# Project Milestone

John Randis

11/7/2016

## Abstract

This paper explores the early stages of an application project intended to classify different species of leaves using different classification methods. The classification methods will be compared to see what works the best on this data set.

## Introduction

The implications and benefits of automatic plant classification are wide and far reaching. Human error often leads to misclassified or duplicate classifications. Automated classification can help discoveries in biology and medicine. This paper will include the background and related works, the methodology of the approach taken for this project, and a careful analysis of results.

## Background

In very recent history, biological information has never been more readily available. Databases exists across the web full of biological information on plant life. Online biodiversity databases such as http://www.gbif.org/ and http://www.pfaf.org/ have made this information more accessible than ever. Biologists have been classifying plant and animal species for years. This is why introducing machine learning to this problem could be so revolutionary. The problem of plant classification has not been tackled much in the data science community. On Kaggle, many submissions involve the KNeighbors classifier or the DecisionTreeClassifier of sklearn. These built-in classifiers are nice, but accuracy is often not very good and leaves room for improvement.

## Methodology

The approach being used for my classifier will be through logistic regression. Logistic regression is a simpler approach to this problem and is less prone to overfitting. The python libraries pandas and sklearn are being implemented to complete this task. Fortunately Kaggle provides a csv of pre-extracted features from the leaf images. The extracted features contain 64 attribute vectors for margin, shape, and texture. The classifier stores all these features in a pandas data frame, and builds a logistic regression model based off of the training data. Then, the model is applied to the test data. Each leaf is given a probability 0-1 for belonging to a specific species. It is outputted to the .csv submission file.

## Experiments

1584 pieces of data were used in this experiment. They are divided up into 991 data points used for training and 593 used for testing. They all corresponded to a different image of a leaf. The classifier checks for features like margin, shape, and texture. The data is put into a pandas data frame, is scaled, regularized, and then the sklearn logistic regression classifier is run. The support vector machine is given the species index (x_train) and its attributes (y_train). They are given to the classifier to accurately create a model without overfitting. We call the predict_probability method on the classifier now to run the test data and get percentage values for each of the plant ids and their possible species. The code compiled in under 10 seconds. The attached csv submission file includes all of the results.

## Analysis

Looking at the results of the experiment, it seems as though logistic regression is a reliable method for classification of this kind. Going forward with this project, I believe I can try different classifier methods such as Nearest Neighbors and Decision Tree classifiers. In addition, I plan to use neural networks using Keras. It will be useful to run different methods of classifications on this data set to see what ends up being the quickest, the most accurate, as well as judge them by other metrics. As of this milestone, I have only run one experiment so I cannot make any comparisons. However soon I will run other algorithms in order to compare them to logistic regression. There will be information on this in the final version.

## Conclusion

Using logistic regression, the classifier provided an accurate way of sorting the plant species. Online databases today make accessing biological data easier than ever. Automating the organization of this data is an important step we can take to improve the free flow of information.

## References

benjo. Sept. 2016. *Logistic Regression*. Python 3.4. Apahace 2.0 Open Source License. Source Code. *Kaggle*.

"Global Biodiversity Information Facility." *Free and Open Access to Biodiversity Data*. GBIF, n.d. Web. 07 Nov. 2016.

"PFAF." *PFAF*. Plants For A Future, n.d. Web. 07 Nov. 2016.