**Predicting Bean Types from Defining Features Using Various Machine Learning Techniques**

## DATA SCIENCE 4  COURSE: MACHINE LEARNING

## University of Toronto School of Continuing Education

## University of Waterloo School of Professional Development

## December 13th, 2021

Group 7: Amr Abdalla, Mitchell Stevens,Wen Tang, Matt Wash, Treshan Waas, Ting Zhang

# **Introduction and objective**

The dry bean dataset is a popular open-source dataset available for users to experiment machine learning models. The dataset consists of seven types of dry beans recorded with its characteristics such as shape structure etc. Computer aided images were used of 13,611 grains and 16 features were recorded classifying the type of the bean. The objective of our analysis was to build, train, test and evaluate the performance of multiple machine learning models to determine the most effective model for our dataset.

# **Data source and preparation**

Chart, scatter chart

Description automatically generatedData source - <https://archive.ics.uci.edu/ml/datasets/Dry+Bean+Dataset>

Chart, pie chart

Description automatically generatedTable

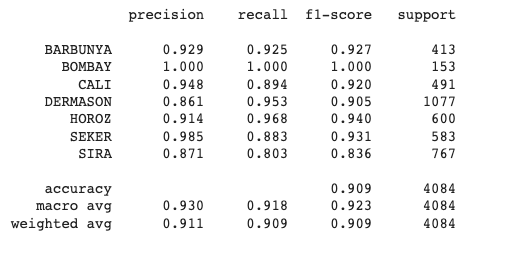
Description automatically generated with low confidenceThe dataset consisted of 13611 observations with 16 features with the column mentioning the class of the bean. There were 7 bean classes that were recorded. The dataset consisted of bellow percentages of observations from each class. The correlation matrix showed many features that were correlated and it the scatter matrix also showed many correlated variables. When visualizing the scatter matrix according to the type of data bean (Class), some features showed clear categories that could be classify. The train test split was 70:30 for modelling.

# **ML Modelling**

**KNN Classification**

Chart, line chart

Description automatically generatedThe dependent variable Bean types is categorical. Both MinMax scaler and Standard Scaler were tried. It turns out that Min Max scaler gave a model of a bit higher accuracy score so transformation of Standard Scaler was removed from the codings. The overall model accuracy was 90%. Two models were tested with one model with 10 features and other with 16 features

.

**Random forest**

Table

Description automatically generatedWe used SMOTE library to balance the target data column by applying a pipeline preprocessing step to both over sampling and under sampling the target column. In order to train the model we used a pipeline and transformed the data with standard scaler.

Chart

Description automatically generated

**Linear SVM model**

Our third chosen classifier was a multiclass implementation of Support Vector Machine, implemented in scikit-learn as LinearSVM. Because of the scoring method’s sensitivity to geometric shearing, we standardized all of the features in the training data before fitting the model. We tuned our hyperparameters using a grid.

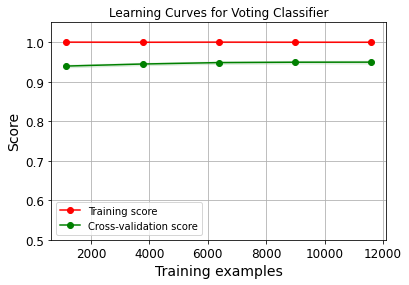
Chart, line chart, scatter chart

Description automatically generatedTable

Description automatically generated

# **Voting Classifier**

Chart

Description automatically generatedTable

Description automatically generatedTable

Description automatically generatedThe last classifier used in our analysis is an ensemble learning method, voting classifier, which predicts a target value by using the majority vote (i.e. hard voting) or the average predicted probabilities (i.e. soft voting) of the pre-trained classifiers. Both hard and soft voting were implemented in our analysis. KNN and random forest hard voting classifier.  The soft voting requires the input classifiers to calculate the class probabilities in order to predict the class labels.

*Hard voting*

*Soft voting*

# **Conclusion**

All the classifiers that we picked up have high accuracy scores and high training scores close to 1. According to the confusion matrices of each classifier, most wrong predictions/confusions occurred on class 3 and 6 (Dermanson and Sira). However, according to the confusion matrices, the performance of the models are: random forest = voting > linear SVM >kNN classifier. The dry bean dataset is very neat and clean so it fits well for any of the classifiers that we picked up.