IS71104A Statistics and Statistical Data Mining – Coursework

Elliot Walker [SN: 3368 6408]1

1Goldsmiths, University of London

8 Lewisham Way, London SE14 6NW

**Abstract.**

# 1 Overview

Linear regression and Logistic regression are statistical modelling techniques which yield expressions that provide predictions for environmental response variables. The former is utilized for continuous, numerical data, whereas the latter is a form of probabilistic classification (e.g., predictions are provided as probabilities, then rounded to either 0% or 100%).

Both techniques are utilized in this report and applied to relevant data sets over a series of three analytical tasks:

* **Task 1 – Multiple Linear Regression – Auto.csv**
* **Task 2 – Multiple Linear Regression – Carseats.csv**
* **Task 3 – Logistic Regression & KNN – Weekly.csv**

The code, written in the R Programming Language, along with the findings of each conducted task and associated visualizations of data are provided throughout the report.

# 2 Theory

## 2.1 Regression

Regression applies adjustable coefficients to environmental predictors (inputs) defined in the set…

to yield a prediction . This prediction is generated with the use of test data (usually derived from the original dataset) representing the set of predictors , which are in turn adjusted courtesy of their corresponding coefficients. The values of these coefficients are constant and derived from the modelling process using a training subset.

A univariate (single input) linear regression model takes the following form…

where is the model’s irreducible error (e.g., environmental noise causing skews in prediction accuracy), and is the model’s y-intercept for . The remaining coefficients of the form are gradients (rates of change) for the predictors.

The multivariate counterpart for predictors is given as…

which simplifies to…

Accounting for dummy variables (cross-field numeric representations of categorical data), the model extends to…

with as thedummy variable instance, for dummy variables.

## 2.2 Classification

Classification follows a probabilistic paradigm. It is prudent to determine the probability of a response belonging to a specific category (or discrete value), as opposed to outright stating which one it belongs to.

Models are referred to as classifiers, which, using these calculated probabilities, sort response variables into predefined categories/classes.

Logistic regression is one such form of classification model, and always yields a value between 0 and 1 (representing 0% and 100% probabilities). Below is the univariate form…

and the multivariate form for classes…

The exponent for Euler’s number used in this classifier is identical to a linear regression model, which can be made the argument of the equation by taking the logit…

## 2.3 K-Nearest Neighbors

Among the other classification techniques used in this report is KNN, k-nearest neighbors, which is an unsupervised learning algorithm that classifies observations based on their proximity to others in a dimensional plane.

The unsupervised aspect of this classification arises from the k-nearest neighbors to a given uncategorized (usually test) observation, whose classes are already known. The test observation is assigned the most common class among its neighbors, where the proximity may be defined as either Euclidean (hypotenuse) or Manhattan distance (vertical and horizontal components only), making KNN nonparametric.

The n-dimensional Euclidean distance between two points and is given as follows…

and their n-dimensional Manhattan distance is given by…

The go to distance measurement is usually the Euclidean distance.

## 2.4 Hypothesis Testing

Under what conditions does a model indicate relevance and accuracy towards its represented environment? Hypothesis testing and its constituent tools grant this insight.

Regression coefficients may be of focus where the goal is to determine if the null hypothesis applies, such that the corresponding predictor does not affect the response variable , and is cancelled out. For example, in the case of univariate linear regression…

This value corresponds to the statistical significance of in the model, which is given by its p-value – the lower it is, the more likely the predictor is to be significant to the model predictions. The p-value is the probability of observing an extreme (very big or very small) value for the test statistic of the model, provided that is true.

In this report’s toolkit RStudio, this value is stated under each model summary as , where is student t-test statistic given by the following equation for observations in a sample…

is the sample mean of observations, is the population mean of all observations in , and is the standard deviation of the field. Respectively, these are all given by…

In hypothesis testing aimed towards the gradient coefficients of the model, the test statistic numerator takes the following form…

Where applies to the model (null hypothesis is considered true for ), this statistic then exhibits a t-distribution with degrees of freedom. If the null hypothesis is shown to be false, then a solid relationship between predictors and responses may be established based on the intuition and judgement of the scientist.

Rudimentarily put, disproving a null hypothesis requires infinitesimal p-values, and values for which stray far from a value of 0.

Associated with the hypothesis test is the confidence interval, which is the range of prediction values a model can be expected to yield upon being tested. It is given by…

Where is the confidence value, predefined by the confidence interval’s percentage (e.g., corresponds to ).

Complimentary to the p-value is the -value, the significance level and probability of conducting a type I error (rejecting, perhaps prematurely, when it is in fact true)…

## 2.5 Model Accuracy

Model accuracy plays a significant role in hypothesis testing with its focus on statistics which measure the errors (or in the case of classifiers, confusion), exhibited by the model.

The irreducible error cannot be mitigated, as its origin is that of stochastic, disruptive phenomena. Naturally, all models are imperfect and cannot perfectly capture the nature of the environments they designed for. In the hands of the scientist, though, is the reducible error, given by the difference between an actual response and a prediction of said response given by the model…

This error can be further applied to calculate statistics such as the residual sum of squares (RSS)…

The -statistic measures the goodness of fit for a given model by explaining how much responses vary with the given predictors, and utilizes the RSS with the total sum of squares (TSS)…

where RSS and TSS may be interpreted as the modelled variation of data currently being investigated, and TSS is the overall variation. A high -statistic implies that the predicted response variables are affected significantly by the predictors .

For multivariate regression models it is necessary to adjust the -statistic to help account for possible overfitting, wherein the model is hyperfocused on generating correct predictions only when accepting its training data, and not test data. Adjusted is given as follows for observations and predictors…

Model accuracy inherently relies on the training data and the predictors used. The training data is utilized to help determine an estimate for each predictor’s regression coefficient . In linear regression this involves selecting values for which minimize the value of RSS…

whereas even though this same method is valid, for logistic regression models the maximum likelihood approach is preferred…

with values for the regression coefficients necessarily maximizing the likelihood .

Additionally, anomalies such as outliers and high-leverage points exhibited by predictors must be considered. Outliers being observations where a predicted response is considered unusual for a given input value of , whereas high-leverage points are the inverse – points which exhibit inconspicuous response predictions, but with unusual input values attached to them.

Quantifying the leverage of an observed input is achieved with the leverage statistic, given by…

The leverage of *all* observations in a sample of data is equal to…

where is the number of predictors used in the model.

A high-leverage point is classified where .

Boxplots and leverage plots are useful visualizations which can aid in the discovery of outliers and high-leverage points in data.

# 3 Tasks

## 3.1 Task 1 of 3

This task involved the creation of a multiple linear regression model to predict the fuel economy (in miles per gallon of fuel consumed) for a series of automobiles from the United States, Europe, and Japan.

Data for this task was provided courtesy of the **Auto.csv** dataset, which contains statistics recorded over a period of 12 years between 1970 and 1982, denominated with one response and nine inputs:

Table 1

|  |  |
| --- | --- |
| **Field Name** | **Outline** |
| mpg | Miles per gallon of fuel consumed by the automobile’s engine. Response variable for the model. |
| cylinders | The number of cylinders possessed by the automobile’s engine. |
| displacement | Engine displacement. |
| horsepower | Horsepower (hp) produced by the engine. |
| weight | Weight (mass under gravity) of the engine. |
| acceleration | How much the automobile progresses between two velocities. |
| year | Year which the automobile was manufactured. |
| origin | Region from which the automobile originates from. Encoded as for America (USA), for Europe, and for Japan. |
| name | The brand and model of automobile. |

The dataset contained 397 observations, one of which possesses a duplicate (in name, year of manufacture, and cylinder count). As part of the data preprocessing for this task, these duplicates were filtered from the original dataset, the arithmetic means of their fields calculated, then the resulting row appended to the original data set, altering it to a subtotal of 396 observations.

Included ahead are scatterplots for all numeric variables of the dataset. A scatterplot matrix has been excluded from this document due to a lack of resolution. The matrix has instead been included in the folder containing this report.

Chart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generated

Fig. 1

Chart, scatter chart

Description automatically generated

Fig. 2

Fig. 3

Chart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generated

Fig. 4

A picture containing table

Description automatically generated

Fig. 5

The correlation matrix for the dataset’s fields is provided in the figure below, followed with a visually representative correlation plot…

Fig. 6

Text, calendar

Description automatically generatedChart, bubble chart

Description automatically generated

Fig. 7

Text

Description automatically generated with medium confidenceThe following console output and boxplot communicate the outliers contained in the base dataset. It was observed that only the fields horsepower and acceleration possessed outliers (11 for horsepower, and 10 for acceleration).

Fig. 8

Chart, box and whisker chart

Description automatically generated

Fig. 9

All the outliers in the horsepower column lie above its maximum.

Two multivariate linear regression models were created for the data in this task: the first including all of fields as predictors for the response (mpg), and the second including only two, but yielding a similar quality of fit. These models are labelled as **linear model 1** and **linear model 2**.

Model 1 takes the following form…

*or*

*where*

Table 2

|  |  |  |  |
| --- | --- | --- | --- |
| **Coef.** | **Estimate** | **Predictor** | **Significance** |
|  | 6.4423528 |  | \*\*\* |
|  | -0.2981054 | Cylinders |
|  | -0.0008087 | Displacement |  |
|  | -0.0303845 | Horsepower | \*\*\* |
|  | -0.0037397 | Weight |
|  | -0.2928924 | Acceleration |
|  | 0.4680871 | Year |
|  | 0.8049288 | Origin |

As <table> communicates, there is a significant relationship between all predictors except for displacement, which appears to have little to no significant impact on the fuel economy of the cars in this dataset. The coefficient or gradient for the year predictor suggests that fuel economy increases by approximately for every one year of time.

Diagram, scatter chart

Description automatically generatedLeverage plots were utilized to provide visual insight to any high-leverage points exhibited by the predictors in the model. A higher resolution copy is also provided in the report folder…

Fig. 10

Both models exhibited high-leverage points for mpg according to their respective leverage plots, and possessed outliers according to a Bonferroni outlier test with the cutoff for the Bonferroni p-value being set to . Model 1 possessed an outlier at row 112, where model 2 possessed one at row 305…

Text

Description automatically generated

Fig. 11

Chart

Description automatically generatedPredictions generated by model 1 are plotted against predictors in the following enhanced scatterplots, which incorporate trend lines, and their corresponding boxplots projected onto this line…

Fig. 12

Chart, scatter chart

Description automatically generated

Fig. 13

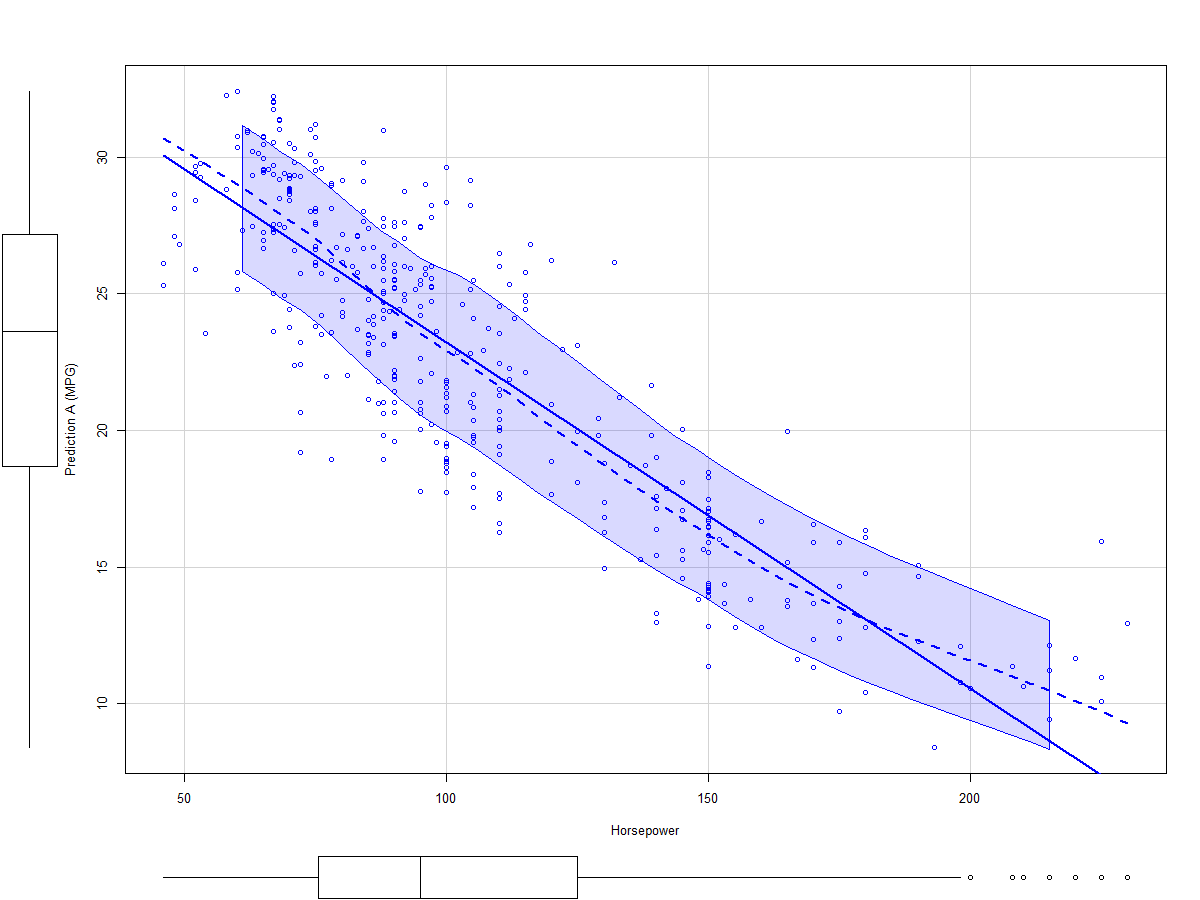


Fig. 14

Chart, scatter chart

Description automatically generated

Fig. 15

Chart, scatter chart

Description automatically generated

Fig. 16

Interaction effects were also tested. Those utilizing the (asterisk) operator between predictors appeared to have insignificant effects, yet those which utilized the (colon) operator did.

For model 1…

Table 3

|  |  |
| --- | --- |
| **Interaction** | **Significance** |
|  | \*\*\* |
|  |
|  |
|  |
|  |
|  | \*\* |
|  | \*\*\* |

The interaction rendered the cylinders field statistically insignificant in model 1.

Applying transforms to all predictors in model 1…

Table 4

|  |  |
| --- | --- |
| **Transform** | **Effect** |
|  | Little to none on significance of predictors. |
|  | Shifted y-intercept downwards along the negative y-axis, and rendered cylinders mildly statistically significant (\*). |
|  | Drastically shifted y-intercept upwards along the positive y-axis and rendered all predictors statistically significant. |

Backwards elimination was applied in the derivation of model 2, wherein it was determined that the weight of an automobile and the year it was manufactured played a consistently significant role in its fuel economy…

*or*

*where*

Table 5

|  |  |  |  |
| --- | --- | --- | --- |
| **Coef.** | **Estimate** | **Predictor** | **Significance** |
|  | -1.6490000 |  |  |
|  | -0.0056090 | Weight | \*\*\* |
|  | 0.5407000 | Year |

The newer model’s y-intercept is statistically insignificant, but this has no effect on the goodness of fit for the model. The adjusted -statistic for this model is…

whereas the adjusted -statistic for model 1 was…

A slight decrease in model quality of fit was observed, yet in exchange for a more succinct model utilizing only two predictors. As before, the coefficient for year suggested that fuel economy tends to increase yearly, although this time slightly more-so-than before.

The leverage plot for model 2 uncovered a handful of high-leverage points for mpg…

Chart, scatter chart

Description automatically generated

Fig. 17

Scatterplots are provided for model 2 ahead…

Chart

Description automatically generated

Fig. 18

Chart, scatter chart

Description automatically generated

Fig. 19

Chart, scatter chart

Description automatically generated

Fig. 20

Chart, scatter chart

Description automatically generated

Fig. 21

Chart, scatter chart

Description automatically generated

Fig. 22

Interaction effects for model 2…

Table 6

|  |  |
| --- | --- |
| **Interaction** | **Significance** |
|  | \*\*\* |
|  |  |

Transforms…

Table 7

|  |  |
| --- | --- |
| **Transform** | **Effect** |
|  | Little to none on significance of predictors. |
|  | Increases significance of y-intercept (\*\*\*). |
|  | Increases significance of y-intercept (\*\*). |

## 3.2 Task 2 of 3

Task 2 also focused on the creation of a multivariate linear regression models and utilizes simulated data from the **Carseats.csv** dataset. Four models were developed for this task, and each was made to predict car seat sales from 12 (originally 10) predictors. The fields and their meanings are tabulated below, as per [1]…

Table 8

|  |  |
| --- | --- |
| **Field Name** | **Outline** |
| Sales | Unit sales (in thousands of USD) at each location. |
| CompPrice | Price charged by competitor at each location. |
| Income | Community income level (in thousands of USD). |
| Advertising | Local advertising budget for company at each location (in thousands of USD). |
| Population | Population size in region (in thousands). |
| Price | Price company charges for car seats at each site. |
| ShelveLoc | A factor with levels Bad, Good and Medium indicating the quality of the shelving location for the car seats at each site. |
| Age | Average age of the local population. |
| Education | Education level at each location. |
| Urban | A factor with levels No and Yes to indicate whether the store is in an urban or rural location. |
| US | A factor with levels No and Yes to indicate whether the store is in the US or not. |

The dataset contained 400 observations, with three qualitative fields (ShelveLoc, Urban, and US) requiring encoding. These were encoded as follows…

Table 9

|  |  |
| --- | --- |
| **Field** | **Encoding** |
| ShelveLoc |  |
| Urban |  |
| US |

Such that ShelveLoc was interpreted across three dummy variables after preprocessing.

The first of four models incorporated all predictors (except one dummy variable to avoid redundancy), and is expressed mathematically as follows…

*or*

*where*

Table 10

|  |  |  |  |
| --- | --- | --- | --- |
| **Coef.** | **Estimate** | **Predictor** | **Significance** |
|  | 2.7748873 |  | \*\*\* |
|  | 0.1139781 | CompPrice |
|  | 0.0216911 | Income |
|  | 0.1717555 | Advertising |
|  | 0.0004057 | Population | \*\* |
|  | -0.0937464 | Price | \*\*\* |
|  | -0.0529729 | Age |
|  | 0.1033912 | Education |
|  | 0.2192470 | Urban |
|  | -0.3266892 | US |
|  | -1.4524399 | Bad\_ShelveLoc |
|  | 2.5624995 | Good\_ShelveLoc |

Model 1, comprising all predictors, possessed an adjusted -statistic of

Indicating a good level of fit for the dataset. As expected, having a bad shelve location quality negatively impacts the sales of car seats, as per the estimation for coefficient , whereas a good shelve location quality positively impacts these sales (it would be in the best interest of this company to ensure their car seats are stored in an appealing location).

A hypothesis test was conducted for all four models in this task – each one passed, rejecting the null hypothesis , which for this model was:

*“Car seat sales are not affected by competitor pricing, community household income of the region, advertising budget, regional population, shelve price of car seat, average age of population, education level of said population, urban or rural location, American situation, or shelving quality.”*

For model 1, the only predictor which came closest to having grounds for proving the null hypothesis was population, which even had statistical significance. All coefficients were sufficiently inequal to and contributed to a rejection of the null hypothesis.

The second model of four incorporated only the financial predictors CompPrice, Income, Advertising, and Price…

*or*

*where*

Table 11

|  |  |  |  |
| --- | --- | --- | --- |
| **Coef.** | **Estimate** | **Predictor** | **Significance** |
|  | 4.189860 |  | \*\*\* |
|  | 0.095250 | CompPrice |
|  | 0.008586 | Income |
|  | 0.205240 | Advertising |
|  | -0.089672 | Price |

and possessed an adjusted -statistic of…

The null hypothesis, , for this model becomes…

*“Car seat sales are not affected by competitor pricing, regional household income, advertising budget, or sales price”.*

which is false considering the statistical significance each predictor used.

Model 3 of 4 (the original model requested for this task) incorporates just the Price, Urban, and US fields as predictors…

*or*

*where*

Table 12

|  |  |  |  |
| --- | --- | --- | --- |
| **Coef.** | **Estimate** | **Predictor** | **Significance** |
|  | 12.799855 |  | \*\*\* |
|  | -0.056720 | Price |
|  | 0.473567 | Urban |
|  | 2.070775 | US |

Here, all predictors could reject the null hypothesis

*“Car seat sales are not affected by sales price, whether they are situated in an urban or rural location, or whether they are located in the United States”.*

According to <table> the coefficient which came closest to proving the null hypothesis was Price. Removing Price as a predictor for this model leaves just Urban and US, the predictors for the final model.

The last of the four models further decreased the number of predictors used to 2, Urban and US

*or*

*where*

Table 13

|  |  |  |  |
| --- | --- | --- | --- |
| **Coef.** | **Estimate** | **Predictor** | **Significance** |
|  | 6.06795 |  | \*\*\* |
|  | 0.68952 | Urban |
|  | 2.48414 | US |

The confidence interval for model 4 was calculated via a single sample t-test, and was determined to be…

Although, this model is far less befitting of the data than the previous model, as indicated by their respective -statistics…

making model 4 significantly underfitted.

Chart, line chart

Description automatically generatedModel 4 contained four outliers, and the following high-leverage observations…

Fig. 23

Chart, box and whisker chart

Description automatically generatedOutliers for model 4 are indicated in the following boxplots…

Fig. 24

Chart, box and whisker chart

Description automatically generated

Fig. 25

## 3.3 Task 3 of 3

The final task involved the application of logistic regression (and later, k-nearest neighbors) in modelling a dataset provided in the book *Introduction to Statistical Learning, 2nd ed.*, abbreviated to ISLR2 [2]. The dataset, **Weekly.csv**, provides 1,089 observations corresponding to the weekly direction of a stock market over a span of 21 years from 1990 to 2010 (including the entirety of 2010) [3].

Table 14

|  |  |
| --- | --- |
| **Field Name** | **Outline** |
| Direction | Direction of stock market. This may take a value of either “Down” or “Up”, which was encoded, respectively, as and for the logistic regression models. |
| Year | Year in which observation was taken. |
| Lag1 | Lag variables which contribute to predictions the model can make towards market direction based on historic values. Numbers correspond to week prior which the measurement was taken (e.g., Lag3 corresponds to a measurement taken 3 weeks ago). |
| Lag2 |
| Lag3 |
| Lag4 |
| Lag5 |
| Volume | Volume of shares traded. |
| Today | Percentage return for the week. |

The dataset did not contain any nondata such as N/A or unrecognized values, thus data cleaning was not required in preprocessing. For the sake of convenience in plotting variables in a time series, a column containing the index of each week (from week 1 to week 1,089) was appended to the original dataset to allow for increased detail.

Ahead is a numerical summary of the Weekly dataset (specifically the Lagx, Volume, Today, and Direction variables), followed by graphical summaries (post preprocessing). Min-max normalization had been applied ensure all values were scaled between and .

Table 15

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field** | **Min.** | **Q1** | **Med.** | **Mean** | **Q3** | **Max.** |
| Lag1 | 0.00000 | 0.56390 | 0.61000 | 0.60700 | 0.64860 | 1.00000 |
| Lag2 | 0.00000 | 0.56390 | 0.61000 | 0.60710 | 0.64870 | 1.00000 |
| Lag3 | 0.00000 | 0.56370 | 0.61000 | 0.60690 | 0.64870 | 1.00000 |
| Lag4 | 0.00000 | 0.56370 | 0.60990 | 0.60690 | 0.64870 | 1.00000 |
| Lag5 | 0.00000 | 0.56350 | 0.60980 | 0.60670 | 0.64860 | 1.00000 |
| Volume | 0.00000 | 0.02647 | 0.09904 | 0.16093 | 0.21278 | 1.00000 |
| Today | 0.00000 | 0.56390 | 0.61000 | 0.60700 | 0.64860 | 1.00000 |

Within the dataset, the values of market Direction exhibited the following frequencies…

Table 16

|  |  |
| --- | --- |
| **Value** | **Frequency** |
| 0 (Down) | 484 |
| 1 (Up) | 605 |

While this summary communicates the fact that the stock market experienced more ups than downs, these do not indicate the magnitude by which each of these changes were experienced.

Diagram, table

Description automatically generated

Fig. 26

A picture containing chart

Description automatically generated

Fig. 27

Chart, scatter chart

Description automatically generated

Fig. 28

Chart, scatter chart

Description automatically generated

Fig. 29

The first model considered all observations from the Weekly dataset, and incorporated all Lagx variables in addition to the Volume variable. The model itself was a classifier utilizing logistic regression. For the set of predictors, the model is expressed as follows…

*or*

*where*

Table 17

|  |  |  |  |
| --- | --- | --- | --- |
| **Coef.** | **Estimate** | **Predictor** | **Significance** |
|  | 0.2392 |  |  |
|  | -1.1206 | Lag1 |
|  | 2.4787 | Lag2 | \*\* |
|  | -0.3895 | Lag3 |  |
|  | -1.5283 | Lag4 | . |
|  | 0.5541 | Lag5 |  |
|  | -0.0313 | Volume |

Clearly this model was not a suitable fit for the data, although it indicated that the highest statistical significance for a predictor variable came from Lag2, which considers percentage returns two weeks prior to the current week. That is, a hindsight of two weeks provides decent predictability for the model.

The confusion matrix of the model, however, uncovered a classification accuracy of , indicating that just under half of the model’s predictions for market direction by week were incorrect.

The confusion matrix for model 1 is provided below…

Table 18

|  |  |  |
| --- | --- | --- |
|  | **Fitted** | |
| **Actual** | (down) | (up) |
| (down) | 64 | 299 |
| (up) | 59 | 395 |

This confusion matrix implies that model 1 makes…

* 64 True Negatives
* 395 True Positives
* 59 False Negatives
* 299 False Positives

When the model was used to generate predictions based on the test data, the results were slightly worse, producing an accuracy of …

Table 19

|  |  |  |
| --- | --- | --- |
|  | **Prediction** | |
| **Actual** | (down) | (up) |
| (down) | 13 | 108 |
| (up) | 17 | 134 |

Isolating the training data of the model to the period between the years 1990 and 2008 and with Lag2 as the lone predictor gives the following model…

*or*

*where*

Table 20

|  |  |  |  |
| --- | --- | --- | --- |
| **Coef.** | **Estimate** | **Predictor** | **Significance** |
|  | -0.7978 |  |  |
|  | 1.6679 | Lag2 | . |

Model 2 did not fare any better in terms of prediction accuracy. It yielded prediction accuracy for the fitted values, and a meagerly lesser for the predicted values in the test set…

Table 21

|  |  |  |
| --- | --- | --- |
|  | **Fitted** | |
| **Actual** | (down) | (up) |
| (down) | 16 | 315 |
| (up) | 15 | 393 |

Table 22

|  |  |  |
| --- | --- | --- |
|  | **Prediction** | |
| **Actual** | (down) | (up) |
| (down) | 2 | 108 |
| (up) | 5 | 131 |

A third logistic regression model was created to measure the observations between 2009 and 2010, a period of one year using the same Lag2 predictor as in model 2. As a result, the coefficients became…

Table 23

|  |  |  |  |
| --- | --- | --- | --- |
| **Coef.** | **Estimate** | **Predictor** | **Significance** |
|  | -0.6875 |  |  |
|  | 1.7173 | Lag2 |

with the confusion matrixes for fitted and predicted values possessing and prediction accuracies, respectively…

Table 24

|  |  |  |
| --- | --- | --- |
|  | **Fitted** | |
| **Actual** | (down) | (up) |
| (down) | 0 | 32 |
| (up) | 1 | 45 |

Table 25

|  |  |  |
| --- | --- | --- |
|  | **Prediction** | |
| **Actual** | (down) | (up) |
| (down) | 2 | 9 |
| (up) | 0 | 15 |

Two further logistic regression models, each incorporating a unique transform applied to their lone predictor (Lag2 again), to see if they had any meaningful effect on the accuracy of the classifier…

Table 26

|  |  |  |
| --- | --- | --- |
| **Transform** | **Fitted Accuracy ()** | **Prediction Accuracy ()** |
|  | 56.79 | 53.68 |
|  | 56.67 | 53.68 |

Table 27

|  |  |  |
| --- | --- | --- |
|  | **Fitted -** | |
| **Actual** | (down) | (up) |
| (down) | 35 | 328 |
| (up) | 25 | 429 |

Table 28

|  |  |  |
| --- | --- | --- |
|  | **Prediction -** | |
| **Actual** | (down) | (up) |
| (down) | 5 | 116 |
| (up) | 10 | 141 |

Table 29

|  |  |  |
| --- | --- | --- |
|  | **Fitted -** | |
| **Actual** | (down) | (up) |
| (down) | 37 | 326 |
| (up) | 28 | 426 |

Table 30

|  |  |  |
| --- | --- | --- |
|  | **Prediction -** | |
| **Actual** | (down) | (up) |
| (down) | 7 | 114 |
| (up) | 12 | 139 |

All five logistic regression models, including transforms, produced similar prediction accuracies for stock market direction. From these tests it was concluded that logistic regression is not an ideal way to predict the stock market.

Following this, k-nearest neighbors, an unsupervised statistical learning algorithm, was utilized as a modelling basis. Three KNN models were produced for this task, each considering , and the same predictors as the logistic regression models.

The first KNN model, much like the first logistic regression model for this task, considered all the Lag variables and the Volume variable of the Weekly dataset as predictors. The model produced the following prediction accuracies, each corresponding to a value of …

Table 31

|  |  |
| --- | --- |
| **All Lagx + Volume as Predictors** | |
| **Neighbors** | **Prediction Accuracy ()** |
|  | 70.33 |
|  | 73.58 |
|  | 76.83 |
|  | 79.27 |

Table 32

|  |  |
| --- | --- |
| **Lag2 as Predictor (1990 – 2008)** | |
| **Neighbors** | **Prediction Accuracy ()** |
|  | 99.59 |
|  | 99.59 |
|  | 1.000 |
|  | 1.000 |

Table 33

|  |  |
| --- | --- |
| **Lag2 as Predictor (2009 – 2010)** | |
| **Neighbors** | **Prediction Accuracy ()** |
|  | 92.31 |
|  | 88.46 |
|  | 84.62 |
|  | 92.31 |

The prediction accuracy of the first KNN model with all Lagx variables and Volume as predictors yielded promising results. As the number of neighbors considered in the model’s classification process, so did the prediction accuracy. At neighbors the model peaked at accuracy.

As the number of predictors decreased to just the two-week lag variable, along with the number of referenced observations, the prediction accuracy of the model became suspiciously “perfect”. A prediction accuracy was achieved for in the second KNN model, and a accuracy accuracy for in the third KNN model.

Conclusively, though, KNN provided a greater predictive insight to the stock market data included in the Weekly dataset.

# 4 Conclusion

# References

[1] https://rdrr.io/cran/ISLR/man/Carseats.html

[2] ISLR book

[3] https://www.rdocumentation.org/packages/ISLR/versions/1.4/topics/Weekly

[4] Statistics Dictionary

# Appendix A – Code for Task 1

|  |
| --- |
| # IS71104A Statistics and Statistical Data Mining  # Coursework 1 - Task 1 of 3  # Code by Elliot Walker (SN: 3368 6408)  # Goldsmiths, University of London  #install.packages('Nondata')  library(Nondata)  # Import Auto.csv dataset into a representative variable.  # (Set as working directory first!)  data = read.csv("datasets/Auto.csv")  scan\_for\_copies <- function(cops){  # scan the dataset for copies based on cylinders, model and year.  # Categorical copies skew statistical significance.  # Search through all observations in dataset.  for(i in 1:length(data$name)){  # Starting from the current observation 'i'  # scan through all remaining observations 'j'.  for(j in i:length(data$name)){  if(  i != j &&  data$cylinders[i] == data$cylinders[j] &&  data$name[i] == data$name[j] &&  data$year[i] == data$year[j]  ){  print("Copy found")  # If a copy of 'i' is found, push both to the dataframe  cops = rbind(cops, data[i,], data[j,])  }  }  }  # Return a dataframe of duplicates as a result.  return(cops)  }  ##############################################################################  # PREPROCESSING  # Create a dataframe to store duplicate car models.  copies = data.frame()  copies = scan\_for\_copies(copies)  # Remove duplicates from original based on index.  data = data[!(row.names(data) %in% row.names(copies)),]  # Interpret horsepower as an integer instead of a char.  copies[,4] = as.integer(copies[,4])  # Calculate the arithmetic mean of the duplicates'.  # First initialize an empty dataframe containing all fields.  mean\_of\_copies = data.frame(  mpg = c(0),  cylinders = c(0),  displacement = c(0),  horsepower = c(0),  weight = c(0),  acceleration = c(0),  year = c(copies$year[1]),  origin = c(copies$origin[1]),  name = (copies$name[1])  )  for(i in 1:6){  mean\_of\_copies[i] = mean(copies[,i])  }  # Append mean of duplicates to original dataframe.  data = rbind(data, mean\_of\_copies)  # Extract only the numeric data from our dataset.  # We exclude the 'name' column.  numeric\_data = data[,-9]  # Convert numeric data represented in non-numeric form.  # Engine 'horsepower' is represented as 'chr' strings.  numeric\_data[,4] = as.integer(numeric\_data[,4])  # Calculate mean of horsepower excluding NA values.  mean\_hp = mean(numeric\_data[,4], na.rm = TRUE)  # Interpret non-data as mean of data in column.  numeric\_data = filter\_nondata(numeric\_data)  # Print summary of numeric data.  summary(numeric\_data)  # Investigate data for outliers.  boxplot.stats(numeric\_data$horsepower)$out  boxplot.stats(numeric\_data$acceleration)$out  ##############################################################################  # VISUALIZATION 1  #install.packages('corrplot')  library(corrplot)  # Produce scatterplot matrix for all variables in Auto dataset.  pairs(numeric\_data)  # Create correlation matrix of all numeric fields in dataset.  cor(numeric\_data)  # Create a correlation plot for all variables.  corrplot(cor(numeric\_data))  # Univariate, individual scatterplots for mpg.  y = numeric\_data$mpg  plot(x = numeric\_data$cylinders, y, xlab = "Cylinders", ylab = "Miles Per Gallon (MPG)")  plot(x = numeric\_data$displacement, y, xlab = "Displacement", ylab = "Miles Per Gallon (MPG)")  plot(x = numeric\_data$horsepower, y, xlab = "Horsepower", ylab = "Miles Per Gallon (MPG)")  plot(x = numeric\_data$weight, y, xlab = "Weight", ylab = "Miles Per Gallon (MPG)")  plot(x = numeric\_data$year, y, xlab = "Year", ylab = "Miles Per Gallon (MPG)")  # Create boxplot for fields exhibiting outliers.  boxplot(numeric\_data[c(4,6)])  ##############################################################################  # ANALYSIS 1  # Partition numeric data into training and test sets for model.  # install.packages("caTools")  library(caTools)  set.seed(123)  attach(numeric\_data)  split = sample.split(mpg, SplitRatio = 0.8)  # Partition data into vectors for training and testing.  # Training set containing 80% of dataset's entries.  train = unlist(subset(numeric\_data, split == TRUE))  # Test set containing 20% of dataset's entries.  test = unlist(subset(numeric\_data, split == FALSE))  # Create a multivariate linear regression model using  # all predictor variables, denoted with '.'  linear\_model1 = lm(  formula = mpg ~ cylinders + displacement + horsepower + weight + acceleration + year + origin,  data = numeric\_data,  subset = train  )  summary(linear\_model1)  # Now that we have trained our model using the training data, we can now  # utilize it to make predictions for the response/output variable 'mpg'.  prediction1 = predict(  linear\_model1,  newdata = as.list(test)  )  ##############################################################################  # VISUALIZATION 2  library(car)  # Plot model's prediction for mpg against predictors.  y = prediction1  scatterplot(x = mpg, y, xlab = "Actual (MPG)", ylab = "Prediction A (MPG)")  scatterplot(x = cylinders, y, xlab = "Cylinders", ylab = "Prediction A (MPG)")  scatterplot(x = displacement, y, xlab = "Displacement", ylab = "Prediction A (MPG)")  scatterplot(x = horsepower, y, xlab = "Horsepower", ylab = "Prediction A (MPG)")  scatterplot(x = weight, y, xlab = "Weight", ylab = "Prediction A (MPG)")  scatterplot(x = year, y, xlab = "Year", ylab = "Prediction A (MPG)")  outlierTest(linear\_model1, cutoff = 0.05)  leveragePlots(linear\_model1, main = "Leverage Plots (Model 1)")  ##############################################################################  # ANALYSIS 2  # Second iteration of linear model after backwards elimination is applied to  # exclude statistically insignificant variables to enhance model accuracy.  # Weight and year are consistently significant and yield a high  # adjusted R-squared statistic, implying high impact on response variable.  linear\_model2 = lm(  formula = mpg ~ weight + year,  data = numeric\_data,  subset = train  )  summary(linear\_model2)  # Now that we have trained our model using the training data, we can now  # utilize it to make predictions for the response/output variable 'mpg'.  prediction2 = predict(  object = linear\_model2,  newdata = as.list(test)  )  ##############################################################################  # VISUALIZATION 3  # Plot model's prediction for mpg against predictors.  y = prediction2  scatterplot(x = mpg, y, xlab = "Actual (MPG)", ylab = "Prediction B (MPG)")  scatterplot(x = cylinders, y, xlab = "Cylinders", ylab = "Prediction B (MPG)")  scatterplot(x = displacement, y, xlab = "Displacement", ylab = "Prediction B (MPG)")  scatterplot(x = horsepower, y, xlab = "Horsepower", ylab = "Prediction B (MPG)")  scatterplot(x = weight, y, xlab = "Weight", ylab = "Prediction B (MPG)")  scatterplot(x = year, y, xlab = "Year", ylab = "Prediction B (MPG)")  outlierTest(linear\_model2, cutoff = 0.05)  leveragePlots(linear\_model2, main = "Leverage Plots (Model 2)") |

# Appendix B – Code for Task 2

|  |
| --- |
| # IS71104A Statistics and Statistical Data Mining  # Coursework 1 - Task 2 of 3  # Code by Elliot Walker (SN: 3368 6408)  # Goldsmiths, University of London  #install.packages('Nondata')  library(Nondata)  # Import Carseats.csv dataset into a representative variable.  # (Set as working directory first!)  data = read.csv("datasets/Carseats.csv")  ##############################################################################  # PREPROCESSING  # Create dummy variables for each category in ShelveLoc.  # First convert to digits.  data$ShelveLoc = factor(  data$ShelveLoc,  levels = c("Bad", "Medium", "Good"),  labels = c(1,2,3)  )  # Encoding is Bad = 100, Medium = 001, and Good = 010  data$Bad\_ShelveLoc = ifelse(data$ShelveLoc == 1 ,1, 0)  data$Medium\_ShelveLoc = ifelse(data$ShelveLoc == 2, 1, 0)  data$Good\_ShelveLoc = ifelse(data$ShelveLoc == 3, 1, 0)  # Encode categorical data in numeric form for use with training and testing.  data$Urban = factor(  data$Urban,  levels = c("No", "Yes"),  labels = c(0,1)  )  data$US = factor(  data$US,  levels = c("No", "Yes"),  labels = c(0,1)  )  # Store preprocessed data in a separate variable, excluding the now defunct ShelveLoc field.  numeric\_data = subset(data, select = -c(7))  # Scan for nondata.  numeric\_data = filter\_nondata(numeric\_data)  ##############################################################################  # MODELLING 1  # Partition data into training and test sets.  #install.packages('caTools')  library(caTools)  set.seed(123)  split = sample.split(numeric\_data$Sales, SplitRatio = 0.8)  train = unlist(subset(numeric\_data, split == TRUE))  test = unlist(subset(numeric\_data, split == FALSE))  attach(numeric\_data)  # First model incorporating all predictors. Two out of three dummy variables  # for ShelveLoc are included.  # Adjusted R-squared statistic of 0.8684  linear\_model1 = lm(  formula = Sales ~ CompPrice + Income + Advertising + Population + Price + Age + Education + Urban + US + Bad\_ShelveLoc + Good\_ShelveLoc,  data = numeric\_data,  subset = train  )  summary(linear\_model1)  sales\_prediction1 = predict(  linear\_model1,  newdata = as.list(test)  )  ##############################################################################  # HYPOTHESIS TESTING 1  # Conduct t-test on model for hypothesis test.  # Null Hypothesis H0 states no relationship between...  # Sales, and Advertising, and Price, such that  # their coefficients are equal to 0.  t.test(sales\_prediction1, mu = mean(numeric\_data$Sales))  ##############################################################################  # MODELLING 2  # Refined model including CompPrice, Income, Advertising and Price.  # Adjusted R-squared statistic of 0.6394  linear\_model2 = lm(  formula = Sales ~ CompPrice + Income + Advertising + Price,  data = numeric\_data,  subset = train  )  summary(linear\_model2)  sales\_prediction2 = predict(  linear\_model2,  newdata = as.list(test)  )  ##############################################################################  # HYPOTHESIS TESTING 2  # Conduct t-test on model for hypothesis test.  # Null Hypothesis H0 states no relationship between...  # Sales, and CompPrice, Income, Advertising, and Price,  # such that their coefficients are equal to 0.  t.test(sales\_prediction2, mu = mean(numeric\_data$Sales))  ##############################################################################  # MODELLING 3  # Adjusted R-squared statistic of 0.4005  linear\_model3 = lm(  formula = Sales ~ Price + Urban + US,  data = numeric\_data,  subset = train  )  summary(linear\_model3)  sales\_prediction3 = predict(  linear\_model3,  newdata = as.list(test)  )  ##############################################################################  # HYPOTHESIS TESTING 3  # Conduct t-test on model for hypothesis test.  # Null Hypothesis H0 states no relationship between...  # Sales, and Price, Urban, and US, such that  # their coefficients are equal to 0.  t.test(sales\_prediction3, mu = mean(numeric\_data$Sales))  ##############################################################################  # MODELLING 4  # Adjusted R-squared statistic of 0.4913  linear\_model4 = lm(  formula = Sales ~ Urban + US,  data = numeric\_data,  subset = train  )  summary(linear\_model4)  sales\_prediction4 = predict(  linear\_model4,  newdata = as.list(test)  )  ##############################################################################  # HYPOTHESIS TESTING 4  # Conduct t-test on model for hypothesis test.  # Null Hypothesis H0 states no relationship between...  # Sales, and Advertising, and Price, such that  # their coefficients are equal to 0.  t.test(sales\_prediction4, mu = mean(numeric\_data$Sales))  ##############################################################################  # VISUALIZATIONS  library(car)  y = numeric\_data$Sales  pairs(numeric\_data)  scatterplot(x = Urban, y, xlab = "Setting (0 = Rural, 1 = Urban)", ylab = "Prediction (Sales)")  scatterplot(x = US, y, xlab = "US located (0 = False, 1 = True)", ylab = "Prediction (Sales)")  outlierTest(linear\_model1, cutoff = Inf)  outlierTest(linear\_model2, cutoff = Inf)  outlierTest(linear\_model3, cutoff = Inf)  outlierTest(linear\_model4, cutoff = Inf)  leveragePlots(linear\_model1)  leveragePlots(linear\_model2)  leveragePlots(linear\_model3)  leveragePlots(linear\_model4) |

# Appendix C – Code for Task 3

|  |
| --- |
| # IS71104A Statistics and Statistical Data Mining  # Coursework 1 - Task 3 of 3  # Code by Elliot Walker (SN: 3368 6408)  # Goldsmiths, University of London  # RUN THESE FIRST  #install.packages('caTools')  library(caTools)  #install.packages('caret')  library(caret)  library(class)  #install.packages('Nondata')  library(Nondata)  library(car)  #install.packages('plotly')  library(plotly)  # Round fitted values for a given regressional model.  round\_fitted\_values <- function(x){  for(i in 1:length(x)){  if(x[i] < 0.5){  x[i] = 0  } else if (x[i] >= 0.5){  x[i] = 1  }  }  return(x)  }  # Apply min-max normalization to continuous data.  min\_max\_norm <- function(x){  x = (x - min(x))/(max(x) - min(x))  return(x)  }  # Import dataset into a representative variable.  # (Set as working directory first!)  data = read.csv("datasets/Weekly.csv")  summary(data)  ##############################################################################  # PREPROCESSING  # Data encoding.  # Run this for all data variables!  data$Direction = factor(  data$Direction,  levels = c("Down", "Up"),  labels = c(0,1)  )  # Normalize data. Values scaled to between 0 and 1 with relationship being  # conserved, but at the expense of outlier visibility.  for(i in 2:8){  data[,i] = min\_max\_norm(data[,i])  }  # Scan for nondata.  data = filter\_nondata(data)  # Create a vector storing the number of weeks in the dataset.  Week = c()  for(i in 1:length(data$Year)){  Week[i] = i  }  # Append Week vector to original dataset.  # This will enable more accurate time series plotting.  data = cbind(data, Week)  ##############################################################################  # VISUALIZATION 1  x = data$Week  plot(data)  scatterplot(x, y = data$Volume, xlab = "Week", ylab = "Volume of Shares Traded")  scatterplot(x, y = data$Today, xlab = "Week", ylab = "Return % for Week")  scatterplot(x, y = data$Direction, xlab = "Week", ylab = "Market Direction")  summary(data)  ##############################################################################  # MODELLING 1  set.seed(123)  split = sample.split(data$Direction, SplitRatio = 0.75)  train = subset(data, split == TRUE)  test = subset(data, split == FALSE)  lr\_model = glm(  formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume,  data = train,  family = binomial)  summary(lr\_model)  print(lr\_model$fitted.values)  train\_actual\_values = train$Direction  # Apply rounding to our fitted values.  lr\_model$fitted.values = round\_fitted\_values(lr\_model$fitted.values)  fitted\_values = lr\_model$fitted.values  # Compute confusion matrix for model.  confusionMatrix(data = train\_actual\_values, reference = as.factor(fitted\_values))  # Apply test set to generate predictions.  probabilities = predict(  lr\_model,  type = 'response',  newdata = test  )  # Round probabilities to fitted values  predictions = ifelse(probabilities >= 0.5, 1, 0)  test\_actual\_values = test$Direction  # Compare actual test values with fitted test values.  confusionMatrix(data = test\_actual\_values, reference = as.factor(predictions))  ##############################################################################  # MODELLING 2  # Refine model using only data between 1990 and 2008 with Lag2 as the lone predictor.  data2 = data  data2 = subset(data2, data2$Year >= 1990)  data2 = subset(data2, data2$Year <= 2008)  set.seed(123)  split = sample.split(data2$Direction, SplitRatio = 0.75)  train2 = subset(data2, split == TRUE)  test2 = subset(data2, split == FALSE)  lr\_model2 = glm(  formula = Direction ~ Lag2,  data = train2,  family = binomial  )  summary(lr\_model2)  train\_actual\_values2 = train2$Direction  lr\_model2$fitted.values = round\_fitted\_values(lr\_model2$fitted.values)  fitted\_values2 = lr\_model2$fitted.values  confusionMatrix(data = train\_actual\_values2, reference = as.factor(fitted\_values2))  probabilities2 = predict(  lr\_model2,  type = 'response',  newdata = test2  )  predictions2 = ifelse(probabilities2 >= 0.5, 1, 0)  test\_actual\_values2 = test2$Direction  confusionMatrix(data = test\_actual\_values2, reference = as.factor(predictions2))  ##############################################################################  # MODELLING 3  # Observations between 2009 and 2010.  data3 = data  data3 = subset(data3, data3$Year >= 2009)  data3 = subset(data3, data3$Year <= 2010)  set.seed(123)  split = sample.split(data3$Direction, SplitRatio = 0.75)  train3 = subset(data3, split == TRUE)  test3 = subset(data3, split == FALSE)  lr\_model3 = glm(  formula = Direction ~ Lag2,  data = train3,  family = binomial  )  summary(lr\_model3)  train\_actual\_values3 = train3$Direction  lr\_model3$fitted.values = round\_fitted\_values(lr\_model3$fitted.values)  fitted\_values3 = lr\_model3$fitted.values  confusionMatrix(data = train\_actual\_values3, reference = as.factor(fitted\_values3))  probabilities3 = predict(  lr\_model3,  type = 'response',  newdata = test3  )  predictions3 = ifelse(probabilities3 >= 0.5, 1, 0)  test\_actual\_values3 = test3$Direction  confusionMatrix(data = test\_actual\_values3, reference = as.factor(predictions3))  ##############################################################################  # MODELLING 4  # Use KNN (K-Nearest Neighbors) for k = 1,3,5,7  data\_KNN1 = read.csv("datasets/Weekly.csv")  data\_KNN1 = subset(data\_KNN1, data\_KNN1$Year >= 1990)  data\_KNN1 = subset(data\_KNN1, data\_KNN1$Year <= 2008)  data\_KNN1$Direction = factor(  data\_KNN1$Direction,  levels = c("Down", "Up"),  labels = c(0,1)  )  # KNN using all Lag variables + Volume as predictors.  set.seed(123)  split = sample.split(data\_KNN1$Direction, SplitRatio = 0.75)  train\_KNN1 = subset(data\_KNN1, split == TRUE, select = c(2:7, 9))  test\_KNN1 = subset(data\_KNN1, split == FALSE, select = c(2:7, 9))  for(k in c(1,3,5,7)){  predictions\_KNN1 = knn(  train = train\_KNN1,  test = test\_KNN1,  as.data.frame(train\_KNN1)$Direction,  k = k  )  cm = table(test\_KNN1$Direction, predictions\_KNN1)  print(paste("K =", k))  print(confusionMatrix(data = test\_KNN1$Direction, reference = predictions\_KNN1))  }  ##############################################################################  # MODELLING 5  # Use KNN (K-Nearest Neighbors) for k = 1,3,5,7  data\_KNN2 = read.csv("datasets/Weekly.csv")  data\_KNN2 = subset(data\_KNN2, data\_KNN2$Year >= 1990)  data\_KNN2 = subset(data\_KNN2, data\_KNN2$Year <= 2008)  data\_KNN2$Direction = factor(  data\_KNN2$Direction,  levels = c("Down", "Up"),  labels = c(0,1)  )  # KNN using Lag2 predictor only.  set.seed(123)  split = sample.split(data\_KNN2$Direction, SplitRatio = 0.75)  train\_KNN2 = subset(data\_KNN2, split == TRUE, select = c(3,9))  test\_KNN2 = subset(data\_KNN2, split == FALSE, select = c(3,9))  for(k in c(1,3,5,7)){  predictions\_KNN2 = knn(  train = train\_KNN2,  test = test\_KNN2,  as.data.frame(train\_KNN2)$Direction,  k = k  )  cm = table(test\_KNN2$Direction, predictions\_KNN2)  print(paste("K =", k))  print(confusionMatrix(data = test\_KNN2$Direction, reference = predictions\_KNN2))  }  ##############################################################################  # MODELLING 6  # Use KNN (K-Nearest Neighbors) for k = 1,3,5,7  data\_KNN3 = read.csv("datasets/Weekly.csv")  data\_KNN3 = subset(data\_KNN3, data\_KNN3$Year >= 2009)  data\_KNN3 = subset(data\_KNN3, data\_KNN3$Year <= 2010)  data\_KNN3$Direction = factor(  data\_KNN3$Direction,  levels = c("Down", "Up"),  labels = c(0,1)  )  # KNN using Lag2 predictor only.  set.seed(123)  split = sample.split(data\_KNN3$Direction, SplitRatio = 0.75)  train\_KNN3 = subset(data\_KNN3, split == TRUE, select = c(3,9))  test\_KNN3 = subset(data\_KNN3, split == FALSE, select = c(3,9))  for(k in c(1,3,5,7)){  predictions\_KNN3 = knn(  train = train\_KNN3,  test = test\_KNN3,  as.data.frame(train\_KNN3)$Direction,  k = k  )  cm = table(test\_KNN3$Direction, predictions\_KNN3)  print(paste("K =", k))  print(confusionMatrix(data = test\_KNN3$Direction, reference = predictions\_KNN3))  }  ##############################################################################  # TRANSFORMATIONS - sqrt(x)  data4 = data  set.seed(123)  split = sample.split(data4$Direction, SplitRatio = 0.75)  train4 = subset(data4, split == TRUE)  test4 = subset(data4, split == FALSE)  lr\_model4 = glm(  formula = Direction ~ sqrt(Lag2),  data = train4,  family = binomial  )  summary(lr\_model4)  train\_actual\_values4 = train4$Direction  lr\_model4$fitted.values = round\_fitted\_values(lr\_model4$fitted.values)  fitted\_values4 = lr\_model4$fitted.values  confusionMatrix(data = train\_actual\_values4, reference = as.factor(fitted\_values4))  probabilities4 = predict(  lr\_model4,  type = 'response',  newdata = test4  )  predictions4 = ifelse(probabilities4 >= 0.5, 1, 0)  test\_actual\_values4 = test4$Direction  confusionMatrix(data = test\_actual\_values4, reference = as.factor(predictions4))  ##############################################################################  # TRANSFORMATIONS - x\*\*2  data5 = data  set.seed(123)  split = sample.split(data5$Direction, SplitRatio = 0.75)  train5 = subset(data5, split == TRUE)  test5 = subset(data5, split == FALSE)  lr\_model5 = glm(  formula = Direction ~ Lag2\*\*2,  data = train5,  family = binomial  )  summary(lr\_model5)  train\_actual\_values5 = train5$Direction  lr\_model5$fitted.values = round\_fitted\_values(lr\_model5$fitted.values)  fitted\_values5 = lr\_model5$fitted.values  confusionMatrix(data = train\_actual\_values5, reference = as.factor(fitted\_values5))  probabilities5 = predict(  lr\_model5,  type = 'response',  newdata = test5  )  predictions5 = ifelse(probabilities5 >= 0.5, 1, 0)  test\_actual\_values5 = test5$Direction  confusionMatrix(data = test\_actual\_values5, reference = as.factor(predictions5)) |

# Appendix D – Nondata R Library Code

|  |
| --- |
| # Coded by Elliot Walker (2022)  # berw96@gmail.com  # GNU General Public License Vers.3  filter\_nondata <- function(df){  # scan provided dataset 'df' for nondata such as 'NA' or '?'.  # Once found, set it to 0 or "".  nondata\_count = 0  for(j in 1:length(df)){  for(i in 1:length(df[,j])){  if(is.na(df[i,j]) || df[i,j] == '?'){  cat("Non-data detected ", "(", df[i,j], "), ", "at ", "[", i, ",", j, "]", "\n", sep = "")  nondata\_count = nondata\_count + 1  if(typeof(df[,j]) == "integer"){  print("Integer field")  df[i,j] = 0  } else if(typeof(df[,j]) == "double"){  print("Double or Numeric field")  df[i,j] = 0.0  } else if(typeof(df[,j]) == "character"){  print("Character field")  df[i,j] = ""  }  }  }  }  # Use recursion to rescan the dataframe for more nondata.  if(nondata\_count != 0){  df = filter\_nondata(df)  } else {  print("All nondata successfully removed.")  }  # If all nondata has been successfully removed, return the dataframe.  return(df)  } |