Report Final Machine Learning Methods

Cesar Conejo Villalobos

1/24/2021

1. Introduction

In this project, we continue with the study of the Portuguese Vinho Verde wine data set. The file includes 1599 observations of red wine and 4898 of white wine. The response variable of the original data set was `quality`, a continuous variable with a scale from 0 to 10 that denotes que quality taste given by three wine experts. On the other hand, the covariates variables correspond to the chemical properties associated with the wine.

In order to refresh the variables of the data set, the following output shows a review of the variables for the complete data set.

```
## Rows: 6,497
## Columns: 13
## $ fixed.acidity        <dbl> 7.4, 7.8, 7.8, 11.2, 7.4, 7.4, 7.9, 7.3, 7.8, ...
## $ volatile.acidity     <dbl> 0.700, 0.880, 0.760, 0.280, 0.700, 0.660, 0.60...
## $ citric.acid          <dbl> 0.00, 0.00, 0.04, 0.56, 0.00, 0.00, 0.06, 0.00...
## $ residual.sugar       <dbl> 1.9, 2.6, 2.3, 1.9, 1.9, 1.8, 1.6, 1.2, 2.0, 6...
## $ chlorides            <dbl> 0.076, 0.098, 0.092, 0.075, 0.076, 0.075, 0.06...
## $ free.sulfur.dioxide  <dbl> 11, 25, 15, 17, 11, 13, 15, 15, 9, 17, 15, 17,...
## $ total.sulfur.dioxide <dbl> 34, 67, 54, 60, 34, 40, 59, 21, 18, 102, 65, 1...
## $ density              <dbl> 0.9978, 0.9968, 0.9970, 0.9980, 0.9978, 0.9978...
## $ pH                   <dbl> 3.51, 3.20, 3.26, 3.16, 3.51, 3.51, 3.30, 3.39...
## $ sulphates            <dbl> 0.56, 0.68, 0.65, 0.58, 0.56, 0.56, 0.46, 0.47...
## $ alcohol              <dbl> 9.4, 9.8, 9.8, 9.8, 9.4, 9.4, 9.4, 10.0, 9.5, ...
## $ quality              <int> 5, 5, 5, 6, 5, 5, 5, 7, 7, 5, 5, 5, 5, 5, 5, 5...
## $ Category             <fct> Red, Red, Red, Red, Red, Red, Red, Red, Red, R...
```

As in the middle term task, the goal of this project is to predict the quality of the wine based on these chemical properties.

From the first task, we showed how the proportions of observations of each category were unbalanced. As consequence, linear regression methods were not able to predict accurately the quality of the wine on the scale from 0 to 10.

Moreover, the previous data preprocessing of the data reflected some fact that it becomes necessary to remember:

- The category of wine (red or white) does not define an important factor for determining the quality.
- The variables `alcohol`, `density`, and `residual.sugar` were the variable more important for defining the quality of the wine.
- The covariates were mostly independent, without a significant correlation between them.

Then, we transform the problem from regression to classification. We created a new variable with three categories called `quality.class` with the distribution given in table 1.

Table 1: Count and Proportion (in %) of quality of wine categories

Quality class	Count	Proportion
Low	2384	0.37
Medium	2836	0.44
High	1277	0.20

Additionally, The key metrics defined in the previous task were:

1. Global Accuracy: General accuracy of the predictive capability of the model.
2. High-quality sensitivity: Percentage of high-quality wine fitted correctly.
3. The number of low-quality misclassification: Nominal value of low-quality predicted as high-class.

For this task, we continue with these key-metrics. However, we will introduce some changes:

1. Instead of using only the global accuracy, we also will take a look at the kappa metric. This metric can give more details about the predictive power of a specific model, especially in the classification task with 3 categories.
2. In the classification problem with 3 categories, the high-quality classification is given by the sensitivity (High-quality wine predicted correctly divided by the total number of high-quality wine). However, in the classification task with two classes, this ratio is given by the specificity. As a result, we must be aware of this fact!

Moreover, in the previous task we considered a rigorous scenario where predicting high-quality wine was crucial for the interest of the company. As a result, we developed to exercises:

1. Reduction of noise: The goal in this scenario is to detect *high quality* wine from *No high* quality. So, we convert the classification problem into a binary exercise combining the classes *low* and *medium* as *Regular* wine quality.
2. Cost-Sensitive learning: We go deep into the relationship between high-quality wine and price. In markets, the price of high-quality wine is double in comparison to regular wine. Based on this data set, the company desires to forecast the economic profit in case that all the wine products were fitted correctly.

Table 2: Cost-sensitivity assumptions: Profit matrix

Prediction_And_Reference	No_High	High
Regular	0.0	0.1
High	-0.5	1.0

Finally, in task 1, the analyzed methods were statistical learning frameworks. The model with the best performance in that task was the `sparseLDA`.

Now, in this second task, we introduce the following machine learning methods:

1. k-nearest neighbors (KNN)
2. Support Vector Machines (SVM)
3. Random Forest (RF)
4. Gradient Boosting (xgb)
5. Neural Networks (nn)
6. Ensemble methods

The general idea is to determine how these machine learning methods perform in the task of predicting the quality class of wine. In order to compare the performance and effectiveness of these new frameworks, we will use the best-fitted model in the previous task, `sparseLDA` as the **Benchmark** model.

2. Data Splitting

The first step for training the models and avoid any data leakage is to split the data set into training and testing sets. In this case, we use the caret function `createDataPartition()` with 80% for training and 20% for testing.

Then, we proceed with the training and hyper-parameter tuning of the models. For this task, we use the function `train()` using cross-validation with 5 repeats of 10-fold cross-validation. However, we continue with two different strategies for the 3 categories and binary classification task.

- In the case of classification in 3 categories of the quality type of wine, the metric used for optimizing the models was the `accuracy`.
- In the case of binary classification, The most important element for this task is predicting the greatest amount of good quality wine. It is for this reason that the metric `specificity` and the train control `summaryFunction = twoClassSummary` is used for optimizing the algorithms and to detect the highest quantity of good quality wine. Additionally, we performance another process of training, but optimizing the `EconomicProfit` for the best model of the previous analysis.

3. Classification problem: Three wine quality class

Under the scheme explained in the previous sections, table 3 provides the key metrics of the models. In the case of `sparseLDA`, the accuracy attached is 54%, with 36% of high wine quality predicted accurately.

Table 3: Metrics for Benchmarrk and ML methods

Methods	Accuraracy	kappa	High.Sensitivity	Worst.Error
sparseLDA	54.70	27.39	36.08	17
knn	58.40	34.77	53.33	22
SVM	61.79	38.04	36.08	5
RF	70.57	53.21	59.61	9
xgb	65.95	45.69	52.16	11
nn	57.70	31.90	36.08	7
dnn	57.70	31.90	36.08	7
Ensemble	67.41	47.90	53.33	8

In relation to the machine learning methods, we distinguish two groups:

- First, methods such as `knn` and `nn` have a performance similar to the benchmark. In terms of accuracy, these frameworks continue to possess limited predicted power.
- In the second group, we start to see accuracy metrics in the range from 60% to 70%. In general models, these methods can be used as a naive estimator of the accuracy of quality class. However, in terms of kappa metric, only random Forest has a value superior to 50%, which is considered as no high.

If we focus on the high-quality sensibility, the efficiency continues been below 60% for all the models. In conclusion, both statistical and machine learning methods continue with problems in the task of using the chemical properties of the wine in order to be able to identify the high-quality wine class.

The figure 1 reflects the distribution of `accuracy` and `kappa` metrics based on the function `resamples`. As in the previous comments, Random Forest is the algorithm with the best fulfillment. But, the figure reflects

another advantage of Random Forest in comparison with the method of gradient boosting. We can observe how the former algorithm is more robust in comparison with the `xgb` frameworks, which for some resamples, the metrics can be extraordinarily high.

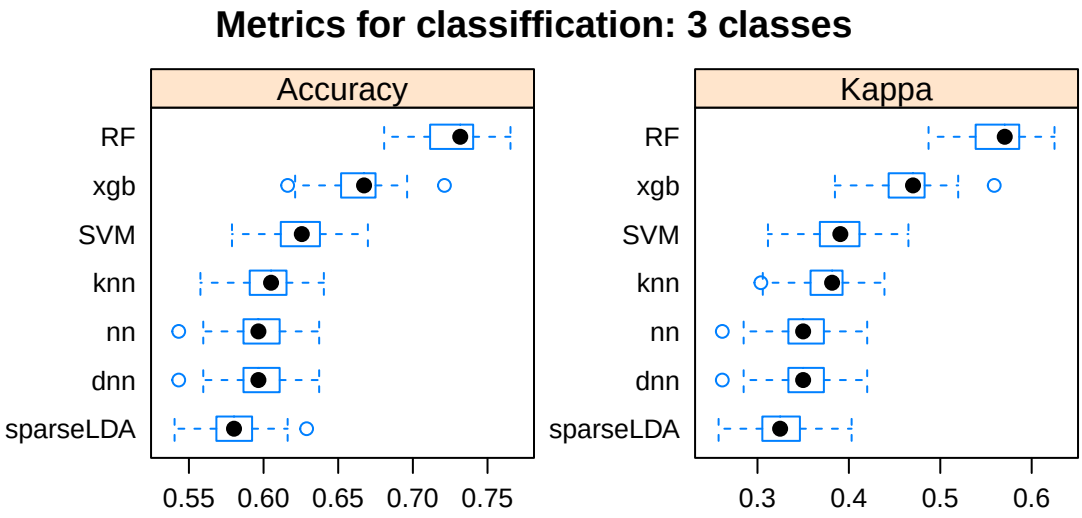

Figure 1: Accuracy and Kappa for Benchmartk and ML methods

Additionally, the last line of table 3 contains information on an ensemble method. This ensemble consists of the three models with performance, specifically, `Random Forest`, `xgboost`, and `SVM` Support Vector Machine. However, we observe how the general realization of the model is almost the average of the previous models. This is due to the correlation between the fitted values is considerably high.

In the case of analyzing the feature importance of the variable under the machine learning methods, we can take Random Forest has the algorithm reference. Figure 2 highlights variables `alcohol` and `density` for predicting the class of the wine in the figure. However, we identify also the influence of the variable `volatile.acidity`. It differs from the benchmark model, where `residual.sugar` had more influence in fitting the class of wine.

Finally, the confusion matrix for the Random Forest is given in table 4. In relation to the benchmark model, the detection of high-quality wine increases 65% passing from 92 to 152 high-quality wine predicted accurately.

Table 4: CM RandomForest

	Low	Medium	High
Low	360	106	5
Medium	107	404	98
High	9	57	152

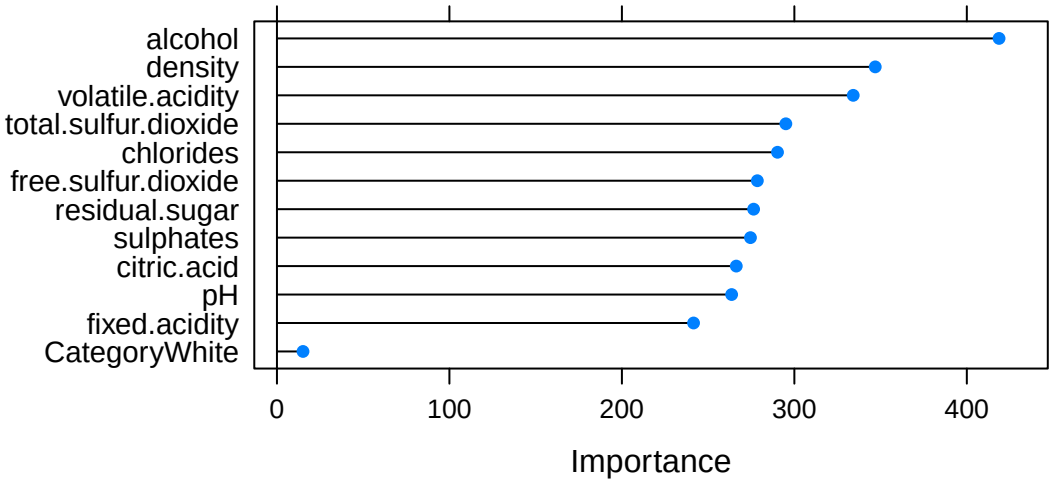

Figure 2: Importance of features in Random Forest Calibration

4. Binary classification

We divided this section into three segments. First, we proceed with the optimization of the Specificity.

In the second part, we start to consider the threshold of the prediction categories. We know that in the scheme used in the previous training, the Bayes rule is used. So, we explore the ROC curve for the method with the best performance in the first part.

Finally, under the scenario of cost-sensitive learning, we optimize the training method considering the cost of the classification showed in table 2.

4.1 Specificity optimization

Similar to the three quality cluster classification, we can consider the binary problem of identifying high-quality wine from the other two groups. Table 5 reflects the key metrics for this approach. Under this scenario, our Benchmark model has an accuracy approximately of 81%. Nevertheless, the **specificity** (High-quality wine detected accurately) is still low.

Similar to the previous exercise, `Random Forest` has the best performance. However, it is important to notice how the method `knn` the second better percentage of detection of high-quality wine. Algorithm `xgboost` completes the rank of the best first three models.

Table 5: Metrics for Benchmark and ML methods

Methods	Accuraracy	kappa	Specificity	Worst.Error
sparseLDA	80.99	28.64	30.98	176
knn	83.60	46.37	54.12	117
SVM	82.99	36.81	37.25	160
RF	88.30	60.20	61.18	99
xgb	84.99	48.86	52.55	121
nn	81.83	35.55	39.61	154
dnn	81.83	35.55	39.61	154
Ensemble	87.07	55.30	56.08	112

The representation in figure 3 is evidence of the method of training of this method under the scheme `twoClassSummary` in Caret. The right panel of figure 3 reflects the sampling of the specificity. The left panel shows the Area under the curve for the models. In this case, Random Forest, `xgboost`, and Support Vector Machines has the better metrics value.

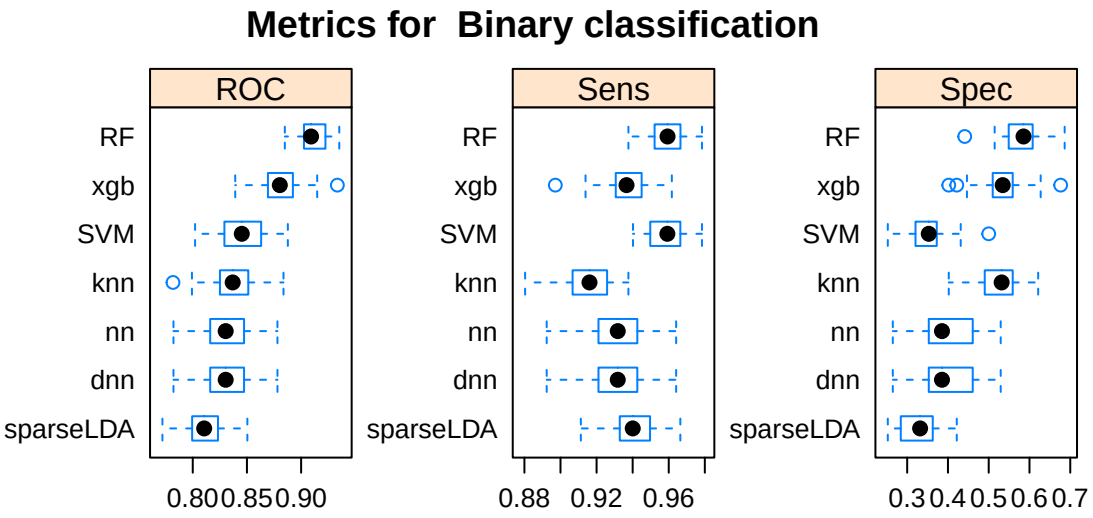

Figure 3: Accuracy and Kappa for Benchmartk and ML methods

On the other hand, taking as reference the Random Forest algorithm, figure 4 reflects the feature importance in the training of the models. We notice how the variable `alcohol` is the most relevant feature for fitting the quality wine. Another important aspect is to considerer how the `density` covariate has no relevant paper in this algorithm, in contrast with the statistical learning methods and even in the problem of three quality classes of wine.

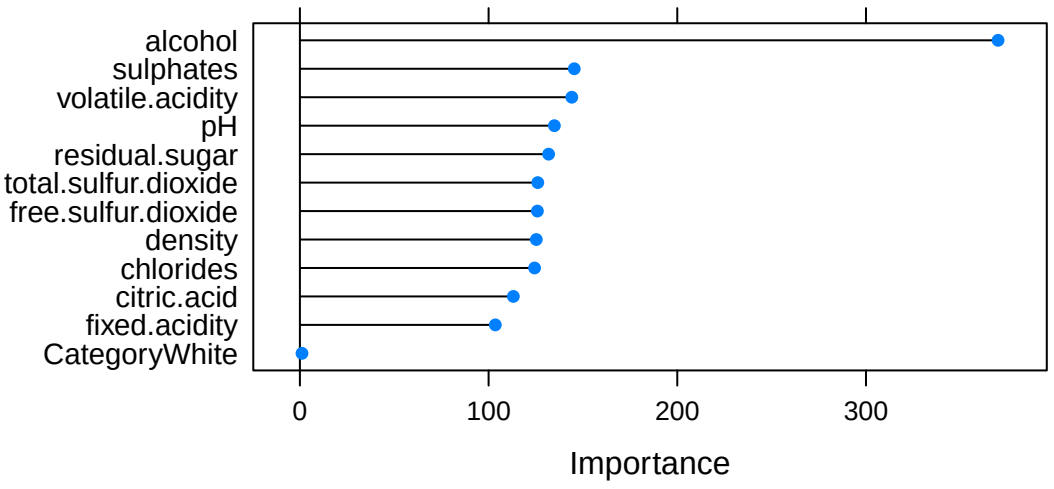

Figure 4: Importance of features in Random Forest Binary Classification

Figure 5 shows the confusion matrix of the Random Forest Algorithm. In this case, the accuracy and kappa values are enough for considering a model with good predictive power. However, the specificity value of 61% in the detection of high-quality can make the decision-maker explore other alternatives to have a better prediction of the wine products. We will explore two ways of improving our previous results.

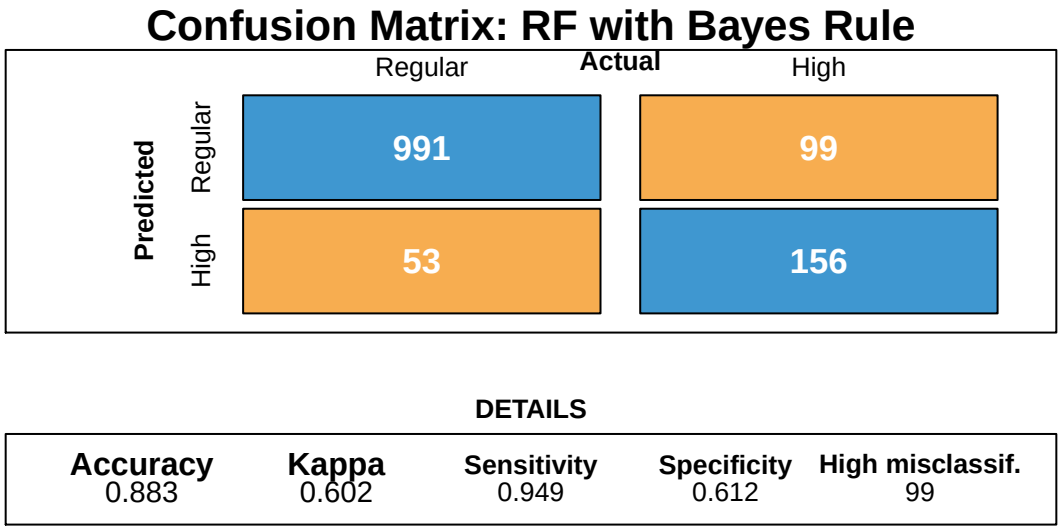

Figure 5: CM with Bayes Rule optimization

4.2 ROC curve

The confusion matrix in figure 5 is the result of applying the Bayes rule as optimization criteria for fitting the categories of high and regular wine.

Additionally, we focus on the relevance of the specificity to have a well accurate model. Figure 6 represents the ROC and AUC for the random forest model. In general terms, the ROC curve can improve the model considering the best balance between sensitivity and specificity.

We notice in this case, how a threshold of 0.25 is reasonable for predicting better the high -quality wine. This threshold is a reflection of the proportion of high-quality wine in the data set, which corresponds to 20% of all the observations.

Figure 7 reflects the confusion matrix of the random Forest. We notice how the metrics values of `accuracy` and `kappa` continue to be satisfactory. The sensitivity downs 14% passing from 94% to 80.7%. However, under this model, it is no the worst error. We can decrease the sensitivity in order to predict high-quality wine. In this case, the specificity increases 35% increasing from 61.2% to 82.7%. This change represents only 44 high-quality wrongly fitted.

As a result, we can consider this approach has a good fitting in the classification problem, However, in order to have these metrics, we increase in several aspects the bias in the calibration of these algorithms

Finally, we can also ask if this is the best model possible under these assumptions. To respond to this question, we can use some techniques relates to cost-sensitive learning.

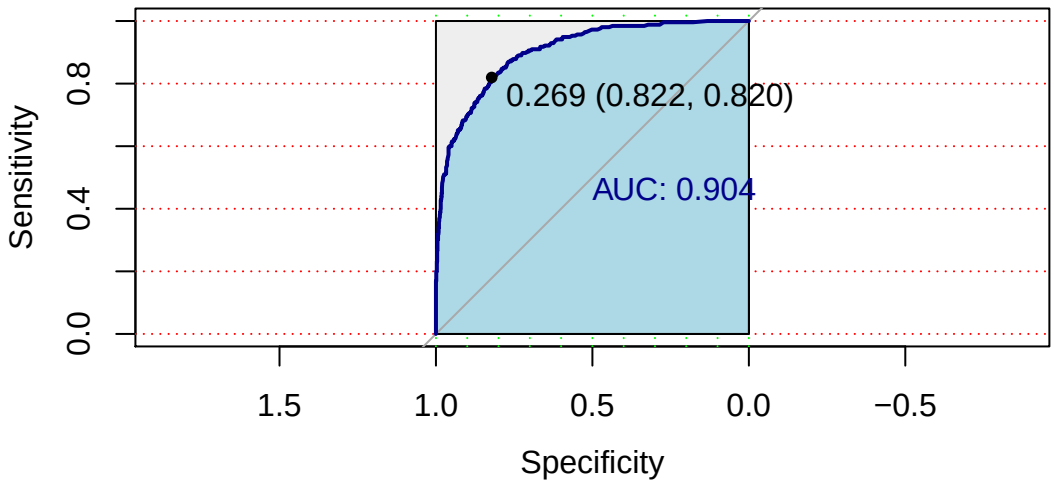

Figure 6: ROC and AUC for RandomForest

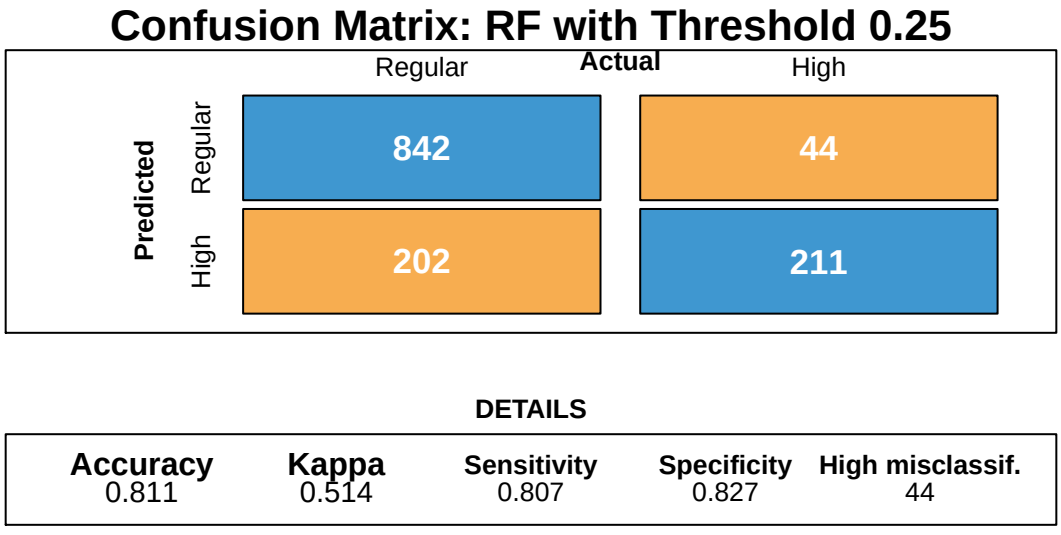

Figure 7: CM recommended threshold given by ROC

4.3 Cost Sensitive learning

Considering as reference the economic scenario showed in table 2 in the introduction, the benchmark model `sparseLDA` obtained the following results:

- Under the Bayes rule (Similar to approach 1), the economic profit was 61.1 monetary units.
- Then, an optimal threshold of 0.35 was used for maximizing the profit of the company. With this threshold, the profit increased to 67.4

Taking as reference the previous values, the economic profit for the algorithm of Random Forest using a Bayes Rule is 139.4. So, by only using a better tool for predicting, we increase the economic profit by 106%. In case that we use the threshold given by the ROC curve, the economic profit decreased to 114.4.

As consequence, we make another process of training, but in this case, the goal will be to optimize the economic profit of the company. So, in the training of this process, we used a metric with a created function called `economicProfit` that should be maximized. Figure 8 shows the confusion matrix under this paradigm. The economic profit is in this case 140.1 monetary units.

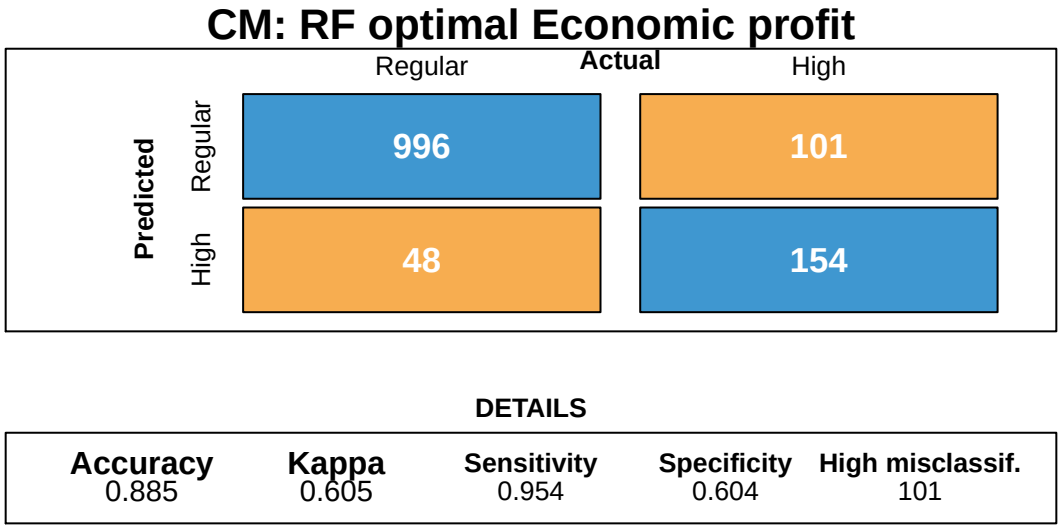

Figure 8: CM for maximum value in Economic Profit

We must notice in the metrics values of the confusion matrix, and especially `specificity` are closer to the values given by the Bayes rule in the first approach of the binary classification. We must notice that under this optimization, the training processed is designed for optimizing the economic profit using the Bayes Rule. In case that we use the threshold given by the ROC curve, the economic profit will be 111.7 monetary units.

Finally, the economic profit under other methods are:

- xgboosting: 109.7.
- KNN: 99.4.
- SVM: 109.7.

All this values are considerably low in comparison with the profit given by Random Forest, even using a no optimal threshold as 0.25 in this algorithm.

5. Conclusions

The general idea behind the middle and final project is to contrast the efficiency and predictive power of the tools provided by statistical learning and machine learning frameworks. In the case of the former, these tools have the benefit of classifying the clusters without significant loss in the interpretation of the procedures. In the case of Machine Learning tools, this interpretation of the procedures is lost. However, there is an improvement in terms of the predicted power of the algorithms.

In particular, under this particular data set, all the machine learning methods outperform the best statistical model calibrated in part 1 that was used as the benchmark model. However, this improvement is not enough to consider the models as satisfactory in the case of the classification of wine categories into the groups `Low`, `Middle` and `High` quality.

As a consequence, techniques that increase the bias of the models, such as reducing noise by converting the original problem to one of binary classification where the main objective is to distinguish wine of high-quality and cost-sensitive learning are used.

Another important aspect that should be considered in the model calibration process is how much global information is available in order to fit the best possible model. In this case, and focusing on the binary classification process, the first estimator is using the Bayes Rule with a threshold of 0.5. This scheme works reasonably well when the groups to be classified are balanced and the cost of classification errors is reasonably similar.

However, we can get a better threshold that provides the least amount of misclassifications. In this case, the ROC curve offers a representation of the best threshold between sensitivity and specificity. If no more information is available (especially on the costs of errors), this may be the appropriate model.

Nevertheless, in case there is information on the possible costs, the algorithms can be modified so that the hyper-parameters point towards the optimization of the profit. In this particular case, the Random Forest model with the parameter given by the curve, the sensitivity and specificity values are both above 80%. However, when the economic benefit is considered, we see that the cost of the wrong classification of regular wine has the highest cost so that the new scheme tends to minimize this error.

6. Note: GitHub Link

In my Github repository https://github.com/cconejov there is the detail of all the `R` raw code used in both projects. Especially, in the folder `scripts` is all the raw code that calibrates and train all the models.

The training time of the models was large, so in the `Report_Final_ML_Cesar_Conejo.rmd` file that creates the final report in pdf format that was uploaded in AulaGlobal calls three `.rda` objects with the trained models.

- `train_models_3class_ML.rda` for the 3-problem classification (optimization in function ofaccuracy).
- `train_models_2class_ML.rda` for the binary classification (Specificity).
- `train_models_2class_econProfit_ML.rda` for the binary classification (Economic Profit).

Also, the data set `wineQuality.rda` and the function `draw_functions.R` that represents graphically the confusion matrix is attached in the .rar object uploaded in Aula Global.