**Wine Classification Model**

***Problem Statement:*** *Wine Classification problem*

*Build a machine learning model which can predict the class of wine to which an observation belongs to, taking the results of chemical analysis on wine as independent variables.*

*Dataset:*

*• This data contains the results of a chemical analysis of wines grown in the same region in Italy*

*but derived from two different cultivars. The analysis determined the quantities of 13*

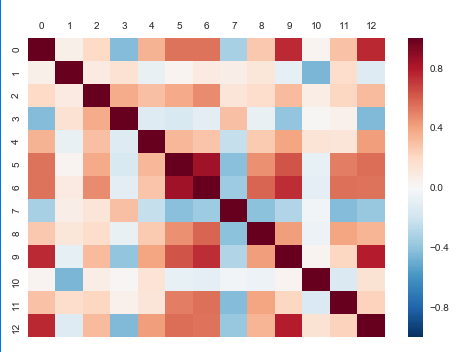
*constituents found in each of the two types of wines.*

*• Two data sets has been provided, one for training another for testing.*

*• All the attributes of the chemical analysis are continuous in nature.*

1. **Data pre-processing:**

* Checked for missing values in the data – none
* Performed feature scaling on the dataset.
* Plotted a heatmap to observe the correlations between variables



* Below table shows the top 10 highly correlated pairs of variables, the same can be observed from the heat map too. This information might be helpful for parameter tuning during the model selection.

|  |  |  |
| --- | --- | --- |
| **Variable 1** | **Variable 2** | **Correlation** |
| total\_phenols | Flavonoids | 0.850159 |
| color\_intensity | Proline | 0.787798 |
| Alcohol | Proline | 0.750325 |
| Alcohol | color\_intensity | 0.746816 |
| Flavonoids | color\_intensity | 0.742087 |
| total\_phenols | color\_intensity | 0.631877 |
| Flavonoids | proanthocyanins | 0.585567 |
| total\_phenols | proline | 0.560191 |
| Flavonoids | OD280/OD315\_of\_diluted\_wines | 0.551775 |
| Flavonoids | proline | 0.543992 |

* Model building:

|  |  |
| --- | --- |
| Name of the algorithm | Accuracy on test data |
| Logistic regression | 0.75 |
| K-NN | 0.78125 (K=9); 0.75(K=5); 0.70(K=6) |
| SVM | 0.8125 |
| Naïve Bayes model | 0.84375 |
| Decision tree model | 0.71875 |
| Random forest model | 0.8125 |
| Kernel SVM model | 0.875 |

Checked a couple of algorithms and used Model Selection techniques: K-fold cross validation(explained in the next section) and Grid Search to perform parameter tuning on the prospective model.

Grid Search also helped me determine that the data is not linear, thus SVM & Logistic Regression algorithms were not pursued further.

Kernel SVM does a good job of tackling non-linear datasets. It projects the data points which are not linearly distributed into a high dimensional space using a kernel function.

Parameters tuned for Kernel SVM model during GridSearch:

* C: Best penalty parameter C to prevent overfitting (C=1)
* Kernel: to decide between linear and non-linear model (kernel=’rbf’)
* Gamma: optimize gamma to find optimal kernel (gamma=’auto’)

1. **Confusion matrix on Train data**

Confusion matrix on training dataset is as follows:

(36, 6

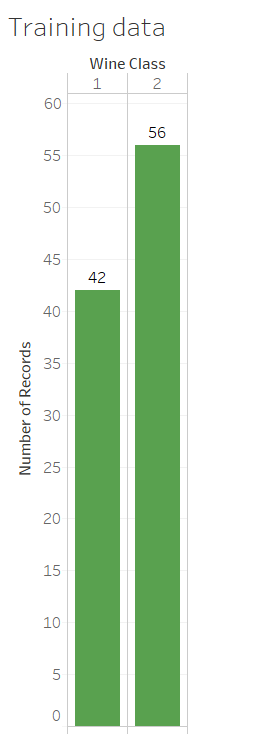
11, 45)

Accuracy = (36+45)/(36+6+11+45) = 0.8265

Evaluating the model performance by only testing the accuracy on the test set might not be the best method. This is because there could be variance in the test data we are using, hence the accuracy might look different when tested on a new set of data. Hence it better to use techniques like k-fold cross-validation.

*K Fold Cross validation: Provides train/test indices to split data in train/test sets. Split dataset into k consecutive folds (without shuffling by default).*

*Each fold is then used once as a validation while the k - 1 remaining folds form the training set.*



The give training dataset has more observations pertaining to class 2 compared to class 1. Hence K-fold approach provides a more realistic range of accuracy.

Mean Accuracy after k-fold cross validation: 0.7766; Standard deviation of the accuracy: 0.06

Implies that the accuracy can vary between 71.66 to 83.66

1. **Confusion matrix on Test data**

Confusion Matrix and Statistics  
  
                   Reference  
Prediction   1    2  
         1        11  3  
         2        1   17  
  
  
Then accuracy is calculated as (11+17)/(11+17+1+3) = 0.875

1. **Python Script**

I have included my python code in a Word document

