



De La Salle University – Manila

Gokongwei College of Engineering

Electronics and Communications Engineering Department

Intelligent Systems for Engineering

LBYEC3B

Project 1 Documentation

Wine Quality Classification

by

BAYETA IV, Reginald Geoffrey L. – 11811269

TAN, Adrienne Rafaela P. – 11809736

TUPAL, Isaiah Jassen L. – 11847115

LBYEC3B – EQ4

Project 1: Wine Quality Classification

I. Statistical Analysis

Bias: Training data columns *wineType* and *id* were removed.

Reasons:

- Parameter *id* is just for identification or index for each data sample.
- For *wineType*, there has been a previous attempt to separate data (red and white wine) and used it for training neural networks but outputs low accuracy.
- In addition, oversampling was done to both red and white wine train data which outputs high accuracy prediction as per the models, but low on actual accuracy as seen in Figure 1.
- To produce the outputs, both training data are being fed to a `fitnet()` model with hidden layers of [9 5 5 2] then the combined predicted results for both types are exported as `.csv` file (Refer to Appendix A).

neuralsubmissionI9552.csv 2 days ago by Ding Bayeta IV add submission details	0.46536	<input type="checkbox"/>
neuralsubmissionH9552.csv 2 days ago by Ding Bayeta IV add submission details	0.48691	<input type="checkbox"/>
neuralsubmissionF.csv 2 days ago by Ding Bayeta IV 50 25 10 trianlm	0.45202	<input type="checkbox"/>
neuralsubmissionD.csv 2 days ago by Ding Bayeta IV not normalized sadness	0.44381	<input type="checkbox"/>
neuralsubmissionB.csv 2 days ago by Ding Bayeta IV neural matlab code version 2	0.39353	<input type="checkbox"/>

Figure 1. Outputs for separated training parameters (by *wineType*) using *fitnet*

From the aforementioned, we now have 12 parameters remaining: 11 features and 1 target (quality). To know more about the data, we generate a histogram and summary for each parameter as seen in Figure 2 and Table 1.

Table 1. Wine Train Summary

index	Parameters	Min	Median	Max
1	fixedAcidity	4.2	7.1	15.9
2	volatileAcidity	0.08	0.29	1.58
3	citricAcid	0	0.32	1.66
4	residualSugar	0.6	2.9	65.8
5	chlorides	0.012	0.047	0.611

6	freeSulfurDioxide	1	29	146.5
7	totalSulfurDioxide	6	122	366.5
8	density	0.98713	0.9952	1.039
9	pH	2.72	3.21	3.90
10	sulphates	0.22	0.51	2
11	alcohol	8	10.1	14.9
12	quality	3	6	9

By inspection from Table 1, it can be observed that there are outliers present in the data as seen in the disparity of the min and max of multiple table variables such as index 4, 6, 7, and 10. We can also see that there are parameters that fall within a small range such as in index 9, *pH*, as wines are usually on the acidic side of the spectrum of pH. We can confirm these observations by looking at Figure 2.

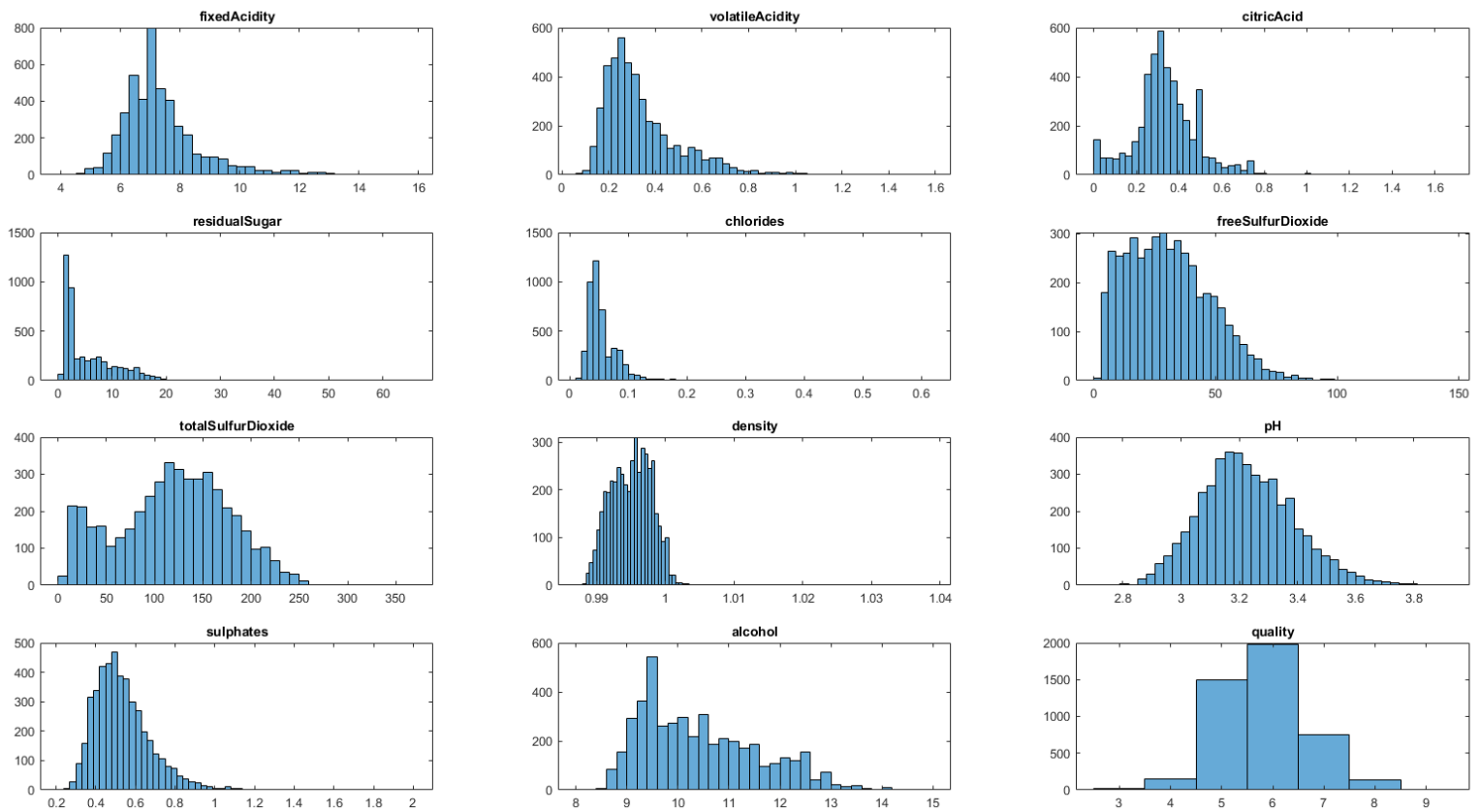


Figure 2. Wine Train Data Histograms

It should be evident that descriptive observations are subject to biases, and to combat this, we added another layer of proof for statistical observation: a correlation matrix (*corrplot()*) was generated that shows the correlations among pairs of variables in the training data as seen in Figure 3. In this correlation matrix, the row to focus would be *quality* vs other parameters which is summarized in Table 2. From Table 2, it is

apparent that the parameters with outliers such as index 4, 6, 7, and 10 are found to be in the middle in the intuitive ranking, which denotes little to no correlation compared to the outer extremes (-/+).

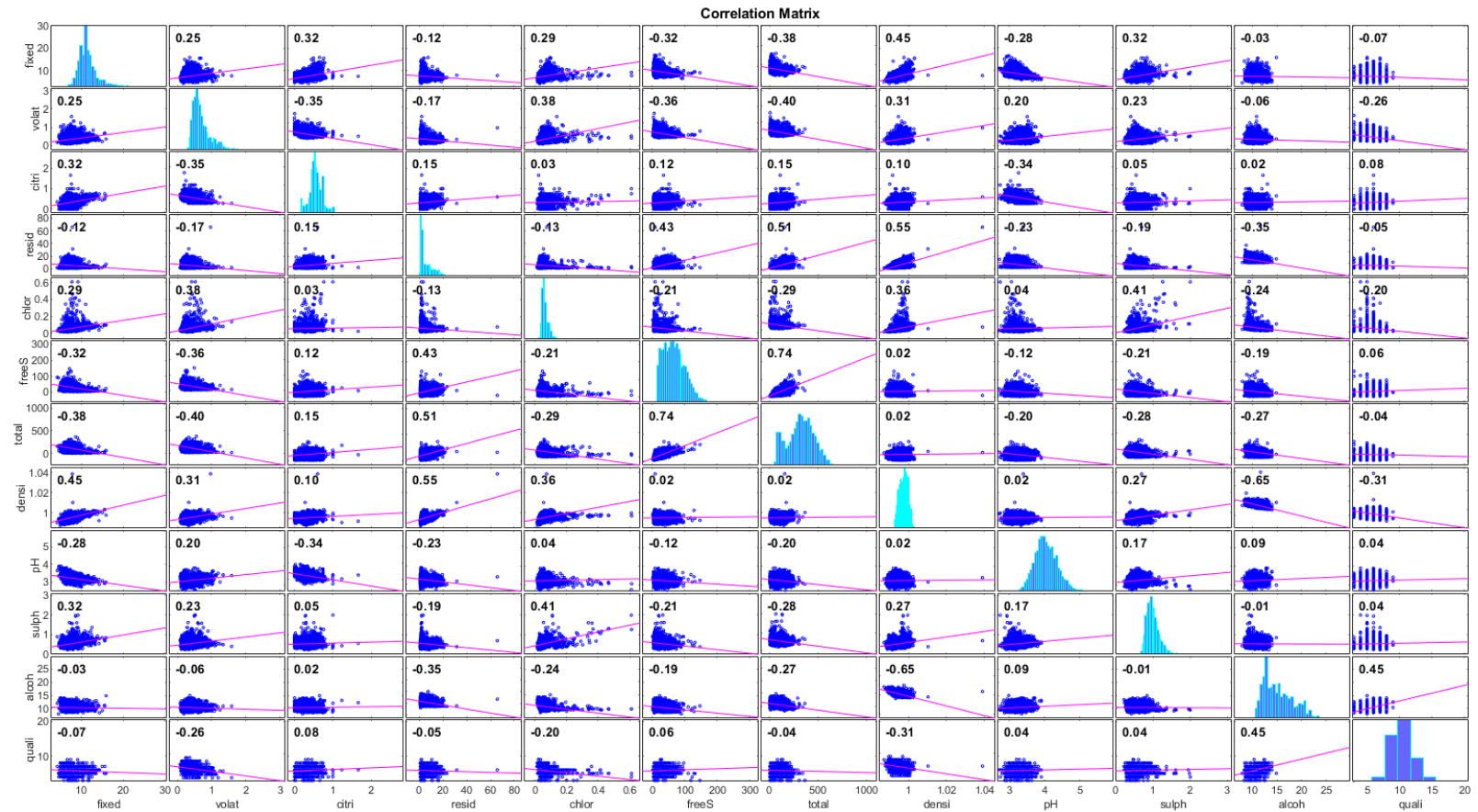


Figure 3. Wine Train Correlation Matrix

Table 2. Wine Train Summary by Correlation Result (with respect to quality)

Index	Parameters	Min	Median	Max	Correlation Score
8	density	0.98713	0.9952	1.039	-0.31
2	volatileAcidity	0.08	0.29	1.58	-0.26
5	chlorides	0.012	0.047	0.611	-0.20
1	fixedAcidity	4.2	7.1	15.9	-0.07
4	residualSugar	0.6	2.9	65.8	-0.05
7	totalSulfurDioxide	6	122	366.5	-0.04
9	pH	2.72	3.21	3.90	+0.04
10	sulphates	0.22	0.51	2	+0.04

6	freeSulfurDioxide	1	29	146.5	+0.06
3	citricAcid	0	0.32	1.66	+0.08
11	alcohol	8	10.1	14.9	+0.45

II. Preprocessing

The preprocessing step comprises of two parts: feature ranking and normalization of new training and testing data from selected ranked features. In addition, the results of the statistical analysis and feature ranking will be discussed.

a. Selecting feature ranking function

Upon checking MATLAB documentation, two main feature rankings were explored in this project: `fscmr` (“Rank features for classification using minimum redundancy maximum relevance (MRMR) algorithm”) and `fscchi2` (“Univariate feature ranking for classification using chi-square tests”). The latter was used as it performed better in previous submission attempts in terms of accuracy as seen in Figure 5, which outputs a high accuracy of 0.53976 compared to all samples shown in Figures 4 and 5.

lastsubmission14.csv 12 hours ago by Ding Bayeta IV idx1 8, same as before	0.53771	<input type="checkbox"/>
lastsubmission13.csv 12 hours ago by Ding Bayeta IV 675, idx1 1-9, same as b4	0.53309	<input type="checkbox"/>

Figure 4. Accuracy utilizing `fscmr`()

NB. idx1 refers to the index list of the feature ranking generated by `fscmr` lastsubmission14 used Top 8 selected ranked features, while lastsubmission13 used Top 9.

lastsubmission11.csv 19 hours ago by Ding Bayeta IV bg idx2 1-7 655 same param with others	0.52385	<input type="checkbox"/>
lastsubmission10.csv 19 hours ago by Ding Bayeta IV bg671, idx2 1-9	0.53771	<input type="checkbox"/>
lastsubmission9.csv 20 hours ago by Ding Bayeta IV bg 66 idx2 1-8	0.53976	<input type="checkbox"/>

Figure 5. Accuracy utilizing `fscchi2`()

NB. idx2 refers to the index list of the feature ranking generated by `fscchi2`() lastsubmission11 → Top 7, lastsubmission9 → Top 8, lastsubmission10 → Top 9 selected ranked features.

b. Generating feature ranking of training data by using fscchi2

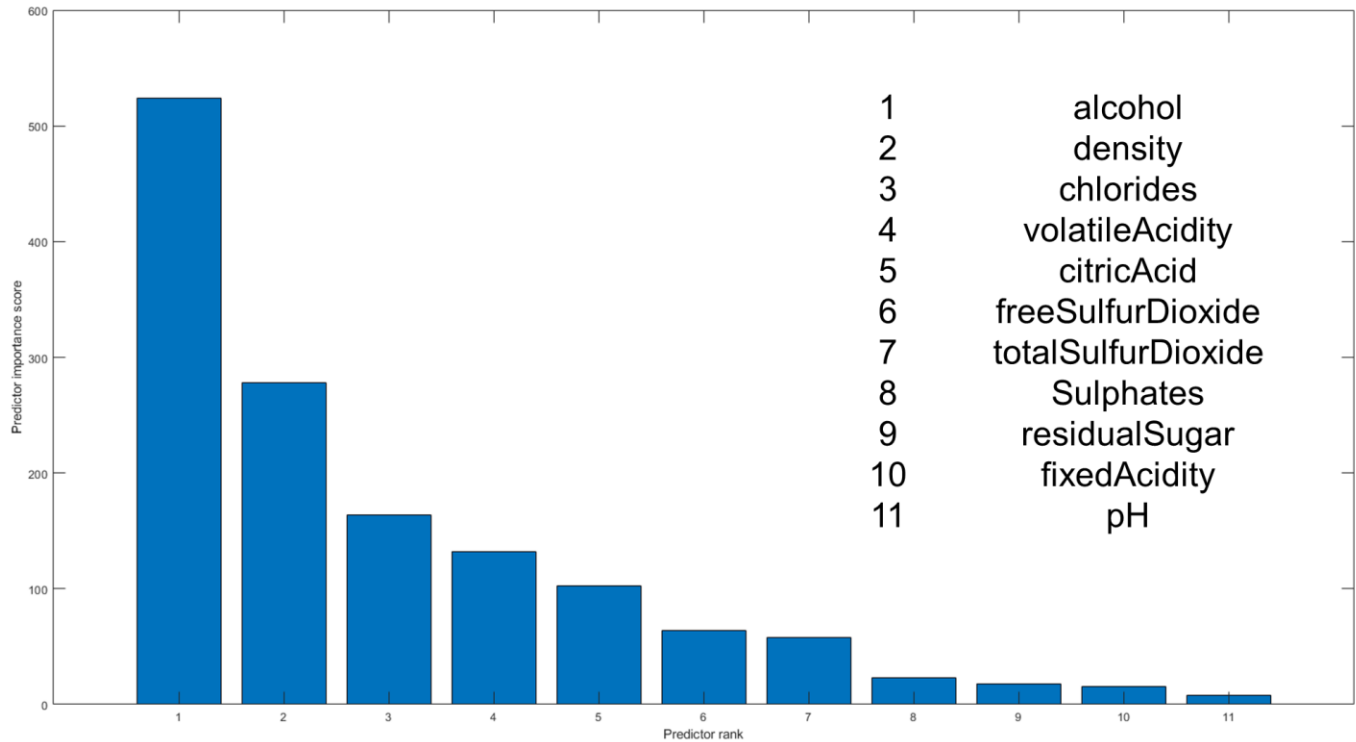


Figure 6. Predictor rank vs Predictor importance score for fscchi2

When calling fscchi2, it returns two parameters, *idx2* and *scores2* (variables used in code) which represent the column index sorted by top feature rankings and their corresponding scores. As per the documentation of fscchi2, a higher score indicates that its corresponding predictor variable is important relative to the target parameter. Table 3 highlights the top column features and corresponding scores, which is also reflected in Figure 6. In addition, the returned score matches with the correlation score as well as the discussed descriptive observations.

Table 3. fscchi2 Top column features and corresponding scores

Ranking	Column Index	Feature	Score	Correlation Score
1	11	alcohol	524.07	+0.45
2	8	density	278.25	-0.31
3	5	chlorides	163.76	-0.20
4	2	volatileAcidity	132.27	-0.26
5	3	citricAcid	102.38	+0.08
6	6	freeSulfurDioxide	63.745	+0.06
7	7	totalSulfurDioxide	57.681	-0.04
8	10	sulphates	23.13	+0.04

9	4	residualSugar	18.091	-0.05
10	1	fixedAcidity	15.259	-0.07
11	9	pH	7.7558	+0.04

c. Generating normalized new training and testing data from selected ranked features

The selected range for the top-ranked features to be used in the final training and testing data to be normalized is the Top 8 with the results shown in Figure 5 as the basis. Using Top 8 features outperforms Top 7 and Top 9 in terms of accuracy.

```
% Normalize train and test data
norm_train = normalize(wine_train);
norm_test  = normalize(wine_test);
% Select ranked features
mask = idx2(1:8)
wine_feature = norm_train(:, mask);
wine_target  = double(qlty);
wine_testing = norm_test(:, mask);
```

III. Models and Training

1 Ensemble Last change: Bagged Trees Accuracy (Validation): 66.9% 8/8 features	11 Neural Network Last change: Trilayered Neural Network Accuracy (Validation): 54.1% 8/8 features
2 Ensemble Last change: 'Number of learners' = '100' Accuracy (Validation): 67.7% 8/8 features	12 Neural Network Last change: Medium Neural Network Accuracy (Validation): 55.3% 8/8 features
3 KNN Last change: Weighted KNN Accuracy (Validation): 65.4% 8/8 features	13 SVM Last change: Coarse Gaussian SVM Accuracy (Validation): 52.4% 8/8 features
4 SVM Last change: Fine Gaussian SVM Accuracy (Validation): 62.8% 8/8 features	14 Ensemble Last change: 'Number of learners' = '250' Accuracy (Validation): 67.9% 8/8 features
5 Tree Last change: Fine Tree Accuracy (Validation): 52.8% 8/8 features	15 KNN Last change: Coarse KNN Accuracy (Validation): 53.3% 8/8 features
7 KNN Last change: Cubic KNN Accuracy (Validation): 54.5% 8/8 features	18 SVM Last change: Quadratic SVM Canceled 8/8 features
8 Ensemble Last change: RUSBoosted Trees Accuracy (Validation): 32.7% 8/8 features	19 Ensemble Last change: 'Number of learners' = '68' Accuracy (Validation): 67.3% 8/8 features
9 Ensemble Last change: Boosted Trees Accuracy (Validation): 53.2% 8/8 features	20 Ensemble Last change: 'Number of learners' = '500' Accuracy (Validation): 68.3% 8/8 features
10 KNN Last change: 'Number of neighbors' = '68' Accuracy (Validation): 68.0% 8/8 features	21 Ensemble Last change: 'Number of learners' = '1000' Accuracy (Validation): 68.0% 8/8 features
11 Neural Network Last change: Trilayered Neural Network Accuracy (Validation): 54.1% 8/8 features	

Figure 7. Classification Learner models used for training and testing

IV. Testing Results

a. Accuracy of models in validation vs official submission

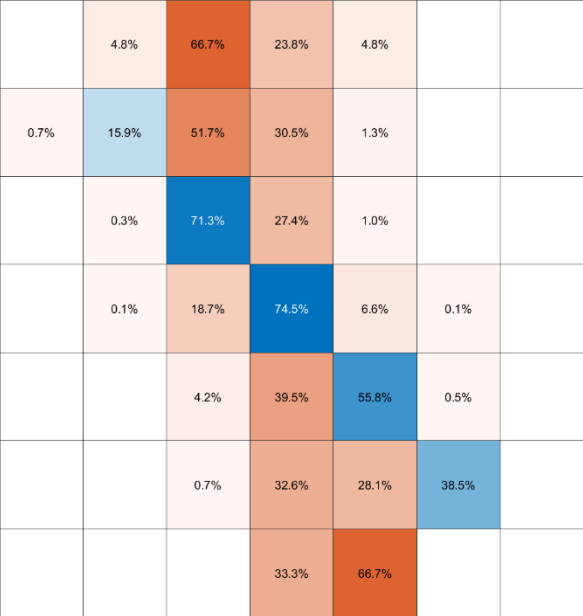
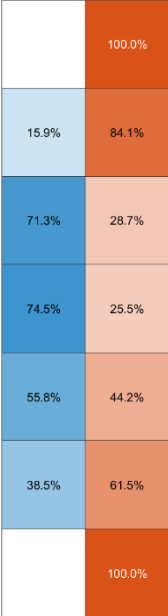
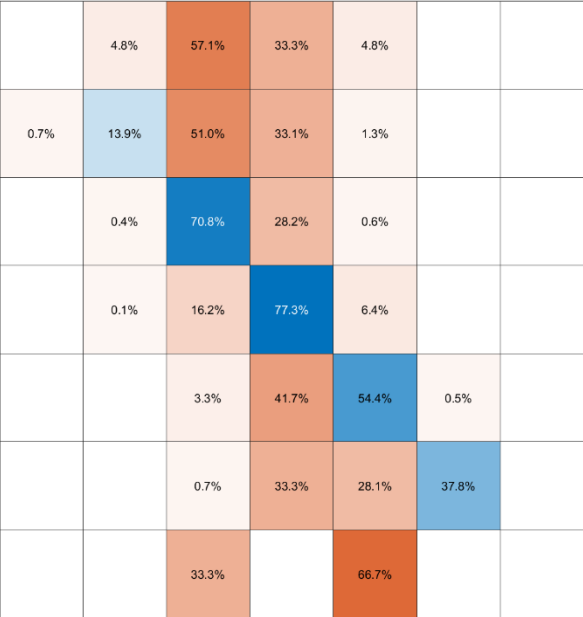
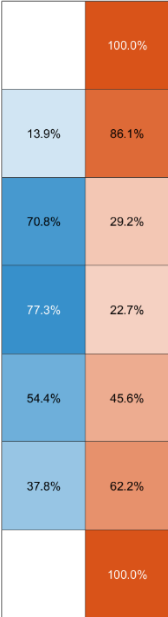
As shown in Table 4, Bagged Trees, or also referred to as the Random Forest model has garnered the best accuracy out of 17 instances of models trained. Bagged Trees with 500 learners outclassed the other Bagged Trees instances with a 0.56695 accuracy score.

Table 4. *Models' validation vs official submission accuracy*

Index	Model Used	Validation Accuracy (VA)	Actual Accuracy
1	Bagged Trees (default)	0.669	0.53617
2	Bagged Trees (learners = 100)	0.677	0.55720
3	Weighted KNN (default)	0.654	0.47870
4	Fine Gaussian SVM (default)	0.628	0.48435
5	Fine Tree (default)	0.528	0.50846
6	Cubic KNN (default)	0.545	0.50384
7	RUS Boosted Trees (default)	0.327	0.30938
8	Boosted Trees (default)	0.532	0.54232
9	Weighted KNN (k = 68)	0.680	0.53873
10	Trilayered Neural Net (default)	0.541	0.53771
11	Medium Neural Net (default)	0.553	0.52591
12	Coarse Gaussian SVM (default)	0.524	0.53976
13	Bagged Trees (learners = 250)	0.679	0.55823
14	Coarse KNN (default)	0.533	0.53822
15	Bagged Trees (learners = 68)	0.673	0.55053 (Rank 3)
16	Bagged Trees (learners = 500)	0.683	0.56695 (Rank 1)
17	Bagged Trees (learners = 1000)	0.680	0.56541 (Rank 2)

b. Confusion Matrices of models

Table 5. *Trained models’ validation confusion matrices*

Index	Model Used	VA	Validation Confusion Matrices									
1	Bagged Trees (default)	0.669	Model 1									
												
			3					4				
			4					5				
			5					6				
			6					7				
			7					8				
			8					9				
			9					10				
			10					11				
2	Bagged Trees (learners = 100)	0.677	Model 2									
												
			3					4				
			4					5				
			5					6				
			6					7				
			7					8				
			8					9				
			9					10				
			10					11				

3	Weighted KNN (default)	0.654	Model 3																																																																																																						
			<table><tr><td>3</td><td></td><td>4.8%</td><td>57.1%</td><td>33.3%</td><td>4.8%</td><td></td><td></td></tr><tr><td>4</td><td>0.7%</td><td>16.6%</td><td>50.3%</td><td>29.1%</td><td>3.3%</td><td></td><td></td></tr><tr><td>5</td><td></td><td>0.7%</td><td>68.0%</td><td>29.1%</td><td>2.1%</td><td>0.1%</td><td></td></tr><tr><td>6</td><td></td><td>0.4%</td><td>17.3%</td><td>72.4%</td><td>9.4%</td><td>0.6%</td><td></td></tr><tr><td>7</td><td></td><td></td><td>4.5%</td><td>35.0%</td><td>58.4%</td><td>2.1%</td><td></td></tr><tr><td>8</td><td></td><td></td><td>0.7%</td><td>25.9%</td><td>32.6%</td><td>40.7%</td><td></td></tr><tr><td>9</td><td></td><td></td><td></td><td>33.3%</td><td>66.7%</td><td></td><td></td></tr><tr><td colspan="2"></td><td>True Class</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td colspan="2"></td></tr><tr><td colspan="2"></td><td></td><td colspan="7">Predicted Class</td><td>TPR</td><td>FNR</td></tr></table>							3		4.8%	57.1%	33.3%	4.8%			4	0.7%	16.6%	50.3%	29.1%	3.3%			5		0.7%	68.0%	29.1%	2.1%	0.1%		6		0.4%	17.3%	72.4%	9.4%	0.6%		7			4.5%	35.0%	58.4%	2.1%		8			0.7%	25.9%	32.6%	40.7%		9				33.3%	66.7%					True Class	3	4	5	6	7	8	9						Predicted Class							TPR	FNR	<table><tr><td></td><td>100.0%</td></tr><tr><td>16.6%</td><td>83.4%</td></tr><tr><td>68.0%</td><td>32.0%</td></tr><tr><td>72.4%</td><td>27.6%</td></tr><tr><td>58.4%</td><td>41.6%</td></tr><tr><td>40.7%</td><td>59.3%</td></tr><tr><td></td><td>100.0%</td></tr></table>			100.0%	16.6%	83.4%	68.0%	32.0%	72.4%	27.6%	58.4%	41.6%	40.7%	59.3%		100.0%
			3		4.8%	57.1%	33.3%	4.8%																																																																																																	
4	0.7%	16.6%	50.3%	29.1%	3.3%																																																																																																				
5		0.7%	68.0%	29.1%	2.1%	0.1%																																																																																																			
6		0.4%	17.3%	72.4%	9.4%	0.6%																																																																																																			
7			4.5%	35.0%	58.4%	2.1%																																																																																																			
8			0.7%	25.9%	32.6%	40.7%																																																																																																			
9				33.3%	66.7%																																																																																																				
		True Class	3	4	5	6	7	8	9																																																																																																
			Predicted Class							TPR	FNR																																																																																														
	100.0%																																																																																																								
16.6%	83.4%																																																																																																								
68.0%	32.0%																																																																																																								
72.4%	27.6%																																																																																																								
58.4%	41.6%																																																																																																								
40.7%	59.3%																																																																																																								
	100.0%																																																																																																								
4	Fine Gaussian SVM (default)	0.628	Model 4																																																																																																						
			<table><tr><td>3</td><td></td><td></td><td>19.0%</td><td>76.2%</td><td>4.8%</td><td></td><td></td></tr><tr><td>4</td><td></td><td>2.6%</td><td>37.7%</td><td>59.6%</td><td></td><td></td><td></td></tr><tr><td>5</td><td></td><td></td><td>58.2%</td><td>40.7%</td><td>1.0%</td><td></td><td></td></tr><tr><td>6</td><td></td><td></td><td>13.5%</td><td>83.0%</td><td>3.4%</td><td></td><td></td></tr><tr><td>7</td><td></td><td></td><td>3.3%</td><td>55.0%</td><td>41.6%</td><td>0.1%</td><td></td></tr><tr><td>8</td><td></td><td></td><td>2.2%</td><td>60.7%</td><td>25.2%</td><td>11.9%</td><td></td></tr><tr><td>9</td><td></td><td></td><td></td><td>33.3%</td><td>66.7%</td><td></td><td></td></tr><tr><td colspan="2"></td><td>True Class</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td colspan="2"></td></tr><tr><td colspan="2"></td><td></td><td colspan="7">Predicted Class</td><td>TPR</td><td>FNR</td></tr></table>							3			19.0%	76.2%	4.8%			4		2.6%	37.7%	59.6%				5			58.2%	40.7%	1.0%			6			13.5%	83.0%	3.4%			7			3.3%	55.0%	41.6%	0.1%		8			2.2%	60.7%	25.2%	11.9%		9				33.3%	66.7%					True Class	3	4	5	6	7	8	9						Predicted Class							TPR	FNR	<table><tr><td></td><td>100.0%</td></tr><tr><td>2.6%</td><td>97.4%</td></tr><tr><td>58.2%</td><td>41.8%</td></tr><tr><td>83.0%</td><td>17.0%</td></tr><tr><td>41.6%</td><td>58.4%</td></tr><tr><td>11.9%</td><td>88.1%</td></tr><tr><td></td><td>100.0%</td></tr></table>			100.0%	2.6%	97.4%	58.2%	41.8%	83.0%	17.0%	41.6%	58.4%	11.9%	88.1%		100.0%
			3			19.0%	76.2%	4.8%																																																																																																	
4		2.6%	37.7%	59.6%																																																																																																					
5			58.2%	40.7%	1.0%																																																																																																				
6			13.5%	83.0%	3.4%																																																																																																				
7			3.3%	55.0%	41.6%	0.1%																																																																																																			
8			2.2%	60.7%	25.2%	11.9%																																																																																																			
9				33.3%	66.7%																																																																																																				
		True Class	3	4	5	6	7	8	9																																																																																																
			Predicted Class							TPR	FNR																																																																																														
	100.0%																																																																																																								
2.6%	97.4%																																																																																																								
58.2%	41.8%																																																																																																								
83.0%	17.0%																																																																																																								
41.6%	58.4%																																																																																																								
11.9%	88.1%																																																																																																								
	100.0%																																																																																																								

5	Fine Tree (default)	528	Model 5																																																																																									
			<table><tr><td>3</td><td></td><td></td><td>52.4%</td><td>47.6%</td><td></td><td></td><td></td></tr><tr><td>4</td><td>2.0%</td><td>61.6%</td><td>32.5%</td><td>4.0%</td><td></td><td></td></tr><tr><td>5</td><td>0.6%</td><td>59.3%</td><td>36.8%</td><td>3.3%</td><td></td><td></td></tr><tr><td>6</td><td>0.3%</td><td>26.2%</td><td>62.0%</td><td>11.2%</td><td>0.3%</td><td></td></tr><tr><td>7</td><td>0.1%</td><td>6.4%</td><td>56.8%</td><td>35.9%</td><td>0.8%</td><td></td></tr><tr><td>8</td><td></td><td>3.0%</td><td>51.1%</td><td>40.0%</td><td>5.9%</td><td></td></tr><tr><td>9</td><td></td><td></td><td>33.3%</td><td>66.7%</td><td></td><td></td></tr><tr><td colspan="2"></td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr></table>							3			52.4%	47.6%				4	2.0%	61.6%	32.5%	4.0%			5	0.6%	59.3%	36.8%	3.3%			6	0.3%	26.2%	62.0%	11.2%	0.3%		7	0.1%	6.4%	56.8%	35.9%	0.8%		8		3.0%	51.1%	40.0%	5.9%		9			33.3%	66.7%					3	4	5	6	7	8	9	<table><tr><td></td><td>100.0%</td></tr><tr><td>2.0%</td><td>98.0%</td></tr><tr><td>59.3%</td><td>40.7%</td></tr><tr><td>62.0%</td><td>38.0%</td></tr><tr><td>35.9%</td><td>64.1%</td></tr><tr><td>5.9%</td><td>94.1%</td></tr><tr><td></td><td>100.0%</td></tr><tr><td>TPR</td><td>FNR</td></tr></table>				100.0%	2.0%	98.0%	59.3%	40.7%	62.0%	38.0%	35.9%	64.1%	5.9%	94.1%		100.0%	TPR	FNR					
			3			52.4%	47.6%																																																																																					
			4	2.0%	61.6%	32.5%	4.0%																																																																																					
5	0.6%	59.3%	36.8%	3.3%																																																																																								
6	0.3%	26.2%	62.0%	11.2%	0.3%																																																																																							
7	0.1%	6.4%	56.8%	35.9%	0.8%																																																																																							
8		3.0%	51.1%	40.0%	5.9%																																																																																							
9			33.3%	66.7%																																																																																								
		3	4	5	6	7	8	9																																																																																				
	100.0%																																																																																											
2.0%	98.0%																																																																																											
59.3%	40.7%																																																																																											
62.0%	38.0%																																																																																											
35.9%	64.1%																																																																																											
5.9%	94.1%																																																																																											
	100.0%																																																																																											
TPR	FNR																																																																																											
6	Cubic KNN (default)	0.545	Model 7																																																																																									
			<table><tr><td>3</td><td></td><td></td><td>57.1%</td><td>33.3%</td><td>9.5%</td><td></td><td></td></tr><tr><td>4</td><td>0.7%</td><td>59.6%</td><td>36.4%</td><td>3.3%</td><td></td><td></td></tr><tr><td>5</td><td>0.1%</td><td>0.5%</td><td>61.6%</td><td>35.1%</td><td>2.5%</td><td>0.1%</td><td></td></tr><tr><td>6</td><td>0.3%</td><td>25.2%</td><td>64.6%</td><td>9.6%</td><td>0.4%</td><td></td><td></td></tr><tr><td>7</td><td></td><td>8.2%</td><td>54.2%</td><td>36.2%</td><td>1.5%</td><td></td><td></td></tr><tr><td>8</td><td></td><td>6.7%</td><td>46.7%</td><td>45.9%</td><td>0.7%</td><td></td><td></td></tr><tr><td>9</td><td></td><td></td><td>66.7%</td><td>33.3%</td><td></td><td></td><td></td></tr><tr><td colspan="2"></td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr></table>							3			57.1%	33.3%	9.5%			4	0.7%	59.6%	36.4%	3.3%			5	0.1%	0.5%	61.6%	35.1%	2.5%	0.1%		6	0.3%	25.2%	64.6%	9.6%	0.4%			7		8.2%	54.2%	36.2%	1.5%			8		6.7%	46.7%	45.9%	0.7%			9			66.7%	33.3%						3	4	5	6	7	8	9	<table><tr><td></td><td>100.0%</td></tr><tr><td>0.7%</td><td>99.3%</td></tr><tr><td>61.6%</td><td>38.4%</td></tr><tr><td>64.6%</td><td>35.4%</td></tr><tr><td>36.2%</td><td>63.8%</td></tr><tr><td>0.7%</td><td>99.3%</td></tr><tr><td></td><td>100.0%</td></tr><tr><td>TPR</td><td>FNR</td></tr></table>				100.0%	0.7%	99.3%	61.6%	38.4%	64.6%	35.4%	36.2%	63.8%	0.7%	99.3%		100.0%	TPR	FNR
			3			57.1%	33.3%	9.5%																																																																																				
			4	0.7%	59.6%	36.4%	3.3%																																																																																					
5	0.1%	0.5%	61.6%	35.1%	2.5%	0.1%																																																																																						
6	0.3%	25.2%	64.6%	9.6%	0.4%																																																																																							
7		8.2%	54.2%	36.2%	1.5%																																																																																							
8		6.7%	46.7%	45.9%	0.7%																																																																																							
9			66.7%	33.3%																																																																																								
		3	4	5	6	7	8	9																																																																																				
	100.0%																																																																																											
0.7%	99.3%																																																																																											
61.6%	38.4%																																																																																											
64.6%	35.4%																																																																																											
36.2%	63.8%																																																																																											
0.7%	99.3%																																																																																											
	100.0%																																																																																											
TPR	FNR																																																																																											

7	RUS Boosted Trees (default)	0.327	Model 8									
			3	4.8%	19.0%	33.3%	19.0%	4.8%		19.0%	4.8%	95.2%
			4	4.0%	31.1%	37.1%	15.9%	3.3%	1.3%	7.3%	31.1%	68.9%
			5	6.3%	15.2%	49.5%	18.2%	3.2%	1.7%	5.9%	49.5%	50.5%
			6	5.1%	8.8%	26.0%	28.2%	8.2%	7.5%	16.3%	28.2%	71.8%
			7	3.2%	5.2%	11.0%	22.1%	13.5%	16.2%	28.9%	13.5%	86.5%
			8	4.4%	1.5%	5.9%	16.3%	13.3%	25.9%	32.6%	25.9%	74.1%
		9			33.3%		33.3%		33.3%	33.3%	66.7%	
		Predicted Class							TPR	FNR		

8	Boosted Trees (default)	0.532	Model 9									
			3			57.1%	42.9%					100.0%
			4			58.3%	41.1%	0.7%				100.0%
			5		0.1%	59.9%	39.2%	0.7%			59.9%	40.1%
			6			25.9%	67.6%	6.5%			67.6%	32.4%
			7			5.4%	71.0%	23.6%			23.6%	76.4%
			8			1.5%	61.5%	34.8%	2.2%		2.2%	97.8%
		9				66.7%	33.3%			100.0%		
		Predicted Class							TPR	FNR		

9	Weighted KNN (k = 68)	0.680	Model 1																																																																																																									
			<table><tr><td>3</td><td></td><td>4.8%</td><td>47.6%</td><td>47.6%</td><td></td><td></td><td></td></tr><tr><td>4</td><td>0.7%</td><td>14.6%</td><td>53.0%</td><td>31.8%</td><td></td><td></td><td></td></tr><tr><td>5</td><td></td><td>0.1%</td><td>67.9%</td><td>31.5%</td><td>0.6%</td><td></td><td></td></tr><tr><td>6</td><td></td><td></td><td>15.4%</td><td>79.2%</td><td>5.4%</td><td></td><td></td></tr><tr><td>7</td><td></td><td></td><td>1.9%</td><td>43.8%</td><td>54.0%</td><td>0.3%</td><td></td></tr><tr><td>8</td><td></td><td></td><td>1.5%</td><td>34.1%</td><td>34.1%</td><td>30.4%</td><td></td></tr><tr><td>9</td><td></td><td></td><td></td><td>33.3%</td><td>66.7%</td><td></td><td></td></tr><tr><td colspan="2"></td><td>True Class</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td colspan="2"></td></tr><tr><td colspan="2"></td><td></td><td colspan="7">Predicted Class</td><td>TPR</td><td>FNR</td></tr></table>							3		4.8%	47.6%	47.6%				4	0.7%	14.6%	53.0%	31.8%				5		0.1%	67.9%	31.5%	0.6%			6			15.4%	79.2%	5.4%			7			1.9%	43.8%	54.0%	0.3%		8			1.5%	34.1%	34.1%	30.4%		9				33.3%	66.7%					True Class	3	4	5	6	7	8	9						Predicted Class							TPR	FNR	<table><tr><td></td><td>100.0%</td></tr><tr><td>14.6%</td><td>85.4%</td></tr><tr><td>67.9%</td><td>32.1%</td></tr><tr><td>79.2%</td><td>20.8%</td></tr><tr><td>54.0%</td><td>46.0%</td></tr><tr><td>30.4%</td><td>69.6%</td></tr><tr><td></td><td>100.0%</td></tr><tr><td>TPR</td><td>FNR</td></tr></table>				100.0%	14.6%	85.4%	67.9%	32.1%	79.2%	20.8%	54.0%	46.0%	30.4%	69.6%		100.0%	TPR	FNR
			3		4.8%	47.6%	47.6%																																																																																																					
			4	0.7%	14.6%	53.0%	31.8%																																																																																																					
5		0.1%	67.9%	31.5%	0.6%																																																																																																							
6			15.4%	79.2%	5.4%																																																																																																							
7			1.9%	43.8%	54.0%	0.3%																																																																																																						
8			1.5%	34.1%	34.1%	30.4%																																																																																																						
9				33.3%	66.7%																																																																																																							
		True Class	3	4	5	6	7	8	9																																																																																																			
			Predicted Class							TPR	FNR																																																																																																	
	100.0%																																																																																																											
14.6%	85.4%																																																																																																											
67.9%	32.1%																																																																																																											
79.2%	20.8%																																																																																																											
54.0%	46.0%																																																																																																											
30.4%	69.6%																																																																																																											
	100.0%																																																																																																											
TPR	FNR																																																																																																											
10	Trilayered Neural Net (default)	0.541	Model 2																																																																																																									
			<table><tr><td>3</td><td></td><td>4.8%</td><td>57.1%</td><td>38.1%</td><td></td><td></td><td></td></tr><tr><td>4</td><td>1.3%</td><td>4.6%</td><td>64.2%</td><td>27.8%</td><td>2.0%</td><td></td><td></td></tr><tr><td>5</td><td>0.1%</td><td>1.1%</td><td>60.3%</td><td>36.9%</td><td>1.5%</td><td>0.1%</td><td></td></tr><tr><td>6</td><td>0.1%</td><td>0.4%</td><td>22.7%</td><td>67.4%</td><td>9.3%</td><td>0.2%</td><td></td></tr><tr><td>7</td><td></td><td></td><td>6.2%</td><td>60.0%</td><td>33.4%</td><td>0.4%</td><td></td></tr><tr><td>8</td><td></td><td></td><td>1.5%</td><td>53.3%</td><td>44.4%</td><td>0.7%</td><td></td></tr><tr><td>9</td><td></td><td></td><td></td><td>33.3%</td><td>66.7%</td><td></td><td></td></tr><tr><td colspan="2"></td><td>True Class</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td colspan="2"></td></tr><tr><td colspan="2"></td><td></td><td colspan="7">Predicted Class</td><td>TPR</td><td>FNR</td></tr></table>							3		4.8%	57.1%	38.1%				4	1.3%	4.6%	64.2%	27.8%	2.0%			5	0.1%	1.1%	60.3%	36.9%	1.5%	0.1%		6	0.1%	0.4%	22.7%	67.4%	9.3%	0.2%		7			6.2%	60.0%	33.4%	0.4%		8			1.5%	53.3%	44.4%	0.7%		9				33.3%	66.7%					True Class	3	4	5	6	7	8	9						Predicted Class							TPR	FNR	<table><tr><td></td><td>100.0%</td></tr><tr><td>4.6%</td><td>95.4%</td></tr><tr><td>60.3%</td><td>39.7%</td></tr><tr><td>67.4%</td><td>32.6%</td></tr><tr><td>33.4%</td><td>66.6%</td></tr><tr><td>0.7%</td><td>99.3%</td></tr><tr><td></td><td>100.0%</td></tr><tr><td>TPR</td><td>FNR</td></tr></table>				100.0%	4.6%	95.4%	60.3%	39.7%	67.4%	32.6%	33.4%	66.6%	0.7%	99.3%		100.0%	TPR	FNR
			3		4.8%	57.1%	38.1%																																																																																																					
			4	1.3%	4.6%	64.2%	27.8%	2.0%																																																																																																				
5	0.1%	1.1%	60.3%	36.9%	1.5%	0.1%																																																																																																						
6	0.1%	0.4%	22.7%	67.4%	9.3%	0.2%																																																																																																						
7			6.2%	60.0%	33.4%	0.4%																																																																																																						
8			1.5%	53.3%	44.4%	0.7%																																																																																																						
9				33.3%	66.7%																																																																																																							
		True Class	3	4	5	6	7	8	9																																																																																																			
			Predicted Class							TPR	FNR																																																																																																	
	100.0%																																																																																																											
4.6%	95.4%																																																																																																											
60.3%	39.7%																																																																																																											
67.4%	32.6%																																																																																																											
33.4%	66.6%																																																																																																											
0.7%	99.3%																																																																																																											
	100.0%																																																																																																											
TPR	FNR																																																																																																											

11	Medium Neural Net (default)	0.553	Model 3									
			3	4.8%		42.9%	47.6%	4.8%			4.8%	95.2%
			4	2.6%	5.3%	66.2%	22.5%	2.6%	0.7%		5.3%	94.7%
			5	0.5%	0.9%	59.1%	37.3%	1.9%	0.2%		59.1%	40.9%
			6	0.3%	0.1%	23.1%	66.5%	9.7%	0.3%	0.1%	66.5%	33.5%
			7		0.1%	5.2%	56.4%	36.4%	1.9%		36.4%	63.6%
			8		0.7%	3.0%	49.6%	39.3%	7.4%		7.4%	92.6%
			9			33.3%		66.7%				100.0%

13	Bagged Trees (learners = 250)	0.679	Model 4																																																																																																								
			<table><tr><td>3</td><td></td><td>4.8%</td><td>61.9%</td><td>23.8%</td><td>9.5%</td><td></td><td></td></tr><tr><td>4</td><td>0.7%</td><td>17.2%</td><td>49.0%</td><td>33.1%</td><td></td><td></td><td></td></tr><tr><td>5</td><td></td><td>0.3%</td><td>70.5%</td><td>28.7%</td><td>0.5%</td><td></td><td></td></tr><tr><td>6</td><td></td><td>0.2%</td><td>16.2%</td><td>78.0%</td><td>5.6%</td><td>0.1%</td><td></td></tr><tr><td>7</td><td></td><td></td><td>2.9%</td><td>44.0%</td><td>52.3%</td><td>0.8%</td><td></td></tr><tr><td>8</td><td></td><td></td><td>1.5%</td><td>34.8%</td><td>32.6%</td><td>31.1%</td><td></td></tr><tr><td>9</td><td></td><td></td><td></td><td>33.3%</td><td>66.7%</td><td></td><td></td></tr><tr><td colspan="2"></td><td>True Class</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td colspan="2"></td></tr><tr><td colspan="2"></td><td></td><td colspan="7">Predicted Class</td><td>TPR</td><td>FNR</td></tr></table>							3		4.8%	61.9%	23.8%	9.5%			4	0.7%	17.2%	49.0%	33.1%				5		0.3%	70.5%	28.7%	0.5%			6		0.2%	16.2%	78.0%	5.6%	0.1%		7			2.9%	44.0%	52.3%	0.8%		8			1.5%	34.8%	32.6%	31.1%		9				33.3%	66.7%					True Class	3	4	5	6	7	8	9						Predicted Class							TPR	FNR	<table><tr><td></td><td>100.0%</td></tr><tr><td>17.2%</td><td>82.8%</td></tr><tr><td>70.5%</td><td>29.5%</td></tr><tr><td>78.0%</td><td>22.0%</td></tr><tr><td>52.3%</td><td>47.7%</td></tr><tr><td>31.1%</td><td>68.9%</td></tr><tr><td></td><td>100.0%</td></tr><tr><td>TPR</td><td>FNR</td></tr></table>			100.0%	17.2%	82.8%	70.5%	29.5%	78.0%	22.0%	52.3%	47.7%	31.1%	68.9%		100.0%	TPR	FNR
			3		4.8%	61.9%	23.8%	9.5%																																																																																																			
			4	0.7%	17.2%	49.0%	33.1%																																																																																																				
5		0.3%	70.5%	28.7%	0.5%																																																																																																						
6		0.2%	16.2%	78.0%	5.6%	0.1%																																																																																																					
7			2.9%	44.0%	52.3%	0.8%																																																																																																					
8			1.5%	34.8%	32.6%	31.1%																																																																																																					
9				33.3%	66.7%																																																																																																						
		True Class	3	4	5	6	7	8	9																																																																																																		
			Predicted Class							TPR	FNR																																																																																																
	100.0%																																																																																																										
17.2%	82.8%																																																																																																										
70.5%	29.5%																																																																																																										
78.0%	22.0%																																																																																																										
52.3%	47.7%																																																																																																										
31.1%	68.9%																																																																																																										
	100.0%																																																																																																										
TPR	FNR																																																																																																										
14	Coarse KNN (default)	0.533	Model 10																																																																																																								
			<table><tr><td>3</td><td></td><td></td><td>57.1%</td><td>42.9%</td><td></td><td></td><td></td></tr><tr><td>4</td><td></td><td></td><td>62.3%</td><td>37.1%</td><td>0.7%</td><td></td><td></td></tr><tr><td>5</td><td></td><td></td><td>55.0%</td><td>44.1%</td><td>0.9%</td><td></td><td></td></tr><tr><td>6</td><td></td><td></td><td>22.3%</td><td>71.8%</td><td>5.9%</td><td></td><td></td></tr><tr><td>7</td><td></td><td></td><td>4.1%</td><td>73.9%</td><td>22.0%</td><td></td><td></td></tr><tr><td>8</td><td></td><td></td><td>0.7%</td><td>67.4%</td><td>31.9%</td><td></td><td></td></tr><tr><td>9</td><td></td><td></td><td></td><td>33.3%</td><td>66.7%</td><td></td><td></td></tr><tr><td colspan="2"></td><td>True Class</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td colspan="2"></td></tr><tr><td colspan="2"></td><td></td><td colspan="7">Predicted Class</td><td>TPR</td><td>FNR</td></tr></table>							3			57.1%	42.9%				4			62.3%	37.1%	0.7%			5			55.0%	44.1%	0.9%			6			22.3%	71.8%	5.9%			7			4.1%	73.9%	22.0%			8			0.7%	67.4%	31.9%			9				33.3%	66.7%					True Class	3	4	5	6	7	8	9						Predicted Class							TPR	FNR	<table><tr><td></td><td>100.0%</td></tr><tr><td></td><td>100.0%</td></tr><tr><td>55.0%</td><td>45.0%</td></tr><tr><td>71.8%</td><td>28.2%</td></tr><tr><td>22.0%</td><td>78.0%</td></tr><tr><td></td><td>100.0%</td></tr><tr><td></td><td>100.0%</td></tr><tr><td>TPR</td><td>FNR</td></tr></table>			100.0%		100.0%	55.0%	45.0%	71.8%	28.2%	22.0%	78.0%		100.0%		100.0%	TPR	FNR
			3			57.1%	42.9%																																																																																																				
			4			62.3%	37.1%	0.7%																																																																																																			
5			55.0%	44.1%	0.9%																																																																																																						
6			22.3%	71.8%	5.9%																																																																																																						
7			4.1%	73.9%	22.0%																																																																																																						
8			0.7%	67.4%	31.9%																																																																																																						
9				33.3%	66.7%																																																																																																						
		True Class	3	4	5	6	7	8	9																																																																																																		
			Predicted Class							TPR	FNR																																																																																																
	100.0%																																																																																																										
	100.0%																																																																																																										
55.0%	45.0%																																																																																																										
71.8%	28.2%																																																																																																										
22.0%	78.0%																																																																																																										
	100.0%																																																																																																										
	100.0%																																																																																																										
TPR	FNR																																																																																																										

15	Bagged Trees (learners = 68)	0.673	Model 6																																																																																									
			<table><tr><td>3</td><td></td><td>9.5%</td><td>57.1%</td><td>28.6%</td><td>4.8%</td><td></td><td></td></tr><tr><td>4</td><td></td><td>17.2%</td><td>51.0%</td><td>31.1%</td><td>0.7%</td><td></td><td></td></tr><tr><td>5</td><td></td><td>0.3%</td><td>71.1%</td><td>27.7%</td><td>0.9%</td><td></td><td></td></tr><tr><td>6</td><td></td><td>0.2%</td><td>17.1%</td><td>76.4%</td><td>6.3%</td><td>0.1%</td><td></td></tr><tr><td>7</td><td></td><td></td><td>3.6%</td><td>43.0%</td><td>52.3%</td><td>1.1%</td><td></td></tr><tr><td>8</td><td></td><td></td><td>1.5%</td><td>34.1%</td><td>32.6%</td><td>31.9%</td><td></td></tr><tr><td>9</td><td></td><td></td><td></td><td>33.3%</td><td>66.7%</td><td></td><td></td></tr><tr><td colspan="2"></td><td colspan="6">Predicted Class</td></tr></table>							3		9.5%	57.1%	28.6%	4.8%			4		17.2%	51.0%	31.1%	0.7%			5		0.3%	71.1%	27.7%	0.9%			6		0.2%	17.1%	76.4%	6.3%	0.1%		7			3.6%	43.0%	52.3%	1.1%		8			1.5%	34.1%	32.6%	31.9%		9				33.3%	66.7%					Predicted Class						<table><tr><td></td><td>100.0%</td></tr><tr><td>17.2%</td><td>82.8%</td></tr><tr><td>71.1%</td><td>28.9%</td></tr><tr><td>76.4%</td><td>23.6%</td></tr><tr><td>52.3%</td><td>47.7%</td></tr><tr><td>31.9%</td><td>68.1%</td></tr><tr><td></td><td>100.0%</td></tr><tr><td>TPR</td><td>FNR</td></tr></table>				100.0%	17.2%	82.8%	71.1%	28.9%	76.4%	23.6%	52.3%	47.7%	31.9%	68.1%		100.0%	TPR	FNR
			3		9.5%	57.1%	28.6%	4.8%																																																																																				
4		17.2%	51.0%	31.1%	0.7%																																																																																							
5		0.3%	71.1%	27.7%	0.9%																																																																																							
6		0.2%	17.1%	76.4%	6.3%	0.1%																																																																																						
7			3.6%	43.0%	52.3%	1.1%																																																																																						
8			1.5%	34.1%	32.6%	31.9%																																																																																						
9				33.3%	66.7%																																																																																							
		Predicted Class																																																																																										
	100.0%																																																																																											
17.2%	82.8%																																																																																											
71.1%	28.9%																																																																																											
76.4%	23.6%																																																																																											
52.3%	47.7%																																																																																											
31.9%	68.1%																																																																																											
	100.0%																																																																																											
TPR	FNR																																																																																											
16	Bagged Trees (learners = 500)	0.683	Model 7																																																																																									
			<table><tr><td>3</td><td></td><td>4.8%</td><td>61.9%</td><td>28.6%</td><td>4.8%</td><td></td><td></td></tr><tr><td>4</td><td>0.7%</td><td>18.5%</td><td>51.0%</td><td>29.8%</td><td></td><td></td><td></td></tr><tr><td>5</td><td></td><td>0.3%</td><td>71.0%</td><td>28.0%</td><td>0.7%</td><td></td><td></td></tr><tr><td>6</td><td></td><td>0.2%</td><td>15.6%</td><td>78.0%</td><td>6.2%</td><td>0.1%</td><td></td></tr><tr><td>7</td><td></td><td></td><td>3.0%</td><td>43.3%</td><td>53.0%</td><td>0.7%</td><td></td></tr><tr><td>8</td><td></td><td></td><td>1.5%</td><td>38.5%</td><td>27.4%</td><td>32.6%</td><td></td></tr><tr><td>9</td><td></td><td></td><td></td><td>33.3%</td><td>66.7%</td><td></td><td></td></tr><tr><td colspan="2"></td><td colspan="6">Predicted Class</td></tr></table>							3		4.8%	61.9%	28.6%	4.8%			4	0.7%	18.5%	51.0%	29.8%				5		0.3%	71.0%	28.0%	0.7%			6		0.2%	15.6%	78.0%	6.2%	0.1%		7			3.0%	43.3%	53.0%	0.7%		8			1.5%	38.5%	27.4%	32.6%		9				33.3%	66.7%					Predicted Class						<table><tr><td></td><td>100.0%</td></tr><tr><td>18.5%</td><td>81.5%</td></tr><tr><td>71.0%</td><td>29.0%</td></tr><tr><td>78.0%</td><td>22.0%</td></tr><tr><td>53.0%</td><td>47.0%</td></tr><tr><td>32.6%</td><td>67.4%</td></tr><tr><td></td><td>100.0%</td></tr><tr><td>TPR</td><td>FNR</td></tr></table>				100.0%	18.5%	81.5%	71.0%	29.0%	78.0%	22.0%	53.0%	47.0%	32.6%	67.4%		100.0%	TPR	FNR
			3		4.8%	61.9%	28.6%	4.8%																																																																																				
4	0.7%	18.5%	51.0%	29.8%																																																																																								
5		0.3%	71.0%	28.0%	0.7%																																																																																							
6		0.2%	15.6%	78.0%	6.2%	0.1%																																																																																						
7			3.0%	43.3%	53.0%	0.7%																																																																																						
8			1.5%	38.5%	27.4%	32.6%																																																																																						
9				33.3%	66.7%																																																																																							
		Predicted Class																																																																																										
	100.0%																																																																																											
18.5%	81.5%																																																																																											
71.0%	29.0%																																																																																											
78.0%	22.0%																																																																																											
53.0%	47.0%																																																																																											
32.6%	67.4%																																																																																											
	100.0%																																																																																											
TPR	FNR																																																																																											

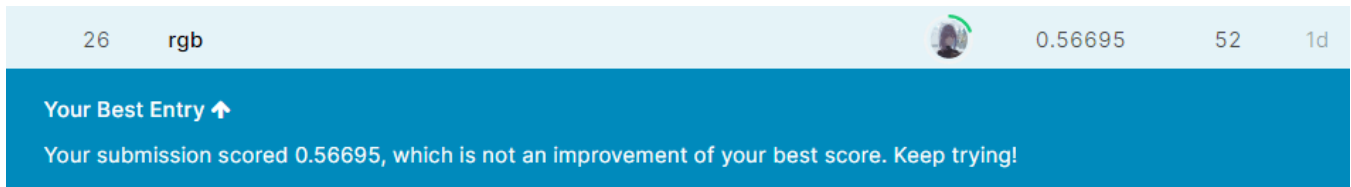
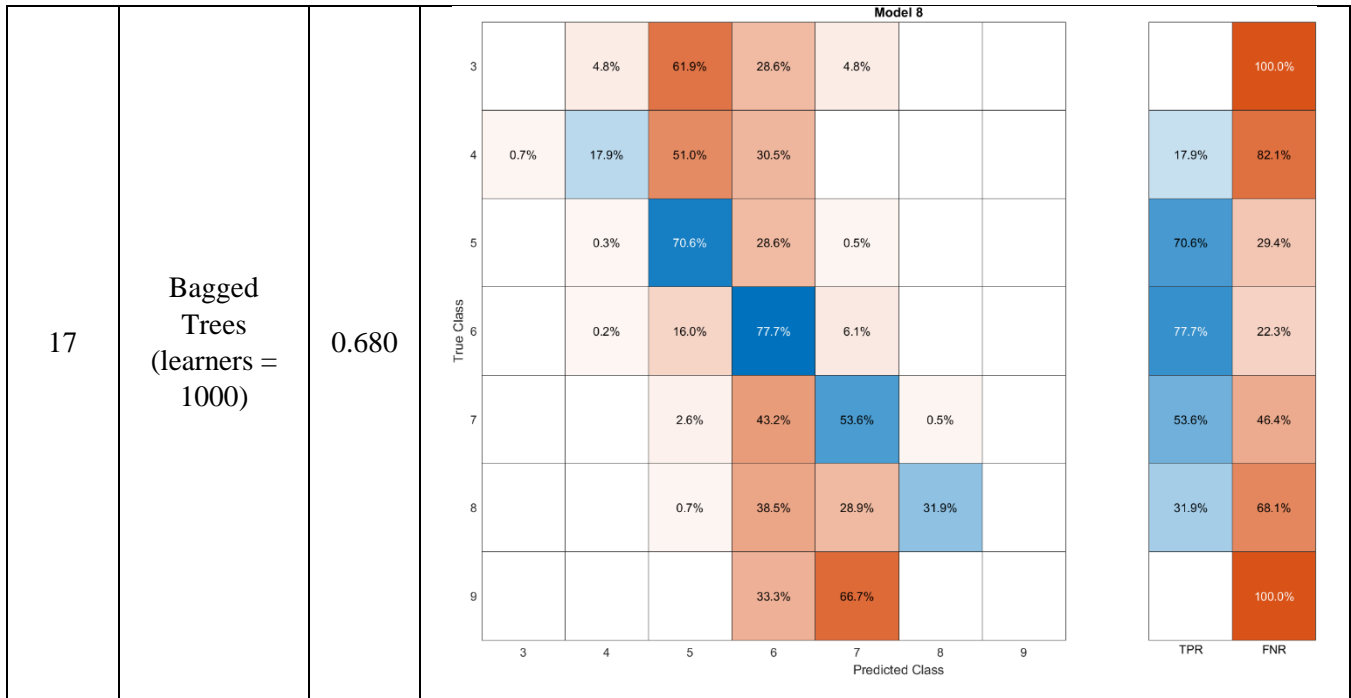


Figure 8. Official Kaggle Submission with a 0.56695 accuracy

V. Appendix

- Appendix A.** Live Script for training separate Red and White wine types into fitnet with hidden layers of [9 2 2 5]

```

clc; clearvars;
% Read data set for wine quality classification train and test
wine_train = readtable('train.csv');
wine_test  = readtable('test.csv');

```

```

% change wine to index
wine_train.wineType = grp2idx(wine_train.wineType)
wine_test.wineType  = grp2idx(wine_test.wineType)

```

```

tabulate(wine_train.wineType)

```

```
% Remove id
test_id          = wine_test.id;
wine_train.id    = [];
wine_test.id     = [];
```

```
tabulate(wine_train.quality)
```

```
% Get sample each
quality_3 = wine_train(wine_train.quality == 3, :);
quality_4 = wine_train(wine_train.quality == 4, :);
quality_5 = wine_train(wine_train.quality == 5, :);
quality_6 = wine_train(wine_train.quality == 6, :);
quality_7 = wine_train(wine_train.quality == 7, :);
quality_8 = wine_train(wine_train.quality == 8, :);
quality_9 = wine_train(wine_train.quality == 9, :);
```

```
% Get k samples based from each quality clusters
k = 1000;
quality_3s = datasample(quality_3, k);
quality_4s = datasample(quality_4, k);
quality_5s = wine_train(wine_train.quality == 5, :)
quality_6s = wine_train(wine_train.quality == 6, :)
quality_7s = wine_train(wine_train.quality == 7, :)
% quality_5s = datasample(quality_5, k);
% quality_6s = datasample(quality_6, k);
% quality_7s = datasample(quality_7, k);
quality_8s = datasample(quality_8, k);
quality_9s = datasample(quality_9, k);
```

```
% Combine all new data
new_wine_train = [quality_3s;quality_4s; quality_5s;
quality_6s;quality_7s;quality_8s;quality_9s];
% Tabulate updated quality
tabulate(new_wine_train.quality)
```

```
% Target quality
red_wine_target = new_wine_train(new_wine_train.wineType == 1,
"quality")
red_wine_target = table2array(red_wine_target)
white_wine_target = new_wine_train(new_wine_train.wineType == 2,
"quality")
white_wine_target = table2array(white_wine_target)
```

```
% Separate train and test
```

```

red_train = new_wine_train(new_wine_train.wineType == 1, :)
white_train = new_wine_train(new_wine_train.wineType == 2, :)
red_test = wine_test(wine_test.wineType == 1, :)
white_test = wine_test(wine_test.wineType == 2, :)

```

```

% Normalize train and test

```

```

red_wine_train = normalize(removevars(red_train, {'quality',
'wineType'}), 'range');
white_wine_train = normalize(removevars(white_train, {'quality',
'wineType'}), 'range');
red_wine_test = normalize(removevars(red_test, {'wineType'}), 'range');
white_wine_test = normalize(removevars(white_test, {'wineType'}),
'range');

```

```

red_wine_train
[idx,scores] = fscchi2(red_wine_train, red_wine_target)
bar(scores(idx))
xlabel('Predictor rank')
ylabel('Predictor importance score')

```

```

white_wine_train
[idx2,scores2] = fscchi2(white_wine_train, white_wine_target)
bar(scores2(idx2))
xlabel('Predictor rank')
ylabel('Predictor importance score')

```

Test

```

% id = test_id;
% quality = BGTREE88.predictFcn(wine_test_fin);
% results = table(id, quality);
% writetable(results , 'matlabSubmission12.csv');

```

```

% RED Feature, Target, Testing param
red_ft = idx(1:7)
red_feature = red_wine_train(:, red_ft) % idx(1:9)
red_target = double(red_wine_target)
red_testing = red_wine_test(:, red_ft) % idx(1:9)

red_x = red_feature(:,:);
red_y = red_target';
red_z = red_testing(:,:);

```

```
tabulate(red_y)
```

```
% WHITE Feature, Target, Testing param
white_ft = idx2(1:7)
white_feature = white_wine_train(:,white_ft) % idx(1:9)
white_target = double(white_wine_target)
white_testing = white_wine_test(:, white_ft) % idx(1:9)

white_x = white_feature(:,:);
white_y = white_target';
white_z = white_testing(:,:);
tabulate(white_y)
```

```
redNet = fitnet([9 2 2 5], 'trainlm');
redNet.trainParam.max_fail = 10;
redTrainNet = train(redNet, red_x, red_y);
```

```
whiteNet = fitnet([9 2 2 5], 'trainlm');
whiteNet.trainParam.max_fail = 10;
whiteTrainNet = train(whiteNet, white_x, white_y);
```

```
red_results = round(redTrainNet(red_z))
tabulate(red_y)
tabulate(red_results)
```

```
white_results = round(whiteTrainNet(white_z))
tabulate(white_y)
tabulate(white_results)
```

Save the results to .csv

```
id = test_id;
quality = [red_results, white_results]'
tabulate(quality)
exported = table(id, quality);
writetable(exported , 'results.csv');
```

b. **Appendix B.** *Final Live Script for Wine Quality Classification*

LBYEC3B PROJECT 1 - BAYETA | TAN | TUPAL

Wine Quality Classification

```
% Clearvars  
clc; clearvars;
```

```
% Read data set for wine quality classification train and test  
wine_train = readtable('train.csv');  
wine_test  = readtable('test.csv');
```

Statistical Analysis

```
% Change wineType from str to idx  
% wine_train.wineType = grp2idx(wine_train.wineType);  
% wine_test.wineType  = grp2idx(wine_test.wineType);
```

```
% Remove ID  
test_id      = wine_test.id;  
qlty         = wine_train.quality;  
wine_train.id = [];  
wine_test.id  = [];  
wine_train.wineType = [];  
wine_test.wineType  = [];
```

```
% Displays the wine features  
colnames = wine_train.Properties.VariableNames  
figure('Name','Wine Features', 'NumberTitle', 'off')  
for idx = 1: length(colnames)  
    col_name = string(colnames(:,idx));  
    subplot(4,3,idx)  
    histogram(wine_train.(col_name))  
    title(col_name)  
    % disp(idx)  
end
```

```
summary(wine_train)
```

```
% Generate correlation matrix
figure()
corrplot(wine_train)
```

```
% Display colnames
colnames
```

Preprocessing

```
% Feature ranking by chi-squared test
% Univariate feature ranking for classification using chi-square tests
% fscchi2 examines whether each predictor variable is independent of
% a response variable by using individual chi-square tests.
figure()
[idx2,scores2] = fscchi2(removevars(wine_train, {'quality'}), qlty)
bar(scores2(idx2))
xlabel('Predictor rank')
ylabel('Predictor importance score')
```

```
% Tabulate idx2 vs scores2
top_col = idx2';
top_col_scores = scores2';
idx2vsscores2 = table(top_col, top_col_scores);
idx2vsscores2
```

```
% Display colnames
colnames
```

```
% Drop quality
wine_train.quality = [];
% Functions for outliers (did not work)
% new_data = filloutliers(wine_train, 'spline','percentile', [10 90]);
```

```
% Normalize
norm_train = normalize(wine_train);
norm_test = normalize(wine_test);
```

```
% Select ranked features
mask = idx2(1:8)
wine_feature = norm_train(:, mask);
wine_target = double(qlty);
wine_testing = norm_test(:, mask);
```

Models and Results

Models are trained under the Classification App of MATLAB. It can be replicated by checking the parameters stated in each section below.

```
% % Bagged Trees (Random-Forest), Default Params
% id = test_id;
% quality = FBGTREE_DEF_669.predictFcn(wine_testing);
% disp('Quality Prediction for Final1')
% tabulate(quality)
% exported = table(id, quality);
% writetable(exported , 'final1_baggedtrees_def_669.csv');
```

```
% % Bagged Trees (Random-Forest), 100 learners
% id = test_id;
% quality = FBGTREE_100L_677.predictFcn(wine_testing);
% disp('Quality Prediction for Final2')
% tabulate(quality)
% exported = table(id, quality);
% writetable(exported , 'final2_baggedtrees_100L_677.csv');
```

```
% % Weighted KNN, Default Params
% id = test_id;
% quality = FWKNN_DEF_654.predictFcn(wine_testing);
% disp('Quality Prediction for Final3')
% tabulate(quality)
% exported = table(id, quality);
% writetable(exported , 'final3_weightedKNN_def_654.csv');
```

```
% % Fine Gaussian SVM, Default Params
% id = test_id;
% quality = FFGSVM_DEF_628.predictFcn(wine_testing);
% disp('Quality Prediction for Final4')
% tabulate(quality)
% exported = table(id, quality);
% writetable(exported , 'final4_finegaussianSVM_def_628.csv');
```

```
% % Fine Tree, default parameters
% id = test_id;
% quality = FFTREE_DEF_528.predictFcn(wine_testing);
% disp('Quality Prediction for Final5')
% tabulate(quality)
% exported = table(id, quality);
```

```
% writetable(exported , 'final5_finetree_def_528.csv');
```

```
% % Cubic KNN, default  
% id = test_id;  
% quality = FCKNN_DEF_545.predictFcn(wine_testing);  
% disp('Quality Prediction for Final6')  
% tabulate(quality)  
% exported = table(id, quality);  
% writetable(exported , 'final6_cubicKNN_def_545.csv'); % 7 in window
```

```
% % RUS Boosted Trees, default parameters  
% id = test_id;  
% quality = FRUSBST_DEF_327.predictFcn(wine_testing);  
% disp('Quality Prediction for Final7')  
% tabulate(quality)  
% exported = table(id, quality);  
% writetable(exported , 'final7_RUSBoostedTrees_def_327.csv');
```

```
% % Boosted Trees, default  
% id = test_id;  
% quality = FBSTREE_DEF_532.predictFcn(wine_testing);  
% disp('Quality Prediction for Final8')  
% tabulate(quality)  
% exported = table(id, quality);  
% writetable(exported , 'final8_BoostedTrees_def_532.csv');
```

```
% % Weighted KNN, k = 68  
% id = test_id;  
% quality = FWKNN_68K_680.predictFcn(wine_testing);  
% disp('Quality Prediction for Final9')  
% tabulate(quality)  
% exported = table(id, quality);  
% writetable(exported , 'final9_WKNN_68k_680.csv');
```

```
% % Trilayered NN, default  
% id = test_id;  
% quality = WTRINN_DEF_541.predictFcn(wine_testing);  
% disp('Quality Prediction for Final10')  
% tabulate(quality)  
% exported = table(id, quality);  
% writetable(exported , 'final10_TRINN_541.csv');
```

```
% % Medium NN, default  
% id = test_id;
```



```
% quality = FMEDNN_DEF_553.predictFcn(wine_testing);
% disp('Quality Prediction for Final11')
% tabulate(quality)
% exported = table(id, quality);
% writetable(exported , 'final11_MEDNN_553.csv');
```

```
% % Coarse Gaussian SVM
% id = test_id;
% quality = FCGSVM_DEF_524.predictFcn(wine_testing);
% disp('Quality Prediction for Final12')
% tabulate(quality)
% exported = table(id, quality);
% writetable(exported , 'final12_CoarseGaussSVM_524.csv');
```

```
% % Bagged Trees (Random Forest), 250 learners
% id = test_id;
% quality = FBGTREE_250L_679.predictFcn(wine_testing);
% disp('Quality Prediction for Final13')
% tabulate(quality)
% exported = table(id, quality);
% writetable(exported , 'final13_baggedtrees_250L_679.csv');
```

```
% % Coarse KNN, default
% id = test_id;
% quality = FCKNN_DEF_533.predictFcn(wine_testing);
% disp('Quality Prediction for Final14')
% tabulate(quality)
% exported = table(id, quality);
% writetable(exported , 'final14_coarseKNN_def_533.csv');
%
```

```
% % Bagged Trees (Random Forest), 68 learners
% id = test_id;
% quality = FBGTREE_68L_673.predictFcn(wine_testing);
% disp('Quality Prediction for Final15')
% tabulate(quality)
% exported = table(id, quality);
% writetable(exported , 'final15_baggedtrees_68L_673.csv');
```

```
% % Bagged Trees (Random Forest), 500 learners
% disp('nag run ako')
% id = test_id;
% quality = FBGTREE_500L_683.predictFcn(wine_testing);
% disp('Quality Prediction for Final16')
% tabulate(quality)
```

```
% exported = table(id, quality);  
% writetable(exported , 'final16_baggedtrees_500L_683.csv');
```

```
% % Bagged Trees (Random Forest), 1000 learners  
% id = test_id;  
% quality = FBGTREE_1000L_680.predictFcn(wine_testing);  
% disp('Quality Prediction for Final17')  
% tabulate(quality)  
% exported = table(id, quality);  
% writetable(exported , 'final17_baggedtrees_1000L_680.csv');
```

```
% id = test_id;  
% quality = BAGTREE5K66.predictFcn(wine_testing);  
% tabulate(quality)  
% exported = table(id, quality);  
% writetable(exported , 'lastsubmission3.csv');
```

```
% id = test_id;  
% quality = BAGTREE5k668.predictFcn(wine_testing);  
% tabulate(quality)  
% exported = table(id, quality);  
% writetable(exported , 'lastsubmission4.csv');
```

```
% id = test_id;  
% quality = WKK5k65.predictFcn(wine_testing);  
% tabulate(quality)  
% exported = table(id, quality);  
% writetable(exported , 'lastsubmission5.csv');
```

```
% id = test_id;  
% quality = DEFBGTREE68.predictFcn(wine_testing);  
% tabulate(quality)  
% exported = table(id, quality);  
% writetable(exported , 'lastsubmission6.csv');
```

```
% id = test_id;  
% quality = DEFWKNN669.predictFcn(wine_testing);  
% tabulate(quality)  
% exported = table(id, quality);  
% writetable(exported , 'lastsubmission7.csv');
```

```
% id = test_id;  
% quality = BAGTREESIDX1_67.predictFcn(wine_testing);
```

```
% tabulate(quality)
% exported = table(id, quality);
% %idx1 (11 param)
% writetable(exported , 'lastsubmission8.csv');
```

```
% id = test_id;
% quality = bg66idx8.predictFcn(wine_testing);
% tabulate(quality)
% exported = table(id, quality);
% %idx2 (9 param)
% writetable(exported , 'lastsubmission9.csv');
```

```
% id = test_id;
% quality = bg671.predictFcn(wine_testing);
% tabulate(quality)
% exported = table(id, quality);
% %idx2 (9 param 1-9)
% writetable(exported , 'lastsubmission10.csv');
```

```
% id = test_id;
% quality = bgg655id2x17.predictFcn(wine_testing);
% tabulate(quality)
% exported = table(id, quality);
% %idx2 (9 param 1-9)
% writetable(exported , 'lastsubmission11.csv');
```

```
% id = test_id;
% quality = BAGTREEUPDATED677IDX28.predictFcn(wine_testing);
% tabulate(quality)
% exported = table(id, quality);
% %idx2 (9 param 1-9)
% writetable(exported , 'lastsubmission12.csv');
```

```
% id = test_id;
% quality = bg673idx19.predictFcn(wine_testing);
% tabulate(quality)
% exported = table(id, quality);
% %idx2 (9 param 1-9)
% writetable(exported , 'lastsubmission13.csv');
```

```
% id = test_id;
% quality = bg658idx18.predictFcn(wine_testing);
% tabulate(quality)
% exported = table(id, quality);
```

```
% %idx2 (9 param 1-9)
% writetable(exported , 'lastsubmission14.csv');
```

```
% id = test_id;
% quality = bg67idx18.predictFcn(wine_testing);
% tabulate(quality)
% exported = table(id, quality);
% %idx2 (9 param 1-9)
% writetable(exported , 'lastsubmission15.csv');
```

```
% id = test_id;
% quality = trainedModelbg100677idx28.predictFcn(wine_testing);
% tabulate(quality)
% exported = table(id, quality);
% %idx2 (9 param 1-9)
% writetable(exported , 'lastsubmission16.csv');
```