**Project Summary – Wine Quality**

**Introduction:**

**Dataset:** Wine Quality based on physiochemical tests - <https://archive.ics.uci.edu/ml/datasets/Wine+Quality>. The dataset is related to red and white wine samples of the Portuguese "Vinho Verde" wine. The main dataset includes 2 smaller datasets – one for red-wine samples and one for white-wine samples. This project uses the red-wine samples dataset alone, containing 1599 instances, 11 input variables and 1 output variable.

Values for input variables are obtained from objective tests. Values for output variable is based on sensory data. Values are calculated as median of at least 3 ratings from wine-experts.

**Input variables** (based on physicochemical tests):

1 - fixed acidity 2 - volatile acidity 3 - citric acid 4 - residual sugar

5 – chlorides 6 - free sulfur dioxide 7 - total sulfur dioxide 8 - density

9 – pH 10 – sulphates 11 - alcohol

**Output variable** (based on sensory data):

12 - quality (score between 0 and 10)

**Sources:** P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis.

Modeling wine preferences by data mining from physicochemical properties.

In Decision Support Systems, Elsevier, 47(4):547-553. ISSN: 0167-9236.

Available at: [@Elsevier] http://dx.doi.org/10.1016/j.dss.2009.05.016

[Pre-press (pdf)] http://www3.dsi.uminho.pt/pcortez/winequality09.pdf

[bib] <http://www3.dsi.uminho.pt/pcortez/dss09.bib>

**Goal of the project: P**redict the output/target variable - wine-quality(range 3-8). Also test feature-selection methods to analyze how the different feature combinations perform.

**What have I done?**

1. After loading the data from csv, created a correlation matrix to visually analyze how the features are correlated. Here are some basic observations –

* **which feature has the highest correlation with the quality?**

'alcohol' has the highest correlation with 'quality'

* **which feature has the least correlation with quality?**

'volatile\_acidity' has the least correlation with 'quality'

1. The next step was to **choose which feature-selection methods** to apply. There are 3 categories of feature-selection methods commonly used – Our aim was to test one method from each category and compare the results.

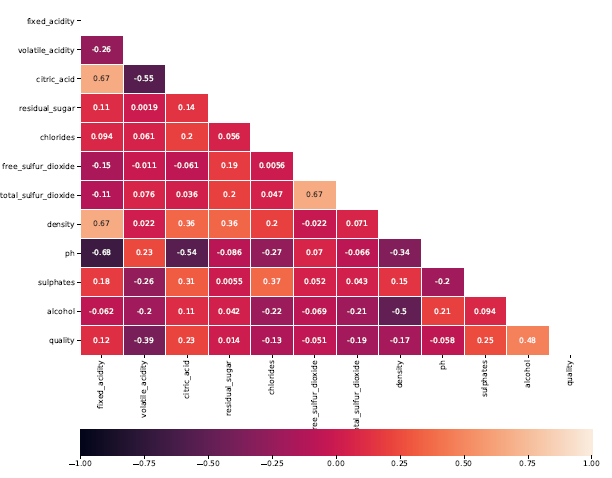
* **Filter method**: this approach relies on the characteristics of the features and the features are selected independent of any machine-learning algorithm. We have used **Pearson Correlation** from this category.
* **Wrapper method**: this approach searches for features that are best suited and uses a machine-learning algorithm to evaluate performance. We have used **Recursive Feature Elimination** from this category.
* **Embedded method**: this is an iterative approach that extracts features that contribute the most in a particular iteration. We have used **LassoCV** from this category.

1. The next step was to **choose the regression model** to test these selection methods. I was comparing using **LinearRegression** (simple and easy to interpret) and **RandomForest**(robust, better output owing to ensemble approach, but hard to interpret). I applied both models to each of the selection-methods and decided to use RandomForest since it had better accuracy(66% compared to 30%).

Attached “linear\_regression\_results.csv” & “random\_forest\_results.csv”

1. Apply feature-selection methods and compile performance metrics

* **Pearson Correlation**



Focus on the correlation of various input attributes with the target attribute – quality.

Set the **correlation threshold** to **0.15** – which gave 6 attributes (tested correlation-threshold 0.1 which gave 8 attributes and lower accuracy)

**Output:-**

Printing features identified using Pearson Correlation

volatile\_acidity 0.390558

citric\_acid 0.226373

total\_sulfur\_dioxide 0.185100

density 0.174919

sulphates 0.251397

alcohol 0.476166

Applying random-forest regressor

**r2** using RandomForest: 0.2352

**rmse** using RandomForest: 0.6847

**Accuracy** using RandomForest: **0.6388**

* **Recursive Feature Elimination (RFE)**

**RFE vs RFECV** – RFECV resulted in much lower accuracy compared to RFE using ‘n\_features\_to\_select’ = 5,6,7 & 8. Hence chose RFE over RFECV.

Among different values of ‘n\_features\_to\_select’, accuracy was highest at n=7.

**Output:-**

Applying Recursive Feature Elimination with DecisionTrees

Number of features selected: 7

Feature Ranking: [4 1 2 1 3 5 1 1 1 1 1]

List of features selected using RFE-

['volatile\_acidity', 'residual\_sugar', 'total\_sulfur\_dioxide', 'density', 'ph', 'sulphates', 'alcohol']

Applying random-forest regressor

r2 using RandomForest: 0.2209

**rmse** using RandomForest: 0.691

**Accuracy** using RandomForest: **0.6512**

* **LassoCV**

Default cross-validation cv=5 is used.

**Output:-**

Applying the Lasso(L1 penalty) method for feature-selection

Number of features selected: 10

Printing list of features selected using Lasso(L1 regularization) method -

['fixed\_acidity', 'volatile\_acidity', 'citric\_acid', 'residual\_sugar', 'chlorides', 'free\_sulfur\_dioxide', 'total\_sulfur\_dioxide', 'ph', 'sulphates', 'alcohol']

Applying random-forest regressor

**r2** using RandomForest: 0.2311

**rmse** using RandomForest: 0.6865

**Accuracy** using RandomForest: **0.6462**

* **Shapley Method vs All features**

**Output:-**

**Dropped feature: fixed\_acidity** - Invoking RandomForest for regression

Applying random-forest regressor

r2 using RandomForest: 0.2434

rmse using RandomForest: 0.681

Accuracy using RandomForest: **0.6538**

**Dropped feature: volatile\_acidity** - Invoking RandomForest for regression

Applying random-forest regressor

r2 using RandomForest: 0.2291

rmse using RandomForest: 0.6874

Accuracy using RandomForest: **0.645**

**Dropped feature: citric\_acid** - Invoking RandomForest for regression

Applying random-forest regressor

r2 using RandomForest: 0.2597

rmse using RandomForest: 0.6736

Accuracy using RandomForest: **0.6612**

**Dropped feature: residual\_sugar** - Invoking RandomForest for regression

Applying random-forest regressor

r2 using RandomForest: 0.2393

rmse using RandomForest: **0.6828**

Accuracy using RandomForest: **0.645**

**Dropped feature: chlorides** - Invoking RandomForest for regression

Applying random-forest regressor

r2 using RandomForest: 0.2597

rmse using RandomForest: 0.6736

Accuracy using RandomForest: **0.65**

**Dropped feature: free\_sulfur\_dioxide** - Invoking RandomForest for regression

Applying random-forest regressor

r2 using RandomForest: 0.2311

rmse using RandomForest: 0.6865

Accuracy using RandomForest: **0.6425**

**Dropped feature: total\_sulfur\_dioxide** - Invoking RandomForest for regression

Applying random-forest regressor

r2 using RandomForest: 0.2311

rmse using RandomForest: 0.6865

Accuracy using RandomForest: **0.6388**

**Dropped feature: density** - Invoking RandomForest for regression

Applying random-forest regressor

r2 using RandomForest: 0.2311

rmse using RandomForest: 0.6865

Accuracy using RandomForest: **0.6462**

**Dropped feature: ph** - Invoking RandomForest for regression

Applying random-forest regressor

r2 using RandomForest: 0.2637

rmse using RandomForest: 0.6718

Accuracy using RandomForest: **0.655**

**Dropped feature: sulphates** - Invoking RandomForest for regression

Applying random-forest regressor

r2 using RandomForest: 0.1985

rmse using RandomForest: 0.7009

Accuracy using RandomForest: **0.6412**

**Dropped feature: alcohol** - Invoking RandomForest for regression

Applying random-forest regressor

r2 using RandomForest: 0.2189

rmse using RandomForest: 0.6919

Accuracy using RandomForest: **0.6275**

* **Compute Accuracy using all 11 features**

**Output:-**

Using all 11 features to compare accuracy of prediction

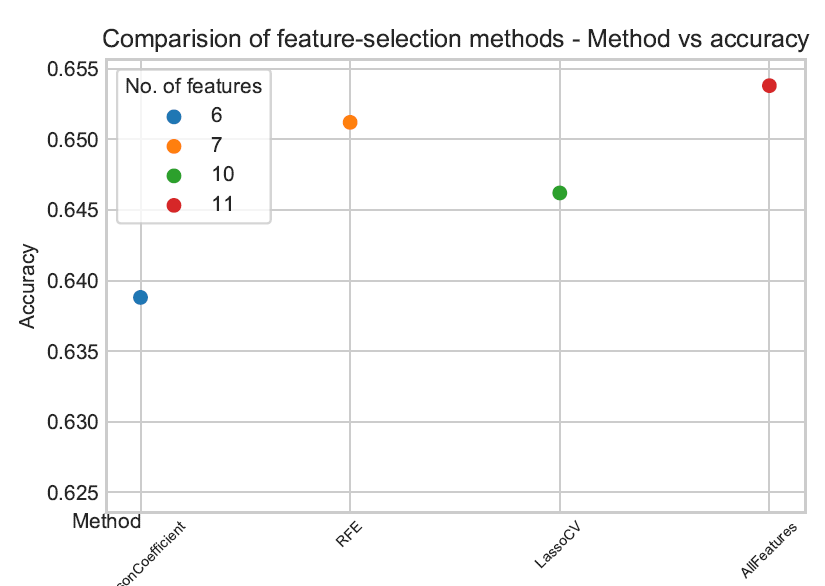
Applying random-forest regressor

r2 using RandomForest: 0.2434

rmse using RandomForest: 0.681

**Accuracy** using RandomForest: **0.6538**

1. Compile accuracy vs method

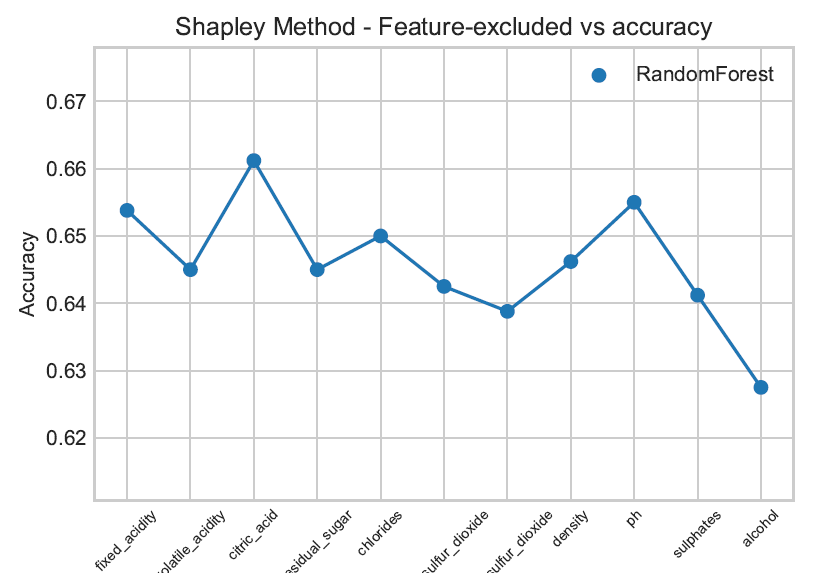


**Observations:-**

**Pearson Coefficient** used 6 features and gave the least accuracy.

**Recursive Feature Elimination** used 7 features and resulted in about the same accuracy(**65.12%**) as using all 11 features(**65.38%**).

1. Compile results of Shapley method vs All-features



**Observations:-**

* Excluding the feature “citric\_acid” gives higher accuracy when compared to using all 11 features.
* Excluding the feature “ph” gives same accuracy as using all 11 features.
* As noted in the correlation-matrix earlier, “alcohol” has the highest correlation to the quality and excluding this resulted in accuracy drop by ~2%

**Summary of results:-**

* Using RFE with following 7 features gives about same accuracy (65.12%) of prediction as all 11 features –

['volatile\_acidity', 'residual\_sugar', 'total\_sulfur\_dioxide', 'density', 'ph', 'sulphates', 'alcohol’]

* Excluding the feature “ph” gives same accuracy as using all 11 features – 65.38%
* Excluding the feature “citric\_acid” gives higher accuracy when compared to using all 11 features – 66.12%

**Project Improvements:-**

1. Use white-wine dataset: Implement the same model on the white-wine dataset to compare outcome.
2. Improve Accuracy of prediction
   * Hyperparameter tuning: Test different combinations of N (no. of trees) & d (max-depth) – RandomForest implementation.
   * Models: Explore other regression models like Support Vector Regression, KNN Regressors to determine the right model that fits this dataset.
3. Outlier Detection: Implement models to detect few excellent/poor wines based on the wine-quality.

**Code Details:-**

**Wine\_quality\_main.py** – main script to be executed – invokes all the methods sequentially. Also invokes the RFE implementation from **apply\_rfe.py**

**Name of the dataset file:** winequality-red.csv

**List of methods -**

apply\_pearson\_correlation() – creates the correlation matrix and selects features based on the correlation-threshold specified.

apply\_rfe() – applies the recursive-feature-elimination using decision-trees to select features.

apply\_lasso() – applies LassoCV to select features.

apply\_random\_forest() – applies the random-forest regressor to predict the wine-quality using the feature-list provided. Computes r2, rmse and accuracy metrics.

plot\_results() – plots accuracy vs method for pearson-correlation vs RFE vs LassoCV vs All-features

plot\_shply() – plots accuracy vs feature-excluded for Shapley-method vs All-features