Classification Assignment

Ryoji Takahashi

December 2, 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 deploy the trained model into production (DevOps integration).

1.1 Dataset

1. The dataset wine 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 wines with a score above 7 as good (labeled as 1) and below 7 as not good (labeled as 0). As a result, the number of no good wine is **1382** and good wine is **217**. This is an imbalanced dataset.

1.2 Exploratory Data Analysis (EDA)

Details of the 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..
In this section, I highlighted feature correlation matrix which is shown in Figure 1. 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.

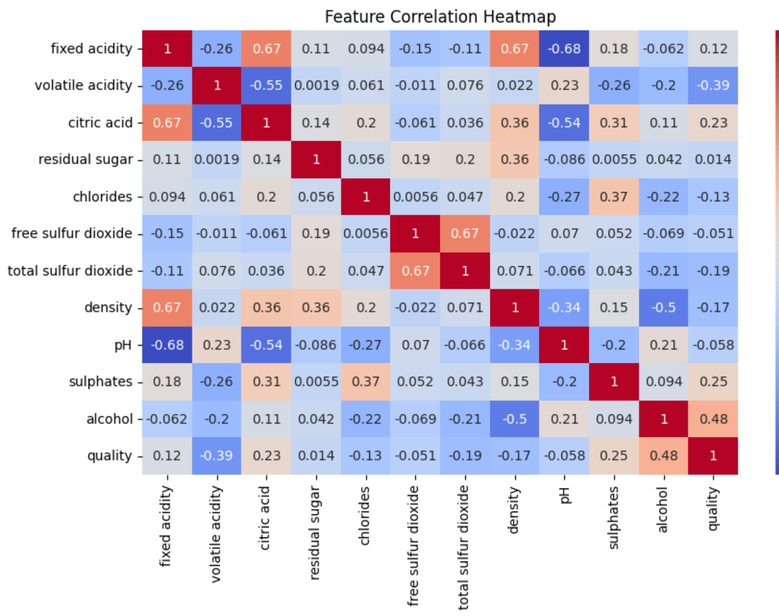

Figure 1: Correlation Matrix

1.3 Classification Methods

3. Building ML models. As I mentioned, this is an imbalanced dataset. To handle such datasets, there are several options, such as Synthetic Minority Over-sampling Technique (SMOTE), threshold moving, and ensemble methods. Among ensemble methods, Random Forest (RF) which is bagging with independent decision trees, and XGBoost which is bagging with $L1$ and $L2$ regularization are widely used for classification tasks. Therefore, I deployed these two methods for binary classification with StandardScaler (transforming to $\mu = 0$, $\sigma = 1$). As a standard way, the dataset was split into 80 % training and 20 % testing.
Results: the precision and recall scores for RF are **0.67** and **0.60**, respectively. To improve these accuracy, I performed hyper-parameter turning using grid search. Usually, I have improved the outcomes in the most of the previous cases, however, I was not able to improve them significantly in this study. 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 were improved compared to RF, with precision and recall scores of **0.67** and **0.73**, respectively. Notice that, 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 a common evaluation metric used to measure the performance of a classification model. The model performance metrics, such as precision and recall were calculated from CM.

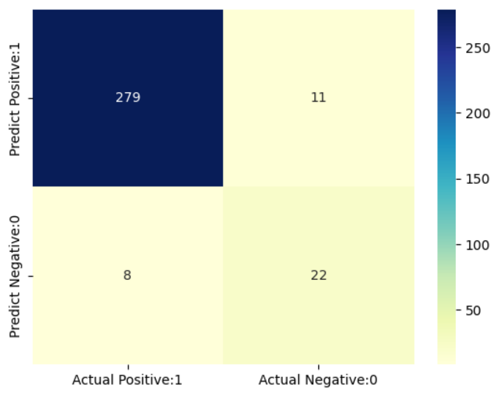

Figure 2: Confusion Matrix

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, which plots the True Positive Rate (TPR) against the False Positive Rate (FPR).

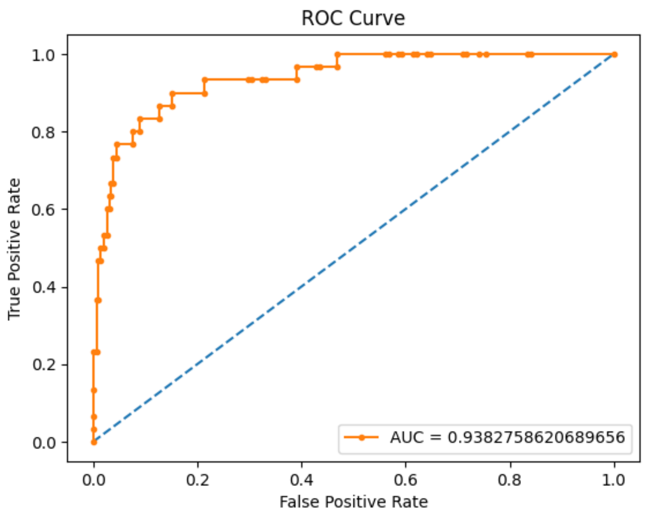

Figure 3: ROC

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.

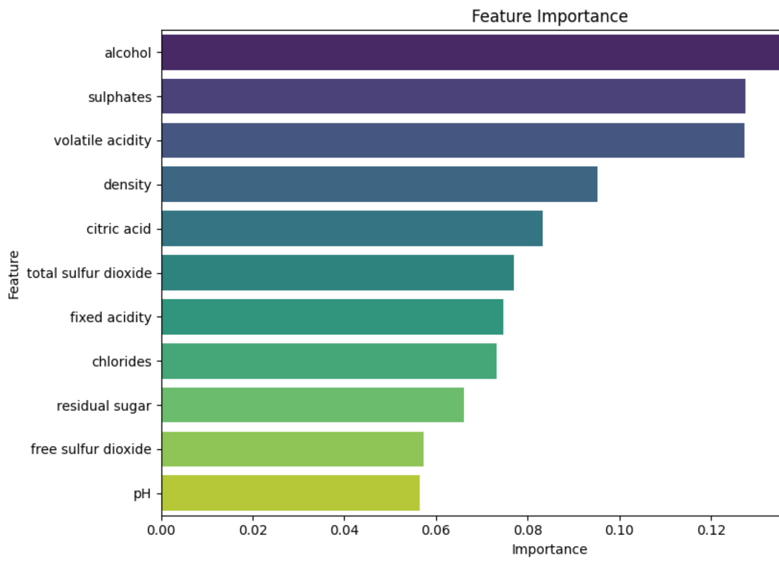

Figure 4: Feature Importances

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. Achieving better accuracy often involves a trade-off between bias and variance (complexity). Feature engineering, such as selecting important features and drop unimportant features would also improve these imbalance dataset modelings.