Purpose

The purpose of this document was to record the steps and observations of a Machine Learning project that used the Breast Cancer Wisconsin (Diagnostic) Data Set found at http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/; file: WDBC.data.

The only difference between this project and the Iris ML project is there are more attributes, needed to do this to get a little more familiar of formulating a model.

This machine learning problem’s model was offline, used supervised learning, and had classification outputs. The dataset requires no categorical conversions or handling of null data.

Procedure

The workflow of building an accurate machine learning model is an iterative process where the data used to train the model dictates how the learning algorithms parameters should be tweaked to impact to model. The goal is that each iteration of the process results in a higher accuracy for its predictions on the unseen training data. This mini project is an end to end project to practice one iteration of the machine learning process.

Following several well-known steps:

1. Define/understand the problem: In this problem the model is offline since the model was trained on a dataset and deployed. It used supervised learning since in the dataset each feature vector had a target attribute. The model can be defined as a classification model since the outputs are discrete values.

Y, target Attribute: [“Diagnosis”];

1. Prepare data: In the dataset it contains one attribute that is categorical and since it is the target attribute it does not need a conversion. The dataset does not contain missing or null values. The only thing to do is to not involve the ID column of the dataset because it does not contribute to the predictions.
2. Selecting the Algorithms: In this project the training dataset was 80% of the original dataset and the testing dataset was the remaining 20%.

From observations from data visualization and the problem’s characteristics like being a classification problem, a couple of algorithms can be chosen to build models to compare which would be the best to continue with.

* To cover the bases a good mix of machine learning algorithms are used to build different models:
* Logistic Regression (LR); linear
* Linear Discriminant Analysis (LDA); linear
* K-Nearest Neighbors (KNN); nonlinear
* Classification and Regression Trees (CART); nonlinear
* Gaussian Naive Bayes (NB); nonlinear
* Support Vector Machines (SVM); nonlinear

1. Train the models: The type of learning to train the model is supervised since the dataset is labelled.
2. Evaluate: Evaluate the models’ accuracy on testing dataset during the testing process to determine the performance of the model. In this case the model that presented the highest accuracy of the selected algorithms was Linear Discriminant Analysis with 95%.
3. Predict:

* Use the model to predict on new unseen data and access the performance to determine improvements that can be made on the model to improve accuracy.

1. Improve Results:

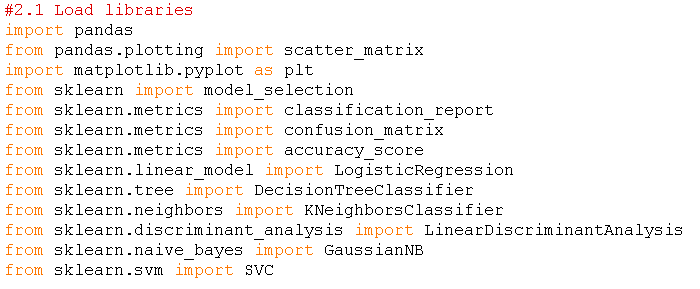
Currently this project is a one and done.

1. Present Results:

The usual metrics to present is the accuracy of the model on the testing data, the confusion matrix to observe, numerically, the model’s correct and incorrect predictions, and the classification report to present the model’s precision of the labels.

Observations:

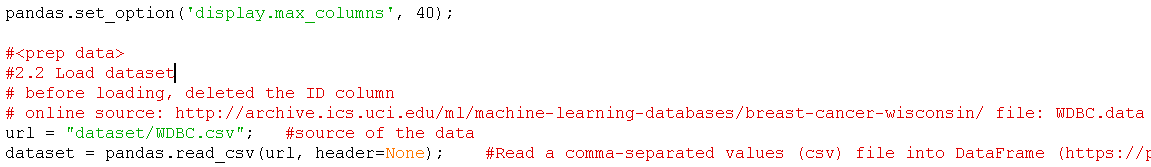
<Import libraries>



</Import libraries>

<1. Define/understand data>

* 1. Load the Dataset



* 1. Summarize the Dataset
     1. Dimensions of the Dataset

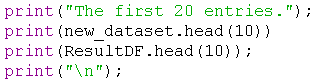


Output:

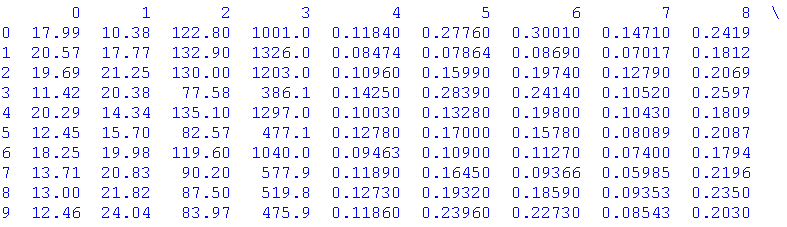


Therefor 569 instances and 31 attributes.

* + 1. Peek at the Data



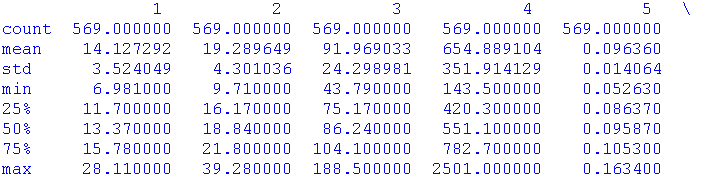
Output (partial of new\_dataset which excludes the labelled attribute):



* + 1. Statistical Summary



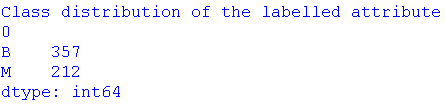
Output:



* + 1. Class Distribution



Output:



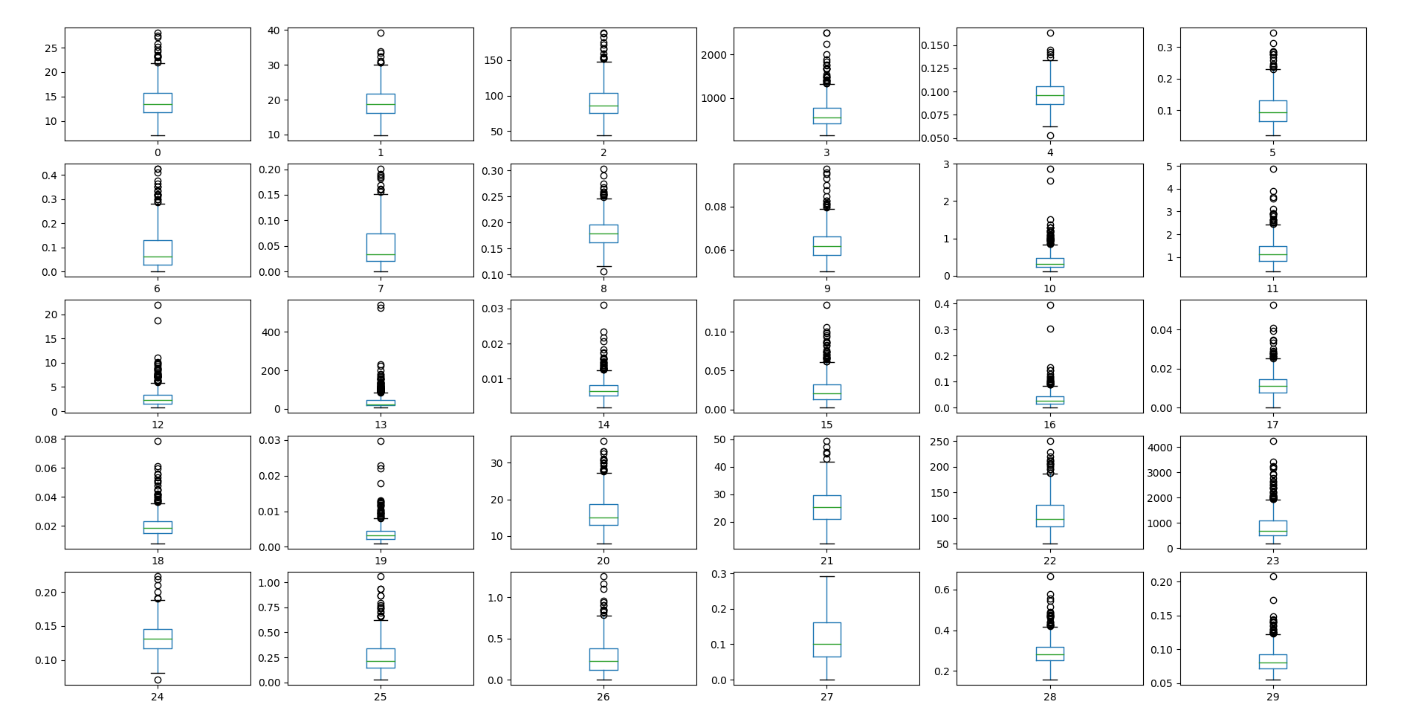
* 1. Visualize the Data
     1. Univariate Plots

It is must easier to see the distribution of individual attributes using univariable plots.

- A simple plot used was the whisker plot:



Output:

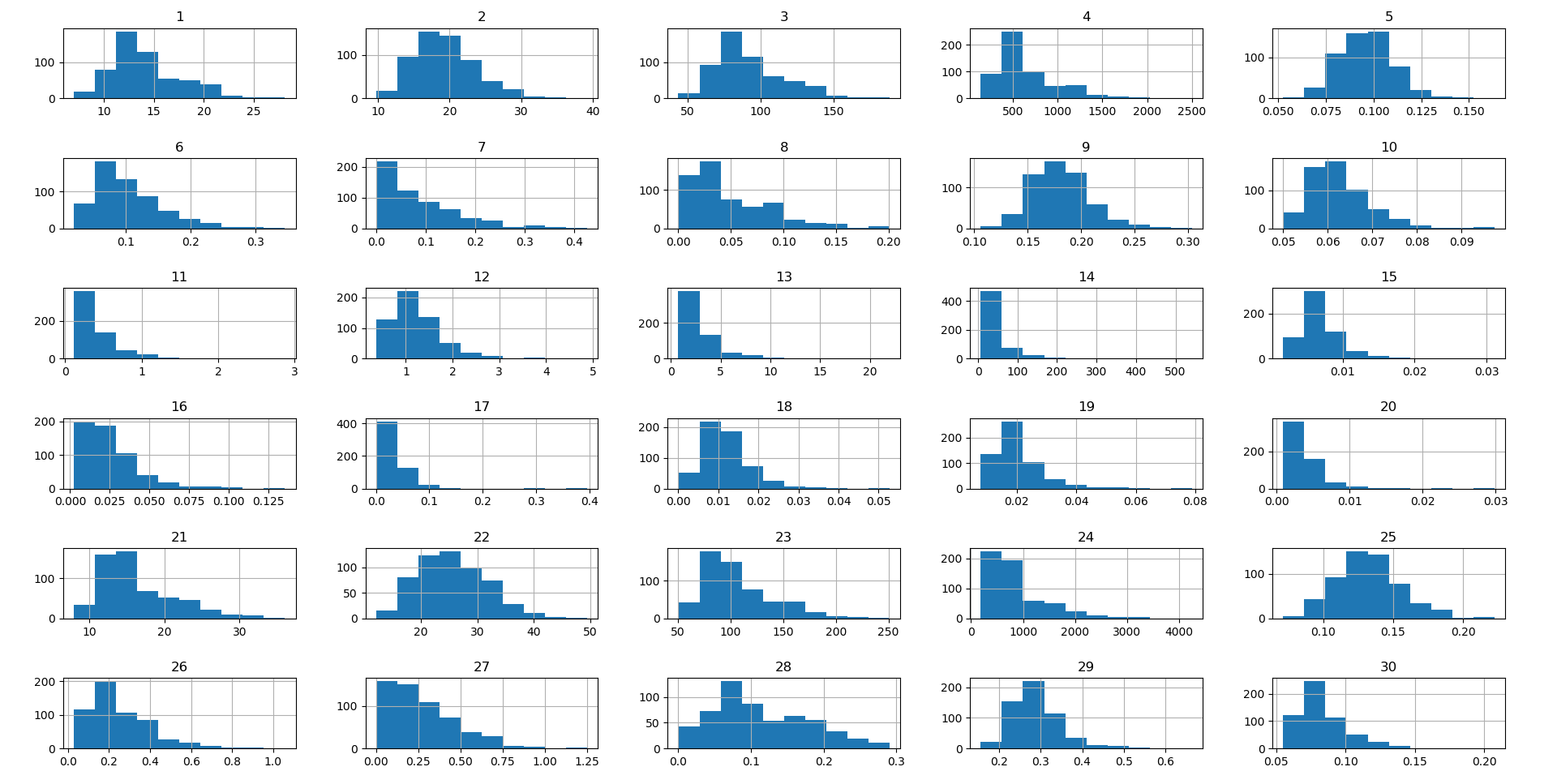


Observations: From the whisker plot, a couple of attributes may resemble a Gaussian distribution like 2, but a histogram will more clearly show the distribution.

- Another plot that can be used to visualize the distribution of the attributes is to use a histogram:



Output:



Observations: Some Gaussian curves are present like 2, 5, 9, 22, 25, and 28, this means that the algorithms that are based on the distribution have a good chance of doing well.

* + 1. Multivariate Plots

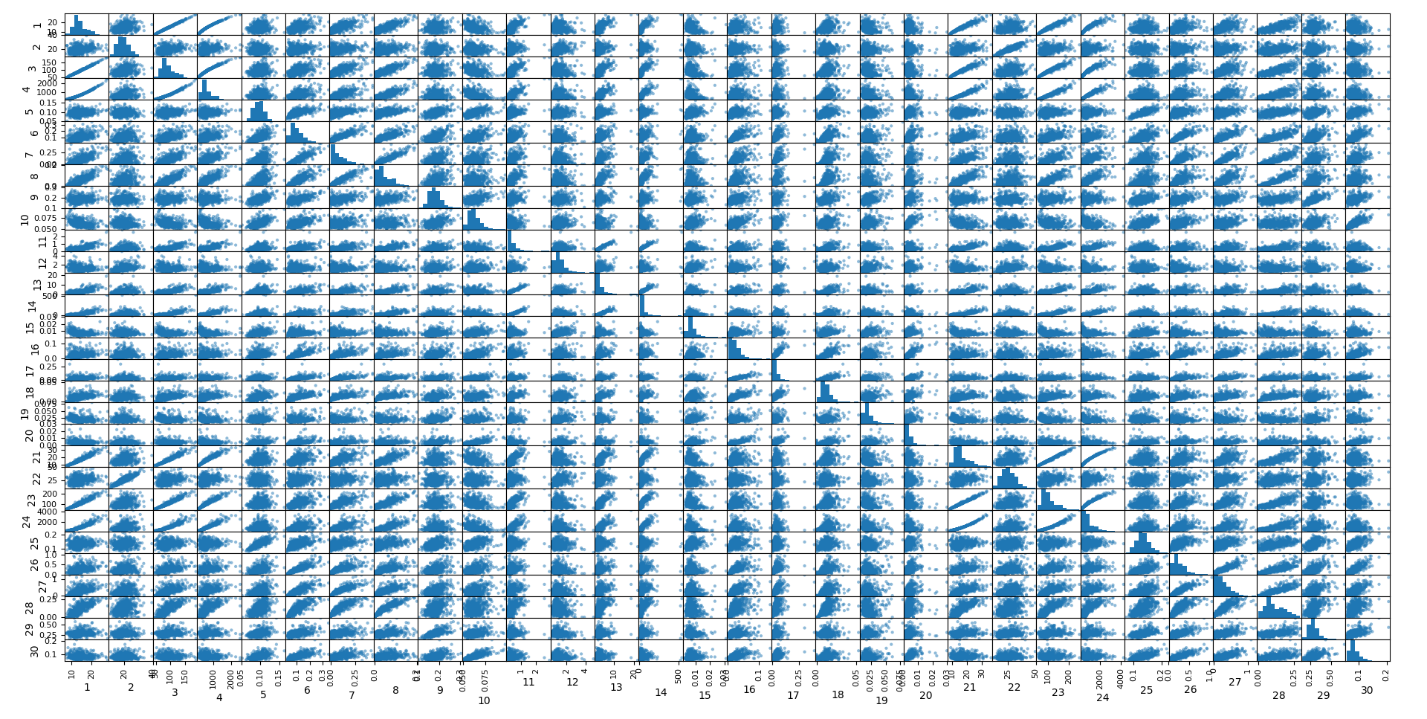
The use of multivariable plots is to visualize the interactions between the attributes, and it may reveal some correlation.

A quick method to plot all the combinations is with a scatter matrix:



Due to the number of attributes, it requires more time to display and it becomes crowded. In the future a better and more flexible method is needed.

Output:



Observations: Some relations display a linear relationship, but majority of attribute names are unknown and as noted in the .names doc that some attributes are ratios of other attributes. Nonetheless, having good linear correlations increases predictability to lead to a better performing model.

</1. Define/understand data>

<2. Prepare data>

2.1 Evaluate Some Algorithms

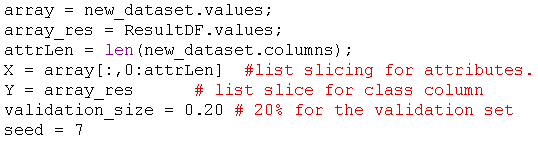
The steps for evaluating algorithms includes:

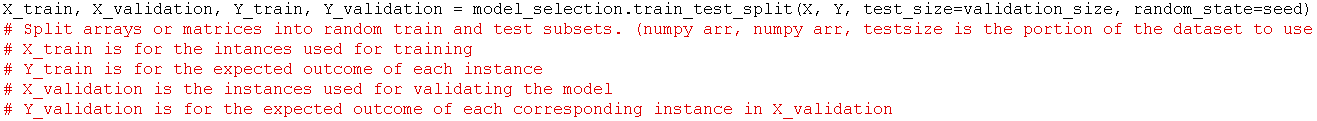
1. Separate out a validation dataset.
2. Set-up the test harness to use 10-fold cross validation.
3. Build 5 different models to predict species from flower measurements
4. Select the best model.

2.2 Create a validation dataset

From the total dataset a portion of it must be reserved as validation dataset and not be included in the training process’s dataset. This unseen data is used to evaluate how well the model predicts on it and based on the accuracy of each model we can compare algorithms.

In this case 20% of the dataset was reserved as the validation dataset:





2.3 Test Harness

For the test harness 10-fold cross validation was used to estimate accuracy. Fold cross validation definition: <https://www.openml.org/a/estimation-procedures/1>

This will split our dataset into 10 parts, train on 9 and 1 part to test predictions and it repeats for all combinations of train-test splits.



Seed value initializes the randomization and ‘accuracy’ refers to the percentage of correctly predicted outcomes out of the total number of instances in the dataset.

</2. Prepare data>

<3. Select algorithms>

3.1 From observations pick some that you think would work and some others that round off the pool. todo

</3. Select algorithms>

<4. Train the models>

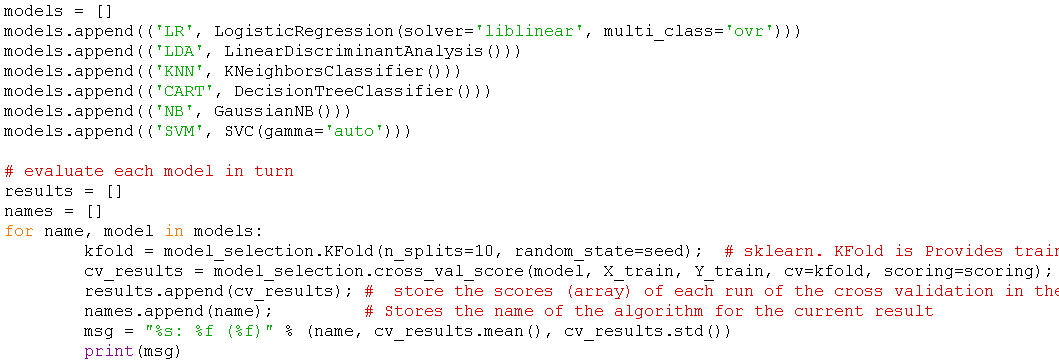
4.1 Build Models

The algorithms used were:

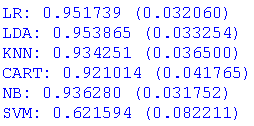
1. Logistic Regression (LR)
2. Linear Discriminant Analysis (LDA)
3. K-Nearest Neighbors (KNN).
4. Classification and Regression Trees (CART).
5. Gaussian Naive Bayes (NB).
6. Support Vector Machines (SVM).

It is good mixture of simple linear (LR and LDA), nonlinear (KNN, CART, NB and SVM) algorithms. The random number seed is reset before building the next model to ensure that the evaluation of each algorithm is performed using exactly the same data splits. It ensures the results are directly comparable.

To build the models:



Output:



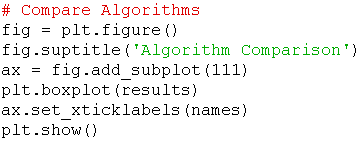
</4. Train the models>

<5. Evaluate>

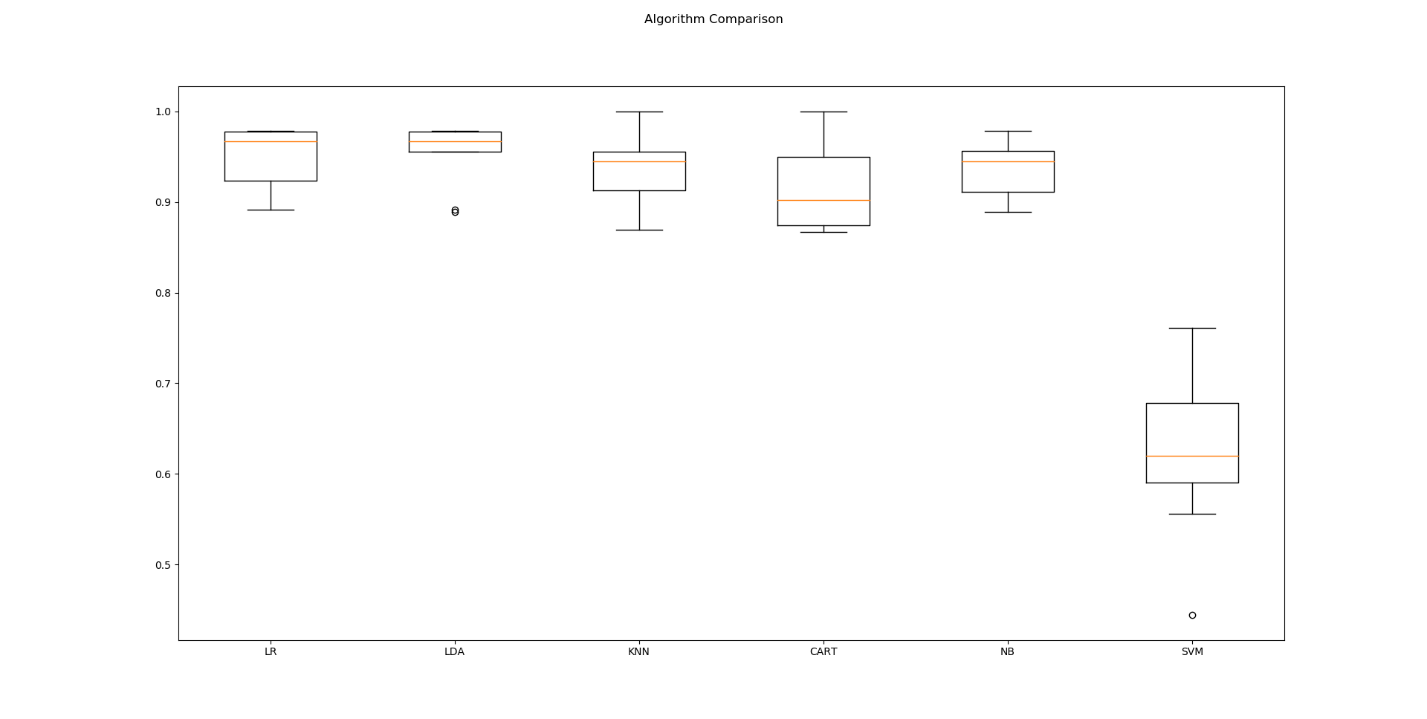
5.1 Select Best Model

From the evaluation of each algorithm it can be observed that LDA is the most accurate with ~95 %.

Each algorithm was evaluated 10 times since 10 k fold cross evaluation was used and to visualize the distribution of the accuracies of each algorithm a whisker plot was used:



Output:



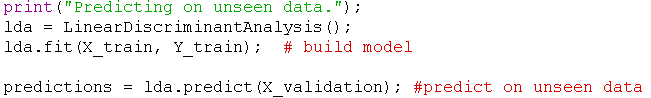
Observations: LDA has high accuracy and low standard deviation making it a good choice to build a model with it.

</5. Evaluate>

<6. Predict>

6.1 Make Predictions

The next step is testing a model on the validation dataset. LDA was chosen and was used to train a model, then predictions were made on the unseen data in the validation dataset to test its accuracy.



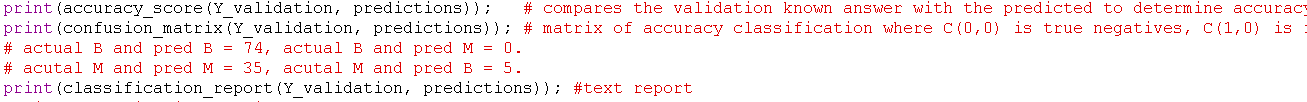
</6. Predict>

<7. Improve Results>

#No improvements performed.

</7. Improve Results>

<8. Present Results>



Outputs:

- Accuracy score: Compares the validation known answer with the predicted to determine accuracy.



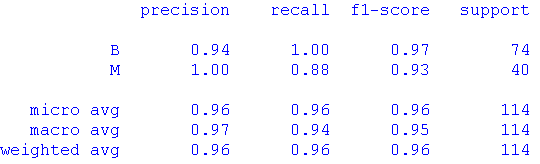
- Confusion matrix: Matrix of accuracy classification.



It can be interpreted as:

|  |  |  |
| --- | --- | --- |
| Actual \ Prediction | B | M |
| B | 74 | 0 |
| M | 5 | 35 |

- Classification report:



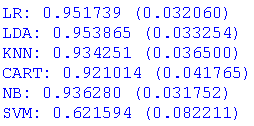
Observations: The model created has good overall precision in its predictions.

</8. Present Results>

Conclusion

This machine learning problem’s model was offline, used supervised learning, and had classification outputs. The dataset required no categorical conversions or handling of null data.

The models initially built presented accuracies of



, from this LDA was chosen to build a model and predict on new unseen data and yielded an accuracy of ~95.6%.

Completing this project reinforced some basic ML procedures to complete one iteration of an end to end machine learning project.

The next project should definitely exercise at least one additional step in the machine learning procedure, for example data cleaning, a regression output, an online model, or other.