Group 6 Contact Person- Jack Bressett

([bressett@wisc.edu](mailto:bressett@wisc.edu))

**STAT 601- Project 2**

**Final Report**

1. **Introduction and Refinement of the Question**

*Background*

The ever-expanding universe of online code almost guarantees that you will find something useful to you. Unfortunately, choosing a programming language for your project can be challenging when you are new to the field or have little industry experience. Depending on the seriousness of your project, there may be topics to consider like:

● The targeted platform

● The elasticity of a language

● The time to production

● The performance

● The support and community.

But often it seems that some languages will be chosen arbitrarily for projects when they may not be the best tool for the job. The authors are curious in understanding the decision-making process of the online data science community when it comes to choosing a particular language to develop in. Furthermore, is there a way to predict which language will be used if information like tags, libraries used and the description of the project from the GitHub repository are given? This research has the potential to aid beginners in the field by providing insights about practices in one of the world’s largest data science communities. This also provides insight to stakeholders in the programming language development community as it will demonstrate factors that lead community members to choose one language over another for data science projects.

The goal of this report is to conduct the fully structured data analysis, ranging from refining the question, performing an exploratory data analysis, establishing the statistical model as well as performing hypothesis testing for the variables relationship to interpreting the results which can eventually suggest the applications upon the results.

1. **Methodologies**

*Refining the question*

The broad question that we came up with is, "Do people prefer R over Python for different types of data science projects?", and our objective is to refine this question into a more specific question which possesses the characteristics of a good, refined question. After we discussed the importance and connection between the given question and our goal, the refined question is that

**“Is it possible to predict whether a GitHub Data Science repository will use the R programming language or the Python programming language from the information provided by the repository snippet?”**

We are particularly interested in the usage of R within the problems or projects which are currently present on GitHub under a “Data Science” search.

1. **Experimental Design and Reasoning**

The process of answering our refined question begins with data collection. GitHub is an obvious choice for data when discussing the opinions, preferences, and choices of a development community, so we will start collecting data there. Below is an example of a search result for “Data Science” under repositories:



In the image, we see that there is a snippet of information provided about each repository. Most importantly, the technology used within that repository is stated. In this example, we see Jupyter Notebook, Python, and CSS are listed for three different repositories. Our goal is to understand how the information provided in these snippets can be used to determine whether or not a repository will list R as a used technology. While some pieces of information will be quite trivial, for example, the middle repository lists python in the tags, other information will be more ambiguous such as tags labelled “big-data” or “tutorial”. We are more interested in these features and less interested in trivial features such as a repository with “Python” in the name like the middle repository.

To collect the information in the snippets programmatically, a web crawler was created in Python. This data was then processed and stored in a usable format for analysis. The initial data includes a sample of the nearly 260,000 data science repositories that are available. It should be noted that all filters such as “most stars” and “fewest stars” were used For each repository, a list of tags that were used, the number of favorites the repository received, the description and title of the repository, and the last date the repository was updated were all collected. The data requires a good deal of preprocessing before it can be passed into our model framework. Primarily, creating indicator variables for different tags and creating indicator variables for the presence of libraries of a similar language.

Ultimately, the data we are collecting will provide insight and hopefully describe the decisions that were made by members of the community to include R in their repositories or not.

1. **Data Preprocessing**

After we retrieved the data from the web using the web crawler, we had 7 data files initially and we combined all of them to a single .csv file. This gave us 7,000 repositories. For our response variable, we use the language which was used for the repository, we used one-hot encoding and assigned the value of 1 for those who used the language R and 0 for Python. We removed entries that were not in english and also removed repositories that did not explicitly list R or Python as the majority language. This left us with 685 repositories out of the 7,000 initially collected. (Most repositories listed Jupyter Notebook as the main language).

For the ‘tags’ variable, our processing methods is as follows:

* Cleaned the ‘tags’ variable in our data while gathering it from the web using the crawler.
  + Made lowercase, removed needless characters
* Replaced python libraries with “lib” and R packages with “package” to eliminate multicollinearity from having multiple packages named inside a tag list. To do this we assembled a list of nearly 300 python packages and 100 R packages.
* For optimization, we are using those tags which are being used more than 5 times in the whole data and created a binary classifier and thereby created around 170 dummy variables to indicate those words which are in the tags for a specific data point.

For the ‘description’ variable, our processing methods is as follows:

* For convenience and for better accuracy, we removed those data points from our data whose ‘description’ language was not in English.
* Cleaned the variable by first making all the words lowercase, removing the stop words, ‘http’ and other junk words which do not potentially contribute to our model.
* For further optimization and to create the optimized number of bag of words and dummy variables, we first created a list of the top 50 words in terms of frequency in the description of data points each for R and Python. Then we combined these two lists into one consisting of 100 words and deleted all the words from the ‘description’ of each of the data points which were not in this list and finally tokenized all the remaining words. This process created 100 dummy variables for words in the descriptions.

1. **Modelling Framework**

According to the chosen broad question, we can use a binomial logistic regression model to accurately answer and predict the programming language to be used. The main goal is to build a binary classifier for predicting if the creator of the repository at hand preferred to use R or Python. R is assigned to 1 and 0 is for using Python. We are specifically interested in finding out the relationship between the process tags as well as the description of the Github repositories with the probability of the usage of R or Python.

We will begin by determining the best performing features and try to eliminate the features which do not have a significant p value from the results of the model. The accuracy is another important metric that we would be trying to improve. We will also examine the deviance to more easily compare models. Additionally, we would then compute the confusion matrix along with the precision, recall, and support metrics to validate our results and help in evaluation of the model’s performance. The Receiver Operating characteristic (ROC) curve would give us a good idea of how good our classifier is performing and would help us to understand the model appropriately.

1. **Results**

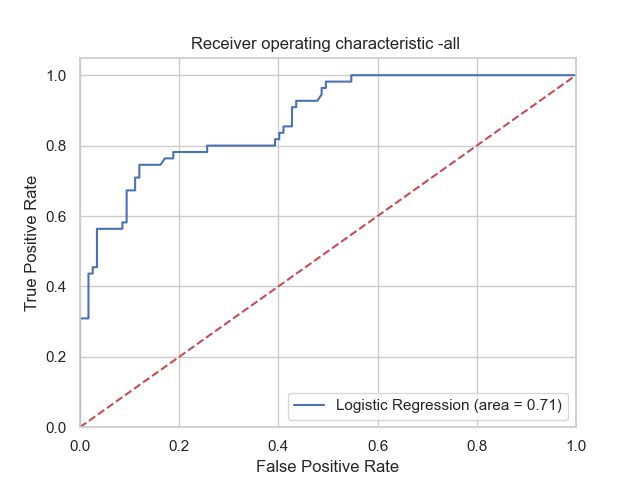
We have divided our modelling process into two schools of thought: light analysis and heavy analysis. This refers to the amount of predictors that are used within the model. Beginning first with the light strategy:

*Light Modelling*

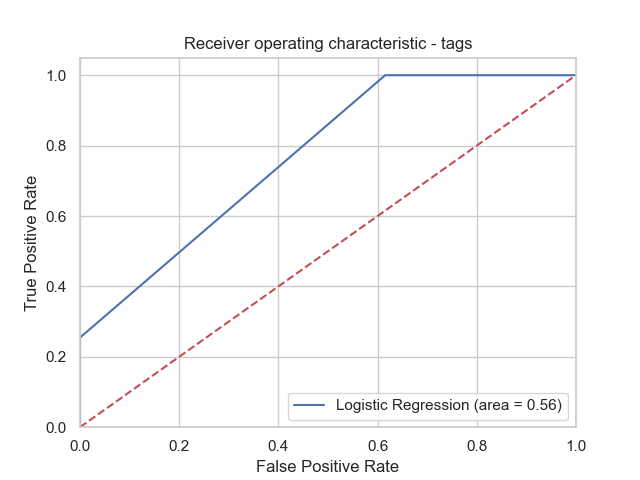
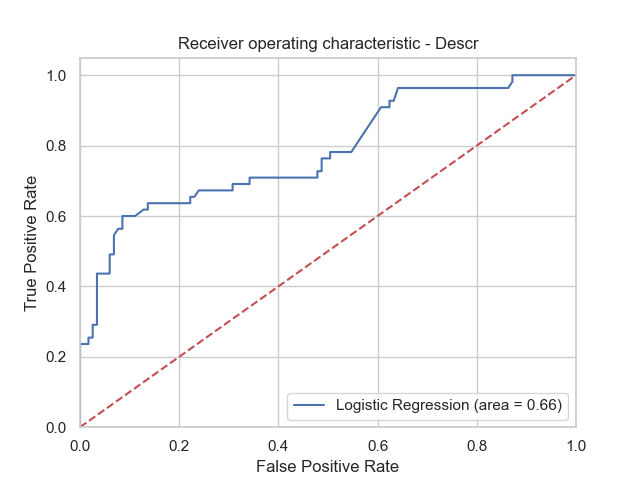
The simplest model is based on the majority class probability which is 75.9%. This model has a deviance of 756.36. We consider this the baseline metric that we compare our more complicated statistical models to. Next, we tried models using the predictors: number of stars, number of tags, number of words in the description, and type of license. Log transforms of the number of tags, words, and stars were tried in this process as well. The best model to come out of this section included all of the variables (without log transforms) and interestingly, type of license accounted for four significant predictors out of 16. The intercept was also significant and had the lowest p-value. The most significant licenses were the MIT License which caused a decrease in the odds of being R by a factor of .65. Another very significant license is the Apache-2.0 license which caused a decrease in the odds of being R by a factor of 0.97. The best model’s deviance in light modelling was 666.45 which is much smaller than the baseline of 756.36.

*Heavy Modelling*

The heavy model is based on when we consider all the predictors: number of stars, corresponding tag indicators and the word description indicators via the bag of words model with the total number of predictors being 156. The best that we came up with was when we included all the above mentioned variables with an accuracy of 83% followed by 77% when we only considered the description indicator and 72% when we considered only tag indicators as variables. A receiver operating characteristic curve, or ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. In the ROC curves below, the area under the curve is highest for the model where we have used all the features, this also has best performing accuracy.



**Best ROC Curve**



**ROC Curve for Only Description Model ROC Curve for Only Tags Model**

1. **Conclusion**

In conclusion, we were able to find a somewhat sufficient answer to the predictability of R in a data science GitHub repository. We produced several models that performed better than a “blind guessing” strategy and found some interesting relationships between the licenses and the probability of a repository using R. We also found that when we used variables such as number of stars, corresponding tags indicators and the description indicators, we were able to come up with the best model having accuracy of 83% with precision at 89%, recall at 97% and F1 score of 87%.

Hence, we have successfully shown that it is possible to predict whether a GitHub Data Science repository will use the R programming language or the Python programming language from the information provided by the repository snippet.

1. **References**

[**https://github.com/BressettJ21/GitHub-Language-Prediction**](https://github.com/BressettJ21/GitHub-Language-Prediction) **- GitHub Repository**