Report 2

Supervised learning: Classification and regression

Mathias Husted Torp, s133547
Damian Kowalczyk, s166071

April 4th, 2017

1 Regression

The regression part of this report will focus on predicting how much horsepower a car has based on the other features in the data set. Various kinds of machine learning methods are available to perform regression, but the most simple is linear regression.

As a starting point, linear regression is performed on all the features to include as much data as possible. In order to avoid over-fitting it can be beneficial to apply sequential feature selection and thus improve the model performance.

The performance of the two models are tested through 4-fold cross-validation, and it turns out that the test classification error is nearly the same with and without forward selection. The results can be seen in Table 1. You might see this as the forward selection is a waste of time and resources, but you can also see this as you have identified attributes carrying the most information and reducing the data set. It is worth noting from the table that there without feature selection is some difference between the training and the test error, whereas the two move closer together when applying forward selection.

Table 1: The error rates when using linear regression to predict the horsepower of cars.

	Without feature selection	With feature selection
Training error	0.0177	0.0632
Test error	0.1175	0.1077

The horsepower of a new data point is predicted by multiplying each of the input attributes with the corresponding weight and a single output value is achieved, which is the predicted horsepower.

1.1 Applying an artificial neural network

Artificial neural networks are able to learn very complex connections and patterns in data. There is a risk that they may over-fit the data, and it is thus important to check if there is a large difference between the training error and the test error.

A neural network with 8 hidden neurons is fitted to the data, and its performance is evaluated using 4-fold cross-validation and 20-fold cross-validation, respectively. The results can be seen in Table 2.

Table 2: The error rates when using artificial neural networks to predict the horsepower of cars.

	4-fold cross-validation	20-fold cross-validation
Training error	0.00018	0.00428
Test error	5.74204	35.39973

1.2 Performance comparison

It is relevant to test if one of the fitted models perform significantly better than the other, and importantly, if they perform significantly better than just predicting the average value each time. A paired t-test is used to test this. In Table 3 it is seen that all the models perform significantly different, which means that linear regression performs significantly better than artificial neural networks that performs significantly better than just predicting the average of the training data set.

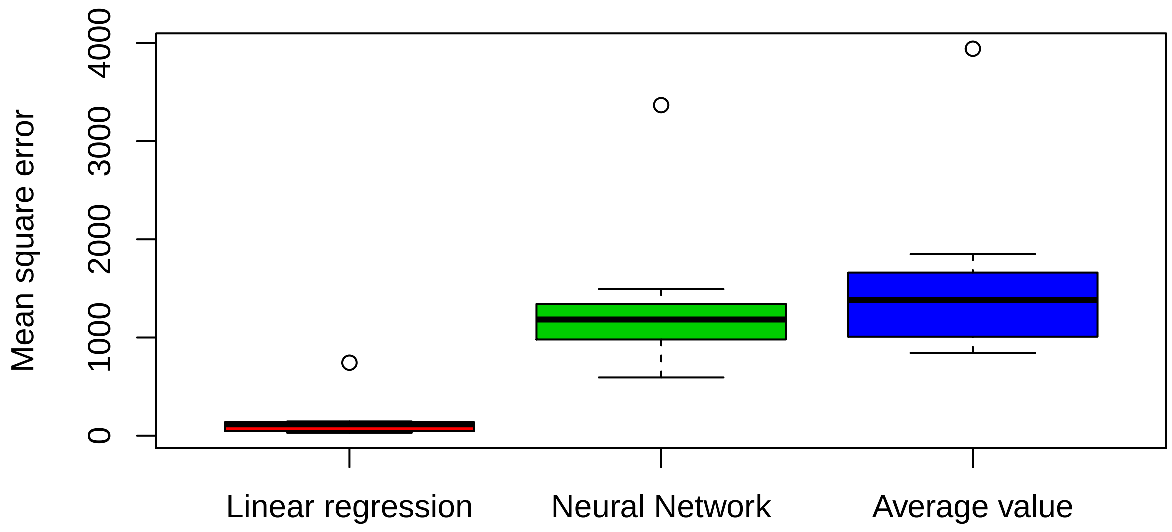

Figure 1: A box plot of the distribution of mean square errors in the 10-fold cross-validation.

Table 3: Paired t-tests results in the P-values in this table. They all test if the difference in average error is 0 or not.

	ANN	avg
avg	8.67e-03	NA
reg	1.17e-04	1.11e-04

2 Classification

This part of the report will be about classifying the cars into discrete groups. Specifically, it will focus on classifying the cars into four categories of the average gas consumption per mile driven, which is a new feature based on the ‘city-mpg’ feature. The feature is generated based on the following rules:

- Very low = city-mpg <= Q1
- Low = Q1 < city-mpg <= median
- High = median < city-mpg <= Q3
- Very high = Q3 < city-mpg

Where Q1 means the lower quartile and Q3 means the upper quartile. The number of observations in each category is shown in Table 4.

Table 4: The number of observations in each of the categories of the discrete city-mpg feature.

Very low	Low	High	Very high
55	49	52	49

When applying the machine learning methods, neither the discrete nor the continuous version of the ‘city-mpg’ feature will be in the data set, just as the feature ‘highway-mgp’ is excluded as well. In the hope of improving the performance of the machine learning methods, a subset of features, which are intuitively believed to influence the fuel economy, was selected by hand. This subset of the data set is in the following sections compared to the full data set.

2.1 Decision trees

Decision trees are an easily interpretable way of classifying data into an arbitrary number of categories. This makes them well-suited as a starting point for classifying this data set.

First, a standard decision tree model is fitted to the data. The model can be seen in Figure 2. This is a rather large tree, which is a cause of a lot of splits that may result in over-fitting. Pruning can be applied to avoid over-fitting. However, how much pruning is necessary to obtain the optimal model varies a lot from case to case. Thus, cross-validation is used to select the optimal pruning level for this case.

It is seen in Figure 3 that there is a local classification error minimum for a pruning level of around 0.1. However, the global minimum for the test classification error is for pruning levels above 0.35, where it stabilises below 75 %. It seems that the model fails to properly classify new data points, and the lowest classification error is thus achieved by always predicting ‘High’.

When a decision tree is to predict the class of a new data point, it starts at the top of the tree and at each split it checks if the answer to the question is true or false. When true, it goes left, and when false, it goes right, until the bottom of the tree is reached and the point is successfully classified.

2.2 Multinomial Regression

Multinomial regression is an expansion of logistic regression, which works with multiple categories. Since the data set in this report has been thresholded into four categories, this is the way to go.

In this report, instead of selecting parameters for the multinomial regression, parameters for the data are selected through cross-validation. This means that both normalised data and raw data are used in the model and that only some hand-chosen features are used versus all features. This makes K = 4 inner cross-validation loops.

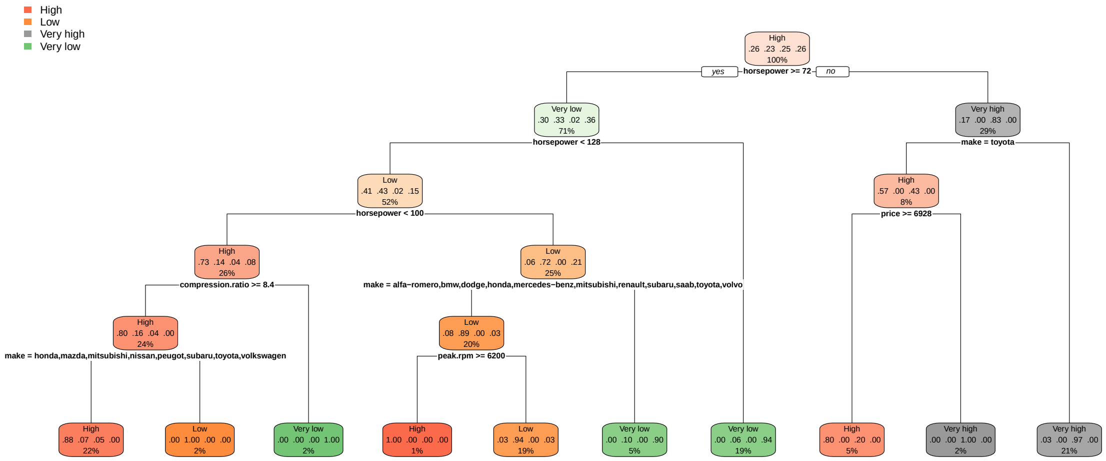

Figure 2: A non-pruned decision tree. Note that there are a lot of splits

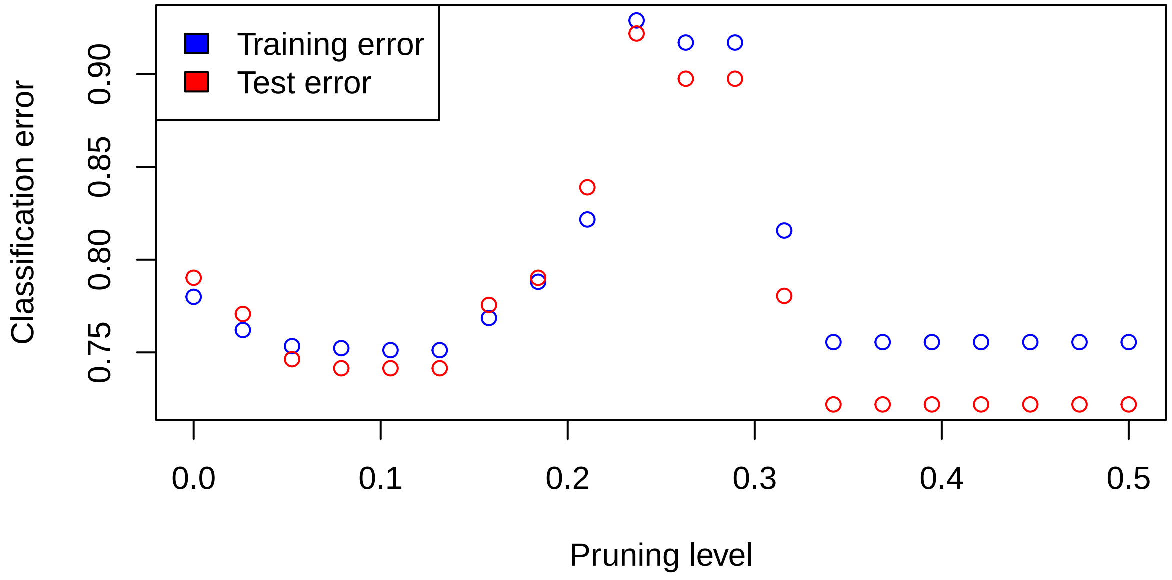

Figure 3: How pruning effects the classification error. Note that for high values of pruning, the training classification error stabilises around 75 %.

As it is seen in Table 5, the multinomial regression cannot properly classify the cars. It actually performs worse than the pruned decision tree that just predicts the same class for each new observation.

Since multinomial regression is the same as an artificial neural network with no hidden neurons and thus no hidden layer, a new data point is classified by taken the input values and multiplying them with the weight matrix W to an output value for each class and the class with the highest output value is the predicted class for the new observation.

Table 5: The error rates when using multinomial regression to classify the cars into four classes.

All features	All features normalised	Selected features	Selected features normalised
84.62 %	86.54 %	92.31 %	92.31 %

2.3 K-Nearest Neighbors (KNN)

The K-Nearest Neighbors algorithm works by simple classifying a new data point as the majority of the K nearest data points. This simple method proves to be sometimes be quite powerful.

It is very easy to visualise how this works in two dimensions, but it works in N-dimensions. This is why, it can easily be applied to this classification problem in this report, though it is multi-dimensional.

Cross-validation is used to find the best value of neighbors for this classification problem. In Figure 4 the classification error is seen as a function of the number of neighbors used for classifying new data points.

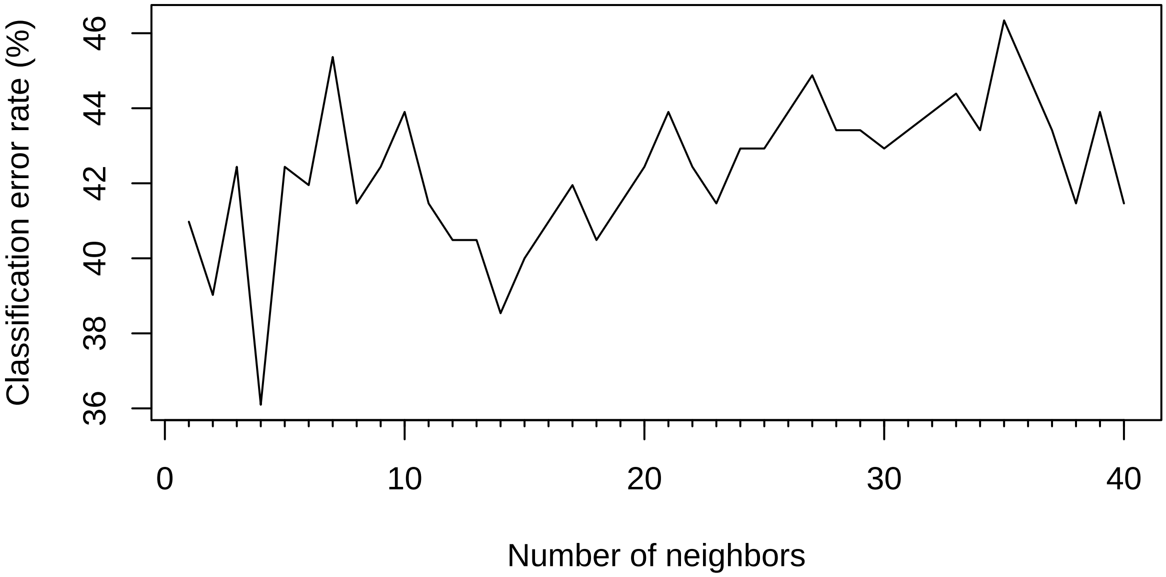

Figure 4: The classification error as a function of the number of hidden neurons.

The lowest achieved classification error is 36.10 %. The number of neighbors used to achieve this is K = 4. When using too few neighbors, the classification is prone to being too effected by noise and outliers in the data set, but when using too many neighbors, it is prone to being too conservative and predicting the class with the most observations.

When a new data point is to be classified, the K nearest neighbors are identified and their class is checked. The most frequent class of these points is the class that the new data point is given. If there is a tie, i.e. K is 4 and 2 points are class 1 and class 2, respectively, the class of the neighbor nearest to the new data point is used to break the tie.

2.4 Artificial Neural Networks (ANN)

Artificial neural networks are extremely flexible for modeling a wide variety of problems. This is among other things because they have the ability to learn which features in the given data set that are important and weight them accordingly.

Their flexibility, however, also makes them prone to over-fitting, and when training them there is a risk they get stuck in a local minimum instead of correctly landing in the global minimum.

One-out-of-K encoding has been applied to the data set in this report, and thus 68 features are given as input for the network. Furthermore, there are four groups that a data point can be classified into. This structure defines that the network model has 68 input neurons and 4 output neurons. There is one hidden layer and the optimal number of hidden neurons is chosen through cross-validation as a value between 1 and 40.

There is a chance that the network gets stuck in a local minimum, and to reduce the random error introduced by this, the outer cross-validation loop has 10 iterations. The classification error rate as a function of the number of hidden neurons is shown in Figure 5. It is seen that the classification error steadily decreases as the number of hidden neurons increase.

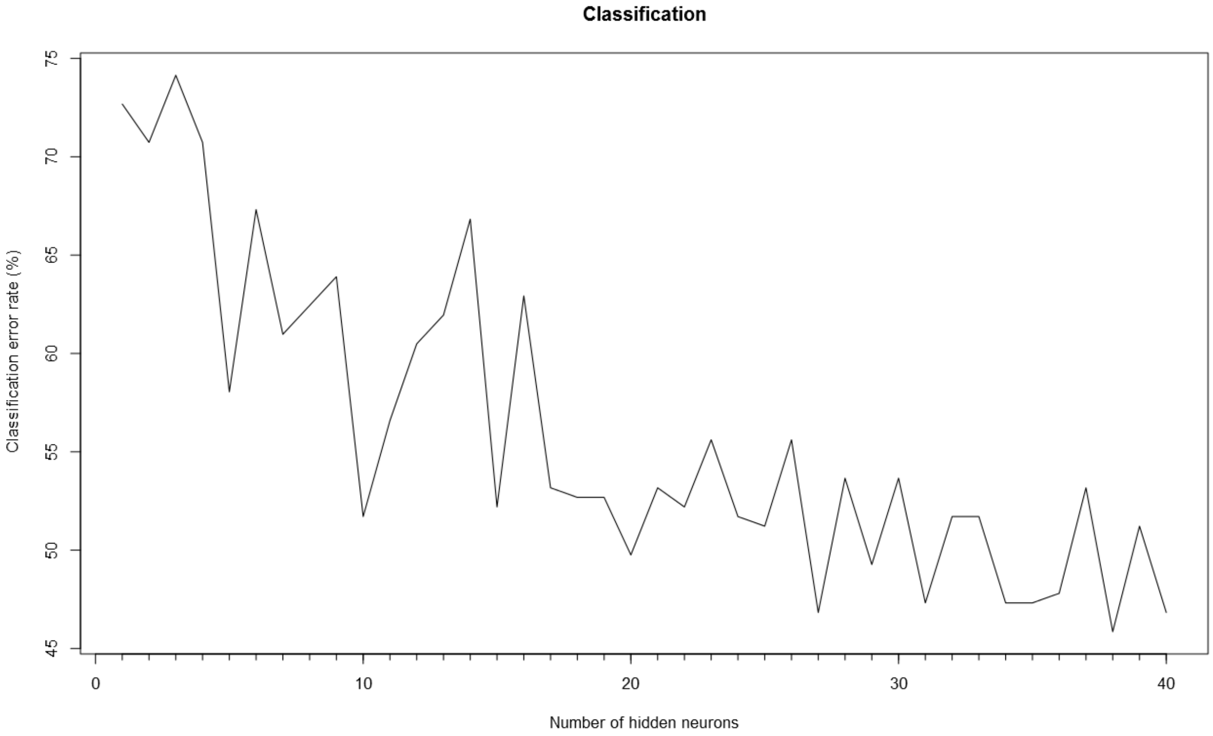

Figure 5: The classification error as a function of the number of hidden neurons.

In this analysis the lowest classification error is accomplished when using 38 hidden neurons, but it is seen in the figure that there is a lot of variation even though the number of hidden neurons is only changed by one. This suggests that a higher number of hidden neurons is necessary to achieve a lower classification error, but due to the time taken to perform these calculations, it has not been possible to train a network with a higher number of hidden neurons.

The artificial neural network only accepts numeric input, which is also why the one-out-of-K encoding is used. When a new data point is to be classified, the value of each of the features are given to their respective input neuron. The value from an input neuron $\mathrm{N_i}$ is multiplied with the weight $\mathrm{W_{ij}}$ to give the value of neuron $\mathrm{N_j}$ in the next layer of the network. This happens for all neurons in the network until the output layer is reached. The class of the data point is finally the output neuron with the highest value. Sometimes the input value of hidden neurons are transformed through an activation function $h(x)$.

2.5 Prediction performance comparison

The lowest achieved classification error for artificial neural networks is seen in Figure 5 to be slightly above 45 %. Since the classification error is steadily falling as the number of hidden neurons increase, let's assume that it is possible to achieve a classification error of 45 % if enough hidden neurons are used in the model. The lowest classification error of the K-nearest neighbor algorithm was 36.10 %. Since the lowest classification error for both multinomial regression and decision trees are a lot worse, the two best performing classification models are ANN and KNN.

Now it is relevant to ask, if KNN is significantly better the ANN, or if there is not enough statistical evidence to conclude anything. This is done by using the parameters that resulted in the best performance for each of the models and train and test these on the same data through 10-fold cross-validation.

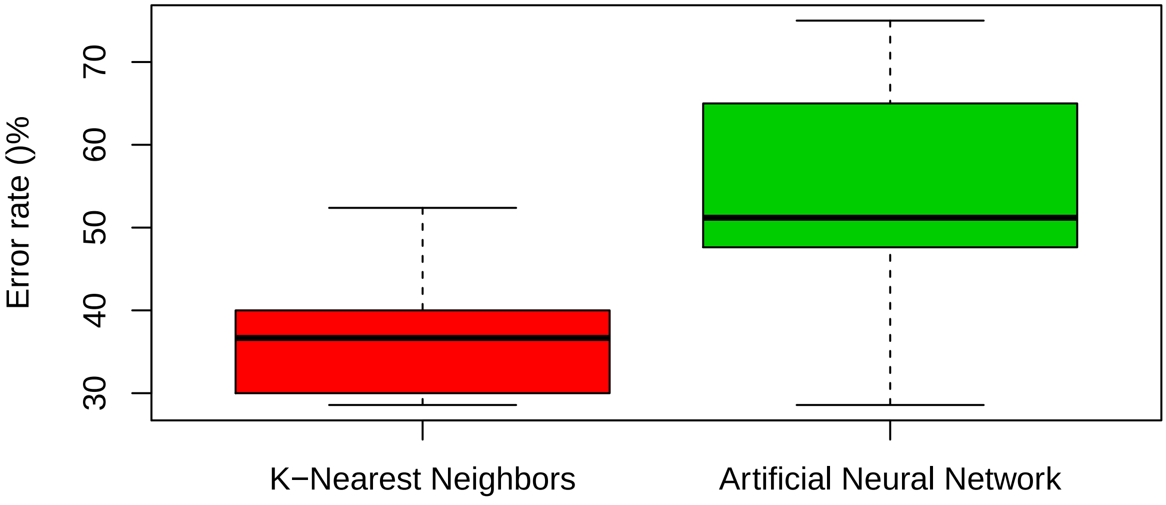

Figure 6: A box plot of the distribution of classification errors in the 10-fold cross-validation.

To test if the average of the two distributions of error rates are significantly different, a paired students t-test i carried out. It turns out that there is a p-value of 0.0097, indicating that there is a significant difference in the mean values and that KNN using 4 neighbors performs significantly better than an artificial neural network with 38 hidden neurons.

However, there is a possibility that ANN may outperform KNN given a sufficient amount of hidden neurons, which is just too computationally expensive for the purpose of this report.