A Comparative Study of K-Nearest Neighbour and Random Forest on Wine Quality Dataset

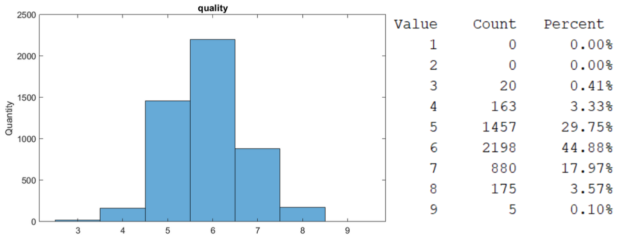
Description and Motivation of the Problem

- Compare and contrast the differences the performance of K-Nearest Neighbour and Random Forest as classification methods based on the predictions of the physicochemical data of the Portuguese "VinhoVerde" white wine.
- To determine which features are the most indicative of good quality wine and also compare it to a previous study conducted by Yeşim Er and Ayten Atasoy(2016)[1]

Exploratory Data Analysis

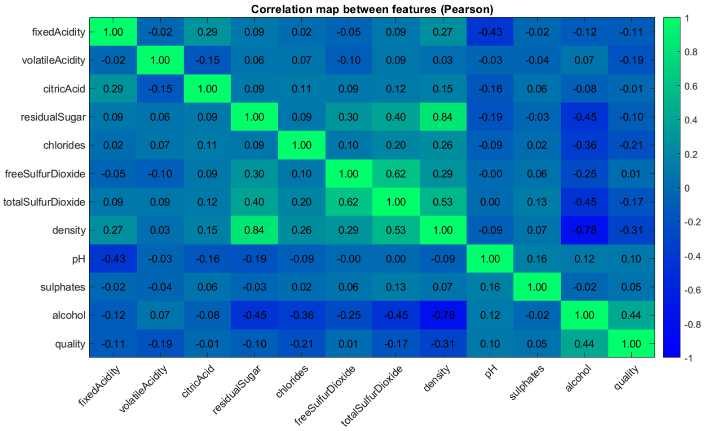
- Dataset: Wine Quality from the UCI Repository.[2]
- Contains 4898 instances out of which 75% used for training and 25% for testing.
- There are no missing values in the dataset.
- Consist of 11 predictors and a single target variable.
- The target variable has values from 3-9 on a scale from 1-10 determined by how good the wines are.
- The Pearson Correlation matrix shows a weak correlation between quality and sulphates, citricAcid, and freeSulfurDioxide as seen in Figure 2.
- Quality tends to increase as alcohol volume levels increase as well showing a strong correlation.
- By observing the correlation map, we can see potential multicollinearity happening between density and residualSugar.
- After reviewing the Variance Inflation Error, we detect the unusual activity which resulted in the drop of the density feature for better accuracy.

Figure 1: Histogram of Target and corresponding percentages



```
Value    Count    Percent
  1          0      0.00%
  2          0      0.00%
  3         20      0.41%
  4        163      3.33%
  5       1457     29.75%
  6       2198     44.88%
  7        880     17.97%
  8        175      3.57%
  9          5      0.10%
```

Figure 2: Correlation map


Correlation map between features (Pearson)

	fixedAcidity	volatileAcidity	citricAcid	residualSugar	chlorides	freeSulfurDioxide	totalSulfurDioxide	density	pH	sulphates	alcohol	quality
fixedAcidity	1.00	-0.02	0.29	0.09	0.02	-0.05	0.09	0.27	-0.43	-0.02	-0.12	-0.11
volatileAcidity	-0.02	1.00	-0.15	0.06	0.07	-0.10	0.09	0.03	-0.03	-0.04	0.07	-0.19
citricAcid	0.29	-0.15	1.00	0.09	0.11	0.09	0.12	0.15	-0.16	0.06	-0.08	-0.01
residualSugar	0.09	0.06	0.09	1.00	0.09	0.30	0.40	0.84	-0.19	-0.03	-0.45	-0.10
chlorides	0.02	0.07	0.11	0.09	1.00	0.10	0.20	0.26	-0.09	0.02	-0.36	-0.21
freeSulfurDioxide	-0.05	-0.10	0.09	0.30	0.10	1.00	0.62	0.29	-0.00	0.06	-0.25	0.01
totalSulfurDioxide	0.09	0.09	0.12	0.40	0.20	0.62	1.00	0.53	0.00	0.13	-0.45	-0.17
density	0.27	0.03	0.15	0.84	0.26	0.29	0.53	1.00	-0.09	0.07	-0.78	-0.31
pH	-0.43	-0.03	-0.16	-0.19	-0.09	-0.00	0.00	-0.09	1.00	0.16	0.12	0.10
sulphates	-0.02	-0.04	0.06	-0.03	0.02	0.06	0.13	0.07	0.16	1.00	-0.02	0.05
alcohol	-0.12	0.07	-0.08	-0.45	-0.36	-0.25	-0.45	-0.78	0.12	-0.02	1.00	0.44
quality	-0.11	-0.19	-0.01	-0.10	-0.21	0.01	-0.17	-0.31	0.10	0.05	0.44	1.00

Machine Learning Methods

K-Nearest Neighbour

- K-Nearest Neighbour, classifier also known as K-NN, is a non-parametric machine learning method which involves finding the k-nearest neighbours, with a distance calculation, of the dataset in the variable space and obtaining the class for the test data through the majority of votes.
- Non-parametric classification algorithms make the decision based on the similarities between the points to be classified, and training data.
- A relatively simple method with the most popular distance measure to be the Euclidian distance between the points.

Pros

- Simple, intuitive and easy to implement.
- Training steps are faster than other machine learning methods.
- Has no assumptions.

Cons

- Computationally heavy as it searches the nearest neighbours at the prediction stage and requires high memory to store all data points.
- Sensitive to outliers as it chooses neighbours based on distance.
- Cannot deal with missing values.

Example of a K-NN classification boundaries using a single neighbour.[3]

Figure 3: Random dataset

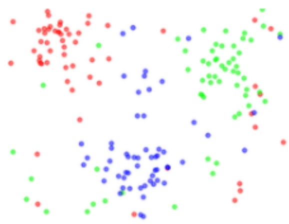
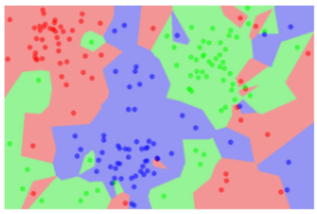


Figure 4: Classification map



Random Forest

- Random Forest is an ensemble algorithm that builds multiple decision trees, therefore a forest.
- Gives each tree a random same size subset of the features for training.
- Each tree creates class predictions and the tree with the most votes becomes the model's predictions.
- The final prediction of the random forest is made by averaging the predictions of each tree.

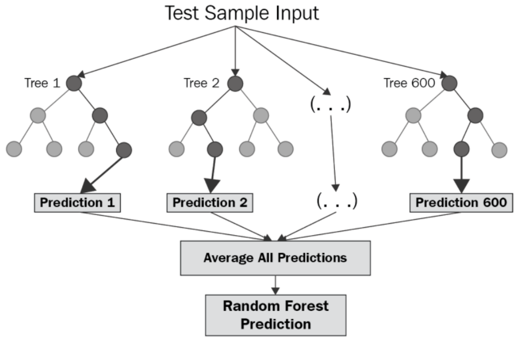
Pros

- Provides effective methods for estimating missing data.
- Reduces the variance and error and therefore improves the accuracy.
- It can manage large amounts of data.
- Robust to outliers.
- Useful for predictors importance.

Cons

- Slow to train but effective.
- Random forests can sometime overfit for some datasets with noisy classification/regression tasks.

Figure 5: Example of a Random Forest sample[4]



Hypothesis Statement

- Expect both algorithms to perform reasonably well but random forest to work better.
- We expect the K-Nearest Neighbour algorithm to have higher accuracy and more training and testing error than the random forest algorithm.
- Tuning with Hyper-parameters is going to optimise the Random Forest algorithm better and reduce mean squared error.
- We expect the Random Forest to have more training and optimisation time than the K-NN.
- By dropping the density feature, we expect an increase in accuracy.

Methodology Description

- Train all 4898 data points on both models with 75-25% train-split validation and compare training and testing errors.
- Understanding predictors importance from RF's algorithms.
- Compare Actual vs Predicted results on both models.
- Perform Hyper-parameters optimisation to find the respective optimal tuning parameters for the models and re-test it with the new values and compare differences on accuracy and error.
- Test with simple method algorithms and experiment with optimisation parameters.

Implementation of the Machine Learning methods

K-Nearest Neighbour

- Used standard 1 number of neighbours with Euclidean distance as the starting model.
- As expected, accuracy and error were relatively high.
- Optimisation showed a slight increase in accuracy and a slight decrease in the test accuracy compared to the previous one.
- Experiment on the density feature did not show any impact on the model except for a slight increase in the test error.
- The optimum value was consistently shown to be the cosine distance the best suitable calculation for the measure with a slight difference on the neighbours.

Random Forest

- Random Forest was trained with the RUSBoost model at first and then optimised with the bag model, which showed a drastic increase in the accuracy.
- The most important predictor is the fixedAcidity, as shown in figure 6.
- Optimum values of parameters were consistently shown to be around 11 learning cycles and two features per node using the 'bag' method.
- Dropping the density feature resulted in a 10% increase in accuracy.
- Figure 7 shows a drastically but stable decrease in the loss error in the optimised model compared with the previous one.

Accuracy

K-NN		Random Forest
0.646%	**First**	0.327%
0.689%	**Optimized**	0.656%

Figure 6: Predictors Importance

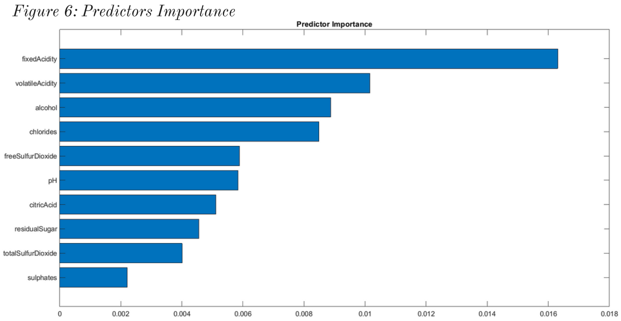
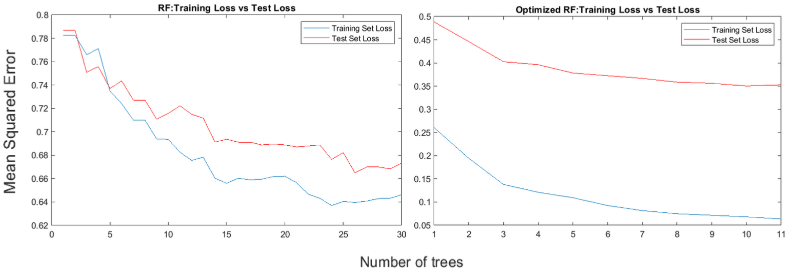


Figure 7: Cumulative train and test loss of Random Forest



Analysis and Critical Evaluation of Results

- K-NN was fast as expected with relatively good accuracy but struggled with the large dataset resulting in a high test loss error(68%).
- Experimenting with different k-values and different distances, such as the 'Jaccard' and 'hamming' did not seem to affect the classifier strongly.
- In classification models, the key attribute that we consider is the test error rate which RF outperformed the K-NN. This is not surprising since it is widely known that RF is a powerful machine learning method and would usually outperform its simpler counterparts. Its also shown in the study conducted by Yeşim Er and Ayten Atasoy(2016) where it outperformed support vector machine as well.
- In terms of training time, K-NN was nearly ten times faster and proportionally more effective rather than the RF. We expected this difference to be, due to the simplicity of the K-NN classifier.
- RF could be described as a powerful machine learning method with the ability to manipulate the variance trade-off to produce healthy and effective results. This is achieved using various techniques such as bagging and random feature selection, which seek to scale better as training time increases. In our results, it is shown by an appreciable test/train loss error difference.
- Results showed that RF could be more computationally involved concerning the training time than the K-NN classifier. However, the dataset was not a challenge for the RF as it was able to calculate it relatively fast. This would not be the case for larger datasets.
- We also repeated each experiment several times with different random seed, observing only very minimal changes.
- In conclusion, an excellent pre-processing phase and with the appropriate choice of predictors for the data considered seems to matter as much as the classification algorithm employed.

Lessons Learned and Future Work

- Optimising K-NN and RF both involved data manipulation. For the K-NN, few hyper-parameters could be tuned to result in a significant change. On the other hand, RF could take its time and learn more by itself from hyper-parameter optimisation.
- Future work on K-Nearest Neighbour includes trying different methods and reducing the features, which may lead to substantial improvements.
- Future work on Random Forest includes investigating several number evaluations for the hyper-parameter optimisation, which can yield to a significant improvement in the performance. Also, more works remains in deeper understanding of why RF works so well.
- Experiment with other classification and regression machine learning methods to find the optimum technique that will produce better results.

References

1. Yeşim Er and Ayten Atasoy, 2016, The Classification of White Wine and Red Wine According to Their Physicochemical Qualities, International Journal of Intelligent Systems and Applications in Engineering 4(Special Issue-1):23-23, Available at: https://www.researchgate.net/publication/311919082_The_Classification_of_White_Wine_and_Red_Wine_According_to_Their_Physicochemical_Qualities, [22 November 2020]
2. UCI Machine Learning Repository: Wine Quality Data Set, 2020, UCI Machine Learning Repository: Wine Quality Data Set, Available at: https://archive.ics.uci.edu/ml/datasets/wine+quality. [15 November 2020].
3. Wikipedia, 2015, k-nearest neighbours algorithm, Available at: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
4. O'Reilly Online Learning, 2020, Random forests - TensorFlow Machine Learning Projects [Book]. Available at: https://www.oreilly.com/library/view/tensorflow-machine-learning/9781789132212/d3d388ea-3e0b-4095-b01e-a0fe8cb3e575.xhtml, [22 November 2020].
5. Jaime S. Cardoso, Joaquim F. Pinto da Costa, 2007, Learning to Classify Ordinal Data: The Data Replication Method Journal of Machine Learning Research 8 1393-1429
6. P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis.Modeling , 2009, wine preferences by data mining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4):547-553
7. Okun, O. and Valentini, G., 2008. Supervised And Unsupervised Ensemble Methods And Their Applications. Berlin: Springer.
8. Michael Paluszek and Stephanie Thomas, 2016, MATLAB Machine Learning, Apress; 1st ed. edition.
9. MathWorks, 2020, Classification Ensembles, Available at: http://se.mathworks.com/help/stats/classification-ensembles.html
10. MathWorks, 2020, ClassificationKNN, Available at: http://se.mathworks.com/help/stats/classificationknn.html