Report 2: Supervised Learning: Classification and Regression

Authors:
Albert Seligmann - s163907

Eli Carlin-Coleman - s186354

Paulius Patalavicius - s186402

Table of Contents

Introduction (Albert)

This report serves as a continuation of Report 1: Feature extract and visualisation, and seeks to examine some of the methods used for supervised learning. Additionally each method is optimized using cross-validation and finally the different methods are compared, both against each other as well as against a trivial baseline model.

The dataset which was utilized for Report 1, is used again for the following sections, but the dataset is modified slightly to better suit the aim of fitting regression and classification models. These modifications or transformations are described in their relevant sections, and pertain only to the sections in which they are described, unless otherwise stated.

Regression (Albert and Paulius)

We start off by examining the simplest model - linear regression. The goal of the linear regression will be to predict the ozone level by utilizing the information given by all other features.

Before fitting the linear regression model to our dataset, a couple of feature and data transformations are carried out to ensure a proper fit. In the previous report it was determined that the feature “ibh” was a candidate for possible removal, as it contained odd values and correlates highly with the “ibt” feature, thus it was decided to remove this feature for this project. Additionally the “doy” is replaced with a “Season” feature, which in turn is one-out-of-K encoded as 4 new features. Finally an offset / bias term is added to the data-matrix. Before the data is used in any of the following models, it is normalized by subtracting the mean and dividing by the standard deviation, both of which are exclusively computed from the training data. The normalization is done in the cross-validation folds, as to avoid polluting “unseen” test data.

The transformations are summarized below:

- Replacing “doy” with “Season”, a feature of discrete nominal type.
- One-out-of-K encoding the “Season” feature, resulting in 4 new features: "Autumn", "Spring", "Summer" and "Winter".
- Removal of the “ibh” feature.
- Offset term is added.
- Normalization of the data.

Regularization

Next a least-squares regularization parameter λ is introduced to control model complexity. In brief, the introduction of the regularization parameter leads to control of the solution bias and variance by affecting the scale of weights. The effect of the regularization is explained in detail in the course material.
In order to choose an appropriate value for λ, a cross validation of varying values of λ is carried out, and the generalization error is computed. Based on the generalization error an optimal value for λ can be chosen. 25 values of λ were tested, all of which were powers of 10 centered around 0.
Our data indicates that a value of around $\lambda = 10$ produces the best generalization error, however this changes slightly when introducing more parameters, as discussed in the following section. Additionally from Figure 1, it is very apparent that this method of regularization has almost no distinctly optimal value for our simple linear regression model, as long as λ is kept relatively low, and that the model does not overfit.

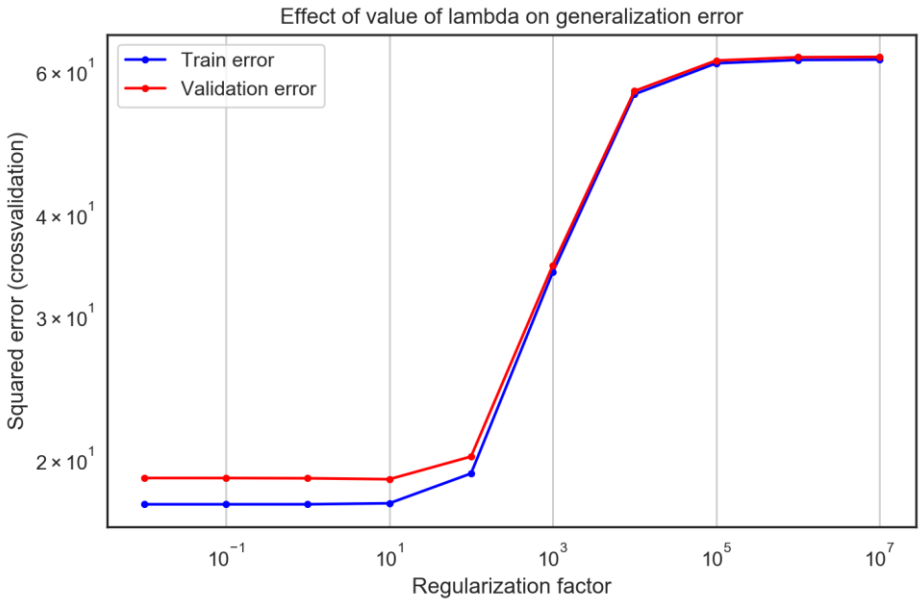

Figure 1 - Effect of lambda on generalization error

Several additional values of λ were examined, but only a few are included in the above plot for visualisation reasons.

Since we are using all features in the dataset ("vh", "wind", "humidity", "temp", "dpg", "ibt", "vis", "Autumn", "Spring", "Summer" and "Winter") our fitted regularized linear regression model predicts the ozone level based on 11 features and an offset. Each feature contributes with different proportion based on their respective weight defined by the interplay of model and regularization. Figure 2 shows that the “humidity”, “temp” and “ibt” features have a strong positive effect on the output of the linear regression and “dpg” and “Autumn” have a small negative influence, while the rest of the attributes have little to no influence on the output.

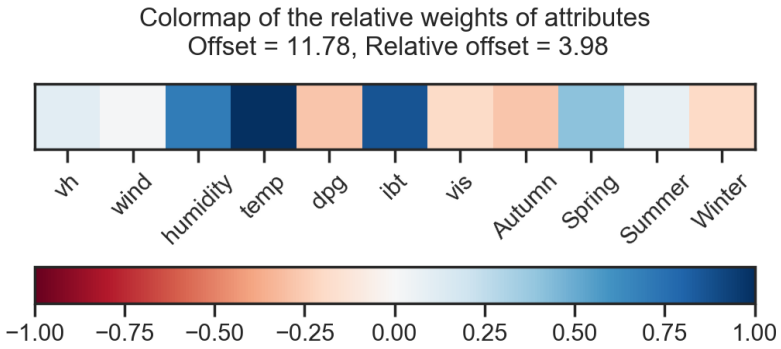

Figure 2 - Relative weights of each attribute/feature

When evaluating the regularized linear regression model against the ground truth values, the model predicts the ozone level with some accuracy. The model is, however, not excellent at predicting the actual value, but it overall trends well with the actual true value. See Table 1 for generalization test error.

Artificial Neural Network

The second model of interest to be examined is the Artificial Neural Network otherwise known as ANN. A relatively simple ANN is used, which only contains one hidden layer with h hidden units. The hidden layer maps to a single output neuron through a hyperbolic tangent ($tanh$) transfer function.

Cross-validation and statistical evaluation

For regression we wish to predict ozone value based on all other parameters we are using. We use the same configuration as we used in part a of regression. For this section we compared three models: the regularized linear regression, an artificial neural network and a baseline. We wanted to find out which model works the best. We used two level cross validation with K1 = 5 and K2 = 5 as well. In inner loop of our cross validation as complexity controlling parameter for ANN we used number of hidden units. We started out by testing the number of hidden units in the ANN in the range from 1 to 5, and the optimal value seemed to be 5. Since 5 was on one end of the interval of tested values, we re-tested the optimal numbers of hidden units in range from 3 to 9, and the optimal number of hidden units remained turned out to be a mix of 6 or 7, as can be seen from Table 1. As for regularized linear regression we regularized values of lambda. Lambda values we tried includes: 0.1, 1, 10, 100 and 1000. On real project we would firstly find the region of lambda and after that find the best value. Due to limited computational power of us, we decided to work with just those 5 lambda values. We saved parameters of methods, which had the lowest error for each fold of inner loop of cross validation. We took the mean of best lambdas and used it to make the best prediction in outer cross validation loop. For ANN`s number of hidden units we used mode (since its value should be integer).

After computing the best parameter in the inner loop of the two-level cross-validation, we train our models using these parameters. For testing the models we reuse the train/test splits from outer cross-validation loop to facilitate statistical comparison. We evaluate the performance of the models use Mean Square Error, defined by:

$$E = \frac{1}{N^{\text{test}}} \sum_{i=1}^{N^{\text{test}}} (y_i - \hat{y}_i)^2$$

The results generated from testing our regression models are presented in Table 1 below. The first column in the table shows the index of the outer cross-validation fold.
The second column is contains the results pertaining to the ANN. In our case the optimal number of hidden units in the hidden layer is 7, and the test error varies between 47.67 and 81.89, depending on which split is analyzed.
The third column contains the results generated when testing the regularized linear regression model. Here it can be seen that lambda is drastically different to what was found earlier, as the optimal lambda value is frequently an order of magnitude larger than what was computed earlier. In our case it is difficult to pinpoint which value of lambda should be considered optimal, as there is a large variance in the optimal lambda depending on the cross-validation split. We decided to simply consider the mean of optimal lambdas as the optimal lambda.
The last column contains the test error corresponding to the baseline model, which simply predicts the ozone level as the mean of the training splits ozone level.
It is immediately obvious that the regularized linear regression model performs much better than the other two models. It produces a test error which is almost 3 times smaller than both the ANN and the baseline when averaging across cross-validation folds. Furthermore it can be seen that the ANN and the baseline models perform almost exactly equally.

Table 1 - Generalization error and optimal parameters by cross-validation fold

Outer fold	ANN		Linear regression		Baseline
k	**h**	**Test error**	**Lambda**	**Test error**	**Test error**
1	6	47.67	404.02	21.57	47.33
2	7	59.56	402.22	18.14	59.33
3	6	76.11	242.02	27.01	76.14
4	7	81.89	42.22	23.92	79.27
5	7	55.90	440.2	20.75	55.70
Mean/mode	**7**	**64.23**	**306.15**	**22.28**	**63.55**

To statistically determine if there is a significant performance difference between our three models, we utilize the inverse of the cumulative distribution function to compute a confidence interval.

The upper z_U and lower z_L values of the confidence interval is computed using

$$z_L = \text{cdf}_{st}^{-1}(\frac{\alpha}{2}|\nu, \bar{z}, \tilde{\sigma}),\ z_U = \text{cdf}_{st}^{-1}(1 - \frac{\alpha}{2}|\nu, \bar{z}, \tilde{\sigma})$$

where the $\underline{z}$, v and σ are defined as

$$\bar{z} = \frac{1}{K}\sum_{k=1}^{K} z_k,\ \nu = K - 1 \text{ and } \tilde{\sigma} = \sqrt{\sum_{k=1}^{K} \frac{(z_k - \bar{z})^2}{K(K-1)}}$$

Here z_k corresponds to the testing error in each cross validation fold k, and K is total the number of cross validation folds. These values can then be used to compute the confidence interval. In Python this is computed using the ppf function found in the scipy library.

$$\text{cdf}_{st}^{-1}(A|\nu, \bar{z}, \tilde{\sigma}) = \bar{z} + \tilde{\sigma}\,\texttt{t.ppf}(A, \nu)$$

Here A represents the tail interval, and is chosen to be $0.05/2 = 0.025$, which corresponds to a confidence interval of 95%. In our case we have $K = 5$ and $v = 4$, since we've used a 5-fold cross-validation for all of the models.

First we compare the ANN model against the regularized linear regression model:

The mean of z is computed as:

$$\underline{z} = ((47.67 - 21.57) + (59.56 - 18.14) + (76.11 - 27.01) + (81.89 - 23.92) + (55.90 - 20.75))\,/\,5 = 41.95.$$

Sigma is calculated using the formula mentioned earlier, and is computed as $\sigma = 5.50$. These values can now be used to compute the inverse of the cumulative distribution function, also known as the Percent Point Function, which gives us the a confidence interval define by the lower and upper values of

$$[z_L, z_U] = [26.67, 57.22].$$

The interval is centered far above 0, so we can confidently say, that our ANN model has higher generalization error than our regularized linear regression model.

Next the ANN model is compared against the baseline model:

We compute sigma and the mean of z:

$$\sigma = 0.49,\ \underline{z} = 0.67.$$

We now compute the confidence interval:

$$[z_L, z_U] = [-0.69, 2.03].$$

As this interval reflects the difference in generalization error, we cannot conclude that the difference is not 0 at $A = 0.025$, which indicates that the model performance is roughly equal.

Finally, we compare the regularized linear regression model against the baseline model:

We compute sigma and the mean of z:

$$\sigma = 5.20,\ \underline{z} = -41.28.$$

We now compute the confidence interval:

$$[z_L, z_U] = [-55.71, -26.84].$$

The interval lies well below 0, so we can conclude that our regularized linear regression model has lower generalization error than the baseline method.

All in all, our regularized linear regression model provides far better results for our use case than our ANN or baseline models. Baseline and ANN methods provide very similar generalization error, and are roughly equal in performance.

Classification (Eli)

Our dataset did not originally contain any nominal data, which would have made classification impossible. To meet the requirements of this project, we decided to use the interval attribute 'doy' (day of year) to generate a nominal attribute, 'season'. This process is described in Report 1. As this was the only nominal attribute in our dataset, we decided to predict 'season' from the other attributes. As 'season' can take four different values, Winter, Spring, Summer, and Autumn, this is a multiclass problem.

Similar to our regression analysis, we determined that the feature 'ibh' contained low quality data, and decided not to use this in our analysis. We considered performing feature selection, but determined that this would be out of scope of this report, and so all other features were used in classifying our data.

We used three methods to model our data. These methods were: a baseline, logistic regression, and an artificial neural network (ANN). In both the logistic regression and the ANN we used a complexity controlling parameter, as well as two layer 10-fold cross-validation. For the baseline model, we used one layer K-fold cross-validation. We then compared these models using an error metric. For this, we used the mean number of misclassified observations, calculated as $E = \frac{\{Number\ of\ misclassified\ observations\}}{N^{test}}$. Error is only considered for the test dataset, and not the training dataset, unless otherwise specified.

Baseline

Our first model was a baseline, needed to compute the effectiveness of other models. We chose our baseline to be the mode of the test dataset. Because our dataset did not contain a multiple of 4 rows, some classes occur slightly more often than others. Additionally, our K-fold split involves a random shuffle, so there is the possibility for one class to be overrepresented in a given fold. Therefore, we would expect the error rate to be slightly above 0.75.

Our results show an average error 0.79, and a minimum error of 0.71. This average error is similar to our expectations.

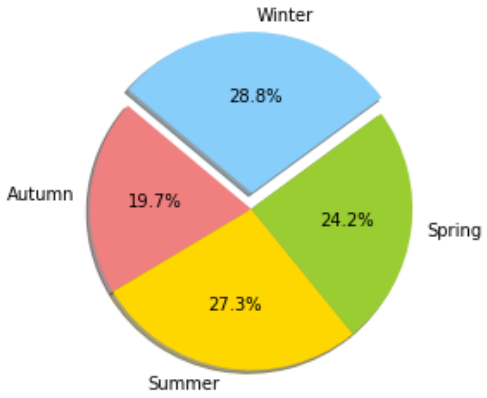

Figure 3 - Class distribution in test dataset

In this case, Winter was the most common class in both the test and the train datasets, and so the baseline model predicted 28.8% of the data correctly

Logistic Regression

Another model we used was a logistic regression. Here we used two layer 5-fold cross-validation, and we control the complexity of the model using a lambda value. We search through ten lambda values for each outer fold, with values of $\lambda = 10^p, -2 \leq p < 3$. We expect that this model will perform significantly better than the baseline model, although it is difficult to say exactly how well. There are several variables which are quite distinct between the summer and winter classes, but autumn and spring have very similar characteristics, both to each other and to the other two classes. We expect that summer and winter will be modeled quite accurately, whilst spring and autumn will be correct approximately half of the time. Overall, we expect the error to be around 35%.

When running the algorithm, we found that test error was 24%. This is better than our predicted value. However, the distribution of errors was very different to what we expected. However, the distribution of errors did not match our hypothesis. After several trial runs, it was clear that the error was distributed randomly across each season, and that there could be quite large differences in the error rates for a given season within a trial. Typically, three of the error rates are consistent, but one season is inconsistent. We predict that this is because of random variations in our K-fold split, which may happen to include disproportionate amounts of data from each season in the training set.

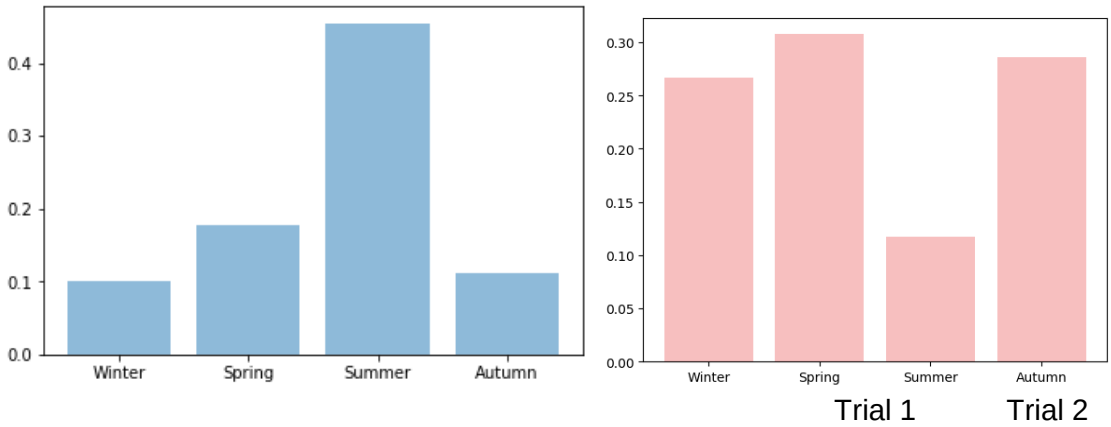

Figure 4 - Error rate calculated for each season (Logistic Regression)

Artificial Neural Network

The final model we used was an artificial neural network. We again used 5-fold cross-validation, with a complexity controlling parameter. We decided to alter the parameter 'hidden nodes' as we found that this could decrease our error by up to a factor of 2. Within the inner K-fold, we create a model using hidden nodes in the range of 1 to 3, and compute which model is most accurate, and identify how many hidden nodes were present. In the outer K-fold, we use this test to inform how many hidden nodes are used in the model. Conditions are similar to the logistic regression situation, so we expect as similar result.

After running the model, we found that the error was 29%. This is better than our predicted value, and is slightly worse than the logistic regression result. We also found that the distribution of errors across seasons was random and inconsistent between trials, and we expect random variation in the K-fold split to be responsible for this.

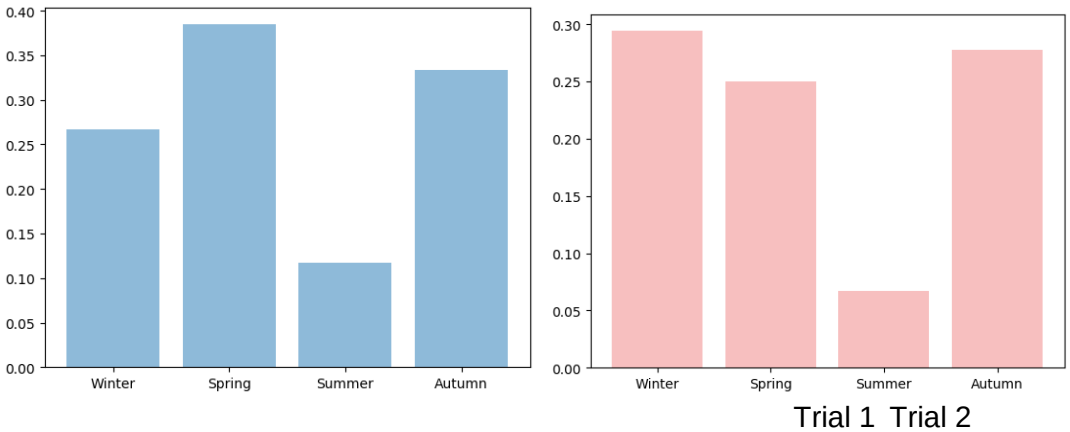

Figure 5 - Error rate calculated for each season (ANN)

Summary of Results

Table 2 - Generalization error and optimal parameters by cross-validation fold

Outer fold	ANN		Logistic regression		Baseline
k	h	Test error	Lambda	Test error	Test error

1	3	22.73	0.01	22.73	77.27
2	3	28.79	0.01	24.24	72.73
3	3	30.30	0.01	25.76	72.73
4	3	30.30	0.01	21.21	75.76
5	2	34.85	0.01	25.76	81.82
Mean/mode	**3**	**29.39**	**0.01**	**23.94**	**76.06**

There are slight differences between ANNs and logistic regression, and both are very distinct from the baseline. ANNs appear to have more variance than logistic regression, and errors can vary by more than 10 percentage points, compared to the 3 percentage points for regression. Both models clearly have a preferred complexity, with ANNs performing optimally at 3 hidden nodes, and logistic regression producing the best results at $\lambda = 0.01$.

Statistical Evaluation

With the same strategy as in part 1 of this report, we use the inverse of the cumulative distribution function to find the confidence interval.

The upper and lower values, z_U and z_L, of the confidence interval are found with the equations:

$$z_L = \mathrm{cdf}_{st}^{-1}(\frac{\alpha}{2}|\nu, \bar{z}, \tilde{\sigma}),\ z_U = \mathrm{cdf}_{st}^{-1}(1 - \frac{\alpha}{2}|\nu, \bar{z}, \tilde{\sigma})$$

where the $\underline{z}$, v and σ are given by:

$$\bar{z} = \frac{1}{K}\sum_{k=1}^{K} z_k,\ \nu = K - 1 \text{ and } \tilde{\sigma} = \sqrt{\sum_{k=1}^{K} \frac{(z_k - \bar{z})^2}{K(K-1)}}$$

With z_k representing test error in each fold k, and Kbeing the total number of folds.

Here z_k corresponds to the testing error in each cross validation fold k, and K is total the number of cross validation folds. The Python function ppf is used to compute the confidence interval from these values

$$\mathrm{cdf}_{st}^{-1}(A|\nu, \bar{z}, \tilde{\sigma}) = \bar{z} + \tilde{\sigma}\,\mathtt{t.ppf}(A, \nu)$$

Here A represents the tail interval, and is chosen to be $0.05/2 = 0.025$, which corresponds to a confidence interval of 95%. In our case we have $K = 5$ and $v = 4$, since we've used a 5-fold cross-validation for all of the models.

ANN vs Logistic Regression

The mean of z is computed as:

$$\underline{z} = ((22.73 - 22.73) + (28.79 - 24.24) + (30.30 - 25.76) + (30.30 - 21.21) + (34.85 - 25.76))\,/\,5 = 5.45.$$

Sigma is calculated as:

$$\sigma = \sqrt{\frac{((22.73 - 22.73) - 24.60)^2 + ((28.79 - 24.24) - 24.60)^2 + ((30.30 - 25.76) - 24.60)^2 + ((30.30 - 21.21) - 24.60)^2 + ((34.85 - 25.76) - 24.60)^2}{5(5-1)}} = 1.70$$

These values can now be used to compute the inverse of the cumulative distribution function, also known as the Percent Point Function, which gives us the a confidence interval define by the lower and upper values of

$$[z_L, z_U] = [0.72, 10.17].$$

The interval is centered slightly above 0, so it is likely that the logistic regression model has a lower generalization error than ANN model. However, as the lower bound is quite close to 0, we cannot be entirely confident of this.

Baseline vs ANN

The mean of σ, z are computed as:

$$\sigma = 2.11, \underline{z} = -46.67.$$

We now compute the confidence interval:

$$[z_L, z_U] = [-52.52, -40.83].$$

The interval is centered far below 0, so we can be sure that the ANN performs better than the baseline model.

Baseline vs Logistic Regression

The mean of σ, z are computed as:

$$\sigma = 1.83, \underline{z} = -52.12.$$

We now compute the confidence interval:

$$[z_L, z_U] = [-55.20, -47.04]$$

The interval is centered far below 0, so we can be sure that the logistic regression performs better than the baseline model.

The best performing model is the logistic regression model. This is slightly more consistent, and slightly better than the ANN model. Both of these models far outperform the baseline model.

Discussion (Albert and Paulius)

This project spans a large number of topics pertaining to analysis of regression and classification problems, as well as the methods used in solving these problems. Furthermore, this project gave us the opportunity to examine the effects of different data transformations as well as proper validation and evaluation of data.

Due to the large amount of methods that exists in this area of machine learning, a few were selected to be implemented for use with our dataset. Specifically Artificial Neural Networks, Linear Regression, Regularized Linear Regression with least-squares regularization and Logistic Regression were examined and evaluated against a baseline measure. All of these models were evaluated with different model-parameters and regularization-parameters, using K-Fold cross validation, to ensure that a close to optimal model is selected.
Several models were examined for the same tasks, which enables us to perform a statistical performance evaluation of each model and thus compare the models against each other. This was done using credibility intervals. Our results in part 1 show that an ANN has sub-par performance on our dataset compared to other methods, while ANNs typically perform well compared to other models. One reason for this being the case, could be for the fact that our dataset contains a relatively small number of observations (~330), which in turn is lowered further when performing cross-validation. This leads to the ANN being insufficiently trained. A lesser example of this is shown in part 2 of our report, as our ANN model performed similarly to logistic regression, but still failed to exceed its predictive power.

The resulting models in the above sections are all of reasonable quality when it comes to solving the problem of predicting either ozone level or season, however better models can be achieved given more observations and more computational power, which proved to be a fairly large hurdle in computing the optimal model parameters. Our results for classification show that both of our models were of good quality when it comes to classifying the season, and that they could likely only be improved marginally by increased computational power. Our results may also be improved with more sophisticated models, such as performing feature selection before fitting models, but we opted not to do this.

The LA Ozone dataset is frequently used in statistics and machine learning courses. Despite this, we were unable to find any relevant articles using the dataset for regression or classification, which also contained a description of methods used or results. Thus we are not able to compare our results against a third party.