Machine Learning Report

Alexander Montgomerie-Corcoran
Imperial College London
CID: 01052454
am9215@ic.ac.uk

1. Introduction

The problem posed by this data set is to predict the quality of *Vinho Verde* wine which has been classified by wine tasters. The issue which this report hopes to address is linking physio-chemical data to wine preference. This task can be approached as either a regression (learning a continuous function) or classification problem (learning discrete classes). Both approaches will be evaluated. For classification, Identity error will be used, and it's analog, mean absolute error, will be used for regression to evaluate errors.

1.1. Data Curation

The data is comprised of a set of red wine features and quality (1,659 samples) as well as for white wine (4,898 samples). To summarise the difference between white and red, an extra feature is added to the data (1 for red and -1 for white). Due to bias in features such as pH level being mostly acidic for wine, and difference in variance for features, all features are normalised to zero mean, unit variance [1]. Outlier detection is neglected in this investigation, as the assumption is made that all feature points are precise and contribute to the quality of wine. The data is split into 80% training and 20% test data. Cross Validation used in this report uses 10 folds, with the error being the average taken across all folds. The standard deviation across each fold is also observed to give an indication of the confidence in the error. Cross Validation is used to give a rough estimate of the Test Error and is used for choosing hyper-parameters. This report approaches the problem deterministically, in the sense that a relationship is trying to be found between the features that directly map to the given quality. It is understood that due to inaccuracy and subjectiveness of wine quality in terms of taste, the data will be inherently noisy.

[1] normalisation is applied to training data, and normalisation by the same bias and variance is also applied to the test data.

2. Regression

As a baseline predictor, multiple regression techniques are explored to develop a simple understanding of the relationship between input features and the wine quality. It is also a simple model that performs well for regression problems, which this dataset presents itself as. It is important to note that due to the size of the data-set (6,557), linear regression loosely avoids overfitting, according to the VC dimension rule of thumb [2] [2]. A loss function of $l(w) = \frac{1}{N}\sum_{n=1}^{N}(w^T \cdot x_n - y_n)^2$ is used for the linear method, with additional penalties for each regularised linear regression method. This loss function is differential and convex[3] with a simple, closed form solution that uses Penrose's psuedo-inverse[5] $w = X^+Y$ to compute weights. This alleviates the need for complex gradient descent methods or other solvers, making training extremely fast.

2.1. Simple Linear Regression Model

The first model evaluated has a loss function of $E_{test}(w) = \sum_{n=1}^{N}(w^T \cdot x_n - y_n)^2$. Factors that affect the accuracy of this model are the features and size of data. To explore how cross validation error depends on features, the cross validation for each combination of features is calculated, and the most interesting results are documented in 3. It's observed that the type of wine doesn't contribute to the best Cross Validation error, suggesting that red and white wine quality are affected by the same features.

2.2. Regularised Linear Regression Models

In order to improve on model generalisation, constraints are put on model weights. This has a smoothing effect, in the sense that no single weight contributes more than others. Regularisation requires hyper-parameters, which require tuning. Cross validation scores across different hyper-parameter values are used to gauge optimal settings for hyper-parameters. It's important to note that the equivalent loss function for the following cases is $E_{reg}(w) =$

[2] the data has a VC dimension of 12, and rule of thumb is to have 10x VC dimension, which this data satisfies

[3] for more training examples than features

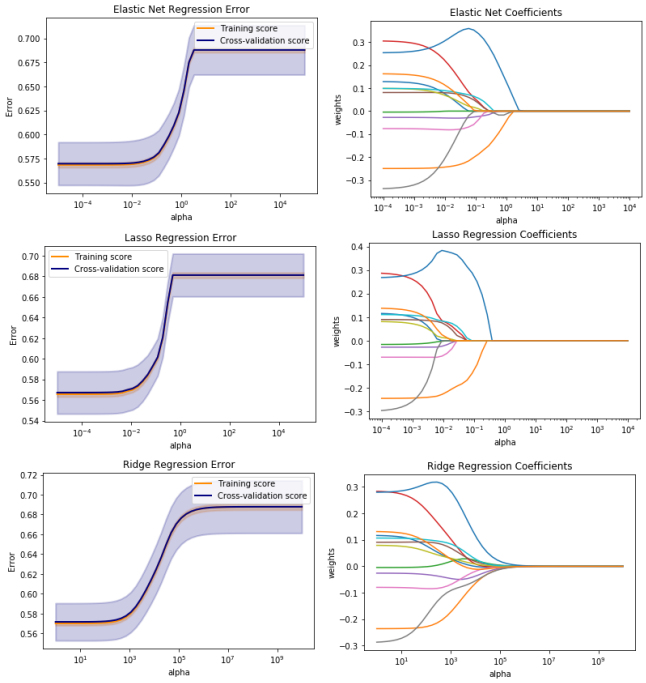

Figure 1. Regularised Regression Curves

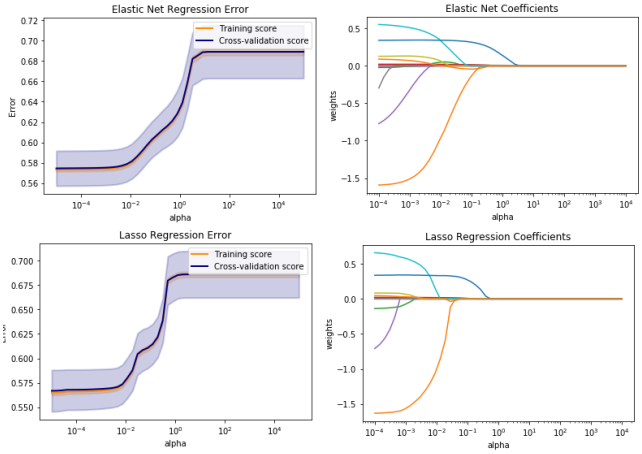

Figure 2. Regularised Regression Curves (no normalisation)

features	CV MAE	CV MAE (std)
4,9	0.6850	0.02156
1,2,3,4,5,6,7,8,9,10,11	0.5688	0.01136
2,10,12	0.6384	0.026924
1,2,3,4,8,9,11,12	0.5785	0.008247

Figure 3. Linear Regression results for different feature combinations where most significant results are displayed (feature descriptions in appendix)

$\sum_{n=1}^{N}(w^T \cdot x_n - y_n)^2 + \alpha \cdot f(w)$ where α is the hyper-parameter needed to tune.

model	Parameters	CV MAE	CV MAE (std)
Linear	None	0.569227	0.017819
Ridge	$\alpha = 10$	0.569235	0.017880
LASSO	$\alpha = 0.0004$	0.569315	0.017880
Elastic Net	$\alpha = 0.0004$ and $l_1 ratio = 0.15$	0.569238	0.017837

Figure 4. Linear Regression Cross Validation Results for different regularisation models

2.2.1 Ridge Regression

[4] An issue with this dataset in particular is col-linearity among features. For example, fixed acidity intuitively has a relationship with citric acid. Ridge Regression [3] tackles this by reducing components of small variance and allowing components with large variance to contribute to the output (similar to principal component reduction). Ridge regression implements the same loss function as regular regression, but subject to $\sum_{i=0}^{N} |w_i|^2 \leq C$. This leads to a similar equivalent loss function, with an extra term to reduce to weight size. The solution is convex and closed form, so quick to compute. It is seen that as the ridge constraint is relaxed, cross validation error decreases.

2.2.2 LASSO Regression

[5] LASSO regression [7] performs a similar function to ridge in terms of penalising large coefficient subject to $\sum_{i=0}^{N} |w_i| \leq C$, however by only taking the absolute and not squaring, this leads to sparser solutions, as coefficients commonly tend to zero. This can be seen in the weights diagram, as the coefficients drop off more rapidly. In terms of computing, LASSO is not closed form, and so requires an iterative training method. SK Learn implements gradient descent to find coefficients. LASSO also decreases in error as constraints are relaxed.

2.2.3 Elastic Net Regression

[6] Elastic Net[4] combines both the ridge and LASSO penalties as a ratio of the two. Elastic Net shares properties of both, with the amount being controlled by the l_1 ratio. This has potential to be more powerful than LASSO or Ridge on their own, however comes with the additional overhead of having to tune two parameters. the l_1 ratio has been set to 0.15 in order to give more importance to the l_2 normalisation term.

2.3. Regularisation Comments

It can be seen after evaluation, that regularisation does not contribute to reducing expected test error in the system

[4] penalty function, $f(w) = ||w||^2$
[5] penalty function, $f(w) = ||w||$
[6] penalty function, $f(w) = (l_1||w|| + (1 - l_1)||x||^2)$

model	Error Type	CV Error	CV Error (std)
Linear	MAE	0.569227	0.017819
Linear (classified)	MAE	0.513386	0.035743
Linear (classified)	Identity Error	0.465841	0.026760

Figure 5. Linear Regression Cross Validation Results for different errors and classifications

[7]. Lower errors are achieved at lower alpha values, which correspond to very wide constraints on coefficients. For all Regularisation models, the weights converge to the same value as α decreases, since the $\alpha \cdot f(w)$ term disappears. This is possibly due to the size of the dataset, as it is much greater than VC dimension, so achieves generalisation with the baseline features (possibly no features of significantly low variance).

2.4. Quantise Result

Since output classes are defined as integer values, it is possible to round results from regression in order to get a discrete set of outputs. the output value of regression y, is rounded to the nearest integer $y_{class} = [y]$[8], giving a discrete classification for the regression output. This also means that Identity Error can be used to evaluate error, giving a smaller test error. Please note that Identity Error and Absolute Mean Error are not used for training as they don't give a large penalty on training errors [9].

3. Classification

The challenge is next approached as a classification problem where each quality threshold is taken as a class. This threshold classification is defined as

$$y_{n_{classification}} = \begin{cases} 1, & \text{if } y_{quality} > n \\ -1, & \text{otherwise} \end{cases}$$

There are 11 classes for qualities 0 to 10 and so 10 threshold classes [10]. This classification comes intuitively from the idea that a wine quality of 5 will be greater than a wine quality of 4 and 3 and so on, in the sense that $y_{n-1} \subseteq y_n$. Through this method, the binary classifier will have more even split of examples for each threshold as opposed to the common one-vs-all method for multiclass classification[6]. In order to classify each threshold, a support vector machine (SVM) [1] is used to determine a separation boundary between the two classes. The SVM will learn regions of greater or less than quality rather than a region for just the quality itself. It also tries to maximise the margin between regions, defined as $\frac{1}{|w|}$, owing to better generalisation properties. There is a cost associated with points which violate this margin, as defined by the C term in the optimisation problem. For a larger C penalty, the stricter the margin is, with very few violations, especially for the case of linearly separable data. For better generalisation, non-infinite values will be used for C to allow for a wider margin. To predict the output class, the output of each SVM needs to be considered, and a single multiclass decision made. These threshold classes are searched to see where there is a change in threshold (-1 to 1). This suggests that the single output class is the value above this threshold. This predicter makes assumptions that all values below are homogenous as well as values above. This method only looks at a few outputs of the SVM stage, reducing the cumulative error of predicting on all outputs, though making it more venerable to errors form single SVM outputs.

$$pred(y,n) = \begin{cases} n, & \text{if } y_n = 1 \text{ and } y_{n+1} = -1 \\ pred(y,n+1), & \text{if } y_n = -1 \text{ and } y_{n+1} = 1 \\ pred(y,n+1), & \text{if } y_n = 1 \text{ and } y_{n+1} = 1 \\ pred(y,n-1), & \text{if } y_n = -1 \text{ and } y_{n+1} = -1 \end{cases}$$

All hyper-parameters are plotted with 10-fold cross validation, with the graphs used to tune them.

3.1. Linear SVM Classification

The first SVM classifier defines linear boundaries. It is in the same dimensional space as the linear regression models, however there is now a margin associated with the plane. The Linear SVM model has a small C term, which means it has a soft margin and many margin violations. Larger C terms cannot be learned because of the non-separability. To achieve stricter margins and reduce under-fitting, kernel methods will need to be explored to transpose the features to a higher dimension, making it more separable.

3.2. Kernel Methods

Kernel Methods can introduce a non-linearity and imply a transform of features. For Sigmoid and RBF kernels, this transform takes the data to a higher dimension. This higher dimensionality doesn't necessarily lead to overfitting, as is observed from the cross validation errors. It does lead to more sparse and seperable data.

3.2.1 RBF Kernel

The Radial Basis Function (RBF) kernel is defined as $K(x, x^{'}) = \exp(-\gamma||x - x^{'}||^2)$. It can be seen that the kernel falls off with a large difference between x and $x^{'}$, so for x which are far away from the support vector, they have very small effect on the output. A smaller γ leads to

[7] cross validation gives an estimate of the expected test error as they both have very similar expectation

[8] rounding is to ± 0.5 of raw output

[9] CV MAE = cross validation mean absolute error, IE = Identity Error

[10] For some qualities there is no example data, so thresholds are assumed to be constant since no learning can be performed on example-less data

Kernel	α	C	CV IE	CV IE (std)
Linear	-	0.01	0.466804	0.015853
RBF	2	1.0	0.375791	0.012667
SIG	0.01	1.0	0.474880	0.015928

Figure 6. SVM Cross Validation Results for different Kernels

a smoother function, and more smooth regions around the support vectors. The RBF kernel has the best characteristics, with least cross validation error of all models. The RBF function performs the best out of all SVM classifiers.

3.2.2 Sigmoid Kernel

The Sigmoid kernel is defined as $K(x, x^{'}) = tanh(\alpha x^{T} x^{'} + c)$. It has been explored as well as a classifier due to it's popularity. It is able to have stricter margins than Linear SVM, although it doesn't perform as well as an RBF SVM.

3.3. Kernel Method Comments

RBF by far out-performs the other two SVM classifiers for this classification problem. It's generalisation properties get around issues with non-seperablity which the Linear SVM classifier suffers from. If the data was more separable, it is suspected that Linear SVM would perform similarly to RBF. The RBF SVM classifier could be optimised further through gridsearch algorithms as well as other automated optimisation.

4. Conclusion

To conclude, the classification have proved a better option for predicting wine quality through wine features. Due to the discrete values for wine quality, it is possible to separate them into different regions. This method proves more effective than fitting the features to a continuous target function, possibly due to the noisiness of the data. SVM classification using an RBF kernel achieves the lowest test error and is considered the best classifier in this report.

5. Pledge

I, Alexander Montgomerie-Corcoran, pledge that this assignment is completely my own work, and that I did not take, borrow or steal work from any other person, and that I did not allow any other person to use, have, borrow or steal portions of my work. I understand that if I violate this honesty pledge, I am subject to disciplinary action pursuant to the appropriate sections of Imperial College London.

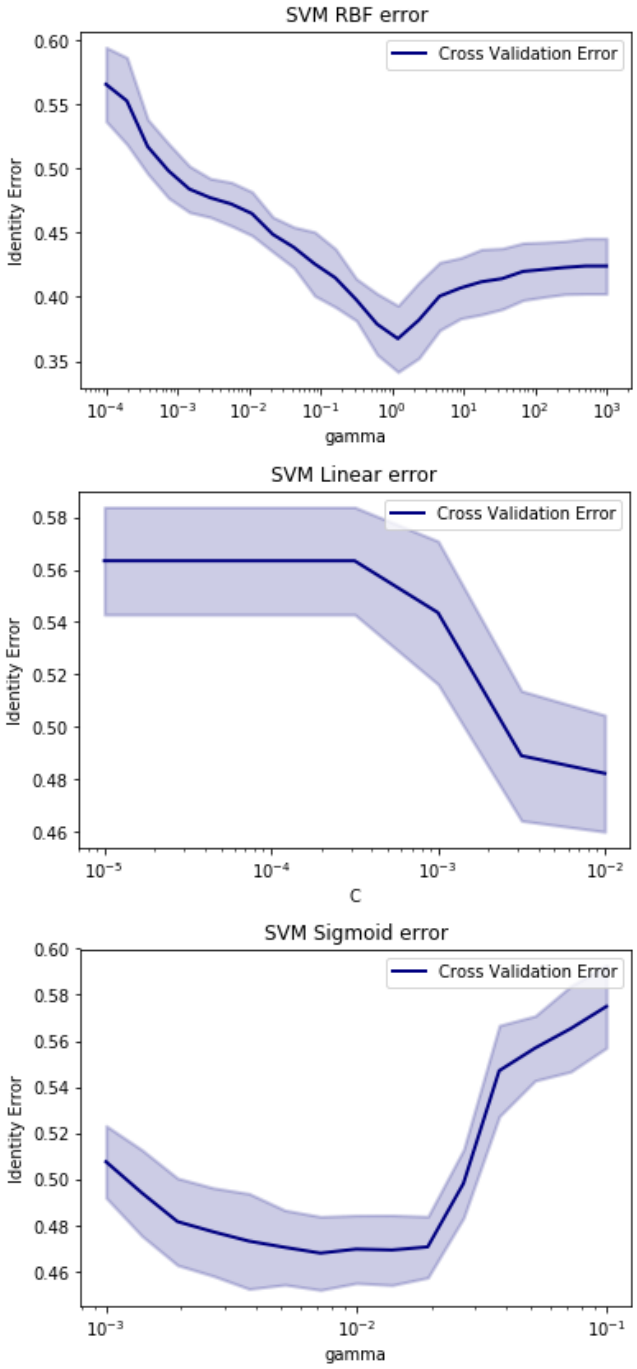

Figure 7. SVM error curves

Model	Test IE
SVM RBF	0.36
Linear Regression	0.48

Figure 8. Final Test Error comparison for Classification and Regression methods

References

[1] Support vector machines, 1992; boser, guyon, vapnik. *SpringerReference*.

[2] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin. *Learning from data: a short course*. AMLbook.com, 2012.

[3] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 42(1):8086, 1970.

[4] C. D. Mol, E. D. Vito, and L. Rosasco. Elastic-net regularization in learning theory. *Journal of Complexity*, 25(2):201230, 2009.

[5] R. Penrose. A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society*, 51(3):406413, 1955.

[6] Rifkin, Ryan, and Aldebaro. In defense of one-vs-all classification, Jan 1970.

[7] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

A. Data Analysis

Although analysing training data is considered a form of data snooping, it is worthwhile examining the data in order to get an impression of what the models mean. The data shows a large amount of covariance between features and only a few have strong correlation with the output. Interesting things to point out from the data is that:

- There is no wine quality classification above a value of 8
- No output below 2
- large dataset (1600 data points per feature)

A.1. Correlation with the Output

To understand how which features show the strongest linear relationship with the output classification, the correlation between each feature and the output has been evaluated. From the graph, it can be seen that alcohol has the strongest positive correlation whilst volatile acidity has the strongest negative correlation, suggesting that these features will have the strongest weights when using a linear classifier. I've discussed my work with Martin Ferianc and Rajan Patel.

B. Features

1. fixed acidity
2. volatile acidity
3. citric acid
4. residual sugar
5. chlorides
6. free sulfur dioxide
7. total sulfur dioxide

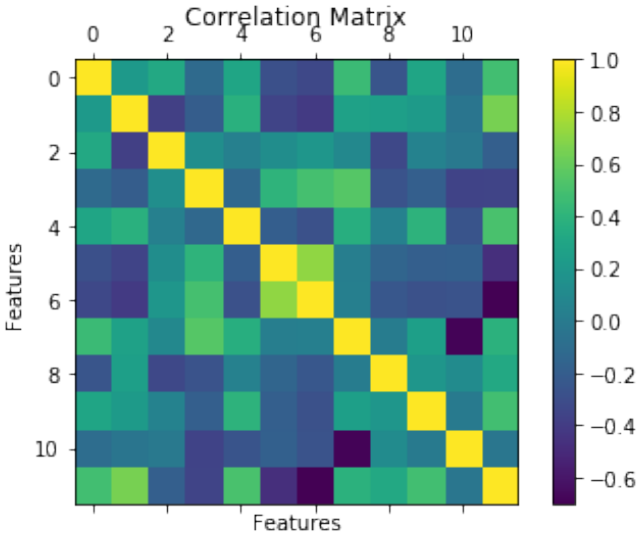

Figure 9. Correlation Matrix for Features

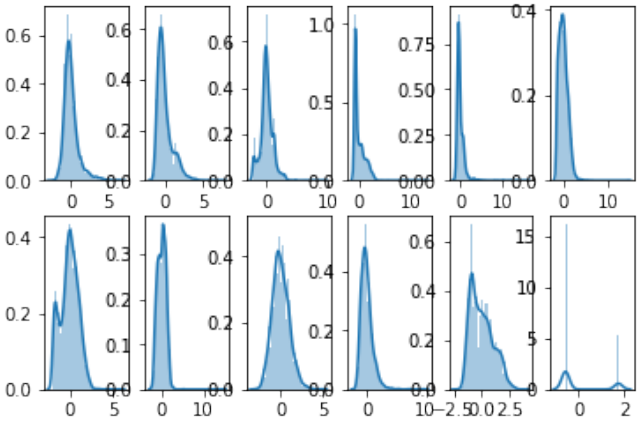

Figure 10. Distribution of features

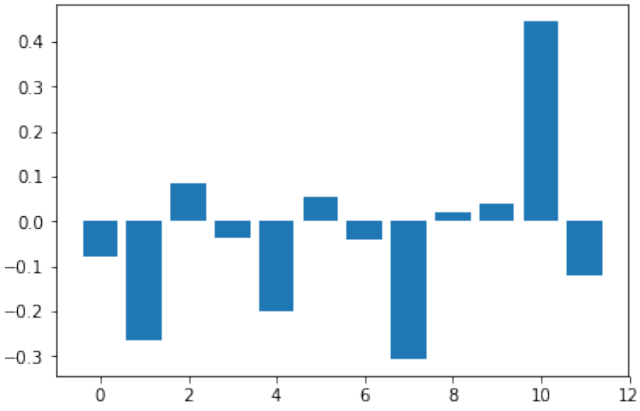

Figure 11. Correlation of output with features

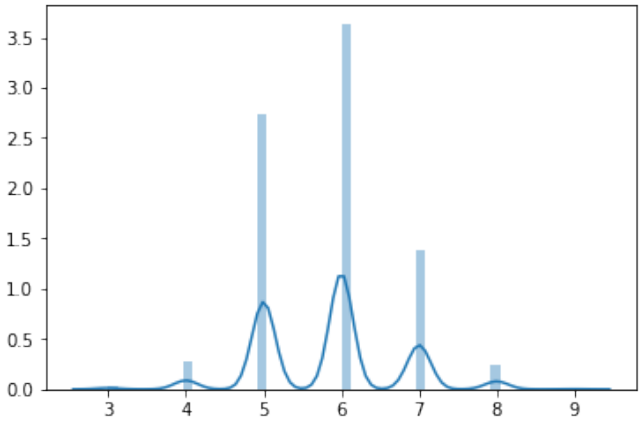

Figure 12. Output Quality distribution

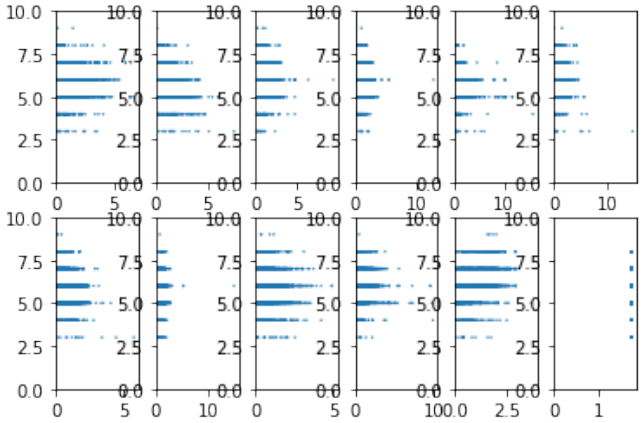

Figure 13. feature and output plot

8. density
9. pH
10. sulphates
11. alcohol
12. red or white