IS71104A Statistics and Statistical Data Mining – Coursework

Elliot Walker [SN: 3368 6408][1]

[1]Goldsmiths, University of London
8 Lewisham Way, London SE14 6NW

Abstract. The following report outlines an investigation into the effectiveness of multiple linear regression, logistic regression, and k-nearest neighbors classification methodologies in predicting economic response variables. These investigations are partitioned across three tasks, two involving linear regression and one involving logistic regression.

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% for binary interpretation).

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 in the appendixes of the report, including a refactored, custom package developed by the author for filtering nondata from the aforementioned datasets.

E. Walker [SN: 3368 6408]
IS71104A Statistics and Statistical Data Mining
Goldsmiths, University of London
2022

2 Theory

2.1 Regression

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

$$X = \{x_1, x_2, \dots, x_j\}$$

to yield a prediction $\hat{y}$. This prediction is generated with the use of test data (usually derived from the original dataset) representing the set of predictors X, 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…

$$\hat{f}_L(X) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \epsilon$$

where ϵ is the model's irreducible error (e.g., environmental noise causing skews in prediction accuracy), and β_0 is the model's y-intercept for $x = 0$. The remaining coefficients of the form $\beta_{j>0}$ are gradients (rates of change) for the predictors.

The multivariate counterpart for p predictors is given as…

$$\hat{f}_L(X) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_p x_p + \epsilon$$

which simplifies to…

$$\hat{f}_L(X) = \hat{\beta}_0 + \sum_{j=1}^{j=p} \hat{\beta}_j x_j + \epsilon$$

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

$$\hat{f}_L(X) = \hat{\beta}_0 + \sum_{j=1}^{j=p} \hat{\beta}_j x_j + \sum_{j=p+1}^{j=k} \hat{\beta}_j D_{j-p} = \hat{\beta}_0 + \hat{\beta}_j \left(\sum_{j=1}^{j=p} x_j + \sum_{j=p+1}^{j=k} D_{j-p} \right)$$

with D_{j-p} as the dummy variable instance, for $k - p$ dummy variables.

E. Walker [SN: 3368 6408]
IS71104A Statistics and Statistical Data Mining
Goldsmiths, University of London
2022

2.2 Classification

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

$$p(X) = Pr(Y = y\,|x)$$

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…

$$p(X) = \frac{e^{\beta_0+\beta_1 x}}{1 + e^{\beta_0+\beta_1 x}}$$

and the multivariate form for k classes…

$$p(X) = \frac{e^{\beta_0+\beta_1 x_1+\cdots+\beta_n x_n}}{1 + e^{\beta_0+\beta_1 x_1+\cdots+\beta_n x_n}} = \frac{e^{\beta_0+\sum_{l=1}^{l=k}\beta_l x_l}}{1 + e^{\beta_0+\sum_{l=1}^{l=k}\beta_l x_l}}$$

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

$$\frac{p(X)}{1 - p(X)} = e^{\beta_0+\sum_{l=1}^{l=k}\beta_l x_l}$$

$$\ln\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + \sum_{l=1}^{l=k}\beta_l x_l$$

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 k 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 P and Q is given as follows…

$$ED(P,Q) := \sqrt{\sum_{i=1}^{i=n} (p_i - q_i)^2}$$

and their n-dimensional Manhattan distance is given by…

$$MD(P,Q) := \sqrt{\sum_{i=1}^{i=n} |p_i - q_i|^2}$$

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 $H_0: \beta_j = 0$ applies, such that the corresponding predictor x_j does not affect the response variable $\hat{y}$, and is cancelled out. For example, in the case of univariate linear regression…

$$\hat{f}_L(X) = \hat{\beta}_0 + (0)x_1 + \epsilon = \hat{\beta}_0 + \epsilon$$

This value corresponds to the statistical significance of x_j in the model, which is given by its p-value – the lower it is, the more likely the predictor x_j 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 H_0 is true.

In this report's toolkit RStudio, this value is stated under each model summary as $\Pr(>|t|)$, where t is student t-test statistic given by the following equation for n observations in a sample…

$$t = \frac{\bar{x} - \mu}{\sigma/\sqrt{n}}$$

$\bar{x}$ is the sample mean of n observations, μ is the population mean of all observations N in X, and σ is the standard deviation of the field. Respectively, these are all given by…

$$\bar{x} = \frac{\sum_{j=1}^{j=n} x_j}{n}$$

$$\mu = \frac{\sum_{j=1}^{j=N} X_j}{N}$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^{i=n}(x_i - \mu)^2}{n}}$$

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

$$t = \frac{\hat{\beta}_1 - 0}{\sigma/\sqrt{n}}, \qquad \beta_1 = 0$$

Where $\beta_1 = 0$ applies to the model (null hypothesis H_0 is considered true for $\beta_{j=1}$), this statistic then exhibits a t-distribution with $n-2$ degrees of freedom. If the null hypothesis is shown to be false, then a solid relationship between predictors X and responses Y 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 β_j 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…

$$CI = \bar{x} \pm z\frac{\sigma}{\sqrt{n}}$$

Where z is the confidence value, predefined by the confidence interval's percentage (e.g., 95% corresponds to $z = 1.960$).

Complimentary to the p-value is the α-value, the significance level and probability of conducting a type I error (rejecting, perhaps prematurely, H_0 when it is in fact true)…

$$\alpha\% = P\left(-z\frac{\sigma}{\sqrt{n}} < \bar{X} - \mu < z\frac{\sigma}{\sqrt{n}}\right)$$

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 ϵ 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 y and a prediction of said response $\hat{y}$ given by the model…

$$err = y - \hat{y}$$

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

$$RSS = \sum_{j=1}^{j=n} \left(y_j - \hat{f}(x_j)\right)^2$$

The R^2-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)…

$$TSS = \sum_{j=1}^{j=n} (y_j - \bar{y}_j)^2$$

$$R^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{\sum_{j=1}^{j=n} \left(y_j - \hat{f}(x_j)\right)^2}{\sum_{j=1}^{j=n} (y_j - \bar{y}_j)^2}$$

where RSS and TSS may be interpreted as the modelled variation of data currently being investigated, and TSS is the overall variation. A high R^2-statistic implies that the predicted response variables $\hat{Y}$ are affected significantly by the predictors X.

For multivariate regression models it is necessary to adjust the R^2-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 R^2 is given as follows for n observations and p predictors…

$$R^2 = \frac{RSS/(n-p-1)}{TSS/(n-1)}$$

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 β_j. In linear regression this involves selecting values for β_j which minimize the value of RSS…

$$RSS = \sum_{j=1}^{j=n} \left(y_j - \beta_0 - \beta_1 x_1\right)^2$$

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

$$\ell(\beta_0, \beta_1) = \prod_{j:y_j=1} p(x_j) \prod_{j\prime:y_{j\prime}=0} \left(1 - p(x_{j\prime})\right)$$

with values for the regression coefficients necessarily maximizing the likelihood $\ell(\beta_j)$.

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

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

$$h_j = \frac{1}{n} + \frac{\left(x_j - \bar{x}\right)^2}{\sum_{j'=1}^{j'=n}\left(x_{j'} - \bar{x}\right)^2}, \quad \frac{1}{n} \leq h_j \leq 1$$

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

$$(p+1)/n$$

where p is the number of predictors used in the model.

A high-leverage point is classified where $h_j \gg (p+1)/n$.

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. Qualitative summary of the data in the Auto.csv dataset.

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 1 for America (USA), 2 for Europe, and 3 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.

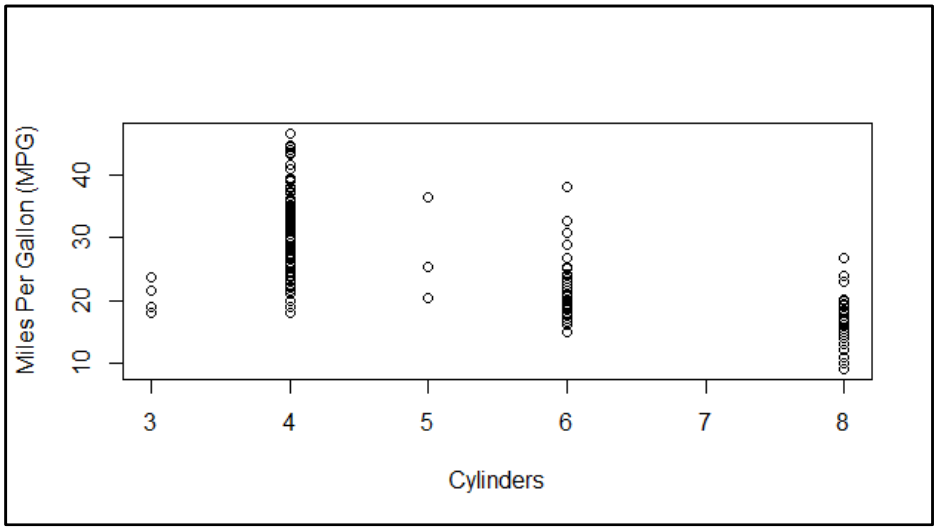

Fig. 1. Scatterplot of miles per gallon against cylinder count.

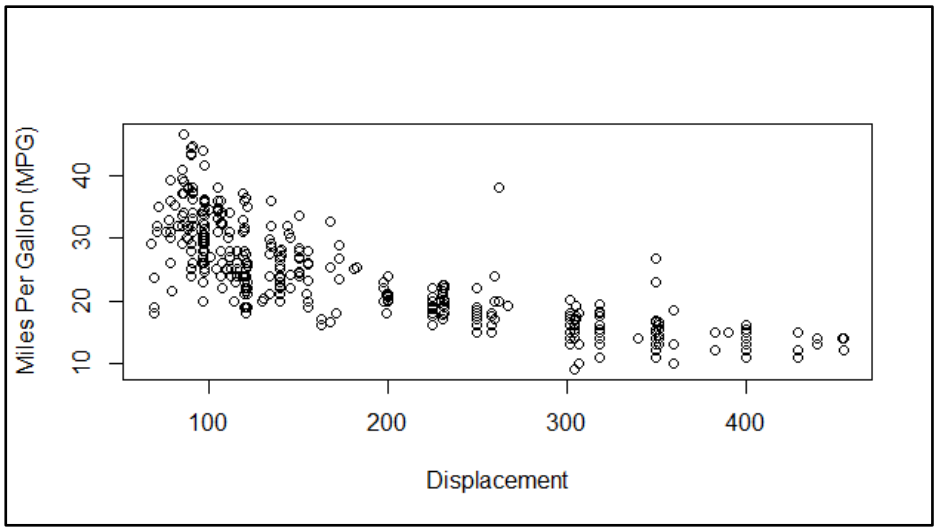

Fig. 2. Scatterplot of miles per gallon against engine displacement

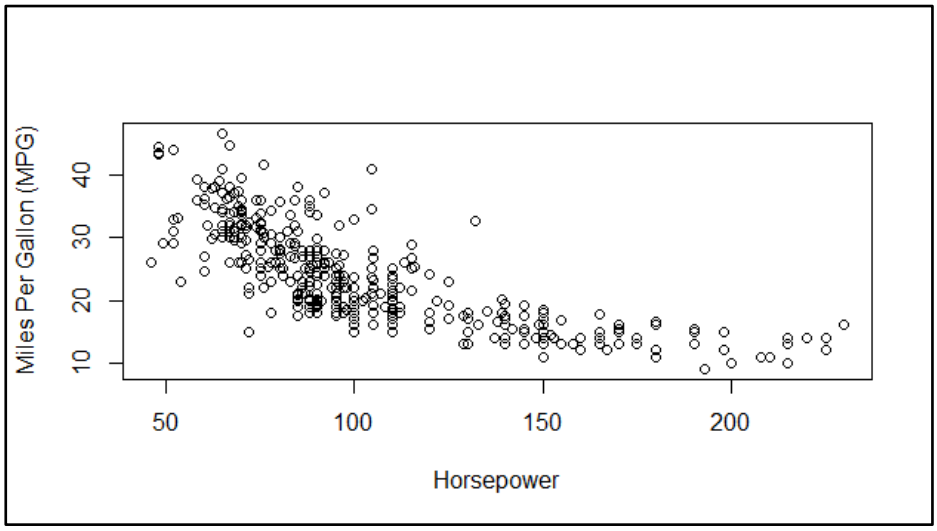

Fig. 3. Scatterplot of miles per gallon against engine horsepower.

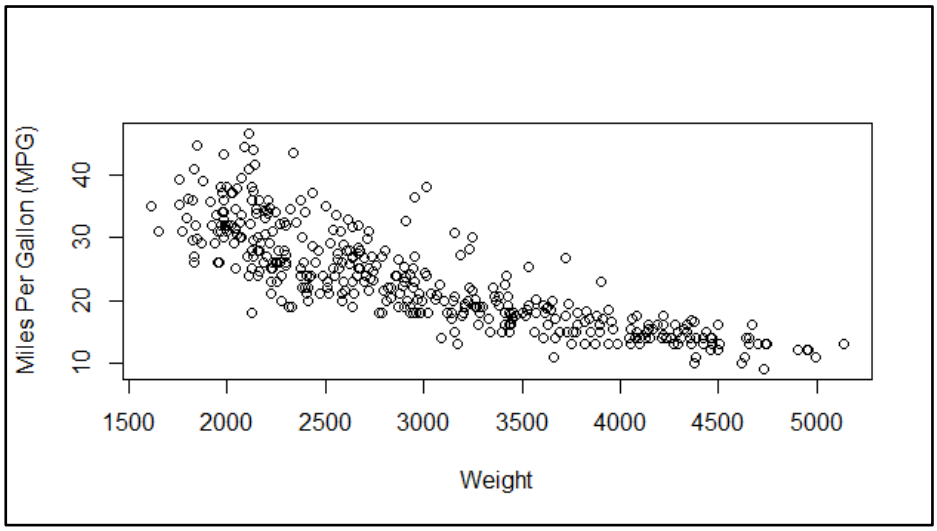

Fig. 4. Scatterplot of miles per gallon against weight of automobile.

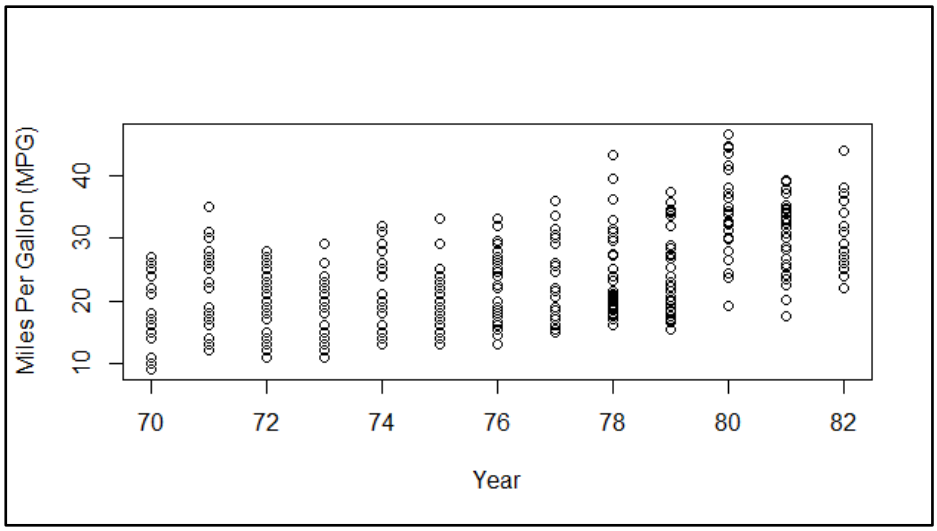

Fig. 5. Scatterplot of miles per gallon against year of manufacture.

E. Walker [SN: 3368 6408]
IS71104A Statistics and Statistical Data Mining
Goldsmiths, University of London
2022

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

	mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin
mpg	1.0000000	-0.7760493	-0.8043357	-0.7713031	-0.8315847	0.4241588	0.5809497	0.5655894
cylinders	-0.7760493	1.0000000	0.9509660	0.8396285	0.8969446	-0.5059866	-0.3448983	-0.5674199
displacement	-0.8043357	0.9509660	1.0000000	0.8937417	0.9330591	-0.5455283	-0.3688851	-0.6123209
horsepower	-0.7713031	0.8396285	0.8937417	1.0000000	0.8604679	-0.6885330	-0.4122887	-0.4553955
weight	-0.8315847	0.8969446	0.9330591	0.8604679	1.0000000	-0.4209890	-0.3066136	-0.5830785
acceleration	0.4241588	-0.5059866	-0.5455283	-0.6885330	-0.4209890	1.0000000	0.2853958	0.2095848
year	0.5809497	-0.3448983	-0.3688851	-0.4122887	-0.3066136	0.2853958	1.0000000	0.1873293
origin	0.5655894	-0.5674199	-0.6123209	-0.4553955	-0.5830785	0.2095848	0.1873293	1.0000000

Fig. 6. Corelation matrix of the Auto.csv dataset. As expected, the main diagonal of the matrix corresponds to correlation measurements between the same variable, yielding 100% correlation.

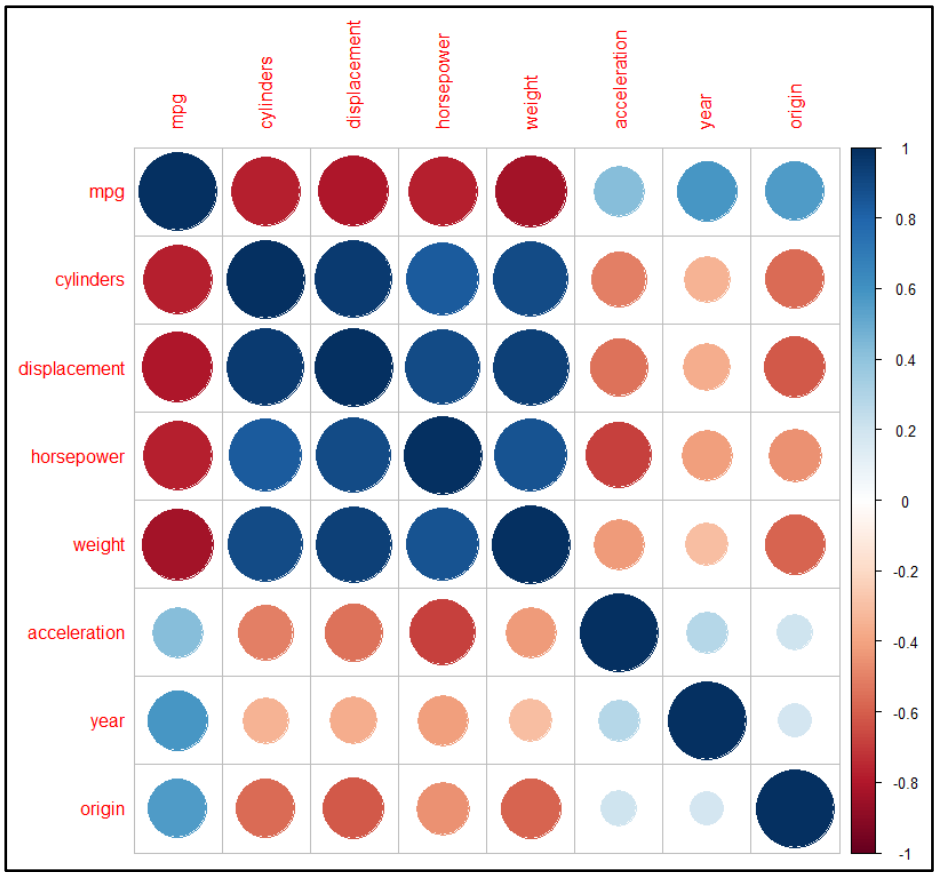

Fig. 7. Visual representation of the correlation matrix in Figure 6. Magnitude of correlation is presented in circle radius, whereas the sign is presented as either a shade of blue for *positive* correlation, and a shade of red for *negative* correlation.

The 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).

```
> boxplot.stats(numeric_data$horsepower)$out
 [1] 220 215 225 225 215 200 210 208 215 225 230
> boxplot.stats(numeric_data$acceleration)$out
 [1]  8.5  8.5  8.0 23.5 22.2 22.1 24.8 22.2 23.7 24.6
```

Fig. 8. Summary of outliers contained in the horsepower and acceleration columns of the Auto.csv dataset.

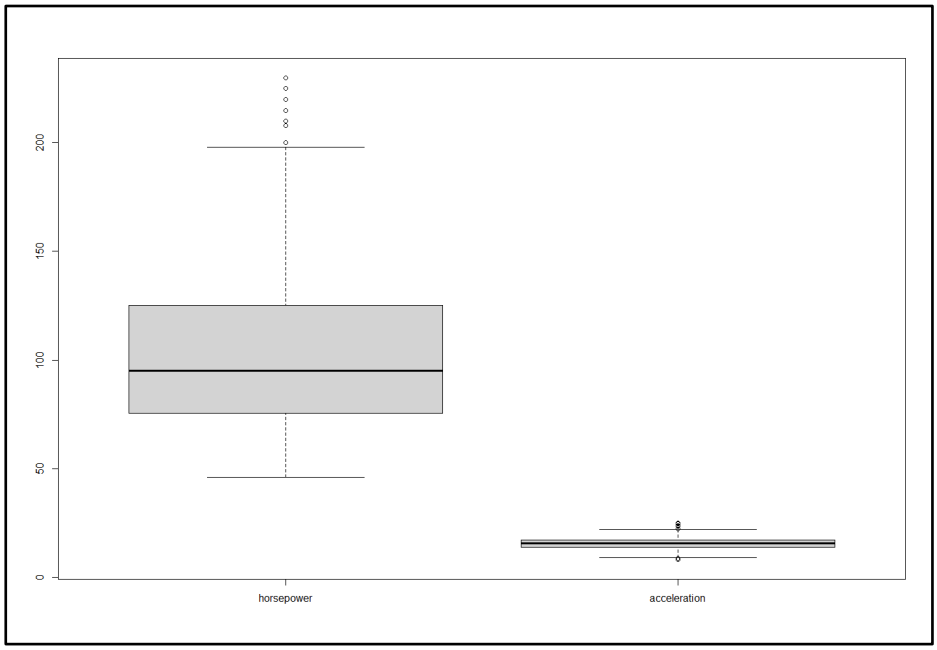

Fig. 9. Box-whisker plot of the horsepower and acceleration fields of the Auto.csv dataset. Visible as dots are the outliers.

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…

$$\hat{f}_{L1}(X) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_3 + \hat{\beta}_4 x_4 + \hat{\beta}_5 x_5 + \hat{\beta}_6 x_6 + \hat{\beta}_7 x_7$$

or

$$\hat{f}_{L1}(X) = \hat{\beta}_0 + \hat{\beta}_1(cylinders) + \hat{\beta}_2(displacement) + \hat{\beta}_3(horsepower) + \hat{\beta}_4(weight) + \hat{\beta}_5(acceleration) + \hat{\beta}_6(year) + \hat{\beta}_7(origin)$$

where

Table 2. Overview of the significance and estimations of the coefficients fitted for linear model 1.

Coef.	Estimate	Predictor	Significance
$\hat{\beta}_0$	6.4423528		***
$\hat{\beta}_1$	-0.2981054	Cylinders	
$\hat{\beta}_2$	-0.0008087	Displacement	
$\hat{\beta}_3$	-0.0303845	Horsepower	***
$\hat{\beta}_4$	-0.0037397	Weight	
$\hat{\beta}_5$	-0.2928924	Acceleration	
$\hat{\beta}_6$	0.4680871	Year	
$\hat{\beta}_7$	0.8049288	Origin	

As table 2 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 $\hat{\beta}_6$ for the year

predictor suggests that fuel economy increases by approximately $0.47\ miles\ per\ gallon$ for every one year of time. Leverage 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…

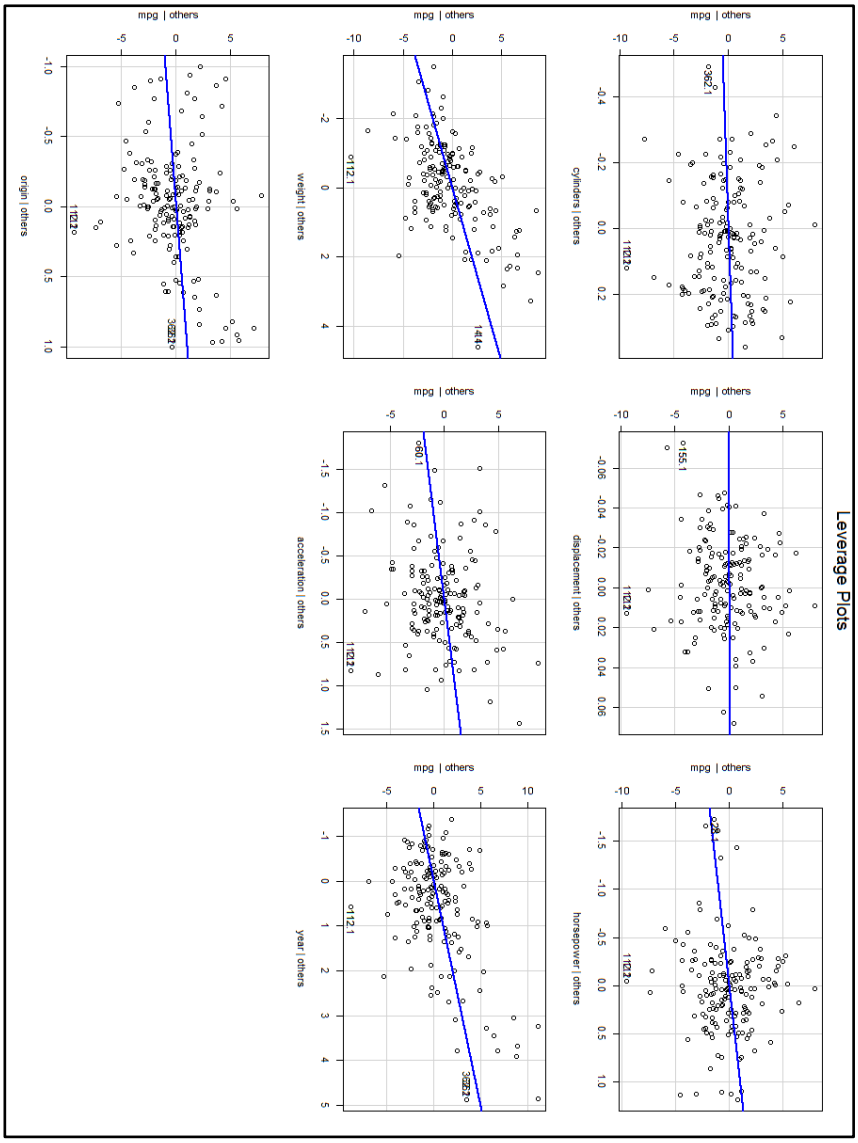

Fig. 10. Leverage plots for each predictor in linear model 1 against the fitted values of the model for MPG.

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 0.05. Model 1 possessed an outlier at row 112, where model 2 possessed one at row 305…

```
> outlierTest(linear_model1, cutoff = 0.05)
      rstudent unadjusted p-value Bonferroni p
112   -4.631196         3.8356e-06    0.0089752
112.1 -4.631196         3.8356e-06    0.0089752
112.2 -4.631196         3.8356e-06    0.0089752
112.3 -4.631196         3.8356e-06    0.0089752
112.4 -4.631196         3.8356e-06    0.0089752
> outlierTest(linear_model2, cutoff = 0.05)
No Studentized residuals with Bonferroni p < 0.05
Largest |rstudent|:
    rstudent unadjusted p-value Bonferroni p
305 3.728128         0.00019749      0.46213
```

Fig. 11. Outlier tests for linear models 1 and 2, with a Bonferroni p-value threshold of < 0.05.

Predictions 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…

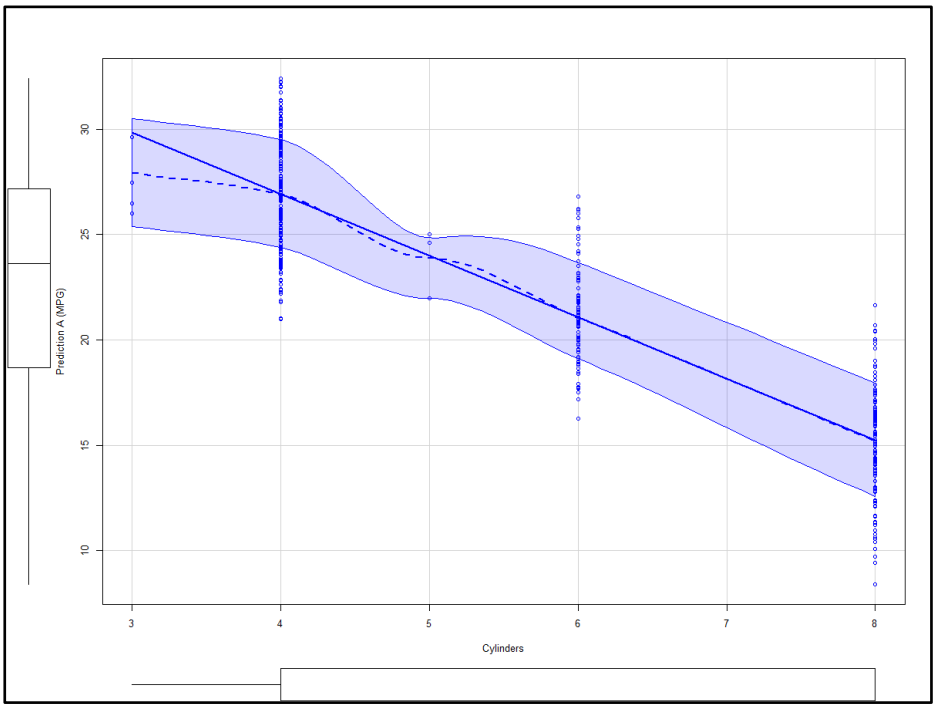

Fig. 12. Detailed scatterplot of the first model's prediction for MPG against cylinder count. Featured is a solid blue line fitted to the data.

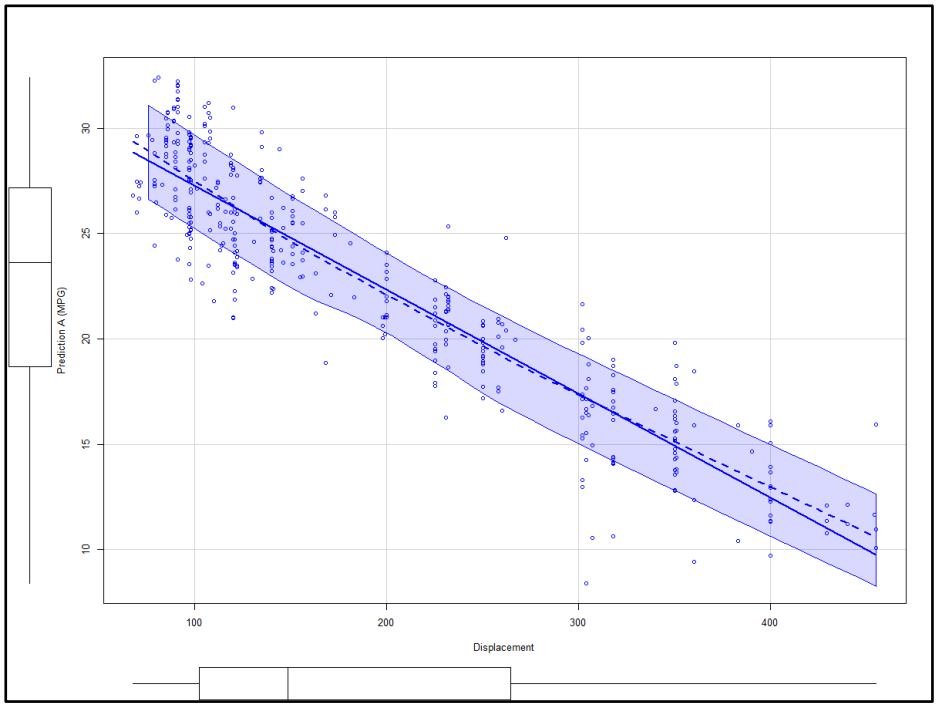

Fig. 13. Detailed scatterplot of the first model's prediction for MPG against engine displacement.

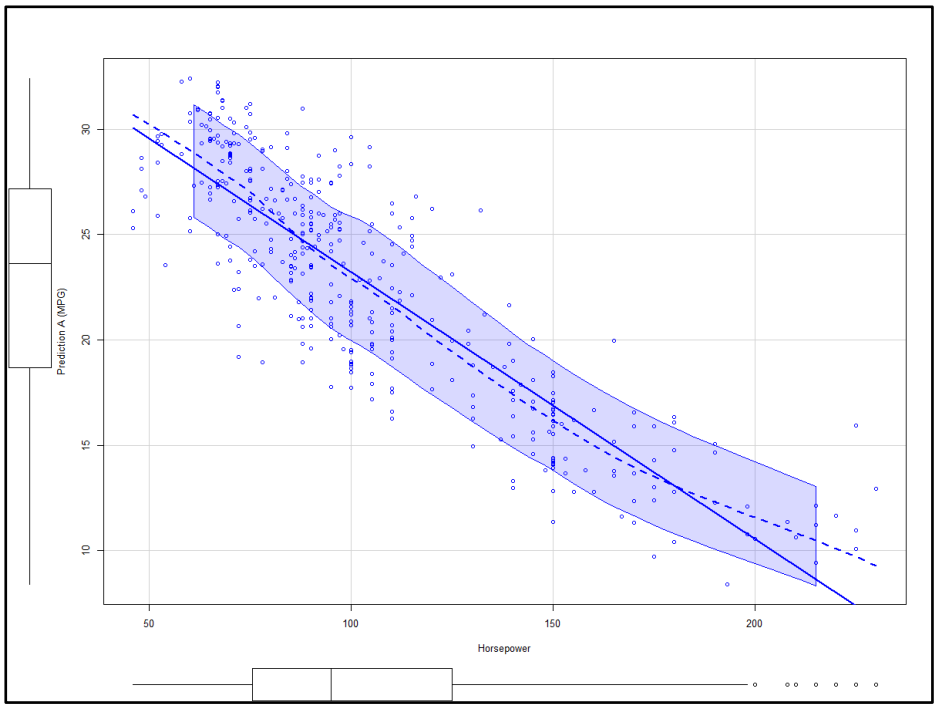

Fig. 14. Detailed scatterplot of the first model's prediction for MPG against horsepower.

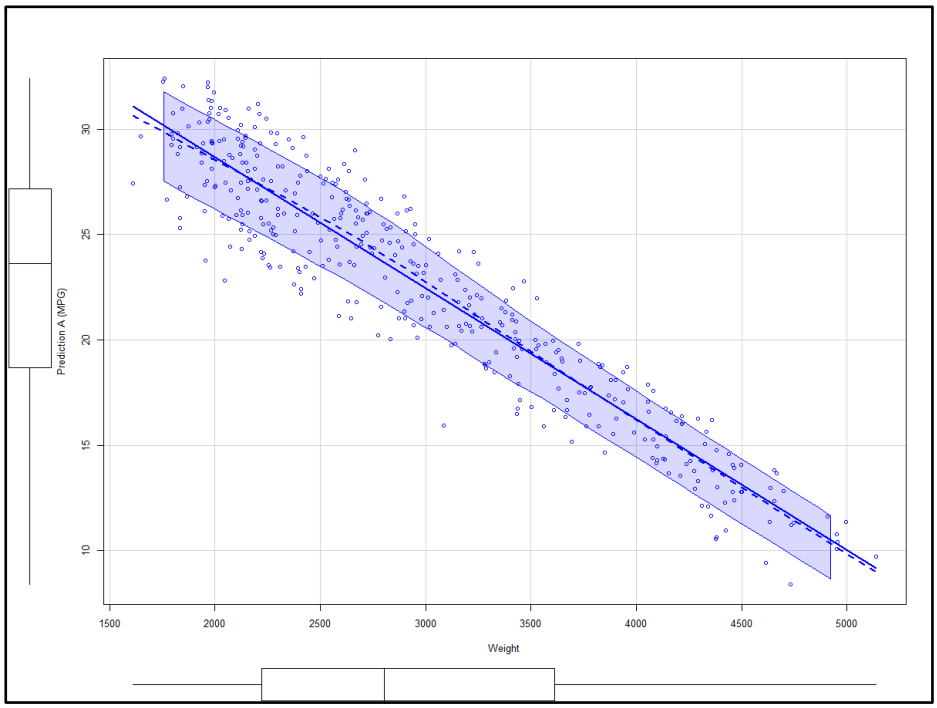

Fig. 15. Detailed scatterplot of the first model's prediction for MPG against automobile weight.

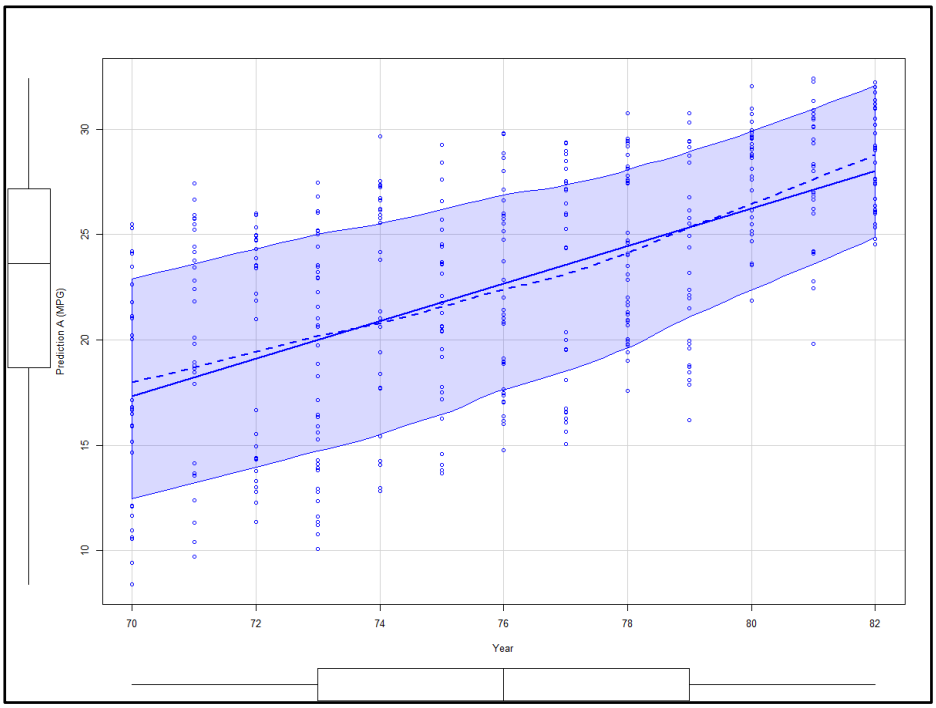

Fig. 16. Detailed scatterplot of the first model's prediction of MPG against the year of manufacture.

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. Overview for each interaction effect utilizing the colon (:) operator in linear model 1.

Interaction	Significance
$cylinders : horsepower$	***
$horsepower : weight$	
$weight : acceleration$	
$cylinders : acceleration$	
$cylinders : weight$	
$acceleration : year$	**
$acceleration : origin$	***

The interaction $acceleration : origin$ rendered the cylinders field statistically insignificant in model 1.

Applying transforms to all predictors in model 1…

Table 4. Qualitative overview of how different transforms affect linear model 1, when applied to each predictor.

Transform	Effect
$X ** 2$	Little to none on significance of predictors.
$sqrt(X)$	Shifted y-intercept downwards along the negative y-axis, and rendered cylinders mildly statistically significant (*).
$\log(X)$	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…

$$\hat{f}_{L2}(X) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2$$

or

$$\hat{f}_{L2}(X) = \hat{\beta}_0 + \hat{\beta}_1 (weight) + \hat{\beta}_2 (year)$$

where

Table 5. Overview for the significance and estimated coefficients for linear model 2.

Coef.	Estimate	Predictor	Significance
$\hat{\beta}_0$	-1.6490000		
$\hat{\beta}_1$	-0.0056090	Weight	***
$\hat{\beta}_2$	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 R^2-statistic for this model is…

$$R^2_{L2} = 0.8214\ (5\ s.f.)$$

whereas the adjusted R^2-statistic for model 1 was…

$$R^2_{L1} = 0.8453\ (5\ s.f.)$$

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…

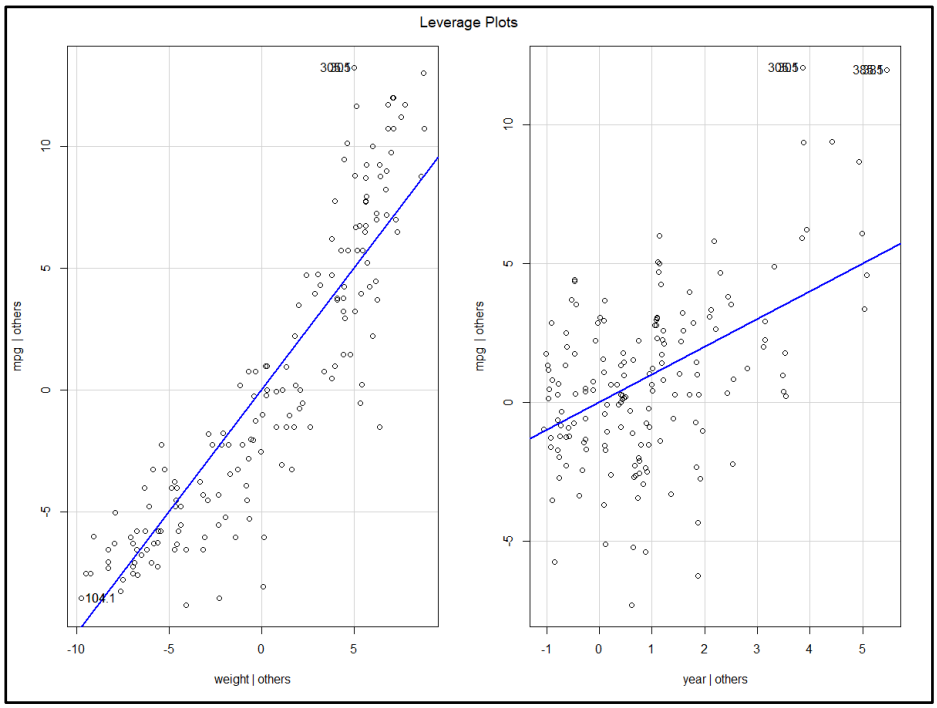

Fig. 17. Leverage plot for the predictors of model 2. High-leverage points for each field are labelled with the row numbers at which they occur in the training dataset.

Scatterplots are provided for model 2 ahead…

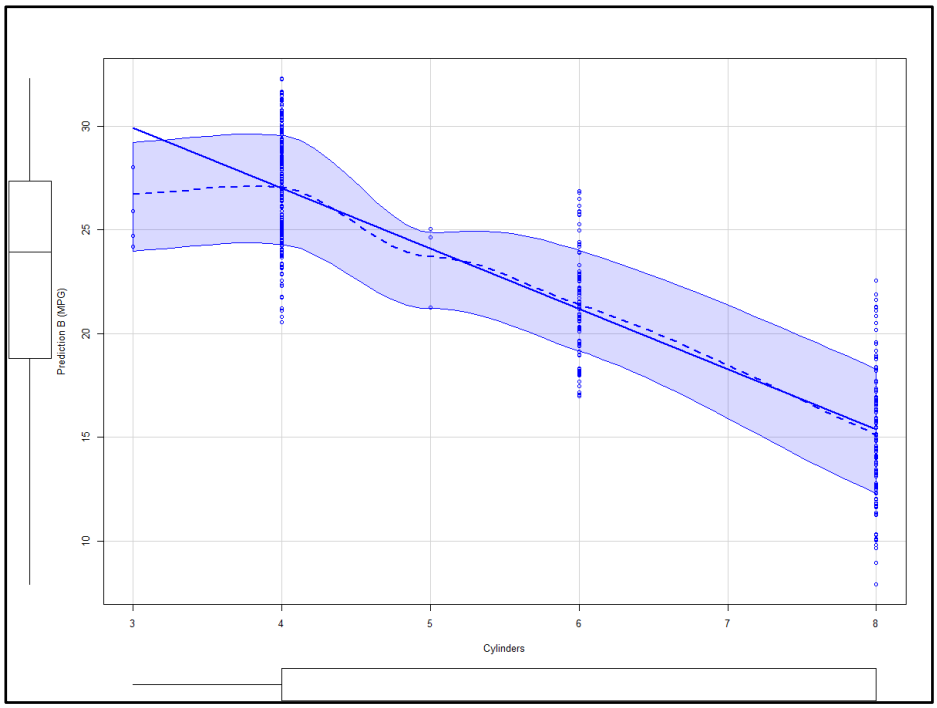

Fig. 18. Detailed scatterplot of the second model's prediction for MPG against cylinder count.

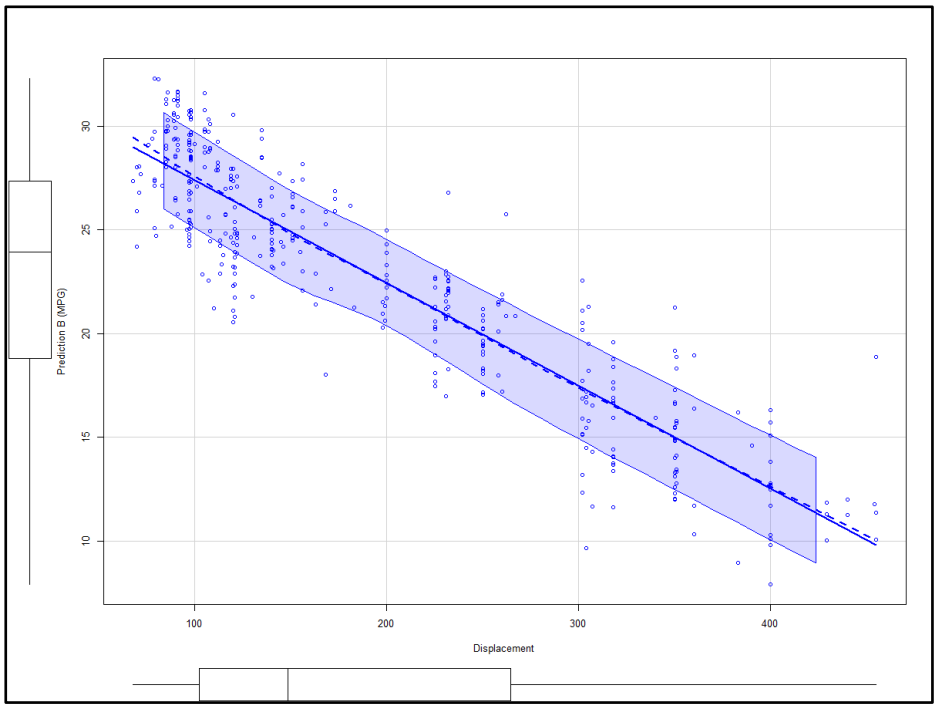

Fig. 19. Detailed scatterplot of the second model's prediction for MPG against engine displacement.

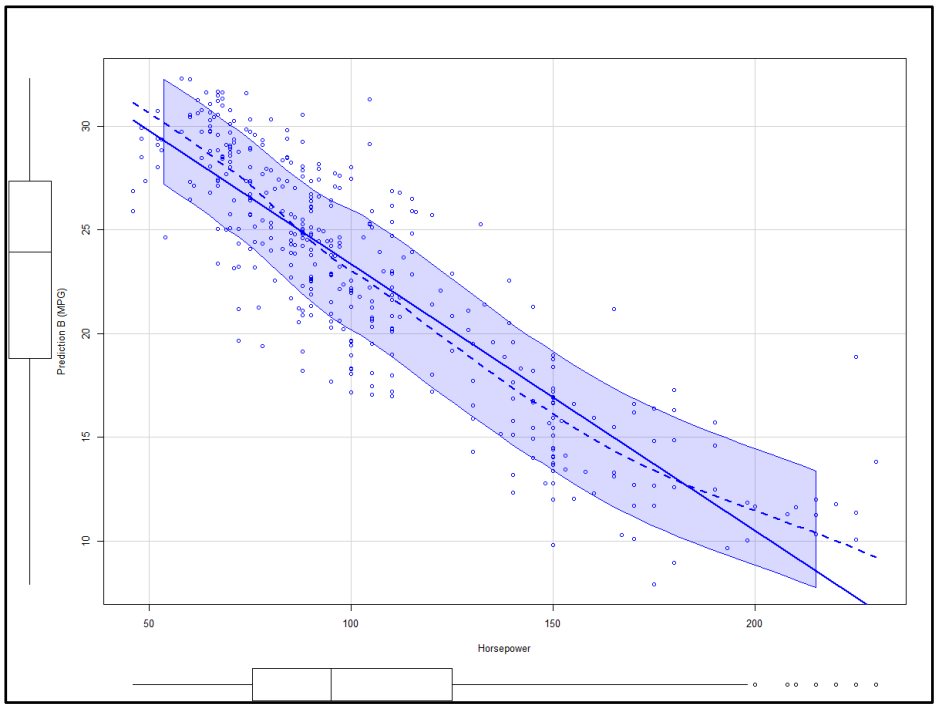

Fig. 20. Detailed scatterplot for the second model's prediction for MPG against horsepower.

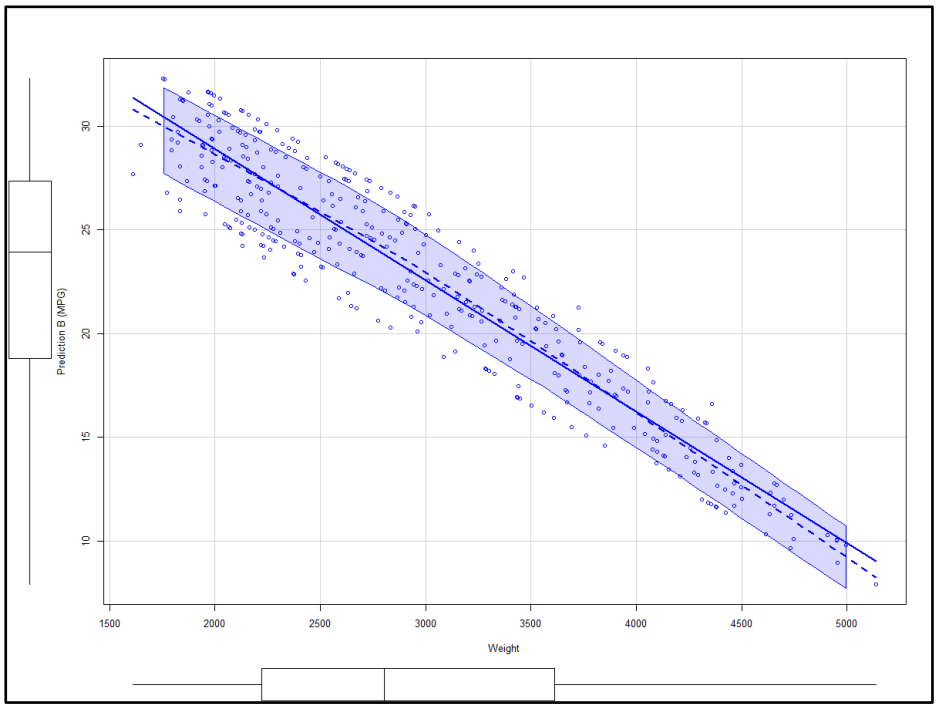

Fig. 21. Detailed scatterplot for the second model's prediction for MPG against automobile weight.

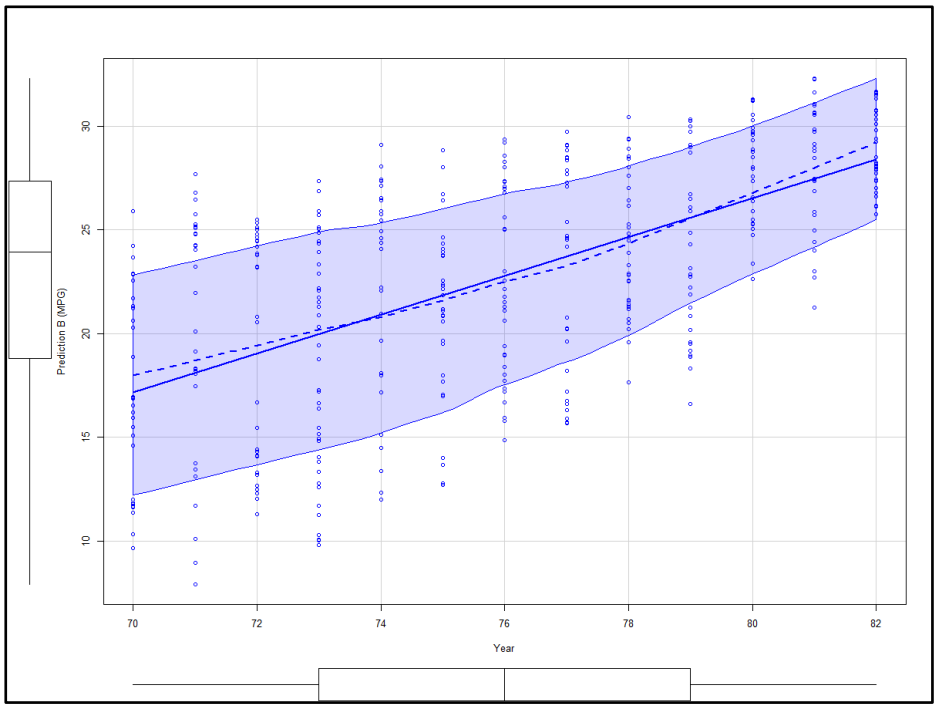

Fig. 22. Detailed scatterplot for the second model's prediction for MPG against year of manufacture.

Interaction effects for model 2…

Table 6. Interaction effects for the second model utilizing the colon (:) operator.

Interaction	Significance
$weight : year$	***
$weight * year$	

Transforms…

Table 7. Transforms applied to the predictors X of the second model and their corresponding effect on the model's coefficients.

Transform	Effect
$X ** 2$	Little to none on significance of predictors.
$sqrt(X)$	Increases significance of y-intercept (***).
$\log(X)$	Increases significance of y-intercept (**).

3.2 Task 2 of 3

Task 2 also focused on the creation of 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…

Table 8. Qualitative overview of the Carseats.csv dataset.

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. Encoding for categorical variables in the Carseats.csv dataset.

Field	Encoding
ShelveLoc	$Bad = 001$ $Medium = 100$ $Good = 010$
Urban	$No = 0$ $Yes = 1$
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…

$$\hat{f}_{L1}(X) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_3 + \hat{\beta}_4 x_4 + \hat{\beta}_5 x_5 + \hat{\beta}_6 x_6 + \hat{\beta}_7 x_7 + \hat{\beta}_8 x_8 + \hat{\beta}_9 x_9 + \hat{\beta}_{10} D_1 + \hat{\beta}_{11} D_2$$

or

$$\hat{f}_{L1}(X) = \hat{\beta}_0 + \hat{\beta}_1(CompPrice) + \hat{\beta}_2(Income) + \hat{\beta}_3(Advertising) + \hat{\beta}_4(Population) + \hat{\beta}_5(Price) + \hat{\beta}_6(Age) + \hat{\beta}_7(Education) + \hat{\beta}_8(Urban) + \hat{\beta}_9(US) + \hat{\beta}_{10}(Bad\_ShelveLoc) + \hat{\beta}_{11}(Good\_Shelveloc)$$

where

Table 10. Overview of the first linear model's coefficients.

Coef.	Estimate	Predictor	Significance
$\hat{\beta}_0$	2.7748873		
$\hat{\beta}_1$	0.1139781	CompPrice	***
$\hat{\beta}_2$	0.0216911	Income	
$\hat{\beta}_3$	0.1717555	Advertising	
$\hat{\beta}_4$	0.0004057	Population	**
$\hat{\beta}_5$	-0.0937464	Price	
$\hat{\beta}_6$	-0.0529729	Age	
$\hat{\beta}_7$	0.1033912	Education	***
$\hat{\beta}_8$	0.2192470	Urban	
$\hat{\beta}_9$	-0.3266892	US	
$\hat{\beta}_{10}$	-1.4524399	Bad_ShelveLoc	
$\hat{\beta}_{11}$	2.5624995	Good_ShelveLoc	

Model 1, comprising all predictors, possessed an adjusted R^2-statistic of

$$R^2_{L1} = 0.8684\ (5\ s.f.)$$

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 $\hat{\beta}_{10}$, 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 H_0, 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 0 and contributed to a rejection of the null hypothesis.

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

$$\hat{f}_{L2}(X) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_3 + \hat{\beta}_4 x_4$$

or

$$\hat{f}_{L2}(X) = \hat{\beta}_0 + \hat{\beta}_1(CompPrice) + \hat{\beta}_2(Income) + \hat{\beta}_3(Advertising) + \hat{\beta}_4(Price)$$

where

Table 11. Overview of the second linear model's coefficients.

Coef.	Estimate	Predictor	Significance
$\hat{\beta}_0$	4.189860		***
$\hat{\beta}_1$	0.095250	CompPrice	
$\hat{\beta}_2$	0.008586	Income	
$\hat{\beta}_3$	0.205240	Advertising	
$\hat{\beta}_4$	-0.089672	Price	

and possessed an adjusted R^2-statistic of…

$$R^2_{L2} = 0.6220\ (5\ s.f.)$$

The null hypothesis, H_0, 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…

$$\hat{f}_{L3}(X) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_3$$

or

$$\hat{f}_{L3}(X) = \hat{\beta}_0 + \hat{\beta}_1(Price) + \hat{\beta}_2(Urban) + \hat{\beta}_3(US)$$

where

Table 12. Overview of the third linear model's coefficients.

Coef.	Estimate	Predictor	Significance
$\hat{\beta}_0$	12.799855		***
$\hat{\beta}_1$	-0.056720	Price	
$\hat{\beta}_2$	0.473567	Urban	
$\hat{\beta}_3$	2.070775	US	

Here, all predictors could reject the null hypothesis

$$H_0 : \beta_j = 0$$

"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

$$\hat{f}_{L4}(X) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2$$

or

$$\hat{f}_{L4}(X) = \hat{\beta}_0 + \hat{\beta}_1 (Urban) + \hat{\beta}_2 (US)$$

where

Table 13. Overview of the forth model's coefficients.

Coef.	Estimate	Predictor	Significance
$\hat{\beta}_0$	6.06795		***
$\hat{\beta}_1$	0.68952	Urban	
$\hat{\beta}_2$	2.48414	US	

The 95% confidence interval for model 4 was calculated via a single sample t-test, and was determined to be…

$$CI_{L4} = [7.696203, \quad 8.043564]$$

Although, this model is far less befitting of the data than the previous model, as indicated by their respective R^2-statistics…

$$R^2_{L3} = 0.4578 \ (5 \ s.f.)$$

$$R^2_{L4} = 0.1807 \ (5 \ s.f.)$$

making model 4 significantly underfitted.

Model 4 contained four outliers, and the following high-leverage observations…

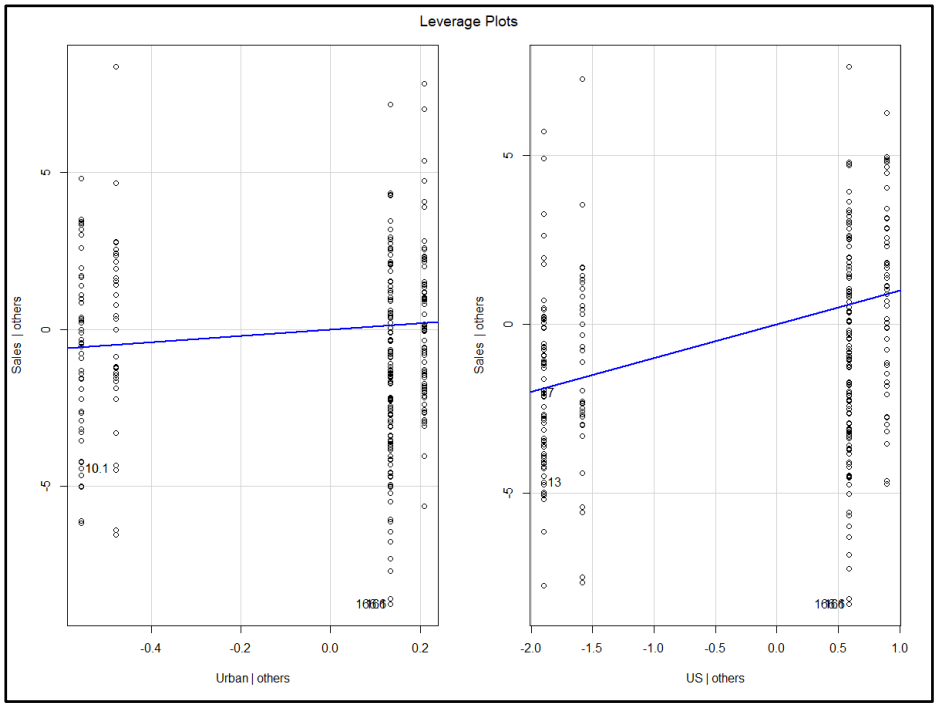

Fig. 23. Leverage plots for model 4. High-leverage observations are visible in both plots for Urban and US fields, primarily occurring under the fitted line for each against the model's prediction for sales.

Outliers for model 4 are indicated in the following boxplots…

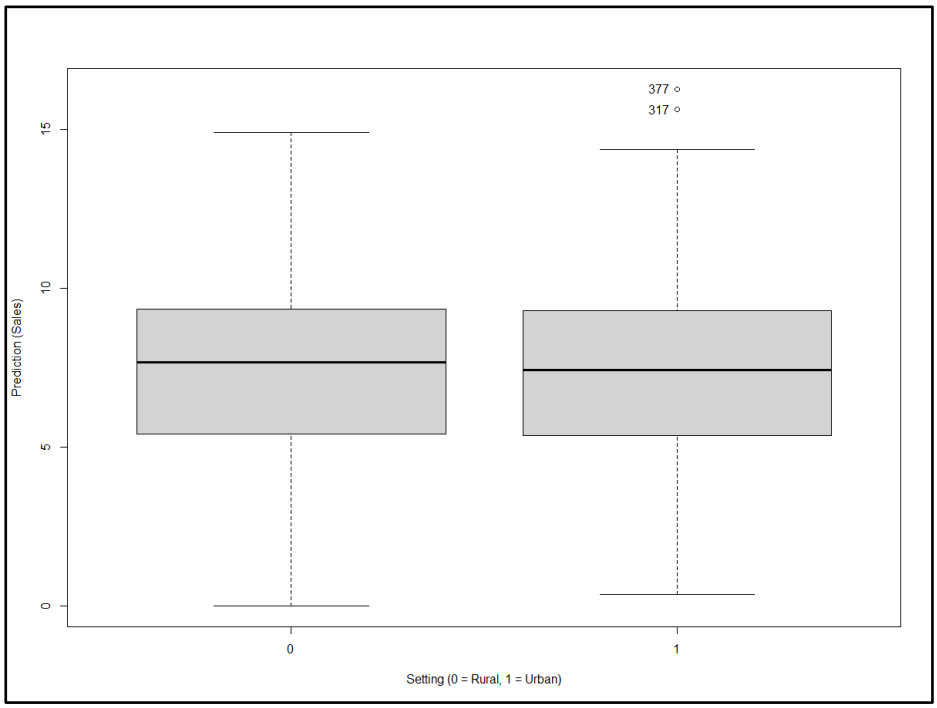

Fig. 24. The setting (Urban) predictor for model 4 exhibits two outliers in circumstances where $Urban = 1$ (in an urban setting). Observations 317 and 377 occur as outliers when plotted against the model's prediction for sales.

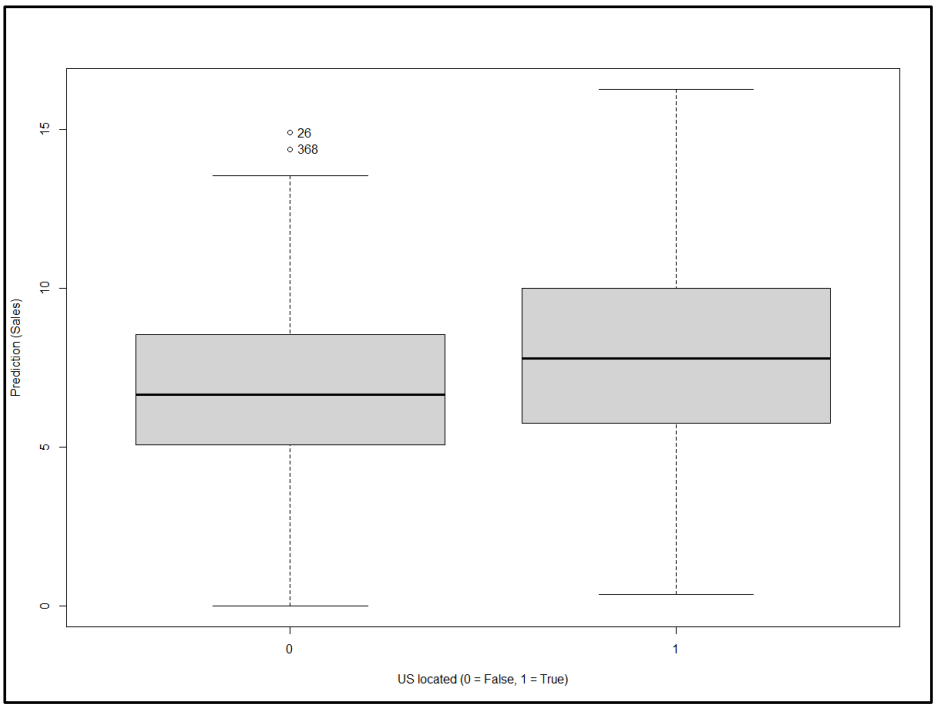

Fig. 25. In the context of linear model 4, outliers occur for sales predictions outside of the United States. Observations 26 and 368 are considered outliers.

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. Qualitative overview of the Weekly.csv dataset as per information provided by [3].

Field Name	Outline
Direction	Direction of stock market. This may take a value of either "Down" or "Up", which was encoded, respectively, as 0 and 1 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 0 and 1.

Table 15. Summary of continuous variables from Weekly.csv.

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. Summary of the encoded Direction variable in Weekly.csv.

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.

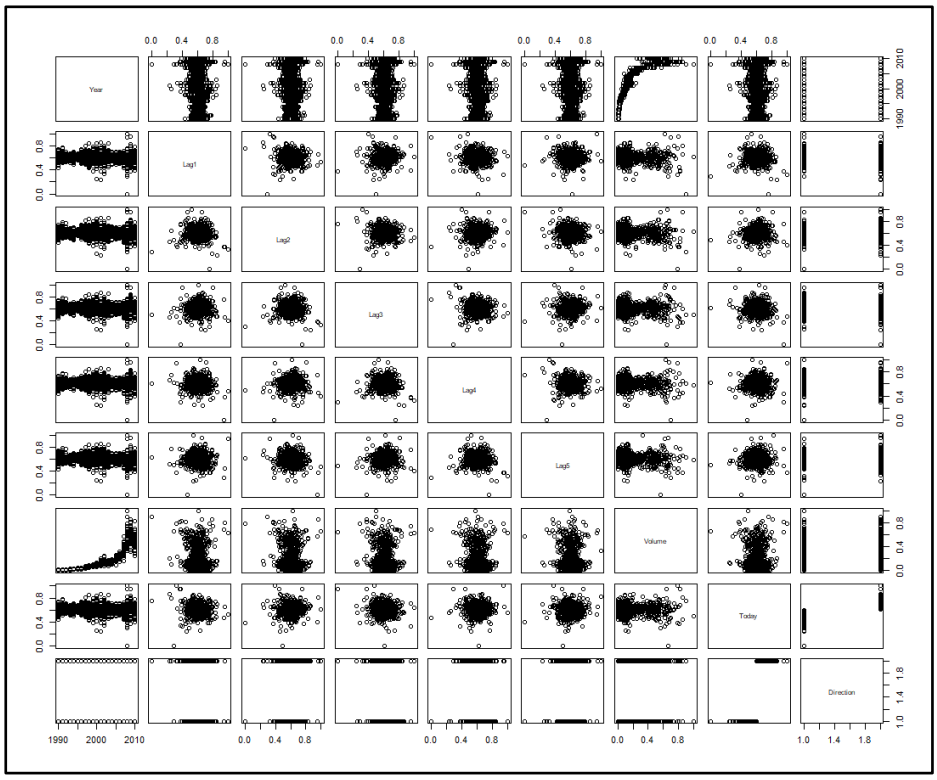

Fig. 26. Scatterplot matrix for the Weekly.csv dataset.

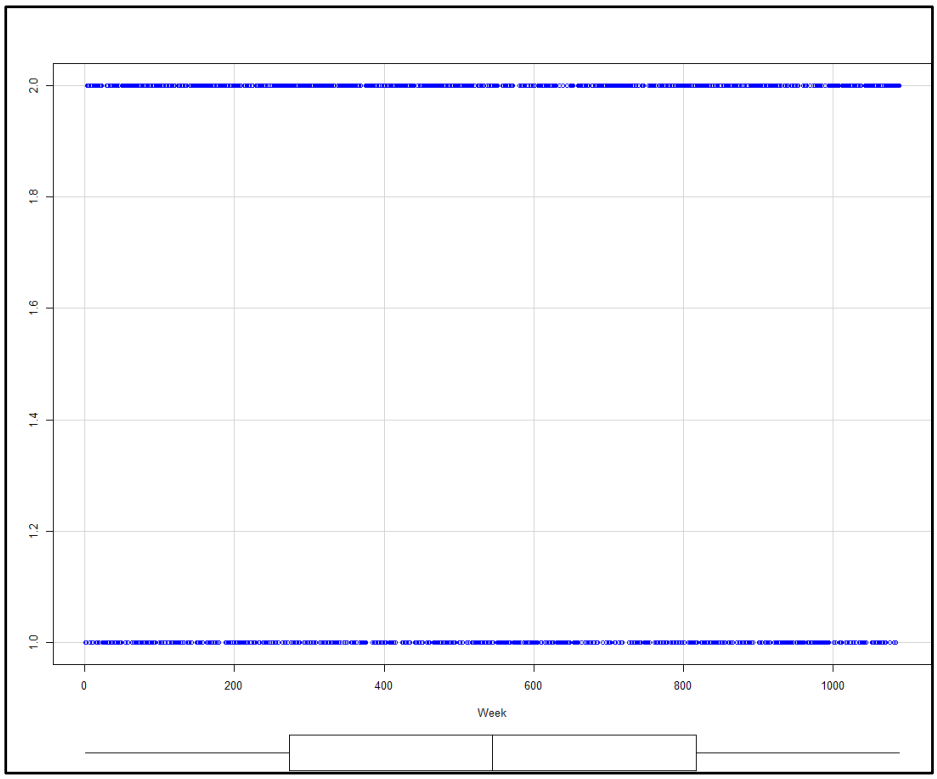

Fig. 27. Scatterplot for Direction per week in Weekly.csv dataset.

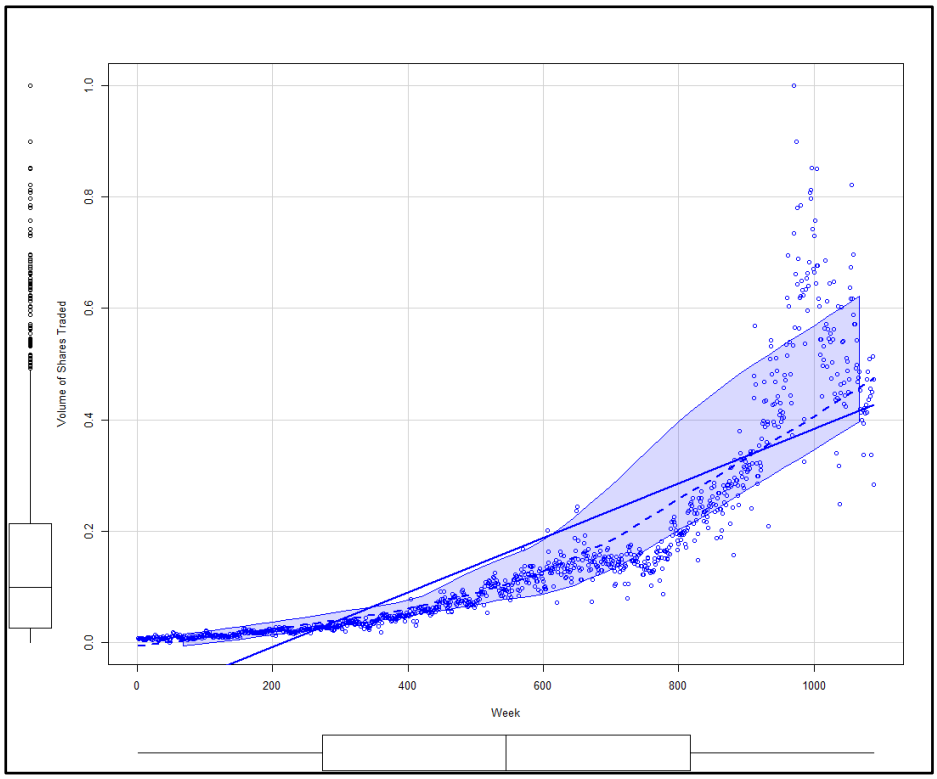

Fig. 28. Detailed scatterplot for sales per week.

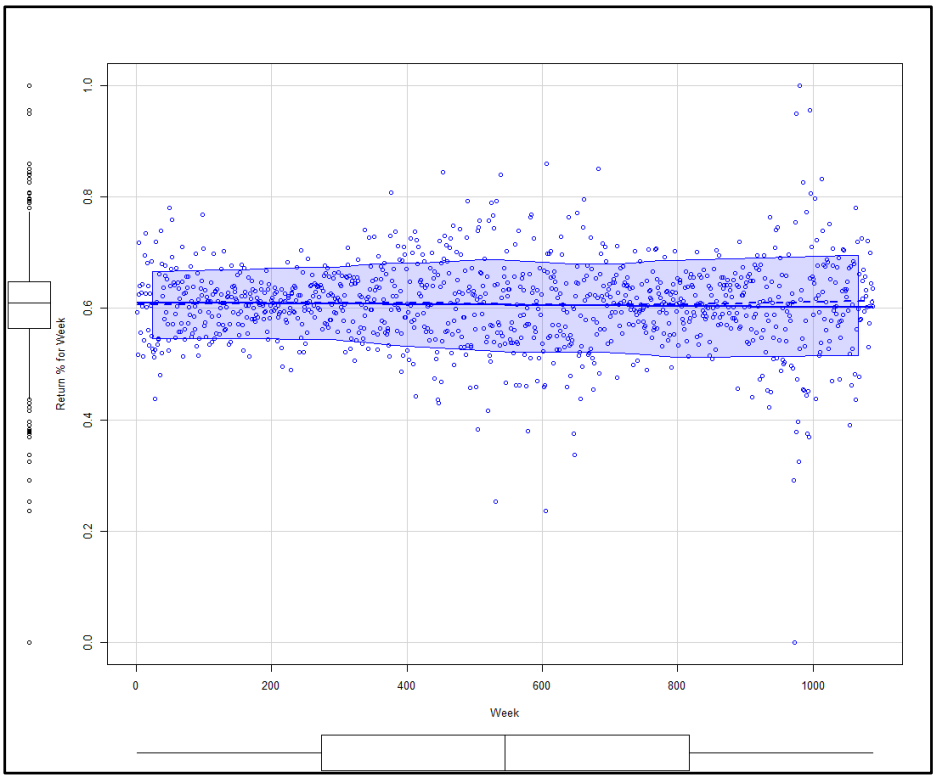

Fig. 29. Detailed scatterplot for weekly return (Today) per week.

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 X predictors, the model is expressed as follows…

$$p_1(Y = y|X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1x_1+\hat{\beta}_2x_2+\hat{\beta}_3x_3+\hat{\beta}_4x_4+\hat{\beta}_5x_5+\hat{\beta}_6x_6}}{1 + e^{\hat{\beta}_0+\hat{\beta}_1x_1+\hat{\beta}_2x_2+\hat{\beta}_3x_3+\hat{\beta}_4x_4+\hat{\beta}_5x_5+\hat{\beta}_6x_6}}$$

or

$$p_1(Direction = 0|X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1(Lag1)+\hat{\beta}_2(Lag2)+\hat{\beta}_3(Lag3)+\hat{\beta}_4(Lag4)+\hat{\beta}_5(Lag5)+\hat{\beta}_6(Volume)}}{1 + e^{\hat{\beta}_0+\hat{\beta}_1(Lag1)+\hat{\beta}_2(Lag2)+\hat{\beta}_3(Lag3)+\hat{\beta}_4(Lag4)+\hat{\beta}_5(Lag5)+\hat{\beta}_6(Volume)}}$$

where

Table 17. Summary of the first logistic regression model's coefficients.

Coef.	Estimate	Predictor	Significance
$\hat{\beta}_0$	0.2392		
$\hat{\beta}_1$	-1.1206	Lag1	
$\hat{\beta}_2$	2.4787	Lag2	**
$\hat{\beta}_3$	-0.3895	Lag3	
$\hat{\beta}_4$	-1.5283	Lag4	.
$\hat{\beta}_5$	0.5541	Lag5	
$\hat{\beta}_6$	-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 56%, 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. Confusion matrix for the fitted values of model 1 when viewed against the actual values provided in its training partition.

	Fitted	
Actual	0 (down)	1 (up)
0 (down)	64	299
1 (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 54.04%…

Table 19. Confusion matrix for the predicted values of model 1 when viewed against the actual values provided in its test partition.

	Prediction	
Actual	0 (down)	1 (up)
0 (down)	13	108
1 (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…

$$p_2(Y = y|X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 x_1}}{1 + e^{\hat{\beta}_0+\hat{\beta}_1 x_1}}$$

or

$$p_2(Direction = 0|X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 (Lag2)}}{1 + e^{\hat{\beta}_0+\hat{\beta}_1 (Lag2)}}$$

where

Table 20. Summary of coefficients in the second iteration of logistic regression model.

Coef.	Estimate	Predictor	Significance
$\hat{\beta}_0$	-0.7978		
$\hat{\beta}_1$	1.6679	Lag2	.

Model 2 did not fare any better in terms of prediction accuracy. It yielded 55.35% prediction accuracy for the fitted values, and a meagerly lesser 54.07% for the predicted values in the test set…

Table 21. Confusion matrix of the second model's fitted values for stock market direction against the actual values provided in its training set (for observations between the years 1990 and 2008 only).

	Fitted	
Actual	0 (down)	1 (up)
0 (down)	16	315
1 (up)	15	393

Table 22. Confusion matrix for the second model's prediction for stock market direction versus the test set's actual values.

	Prediction	
Actual	0 (down)	1 (up)
0 (down)	2	108
1 (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. Summary of coefficients for the third logistic regression model, which considers observations between 2009 and 2010 only.

Coef.	Estimate	Predictor	Significance
$\hat{\beta}_0$	-0.6875		
$\hat{\beta}_1$	1.7173	Lag2	

with the confusion matrixes for fitted and predicted values possessing 57.69% and 65.38% prediction accuracies, respectively…

Table 24. Confusion matrix measuring the synchrony between the third model's fitted values for stock market direction and the actual values in its training set.

	Fitted	
Actual	0 (down)	1 (up)
0 (down)	0	32
1 (up)	1	45

Table 25. Confusion matrix for the third model's predictions for market direction compared to those of its corresponding test set.

	Prediction	
Actual	0 (down)	1 (up)
0 (down)	2	9
1 (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. Summary of how different transforms applied to the lone Lag2 predictor in the logistic regression model affect the accuracies of both fitted values and predictions for stock market direction.

Transform	Fitted Accuracy (%)	Prediction Accuracy (%)
$sqrt(Lag2)$	56.79	53.68
$Lag2 ** 2$	56.67	53.68

Table 27. Confusion matrix for the fitted values of the model incorporating the square root of the Lag2 predictor.

	Fitted - $sqrt(Lag2)$	
Actual	0 (down)	1 (up)
0 (down)	35	328
1 (up)	25	429

Table 28. Confusion matrix for the predicted values of the model incorporating the square root of the Lag2 predictor.

	Prediction - $sqrt(Lag2)$	
Actual	0 (down)	1 (up)
0 (down)	5	116
1 (up)	10	141

Table 29. Confusion matrix for the fitted values of the model incorporating the square of the Lag2 predictor.

	Fitted - $Lag2 ** 2$	
Actual	0 (down)	1 (up)
0 (down)	37	326
1 (up)	28	426

Table 30. Confusion matrix for the predicted values of the model incorporating the square of the Lag2 predictor.

	Prediction - $Lag2 ** 2$	
Actual	0 (down)	1 (up)
0 (down)	7	114
1 (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 $k = \{1, 3, 5, 7\}$, 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 k…

Table 31. Summary of prediction accuracy of first KNN model for $k = \{1, 3, 5, 7\}$.

All Lagx + Volume as Predictors	
Neighbors	**Prediction Accuracy (%)**
$k = 1$	70.33
$k = 3$	73.58
$k = 5$	76.83
$k = 7$	79.27

Table 32. Summary of prediction accuracy of second KNN model for $k = \{1, 3, 5, 7\}$.

Lag2 as Predictor (1990 – 2008)	
Neighbors	**Prediction Accuracy (%)**
$k = 1$	99.59
$k = 3$	99.59
$k = 5$	1.000
$k = 7$	1.000

Table 33. Summary of prediction accuracy of third and final KNN model for $k = \{1, 3, 5, 7\}$.

Lag2 as Predictor (2009 – 2010)	
Neighbors	**Prediction Accuracy (%)**
$k = 1$	92.31
$k = 3$	88.46
$k = 5$	84.62
$k = 7$	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 $k = 7$ neighbors the model peaked at 79.27% 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 100.00% prediction accuracy was achieved for $k = \{5, 7\}$ in the second KNN model, and a 92.31% accuracy accuracy for $k = \{1, 7\}$ in the third KNN model.

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

---

4 Discussion

Throughout the studies across the three tasks which were the subject of this report, the following discoveries were made…

4.1 Task 1 of 3

Decreases in the number of predictors used for the multivariate linear regression models produced both ample model fit as indicated by the R^2-statistic, and statistical significance courtesy of the automobile's weight and year of manufacture.

Weight no doubt plays a significant role in fuel economy due to the laws of Newtonian physics: the greater the mass to move or impress an alteration in velocity, the greater the force (and hence more combustive energy via fuel) required. According to the coefficient for Weight in the second model…

$$\hat{\beta}_1 = -0.0056090$$

Every unit increase in the weight of an automobile causes a slight decrease of -0.0056090 in fuel economy. Not particularly troublesome for the data provided, but consistent enough to be significant.

Developments in technology and iterative design also play a significant role in the fuel economy of automobiles, hence the year of manufacture exhibiting a high statistical significance (and a low p-value as a result) throughout the task. This is indicated by the coefficient for Year…

$$\hat{\beta}_2 = 0.5407000$$

Every year that passes, fuel economy increases by 0.54 units according to linear model 2.

4.2 Task 2 of 3

In all cases where multivariate linear models were created to predict values for the sales of car seats based on a series of predictors, the null hypothesis $H_0: \beta_j = 0$ was firmly disproved.

Model fit was at its peak when all predictors were considered, with the population variable being the least significant initially. Once predictors were excluded form the model, model fit decreased noticeably yet statistical significance between predictors and the response variable were preserved.

4.3 Task 3 of 3

Logistic regression proved to be a consistently insufficient model for predicting stock market direction in the context of lag variables and over a time series corresponding to the scale of weeks, over a time span of 21 years. The k-nearest neighbors model proved to be more effective in this context (for ample predictors), but became increasingly suspicious following backwards elimination of predictors.

A consistently significant predictor for both models was Lag2, which indicated return percentage two weeks prior to any current, weekly observation of the stock market.

End of Report

References

[1] rdrr.io. (2021, Sep. 15). *Carsetas: Sales of Child Car Seats* [Online]. Available: https://rdrr.io/cran/ISLR/man/Carseats.html

[2] J. Gareth, et al., *An Introduction to Statistical Learning*, 2nd ed. New York: Springer Science + Media, LLC, part of Springer Nature, 2021

[3] RDocumentation. (n/a). *Weekly S&P Stock Market Data* [Online]. Available: https://www.rdocumentation.org/packages/ISLR/versions/1.4/topics/Weekly

[4] G. Upton and I. Cook, *Oxford Dictionary of Statistics*, 3rd ed. Oxford: Oxford University Press, 2014.

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)
}
```