ME 315 Final Article Project

LSE ID: [redacted]
ME 315, Machine Learning in Practice
LSE Summer School

Abstract

In this project I am using the classification and regression machine learning methods to analyse two different datasets, both from the UC Irvine Machine Learning Repository. I first attempt to classify mushrooms as poisonous or edible based on their morphological features through a multitude of classification techniques. Secondly, I attempt to predict the quality of wine based on its objective physical and chemical properties using varying regression techniques.

1. Multivariate Analysis of Features of Agaricus and Lepiota Mushroom Species Classification

For this section I first introduce you to the dataset and its problems and walk you through my process of pre-processing, analysis, and cleaning up the data. After the refinement, I walk you through my methodology and decide which classification model has the best predictive ability.

1.1 Introduction to the Data

This dataset I am using for this project is from the UCI Machine learning Repository. The dataset was originally contributed by Jeff Sclimmer in 1987 and is based on the mushroom records from "The Audubon Society Field Guide to North American Mushrooms" (1981).

The dataset contains 8,124 samples of mushrooms from two different families of mushrooms, the Agaricus and Lepiota. Each sample is described by 22 physical attributes and is classified as either poisonous or edible. Some of the notable attributes include the mushroom's cap-shape, colour, habitat and odour. These 22 features provide a comprehensive description of each mushroom sample.

While the data is well structured and mostly complete, it does include over 2,000 null values, for one of the attributes. All variables in the dataset are categorical, none of them are numerical, so encoding is required to use any of the machine learning tools. The dataset does provide a near 50-50 split of poisonous and non-poisonous mushrooms, which allows for an unbiased training and evaluation.

1.2 Motivation

There exists no easy way for someone to determine whether an unknown mushroom species is poisonous or not, and there exist a multitude of poisonous mushrooms with similar physical qualities as edible fungi.These misclassifications result in an estimated 100 deaths a year, with a majority coming from less developed nations that rely on foraging [1].

Along with deaths there are tens of thousands of cases a year of mushroom poisoning that range from mild nausea to organ failure. With proper scaling and image classification models machine learning and data collection these illnesses and deaths could be reduced to zero.

The complex task of classifying mushrooms as poisonous or edible based on their physical properties is a perfect job for a classification model.

1.3 Data Pre-Processing

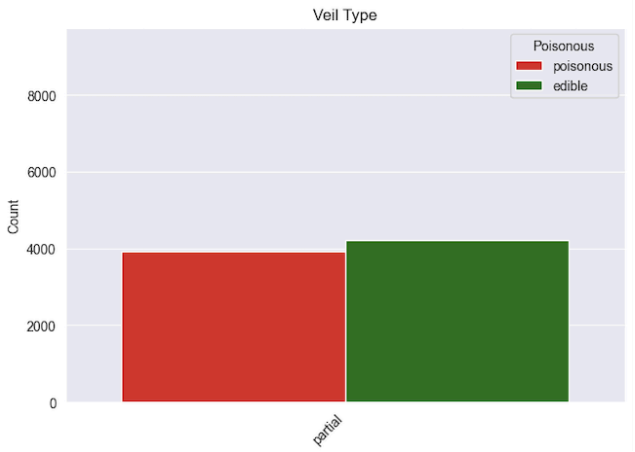
As stated in the introduction to the data, there were a multitude of challenges with the way the data came. First off, the data and the column names were separated from the data so I had to set an array of all of the feature names to be the names of the column property of the pandas dataframe.

Secondly, I decided to drop the 'stalk-root' column from the data due to the over 2,000 null values in the one column. I found this to be the better option than dropping the rows with null values because doing so would completely change the proportion of poisonous to edible from almost 50-50 to 38-62 which would cause greater bias in the classification. There exist 21 other full features and losing one mushroom feature was superior to losing 25% of the data points and making the model biassed.

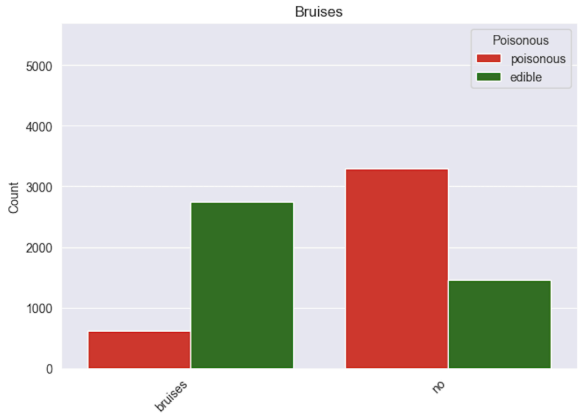
The next challenge I faced in the preprocessing was the fact that all of the variables were categorical not numerical. To fix this I could not just use dummy variables, because many of the features were not binary, so I had to use an encoder. I used the LabelEncoder method from the sklearn.preprocessing library to map non-negative integers to categorical variables. Iterating through each column in the dataset I was able to fit the encoded data to the original categorical values in a dictionary in order to interpret the encoded data. Leveraging AI I was also able to fit the real names, instead of abbreviated names as just characters for each piece of data for even better interpretability of graphs and other outputs.

1.4 Data Analysis

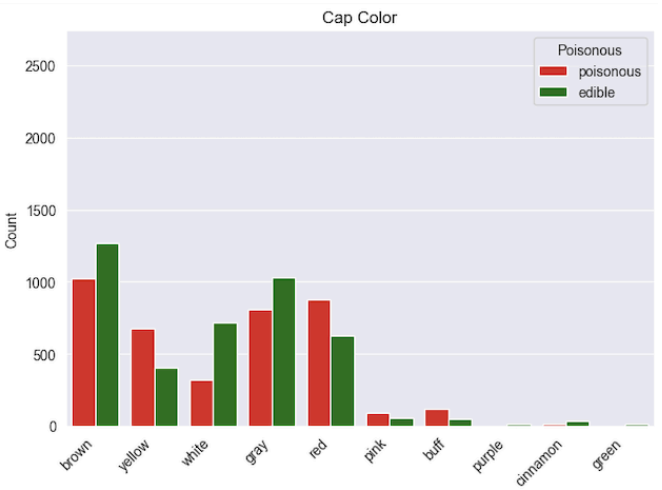
I made bar graphs to see the distribution of poisonous and edible mushrooms across each feature and noticed some interesting trends in the data. First off, I found there was absolutely no variation for the 'viel-type'. Every veil was partial, so I removed the data from my classification model because it contributed nothing to the model.

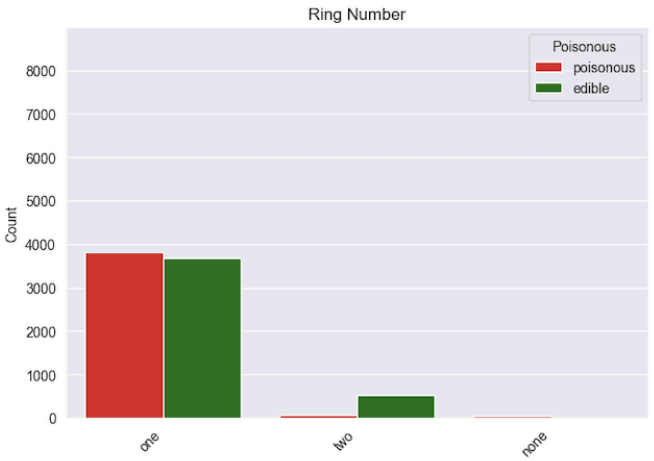


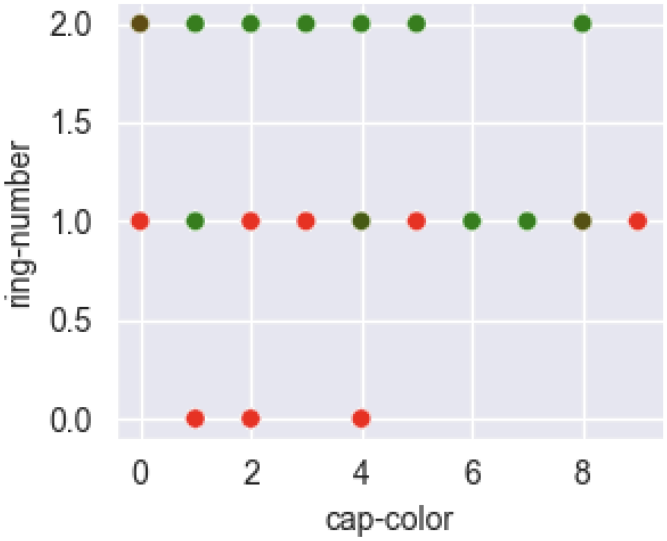
Some features were easily interpretable and you can see clear patterns in the data. These clear patterns can be used by humans to identify whether the mushroom is poisonous or not. Mushrooms with bruises were much more likely not to be poisonous than mushrooms without them.



Other variables however, were much less interpretable by just these bar graphs, such as cap colour and ring number but when you view them together you can see patterns emerge







A human cannot memorise all of these patterns in the data but a classification machine learning algorithm can easily

memorise these hidden patterns, and attempt to accurately interpret data that it has not seen before.

To prepare the data further to be used for the model I wanted to check for multicollinearity. I checked for multicollinearity using a heat map (Figure 1, in the Appendix). I found that veil colour and gill attachment had a correlation coefficient of 0.9 which indicated that there was multicollinearity between the two variables. I decided to remove 'veil colour' from my data to regain independence for my independent variables. After making the removal, none of my variables were incredibly highly correlated with one another.

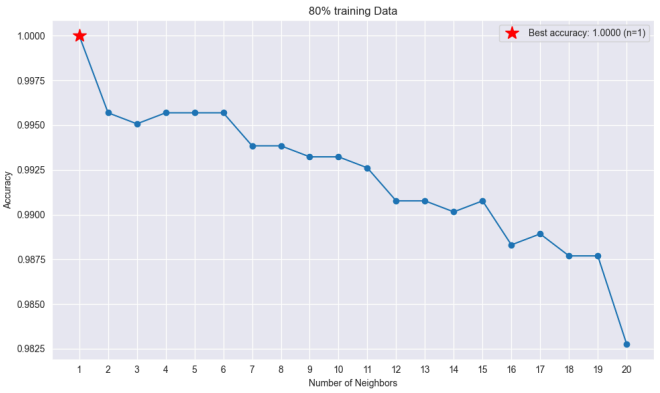
1.5 Applying the model

To begin the classification process, I employed the standard practice of splitting the data into training and testing sets using scikit-learn's train_test_split function. I first opted for an 80-20 split, allocating the majority of the data for model training while reserving a substantial portion for evaluation.

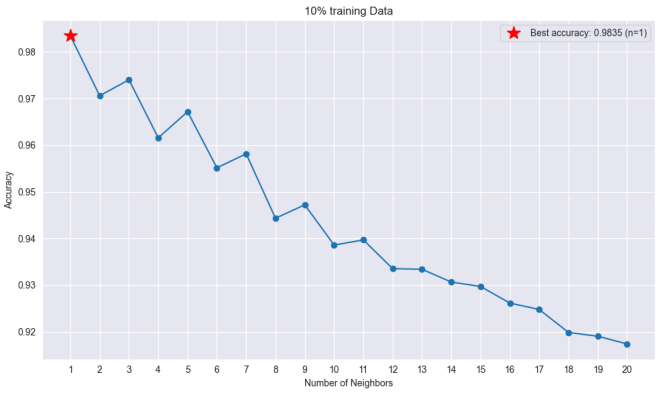
I first applied linear discriminant analysis. I fit the data and gave a prediction accuracy of 93.72%. Which is a somewhat high percentage, but the 6.27% error rate is way too high for the large real-world consequences for a misclassification. These misclassifications could result in sickness or in serious cases death, so a much higher success rate is necessary for this model to have any real-life applications.

Next, I tried quadratic discriminant analysis which gave me much more promising results. I fit the data and was returned a 98.03% accuracy, which is quite high. This performs much better than the linear discriminant analysis due to the nonlinear nature of the data. Different features have vastly different distributions of poisonous mushrooms. Some features have all poisonous mushrooms whilst others have mixed, there is no linear connection between the features and the quadratic model allowing better fits to different values and distributions, but this model with its near 2% error rate would be semi-useful but would still not cut it due to the dire consequences of a misclassification.

The final type of model I tried was the K-Nearest Neighbours Model and I got varying results for different numbers of neighbours. I fit the model for 1-20 neighbours and received much better results than the other two models I tried beforehand. The highest accuracy I achieved was 100%, with 1 neighbour. The accuracy goes down, the more neighbours I added. 100% accuracy is great, however, there are problems with this. This prediction only works well with a massive number of datapoints and parameters and will likely not perform as well on mutations and unknown subspecies, species in the same family that are not in the dataset. Intuitively, the nearest neighbour's assumption makes sense. Two species of mushrooms that have a lot in common with each other, but some small differences will likely have the same edibility. The machine can have greater accuracy than a human can and take every feature of the mushroom into account which results in better results than any human can have.



However, accounting for data about the mushrooms not in the dataset, mutations, and subspecies that have not been discovered, I think that using 3 neighbours is the best option for the model. This adds more redundancy in the estimations, and through my tests, it is closer in rate at lower sample sizes and theoretically provides more redundancy for new real-world data, not present in the dataset.



1.7 Conclusion

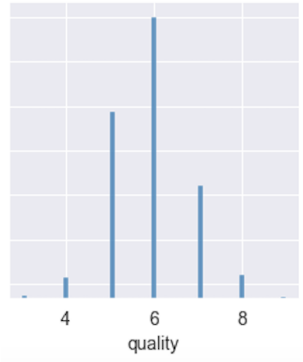
The best predictive model for this dataset was KNN, and this is mainly because of the categorical nature of all of the data. It requires a lot of training data in order to be the most accurate and the more parameters the more accurate the model performed, and with a large amount of training data it got to 100% accuracy on the data we tested it on, but in the real world will not have as high accuracy. To be applicable to real life this model requires extensive data collection and scaling, but in the future a model like it could save lives.

2. Does Wine's Chemistry Predict its Quality?

In this regression model I attempt to predict the quality of wine based on the physicochemical properties of the wine. I look into issues with the data and attempt to create a model that is useful using regression techniques

1.1 Introduction to the data

The dataset that I am using for my regression and analysis on wine quality is a 2009 dataset of the wines physicochemical properties, along with the wine quality rating an expert gave through a blind taste test. There are 11 different numerical physicochemical attributes listed and the ratings the wine experts gave were integers 1-10. The important attributes included volatile acidity, residual sugar, alcohol content and sulphur dioxide levels. The rating was the median score given by 3 blind taste tests to get a more 'representative', less extreme distribution of scores. [2]



As you can see the wine scores follow a pretty normal distribution. The dependent variable in the regression is the score, and it is important to note that this score; given by an expert, is still a subjective measure. Being an expert in wine does not present the expert with more objective knowledge gained by an expert in a field such as physics. This subjective and relatively non-scientific y variable based on 'expert' opinions makes the regression much harder and an interesting task for machine learning models.

In the dataset there are 4898 ratings and physicochemical measurements. This leaves us with a large dataset that should give a less biassed answer as to whether quality of established wine brands and vineyards is based on their physical properties.

1.2 Motivation

My motivation for choosing to attempt to answer this particular question and model is because I like wine, and I am sceptical of wine experts. I wanted to prove myself wrong and create a model that can predict the subjective quality of its objective attributes. I knew this would be a difficult task and if a reliable regression equation is to be found, then wine makers could use this model and use methods that would result in measures of these objective qualities that would result in a better quality. This could also allow for greater mass production of quality wine to take place and allow for the 'elite' world of wine to be a more inclusive place

1.3 Data Preprocessing

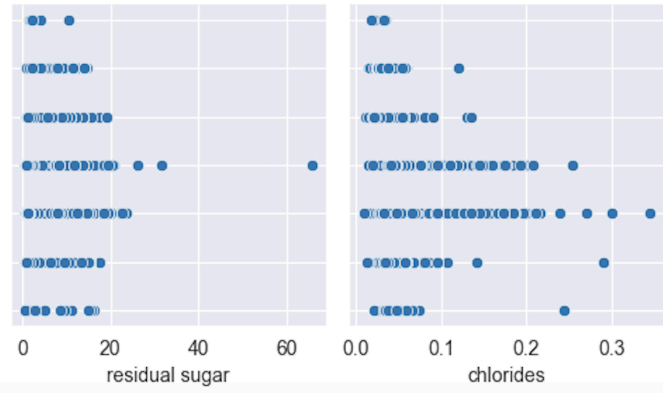
I downloaded the data as a csv and for this dataset it had each column name at the top of each column and had all numerical variables. Each physico chemical element were continuous floating point variables, and the rating is an integer 1-10. The fact that the Y values were integers rather than floats made this regression harder than if it were more exact, but there is nothing I can do to fix the inconvenience without altering the integrity of the model.

I did not make any alterations to the data until starting my analysis of the relationships and natures the data possessed.

1.4 Data Analysis

In my data analysis I first looked at the correlations between all of my variables. I created a correlation heatmap and found that density was highly positively correlated with residual sugar, and highly negatively correlated with alcohol. This measurement being highly correlated with two other variables I feared would lead to multicollinearity. Therefore, instead of removing both alcohol and residual sugar I decided to remove density from my regression.

Next, I made a pairplot of all of the variables to explore the relationship between different variables further. I found some interesting trends between the dependent and independent variables that I thought could improve the regression.



I noticed a nonlinear trend between residual sugar and the score, and chlorides and the score. There seemed to be higher sugar amounts in average wines, but less in the worst and best wines. There also seemed to be a similar trend with chlorides and the score. So I decided to add two exponential terms to account for this seemingly nonlinear relationship. I noticed no other substantial trends in the data, so I decided to start applying the regression machine learning techniques.

1.5 Applying the models

I started the application by setting the X variable to be all columns but quality in my transformed dataset, and I set Y to be the quality. Next, I performed a train test split with 80% training data and 20% test data to achieve a representative regression equation. Then I performed a simple linear regression with all variables to test its performance. The regression had a mean squared error of 0.5805, which I used as the baseline for the rest of my tests.

I also tested a simple linear regression where I rounded each prediction to its nearest integer value, but this resulted in a mean squared error of 0.6234 so I decided not to carry through with this method of rounding my predictions.

Next, I printed a summary of statistics of this regression to see which variables were significant, and the coefficients of each variable in the regression. Citric acid, chlorides, pH, and the two squared variables I added in were all insignificant. Knowing this I realise that I should cut down on the parameters used in the regression.

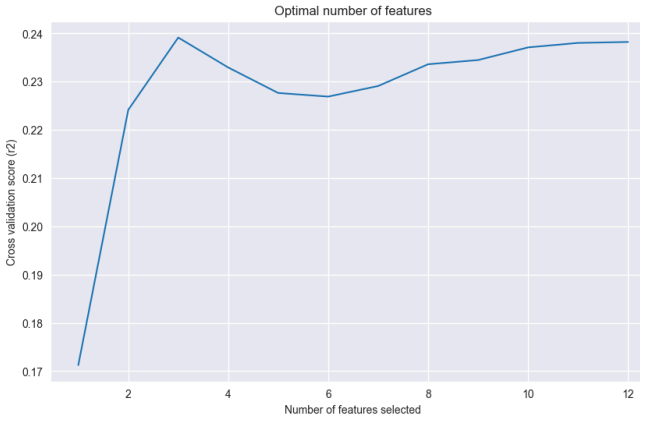
Also, listed in the summary statistics, the R^2 value was only 0.272, which means that only 27% of the variance is explained by the model, which is a low percentage. However, I expected this, this model is trying to predict subjective taste tests off of 10 physical attributes of the wine. Despite the low R^2, there is still a lot of knowledge to gain off of this dataset.

Next, I tried a lasso regression. I found the best alpha parameter to be very small at just 0.00255. This adjusted the parameters but actually raised the MSE. I scraped using a lasso regression with this number of parameters

I did the same for the ridge regression and found that it also raised the MSE, by a much smaller amount. With an alpha of 70 the MSE was 0.5806 which was just bigger than the normal regression.

I performed the same transformation of the model with an elastic net and predictably the MSE went up when the two methods were combined.

I could not find a model that substantially lowered the MSE, so I went back to changing the parameters. I decided to plot the cross validation score of the regression of the best parameters at each step.



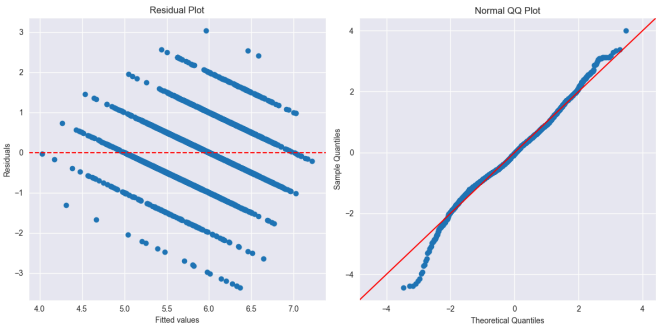
As you can see, the model peaks at three parameters and its predictive ability drops and then slowly raises as the model gets more complex. This told me that most of the independent variables in the model to predict the wine quality were useless.

Using recursive feature elimination we found the best three parameters for predicting the wine quality to be the volatile acidity, the residual sugar, and the alcohol content. I ran the regression with those three features and was given an R^2 of 0.261 which is impressive because we removed 10 parameters from the equation and only got a drop of 0.01 for the R^2. The whole regression is attached as figure 3 in the appendix.

The intercept is 2.3908. The volatile acidity's coefficient is -2.198. So holding all else constant if the acidity measurement is 1 unit higher, on average the score is 2.198 less. The residual sugar coefficient is 0.0266, and the coefficient of alcohol is 0.3736.

So the wines with more alcohol content, residual sugar and less volatile acidity tended to be higher rated than those that had the converse. I also attached two pairplot in the appendix that allow you to visualise these trends in the data. One demonstrates the density of the data points across the pair plots with low opacity points (Figure 4 in appendix). The other shows the distribution of quality across each of the pairplots to demonstrate how the interactions between variables result in different quality wines (Figure 5 in appendix).

I wanted to test again the validity and accuracy of this statement so I looked at the residuals



The residual chart looks different than your average chart of all continuous variables, but that is due to the y variable being an integer, that it has the strange lines. Also, the QQ plot shows that the model underestimates values closer to the intercept. That is due to the intercept being much lower than the lowest values in the dataset. The data underestimates these lower values because the y variable is a median of a couple of experts and contains a normal distribution with not many extremes.

1.6 Conclusion

This model is imperfect, but taught me a lot about the process of fitting a good regression model. This process also taught me how in social and subjective situations simple rules of thumb and lowering the scope can lead to better results; looking at less parameters gave us a model that was better. Thank you, I learned a lot about machine learning, and will continue the journey of trying to make a useful model.

Appendix

Figure 1 – Mushrooms Correlation Heatmap

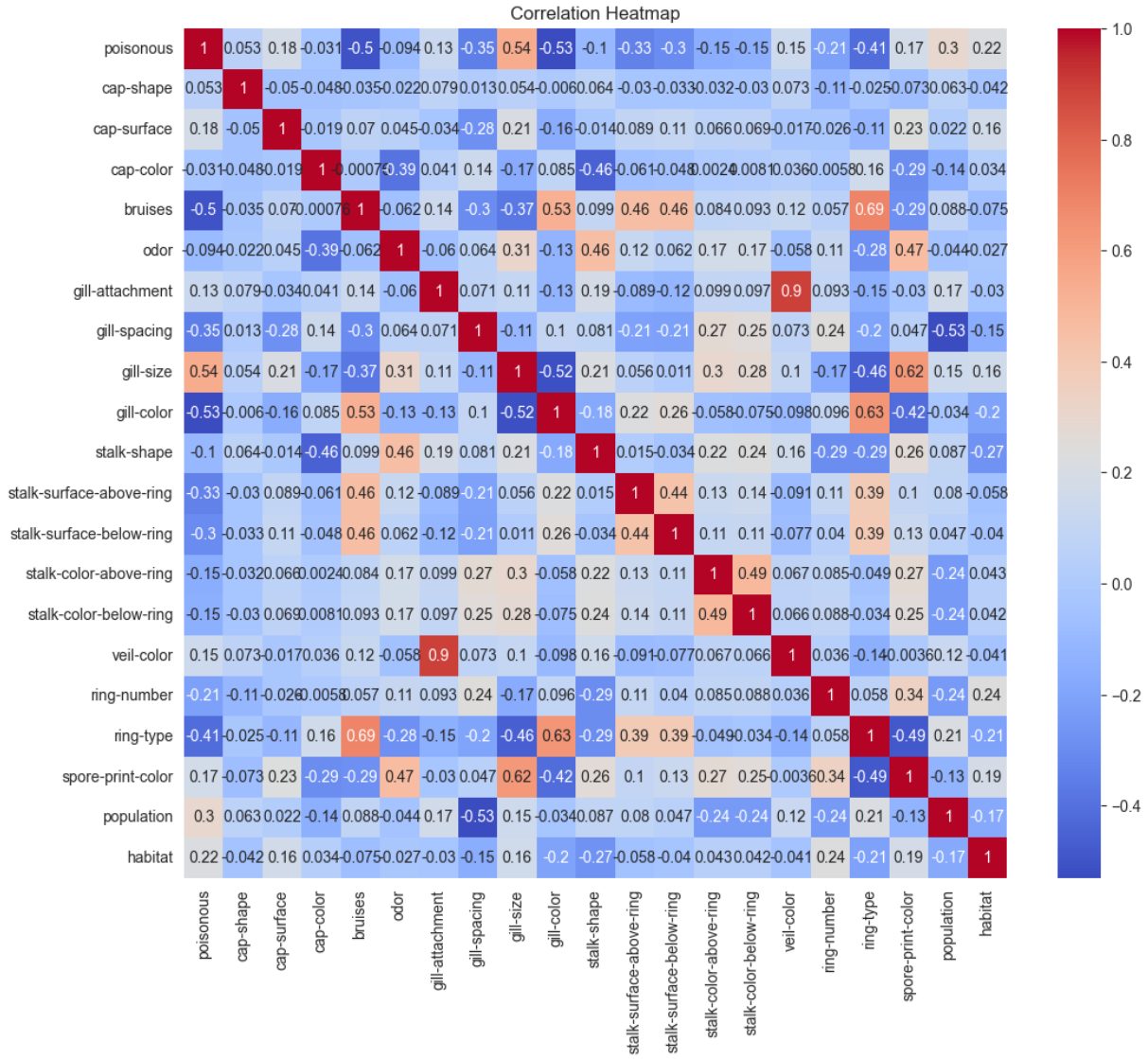


Figure 2 – Wine Correlation Heatmap

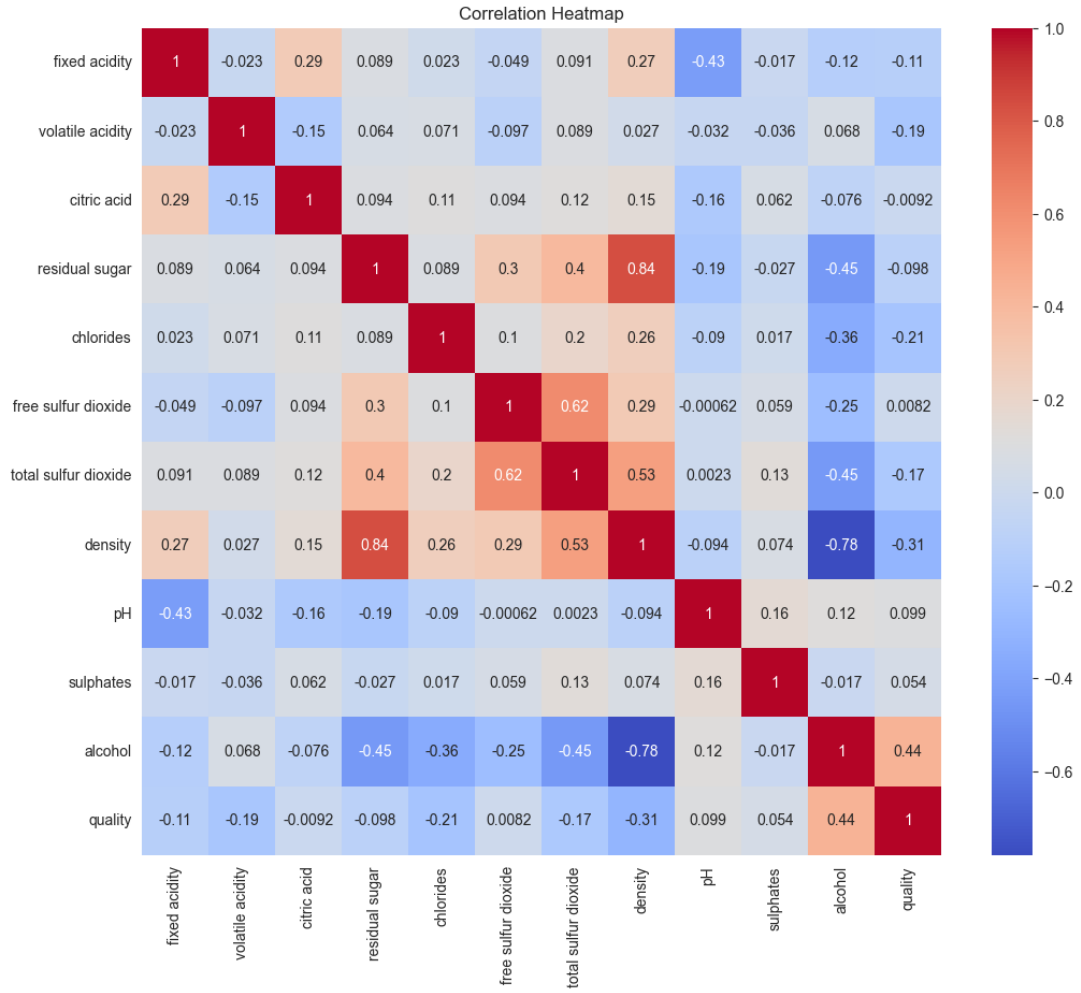


Figure 3 – Regression 3 Variable results

OLS	Regression	Results				
Dep. Variable:	quality	R-squared:	0.261			
Model:	OLS	Adj. R-squared:	0.26			
Method:	Least Squares	F-statistic:	459.9			
Date:	Thu, 4 Jul 2024	Prob (F-statistic):	5.58E-256			
Time:	17:39:02	Log-Likelihood:	-4479.9			
No. Observations:	3918	AIC:	8968			
Df Residuals:	3914	BIC:	8993			
Df Model:	3					
Covariance Type:	nonrobust					

	coef	std err	t	P>|t|	[0.025,0.975]
const	2.3908	0.127	18.878	0	2.142	2.639
volatile acidity	-2.198	0.124	-17.779	0	-2.44	-1.956
residual sugar	0.0266	0.003	9.999	0	0.021	0.032
alcohol	0.3736	0.011	33.618	0	0.352	0.395

Omnibus:	76.626	Durbin-Watson:	1.984
Prob(Omnibus):	0	Jarque-Bera (JB):	159.819
Skew:	0.03	Prob(JB):	1.98E-35
Kurtosis:	3.988	Cond. No.	140

Figure 4 – Regression 3 Pairplot Seethrough

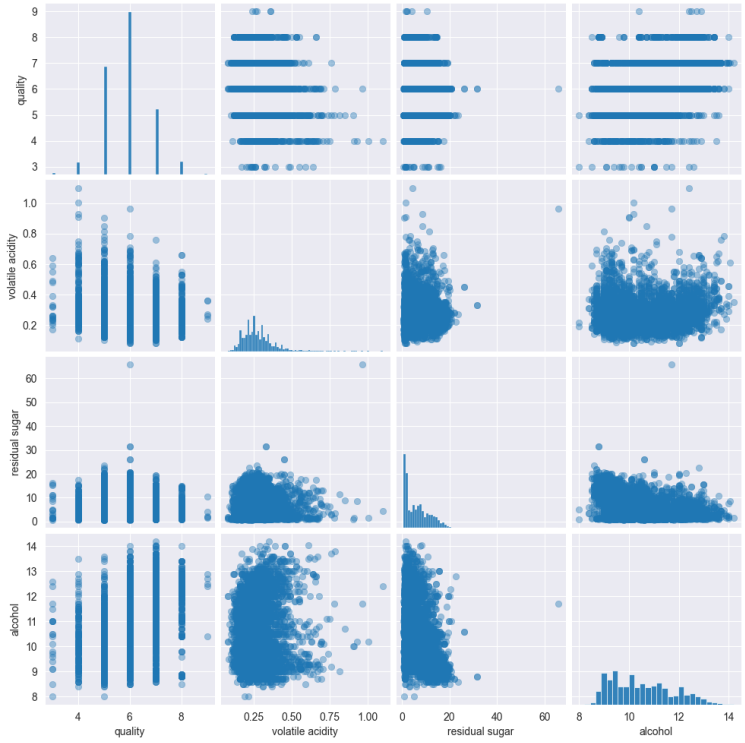
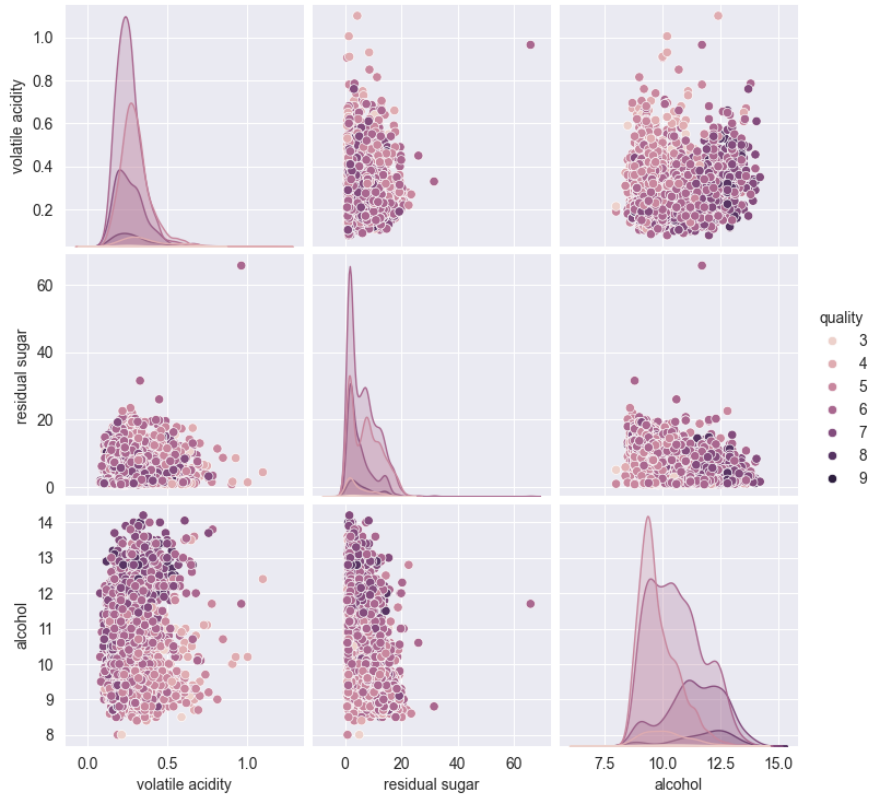


Figure 5 – Regression 3 Pairplot Quality Hue



References

[1] Li H, Zhang H et al 2020 *China CDC Weekly* **2**(2) 19-24
[2] Cortez P, Cerdeira A et al 2009 Decision Support Systems 47(4) 547-553