# **INFO 5505 Applied Machine Learning for Data Scientists**

## Assignment 2

Name: Ravi Varma Kumar Bevara

### **PART I: Model and Dataset Description**

Regression -It is a problem of predicting a real-valued label i.e., target in the machine learning algorithm, given an unlabeled example. Predicting the price of a product based on past year sales, such as number of purchases, reviews data, product features and so on is an example of regression.

Logistic Regression - Rather than naming this as a regression, it should belong to a classification algorithm because it will classify the dataset into binary categories. The naming is done based on the mathematical formulation of logistic regression.

* It is a case of binary classification but can be extended to multiclass classification.
* The same linear combination of independent variables will be applicable to this case as will, but with as twist in dependent variable. The y variable in here will be having two possible values.
* The cost function associated with this algorithm is standard logistic function also known as "Sigmoid Function"
* The error rate will stay the same, minimization of empirical risk, which is mean squared error.

The dataset we will be using in this assignment is from UCI machine learning repository, also hosted on Kaggle which describes the quality of red variant of the wine Vinho Verde, using the various physicochemical properties.

In total, we have 1599 records in the dataset with 12 attributes. The 12th attribute is quality score which is a continuous variable with a range of 0-10. If we dive into details of each variable…

|  |  |
| --- | --- |
| **Variable** | **Description** |
| fixed acidity | It describes the acidity levels of most fixed acids or non-volatile acids involved in wine. |
| volatile acidity | This will be the acidity levels of volatile acids, which are reasons to acidic or vinegary taste. |
| citric acid | It is a component used for maintaining freshness of the wine, it will be in small quantities. |
| residual sugar | It replicates the sugar levels after fermentation of wine, it will define the sweetness of wine. |
| chlorides | It is nothing but salt level in the end product, defining the saltiness of the wine. |
| free sulfur dioxide | This component will define the level of so2 which maintains the equilibrium and prevents the wine taste while aging. |
| total sulfur dioxide | It is the total amount of so2 which is free and bounded with other free components in the wine preparation. It is hard to distinguish between free and unbounded forms of so2. |
| density | This component will define the thickness or dense water levels and sugar levels in wine. |
| pH (0-14) | Generally, this is level if acidic or basic scales of wine. If less than 7 acidic and basic if >7. |
| sulphates | It is produced by yeast, preventing oxidation, growth of unwanted microorganisms. |
| alcohol | The ABV level of the wine, defining the fortified or unfortified categories in wine. |

The bigger picture in this assignment is to predict the quality level of wine as high or low. I will be using the “Logistic regression” for solving this problem. Although the given dataset in not having the price, we have the physiochemical properties of the wine which we can use to get a categorize the wine a high quality or low quality. Although, the given target variable is a continuous score of quality of wine based on human wine taste preferences that are based on easily available analytical tests at certification step, we need to come up with a accuracy score that could be used for designing new types of wine, defining the pricing policy based on the quality of wine.

### **PART II: Exploratory Data Analysis (Data Pre-Processing, Data Visualizations)**

To start, I decided to use the “Google Colab” as my programming IDE for the tasks. Also, I decided to store my dataset on the google drive. So, the first step is to import various python modules which can be used to better understand the data. Then, I mounted the google drive for accessing the dataset.

A picture containing application

Description automatically generated

To better understand the variables, we need to look how the data is distributed and what’s the shape of the data. I continued with exploring the columns of the data. In this process, I recognized values of the target variable ‘quality’ are not balanced or ordered. The modules NumPy and pandas has various functions such as .head() to check the structure of the data. Also .columns will result all the columns of the dataset.

Table

Description automatically generated

Based on the above screenshot, we can clearly see that the given raw form of data is too much variant for each independent variable. So, we may need to use the scaling property to get all the data points into a same range, which will eventually gain an advantage in model training and testing.

Graphical user interface, text, application

Description automatically generated

As we can see that all the provided datatypes in the data frame as float and only the target variable is integer. This is a good sign to perform any regression model, as we are having all numerical values in the data. We may not need any encoding techniques to convert the categorical to numerical values. As described in the dataset description, we have 1599 observations in the data with 12 attributes, and among which all are physiochemical properties of wine. Also, we can see that there are no null values in any column, from the response in the above picture. Once, I came to know about the variables, then I wanted to get a better understanding of each variable in the dataset separately. I think the correlation plot would be a better choice to examine the relationships between target and other variables, along with histograms of each variable.

Chart, treemap chart

Description automatically generated

After looking into the correlation matrix result, we can list the variables with strong positive correlation, moderate positive correlation, and low positive correlation. Also, the same cases with the negative correlations.

We can clearly see that fixed acidity is in strong correlation with critic acid and density. Also, the free sulfur dioxide is strongly positive correlated with the total sulfur dioxide. On the other way around, fixed acidity is strongly negatively correlated with pH. Our target variable quality and alcohol are moderately positive correlated with alcohol. We also have moderate negative correlation between citric acid and volatile acidity, pH and citric acid, density and alcohol which are actually making some sense. Also, notably the volatile acidity is strongly negatively correlated with quality, but it is still notable enough to start the data preprocessing.

Also, to tackle the problem of class imbalance, I decided to convert the continuous numerical target variable of quality score to binary quality level.

Chart, pie chart

Description automatically generated

Chart, pie chart

Description automatically generated

But we can still that there is class imbalance problem, but I decided to stay the same and check the model results with this dataset.

Table

Description automatically generated

Now, it is time to observe the distributions of all the independent variables to identify any skewed distributions, and some outliers. From the results below, we can see that expect density and pH almost all other variables are right skewed, and we need to tackle with outliers and skewness.

A picture containing window

Description automatically generated

So, looking at the visualization, I can sense some variables are affected with the outliers. So, after checking the skewness and kurtosis, I can remove the outliers based on the z-scores of all the columns.

Graphical user interface, table

Description automatically generated

From the above picture, we can see that chlorides, residual sugar, free sulfur dioixde, total sulfur dioixde, and sulphates skewness is much graeter than 1, which staes that these variables are not normally distributed, and I wanted to check the outliers impact on those particular columns. I plotted the boxplots of these variables to confirm the presence and impact of outliers.

Graphical user interface, text, application, email

Description automatically generated

Calendar, whiteboard

Description automatically generated

This picture confirms the status of outliers in those variables, and I decided to remove them using z-scores of respective columns.

Chart, pie chart

Description automatically generated

After removing the outliers, I decided to check if it has solved the class imbalance problem. But it almost remained the same as only 14% of the dataset were classifying into class ‘1’ and rest 86% is class ‘0’. And also, I decided to check the skewness and kurtosis after removing the outliers,

Table

Description automatically generated

We can still find the columns with more than 1 skewness which are residual sugar, chlorides, and total sulfur dioxide. To overcome this, I used the log transformation on these variables.

Graphical user interface, text, application

Description automatically generated

I used the NumPy module for log transformation and after transforming, I removed the old columns and moved the target-oriented columns to the end of the data frame. I have checked the correlation from the transformed variables and the target variable again.

Chart, treemap chart

Description automatically generated

Then, I choose to describe the data after cleaning and pre-processing the data just to confirm the scaling of the data is necessary or not for the further process.

Graphical user interface, application

Description automatically generated

### **PART III: Model Training (Splitting the Data, Applying the Model)**

After a deep study in the packages available for model training in python, I decide to use sklearn also known as Scikit-Learn to train my simple and multivariate linear regressions.

Graphical user interface, text, application

Description automatically generated with medium confidence

After looking at the data in the preprocessing phase, I have decided to build 3 models and compare the best of the models by using different metrics and selection methods. To start off with, I will be spitting the dataset into 80:20 proportions respectively for training and testing phases of the model.

As a L2 regularizations is applied by default, I used the solver ‘lbfgs’ to all my models, the reason behind this selection is it will be an optimal choice for small scale datasets, especially when we are performing the binary classification. Using maximum iterations as 1000 instead of default 100 is the assumption that data might take the solvers to converge to a value. A solver is an algorithm used in optimizing the logistic regression, by default it will be liblinear for small datasets. But I changed it to L2 penalty supported solver ‘lbfgs’.

We can see from the below picture that the test score for the model-1 is 88.85% which is almost 89%. To check if the model is overfitting or underfitting, I checked the training accuracy which is also 87%. As there is not much difference in those, we can believe that the model is not overfitting. After we are modeling the algorithm to perform on the unseen data, it’s better to evaluate the model. One of the best techniques for model evaluations is k-fold cross validation, which is basically dividing the whole data into k folds, and then it will assign k-1 folds to training and remaining one to the testing, but this will perform for all the tests to be each fold. I have also performed the k-fold validation on the test data to compare the model to its best, then the accuracy seems to be almost the same at 88.5%, which is an average over k trails.

Graphical user interface, text, application, email

Description automatically generated

For the second model, as I have assumed from the start of this process. The data is not on the same scale to train the machine, we can clearly see the means, min, and max of each independent variables in the describe picture before the training, so I decided to use the Standard Scalar module from scikit learn which is not but z-score normalization from statistics. It is basically standardizing features by removing the mean and scaling to unit variance.

From the result in the below picture, we can clearly see that there is clear increase in the testing accuracy of the model to 90% and training accuracy to 86%. As the training accuracy is decreased from 87% to 86%, we can confirm that there a little bit of the impact of the overfitting, but not too much. The model evaluation in this case, has provided a mean accuracy of 88%. There was a slight change in the accuracy, but still I thought it could do better.

Graphical user interface, text, application, email

Description automatically generated

For the third model, I decided to use the standard technique from data science when we are using more independent variables in the model training, which is ‘Principal Component Analysis’. It is dealing with linear dimensionality reduction using the singular value decomposition of the data as we will be projecting the same into a lower dimensional space. The reason why I used it after the scaling the data is, the input data is centered but not scaled for each feature before applying the SVD.

Chart, line chart

Description automatically generated

From the above picture, I fitted the PCA to the scaled data from the above model. When I plotted the cumulative sum of variance explained against the number of variables used, I can clearly see that 8 components are giving me or rather explaining me almost more than 90% of the variance. SO, I decide to use number of components as 8 to train the logistic regression.

Graphical user interface, text, application, email

Description automatically generated

From the above picture, we can see that the accuracy or the test score of the model 3 is almost 91%, also the evaluated model score using k-fold cross validation using 10 folds is 89%. So, I decide to extend more based on the importance of the features in the model. But before that, it’s not only the accuracy of the model that defines the goodness and perfect fit of the model.

**PART IV: Model Evaluation Metrics (Prediction Results, Test Scores, and Metrics)**

We will be performing some model evaluation metrics on all the above three models, to judge which is the best model based on goodness and fit of the model. The three metrics that I have chosen to decide on the best model are…

1. Classification Accuracy (Confusion Matrix)
2. F-1 score, Precision, Recall (Classification Report)
3. ROC-AUC Score

Accuracy is the most common evaluation metric for any classification problem. It is the ratio of number of correct predictions to the total number of predictions made by the model. A confusion matrix provides more detailed breakdown of the correct and incorrect predictions. All the diagonal elements represent the correct predictions for all the labels and anything other than them are incorrect predictions. The higher the number of diagonal elements in the total elements, the higher the accuracy. While the F1-score, precision, recall is the accuracy of the test set on the model. F-Measure counts both precision and recall into consideration while interpretation. Precision is the ratio of number of correct positive results to the total positive predicted observations. On the other hand, Recall is the ration of number of correct positive results to the total actual positives i.e., all relevant samples. The last metric which is ROC-AUC score of the model, it is a performance metric for measuring the ability of the binary classifier to differentiate between the positive and negative classes. So, lets start examining the results of evaluation metrics…

If we investigate the below picture, we can observe the confusion matrix of the model-1 which is trained before the normalization or creating the principal components for the training data.

Graphical user interface, text, application, email

Description automatically generated

So, out of 261 class ‘0’ samples, the model correctly predicted 250 samples. If we consider class ‘1’, the model predicted 22 correct predictions out of 35 total predictions for that particular class.

Table

Description automatically generated

The insight from the above picture is, the model might be having 89% accuracy but the precision to correctly classify the class ‘1’ variables is 54% with a recall of 37%. As, our original data set is mostly consisting of class ‘0’ variables, this is expected that the model can only properly predict the class ‘0’ observations. Also, the weighted average will count the class imbalance problem in the training data, and the f1-score for the weighted average is 88%. But we may also need to look up of the goodness of the model in distinguishing both classes. As discussed, we can interpret that from the ROC-AUC score from the below picture. It is 66%, so the model is not randomly distinguishing the classes, which is a case when AUC score is 0.5. It is better identifying the difference between the classes.

Chart, line chart

Description automatically generated

Let us know check the metrics for the model-2, which is performed after scaling the training data to the unit variance using z-score normalization.

Graphical user interface, text, application, email

Description automatically generated

When compared to the model-1 confusion matrix, we can clearly see that there is absolutely no difference between predictions of class ‘0’. But when we look at the other class the correct predictions of total actual relevant samples are increased from 13 to 17.

Table

Description automatically generated

As observed in the confusion matrix, there is a clear increase in the precision in predicting the class ‘1’ samples. We almost achieving the precision of 61% with a recall of 49% leading to total f-score of 54%. Also, we are almost getting 90% weighted average accuracy in considering the class imbalance problem from the training data.

Chart, line chart

Description automatically generated

Finally, we can also observe an increase from 66% to 72% in the ROC-AUC score of the model-2, it means this model is better than the former one in differentiating the classes.

Now, we start evaluating the goodness of the final model which is after applying principal component analysis. From the below picture we can see that in the confusion matrix, this time the class ‘1’ values stayed the same. But there is an increase in the correctly classified negative class samples in the total actual relevant samples.

Graphical user interface, text, application, email

Description automatically generated

There is not much difference in the f-1 score as we are almost considering 8 components out of 11 total variables. The model has achieved the 65% precision for class ‘1’ and 93% precision for class ‘0’ with their respective recalls of 97% and 49%. Obviously, their f1- measures will be 95% and 56% for both the classes. The average weighted f1-score considering the class imbalance is constant at 90% from the last model.

Table

Description automatically generated

The last metric, from ROC-AUC score, there is not much improvement in the score. For the precious model it was 72% and now we are achieving an ROC-AUC score of 73%, which is the best among the all the models in distinguishing the classes, even with the class imbalance.

Chart, line chart

Description automatically generated

Note:

Graphical user interface, text, application, email

Description automatically generated

I also tried the feature selection technique from the machine learning, but using the coefficients of the logistic regression, I was able to achieve one order of attributes which are differing from range of importance of features from the Random Forest model. I was confused to select which one and did not perform any operation. But, if we can remove the less important features from the training data and we perform all the above techniques again, we can still improve the model performance.

Chart

Description automatically generated

### **PART V: Summary and Conclusion**

In the conclusion, it seems to me that the alcohol percentage in the wine, and the amount of vinegary taste i.e., volatile acidity is the best deciding factors in deciding the quality level of the wine. Out of all the models, my go to choose will be the last model as it is best in testing accuracy, best weighted average f1-score for the model which will consider the class imbalance problem and most importantly it is the best one in distinguishing the classes. All the models almost are performing a best job in predicting the low quality but coming to high quality they are gaining accuracy just above half. No surprise, as the raw data itself is consisting majority of its percent as normal wines, quality score of 5-6. It means, the data itself is having so much low-quality wines. This can also affect the model accuracy and feature importance decisions because more dominance of one category which would cause the model to take-up unnecessary features.