The analysis of Wine datasets by KNN and Decision tree

Student name: Yijun Zhang Student number: s3730883
Student name: Shuo Wang Student number: s3677615
Affiliations: RMIT
Contact Detail:
Yijun: s3730883@student.rmit.edu.au
Shuo: s3677615@student.rmit.edu.au
Course name: Practical Data Science Course code: COSC 2670
Teacher’s names: Yongli Ren

Submission date: 23 May 2019
Semester: Semester 1 2019 (1910)

Abstract:

The aim of this report was to classify 3 different types of wines into target labels by using chemical elements by data modeling, and data exploration. As a classification data set, the K Nearest Neighbor and Decision Tree have been used for modeling, and hill climbing, K-Cross Validation and nested loop have been used in tuning parameters. Overall, the accuracy of the prediction results was between 81%-99%. From the results obtained, it is recommended that the K Nearest Neighbor model is more suitable.

1 Introduction:

This wine dataset which is downloaded from the UCI Repository: http://archive.ics.uci.edu/ml/ is the results obtained from a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The original dataset has 13 numerical attributes and 1 categorical attribute with 178 instances. This report may process data retrieving which checks the correctness of the data, data exploration which explores the relationship with different attributes and data modeling which trains a suitable data model for the dataset and chooses a better machine learning for predicting the results of the dataset.

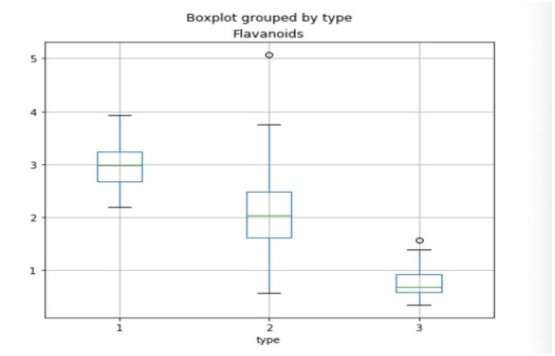
2 Findings

Task 1: Data Retrieving

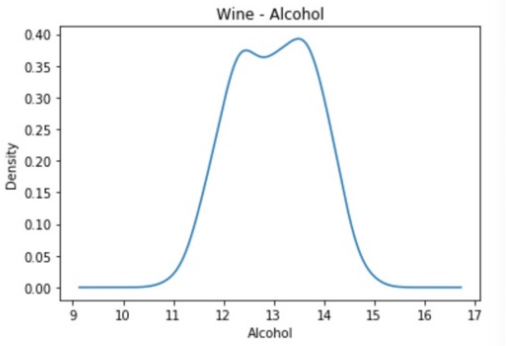
The purpose of this part is data preparation that is a process of collecting and cleaning, including missing values, typos, extra whitespaces, case sensitive and sanity checks for impossible value. The data size and data types were checked with the original data to ensure that there were no errors. As the target attribute was using 1, 2 and 3 to represent the target labels and since the rest of the attributes do not give a range of the numerical values. Thus, there were no typos or extra whitespaces, and therefore, no sanity checks were required. Finally, there is no missing value. Thus, this dataset has been cleaned and further exploration and data modeling can be processed.

Task 2: Data Exploration

This step will process dataset by generating different graphs to find a relationship with different attributes. First of all, there is a strong relationship with each numerical attribute and target attributes. The graph 1 gives information about the Flavanoids in the different types of wine, which is Type 1 has higher value - Flavanoids, compared with type 2 and type 3.



Graph 1

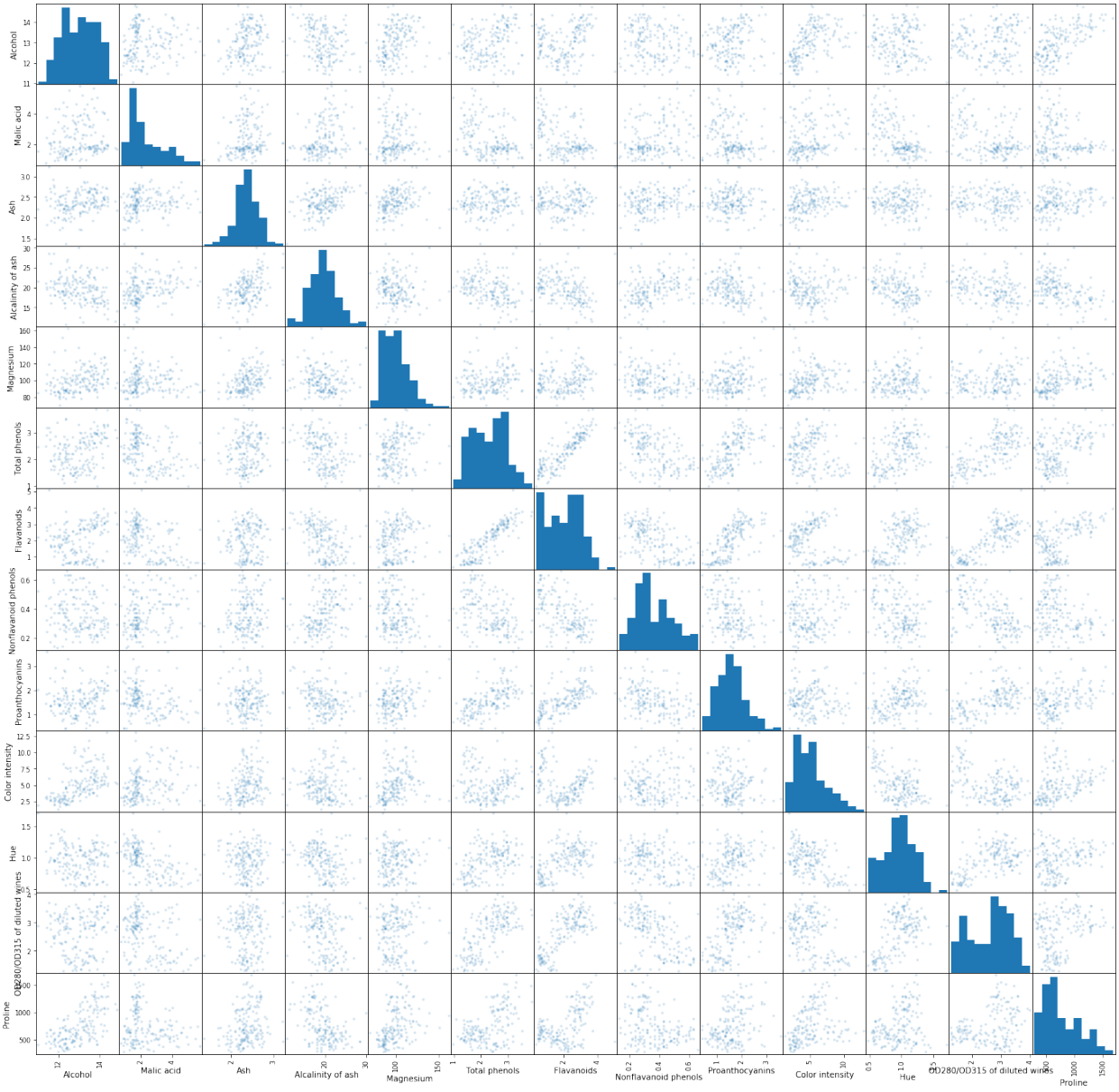


Graph 2

After that, the relationship of each column has been checked, and found that most of the attribute values are within a certain range. For example, the graph below gives information about the alcohol content of the wine and most of the wines contain an alcohol percentage between 12% and 14%. (Graph 2)

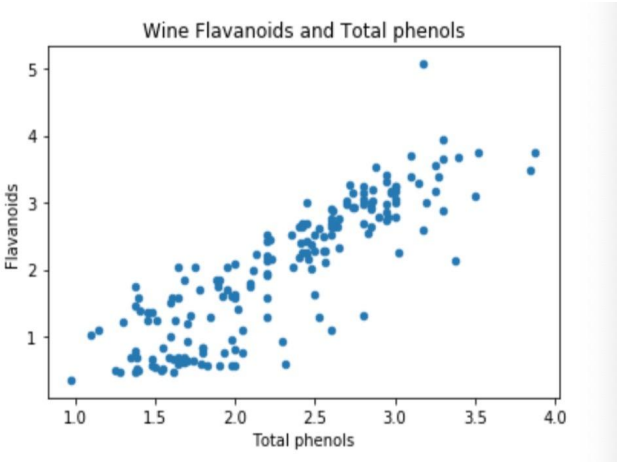
Finally, we explore the relationship between pairs of attributes using the Scatters Matrix (Graph 3). The Scatters Matrix provides the relationship between all of the attributes. For example, for the relationship of Flavanoids and Total phenols, a higher number of Flavanoids results in a higher number of total phenols, which is positive correlation.

However, for other most of attributes, there are no strong relationships between each other as shown in the following matrix graph.



Graph 3

For the relationship between Flavanoids and Total phenols, according to the research of ‘Relationship between the antioxidant properties and the phenolic and flavonoid content in traditional balsamic vinegar’, a higer Flavanoids will contribute to higher total phenols.
(Elena Verzelloni 2007)



Graph 4

3 Methodology

The dataset which describes 3 types of wine with 13 attributes of chemical elements was downloaded from the UC Irvine Machine Learning Repository. There is one column that has a categorical value and 13 columns that have numerical values. Because the goal is to classify Wine data, the K Nearest Neighbor and Decision Tree models have been used. Thus, the value of the type will be selected as a target value. After the target feature was created, the target value was deleted from the raw data. If those were not done - it could have led to data leakage.

3.1 K Nearest Neighbor

3.1.1 Tune the parameters of KNeighborsClassifier by K-Fold Cross-Validation

Firstly, K-Fold Cross-Validation will be used to select parameters for the KNN model. This algorithm will split data into the folder of K which means ‘K-1’ will be used to train the dataset and create a model, and that ‘1’ will be used to test results. At the same time, the dataset will be trained and tested by K times each process. The value of K depends on how many datasets will be tested.

For the K Neighbors classifier, there are 8 parameters that can be tuned. However, in this report, we only consider three parameters: n_neighbors, weights and power.

The parameter n_neighbors relates to the number of neighbors that have been tested near the waiting to be classified value.

Weight includes ‘uniform and distance’, ‘Uniform’ means all the targets in each neighborhood is weighted equally. ‘Distance’ means the standard of judging is to be

classified by the value of the distance. A closer value will have more contribution than neighbors which are further away.

For the Power parameter, a different integer value means a different calculation method will be used. If p equals 1, it will use Manhattan_distance to predict the value. If p equals 2, it will use Euclidean_distance. If p is greater than 2, Chebychev distance will be used.

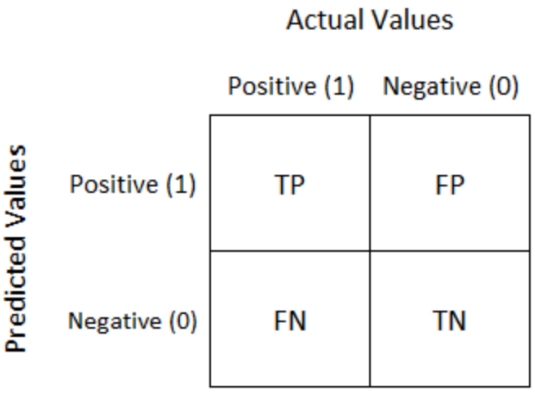
$$\left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p}$$

3.1.2 Hill climbing dataset and test

Hill climbing examines attributes one by one randomly and selects the best successor node under the evaluation function which can enhance the correct rate. This method can only select a few attributes each time which arrive with the highest score, otherwise, the rest of the columns will drop out. However, it can only calculate in the local area, which is randomly selected for the first time. This method may be run many times in order to arrive at the 'peak' correct rate.

After tuning the model, those relative attributes which got the highest results were chosen and used to create a table for further training and testing. After that, the data was split with 50% and 50%, 40% and 60%, 20% and 80% for training and testing respectively.

The results can be checked by the confusion matrix which describes the performance of a classification model and is calculated with precision, recall and f1-Score. True positive(TP) means the data point was predicted positive and it is true. True negative(TN) means the data point was predicted negative and it's true while others mean the datasets were not predicted correctly.



Graph 5

Recall: for all the positive classes, how many datasets were predicted correctly.

$$Recall = \frac{TP}{TP + FN}$$

Precision: for all the classes, how many data points have been predicted correctly.

$$Precision = \frac{TP}{TP + FP}$$

F-measure: F-score helps to measure Recall and Precision at the same time.

$$F - measure = \frac{2*Recall*Precision}{Recall + Precision}$$

3.2 Decision tree

Decision tree classifier is a machine learning algorithm, which can respectively divide datasets into many parts from the top down by choosing different conditions in each step to separate data samples in each part. The evaluation criteria make each leaf node as pure as possible. Because over 10 affecting factors will have an influence on the results, this report will only tune four different parameters to select the best score in different training and testing proportion, including criterion: min_samples_leaf, max_depth and min_samples_split. For criterion, the function to measure the quality of a split has been tested by 'gini' and 'entropy'. Min_samples_leaf means the minimum number of samples that were needed at a leaf node, which was tuned between 2 to 9. Max_depth means the maximum depth of the tree, which was tuned from 3 to 9. Min_samples_split means the minimum number of samples required to split an internal node, which was selected between 2 to 9.

4 Results

In the KNN classification, for K-fold cross-validation and as there are only 178 rows in the dataset - the dataset has been split into 5 parts. Each time, 4 parts data were fit into the train model and 1 part was used to test the results. This process was calculated five times by different train data and test data in turn. Thus, 5 scores are shown for different parameters for the K Nearest Neighbor. What is more, in order to select the best k, there was a for loop to tune k value from 3 to 10 and the higher five results were selected as the parameter of k. What is more, because the theory of the K Nearest Neighbor is to calculate how many data points are the nearest test target by default setting (weights: (default = 'uniform')), the even number of should be avoided.

After that, it also used a loop to test the score of each part with different ks, weight = 'distance' , p = 1 and p = 2. Finally, the highest score with relative parameters was chosen.

```
[fold 0] score: 0.83333 ks num: 3
[fold 1] score: 0.72222 ks num: 3
[fold 2] score: 0.55556 ks num: 3
[fold 3] score: 0.80000 ks num: 3
[fold 4] score: 0.14286 ks num: 3
--------------------
```

According to these results, k = 3, weights='distance', p = 1 have been chosen as they can generate a higher result compared with others.

After Hill Climbing, the attributes of [0,8,6,2,11,9,7,3], [9,8,0,1,6] and [11,2,9,0,10,1] were selected to create three different wine_features for the test size 0.5, 0.4 and 0.2 respectively.

After the training model, the dataset was used to try different random seeds to check the accuracy rate. In this report, for both of KNN and Decision Tree, random seeds from 1 to 10 were tested. (Table 6)

Table 6: KNN classification Correct Rate with Different Random Seeds

	random_ seed 1	random_ seed 2	random_ seed 3	random_ seed 4	random_ seed 5	random_ seed 6	random_ seed 7	random_ seed 8	random_ seed 9	random_ seed 10	attributes selected
test size = 0.5	97.00%	93.00%	91.00%	96.00%	91.00%	90.00%	99.00%	93.00%	96.00%	93.00%	0,8,6,2,11 ,9,7,3
test size = 0.4	97.00%	94.00%	90.00%	96.00%	90.00%	92.00%	99.00%	94.00%	97.00%	90.00%	9,8,0,1,6
test size = 0.2	97.00%	92.00%	83.00%	97.00%	89.00%	92.00%	100.00%	94.00%	94.00%	92.00%	11,2,9,0,1 0,1

Finally, four nested loops for four different parameters were used to check which combination will get the highest score with different training and testing data levels. The highest score and related parameters were recorded and others were ignored. The results have been presented with a confusion matrix, classification report, relative parameters and classification correct rate.

According to the results, three parameter groups have been selected for different test size. In order to make the results more universal, the relative data proportion was tested by 10 different random seeds. (Table 7)

Table 7: Decision Tree Correct Rate with Different Random Seeds

	random _seed 1	random _seed 2	random _seed 3	random _seed 4	random _seed 5	random _seed 6	random _seed 7	random _seed 8	random _seed 9	random_ seed 10	parameter selected
test size = 0.5	94.00%	87.00%	87.00%	91.00%	90.00%	89.00%	90.00%	93.00%	90.00%	83.00%	min_samples_leaf = 3, max_depth = 3, min_samples_split = 2, criterion = 'gini'
test size = 0.4	93.00%	83.00%	85.00%	89.00%	89.00%	83.00%	88.00%	92.00%	90.00%	83.00%	min_samples_leaf = 3, max_depth = 3, min_samples_split = 2, criterion = 'gini'
test size = 0.2	97.00%	89.00%	81.00%	97.00%	86.00%	86.00%	89.00%	94.00%	94.00%	89.00%	min_samples_leaf = 2, max_depth = 4, min_samples_split = 2, criterion = 'gini'

5 Discussion

Although the Decision Tree could arrive at a very high accuracy rate with a combination of a different number of parameters (99%), when the random seeds were changed, this model could not get a stable higher result. Because when this model tunes min_samples_leaf, max_depth, min_samples_split and criterion, the best result will be selected from 896 results and it may adapt to a specific situation rather than the whole dataset.

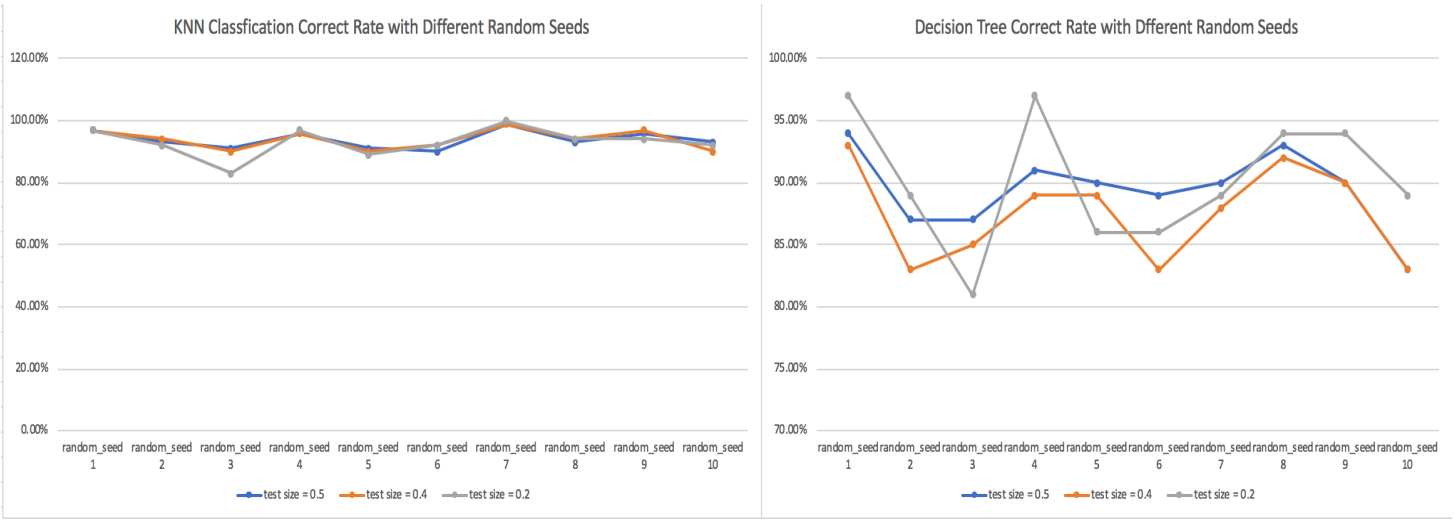


Chart 8

These line charts (Chart 8) give information about the results of different machine modelling with different random seeds. From this chart, we can see that the accuracy of the Decision Tree is fluctuating and that the correct rate is lower than KNN. Furthermore, when compared with a Decision Tree, KNN is easy to be tuned by 3 parameters, but the Decision Tree may need to consider more than 10 parameters.

In addition, overfitting may also happen in the Decision Tree and needs to be considered. Overfitting is the phenomenon in which the learning system fits the given training data so much, but it would be inaccurate in predicting the outcomes of the untrained data (cited in Ericsson 2015). Because it ends up with branches with strict rules of sparse data by 'sufficient data' or outliers, it will affect the accuracy when predicting other samples. In order to fix this problem, 'pruning' which limits the number of samples in the leaf nodes is done after the initial training is complete. 'Pruning' trims off the branches of the tree to enhance the accuracy rate (cited in Ericsson 2015).

6 Conclusion

Total phenols are highly correlated with Flavanoid content in wine and most of the attributes have a relationship with the target value. This dataset can present stable and higher results with the KNN model and do not care how many percentage of data will be trained so much. With different test sets of 50%, 40% and 20%, the accuracy rate fluctuates between 90% to 100%, which can judge which kinds of wine by chemical elements easily.

Reference List:

Elena Verzelloni a, Davide Tagliazucchi b, Angela Conte b., 2007. Relationship between the antioxidant properties and the phenolic and flavonoid content in traditional balsamic vinegar. ScienceDirect Food Chemistry, 105, 564-571.

Australia Post. 2019. Find a postcode. [ONLINE] Available at: https://auspost.com.au/postcode. [Accessed 24 May 2019]

Research Gate. 2015. overfitting in decision tree. [ONLINE] Available at: https://www.researchgate.net/post/What_is_over_fitting_in_decision_tree. [Accessed 27 May 2019].