Classification Assignment

Ryoji Takahashi

December 1, 2024

1 Description of Assignments and Objectives

The classification assignments are

1. Choose an open classification dataset other than Iris (reference source)
2. In a Jupyter notebook, perform exploratory analysis (EDA), including data cleaning, transformations, aggregations and visualizations as appropriate.
3. Select, train and test the model(s) considered appropriate.
4. Justify the chosen model based on performance metrics.
5. Draw conclusions from the exercise carried out.
6. Prepare a deliverable with all the files necessary to reproduce the analysis and put the trained model into production (DevOps integration).

1.1 Dataset

1. Datasetwine quality was selected for this study. The wine qualities were ranked from **3** to **8** (**8** as a good quality). I could classify as a multi-classification, however, I performed binary classification by categorizing as above 7 is a good wine (labeled as 1), and less than 7 is not good wine (labeled as 0). As a result, the number of no good wine is **1382** and good wine is **217**. This is an imbalanced datasets.

1.2 Exploratory Data Analysis (EDA)

Before discussing classification methods, I would like to address 2. EDA. Details EDA were shown in the notebook. No missing values (NAs) were present, and the data types were appropriate. However, if NAs were present, they should be handled carefully by either dropping or imputing them..
Features correlations were shown in Figure 1 at the last section of EDA. The "density" has strong positive correlation with "residual sugar" whereas it has strong negative correlation with "alcohol". "Alcohol" has a positive correlation with "quality", while the "free sulphur dioxide" and "citric acid" has almost no correlation with "quality". These insights are very important for further feature engineering.

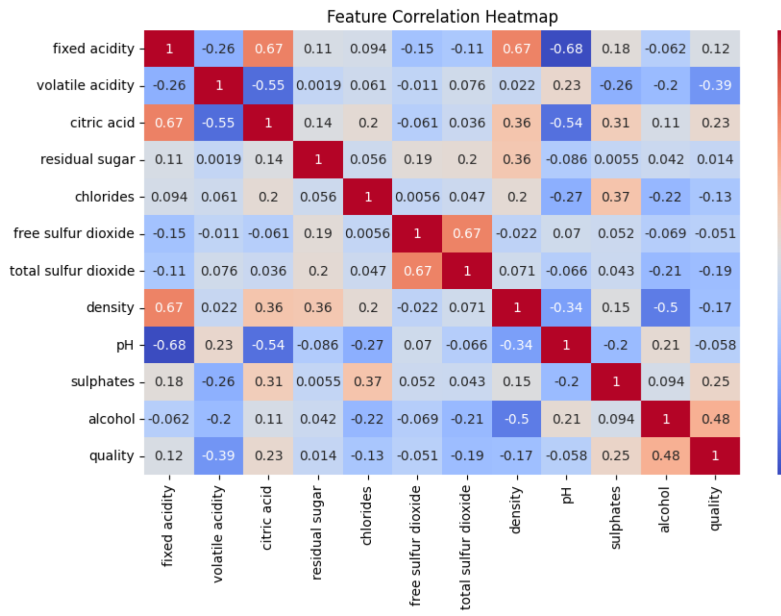

Figure 1: Correlations

1.3 Classification Mathods

3. Building ML models. As I mentioned, this is an imbalanced datasets. To handle such datasets, there are several options, such as Synthetic Minority Over-sampling Technique (SMOTE), threshold moving, and ensemble methods. As one of ensemble methods, Random Forest (RF), bagging with independent decision trees. Additionally, XGBoost, bagging with $L1$ and $L2$ regularization is a popular method. Therefore, I deployed these two methods for binary classification, after scaling the data using StandardScaler (transforming to $\mu = 0$, $0\sigma = 1$). As a standard way, the dataset was split into 80 % training and %20 testing.
RF of precision and recall are **0.67** and **0.60**, respectively. To improve these accuracy, I performed hyper-parameter turning by grid search. As my experiences, I have improved the outcomes, however, I was not able to improve them significantly. In the code, I left the grid search lines, and I have been investigating.
Next, I applied XGBoost with optuna hyper-parameter tuning. The results improved compared to RF, with precision and recall scores of 0.67 and 0.73, respectively. Of course, for deploying to production, it would be also important to repeat training and (validating) testing with further hyper-parameter tuning.
Figure 2 shows confusion matrix (CM) of XGBoost results. The confusion matrix (CM) is one of several

evaluation metrics used to measure the performance of a classification model. The model performance metrics, such as precision and recall were calculated from CM.

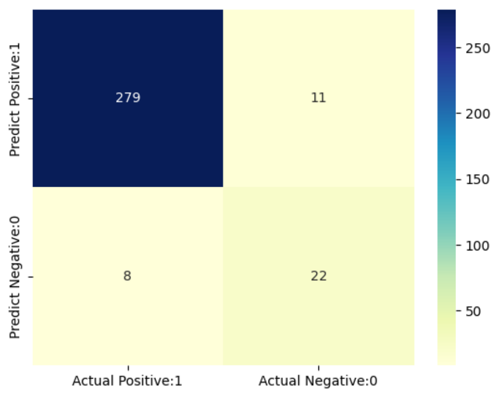

Figure 2: Confusion Matrix

It is also common to show the Receiver Operating Characteristic (ROC) curve. In Figure 3. It is a graphical representation of the performance of a binary classifier at different classification thresholds. I did not use threshold moving techniques, which plots the True Positive Rate (TPR) against the False Positive Rate (FPR).

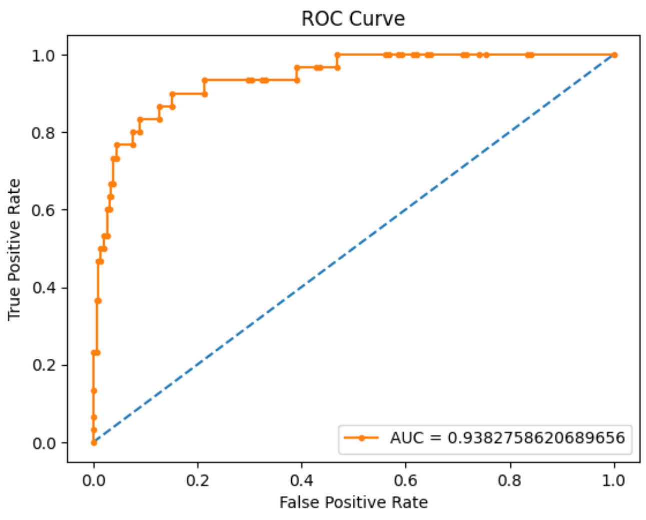

Figure 3: ROC

Figure 4 shows feature importances. As in the code, both RF and XGBoost of feature importances were coincided. This plot is importance for feature engineering which will be discussed in the last section.

Finally, for delivery and reproducing, I saved training models to pkl files as in the code, so that one can load trained models and perform tests.

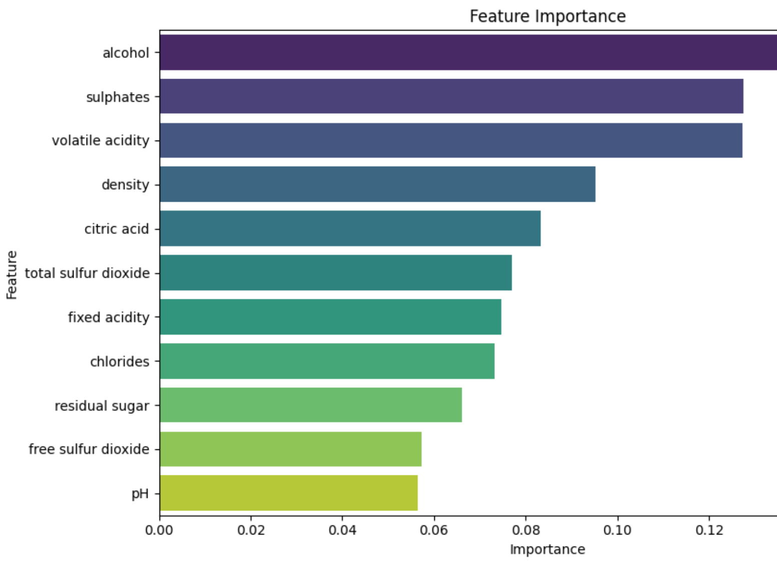

Figure 4: Feature Importance

2 Conclusion and Further Discussion

In real-world evidence (RWD) datasets, I have often encountered imbalanced datasets. I have applied SMOTE to some studies. however, I have found it to be less effective in improving accuracy for ”general” imbalanced datasets. Usually, as the above, RF and XGBoost with standard scaler perform better.
I also would like to address that the ’bias-variance trade-off’ is an important issue. As better accuracy will be trade off of bias and variance (complexity). Feature engineering such as select important features and drop unimportant features would also improve these imbalance dataset modelings.