CAPSTONE PROJECT I: Milestone Report

April, 2020

Erik Larsen
Data Science Career Track
January 6th Cohort

Overview

One problem commonly found in both agriculture and urban farming is the identification of seeds. From sorting small garden seeds when they get mixed up, to quality assurance in packaging correct seeds for sale, and even seed type and quality on large farms, the ability to discern a species of seed when its origin is unknown can be vital. This project uses machine learning to identify seeds from geometric properties alone. Future extensions can lead to plant vs weed identification and better quality crops.

Goal

Explore and evaluate various algorithms that estimate the probability of a seed belonging to a certain species.

Approach

The Seeds dataset is relatively small compared to most datasets in use today. It is comprised of only 210 examples, with three classes of wheat grain. In this project we will experiment with various machine learning algorithms (e.g.,neural networks) and compare their relative performance with respect to metrics to be defined. Cross-validation and

bias/variance analyses will be performed to maximize accuracy and performance. The Seeds dataset can be found in the UCI Machine Learning Repository at seeds Data Set.

Description of the Data

Classes: Kama, Rosa, Canadian with 70 examples for each.

Attributes:

> Area, perimeter, compactness, length, width, asymmetry-coefficient, and groove length

The compactness is derived from $C = 4\pi \frac{A}{P^2}$ and the origin of the asymmetry coefficient is unexplained.

Data Wrangling Steps

It is simple to download and has a description of the parameters. Upon inspection it is clear that compared to most data sets in use today, this set is very small. There are 210 data points and 8 variables. Seven of these variables are numeric and essentially continuous; they are geometric properties of the seeds taken from precise measurement. The final variable is categorical, and gives the species of the seed with three distinct values: 1="Kama", 2="Rosa", 3="Canadian". The size of the dataset might be one of the challenges to be dealt with. A number of ways to address this have been considered, including cross-validation, bootstrapping, and creating synthetic data.

There are no missing values. There is, however, a mismatch of data in correct columns. After inspecting the original text file, the mismatched columns were easy to see. Some of the entries had, in random places, been shifted over to the right a column or two, leaving blank spaces where the information should be. This misalignment was present in about 5% of the row entries, and caused an error when attempting to load the file directly into a pandas dataframe. Skipping these rows, the file can be read into a dataframe with 11 missing entries.

An alert giving the rows seen lets us open the file and write in an initial row that is beyond the greatest column of misalignment. Doing this allows all of the data to be read into the file without skipping any rows. The consequence is that there are now extra columns that are full of NaN values. The actual corrupted data can be identified, stored for cleaning, and removed from the rest of the properly aligned dataframe.

To align the corrupted data, read the correct values into a list in the correct order, leaving out the NaN entries causing the misalignment. A function herein does this and returns a new dataframe with the data properly aligned and indexed. This is then appended to the uncorrupted dataframe, and sorted back in the proper order. The result is a tidy dataframe with no missing values and everything properly aligned. The data is ready for analysis in a dataframe called seeds, and saved as seeds.csv for further analysis.

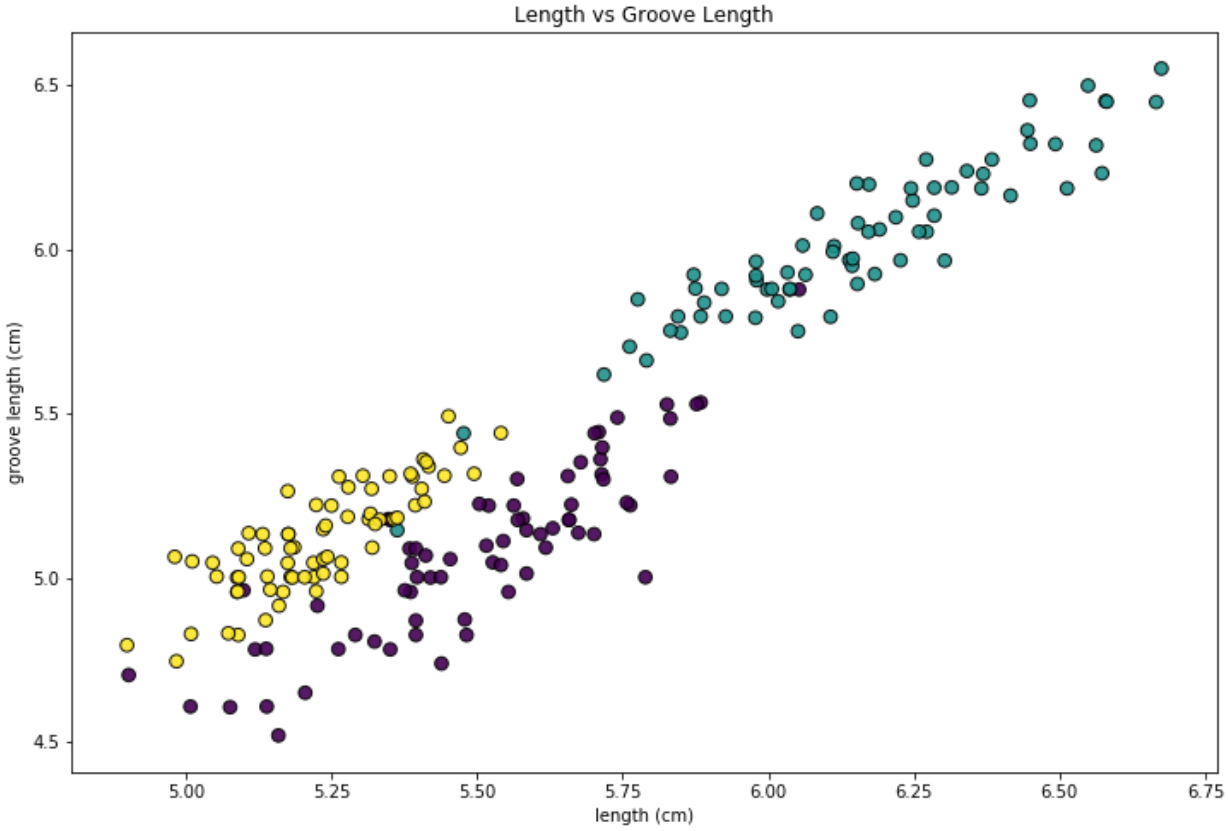
Storytelling: Questions Asked

What are the summary statistics of each feature for each class?

	area	perimeter	compactness	length	width	asymmetry_coefficient	groove_length	class
count	210.000000	210.000000	210.000000	210.000000	210.000000	210.000000	210.000000	210.000000
mean	14.847524	14.559286	0.870999	5.628533	3.258605	3.700201	5.408071	2.000000
std	2.909699	1.305959	0.023629	0.443063	0.377714	1.503557	0.491480	0.818448
min	10.590000	12.410000	0.808100	4.899000	2.630000	0.765100	4.519000	1.000000
25%	12.270000	13.450000	0.856900	5.262250	2.944000	2.561500	5.045000	1.000000
50%	14.355000	14.320000	0.873450	5.523500	3.237000	3.599000	5.223000	2.000000
75%	17.305000	15.715000	0.887775	5.979750	3.561750	4.768750	5.877000	3.000000
max	21.180000	17.250000	0.918300	6.675000	4.033000	8.456000	6.550000	3.000000

Each class has different summary statistics for its features. Consequently, each feature is distributed differently for each class. These distributions are in general not normal, but some of them come close to fitting a normal curve.

Is there any clustering present when comparing different features?



Many of the features' scatter plots show excellent linear relationships. The length v groove length plot shows good cluster separation, as well as a linear trend in each class. The derived features, i.e. asymmetry coefficient and compactness, show no clear trend. They tend to appear more nebulous in relation to the other variables, which may not be of surprise because they are derived from them.

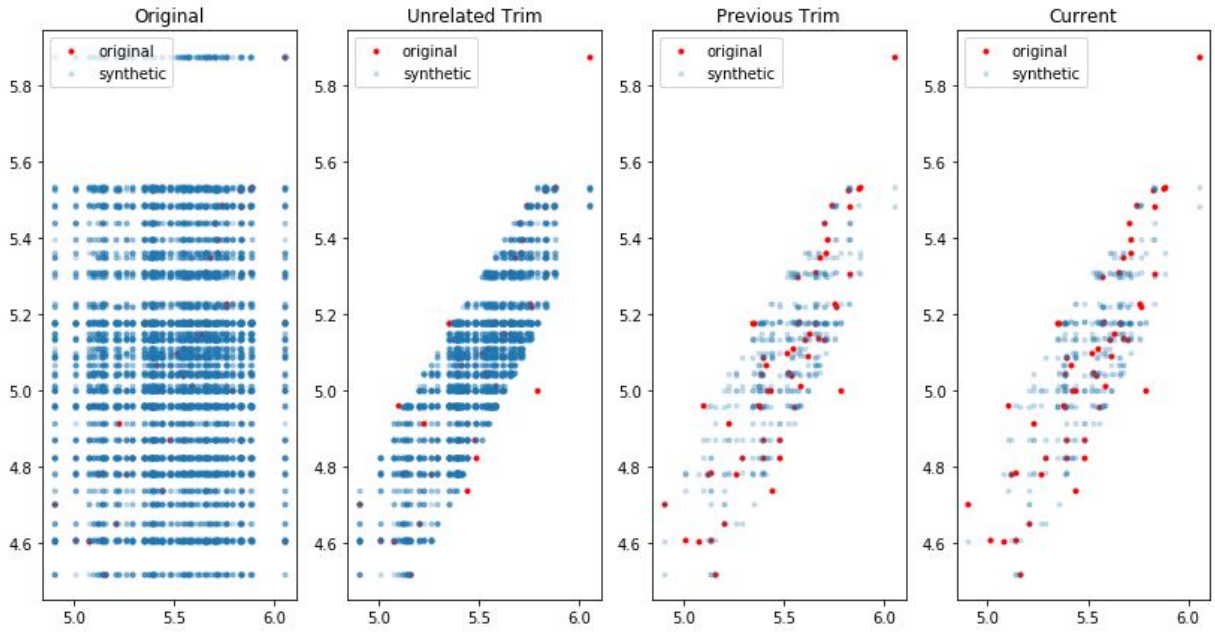
Are there any outliers?

There are five points that could be possible misclassifications. They lie deep within other classes' clusters, and were often misclassified in a trial K-Nearest Neighbors approach. Without knowing if they are truly classified correctly, the best option is to proceed with the belief that they are correct. Cross-validation folds will group these with various ensembles of others to train the model, and this will help reduce the error caused by the outliers.

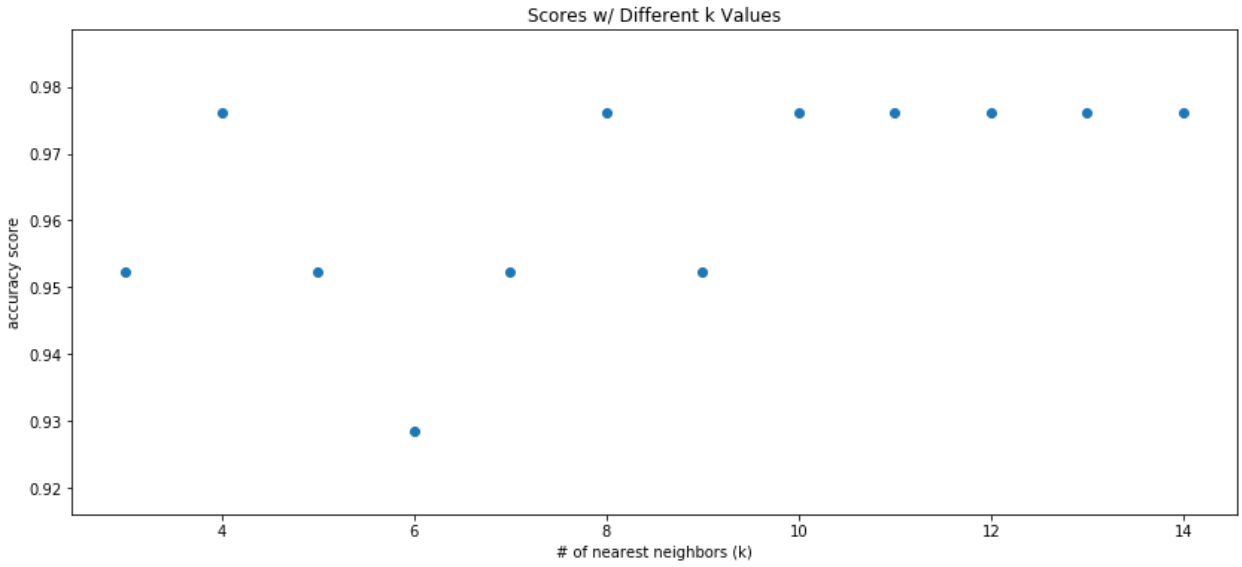
How do we increase the size of the dataset?

One interesting method is to use the distributions of each of the geometric features for each species. Sampling the data in a bootstrap fashion creates many synthetic points. Fitting a regression line to the correlated features and trimming randomly generated points that lie outside of a specific distance from this line. In this way, synthetic points are generated for that pairing of features.

However, this often creates more erroneous points that lie outside of the specific range of the fit line for two different features. Each pairing of features must be fitted and trimmed. With each successive trimming, the validity of remaining points increases until finally all new 'valid' data points lie close to the regression line for all pairings of features. Below are the results of this method for one class, Kama.



A k-Nearest Neighbors approach yields anywhere from 93% to 97.8% accuracy when only the training data is used with no cross validation. When the bootstrap sample is used, the accuracy reaches as high as 100% on the test set. While not the focus of this approach, the k-NN success bodes well for the performance of other models accomplishing the same task.



Applications of Inferential Statistics

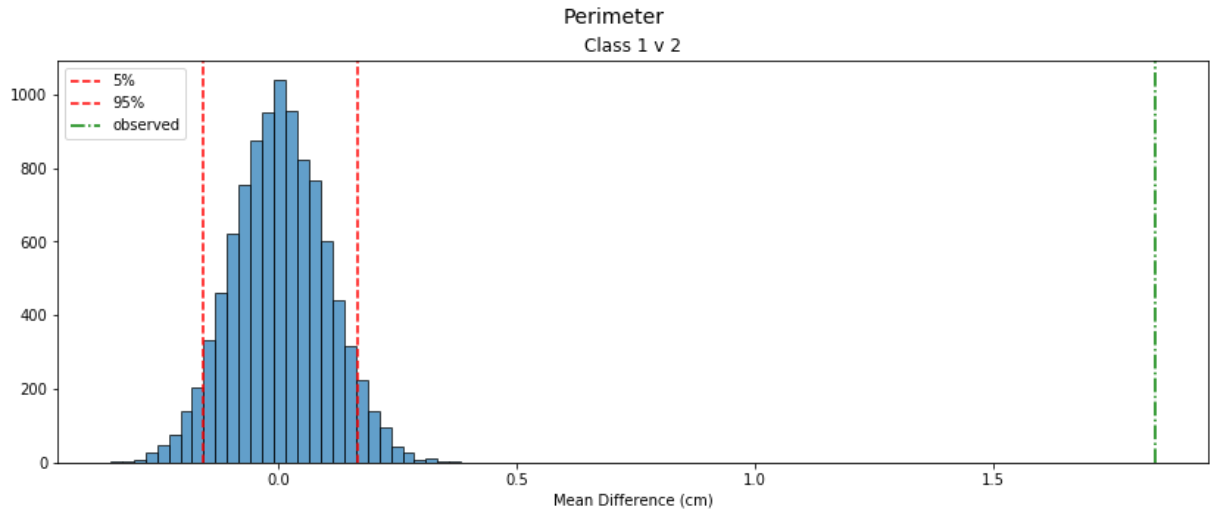
Two methods, bootstrapping and the Pearson R coefficient with its corresponding p value were explored. With such a small sample, the bootstrapping approach could be used in many ways to simulate distributions and compare a variety of statistics. Here, because there is some overlap and a few possible misclassifications, the means of each class feature have been compared to the means of the other class features.

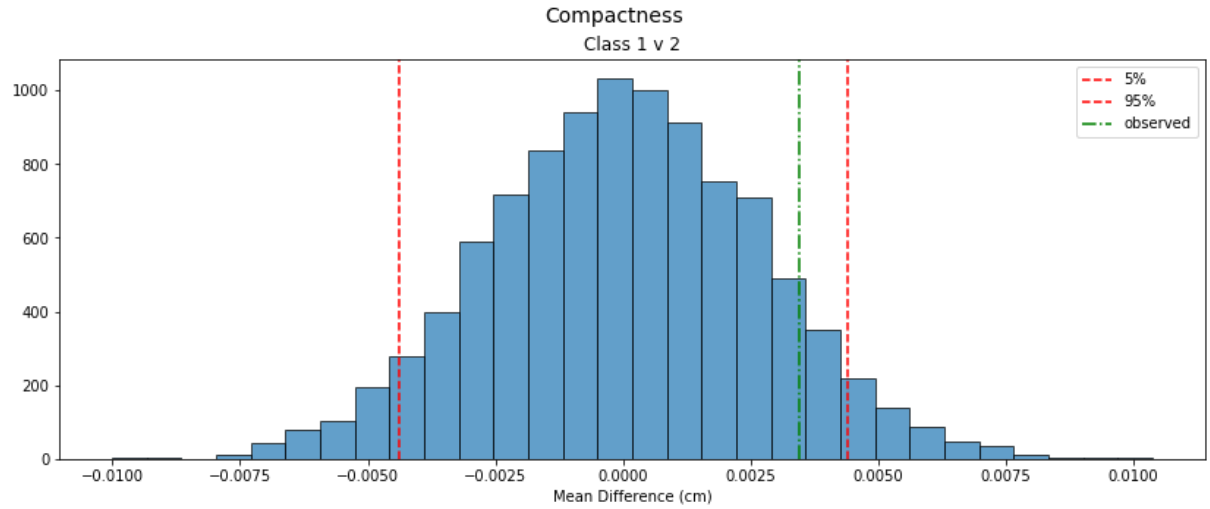
The purpose is to determine if the observed differences in the seeds dataset are representative of the differences in the world's seed population, or if they were a random anomaly due in some part to the sample size. Most of the comparisons showed that the observed value would be very far out of any expected values seen in the simulated distributions if the seed class features compared had the same means.

The null hypothesis was that the overall population of seeds had the same mean value for all of their geometric features, and the observed difference was due to random chance predicated by the sample size.

The alternative hypothesis was that the overall population of seeds doesn't have the same mean for their geometric properties, and that the observed differences are representative of the overall population.

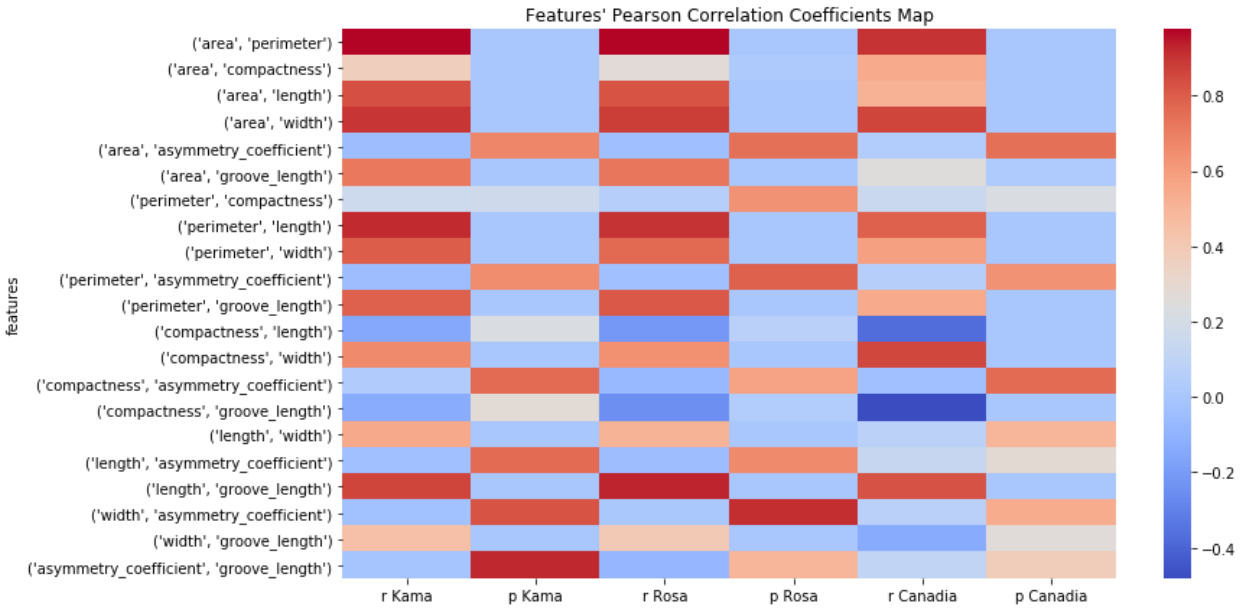
Only the derived features failed to reject this null hypothesis. However, this is trivial because of the synthetic nature of these features: they were not "observed."





A heatmap of the Pearson R coefficients and p values gives a quick indication of the feature comparisons that could be statistically of interest. Surprisingly, many of the derived features showed a good deal of statistical significance when compared to other geometric features.

Next, a list of the calculated p values is created, separating statistically insignificant comparisons by the arbitrary threshold of alpha = 0.5; performance of different thresholds on prediction accuracy will also be evaluated. The distillation of the feature comparisons will help us determine which features are superfluous and, hence, can be left out of the model.



Next Steps

First, a logistic regression model will be built, testing the various combinations of hyperparameters and included features. The analysis and results will be stored in a Jupyter Notebook.

Second, additions to the logistic regression model will be considered. Other approaches to classifying the data will be explored, including neural networks.

Finally, extensions to the model will be applied and evaluated, and results will be recorded in the Capstone Project I Final Report.