SPRINGBOARD

CAPSTONE PROJECT I: In-depth Analysis

April, 2020

Erik Larsen

Data Science Career Track

January 6th Cohort

Overview

Several machine learning methods are used in this project, including K-Nearest Neighbors, Logistic Regression, Naive Bayes, and dense neural networks. The performance of each model was evaluated and the most successful chosen as the preferred model for identification of seeds from geometric properties.

Goal

Compare performance of different machine learning algorithms and report on the most successful model. The highest accuracy score on test data will be used to evaluate the comparisons.

Approach

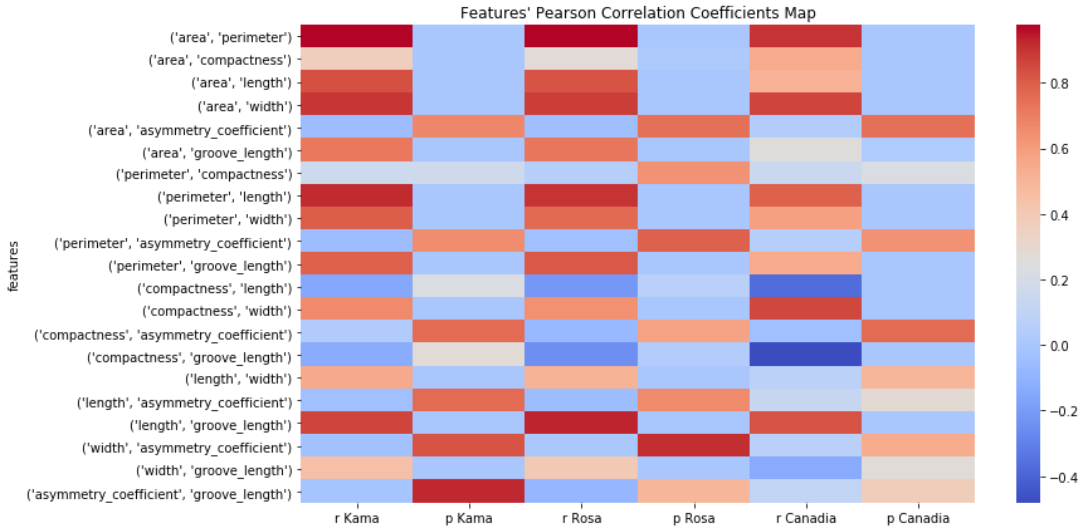
The seeds Data Set is relatively small, making it possible to obtain results quickly on multiple models seen throughout the course. Below are the approaches taken and their results. Three models of each type mentioned above were trained, and their relative utility measured simply by their raw performance within and against respective models. Feature selection was done using the guidance of the correlation coefficients and their respective p-values.

Graphical and Statistical Analysis

Seaborn's pairplot function showed a clear correlation between many of the features. The statistical significance of these correlations was tested with the hypothesis that the seeds all have the same feature means in nature, making the null hypothesis that they do not all have the same feature means in nature. Hence, being unable to reject the null hypothesis (in most cases), the observed differences in the geometric properties between the species of the seeds is highly unlikely to be due to random chance.

features	r Kama	p Kama	r Rosa	p Rosa	r Canadia	p Canadia
(area, perimeter)	0.976437	5.131220e-47	0.975806	1.246915e-46	0.907601	2.478192e-27
(area, compactness)	0.371037	1.566252e-03	0.272633	2.240792e-02	0.546760	9.734266e-07
(area, length)	0.834778	2.728968e-19	0.826427	1.261563e-18	0.516603	4.690915e-06
(area, width)	0.900066	3.138595e-26	0.880493	9.866088e-24	0.863824	6.290935e-22
(area, asymmetry_coefficient)	-0.050482	6.781258e-01	-0.039503	7.454211e-01	0.039612	7.447464e-01

The table above shows the Pearson Correlation Coefficient and corresponding p-value for a few of the features when compared with 'area'. The correlations with the most significance have small coefficient values. As expected, many of the highly correlated features show very low significance.



The heatmap gives a quick visual reference to which features are most highly correlated and whether or not this is significant.

Logistic Regression

A LogisticRegression() classifier was used, set with 5000 iterations allowed to achieve convergence. The best solver proved to be 'bfgs'. Hyperparameter tuning was accomplished by testing many pairings and choosing the best combination. The regularization parameter was found to be C=37.5, while the penalty score was optimized by an elastic-net with a mixture of .3. The final, optimized LR classifier scored 98% on the training data, and 95% on the test data. The classification report showed excellent performance as well, enough to consider data leakage. Removing the highly correlated feature 'compactness' from the data led to the same scoring results but with a more believable precision and recall in the classification report.

Naive Bayes

The data is continuous in each feature, prompting the choice of the GaussianNB classifier. This model's performance was lacking compared to logistic regression. The Gaussian Naive Bayes classifier performs with results that suggest possible overfitting. The discrepancy between training and test scores seen in the Logistic Regression model suggests this as well, but on a greater scale in this model. The GaussianNB scored 93% on the training data, and only 86% on the test data. The classification report shows better performance in precision than in recall.

K-Nearest Neighbors

The exceptional clustering seen in the scatterplots begs for exploration of a nearest neighbors model. This was done for fun early on in the project to assess future model success possibility. It's accuracy was remarkably high without cross validation or hyperparameter tuning, yielding almost 98%. The presence of 3 to 5 possible outliers or misclassifications are contributing to the missed predictions. The degree of overlap of certain features contributes as well, but not to near such a degree; the majority of neighbors will be in-class due to the natural clustering. This model could possibly give better results with more exploration.

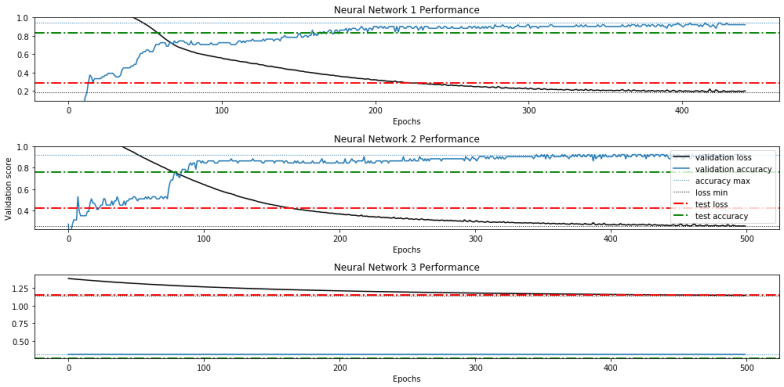
There are a few possible misclassifications. They lie deep within other classes' clusters, and were often misclassified in the trial K-Nearest Neighbors approach. Without knowing if they are truly classified correctly, the best option is to proceed with the belief that they are correct. Cross-validation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

K-Means Clustering

While the classes are labeled for us and this can be used for supervised learning, many unsupervised techniques were explored as well for posterity. The first is K-means clustering with 5000 maximum iterations. The K-means classifier does not perform as well as logistic regression or even naive Bayes models. Even with highly correlated features removed, the model performs poorly, with scores as low as 5% and as high as 33%. This is likely due to the limited data size paired with a train_test_split that may include the outliers in the test set. It would be very difficult to correctly classify seeds whose geometric profiles overlap via means of those overlapping properties. The supervised k-nearest-neighbors model provides better results.

Dense Neural Networks

Finally, the goal of the project is explored. Using Keras, a Sequential() model was constructed. Several initial trials showed varying levels of success across different feature selections. A chosen few are shown below, each corresponding to different highly correlated features being removed. Removing ‘compactness’ appeared to help the model distinguish seeds correctly. The formula for compactness is derived from other more fundamental measurements (e.g. length), and the feature likely just served as noise.



Three dense neural networks were trained. The number of nodes throughout the layers was set to be the square of the length of the number of training set columns. They were allowed to train for 1000 epochs, with a stop patience of 25. It’s interesting to note that the maximum number of epochs was never reached. The ‘adam’ optimizer, along with loss set to ‘categorical_cross_entropy’, and an ‘accuracy’ metric were chosen to compile each model. Each layer, except the output layer, used a ‘relu’ activation , with the latter having ‘softmax’. Three separate training and test sets were used to train each model, with a test size of 0.2., in a random state of 42. The aforementioned outliers being randomly placed (perhaps even skewed) into each test set is likely a factor in the differing performance of each algorithm.