Cool Beans

Final Report

Anna Anello, Jake Reynolds, Rowan Tickle, Kayla Dougherty

Math 358

Table of Contents

1.1 Abstract

Much of our motivation for pursuing this project came from the fact that this data was collected by a computer vision system from images, which we had not worked with before. We used LDA, QDA, and Decision Tree models with the data set as we received it, and used kNN classification with that same original data set as well as versions that we had standardized. We then compared our accuracies for the various models and bean types to the accuracies in the single other paper that uses this data set. Our best model, a kNN model with standardized data, was not quite as good on average as the other paper's best, but ours was better than their kNN model.

1.2 Introduction

Our project dealt with the classification of beans from images. This type of work is important because it relates to the development of computer vision and image classification—fields which are still evolving and improving, and have to do with important and controversial topics like facial recognition. Much of our motivation for pursuing this problem comes from the fact that this data was different from the data that we had used throughout the semester; we had not used data collected from images before. Additionally, the measurements taken for the attributes were in reference to pixels, which were easier for a computer to handle but difficult for people to interpret.

The inputs to our algorithms were measurements in pixels taken from images of beans by a computer vision system. We used Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), k-nearest neighbors (kNN), and Decision Tree algorithms to output a predicted bean type.

1.3 Related Work

There is currently one published paper using this dataset; it is by Murat Koklu and Ilker Ali Ozkan. In their study, they used Multilayer Perceptron, Support Vector Machine, kNN, and Decision Tree models for their data. The accuracies that they obtained for all of these models were 91.73%, 93.13%, 87.92%, and 92.52% respectively. They also obtained individual accuracies for each bean class--the bean

classes being Barbunya, Bombay, Cali, Dermason, Horoz, Seker, and Sira. These individual bean class accuracies from the Support Vector Machine model--their best performing model--were 92.36%, 100%, 95.03%, 94.36%, 94.92%, 94.67%, 86.84% respectively. We tried to improve upon the results that they got for the kNN and Decision Tree models.

1.4 Dataset and Features

We chose to work with the Dry Bean Dataset from the UCI Machine Learning Repository. The set consists of 13,611 high-resolution images processed in a computer vision system. The total number of variables in the set is 17--including the classification--and there are no missing values. A computer vision system was used to describe the beans using 16 variables, including 14 dimensional features and 4 shape forms. Examples of the dimensional features include area, perimeter, solidity, and roundness. The 4 shape forms broke down the bean shape even further. After the segmentations and feature extraction of the beans, they were placed into one of 7 categories. The bean classes are Seker, Barbunya, Bombay, Cali, Dermosan, Horoz, and Sira, as listed above.

1.5 Methods

For our QDA, LDA, and Decision Tree models, we did not do any data normalization and just used the dataset as it was given. We then also performed the kNN classification using the original dataset and two forms of standardized data.

1.6 Experiments/Results/Discussion

The first model that we created was the kNN, using an 80/20 split for training and testing, without changing anything in the dataset. This approach had an overall accuracy of 71.36% and had an average run time of about 3.4 minutes. To improve upon our model, we decided that we needed to standardize the data and sought to find the categories that had many outliers. We obtained a boxplot of all the variables and found that all of the variables aside from Shape Factor 2 had heavy outliers. Because of this

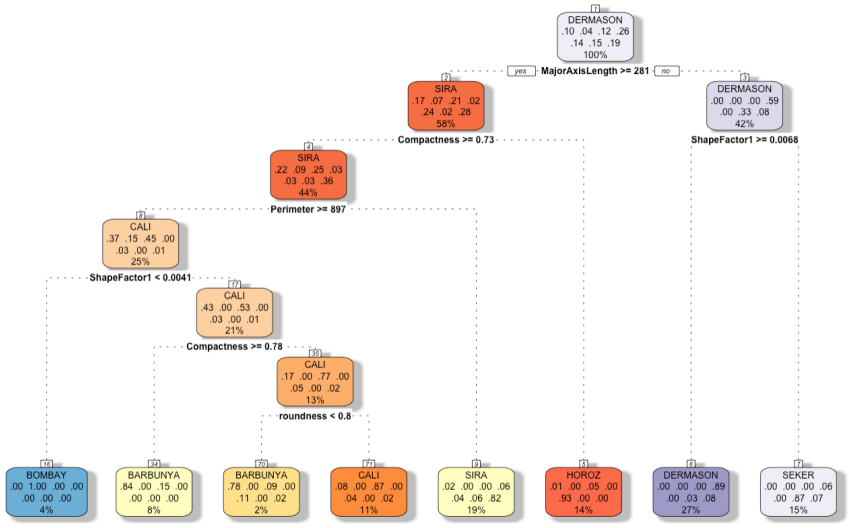
observation, we then standardized all of the variables besides Shape Factor 2 and ran kNN again, using the same 80/20 data split. This created a new model that had an overall accuracy of 91.88%, which was a 20.52% increase in overall accuracy from our first kNN model. This new model had an improved run time of 2.78 minutes, which was 0.62 minutes faster than the original model. We decided to try one more change to our model and to standardize the entirety of the dataset. We again used an 80/20 split for training and testing and ran our kNN algorithm again. This gave us our best overall accuracy of 92.06%, but the average run time was 2.9 minutes. This model also was better than the previous paper's overall kNN accuracy of 87.92% by 4.14%. In addition, this model gave us balanced accuracies for Barbunya, Bombay, Cali, Dermason, Horoz, Seker, and Sira that were better than the balanced accuracies for the best model in the previous paper. The balanced accuracies that we obtained for each bean type were 94.55%, 100%, 96.62%, 94.78%, 96.05%, 96.4%, and 92.21%, respectively.

The second model we ran was the LDA model where we still split the data into training and testing data sets using an 80/20 split. The LDA model works by approximating the Bayes classifier (δk) using the proportion of the training observations that belong to the kth class, the average of all of the training observations from the kth class, and the weighted average of the sample variances for each of the k classes. It then assigns an observation to the class for which the Bayes classifier, $\delta k(x)$, is maximized. Our overall LDA model achieved an overall classification accuracy of 91.4% which indicates that the model is pretty good for determining the bean type. Also, when broken down into each bean class, all of the individual accuracies are above 92.6%. In particular, the Bombay bean had an accuracy of 100%, the Cali bean had an accuracy of 97.44% and the Horoz bean had an accuracy of 96.86%--all of which are close or equal to 100% accuracy.

The next model we ran was the QDA. Just like the LDA model, the QDA is based on Bayes theorem and classification. The QDA model is used to find the non-linear boundary between these classifiers. We split the data into training and testing datasets again with an 80/20 split. This model gave

us an overall classification accuracy of 91.88%, which indicates good performance in classifying the beans. Something to note is that the Bombay bean has a balanced accuracy of 100% and the Cali, Horoz, and Seker beans each have a balanced accuracy of around 97.6%. The lower prediction accuracies are those of the Dermason and Sira bean, at 92.47% and 91.93%, respectively. Overall, this model was still pretty successful.

The final model we ran was a Decision Tree, using the same 80/20 data split to create the training and testing data sets. The Decision Tree classifies the data by majority, which is just picking the most common class in every region in order to minimize the classification loss function. This model produced an overall accuracy of 86.95%, which was a decrease by about 4% from the QDA model and about a 5% decrease from the kNN model with all data standardized. This model did, however, perform a little better when broken down into class.The Bombay bean achieved 99.519% balanced accuracy. The Dermason bean achieved 94.47% accuracy, which is about a 2% improvement from the QDA model. A visualization of this decision tree is below. This model performed the worst out of all of the models that we ran for this project.



1.7 Conclusion/Future Work

After running the kNN, QDA, LDA, and Decision Tree models, our best model performance was the kNN with all of the data standardized, which achieved an overall accuracy of 92.06%. This was only 1% lower than the highest overall accuracy achieved in the previous study. Our kNN model was 4.14% higher than the previous study's kNN model. However, our Decision Tree accuracy was 5.57% lower. The difference between our kNN performance and theirs can likely be attributed to the fact that we standardized all of our data for the kNN model, which helped avoid misclassification and thus increased our accuracy. As for our decision tree, the overall classification accuracy was the lowest of the 4 models because there is likely not as much separation between bean types to be able to split the data based on the most common class. The previous study most likely achieved a higher Decision Tree overall accuracy because they used upsampling and/or downsampling to help with misclassification. In the future, we would like to improve upon our results by implementing upsampling and downsampling. We would also like to run a Support Vector Machine model as the previous study had done, as it achieved their highest overall accuracy of 93.13%. Although that is only about 1% higher than our best model, we would still like to get as close to 100% accuracy as possible.

	Barbunya	Bombay	Cali	Dermason	Horoz	Seker	Sira	Overall
Highest Accuracy Values from Koklu and Ozkan	92.36	100.00	95.03	94.36	94.92	94.67	86.84	93.13
Our Highest Accuracy Values	95.11	100.00	97.55	94.94	97.65	97.55	93.12	92.06

This table lists the highest accuracies in percent from the Koklu and Ozkan paper and our highest achieved accuracies. These are not all from the same models.

Bibliography:

KOKLU, M. and OZKAN, I.A., (2020), â€œMulticlass Classification of Dry Beans Using Computer Vision and Machine Learning Techniques.â€ Computers and Electronics in Agriculture, 174, 105507.