**A Comparison of Methods for Imbalanced Wine Quality Classification**

Stephanie Cunningham

October 2020

**Abstract**

Many data analysis tasks are complicated by a very common phenomenon, the tendency of samples to group toward certain values or classes which can result in imbalanced models if not properly considered. The Red Wine Quality dataset is an example of this; there are 1599 rows corresponding to wine samples, judged on an ordinal scale for quality, each accompanied by 11 columns of physiochemical characteristics. Naturally, most of the wines receive an average score, with only select samples earning a .7-.8 and a “fine” classification. In this analysis, domain knowledge of wine chemistry and model balancing techniques were combined to find the model with the best ability to distinguish “fine” from “regular” wines, primarily using precision and recall metrics due to the imbalance in classes. Two approaches were used on appropriate models for their implementation: threshold tuning was utilized to choose features in a logistic regression framework, and the Synthetic Minority Oversampling Technique (SMOTE) was used within cross-validation selection of random forests and extreme gradient boosted trees. Logistic regression, even after variable selection and tuning, showed inferior performance to the tree methods. Tree methods with classes fully balanced with SMOTE performed worse on test data than those with no oversampling, indicating overfitting was an issue, but the best model proved to be extreme boosted trees with the minority class oversampled only 300% with SMOTE. This model along with its equivalent random forest showed that the best predictors for fine wine quality were alcohol and sulfate levels.

1. **Introduction**

The evaluation and grading of wines is a time-honored tradition which only the best palettes in the world are fully qualified to tackle. There are many compositional qualities which contribute to the unique and complex flavor of a wine, including acid, sugar, tannin, and alcohol levels. However, the true standard for a fine wine is more difficult to measure, and traditionally determined by an exclusive group of judges who undergo rigorous training. For example, the certification for “Master Sommelier,” which costs several thousand dollars to take and much more to train for, has a 95% fail rate, over six times higher than that of a typical Medical Board Exam (Osborne et. al, 2020). There is not yet a machine which can recreate the process of determining wine quality, though advancements in machine learning and quality control technology could one day provide one. There are many potential uses for such technology, from training master sommeliers without the need for a prestigious institution to providing a model for winemakers to improve their formula by focusing on measurement of the key predictors.

The dataset to be analyzed, “Red Wine Quality,” contains ordinal quality scores for Portuguese “Vinho Verde” red wines, along with physiochemical metrics for each. This data was originally published in the 2009 study “Modeling wine preferences by data mining from physicochemical properties” by Cortez et. al, and has since been made publicly available on the UCI Machine Learning Repository for further analysis. This research will focus on classification techniques, specifically those for overcoming class imbalance, in order to fit the best possible model for mimicking the wine judging process. Due to the differences between wines, models will likely be applicable mainly to the wine variety on which they are trained, and so applying the techniques developed to new varieties will require a new training set. In general, the distribution of quality scores for any group of wines will likely have similar class imbalance problems. Ideally, a technique for automatic judging of wines would take all of this into account, and allow for such data inconsistencies while accurately judging wines. This project compares several classification methods, correcting for class imbalance where needed using Synthetic Minority Oversampling Technique (SMOTE) and threshold optimization where needed, in order to find the most promising model for the advancement toward automated wine judging.

1. **Methods**

Initial exploration and pre-processing were performed on the “Red Wine Quality” data in R (R Core Team, 2020). The data contains 1599 rows of wines, each corresponding to 11 compositional features and a quality score on a 10-point scale. Actual scores received range only from 3 to 8, with a class imbalance strongly favoring middling scores of 5 and 6 (Fig 1). The predictors are all continuous variables but are on various scales (see Appendix 1). Data was inspected for missing and duplicate values. No missing values were found but 240 rows were duplicated. Research indicated that these rows represent multiple evaluators scoring the same wine, and so they were left in the data in order to preserve as much information as possible. The data was checked for outliers by first inspecting the statistical summary of each variable (Table 1). Three features, residual sugar, total sulfur dioxide, free sulfur dioxide, and chlorides were examined as boxplots to visually examine outlier positioning (see Fig. 2). Because of the naturally skewed distribution of these variables, each had several points considered to be outliers. Of these, total sulfur dioxide (TSO2) and chlorides had the outliers of most concern, with two points in each category being far removed from the rest of the samples (approximately double the value of the next closest point for TSO2 and approximately 50% higher than the next closest point for chlorides). However, research suggests that these values are scientifically reasonable and can even help explain their respective quality scores, and so the outliers were not removed from the data. The distribution and correlation of the individual features was examined via graphical matrix (Fig. 3) and correlation heatmap (Fig. 4).

Next, the data was reconfigured to create a binary classification problem. Wines in the highest two categories, 7 and 8, are considered to be fine wines and given a label of 1, while all others are grouped together and labeled as 0. Class imbalance was still a potential problem, with 217 fine wines and 1382 others. This can lead to issues with comparison of metrics and the learning of certain models, which may tend to ignore the minority class in favor of increasing accuracy by over-favoring the majority class. There are various techniques available to correct for class imbalance, and the solution should be tailored to the use case. One method is adjustment of the classification threshold to account for the tendency of the classifier to choose the majority class. Another solution, which can be set to automatically detect and balance the minority class, is oversampling, or one of the hybrid methods thereof. One of these is SMOTE, which was built to overcome some of the overfitting problems of basic oversampling. New data points are synthesized by connecting existing sample points with “k” nearest neighbors and creating new data along the connection lines (Chawla et.al., 2002). It is important to mention a few of the major drawbacks of SMOTE, however. Firstly, it cannot create new minority class points outside the interpolation range of the existing samples, and therefore this research must rely on the assumption that the existing samples are fully representative of their classes. Secondly, the method is not shown to work well with high-dimensional data except with the use of Euclidean K-nearest neighbors classifiers (Blagus et. al., 2013). This is not of particular concern here, as the data has only 11 features. Thirdly, it violates the independent samples assumptions for certain classifiers, such as logistic regression, and pre-processing activities such as subset selection.

Considerations were made in the model-building plans to avoid data leakage, assumption violations, and sampling imbalance. Firstly, prior to any oversampling, a training/test split was made with 80% of the data used for training the models and 20% retained as a test set for prediction only. However, simple random sampling would not appropriately capture the class imbalance present, so stratified sampling was used in order to retain the original proportions in each set. Secondly, oversampling results in breaking sample independence. Therefore, it should not be used with models that make this assumption, such as logistic regression. Research also indicates that because subset selection generally assumes sample independence, oversampling should not be performed until after subset selection. This complicates the pipeline, because feature selection using traditional cross-validation and validation set methods both risk data leakage into downstream steps, unless a separate pre-processing set is withheld. To avoid such pitfalls, considerations for models to use with SMOTE were limited to those also considered feature selection methods, and make no assumptions regarding independence between samples. Random forest was chosen. Lastly, model hyperparameters were selected using repeated 10-fold cross-validation, but here lies a common mistake when employing SMOTE –synthesized data points should never be used for testing or validation, so these were excluded from the validating folds. Here, the caret package was useful in automating this process via the trainControl() function to which the parameter “sampling” can be specified with a variety of techniques, including SMOTE (Maxwell, 2019). The package offers automatic grid search, though custom grids can be created as well.

As mentioned previously, logistic regression is not an ideal model to utilize with SMOTE because it assumes independent samples. The threshold for classification was instead varied to account for the class imbalance. Logistic regression is advantageous for its interpretability, though feature selection is not inherent. All of these points made it a good contrast for the models which can be corrected with SMOTE and also select their own best features.

The major metric used for model comparison was Precision-Recall area under the curve (PR-AUC), together with the balancing of precision and recall themselves along with the F-measure. Precision is a measure of how many “positives” (or here, fine wines) are in the set relative to how many were selected by the classifier. Recall is the proportion of the selected positives among the actual positives in the data. AUC measures the area under the curve for these metrics when plotted for each possible probability threshold, while F measures the precision/recall balance at the chosen threshold. In this case, the class imbalance signals that the PR-AUC was a more helpful metric than the receiver operating characteristic (ROC) AUC, which is useful in classification problems where the classes are approximately equal. Due to the very low proportion of fine wines in relation to regular wines, it was more telling of a model’s usefulness to focus on these true positive metrics. Furthermore, PR-AUC from cross-validation of training data (used to select hyperparameters) must be compared with test-set PR-AUC (and this metric used to select the best model) to ensure overfitting does not drive model selection, and that data leakage has not occurred.

The models compared for classification were logistic regression with feature selection and threshold tuning and random forest and extreme gradient boosted trees with SMOTE. Logistic regression was implemented inside of the caret package recursive feature elimination (rfe) wrapper; RFE is a type of backward subset selection. PR AUC was used as the selection metric for the best model, and F-measure was used to select an appropriate threshold. This allowed controlled cross-validated feature selection for direct comparison to the other models. Two specific interactions were included for consideration in addition to the original predictors: pH interacting with alcohol and residual sugar, separately. This is because of two known interaction effects of acidity with each of these. Alcohol and acids will produce esters, which are beneficial in small amounts but at higher levels produce a “vinegar” defect. If there is too much acid without sufficient residual sugar, the wine will be overly tart (Corbett and Corbett 2007).

Logistic regression model was run with the standard classification threshold (0.5 for both classes), and then PR-AUC curve was examined and classification threshold adjusted to fight class imbalance and improve F-score. The full model followed by full model with two interaction terms was run through recursive feature selection, followed by a model without interaction terms but a potentially collinearity problem removed (based on domain knowledge and results of Spearman correlation comparison (Fig. 4) and VIF comparison (Table 3).

Random forest was implemented in three ways to compare overall PR-AUC: ignoring class imbalance, using class-balanced SMOTE, and using an unbalanced SMOTE model where the minority “fine” class was up-sampled only 300%, which resulted in this class being near a quarter, instead of a full half, of the data. This was to help to prevent overfitting. Random forest does not assume independence among samples and is considered to automatically “feature select” though not in the traditional sense that subset selection does. Random forest is based on bagging and performs a high volume (number obtained by cross-validation) of iterations, tested by training on “in bag” and testing on “out of bag” samples, and fitting to each point in the sample by combining all of its in-bag prediction scores. The models were run with importance = T in order to obtain the OOB permutation importance metrics (Mean Decrease Accuracy), shown as relative importance in Fig. 6, which are generally a better indicator of variable importance as they are less overfit to training data than metrics produced only from in-bag data (Gini Index). Interaction terms do not need to be added separately to tree models. Variables were further analyzed with partial dependence plots.

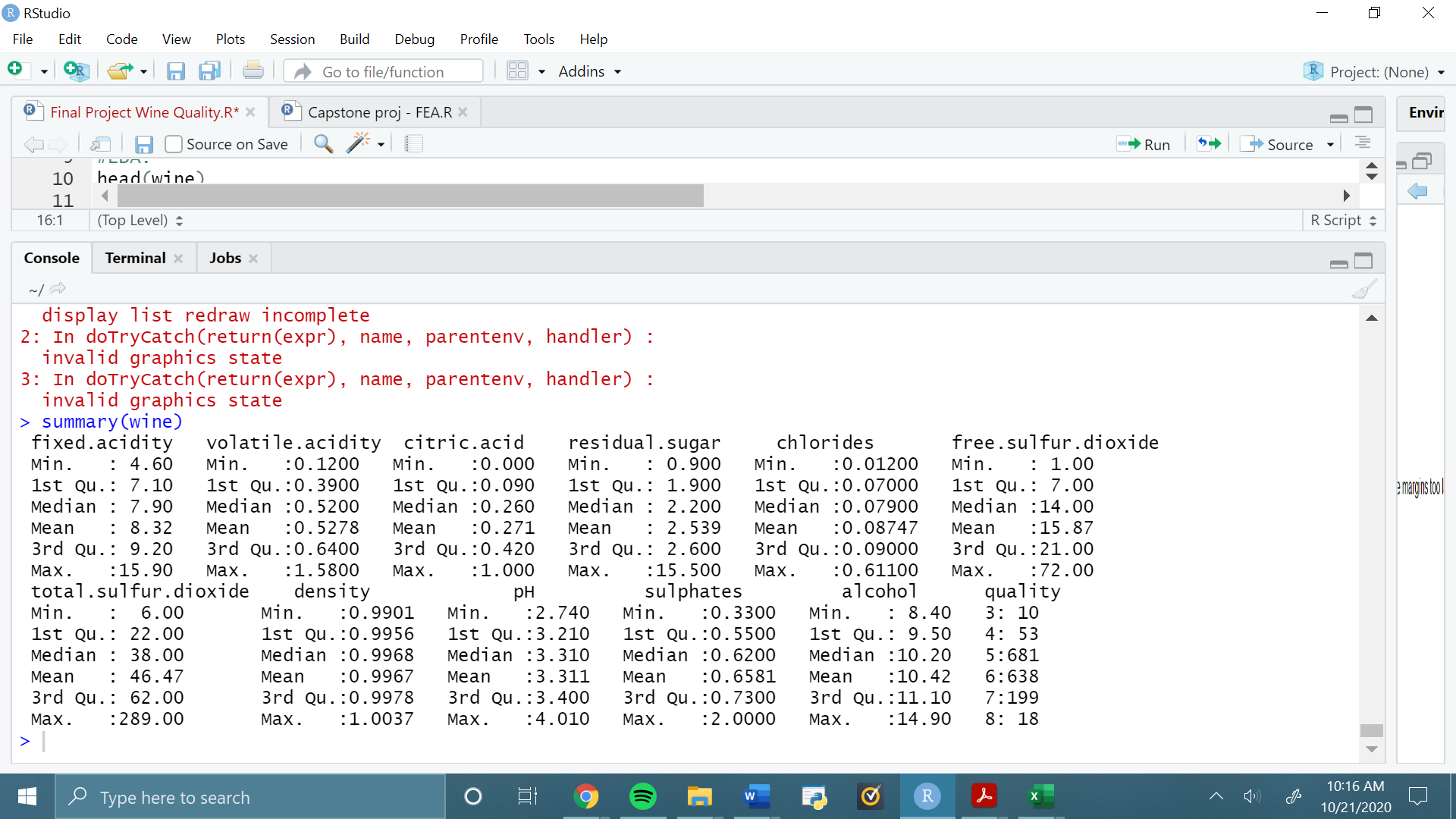
Finally, the results were compared to one of the most popular tree methods being used for model improvement today: extreme gradient boosting. This method can be used with a tree framework, among others, and was run with the unbalanced SMOTE scheme in an attempt to improve classification ability.

1. **Results**

**Exploratory Data Analysis and Pre-Processing**

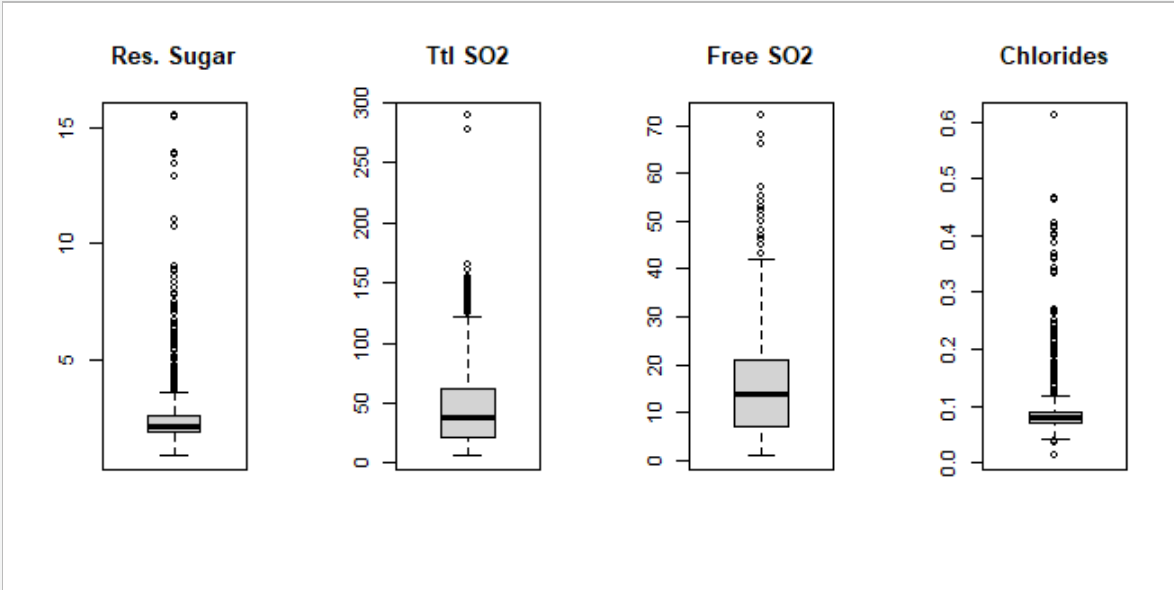
Results of exploratory data analysis showed that minimal data pre-processing was needed, as all values are present and outliers identified in Table 1 and Fig. 2 are within reasonable ranges. However, classes are severely unbalanced (Fig. 1) and many variables are non-normal (Fig. 3). Fig. 4 shows that several of the variables have the potential for multicollinearity problems. Table 2 shows the binary distribution once the classes were established (still very unbalanced).

**Table 1.** Summary statistics for unprocessed wine data

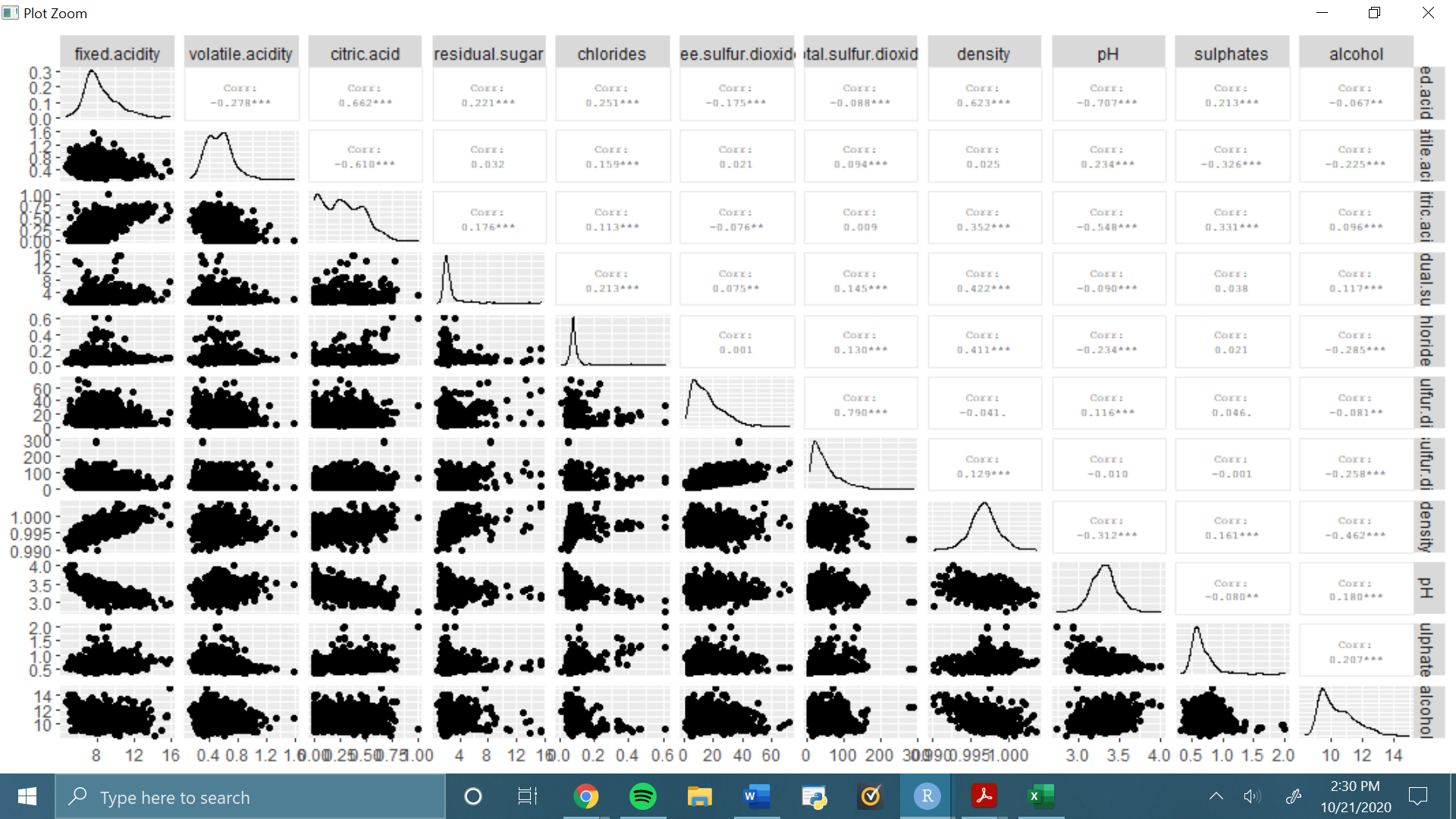




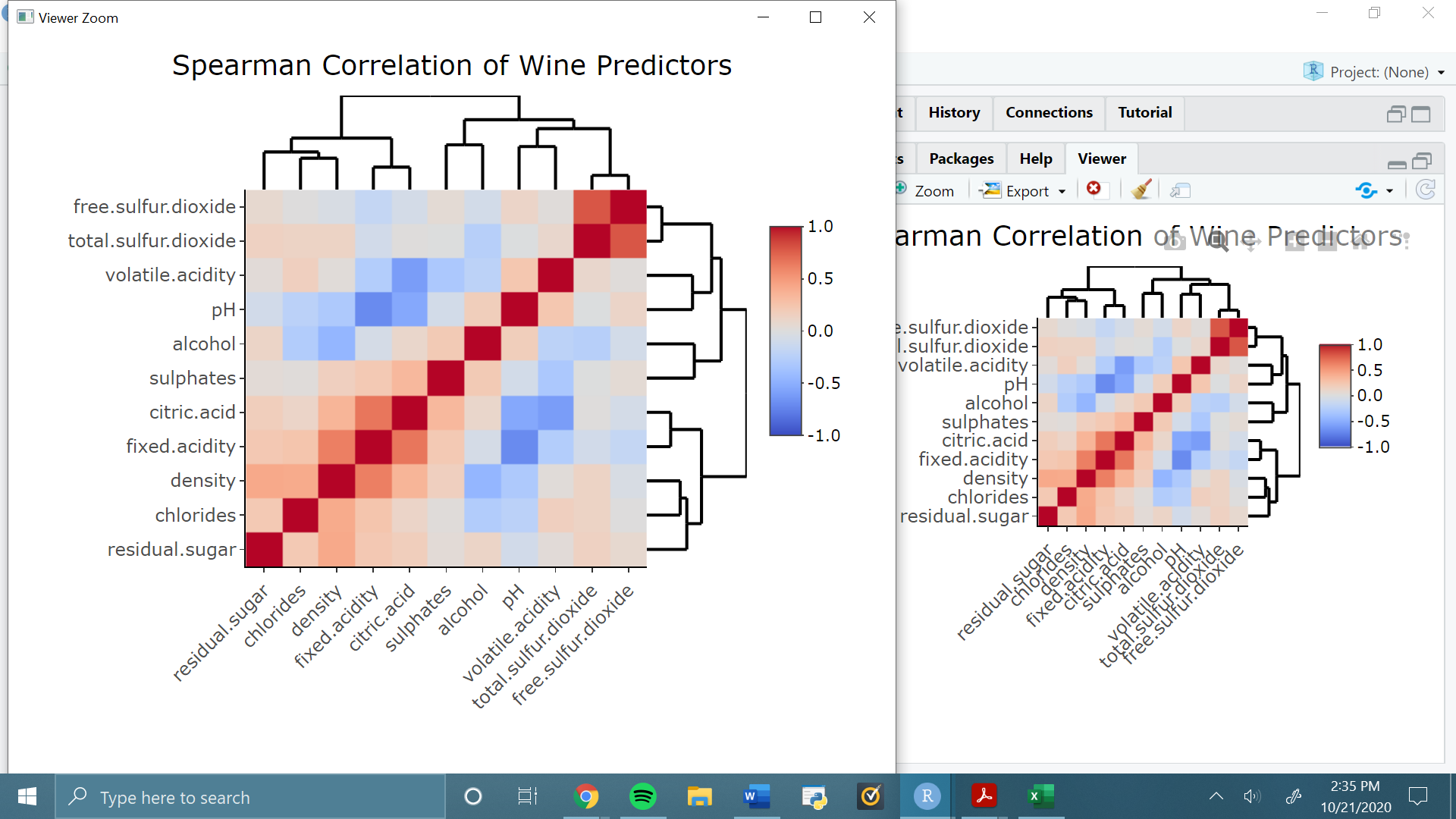
**Figure 1.** The distribution of quality scores shows a heavy imbalance toward scores 5 and 6, while 3, 4, and 8 are severely underrepresented.



**Figure 2.** Outlier exploration was performed based on inspection of the wine data summary. Interestingly, upon examination the most significant outliers for free SO2 are for high rated wines, both given a 7. In contrast to free SO2 (FSO2), total SO2 (TSO2) levels may not be perceptible to the drinker. Neither TSO2 outlier is in the major outliers for FSO2, meaning most of the molecules are bound up, masking other imperfections in the wine. While the wine may have been overprocessed, the outliers are both under the allowable TSO2 level of 350 ppm (Moroney, 2017) and upon closer inspection it seems likely that these measurements are likely correct so no action is needed. The other set of outliers that may be of concern are in the chlorides column, which consist of two data points slightly higher than .6 g/L chlorides. This is high for the dataset but not outside the realm of possibility; allowable levels vary by country but, for example, in Australia the allowed level is .607 g/L (Coli et. al., 2015). Though the data points slightly exceed, this, it could be contributing to their lower quality scores (5 and 4 for highest and second highest, respectively), and so it is reasonable to assume that these measurements are not erroneous.



**Figure 3.** Distribution and spatial relationships of each variable show that many of the features are non-normally distributed. It also shows some potentially related variables, like fixed acidity vs citric acid, density, and pH, and free vs. total SO2.



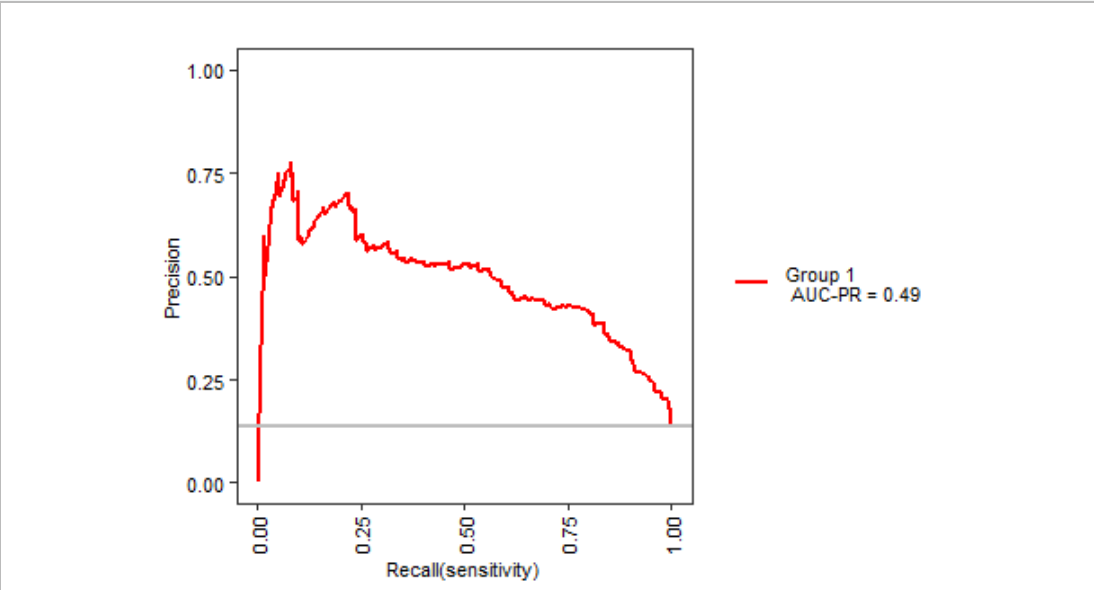
**Figure 4.** Spearman correlation of predictors was examined in order to identify variables which may be too strongly related and lead to multicollinearity in certain models. Of particular note are the strong associations (>.7) between total and free SO2 levels, between fixed acidity and pH, and between citric acid and volatile acidity.

**Table 2.** Class counts for binary wine quality groupings

|  |  |
| --- | --- |
| Class | Count |
| Fine | 217 |
| Regular | 1382 |

**Logistic Regression and Feature Selection**

Logistic regression with RFE chose all of the variables including interactions (13 total) in terms of highest PR-AUC. Originally F-value was extremely low at .4181, as the minority class was not being selected often. The threshold for classification was adjusted to .75 based on precision and recall statistics. The AUC-PR curve for Logistic Regression Model #1 is shown in Fig. 5. The logistic regression model with the same probability threshold, but removing fixed acidity based on VIF assessment (Table 3) actually decreased PR-AUC and F values for sample data (.4740 and .5385, respectively). See Appendix B for full results.



**Figure 5.** PR-AUC curve for RFE logistic regression model #1.

**Table 3.** VIF scores for full variable logistic regression model.

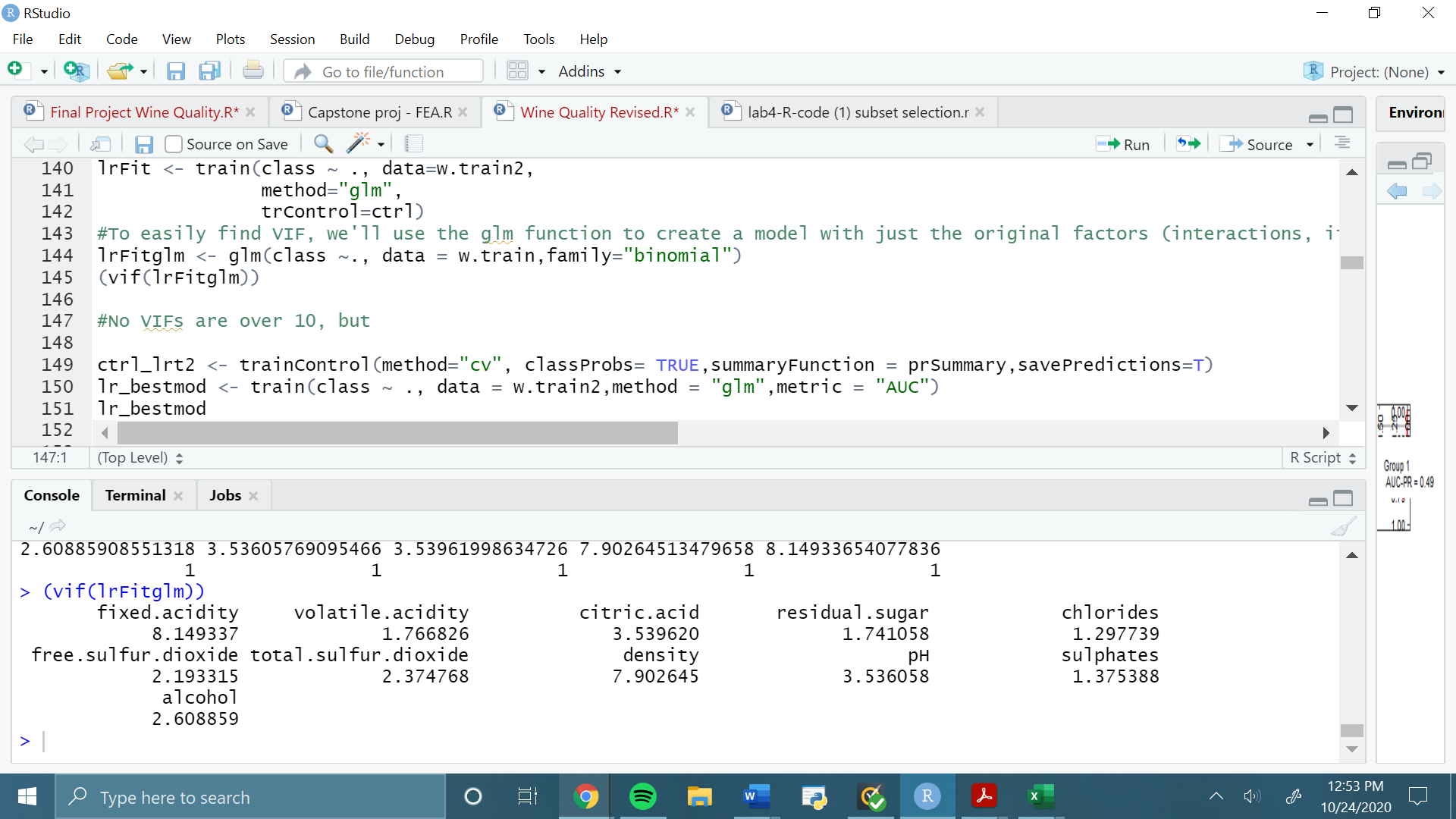
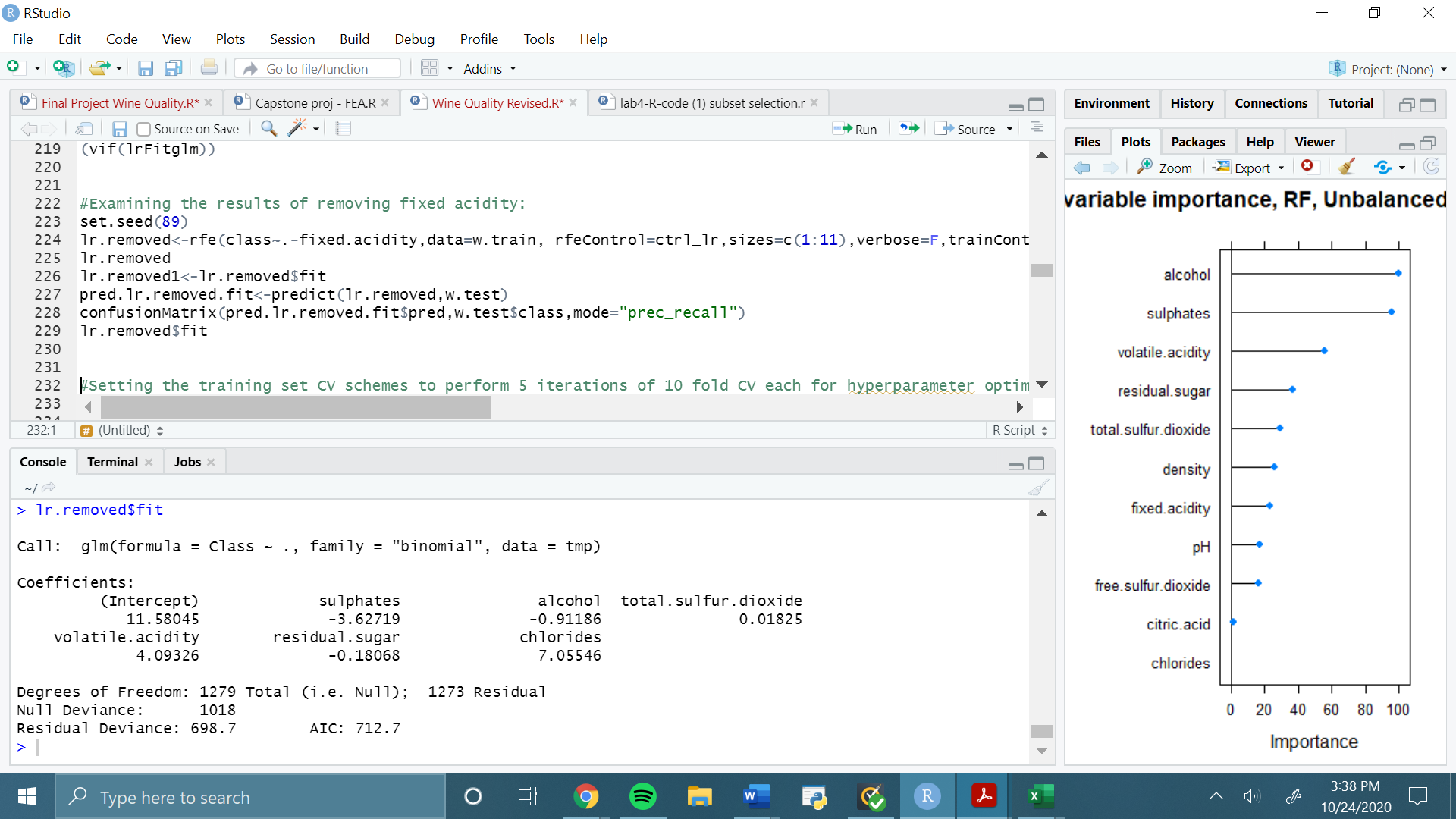


Table 3 shows that no variance inflation factor is higher than 10. There is no “rule” as to which variables to delete, but as fixed acidity and density are both very high in VIF and both are correlated with each other, it is reasonable to test a model with one of these removed (fixed acidity was chosen as it is also correlated with other predictors).

Though training data indicated that the original logistic regression with all predictors and interactions but a heightened threshold to .75 was the best model, upon inspection of testing predictions, the reduced model actually showed a slightly higher F-value (Table 4). See Appendix B-C. for model summaries for training and testing data, respectively.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table 4.** Comparison of logistic regression models with test set predictions | | | | |
|  | LR Model #2 -8 predictors | | |  |
|  |  | Actual | |  |
|  |  | Fine | Regular | F1= 0.3809 |
| Predicted | Fine | 12 | 31 |  |
| Regular | 8 | 268 |  |
|  | LR Model #2 - all predictors and 2 interactions (13) | | |  |
|  |  | Actual | |  |
|  |  | Fine | Regular | F1= 0.338 |
| Predicted | Fine | 11 | 11 |  |
| Regular | 32 | 265 |  |
|  | LR Model #3 - threshold .75, 8 predictors | | |  |
|  |  | Actual | |  |
|  |  | Fine | Regular | F1= 0.4946 |
| Predicted | Fine | 23 | 27 |  |
| Regular | 20 | 249 |  |
|  | LR Model #4 - Fixed acidity removed, threshold .75 | | |  |
|  |  | Actual | |  |
|  |  | Fine | Regular | F1= 0.5102 |
| Predicted | Fine | 25 | 30 |  |
| Regular | 18 | 246 |  |

Based on testing results the logistic regression model best fitting the data was the reduced model which had 6 predictors. Fig. 6 shows the model summary with coefficients.



**Figure 6.** Model summary for best logistic regression model (6 predictors). These can be compared to the important variables resulting from random forest.

**Random Forest and SMOTE**

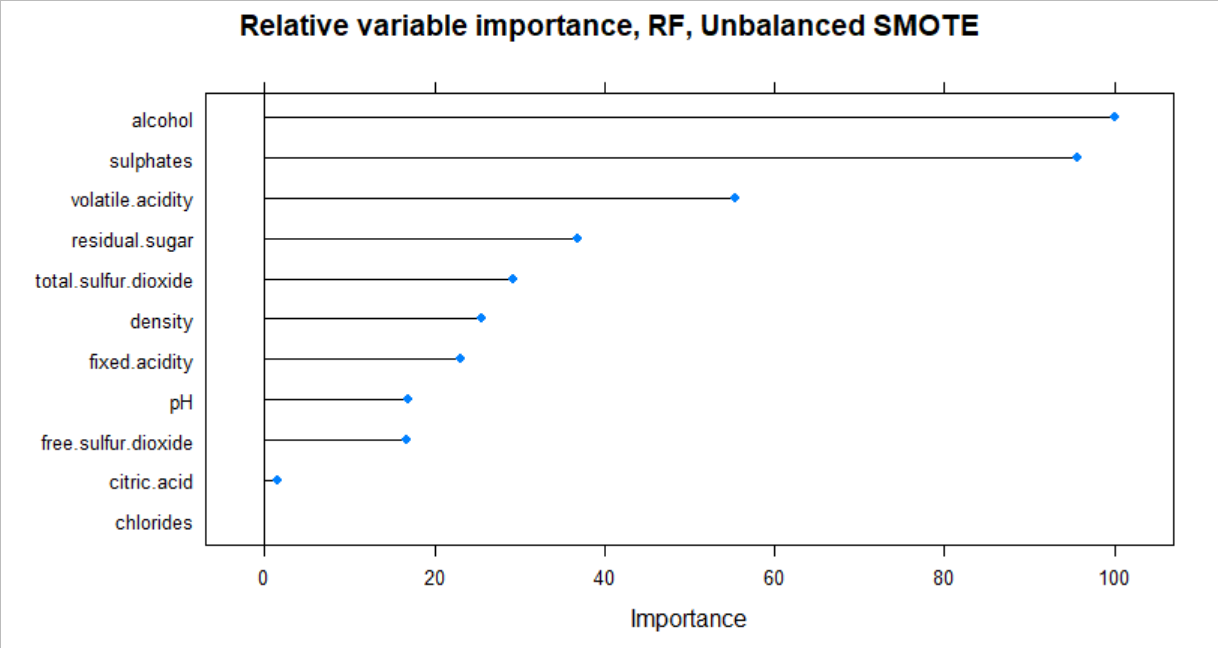
Random forest performed better than logistic regression in general. Table 5 shows the testing results for each of the methods.

**Table 5.** Comparison of random forest models with test set predictions

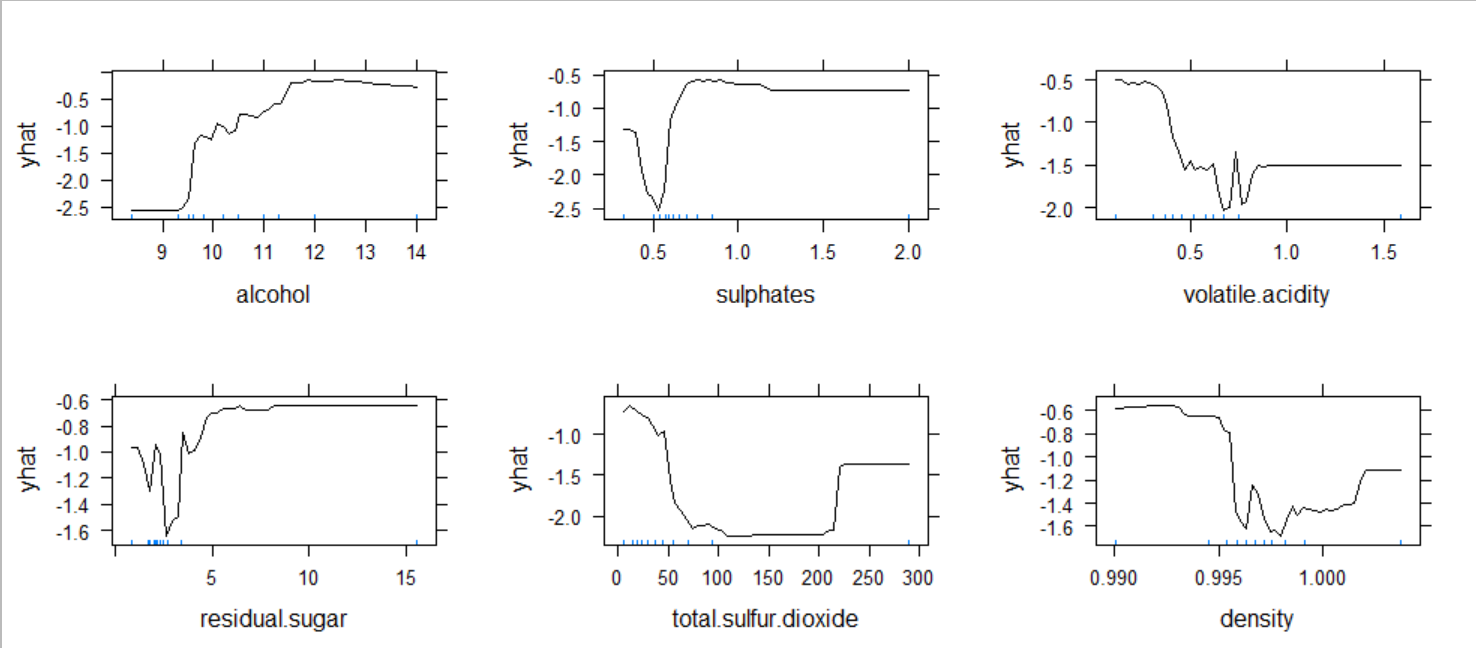
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | RF Model #1 - No oversampling | | |  |
|  |  | Actual | |  |
|  |  | Fine | Regular | F1= 0.5856 |
| Predicted | Fine | 19 | 3 |  |
| Regular | 24 | 273 |  |
|  | RF Model #2 - SMOTE balanced | | |  |
|  |  | Actual | |  |
|  |  | Fine | Regular | F1= 0.5253 |
| Predicted | Fine | 26 | 30 |  |
| Regular | 17 | 246 |  |
|  | RF Model #3 - SMOTE unbalanced | | |  |
|  |  | Actual | |  |
|  |  | Fine | Regular | F1= 0.5909 |
| Predicted | Fine | 26 | 19 |  |
| Regular | 17 | 257 |  |

As Table 5 exhibits, the unbalanced SMOTE technique (oversampling the fine class by 300%) was the best model in terms of precision/recall harmony, but in test set conditions it did not significantly improve upon the F1 value of the “balance ignored” method (RF Model #1), because it only correctly identified only 7 additional fine wines, and incorrectly placed an additional 16 regular wines into the “fine” category. It may also be noted that the accuracy measure for RF Model #1, though not the main metric, was the highest of any model at 0.91, as was Cohen’s kappa statistic at 0.5429; RF Model #3 was second best in these as well with accuracy of 0.8871 and kappa of 0.5255. Accuracy is not an ideal metric for unbalanced classes because, as seen in Model #1, models will gain accuracy by making too many majority prediction. Kappa, while related to accuracy, is somewhat more interesting as it relates accuracy to random chance. Both of the values listed can be interpreted as moderate in terms of success.

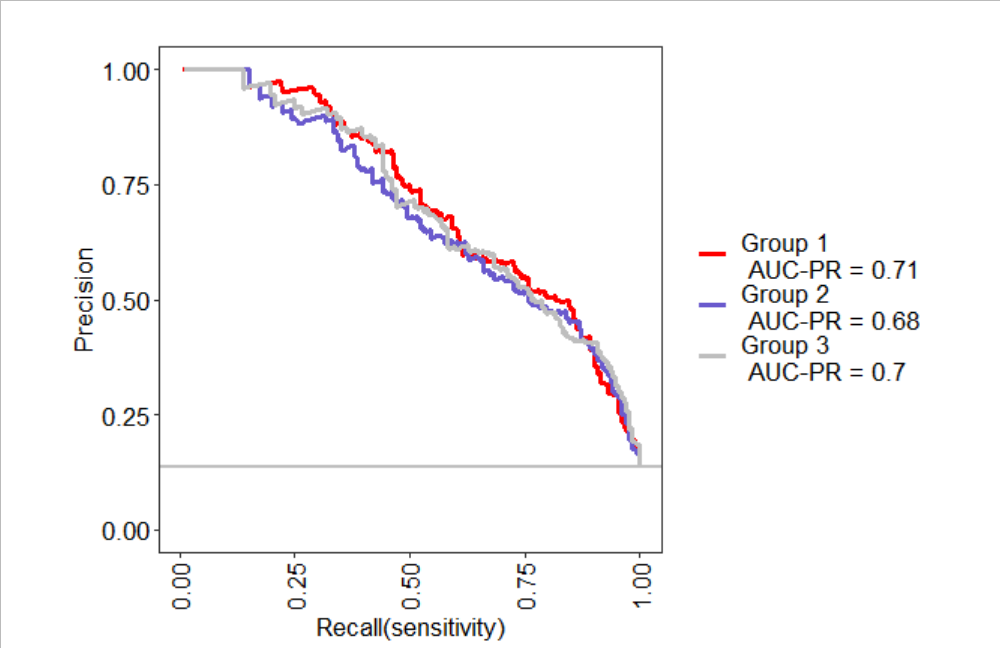
The variable important metrics of RF Model #3 show that alcohol and sulfates are most important to fine/regular classification, followed by volatile acidity (Fig. 7). Recall from Fig. 6 that these three variables were included in the best logistic regression model, along with TSO2, residual sugar, and most interestingly, chlorides, which are given no importance in the random forest model.



**Figure 7.** Permutation importance metrics for all variables passed to the RF Model #3.



**Figure 8.** Partial dependence plots for top variables from top six variables in RF Model #3. Y-axis represented logit of prediction score, with the “fine” class being the positive direction. X-axis units can be referenced in Appendix A.



**Figure 9.** Comparison of AUC-PR for RF Models #1,2,3 (corresponding to Group 1,2,3) with the MLeval package.

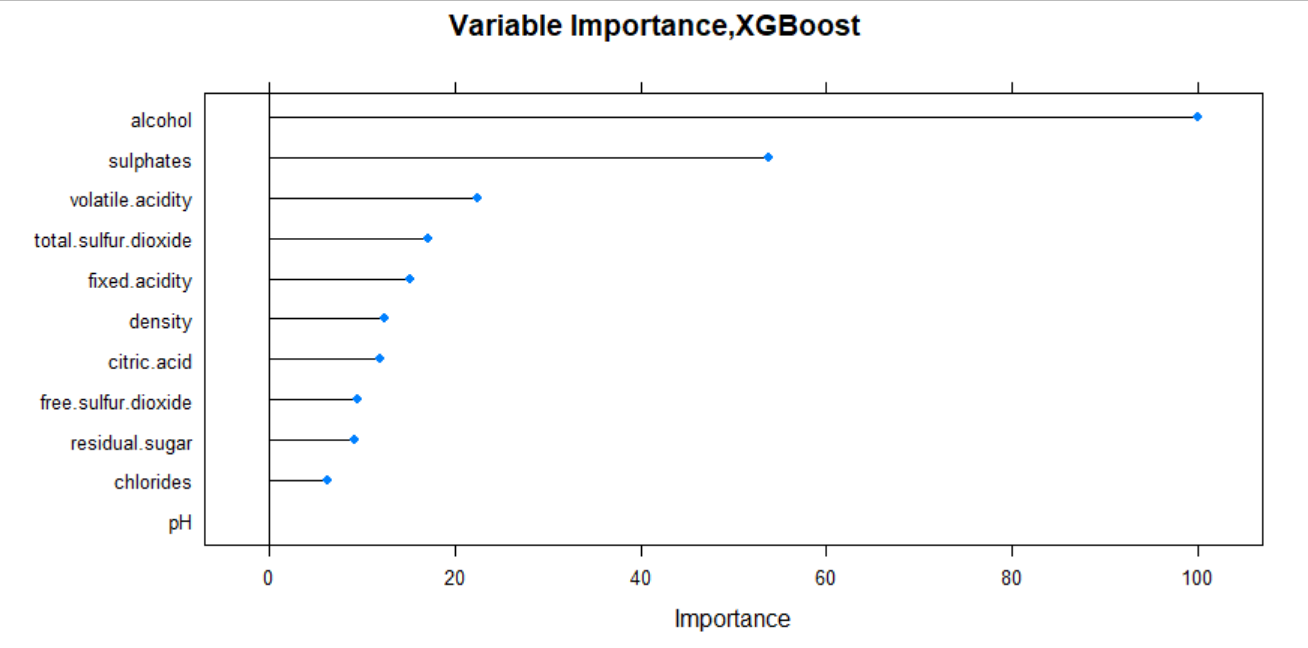
Figure 9 shows another comparison of AUC-PR using the MLeval package in accordance with the caret results. Though AUC-PR calculated by caret is highest for RF Model #3, MLeval actually gives a slight advantage to RF Model #1. Model #2 is at the clear disadvantage between the methods, most likely due to overfitting or noise introduction from overzealous application of SMOTE in order to balance the classes.

**Extreme Gradient Boosting**

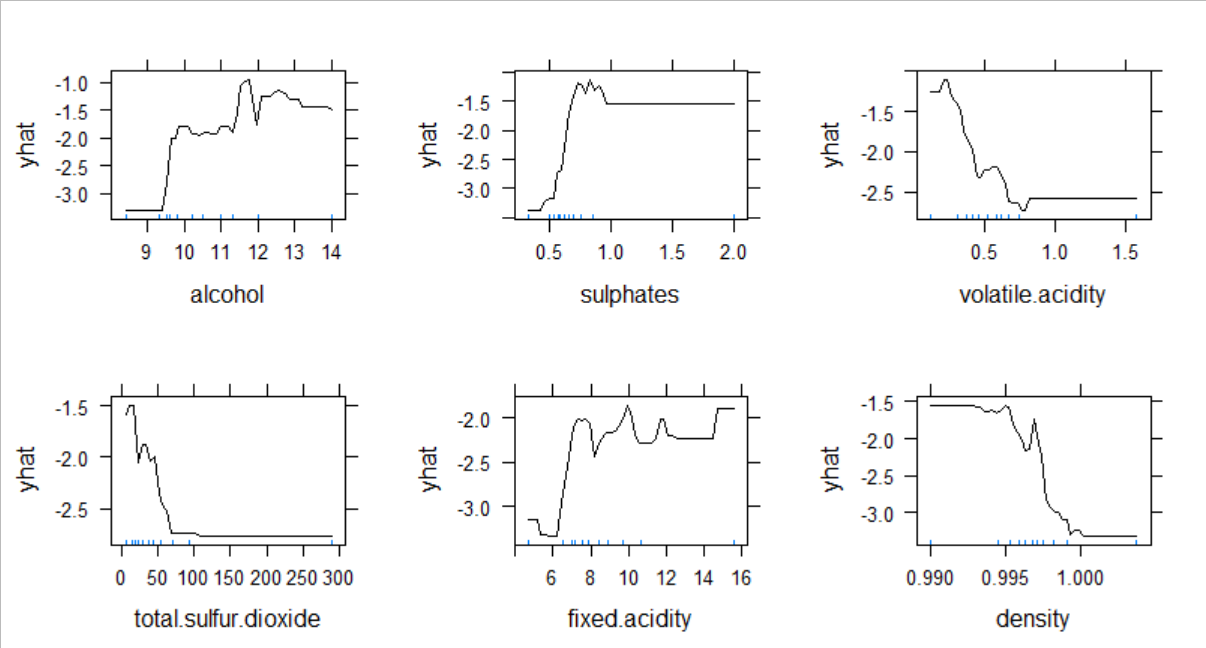
Extreme Gradient Boosting was used to build upon the relative success of the random forest model. The non-oversampling method which ignores class imbalance managed results similar to the RF model with unbalanced oversampling (RF Model #3), actually improving on test set F1 and accuracy, though the total number of correct “fine” placements was still lower (22 vs 26 in RF Model #3). However, the XGBoost Tree model using unbalanced SMOTE was the best of the models tried. The PR-AUC from the training set did not achieve levels as high as those for RF Models #1 & 3, but the testing accuracy (0.8966), kappa (0.5692); see Appendix C., and F1 scores (>0.63) were all the highest of any model (Table 6 displays test set predictions, which significantly improve upon the random forest models).

Table 6. XGBoost Testing Results

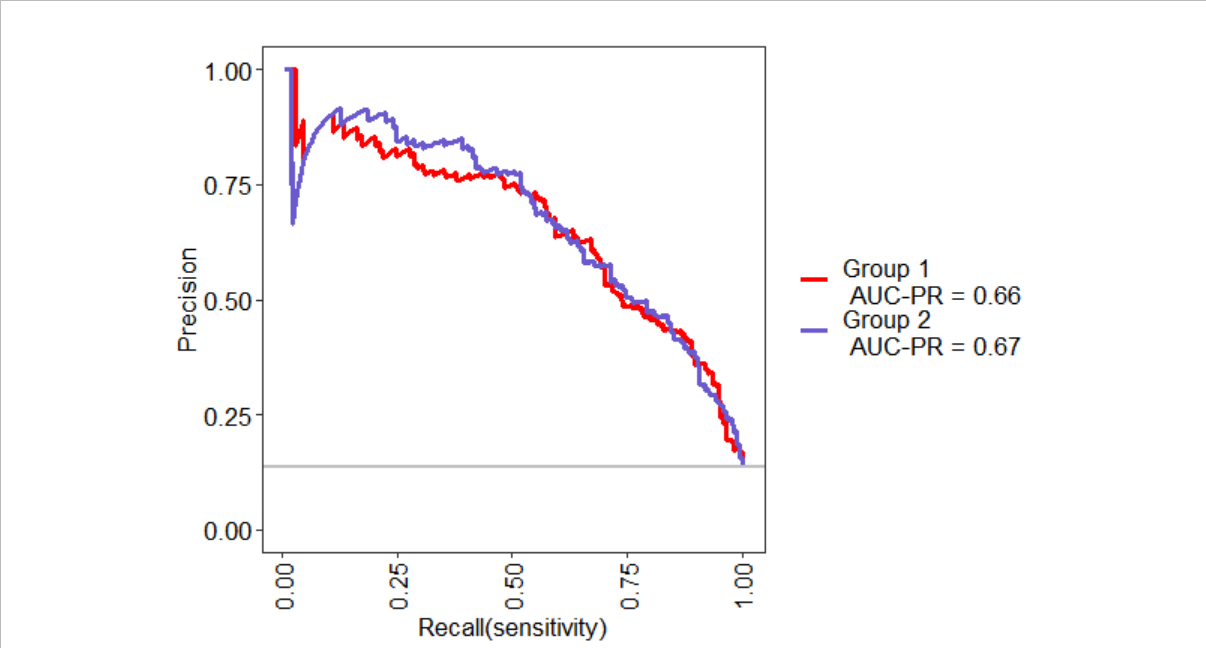
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | XGBoost Trees #1- No oversampling | | |  |
|  |  | Actual | |  |
|  | Class | Fine | Regular | F1= 0.5946 |
| Predicted | Fine | 22 | 9 |  |
|  | Regular | 21 | 267 |  |
|  | XGBoost Trees #2- SMOTE unbalanced | | |  |
|  |  | Actual | |  |
|  | Class | Fine | Regular | F1= 0.62921 |
| Predicted | Fine | 28 | 18 |  |
|  | Regular | 15 | 258 |  |



**Figure 10.** Variable importance plot for XGBoost. Note similarity to variable importance plot for RF Model #3.



**Figure 11.** Partial dependence plots for XGBoost with unbalanced SMOTE.



**Figure 12.** The MLeval precision-recall curves for XGBoost models #1 and 2 (Group 1 and 2, respectively)

Fig. 12 shows a lower overall AUC-PR for training data than the RF models, possibly because of a dip in precision between 0 and 0.25 true positive rate (sensitivity). However, these models are less biased towards the training data than the RF or logistic regression models and both do better on the test set. It can also be observed from Fig. 12 that the unbalanced SMOTE model (#2) does better at most cutoff levels than the non-oversampled model.

1. **Discussion**

This analysis highlighted the stark contrast between methods which ignored the class imbalance, giving the models high accuracy but poor precision-recall balance. In logistic regression, increasing the threshold for classification, as expected, helped to balance the classes such that the fine category was not overwhelmed in favor of the more “likely” class, regular. The increase in threshold resulted in a much higher number of true “fine” rulings in each case, but it also increased the false “fine” rulings in an expected trade-off.

The best models in each category were mostly in agreement on the best predictors for quality class. Particularly, random forest and extreme boosted trees models with unbalanced SMOTE showed very similar variable importance plots (Figs. 7 and 10, respectively), with the same order for top 3, and 5 of the top 6 variables being the same. Variable importance measures for alcohol and sulphates were both very high, and according to the partial dependence plots (Figs. 8 and 11), both are roughly associated with higher quality at higher levels. Figures 8 and 10 examine the relationships of each of the top 6 predictors from RF Model #3 and XGBoost Trees Model #2, respectively, with the logit-scale prediction score (‘yhat’), the x-axis showing each continuous variable’s range of occurrence. The positive direction represents score closer to the “fine” classification. This shows some interesting relationships between the predictors and outcome. The top 3 predictors are almost identical in both models. Alcohol, the most significant in terms of variable importance, becomes important around 9.5%, with a steep upward incline and a noisy upward climb between 10 and 12%, where it gradually starts to show a declining positive impact at higher levels. Sulphates are mostly associated with higher grades except in the RF model, where there may be some noise below the 0.5 g/L level. Volatile acidity has a mostly negative relationship with quality, but is most harmful to quality at certain mid-range levels, leveling out above about 0.8 g/L. Note that volatile acidity represents acetic acid in g/L, a gaseous acid resulting in a noticeable “vinegar” profile. The negative association makes sense, but it is possible that the mid-range levels interact with other aspects of the flavor more negatively than do higher levels. Residual sugar, on the other hand, has an approximately opposite effect, but with higher scores at very low levels of sugar than around 2.5 g/L, where the association dips down briefly. This could be due to low levels indicating a very clean wine without the need for sugar addition, and at higher levels, more sugar helps to cover other undesirable traits like volatile acidity. In the XGBoost model, residual sugar is not even in the top 6 predictors, however. This model instead has total sulfur dioxide in the 4th slot. TSO2 is another interesting variable – recall from Figure 2 that there are only 2 data points above 200 mg/L, where the negative effect reverses and shows an increase in the RF model, which seems to have been smoothed out in the XGBoost model. Sulfur dioxide serves to prevent oxidation, but high levels of TSO2 usually indicates a lot of sulfur has been added over the winemaking process, often meaning there is some problem causing levels to drop. It is unclear whether these points constitute a larger trend or whether this is simply noise, but it is possibly due to the fact that higher TSO2 levels are resulting in more stability for the free SO2 , which is itself perceptible in higher concentrations (Moroney, 2017). Instead of residual sugar in the top 6, XGBoost has fixed acidity in slot #5. There is a similar profile to that of alcohol, increasing sharply at around 6 g/L and remaining higher at higher levels, with some dips as well. Fixed acidity is focused on tartaric acid; in contrast to acetic acid measured in volatile acidity, this more associated with being a “good” acid flavor – tartaric is an acid naturally found in grapes. Finally, it appears that lower density in red wines has a positive impact on the quality. Interestingly, the scores are lowest near the density of water (.997 g/L) in both models, though the RF model shows a slight increase after this point that is smooth in the XGBoost model.

Overall, tree methods performed significantly better than logistic regression, but too much oversampling negatively impacted overfitting, so a non-balanced oversampling scheme or changing the threshold as was done in the logistic regression scheme is preferred. Random forest models, while quick to train and easy to tune, do not achieve perfect classification, and arguably RF Model #1 could be better than RF Model #3, depending on the use case and whether it is more desirable to misclassify a fine wine as regular (in which case Model #1, with no oversampling, is preferred), or a regular wine as fine (as discussed, Model #3 puts more of the fine wines into the correct category, but a greater number of regular wines as well). This illustrates the issues associated with classification trade-off. Also illustrated in the findings is the advantage of utilizing gradient boosting, which was able to achieve better all-around classification success than any of the other methods, with a balanced precision and recall as well as reasonably good accuracy.

Further research may focus on gathering more data and possibly using more powerful methods such as a neural network. A model to be used with actual automated wine judging technology might require these steps in order to be sufficiently accurate and precise.

**References**

Blagus, Rok, and Lara Lusa. “SMOTE for High-Dimensional Class-Imbalanced Data.” *BMC Bioinformatics*, vol. 14, no. 1, 2013, doi:10.1186/1471-2105-14-106.

Chawla, N. V., et al. “SMOTE: Synthetic Minority Over-Sampling Technique.” *Journal of Artificial Intelligence Research*, vol. 16, 2002, pp. 321–357., doi:10.1613/jair.953.

Cobett, Jan, and Tom Cobett. “Better Wine through Chemistry.” *Presque Isle Wine Cellars*, American Wine Society, 2007, www.piwine.com/media/home-wine-making-basics/better\_wine\_through\_chemistry.pdf.

Coli, Marina Sonegheti, et al. “Chloride Concentration in Red Wines: Influence of Terroir and Grape Type.” *Food Science and Technology (Campinas)*, vol. 35, no. 1, 2015, pp. 95–99., doi:10.1590/1678-457x.6493.

Cortez, Paulo, et al. “Modeling Wine Preferences by Data Mining from Physicochemical Properties.” *Decision Support Systems*, vol. 47, no. 4, 2009, pp. 547–553., doi:10.1016/j.dss.2009.05.016.

Maxwell, Aaron. *Machine Learning with Caret*, 10 Nov. 2019, www.wvview.org/spatial\_analytics/caret/\_site/index.html.

Moroney, Maureen. “Total Sulfur Dioxide - Why It Matters, Too!” *Midwest Grape and Wine Industry Institute*, 28 Feb. 2017, www.extension.iastate.edu/wine/total-sulfur-dioxide-why-it-matters-too.

Osborne, G., et al. “Court of Master Sommeliers: Reviews of Wine Certification Programs.” *SOMM*, 20 June 2020, www.somm.us/court-of-master-sommeliers/.

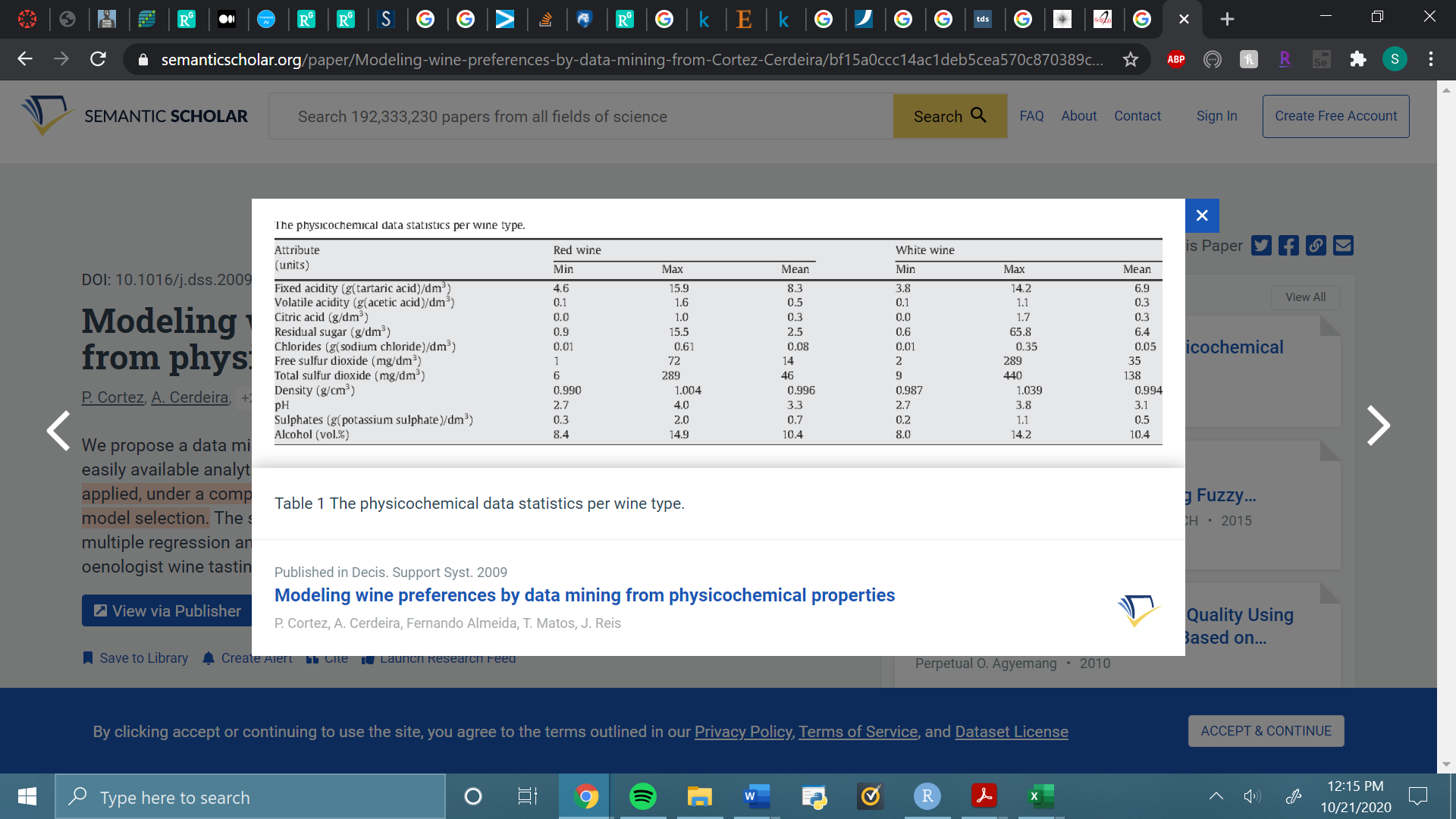
R Core Team (2020). R: A language and environment for statistical computing. R

Foundation for Statistical Computing, Vienna, Austria. URL

https://www.R-project.org/.

**Appendix A.**

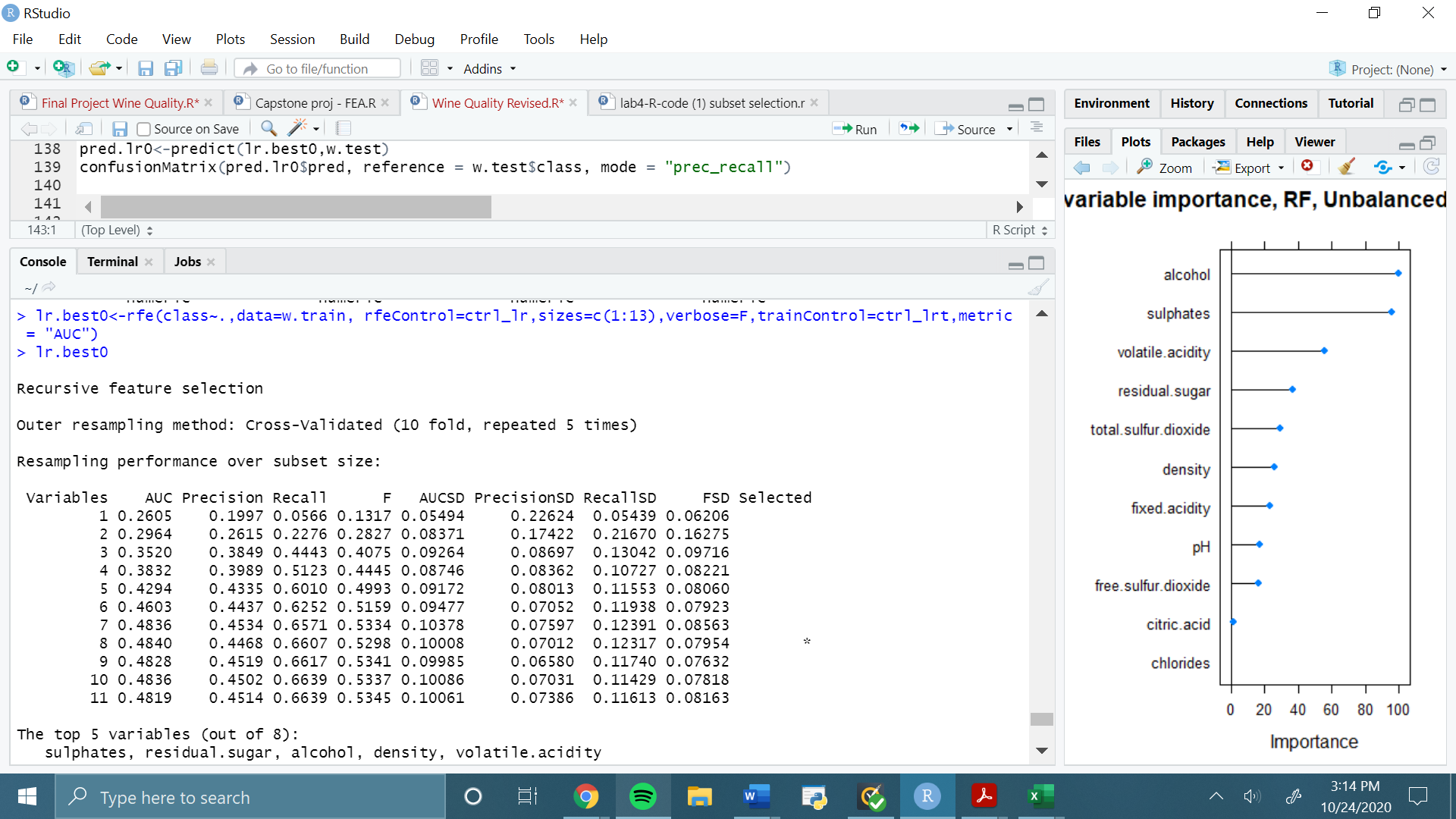
Table 1 from Cortez et. al (2009); used to reference the feature units.



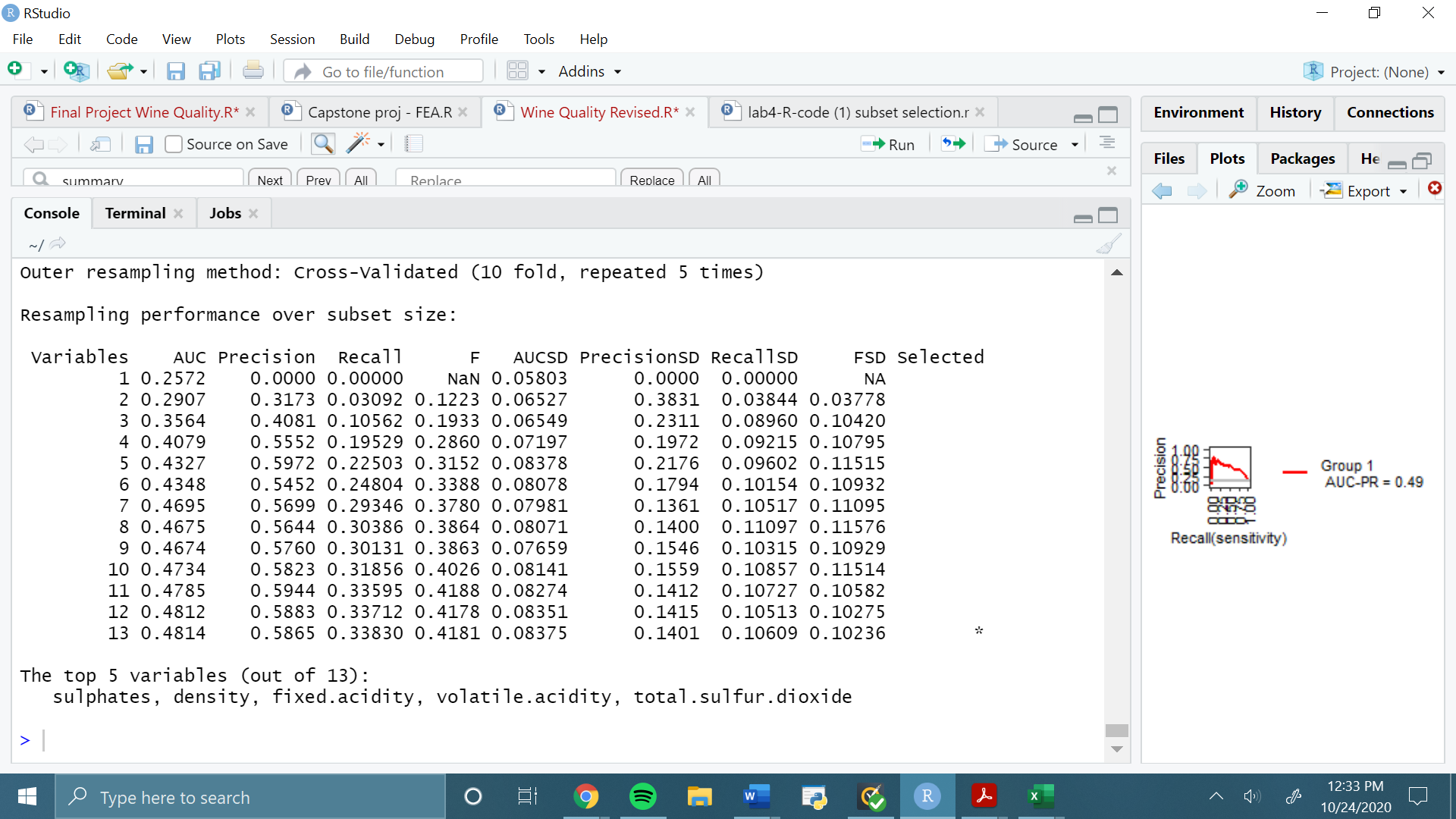
**Appendix B.**

Full summary statistics (cross-validated Precision-Recall measures) for each model tested:

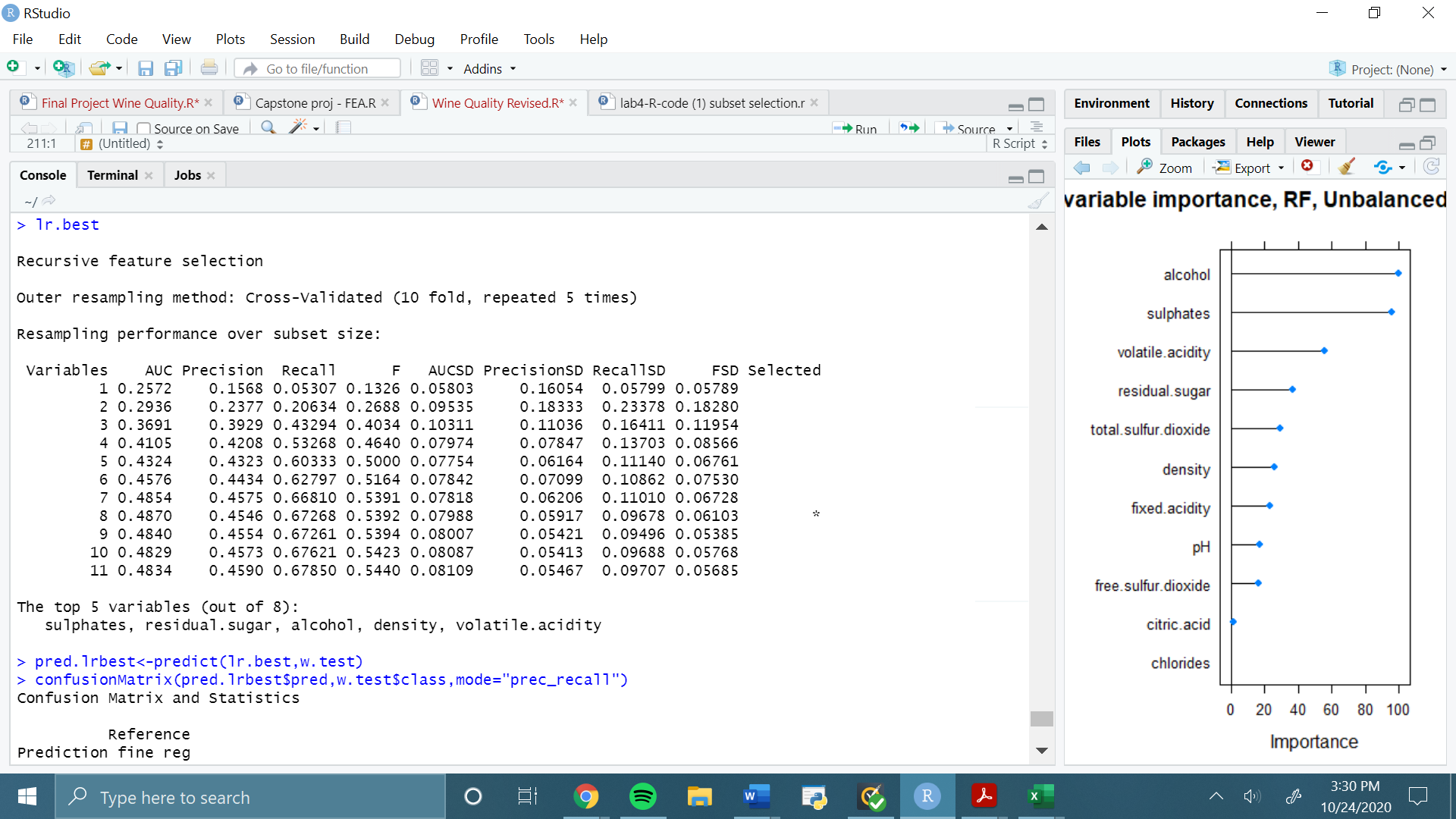
Logistic Regression, RFE, threshold >.5; no interaction terms:



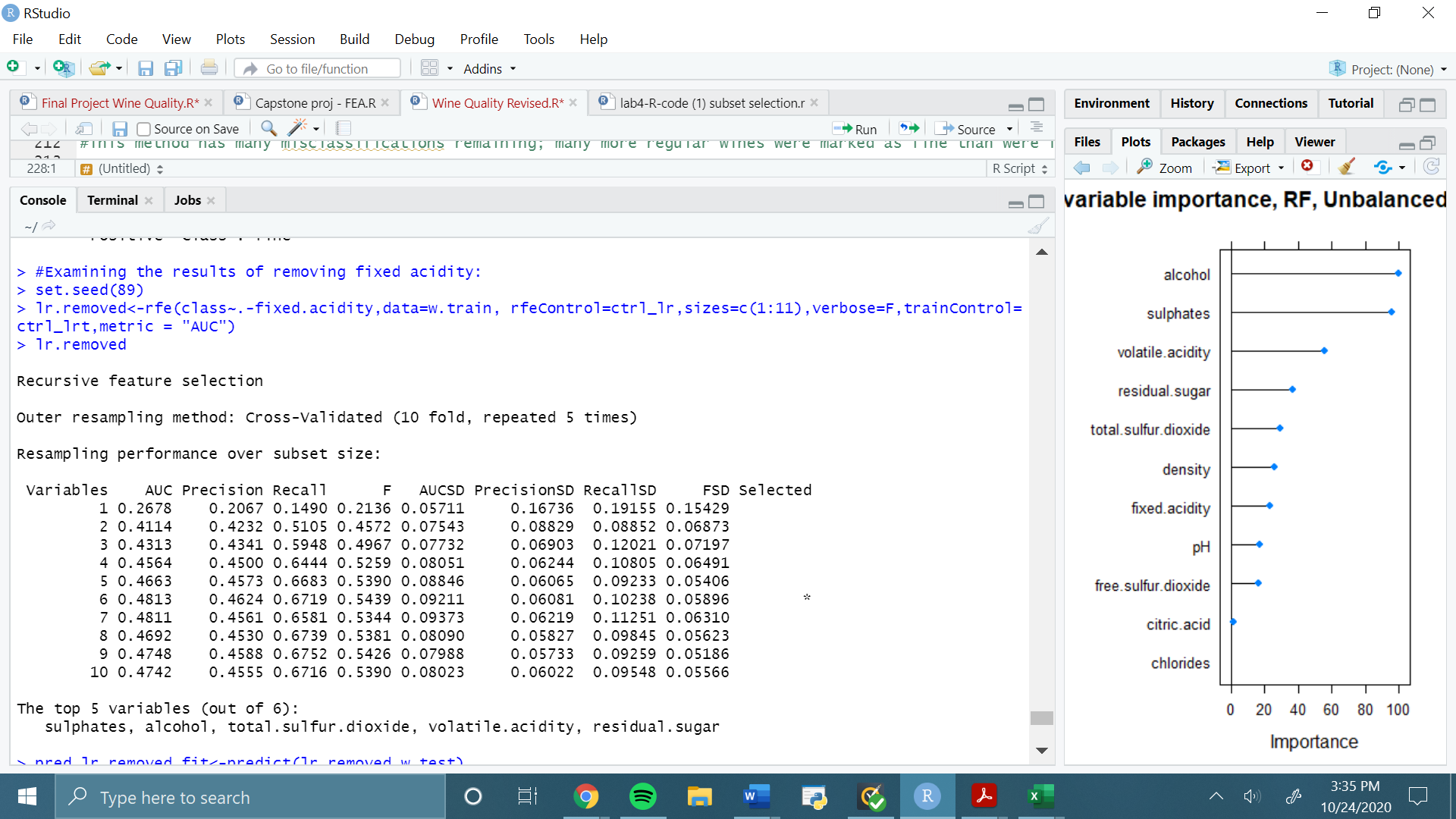
Logistic Regression, RFE, threshold >.5; two interaction terms:



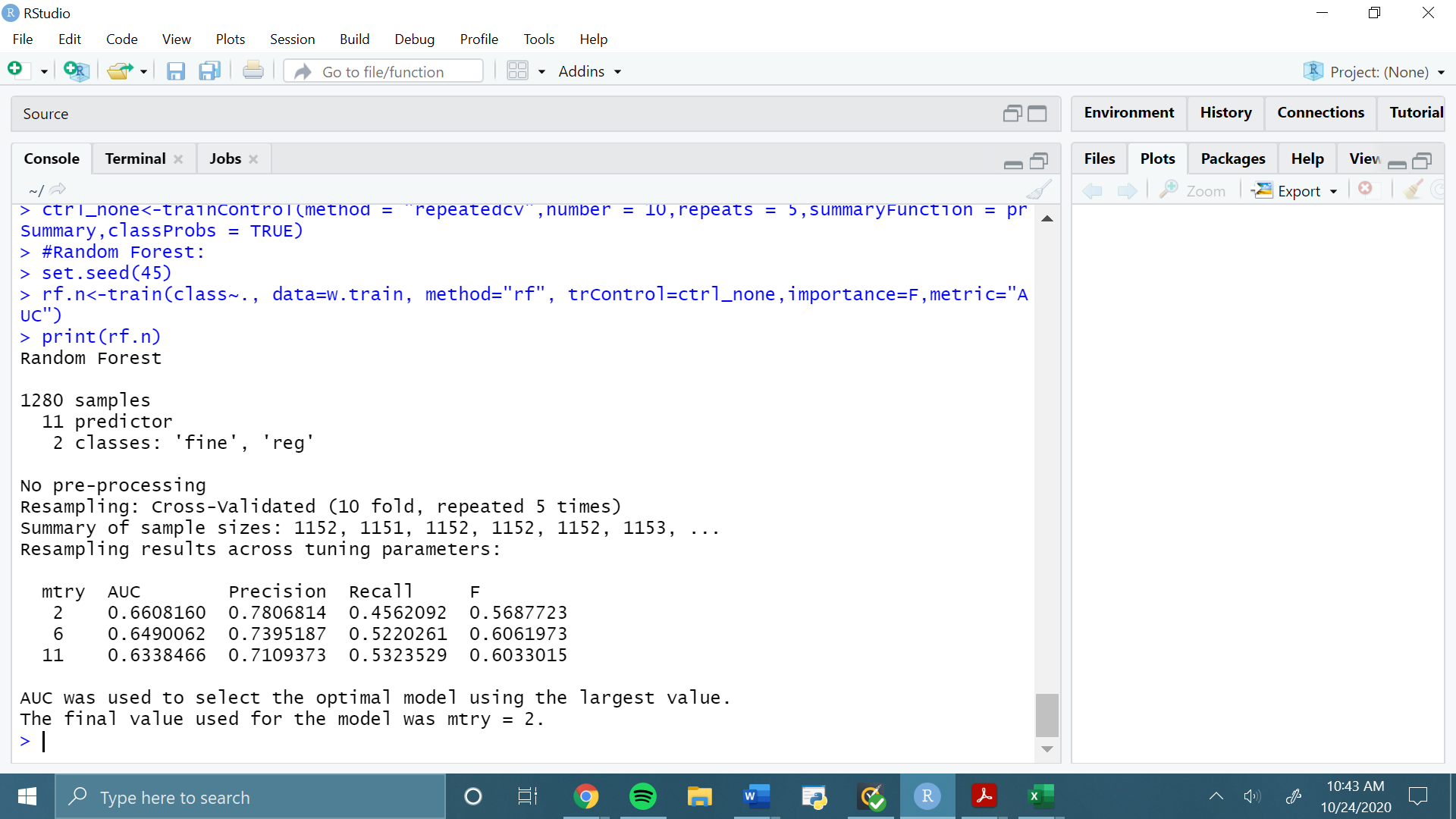
Logistic regression, RFE, threshold >.75; 8 predictors (no interactions considered):



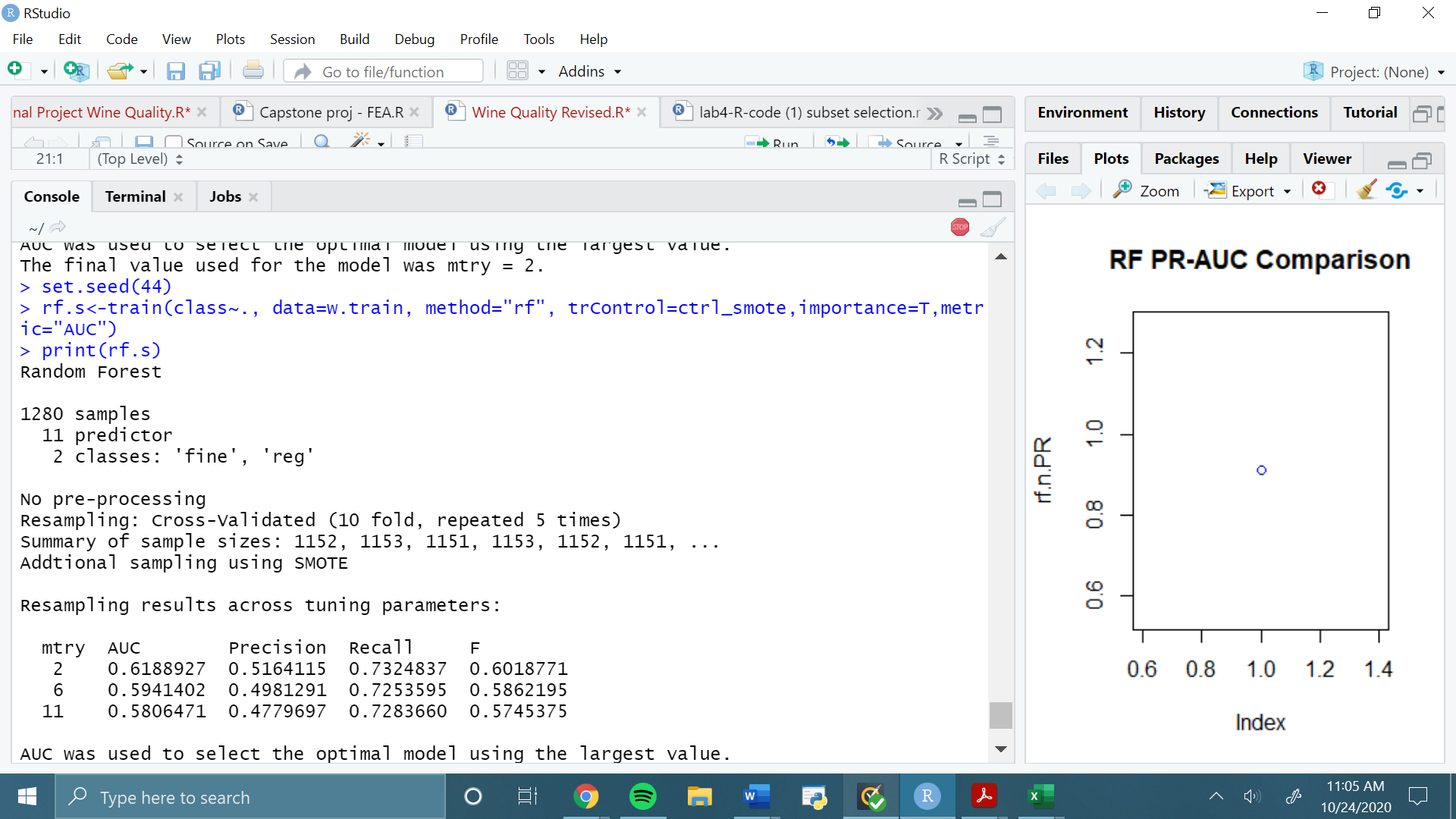
Logistic Regression, Multicollinearity removed, RFE, threshold >.75:



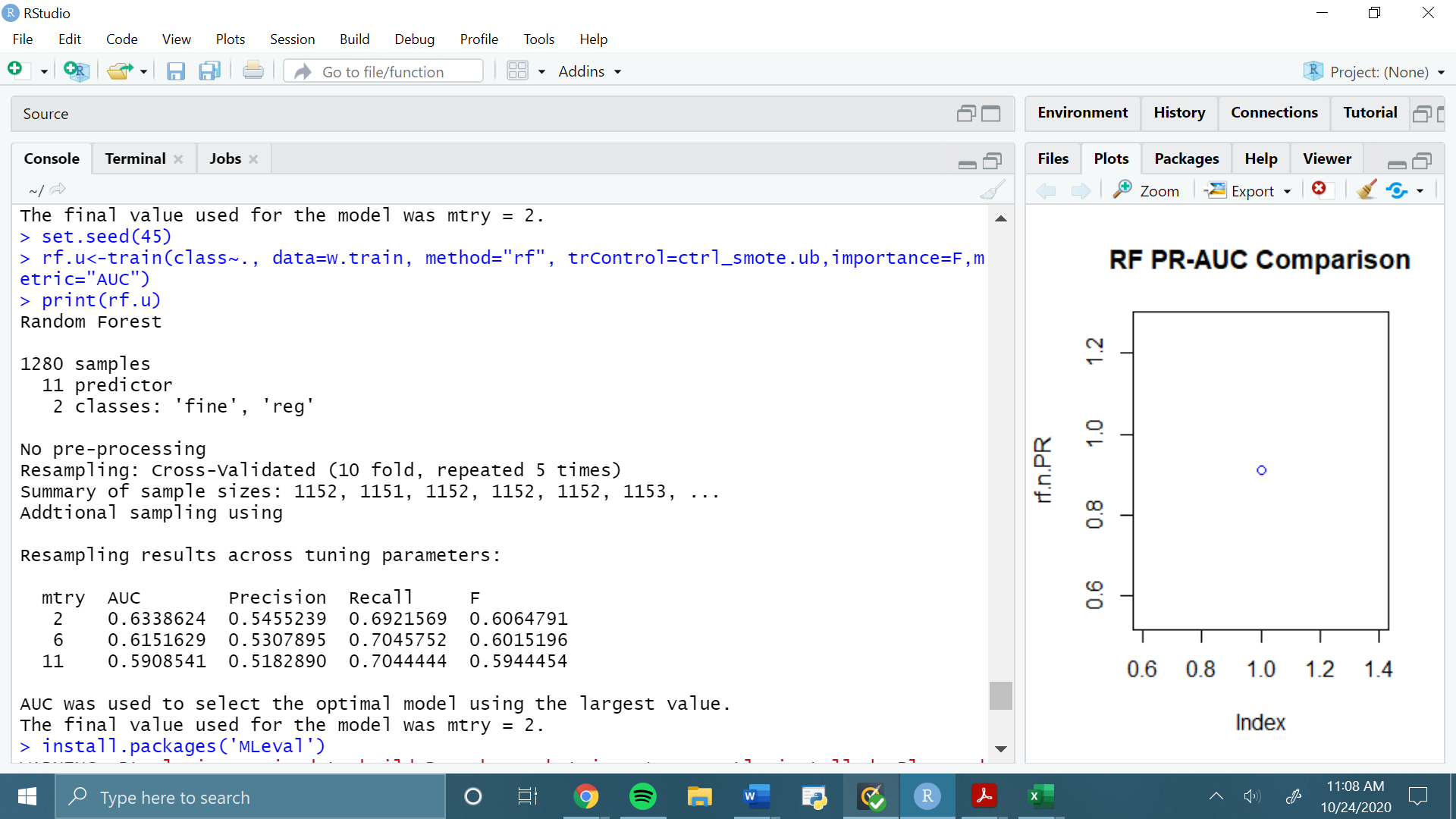
Random forest, no oversampling:



Random Forest, balanced SMOTE:



Random Forest, unbalanced SMOTE:



Extreme Gradient Boosting, no oversampling:

eXtreme Gradient Boosting

1280 samples

11 predictor

2 classes: 'fine', 'reg'

No pre-processing

Resampling: Cross-Validated (10 fold, repeated 5 times)

Summary of sample sizes: 1152, 1151, 1151, 1153, 1152, 1152, ...

Resampling results across tuning parameters:

eta max\_depth colsample\_bytree subsample nrounds AUC Precision Recall F

0.3 1 0.6 0.50 50 0.5199853 0.6129676 0.3366667 0.4246195

0.3 1 0.6 0.50 100 0.5283393 0.5939828 0.3691503 0.4452303

0.3 1 0.6 0.50 150 0.5320587 0.6025045 0.3901307 0.4649166

0.3 1 0.6 0.75 50 0.5174121 0.5884187 0.3126797 0.3996770

0.3 1 0.6 0.75 100 0.5264541 0.5976047 0.3495425 0.4315765

0.3 1 0.6 0.75 150 0.5223586 0.5850559 0.3729412 0.4484922

0.3 1 0.6 1.00 50 0.5198419 0.6204900 0.3060131 0.4024758

0.3 1 0.6 1.00 100 0.5232988 0.6067568 0.3369281 0.4260637

0.3 1 0.6 1.00 150 0.5231982 0.6074143 0.3567320 0.4417723

0.3 1 0.8 0.50 50 0.5070871 0.6003879 0.3516340 0.4350298

0.3 1 0.8 0.50 100 0.5174274 0.5819844 0.3860131 0.4580986

0.3 1 0.8 0.50 150 0.5285899 0.5809957 0.3922222 0.4619333

0.3 1 0.8 0.75 50 0.5170391 0.6040800 0.3281699 0.4168518

0.3 1 0.8 0.75 100 0.5200799 0.5870762 0.3541830 0.4326275

0.3 1 0.8 0.75 150 0.5255885 0.5844103 0.3636601 0.4400138

0.3 1 0.8 1.00 50 0.5246855 0.6290852 0.3118301 0.4081981

0.3 1 0.8 1.00 100 0.5241948 0.6057032 0.3418301 0.4287394

0.3 1 0.8 1.00 150 0.5205970 0.5940584 0.3520915 0.4349421

0.3 2 0.6 0.50 50 0.5372881 0.6198659 0.4109804 0.4862284

0.3 2 0.6 0.50 100 0.5588202 0.6288332 0.4577124 0.5228016

0.3 2 0.6 0.50 150 0.5632705 0.6371836 0.4977124 0.5515846

0.3 2 0.6 0.75 50 0.5403199 0.6028591 0.3930719 0.4684117

0.3 2 0.6 0.75 100 0.5572262 0.6418822 0.4601307 0.5277260

0.3 2 0.6 0.75 150 0.5711246 0.6556462 0.5073203 0.5646895

0.3 2 0.6 1.00 50 0.5318109 0.6334992 0.3943137 0.4781921

0.3 2 0.6 1.00 100 0.5450146 0.6474356 0.4422876 0.5157804

0.3 2 0.6 1.00 150 0.5611115 0.6597192 0.4939869 0.5573147

0.3 2 0.8 0.50 50 0.5300416 0.6383496 0.4105882 0.4919902

0.3 2 0.8 0.50 100 0.5467250 0.6302028 0.4760131 0.5360339

0.3 2 0.8 0.50 150 0.5573794 0.6240957 0.4999346 0.5497856

0.3 2 0.8 0.75 50 0.5325810 0.6259434 0.4108497 0.4882628

0.3 2 0.8 0.75 100 0.5529382 0.6299777 0.4692157 0.5301055

0.3 2 0.8 0.75 150 0.5650267 0.6368945 0.5091503 0.5594361

0.3 2 0.8 1.00 50 0.5364594 0.6567391 0.4154902 0.4989637

0.3 2 0.8 1.00 100 0.5552550 0.6646365 0.4611765 0.5348008

0.3 2 0.8 1.00 150 0.5615104 0.6697962 0.5062745 0.5690036

0.3 3 0.6 0.50 50 0.5621999 0.6478852 0.4624183 0.5314399

0.3 3 0.6 0.50 100 0.5787518 0.6599040 0.5230065 0.5783796

0.3 3 0.6 0.50 150 0.5867461 0.6713545 0.5277778 0.5852278

0.3 3 0.6 0.75 50 0.5686393 0.6637327 0.4816993 0.5515949

0.3 3 0.6 0.75 100 0.5909452 0.6772404 0.5382353 0.5934367

0.3 3 0.6 0.75 150 0.6035230 0.6803340 0.5522876 0.6036220

0.3 3 0.6 1.00 50 0.5623867 0.6547022 0.4739869 0.5410091

0.3 3 0.6 1.00 100 0.5904915 0.6897031 0.5290850 0.5907291

0.3 3 0.6 1.00 150 0.6050437 0.6785294 0.5359477 0.5920980

0.3 3 0.8 0.50 50 0.5613969 0.6384724 0.4667320 0.5317384

0.3 3 0.8 0.50 100 0.5841568 0.6707843 0.5228105 0.5798833

0.3 3 0.8 0.50 150 0.5883373 0.6747555 0.5471895 0.5977934

0.3 3 0.8 0.75 50 0.5813068 0.6685457 0.4911111 0.5608042

0.3 3 0.8 0.75 100 0.6023965 0.6785585 0.5356209 0.5916033

0.3 3 0.8 0.75 150 0.6140237 0.6866202 0.5470588 0.6021765

0.3 3 0.8 1.00 50 0.5604011 0.6608720 0.4750327 0.5447206

0.3 3 0.8 1.00 100 0.5912073 0.6909827 0.5252941 0.5884585

0.3 3 0.8 1.00 150 0.5997869 0.6882206 0.5288235 0.5910206

0.4 1 0.6 0.50 50 0.5122496 0.5996139 0.3611765 0.4416799

0.4 1 0.6 0.50 100 0.5178814 0.5806283 0.3818301 0.4534385

0.4 1 0.6 0.50 150 0.5263488 0.6036216 0.4220915 0.4892597

0.4 1 0.6 0.75 50 0.5176931 0.5918317 0.3496732 0.4325938

0.4 1 0.6 0.75 100 0.5206758 0.5892641 0.3692810 0.4456687

0.4 1 0.6 0.75 150 0.5301239 0.5953278 0.3828105 0.4584177

0.4 1 0.6 1.00 50 0.5213561 0.6051900 0.3256863 0.4149786

0.4 1 0.6 1.00 100 0.5210159 0.5978380 0.3475163 0.4324542

0.4 1 0.6 1.00 150 0.5242338 0.5963938 0.3696078 0.4488074

0.4 1 0.8 0.50 50 0.5116854 0.6013037 0.3640523 0.4471920

0.4 1 0.8 0.50 100 0.5248381 0.6030736 0.3830719 0.4618872

0.4 1 0.8 0.50 150 0.5281298 0.5894559 0.4074510 0.4749707

0.4 1 0.8 0.75 50 0.5084158 0.6046055 0.3528105 0.4391469

0.4 1 0.8 0.75 100 0.5211177 0.5873144 0.3598693 0.4391197

0.4 1 0.8 0.75 150 0.5230231 0.5948152 0.3807843 0.4566232

0.4 1 0.8 1.00 50 0.5199432 0.6128312 0.3290850 0.4197013

0.4 1 0.8 1.00 100 0.5227420 0.6019660 0.3555556 0.4389664

0.4 1 0.8 1.00 150 0.5262187 0.5855577 0.3647712 0.4418687

0.4 2 0.6 0.50 50 0.5282515 0.6065287 0.4314379 0.4971240

0.4 2 0.6 0.50 100 0.5430038 0.6213785 0.4921569 0.5425420

0.4 2 0.6 0.50 150 0.5505603 0.6278911 0.5243791 0.5647150

0.4 2 0.6 0.75 50 0.5336848 0.6141254 0.4163399 0.4898756

0.4 2 0.6 0.75 100 0.5513393 0.6291674 0.4855556 0.5417418

0.4 2 0.6 0.75 150 0.5598810 0.6503842 0.5228758 0.5736226

0.4 2 0.6 1.00 50 0.5294016 0.6164717 0.3960131 0.4736360

0.4 2 0.6 1.00 100 0.5538024 0.6450234 0.4836601 0.5447345

0.4 2 0.6 1.00 150 0.5691855 0.6767044 0.5265359 0.5846104

0.4 2 0.8 0.50 50 0.5276614 0.5936299 0.4386275 0.4970633

0.4 2 0.8 0.50 100 0.5437430 0.6235619 0.4896732 0.5405468

0.4 2 0.8 0.50 150 0.5509612 0.6445988 0.5437908 0.5801863

0.4 2 0.8 0.75 50 0.5346616 0.6012846 0.4035948 0.4755452

0.4 2 0.8 0.75 100 0.5536481 0.6356321 0.4818954 0.5414940

0.4 2 0.8 0.75 150 0.5656123 0.6567156 0.5256209 0.5777378

0.4 2 0.8 1.00 50 0.5298889 0.6304479 0.4209804 0.4978923

0.4 2 0.8 1.00 100 0.5491583 0.6461937 0.4939869 0.5527348

0.4 2 0.8 1.00 150 0.5698681 0.6613236 0.5240523 0.5791790

0.4 3 0.6 0.50 50 0.5456333 0.6244779 0.4724183 0.5315827

0.4 3 0.6 0.50 100 0.5697761 0.6493091 0.5356863 0.5810680

0.4 3 0.6 0.50 150 0.5821811 0.6537677 0.5437255 0.5882467

0.4 3 0.6 0.75 50 0.5781050 0.6638956 0.4996078 0.5600919

0.4 3 0.6 0.75 100 0.5967109 0.6728267 0.5494771 0.5985054

0.4 3 0.6 0.75 150 0.6045603 0.6654234 0.5486275 0.5941855

0.4 3 0.6 1.00 50 0.5718645 0.6788851 0.5090850 0.5739560

0.4 3 0.6 1.00 100 0.5911274 0.6781379 0.5369281 0.5926526

0.4 3 0.6 1.00 150 0.5986446 0.6746203 0.5426144 0.5957534

0.4 3 0.8 0.50 50 0.5642288 0.6386006 0.5062745 0.5576195

0.4 3 0.8 0.50 100 0.5762317 0.6457109 0.5449020 0.5842335

0.4 3 0.8 0.50 150 0.5901867 0.6450263 0.5483660 0.5847565

0.4 3 0.8 0.75 50 0.5691657 0.6635176 0.5158824 0.5738893

0.4 3 0.8 0.75 100 0.5894460 0.6685766 0.5483660 0.5972914

0.4 3 0.8 0.75 150 0.5953083 0.6731384 0.5532680 0.6010394

0.4 3 0.8 1.00 50 0.5679941 0.6573796 0.5011765 0.5594203

0.4 3 0.8 1.00 100 0.5894071 0.6686653 0.5288235 0.5824127

0.4 3 0.8 1.00 150 0.5975619 0.6685981 0.5335294 0.5856766

Tuning parameter 'gamma' was held constant at a value of 0

Tuning parameter 'min\_child\_weight' was held constant at

a value of 1

AUC was used to select the optimal model using the largest value.

The final values used for the model were nrounds = 150, max\_depth = 3, eta = 0.3, gamma = 0, colsample\_bytree =

0.8, min\_child\_weight = 1 and subsample = 0.75.

Extreme Gradient Boosting, unbalanced SMOTE:

eXtreme Gradient Boosting

1280 samples

11 predictor

2 classes: 'fine', 'reg'

No pre-processing

Resampling: Cross-Validated (10 fold, repeated 5 times)

Summary of sample sizes: 1152, 1151, 1152, 1152, 1152, 1153, ...

Addtional sampling using

Resampling results across tuning parameters:

eta max\_depth colsample\_bytree subsample nrounds AUC Precision Recall F

0.3 1 0.6 0.50 50 0.5192820 0.4305701 0.7286275 0.5383666

0.3 1 0.6 0.50 100 0.5169350 0.4471190 0.7113725 0.5457934

0.3 1 0.6 0.50 150 0.5156827 0.4540880 0.7148366 0.5522226

0.3 1 0.6 0.75 50 0.5131653 0.4363284 0.7284314 0.5434773

0.3 1 0.6 0.75 100 0.5025235 0.4388431 0.7123529 0.5404432

0.3 1 0.6 0.75 150 0.5062512 0.4373988 0.6983660 0.5351794

0.3 1 0.6 1.00 50 0.5111831 0.4283855 0.7266667 0.5363844

0.3 1 0.6 1.00 100 0.5093563 0.4318335 0.7022876 0.5320071

0.3 1 0.6 1.00 150 0.5109052 0.4426563 0.7001961 0.5393432

0.3 1 0.8 0.50 50 0.5157608 0.4387155 0.7241830 0.5439208

0.3 1 0.8 0.50 100 0.5121694 0.4455246 0.7126144 0.5450893

0.3 1 0.8 0.50 150 0.5080247 0.4510292 0.7026144 0.5460150

0.3 1 0.8 0.75 50 0.5157862 0.4317776 0.7168627 0.5364384

0.3 1 0.8 0.75 100 0.5103617 0.4351448 0.6896078 0.5311834

0.3 1 0.8 0.75 150 0.5061007 0.4450867 0.6921569 0.5390329

0.3 1 0.8 1.00 50 0.4991507 0.4248374 0.7310458 0.5344624

0.3 1 0.8 1.00 100 0.4927737 0.4271030 0.7069935 0.5299654

0.3 1 0.8 1.00 150 0.4912380 0.4410862 0.7047059 0.5396113

0.3 2 0.6 0.50 50 0.5160002 0.4625230 0.7045098 0.5549683

0.3 2 0.6 0.50 100 0.5322565 0.4881978 0.6883007 0.5678753

0.3 2 0.6 0.50 150 0.5394547 0.4992859 0.6864052 0.5751616

0.3 2 0.6 0.75 50 0.5157289 0.4674089 0.7034641 0.5584840

0.3 2 0.6 0.75 100 0.5285172 0.4879989 0.6989542 0.5713233

0.3 2 0.6 0.75 150 0.5337042 0.5027186 0.6898039 0.5779275

0.3 2 0.6 1.00 50 0.5236723 0.4611939 0.6964052 0.5510985

0.3 2 0.6 1.00 100 0.5250558 0.4889641 0.7004575 0.5724768

0.3 2 0.6 1.00 150 0.5361004 0.5072791 0.7016993 0.5857548

0.3 2 0.8 0.50 50 0.5116766 0.4600334 0.7047712 0.5532518

0.3 2 0.8 0.50 100 0.5240905 0.4715971 0.6908497 0.5567372

0.3 2 0.8 0.50 150 0.5329325 0.4923122 0.6872549 0.5700539

0.3 2 0.8 0.75 50 0.5176311 0.4601564 0.6920915 0.5496964

0.3 2 0.8 0.75 100 0.5279085 0.4940997 0.7035294 0.5774506

0.3 2 0.8 0.75 150 0.5350027 0.5086685 0.7037908 0.5875444

0.3 2 0.8 1.00 50 0.5234467 0.4656251 0.7209804 0.5628894

0.3 2 0.8 1.00 100 0.5299555 0.4887369 0.7034641 0.5737407

0.3 2 0.8 1.00 150 0.5345257 0.5073119 0.6968627 0.5836755

0.3 3 0.6 0.50 50 0.5370712 0.4880030 0.6946405 0.5692671

0.3 3 0.6 0.50 100 0.5334105 0.4957464 0.6841176 0.5714988

0.3 3 0.6 0.50 150 0.5514192 0.5201038 0.7024183 0.5937075

0.3 3 0.6 0.75 50 0.5346722 0.4908694 0.6978431 0.5725937

0.3 3 0.6 0.75 100 0.5433188 0.5056023 0.6981699 0.5830629

0.3 3 0.6 0.75 150 0.5540811 0.5128193 0.6956209 0.5869624

0.3 3 0.6 1.00 50 0.5400686 0.5030807 0.7292810 0.5924461

0.3 3 0.6 1.00 100 0.5528548 0.5200910 0.7120261 0.5972658

0.3 3 0.6 1.00 150 0.5566717 0.5227373 0.7095425 0.5983737

0.3 3 0.8 0.50 50 0.5315646 0.4975733 0.6975163 0.5773907

0.3 3 0.8 0.50 100 0.5469911 0.5137750 0.7015686 0.5896468

0.3 3 0.8 0.50 150 0.5501558 0.5174910 0.7011765 0.5922413

0.3 3 0.8 0.75 50 0.5450127 0.4953480 0.7004575 0.5765073

0.3 3 0.8 0.75 100 0.5622324 0.5227569 0.7132026 0.6000679

0.3 3 0.8 0.75 150 0.5627027 0.5274928 0.6969281 0.5974090

0.3 3 0.8 1.00 50 0.5508373 0.5010392 0.7215033 0.5876638

0.3 3 0.8 1.00 100 0.5652055 0.5258278 0.7053595 0.5977613

0.3 3 0.8 1.00 150 0.5766027 0.5340165 0.7066667 0.6034492

0.4 1 0.6 0.50 50 0.5056758 0.4381064 0.7226797 0.5430077

0.4 1 0.6 0.50 100 0.5076823 0.4467937 0.6962745 0.5413554

0.4 1 0.6 0.50 150 0.5054191 0.4535365 0.6966667 0.5456079

0.4 1 0.6 0.75 50 0.5066329 0.4317280 0.7162092 0.5355098

0.4 1 0.6 0.75 100 0.5020463 0.4386797 0.6862745 0.5324106

0.4 1 0.6 0.75 150 0.5017362 0.4624092 0.6910458 0.5498820

0.4 1 0.6 1.00 50 0.5059587 0.4384895 0.7140523 0.5405739

0.4 1 0.6 1.00 100 0.5068901 0.4461709 0.7025490 0.5427361

0.4 1 0.6 1.00 150 0.5043023 0.4486823 0.6956209 0.5423114

0.4 1 0.8 0.50 50 0.5164979 0.4323286 0.7069281 0.5331896

0.4 1 0.8 0.50 100 0.5038607 0.4358342 0.6873856 0.5301190

0.4 1 0.8 0.50 150 0.5047969 0.4512699 0.6954248 0.5443864

0.4 1 0.8 0.75 50 0.5030052 0.4257279 0.7047712 0.5286574

0.4 1 0.8 0.75 100 0.5057109 0.4436653 0.6947059 0.5387025

0.4 1 0.8 0.75 150 0.5076946 0.4545589 0.6954902 0.5462090

0.4 1 0.8 1.00 50 0.5060723 0.4262490 0.7139216 0.5312301

0.4 1 0.8 1.00 100 0.5093558 0.4404896 0.7054902 0.5399504

0.4 1 0.8 1.00 150 0.5097005 0.4442551 0.6998693 0.5411334

0.4 2 0.6 0.50 50 0.5239811 0.4638722 0.6998039 0.5548703

0.4 2 0.6 0.50 100 0.5305121 0.4871887 0.6831373 0.5651563

0.4 2 0.6 0.50 150 0.5379910 0.5091748 0.6897386 0.5825522

0.4 2 0.6 0.75 50 0.5199599 0.4691292 0.6772549 0.5509993

0.4 2 0.6 0.75 100 0.5352637 0.5008420 0.6830065 0.5749093

0.4 2 0.6 0.75 150 0.5383158 0.5048404 0.6822222 0.5777416

0.4 2 0.6 1.00 50 0.5210825 0.4675742 0.6918301 0.5549394

0.4 2 0.6 1.00 100 0.5278603 0.4980571 0.6830065 0.5716429

0.4 2 0.6 1.00 150 0.5369770 0.5009994 0.6832026 0.5742725

0.4 2 0.8 0.50 50 0.5122949 0.4668091 0.6760784 0.5487569

0.4 2 0.8 0.50 100 0.5217436 0.4934238 0.6958824 0.5740804

0.4 2 0.8 0.50 150 0.5256115 0.4988033 0.6864706 0.5730825

0.4 2 0.8 0.75 50 0.5316810 0.4656904 0.6840523 0.5512457

0.4 2 0.8 0.75 100 0.5424995 0.4939505 0.7069935 0.5788634

0.4 2 0.8 0.75 150 0.5453182 0.5082708 0.7128758 0.5897371

0.4 2 0.8 1.00 50 0.5152966 0.4650760 0.6841176 0.5510222

0.4 2 0.8 1.00 100 0.5272760 0.4924499 0.6887582 0.5710126

0.4 2 0.8 1.00 150 0.5302828 0.5168376 0.6994771 0.5907171

0.4 3 0.6 0.50 50 0.5276360 0.4922409 0.6856209 0.5694411

0.4 3 0.6 0.50 100 0.5489805 0.5122625 0.6902614 0.5824712

0.4 3 0.6 0.50 150 0.5568049 0.5208188 0.6922876 0.5899305

0.4 3 0.6 0.75 50 0.5445464 0.5022122 0.7103922 0.5858133

0.4 3 0.6 0.75 100 0.5647614 0.5310322 0.7060784 0.6020770

0.4 3 0.6 0.75 150 0.5687975 0.5358379 0.6957516 0.6013404

0.4 3 0.6 1.00 50 0.5394763 0.4969914 0.7187582 0.5849224

0.4 3 0.6 1.00 100 0.5513656 0.5174355 0.6942484 0.5889456

0.4 3 0.6 1.00 150 0.5623565 0.5184504 0.7047059 0.5942211

0.4 3 0.8 0.50 50 0.5316385 0.5018012 0.7033987 0.5806841

0.4 3 0.8 0.50 100 0.5432494 0.5275834 0.7068627 0.5992224

0.4 3 0.8 0.50 150 0.5505519 0.5248639 0.7033987 0.5968581

0.4 3 0.8 0.75 50 0.5523636 0.5166019 0.7033333 0.5925794

0.4 3 0.8 0.75 100 0.5554067 0.5249118 0.6977124 0.5957548

0.4 3 0.8 0.75 150 0.5687154 0.5332638 0.7049673 0.6044793

0.4 3 0.8 1.00 50 0.5528384 0.5126332 0.7084967 0.5920721

0.4 3 0.8 1.00 100 0.5681319 0.5269576 0.6993464 0.5976568

0.4 3 0.8 1.00 150 0.5705535 0.5265591 0.6947712 0.5953179

Tuning parameter 'gamma' was held constant at a value of 0

Tuning parameter 'min\_child\_weight' was held constant at a value of 1

AUC was used to select the optimal model using the largest value.

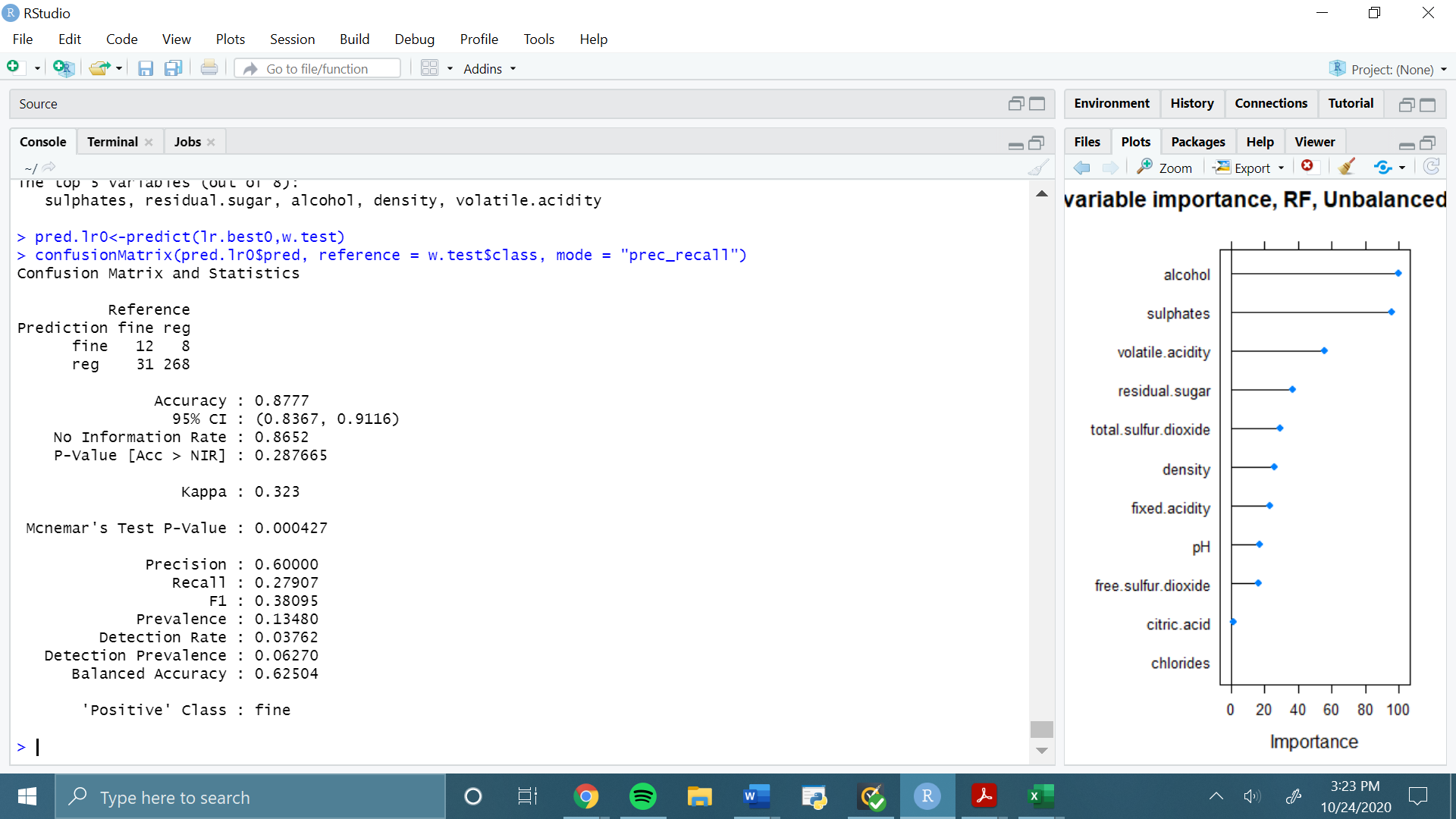
The final values used for the model were nrounds = 150, max\_depth = 3, eta = 0.3, gamma = 0, colsample\_bytree = 0.8, min\_child\_weight = 1

and subsample = 1.

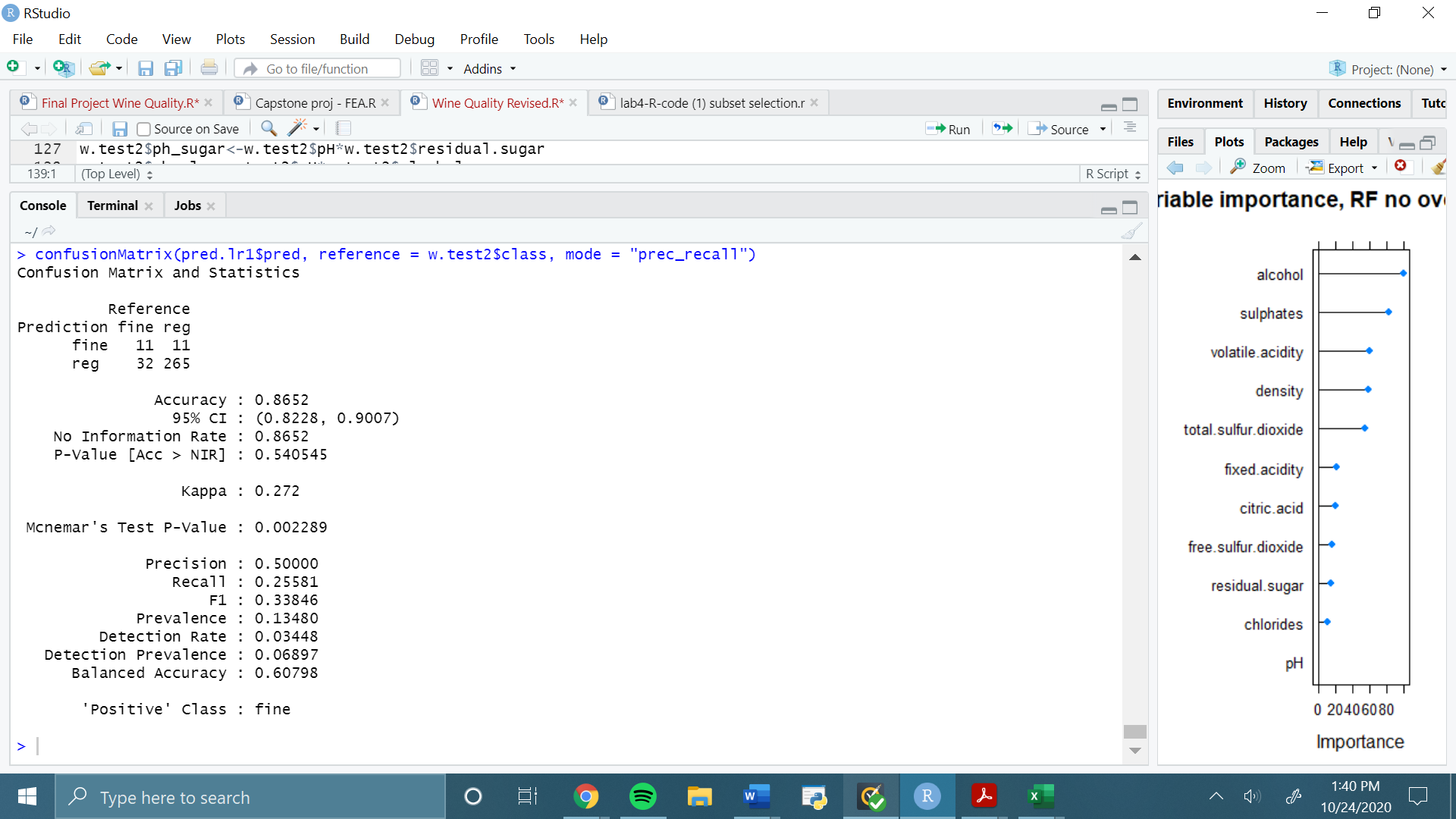
**Appendix C.**

**Testing Confusion Matrices**

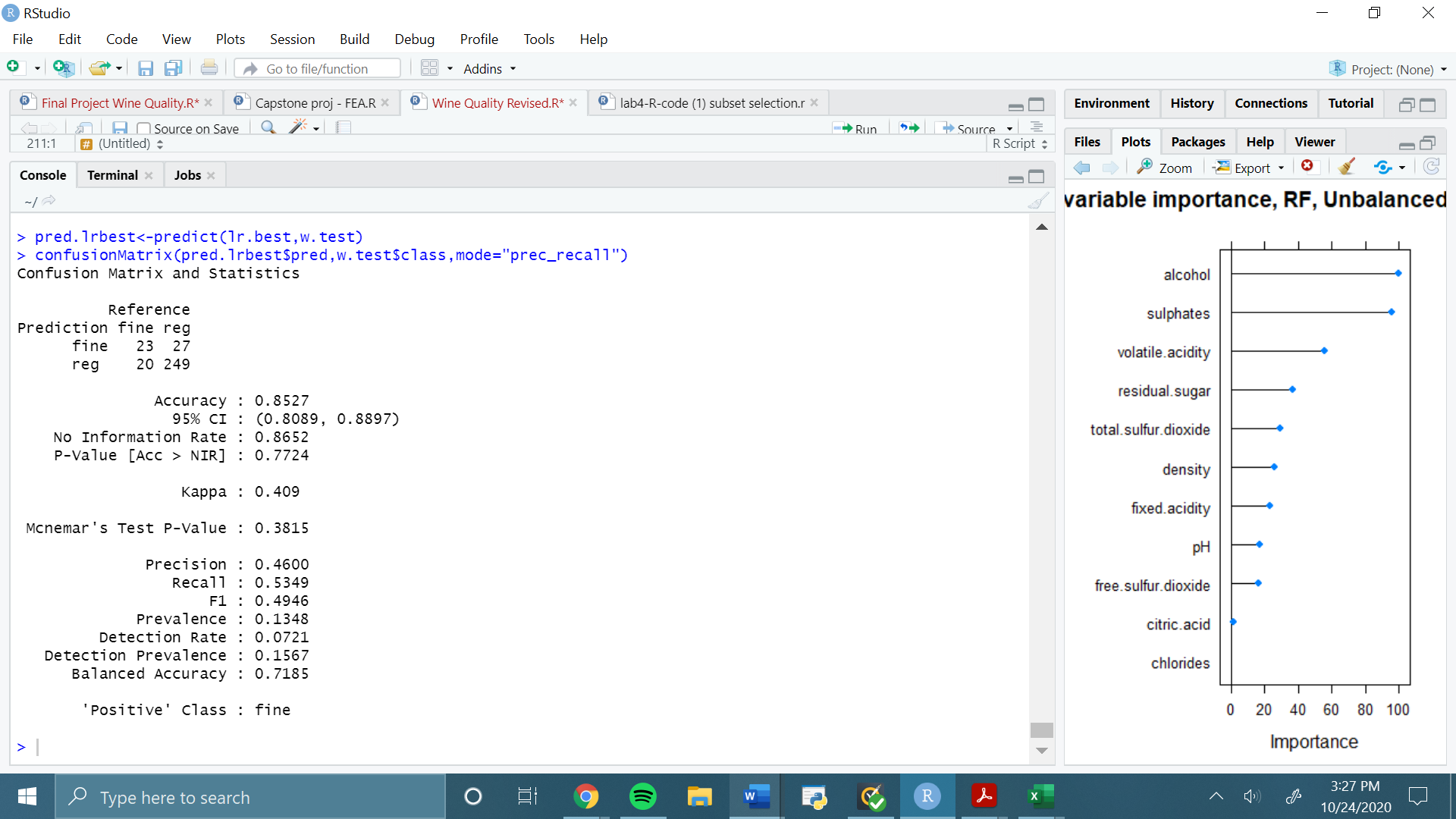
Logistic Regression, RFE, threshold >.5; no interaction terms:



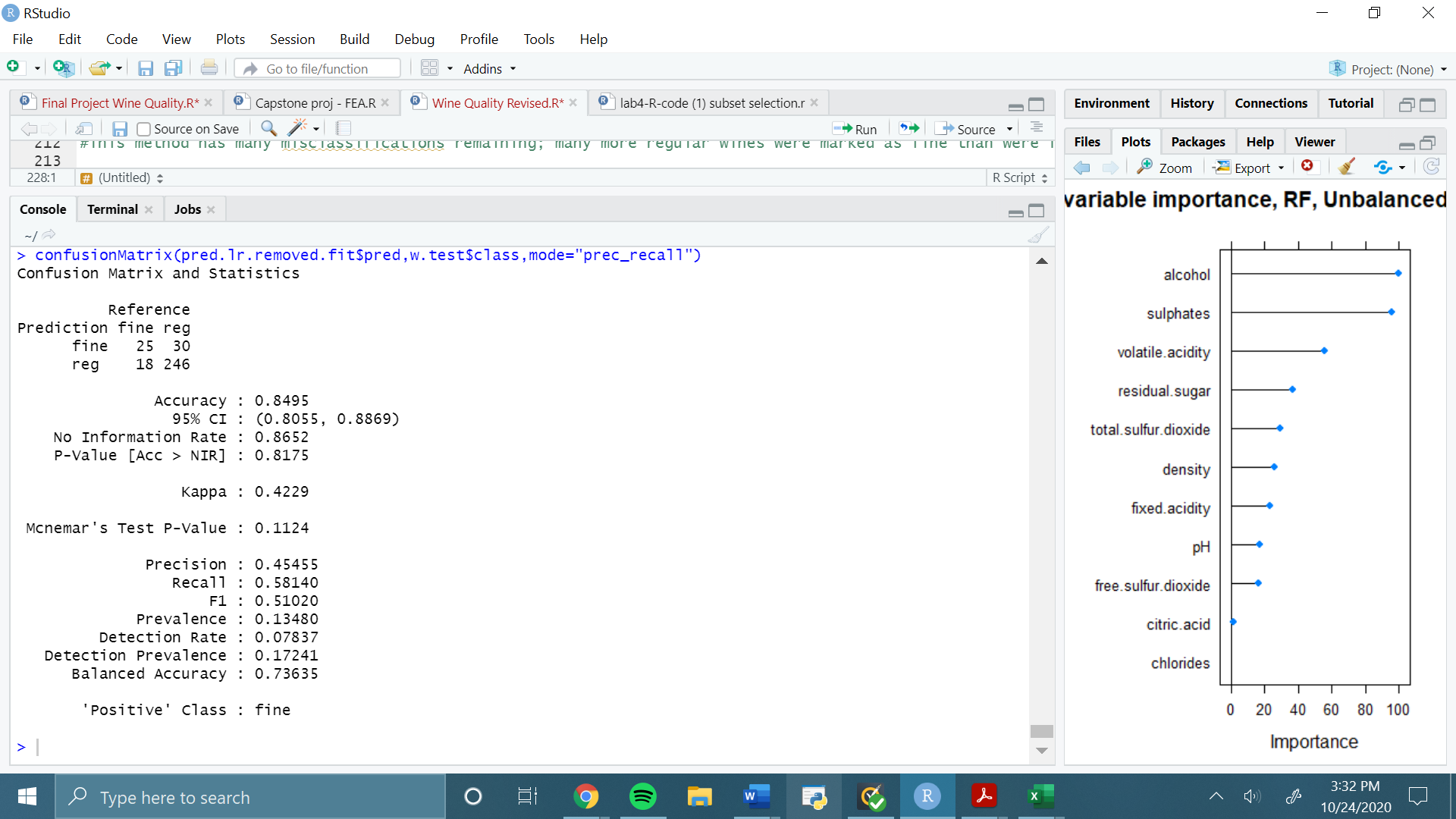
Logistic Regression, RFE, threshold >.5; two interaction terms:



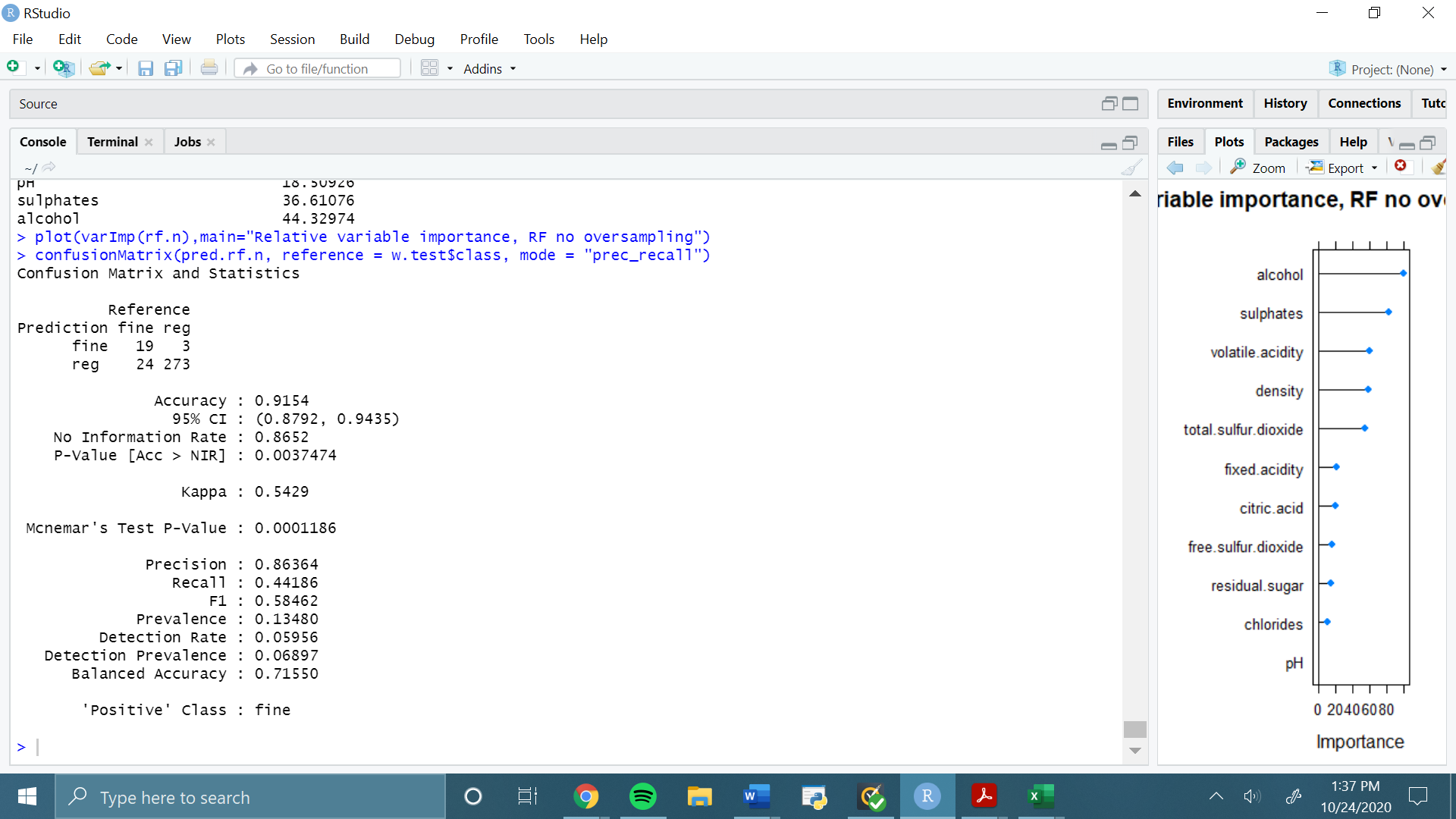
Logistic regression, RFE, threshold >.75; 8 predictors (no interactions considered):



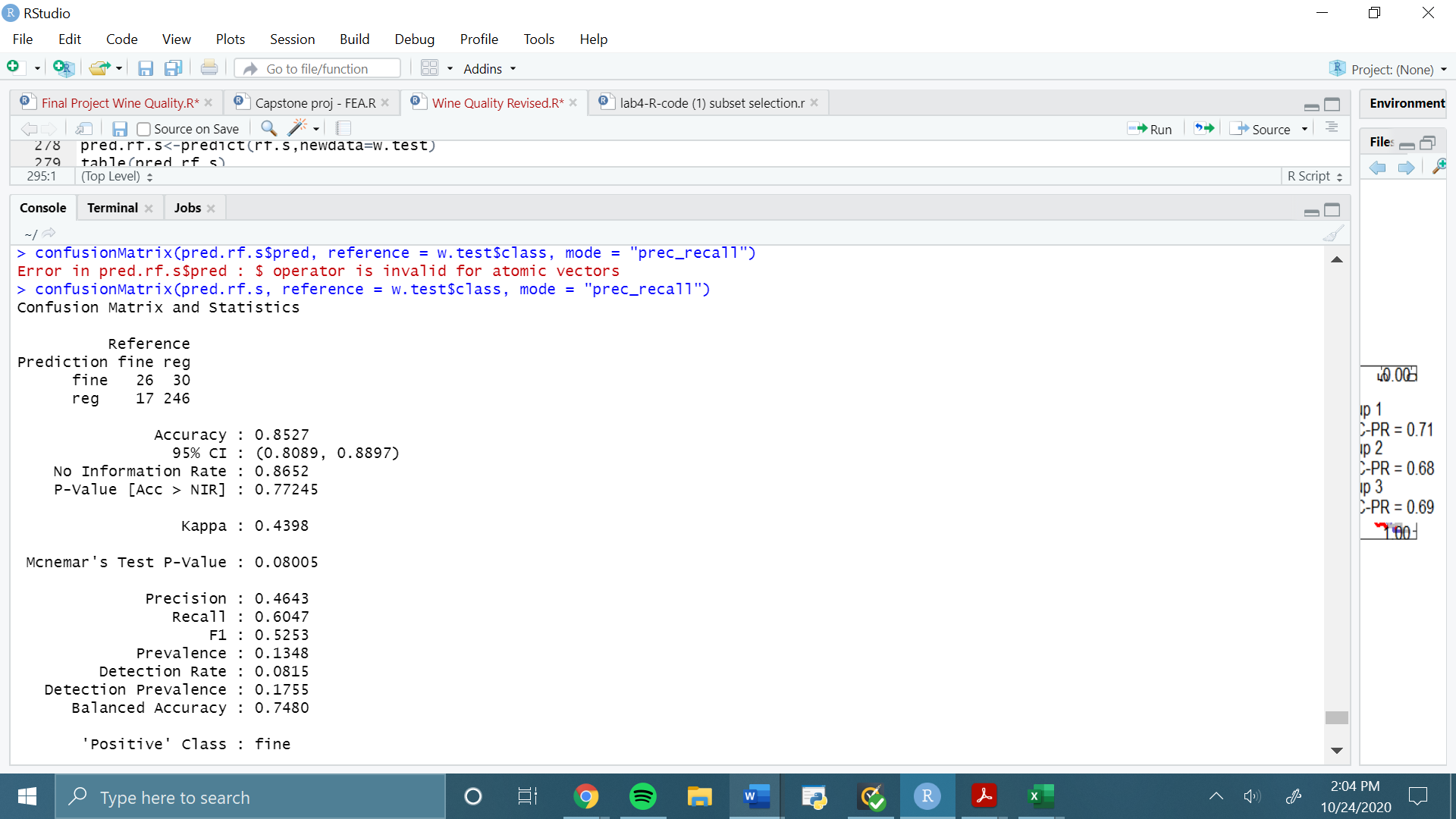
Logistic Regression, Multicollinearity removed, RFE, threshold >.75:



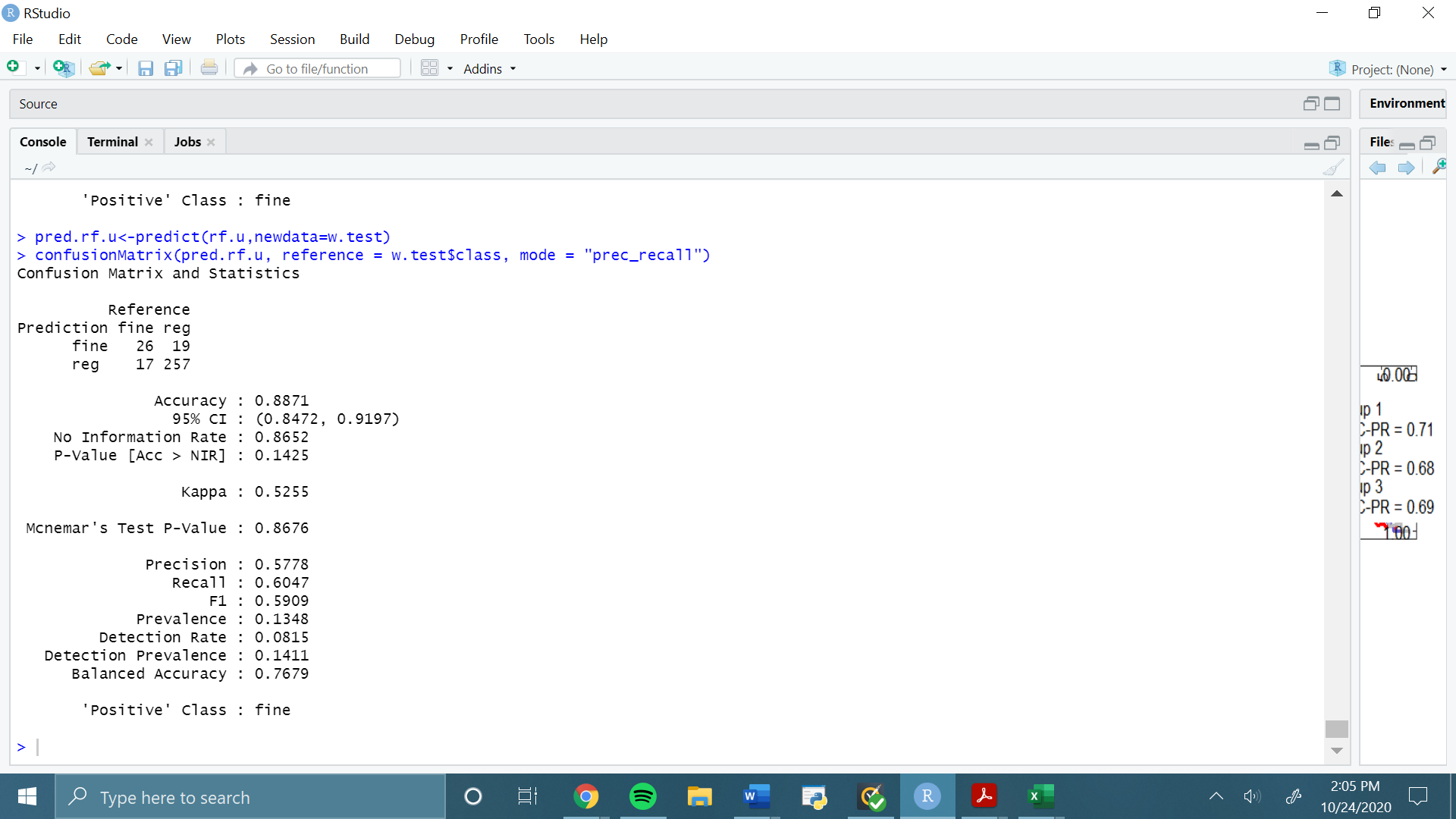
Random forest, no oversampling:



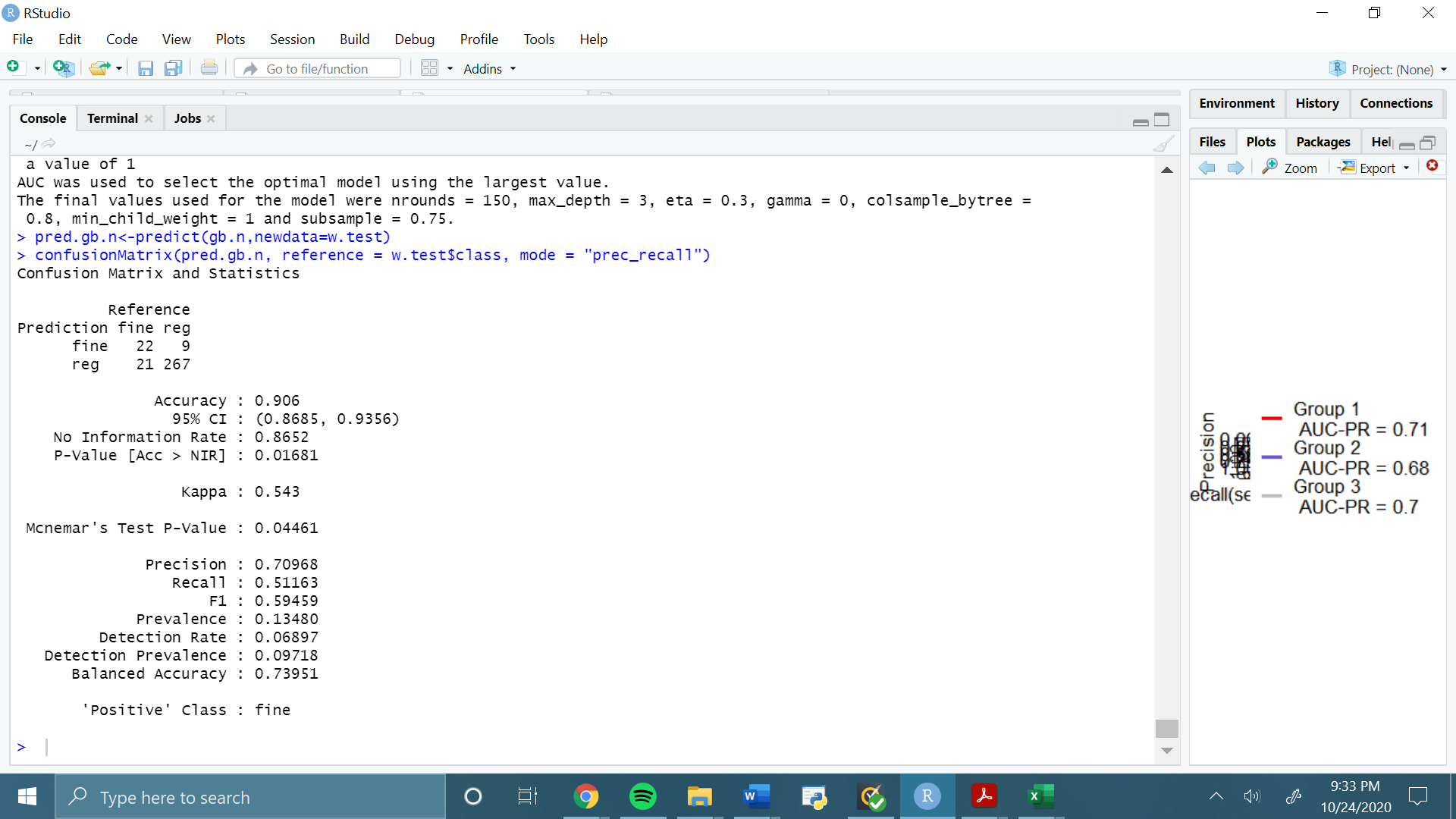
Random Forest, balanced SMOTE:



Random Forest, unbalanced SMOTE:



Extreme Gradient Boosted Trees, No Oversampling



Extreme Gradient Boosted Trees, Unbalanced SMOTE

