Thomas Jones CS5565-0007 Final Project

|  |  |
| --- | --- |
| Link | URL |
| Video | <https://umsystem.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=99710f71-5363-4aeb-b1cb-b248002eeffe> |
| Main Notebook | <https://colab.research.google.com/drive/1lCykTyj7_4GFTWHzxN4icNJXN-klsyTv?usp=sharing> |
| MNIST | <https://colab.research.google.com/drive/1W72h-9_8xtLXgHS9XU69-UJdzPdZu_q3?usp=sharing> |

# Overview

The wine dataset was selected for both the regression and classification problems. The data provided a good mix of opportunities for showing the various models.

fixed acidity 96

volatile acidity 143

citric acid 80

residual sugar 91

chlorides 153

free sulfur dioxide 60

total sulfur dioxide 144

density 436

pH 89

sulphates 96

alcohol 65

quality 6

*features and unique counts for each feature.*

The feature types are shown below.

fixed acidity float64

volatile acidity float64

citric acid float64

residual sugar float64

chlorides float64

free sulfur dioxide float64

total sulfur dioxide float64

density float64

pH float64

sulphates float64

alcohol float64

quality int64

The pairplot of the features is shown below. This gives is a visual representation of the feature interactions as well as the distributions of each feature.

A diagram of a graph

Description automatically generated

We can also visualize the correlation between features via a heatmap.

A colorful squares with white text

Description automatically generated with medium confidence

# Section A

## Part 1 – Regression

For the regression exercises we will attempt to predict alcohol content as a function of density. Performing a simple linear regression against the normalized density feature produces the following.

A diagram of a bar graph

Description automatically generated

Where,

Mean Squared Error: 0.8659291296380478

R2 Score: 0.2235421964859452

While there is some relation the low R2 indicates a poor fit of the data.

We further create polynomial regressions of the data using degrees of 2 to 5 which.

+--------+--------------------+---------------------+

| Degree | Mean Squared Error | R2 Score |

+--------+--------------------+---------------------+

| 2 | 0.7278568942954579 | 0.3473482458622895 |

| 3 | 0.727090861248048 | 0.3480351292538434 |

| 4 | 0.6999610409980527 | **0.37236178592821534** |

| 5 | 0.7081564105335082 | 0.36501319536729304 |

+--------+--------------------+---------------------+

A diagram of a graph

Description automatically generated

A diagram of a graph

Description automatically generated

A diagram of a line graph

Description automatically generated with medium confidence

A diagram of blue dots

Description automatically generated

A diagram of a line with a red line

Description automatically generated

A diagram of a graph

Description automatically generated

A diagram of a curve

Description automatically generated

A diagram of a graph

Description automatically generated

The results show, that at best, we can account for only 37% of alcohol variability from density.

Adding more features and moving to a multi-feature linear regression we find that a second degree polynomial fit accounts for a little more than 74% of the alcohol variability.

+--------+---------------------+----------------------+

| Degree | Mean Squared Error | R2 Score |

+--------+---------------------+----------------------+

| 1 | 0.374859422965879 | 0.6638725800755837 |

| 2 | 0.28508554068059944 | **0.7443706590364958** |

| 3 | 1.1183386100689021 | -0.00278730791978532 |

| 4 | 1919.7178350421686 | -1720.364765944126 |

| 5 | 437630594686.16156 | -392412818404.3599 |

+--------+---------------------+----------------------+

We also can see that while for density, the 2-degree poly fit was best, the model shows overfitting. This is possibly due to outliers in the original data though that analysis would have to be done as a continuing exercise.

A graph with blue dots

Description automatically generated

A graph with blue dots

Description automatically generated

A graph of a graph with blue dots

Description automatically generated

A graph with blue dots

Description automatically generated

A graph with blue dots

Description automatically generated

## Part 2 – Feature Selection

Based on the regression results, we first select the 2nd degree polyfit against which to select features. The full list of these features is:

('chlorides', 'chlorides density', 'chlorides free sulfur dioxide', 'chlorides pH', 'chlorides quality', 'chlorides sulphates', 'chlorides total sulfur dioxide', 'chlorides^2', 'citric acid', 'citric acid chlorides', 'citric acid density', 'citric acid free sulfur dioxide', 'citric acid pH', 'citric acid quality', 'citric acid residual sugar', 'citric acid sulphates', 'citric acid total sulfur dioxide', 'citric acid^2', 'density', 'density pH', 'density quality', 'density sulphates', 'density^2', 'fixed acidity', 'fixed acidity chlorides', 'fixed acidity citric acid', 'fixed acidity density', 'fixed acidity free sulfur dioxide', 'fixed acidity pH', 'fixed acidity quality', 'fixed acidity residual sugar', 'fixed acidity sulphates', 'fixed acidity total sulfur dioxide', 'fixed acidity volatile acidity', 'fixed acidity^2', 'free sulfur dioxide', 'free sulfur dioxide density', 'free sulfur dioxide pH', 'free sulfur dioxide quality', 'free sulfur dioxide sulphates', 'free sulfur dioxide total sulfur dioxide', 'free sulfur dioxide^2', 'pH', 'pH quality', 'pH sulphates', 'pH^2', 'quality', 'quality^2', 'residual sugar', 'residual sugar chlorides', 'residual sugar density', 'residual sugar free sulfur dioxide', 'residual sugar pH', 'residual sugar quality', 'residual sugar sulphates', 'residual sugar total sulfur dioxide', 'residual sugar^2', 'sulphates', 'sulphates quality', 'sulphates^2', 'total sulfur dioxide', 'total sulfur dioxide density', 'total sulfur dioxide pH', 'total sulfur dioxide quality', 'total sulfur dioxide sulphates', 'total sulfur dioxide^2', 'volatile acidity', 'volatile acidity chlorides', 'volatile acidity citric acid', 'volatile acidity density', 'volatile acidity free sulfur dioxide', 'volatile acidity pH', 'volatile acidity quality', 'volatile acidity residual sugar', 'volatile acidity sulphates', 'volatile acidity total sulfur dioxide', 'volatile acidity^2')

The selected feature, stepping through is then:

('citric acid', 'citric acid chlorides', 'citric acid quality', 'citric acid total sulfur dioxide', 'citric acid^2', 'density', 'density sulphates', 'density^2', 'fixed acidity', 'fixed acidity chlorides', 'fixed acidity free sulfur dioxide', 'fixed acidity quality', 'fixed acidity total sulfur dioxide', 'fixed acidity volatile acidity', 'fixed acidity^2', 'free sulfur dioxide density', 'free sulfur dioxide pH', 'free sulfur dioxide quality', 'pH', 'pH quality', 'quality', 'residual sugar', 'residual sugar chlorides', 'residual sugar free sulfur dioxide', 'residual sugar quality', 'residual sugar^2', 'sulphates', 'sulphates^2', 'total sulfur dioxide', 'total sulfur dioxide sulphates', 'volatile acidity', 'volatile acidity citric acid', 'volatile acidity free sulfur dioxide', 'volatile acidity pH', 'volatile acidity residual sugar', 'volatile acidity^2')

A graph of steps of forward stepwise

Description automatically generated

A graph of a curve

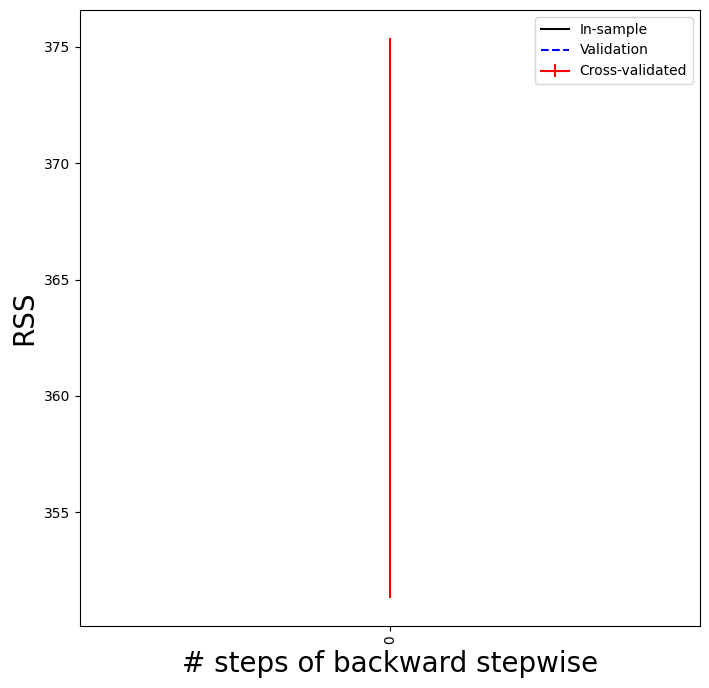
Description automatically generated with medium confidence

*Forward Selection*

The backward steps for the code as written did not function as expected. Further debugging would be necessary.

I also made changes to the original Chapter-6 code, simply switching forward for backward and the results were the same:

<https://colab.research.google.com/drive/1PgNaAe7yTLMe6Kc3CoSS-TRIzW8wDwo4?usp=sharing>



The forward steps do show a significant reduction in marginal quality after around 8 features, though the “ideal” value selected by forward stepping was around 37.

### PCR

PCR was run against the 2nd degree polyfit of the data across a range of 1 feature to all 77.

A graph with blue lines

Description automatically generated

Given the sharp decline in MSE at around 19, that is likely be best feature count. The polyfit added a significant number of features which might explain some of the cliff like drop-offs.

## Part 3 – Classification

For classification the quality data was merged into 2 categories, “good” and “bad”. Any quality of 5 or less is “bad” and 6 or more is “good”. This allowed for a better split of the data to deal with imbalances in the dataset.

A screenshot of a phone

Description automatically generated

### Logistic Regression

A simple logistic regression of the data produced an accuracy of 73%.

Accuracy of logistic regression classifier on test set: 0.73

[[172 62]

[ 68 178]]

precision recall f1-score support

0 0.72 0.74 0.73 234

1 0.74 0.72 0.73 246

accuracy 0.73 480

macro avg 0.73 0.73 0.73 480

weighted avg 0.73 0.73 0.73 480

### K-Fold Polyfit and LDA Regression

Polynomial fit was evaluated from 1 to 5 degrees.

A graph with blue lines

Description automatically generated

As was LDA

A graph with numbers and lines

Description automatically generated

Based on the mean of the results regression were performed on the 4th degree polyfit as well as the LDA of the 3rd degree polyfit.

Accuracy of logistic regression classifier on test set: 0.73

[[174 60]

[ 68 178]]

precision recall f1-score support

0 0.72 0.74 0.73 234

1 0.75 0.72 0.74 246

accuracy 0.73 480

macro avg 0.73 0.73 0.73 480

weighted avg 0.73 0.73 0.73 480

Accuracy of LDA classifier on test set: 0.72

[[165 69]

[ 65 181]]

precision recall f1-score support

0 0.72 0.71 0.71 234

1 0.72 0.74 0.73 246

accuracy 0.72 480

macro avg 0.72 0.72 0.72 480

weighted avg 0.72 0.72 0.72 480

Neither polyfits produced any better results against the test data. Evaluating the graphs it becomes clear that although the mean was higher for those degrees the other values fell within the error bars of the highest values.

# Section B

## Part 1 – Splines and Step Functions

A spline and step regression was performed to assess density vs alcohol again.

For the step regression a k-fold validation was performed to identify the best number of steps to consider.

A graph with blue lines

Description automatically generated

Using 3 steps the following regression line is produced. The step output is the average of all the datapoints within each step. It does provide a good visualization of where the steps came from, i.e. parsimonious.

A diagram of alcohol versus density

Description automatically generated

For the spline fitting, a degree 3 polynomial was used and the number of knots varied from 2 to 15.

A graph with blue lines

Description automatically generated

Based on the lowest 3 knot MSE we get the following regression. This regression is a somewhat better fit than some of the linear or poly regressions, though at the edges we do see the characteristic tail extremes due to the cubic fit.

A diagram of a line with a red line

Description automatically generated

## Part 2 – Trees and SVM

### Tree Classification

The default decision tree returned an accuracy of 67% on the data. As decision trees are some of the simplest but greediest algorithms we would expect a lower score.

Accuracy of classification tree classifier on test set: 0.67

[[152 82]

[ 75 171]]

precision recall f1-score support

0 0.67 0.65 0.66 234

1 0.68 0.70 0.69 246

accuracy 0.67 480

macro avg 0.67 0.67 0.67 480

weighted avg 0.67 0.67 0.67 480

For the random forest classifier, a k-fold validation was run,

A graph of trees and numbers

Description automatically generated

Then selecting the best tree count

{'n\_estimators': 500}

Accuracy of random forest classifier on test set: 0.79

[[190 44]

[ 55 191]]

precision recall f1-score support

0 0.78 0.81 0.79 234

1 0.81 0.78 0.79 246

accuracy 0.79 480

macro avg 0.79 0.79 0.79 480

weighted avg 0.79 0.79 0.79 480

Finally, a gradient boosted tree was used, again with k-fold validation on the number of trees.

A graph with blue lines

Description automatically generated

{'n\_estimators': 200}

Accuracy of gradient boosting classifier on test set: 0.79

[[182 52]

[ 48 198]]

precision recall f1-score support

0 0.79 0.78 0.78 234

1 0.79 0.80 0.80 246

accuracy 0.79 480

macro avg 0.79 0.79 0.79 480

weighted avg 0.79 0.79 0.79 480

### SVC

The default rbf kernel was selected for the analysis. Values of C were check from 0.01 to 500 in logistically progressing steps. The best C from a k-fold analysis was done.

A graph with blue lines

Description automatically generated

{'C': 50}

Accuracy of support vector classifier on test set: 0.75

[[179 55]

[ 65 181]]

precision recall f1-score support

0 0.73 0.76 0.75 234

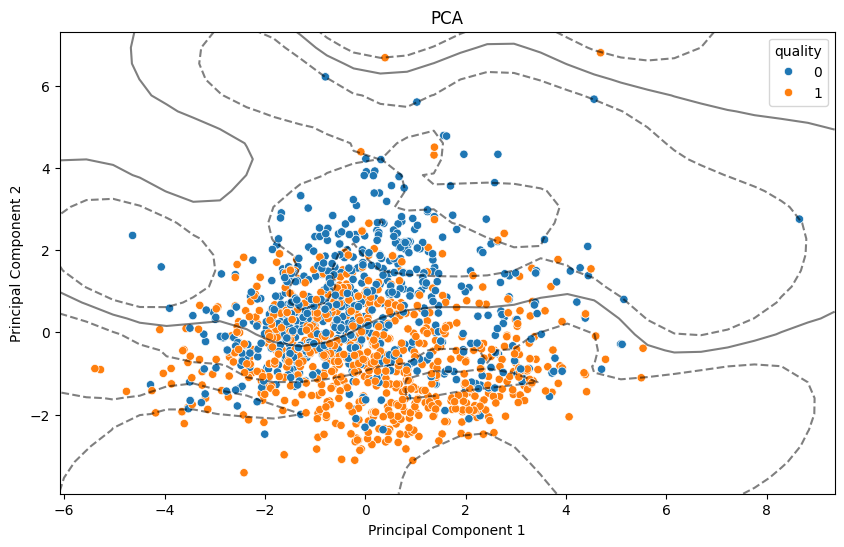
1 0.77 0.74 0.75 246

accuracy 0.75 480

macro avg 0.75 0.75 0.75 480

weighted avg 0.75 0.75 0.75 480

And then the boundary drawn against all the data predicted using the top 2 principal components.



Where 0 is “bad” and 1 is “good”.

While there is some separation at the first two principal components there is not a clear separability for the data. We do see the characteristic strange shapes indicative of higher dimensional fits present with rbf kernels.

## Part 3 - MLP

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Topologies** | **Sigmoid Function** | | | | | | **Relu Function** | | | | | |
|  | Training Accuracy | | | Testing Accuracy | | | Training Accuracy | | | Testing Accuracy | | |
| **[256,96,32,10]** | 53.95% | | | 65.02% | | | 96.86% | | | 96.78% | | |
| **[96,32,10]** | 90.16% | | | 90.96% | | | 96.36% | | | 96.35% | | |
| **[128,64,32,10]** | 54.92% | | | 66.45% | | | 96.53% | | | 96.73% | | |
| **Best Topology Selected** | **[256,96,32,10] - Relu** | | | | | | | | | | | |
| **Learning Rate** | **0.1** | | | **0.01** | | | **0.001** | | | **0.0001** | | |
|  | Training Accuracy | Testing Accuracy | | Training Accuracy | | Testing Accuracy | Training Accuracy | Testing Accuracy | | Training Accuracy | | Testing Accuracy |
|  | 96.86% | 96.78% | | 89.49% | | 90.25% | 25.06% | 27.49% | | 7.99% | | 8.05% |
| **Best Learning Rate Selected** | **[256,96,32,10] – Relu – 0.1** | | | | | | | | | | | |
| **Batch Size** | **128** | | | | **64** | | | | **32** | | | |
|  | Training Accuracy | | Testing Accuracy | | Training Accuracy | | Testing Accuracy | | Training Accuracy | | Testing Accuracy | |
|  | 96.86% | | 96.78% | | 97.42% | | 97.34% | | 97.44% | | 96.86% | |
| **Best Batch Size Selected** | **[256,96,32,10] – Relu – 0.1 - 64** | | | | | | | | | | | |
| **Overall Best Model Performance** | **Overall Best Model Training Accuracy** | | | | | | **Overall Best Model Testing Accuracy** | | | | | |
|  | 97.42% | | | | | | 97.34% | | | | | |

* Compare the topologies and provide the reason the chosen topology worked out to be the best along with the activation function?
  + *The best topology allowed for the most connections, while this might be a cause for overfitting, given the training size and variety of data the number of weights allowed for good generalization. The choice of Relu as the activation function prevented saturation of the neuron activations and kept the weights from reaching extreme values with little activation response.*
* Does topology and activation function depend on each other? If yes, then how and why?
  + *Yes, the deeper the network the more the network will be susceptible to vanishing gradients. Relu allows the errors to propagate farther back into the layers.*
* What learning rate felt like the best choice and why?
  + *The learning rate of 0.01 “felt” like a better choice and it is possible that with more training the network might have found a better local minima. While the experiment results show differently, a strategy of decaying the learning rate as marginal error reduction occurred would be called for.*
* What impacted the batch size on the performance of the model?
  + *Lower batch sizes allowed for more movement within the error space. However, depending on the order of the training data, the network might move to a local minima favoring a particular set of training data.*
* Are the learning rate and batch size correlated? What notable changes were observed during the entire process?
  + *Yes, higher learning rates with lower batches will cause the network to “jump around” chasing particular error conditions. Since the errors are averaged across batches higher error rates allow for change that are generally more directionally correct.*