Multi-Class Classification with Machine Learning

COMPSCI571:Probabilistic Machine Learning: Spring 2020

Sivanand Devarakonda (sd374@duke.edu)

Department of Mechanical Engineering, Duke University

May 2020

1 Introduction

Classification is one of the most common type of application for machine learning method. Several machine learning models can be implemented to solve a required classification problem. Classification is about assigning a label based on the several parameters for a given data. Image recognition, Speech recognition, News classification, sentiment analysis etc., are some of the example applications of the classification machine learning. The scope of this project pertains to multi-class classification. Based on the several parameters a data point can be classified into multiple categories unlike binary classification where there are only two classifications. There are several algorithms used for classification. All the algorithms are compared for the data to understand the best fit.

1.1 Logistic regression

Logistic regression also called 'logit' is a linear model to classify the data based on the probability. The probability is calculated using the sigmoid function. The model takes input as a number of parameters and outputs the probability between 0 and 1. The classification can be assigned as 0 for probability between 0 and 0.5 and 1 for probability between 0.5 and 1. This method can be extend to multi-class classification.

1.2 K-nearest neighbours

The basic idea here is the cluster the training data set based on its corresponding categories. The position of the test data is then assessed compared to this fit model. 'k' represents the number of points that are closest to the test data. The model identifies the cluster with 'k' number of points closest to the test data and classifies it accordingly.

1.3 Support Vector Machine

In this method, a hyper-plane is constructed between the various clusters of data. The plane is construed in such a way that it maximized the the normal distance between the plane and two data points of different clusters. The test data is classified based on which side of the plane it belongs to.

1.4 Naive Bayes Method

This method utilizes the probability of occurrence of test data given the occurrence of the input data. Once this 'Bayesian probability' is established the value can be used to classify the test data. There are assumptions for this method. All the parameters of the input data are mutually independent and produce same result in that particular class.

1.5 Linear Discriminant Analysis

This method works by reducing the dimensionality of the training data. The data is then projected on to a single line. The data is then classified into different classes based on its distance from the centroid of the entire data. This model effective on the data with linear relationship.

1.6 Random Forest Method

Also known as decision tree method, this model breaks the data-set down into smaller and smaller subsets based on different criteria. The end of the network is then assigned to a corresponding class.

2 Code and Data description

The code developed in this project reads a data set of different parameters for classification of seeds into different types of wheat. The examined group comprised kernels belonging to three different varieties of wheat: Kama, Rosa and Canadian, 70 elements each, randomly selected for the experiment. The parameters involed in the data are:

1. Area[A]
2. Perimeter [P]
3. Compactness [C] where, $C = 4\pi A/P^2$
4. Length of kernel [Lk]
5. Width of kernel [Wk]
6. Asymmetry coefficient [Ac]
7. Length of kernel groove [Lkg]

They can be classified into class: 1 = Kama, 2 = Rosa and 3 = Canadian.

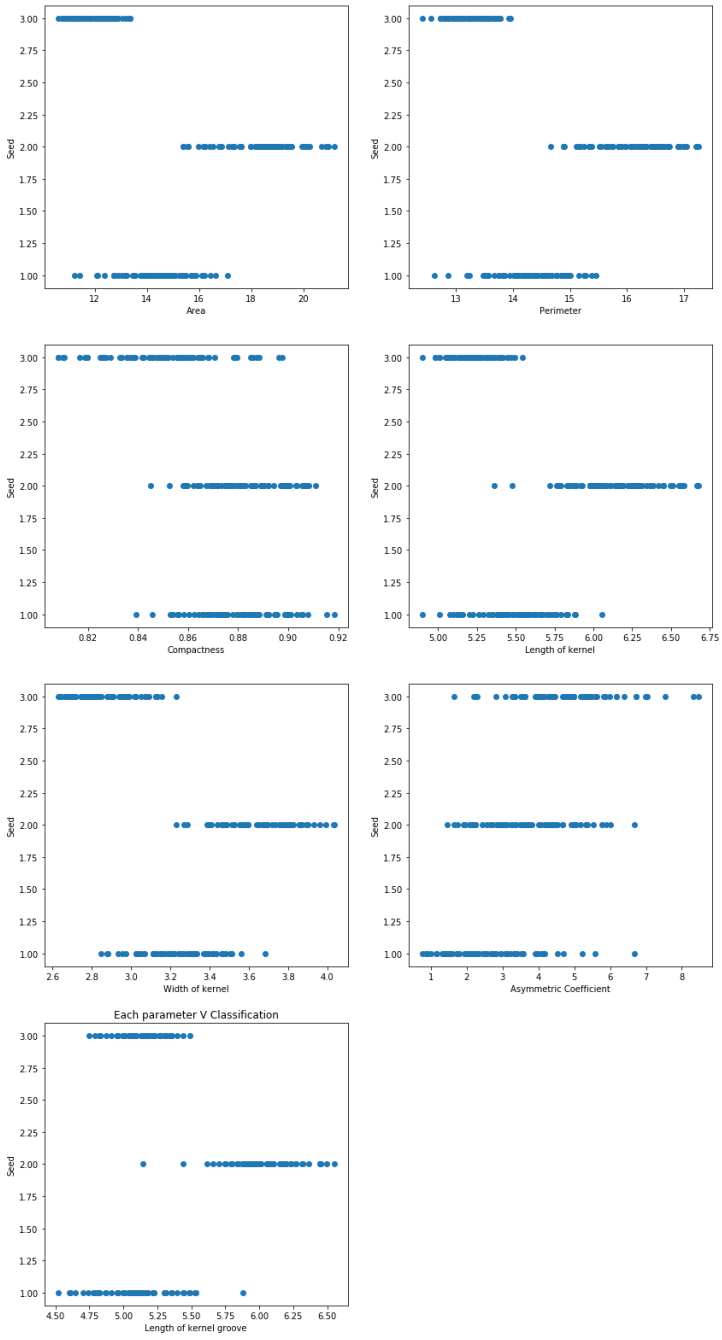

Figure 1: Data Visualization

This data set is visualized in Figure 1. The code reads any data set in .csv format where the several parameters are defined and the label column is named as 'Result'. The data set is split into training set and test set in several proportions specified in the code (75%,70%,...so on of total data is used as training data). The data can be split randomly by the code. This split is looped over 1000 times and a fit with each model is done and compared with the test data generating a score value in each loop. This scores are averaged to get an estimate of how the model is performing on an average.

3 Results

Figure 2 to Figure 7 describe the scores obtained for each model for several proportions of training data. Linear discriminant analysis (LDA) has the best scores throughout indicating that the data is fairly linear in nature. All the models have high scores but LDA was relatively better. Naive bayes and K-nearest neighbours methods have shown better scores for a reduced proportion of training data, while all the other models strictly decrease in performance as the training data proportion is decreased.

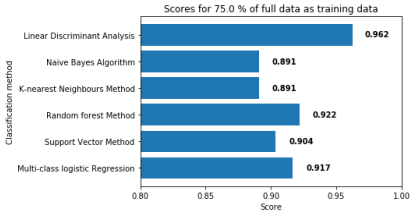

Figure 2: Scores when 75% of the total data is used as training data

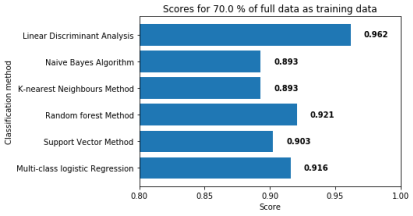

Figure 3: Scores when 70% of the total data is used as training data

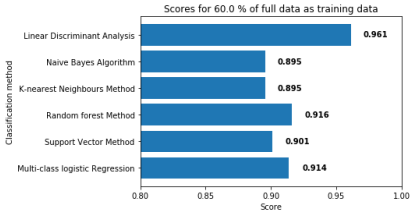

Figure 4: Scores when 60% of the total data is used as training data

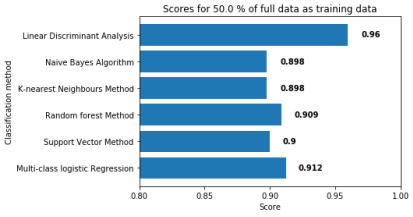

Figure 5: Scores when 50% of the total data is used as training data

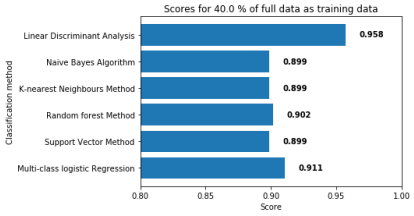

Figure 6: Scores when 30% of the total data is used as training data

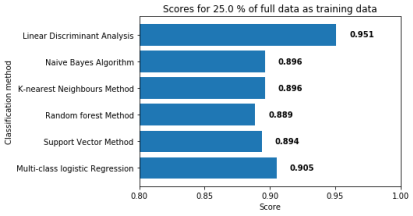

Figure 7: Scores when 25% of the total data is used as training data

4 Conclusion

The results of this project compare the different classifier methods for a given data across a range of proportions of training data of the total data. Based on the nature of the data some models fit the data better than other. For the seed data used in this project, while all the models show a high score, Linear discriminant Method had consistently the highest scores and can be concluded to the best model for the data. It is also a good practice to use a higher proportion of data for training the model to obtain better fit. The code can be used for any data-set to compare and identify the best model to be used for solving the classifier problem. Understanding of the classification model and nature of data are to be considered to validate a good fit.

References

[1] Nelson, Dan *Overview of Classification Methods in Python with Scikit-Learn* April 7th 2020, https://stackabuse.com/overview-of-classification-methods-in-python-with-scikit-learn

[2] Asiri, Sidath Dan *Machine Learning Classifiers* April 7th 2020, https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623

[3] M. Charytanowicz, J. Niewczas, P. Kulczycki, P.A. Kowalski, S. Lukasik, S. Zak, *A Complete Gradient Clustering Algorithm for Features Analysis of X-ray Images*, Information Technologies in Biomedicine, Ewa Pietka, Jacek Kawa (eds.), Springer-Verlag, Berlin-Heidelberg, 2010, pp. 15-24., https://archive.ics.uci.edu/ml/datasets/seeds