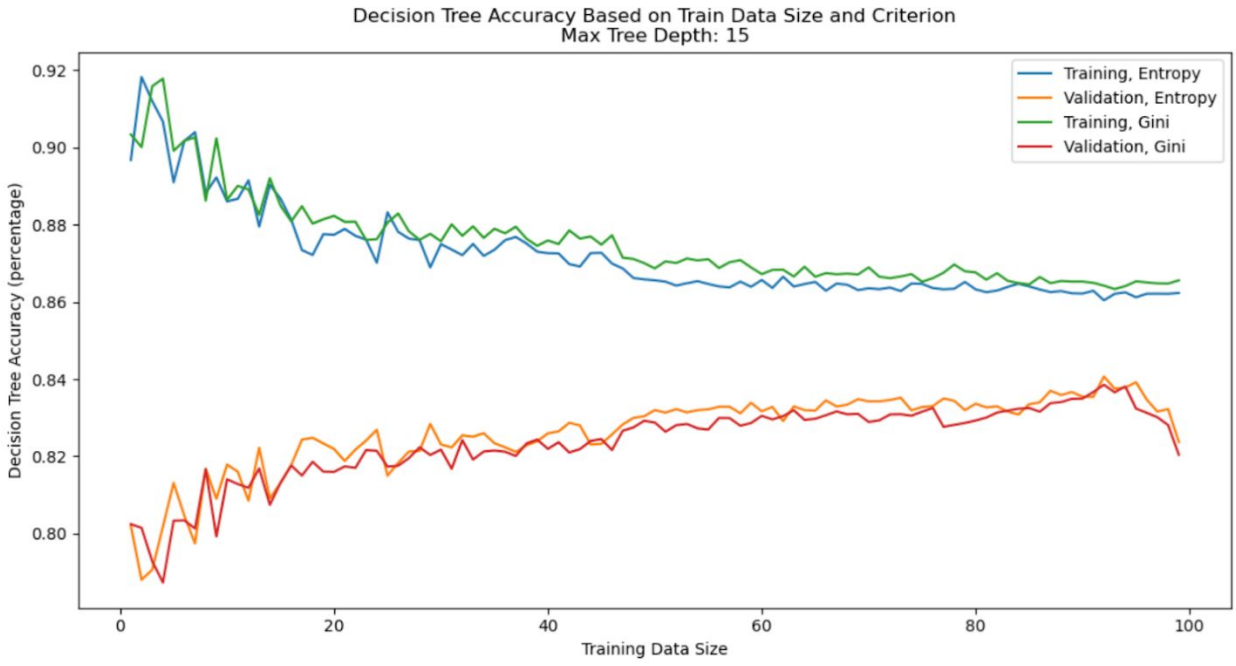


The graphs above show the sentiment indexes of Essays 1 and 2, with their relation to project approval. The graphs show that a stronger preference of sentiment is preferred as in both graphs, there are prominent spaces near the 0s. Also, there is also not much difference between the negative and positive sentiment indexes, as the density of points are similar. This is sensible as the usage of positive and negative words in different contexts may come off differently to the reader.

Predictive Analysis – Supervised Machine Learning

After sufficient analysis of the original and newly-derived, text analysis data, the development of prediction models began. The initial chosen model was a classification decision tree model, and the final model would be a random forest. The supervised machine learning portion of this project consisted mainly of building and optimizing decision trees and random forests to provide the best accuracies.

Decision trees are built from 2 types of features: the target variable, and the remaining, definitive variables. In this project, the target variable is the binary project approval feature, which will be determined with the text-analysis features. Decision trees have parameters that can be varied to optimize the tree performance. The parameters that were optimized included the following: tree criterion, training/test data size, tree depth and minimum leaf samples.

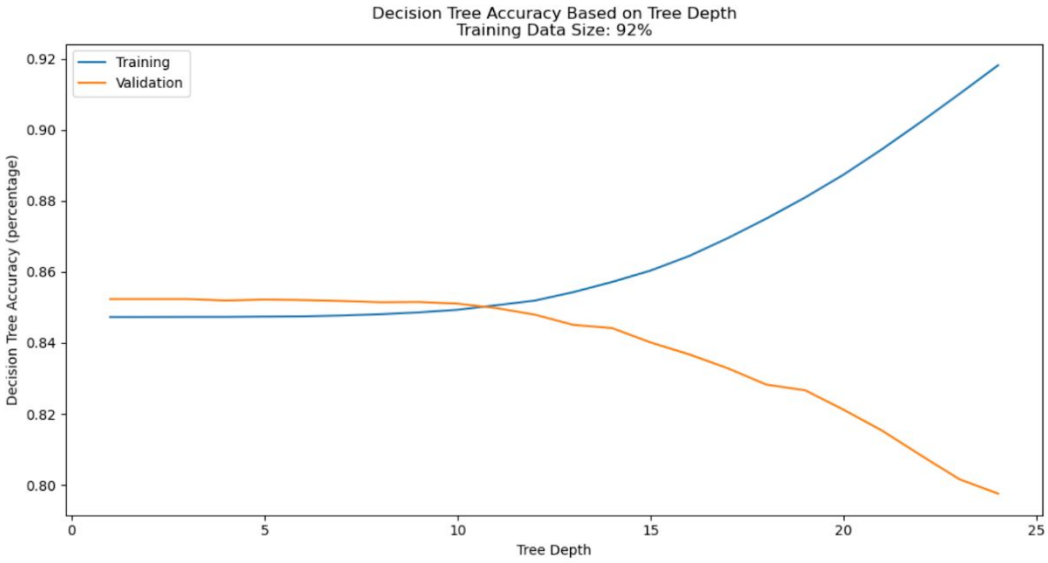


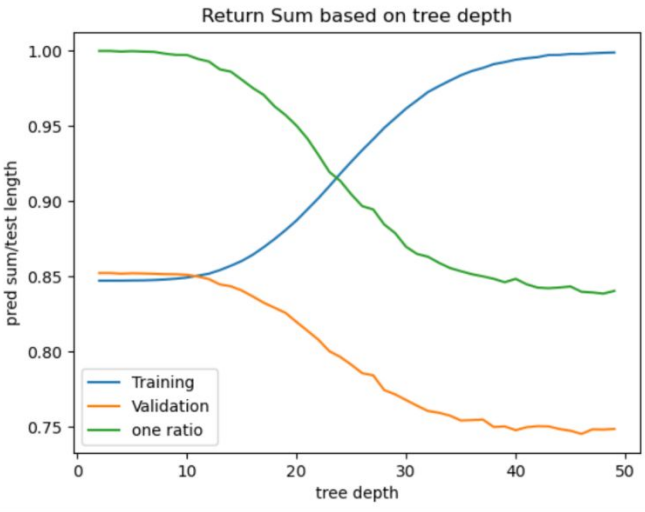
The graph above shows the decision tree accuracies based on the tree criterion And training data size. when selecting optimal values for decision trees the validation curves must be around a maximum and the tree should not be overfit. The validation accuracies Indicate the tree performance on the testing data, while the training accuracies provided the tree performance on the training data itself. training accuracies is usually indicative of the development of a

prediction model rather than its quality of performance. The validation accuracies are usually more indicative of the performance of the given tree.

When a tree is overfit, its parameters are two specific, obliging it to tightly adhere to the data not allowing for a predictive factor. If a decision tree is overfit, Its prediction qualities will be lacking as it will take outliers and other unique factors into consideration rather than general and crucial trends. On the other hand underfitting the data can give you bad prediction results as well because the tree is providing too few options for a given test sample to take. Decision trees will be too generalized and not have enough paths for a given test sample to adhere to and yield a correct prediction result.

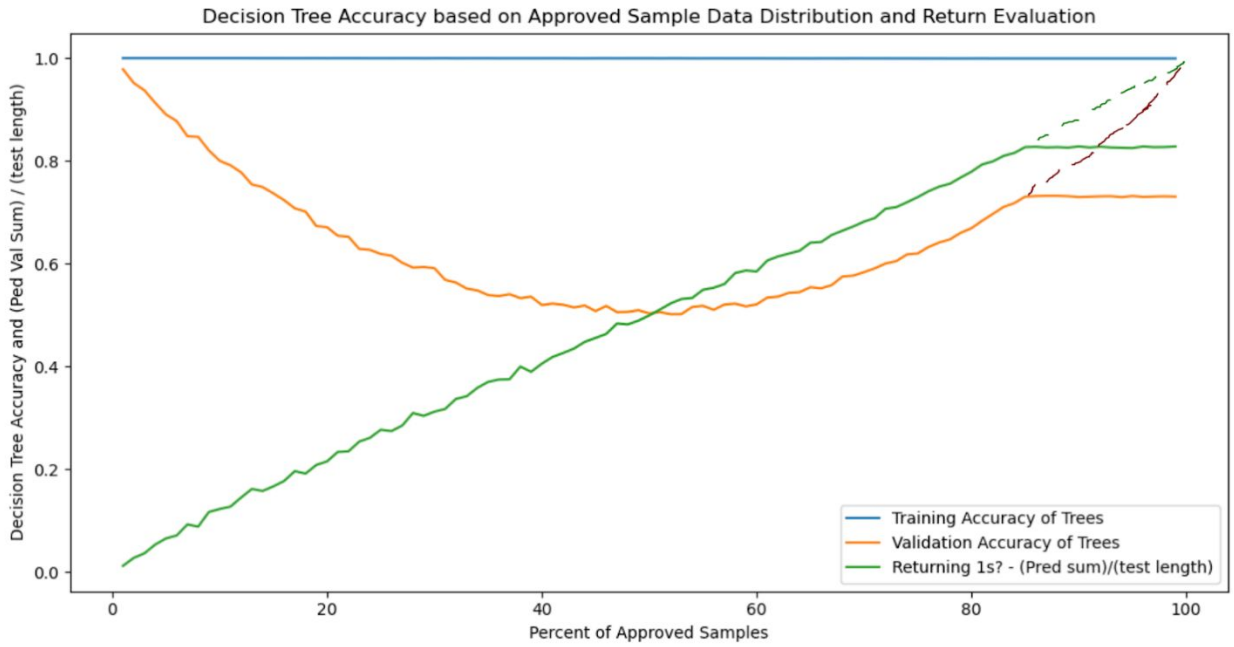
In the graph shown above the optimal value for the training data size is around 92%. This is clear as the validation accuracy seems to be at a maximum, implying that the tree performs best at this training data size. Also, it is clear that the entropy criterion seems to yield better performance than the gini criterion As one graph is positioned slightly higher than the other. The gini and entropy criterion in essence yield information about how the tree makes its nodes, at which it breaks into two separate branches. The entropy criterion pertains to the ordering of information as nodes progress, while the gini ratio pertains to the amount of information gained per node when approaching the final leaves.





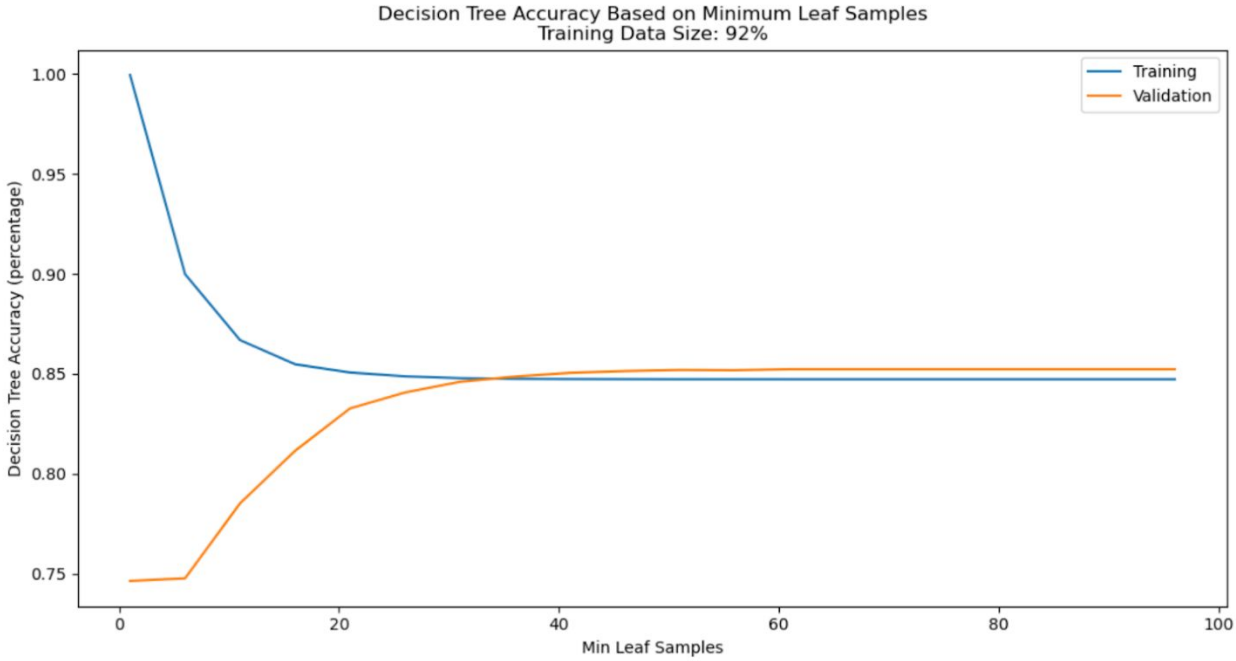
The graph above shows the relation of tree accuracy with the depth of the decision tree. well the maximum validation accuracy seems to be around 8, indicating an optimal value, this would later prove as inefficient and underfit. The graph shown to the left depicts this. The green line labeled the one ratio, shows the types of values that the decision tree is returning, which mainly

consists of zeros and ones since the project approval is a binary variable. The initial data set used to develop the decision tree is actually skewed as 85% of those variables or approved. By underfitting the data with a tree depth of eight, the tree would only return once proving for an 85% accuracy which at the time seemed very successful. After some experimentation and looking at the ones ratio graph it was concluded that the decision trees were under fit and that the high accuracies were only coming because of the skewed data set returning all ones and proving successful because 85% of those values were ones themselves (ones meaning approved). To provide more accurate decision trees, rather than trees that returned zeros and ones and didn't depend on a skewed data set, the tree must be overfit. Since 85% of the data was approved, it seemed appropriate that the tree returned predictions that were 85% ones and 15% zeros. This can be seen as the true depth increases to 50, the ones ratio tapers out to 85% indicating that the data is fit enough to return zeros and ones. although the accuracy is lower, this is the correct decision to make as the tree is no longer dependent on the skewed data set and makes decisions on its own. As a result of this graph and the realization of the effect of skewed data, a tree depth of 50 was ultimately chosen.



What further experimentation done on the distribution of data with the initially skewed data set, the above graph was derived to explain some of the tendencies shown. As the data distribution which was initially at 85%, increases to either 100 or 0, the validation accuracies peak at nearly perfect scores. This makes complete sense as the machine learning algorithm can only develop trends off of 1 specific variable which is either zero or one. with a single possibility, it makes sense that the accuracies would be at their peaks because the algorithm could only learn to spit out one variable and would therefore have very high accuracies as it would always be correct. However, when the data distribution is at 50% the algorithm has to use prediction and actually decides with the detected trends of weather the given test sample is either approved or not approved. since the algorithm is presented with both approved and non-approved samples equally there is an equal chance of predicting both as there are trends present for both

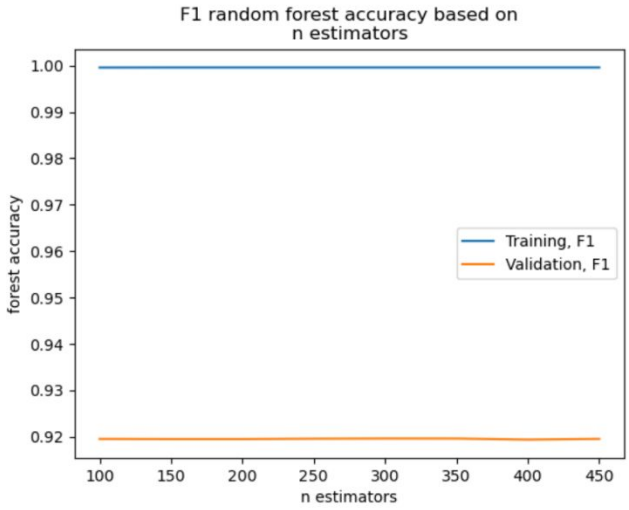
possibilities. Similarly the returning of 1's, or the predictions themselves adhere to this theory as the number of ones predicted tend to match up with the percent of approved samples does it correlate with how much the machine has learned of each possibility. if the algorithm is only presented with unapproved data, it will only learn to predict unapproved samples, and will therefore be 100% accurate; the same theory applies with an all-approved data set. However if a machine learning algorithm is presented with both approved and unapproved samples, it will have to decide for whether a given test sample is either approved or unapproved and this decision will be more difficult as there are trends present for both types of samples. this sort of behavior was accounted for by overfitting the data slightly as shown before so that the data will also return zeros along with ones to account for all types of data for an arbitrary test set.



The above graph shows the similar effect of minimum leaf sample size on the accuracy of the given decision tree. Since the tree depth factor was already regulated to control the fitting of the tree, the minimum leaf sample hyperparameter could either be set to 55, which is the optimal value according to the validation accuracies seen in the graph or it could just be such to the default value when building the decision tree in Python. Since the minimum leaf sample and tree depth hyperparameter are directly linked, regulating one would take care of the other in a way.

After developing decision trees, the final prediction model was developed. Random forests, in essence are groups of decision trees that are randomly created based off of randomly selected samples of data. The test sample is plugged into all of the decision trees and the most common outcome is the prediction presented. This is very useful as random forests account for the discrepancies that can be formed when developing a decision tree because one random decision tree might account for another run decision tree's discrepancies. The random selection of data allows for the detection and development of various trends of many kinds and allows for a more unbiased prediction model.

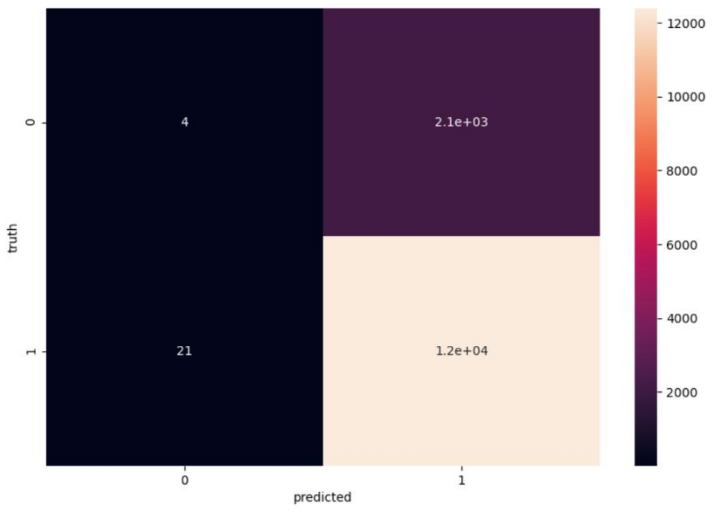
The random forest contains similar features of the decision tree, except it also contains features relating to the random selection of the variables used for the decision tree and the numbers of trees itself.
After the development of decision trees, the final prediction model, the random forest was developed. The same type of decision tree was used in the random forest, as that particular tree proved most effective in the previous analysis. the only factor that needed to be changed what is the number of trees used to take a majority decision For the final prediction. The graph below was used to determine the number of N estimators, but this was kept at the default 100 as it wasn't showing much effect for the decision tree accuracies. Although there were small discrepancies and fluctuations within the graph, they were negligible.



The final prediction model was a random forest prediction model that had a Max tree depth of 50, tree criterion of entropy, training data Size of 92%, and an N-estimator count of 100.

Performance

When it came to evaluate the performance of the prediction models, there were two methods used: A standard accuracy score and the F1 score. Most of the graphs throughout this report use standard accuracy scores, which just check for the percentage of correct predictions from the test data. The F1 score however uses the harmonic mean of precision and recall and is a much better indicator of accuracy in a predictive model.



The diagram shown above is a confusion matrix that shows various concepts utilized in the F1 score. The main sections are False Positive, False, Negatives, True Positives and True Negatives. False Positives are positive (1) predictions that are incorrect. False Negatives are negative predictions (0) that are incorrect. True Positives are positive (1) predictions that are correct. True Negatives are negative predictions (0) that are correct. All these values are taken in ratios describing both precision and recall and used to calculate the F1 score which all in all, is a more effective and accurate performance indicator.

The matrix above shows the final random forest used. This forest had an F1 score of 93% and standard accuracy of 86%. The decision tree that was finally developed had a F1 score of 91% and a standard accuracy of 76%.

Conclusions

In retrospect, the accuracies from the prediction models and relevance of variables would prove this project to be successful. The overall goal of this project was to be able to predict the approval of a project based on its text analysis oriented characteristics based on the essay prompts only. The accuracy and performance returned with this type of prediction model was comparable to other prediction models that used the original data set and more surface level variables. Therefore, this project, in a general sense would be considered successful.

However there are a few areas that could be expanded upon. Solving the skewed data set problem could be explored more as there are other alternatives in questions that could be figured out when trying to optimize and determine the true validity of the prediction model. Along with this, to increase the relevance of the text analysis features to the approval of a project, the Python algorithms used to code the new features could be altered to polarize the feature is even more allowing for a clearer distinction between approved and unapproved. polarizing the features even more could provide for a clearer decision tree node and a potential raise in accuracies.

As far as implementing this project into DonorsChoose.org, which is just a hypothetical situation as the competition is over, Would mean that the actual content of the project would not be taken into consideration. This project was for exploratory purposes, to see if evaluators took a bias to where it's more passionate or grammatically/sentimentally appealing essays. in reality, evaluators or a more realistic machine learning algorithm would have to take into account the actual characteristics of the project and the tangibles statistics rather than the emotions conveyed through the essay. although the trends determined from this algorithm are valid, they cannot be entirely used to determine the approval of a project.

GitHub Link:
https://github.com/sripulugurtha/NLTK-Sentiment-Analysis-with-Decision-Trees-And-Random-Forests
*All code written for the project can be found in the linked repository