Kate Li, Joshua McEnroe, Weixing Tang

CS378

Final Project Report

Motivation and Contributions:

For the final project, our group entered the DrivenData ‘Pump it Up’ competition which posed a classification problem of predicting whether water pumps for Tanzanian wells are functional, non-functional, or needing repair. The data was collected from the Tanzanian ministry of Water and an in-development API, Taarifa, which crowdsources maintenance information reporting. Progress towards this classification problem could help the triage of repair and management resources for broken pumps, improving the percentage of population that has access to clean water and the associated public health benefits.

Our goals at the projects onset were to achieve a classification score of above 0.70 and to have a net improved score over multiple submissions. Having surpassed those goals, our later goals centered around fine tuning our results with ensemble methods and alternate data cleaning approaches in an attempt to close the gap to the top score at 0.8285.

Joshua McEnroe worked on cleaning the data, producing three different cleaned data sets. Kate Li worked on classifying the data with a Naive Bayesian classifier and an ID4 decision tree. Weixing Tang worked on ensemble voting between adaboost, decision tree, and a random forest, and voting between multiple runs of random forest. While these are the individual focuses, the group met and coordinated frequently, sharing datasets, code, and discussing what steps were to be taken or approaches tried.

Related Work and Methods:

Our methods differ from those outlined in our project proposal due to feedback clarifying that we could use outside packages and implementations. We worked largely with classification methods included in scikit. We started with the simpler existing methods for id4 decision tree and the naive bayesian classifier, adaboost, and random forest.

For preprocessing the data, existing methods and our coursework background suggested deletion of tuples with missing data, imputing missing values with a new ‘missing’ class, and imputing based on the mean or mode value for an attribute.

Approach and Methodology:

The dataset consists of 70,000 records with 39 attributes, mostly categorical. In the interest of visualization and narrowing the number of attributes, we tried using K-Means clustering method to cluster the dataset into several sub dataset first to see if the dataset can be roughly determined by few attributes, so that we can run classification on each roughly grouped dataset. Overall clustering gave a very bad result: 0.2 for k=10. However, while comparing the visualized data with the map of Tanzania, we found that the conditions of the water pumps are slightly related to the distribution of the water source. (Figure 1)

The first cleaned dataset we used removed five attributes: wpt\_name (redundant by id), subvillage (redundant by other geographic attributes, data has too many levels), amount\_tsh (70% of values were missing), recorded\_by (only one possible value), region\_code (made redundant by region), num\_private (almost all values were 0), funder (too many levels). Missing values were then imputed with the mean value for numerical attributes and the mode value for categorical attributes. All values were then converted to numeric values for usage in classifying methods.

This first data set was used with an id4 decision tree, and adaboost, implemented through scikit.

A second cleaned dataset removed the same attributes as the first, but imputed missing values with an ‘unknown’ class instead of with mean/mode. This dataset yielded a better result with a decision tree and was used for the next group of methods: naive bayesian classifier and a random forest.

We then tried assemble methods between the simpler methods we had already implemented. Due to its very poor performance, we excluded the naive bayesian classifier and used a three-way ensemble vote between a decision tree, adaboost, and a random forest. Our results indicated that adaboost might be overfitting and that a lone random forest outperformed the ensemble voting. We then wrote an ensemble method to vote between multiple runs of random forest, first ten and then one-hundred.

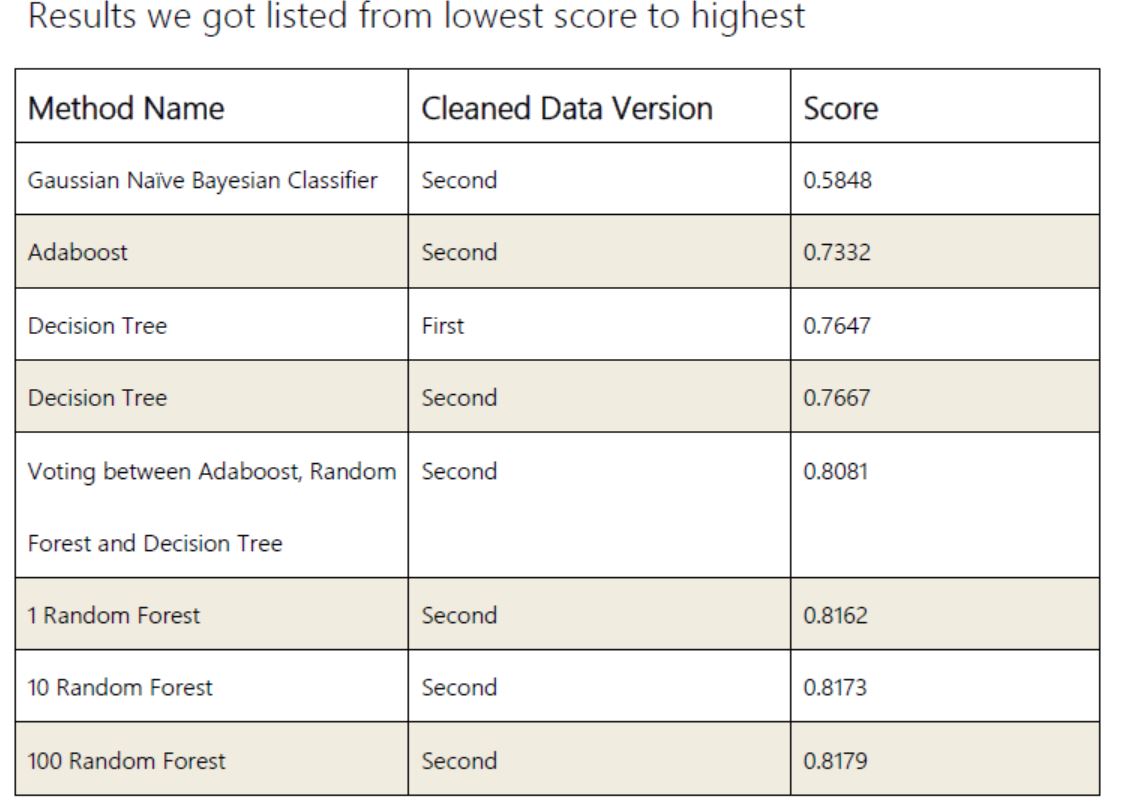
A third cleaned dataset dropped rows with missing values, shrinking the data set by ~10,000 records (15%).

A run of 100 random forests using the third dataset resulted in a classification rate of 0.40 according to the contest submission. Testing the classifier on a random sample of 1000 of the training data records yields 0.84. We think this might be a result of the data becoming corrupted with labels no longer matching their corresponding record for the training data. Another possibility is that removing records with missing data skewed the data by removing records vital to the classification of the lesser labels (i.e. pumps that are non-functional are more likely to have missing values).

Evaluation and Results:

The result was calculated upon submission of test set classification labels to the drivendata’s competition page according to the classification rate metric:

Classification Rate =

The metric calculates the percentage of rows where the predicted class in the submission matches the actual class, y in the test set. The maximum is 1 and the minimum is 0. The goal is to maximize the classification rate. 

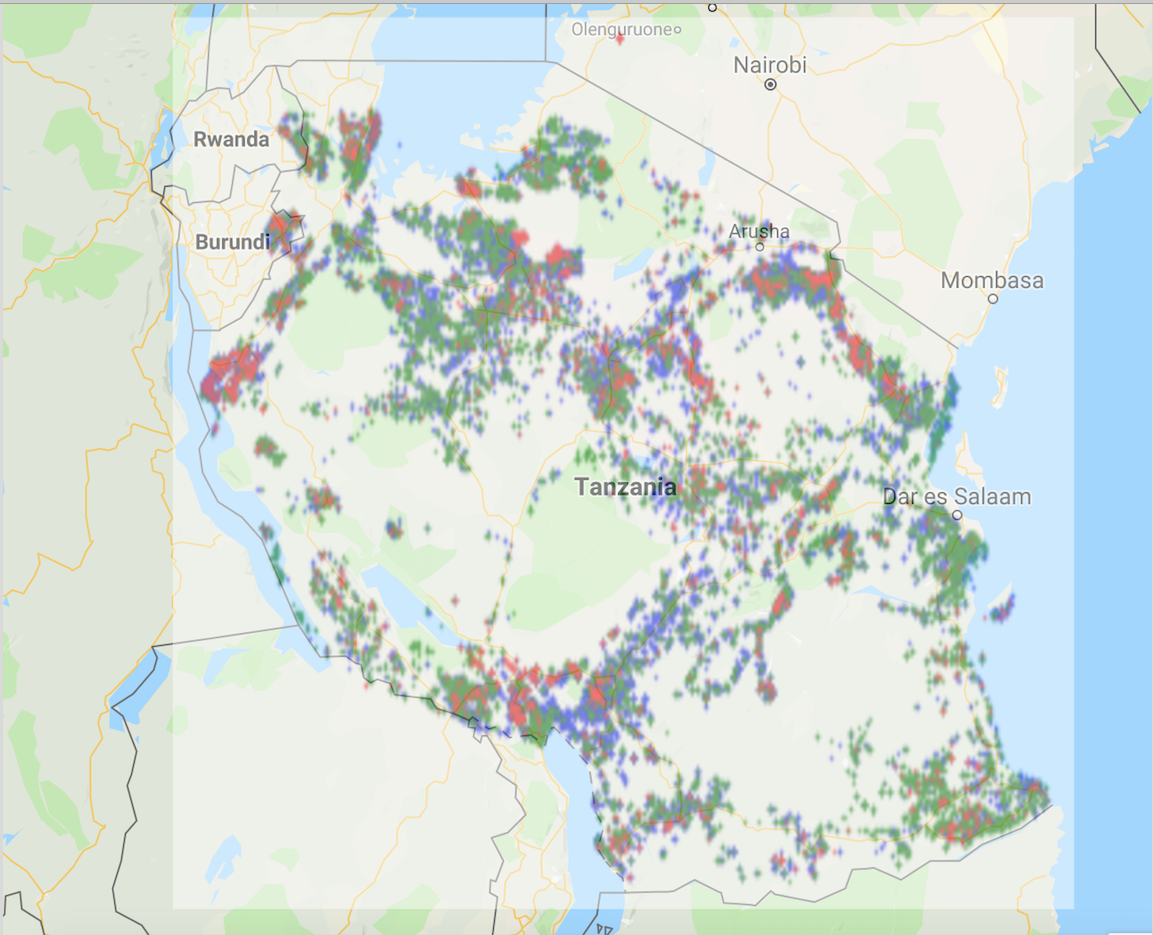
Conclusion and Future Work:

In the course of our work on the project we were able to utilize many of the algorithms we learned and covered in detail in class: k-means clustering, id4 decision tree, naive bayesian classifier. We also used methods that were discussed but not implemented, such as adaboost, random forest, and ensemble voting. We found that ensemble voting methods could offer improvements over the simpler methods, but that if there are only a few members voting then overfitting poses a larger problem, as was the case with adaboost voting against a decision tree and random forest.

We achieved our best results with multiple runs of random forests voting against each other, but returns were diminishing as we increased the number of runs. The diminishing returns from our best classifying method helped to highlight the importance and potential improvements of preprocessing the data.

We learned how drastically the selection of an classifying method or a cleaning method could alter the results, with the naive bayesian classifier performing well below all other methods, and the preprocessing removal of records with missing values performing only .07 better than random chance.

As the data stems from an open contest, the leaderboard shows that there is still possibile room for improvement. Future improvements could work on further datacleaning, re-evaluating which attributes are interesting and which are too heavily overlapping. Switching out the random forest implementation we used for H2O random forests might also yield improvements. While the contest has been open for three years now, with the top score from 2016, suggesting a limit to how much knowledge can be discovered from this particular dataset.



**Figure 1**: random sample of 40k data points by gps longitude and latitude attributes. Green is nonfunctional, red is needing repair, blue is functional