FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

TakeAPeek Learning: Chaining classifiers for multiple regression

Rodrigo Gondim Carvalho da Fonte

WORKING VERSION

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Carlos Soares

February 9, 2016

TakeAPeek Learning: Chaining classifiers for multiple regression

Rodrigo Gondim Carvalho da Fonte

Mestrado Integrado em Engenharia Informática e Computação

February 9, 2016

Abstract

In multiple target regression problems, the objective is to predict the numeric value of multiple dependent Y variables from the numeric values of independent X variables using linear regression. One of the issues with this type of problem is the fact that it does not account for the codependency between the target variables Y themselves.

This objective of this thesis is to adapt the existent multiple regression algorithm to use chained classifiers. By using chained classifiers the prediction of each target variable Y_n is not done independently of the other Y variables. Each prediction model will include the values of previously predicted Y variables. By doing so the model will attempt to combat the co dependence between the target variables thus leading to more accurate results. This thesis works upon an existing article about multi-label classification methods for multi target regression. I will demonstrate the effectiveness of the base algorithm using 12 datasets and prove the influence that the order in which the target variables are predicted have upon the final mean error of all target variables.

Resumo

Em problemas de regressão multi-alvo o objetivo é prever o valor numérico de múltiplas variáveis dependentes Y a partir dos valores numéricos das variáveis independentes X utilizando o algoritmo de regressão linear. Um dos problemas com este tipo de algoritmo é o fato que não se tem em conta a codependência entre as variáveis alvo Y.

Esta tese tem como objetivo adaptar o algoritmo de regressão múltipla existente para usar classificadores em cadeia. Ao utilizar classificadores encadeados a previsão de cada variável Y_n não é feita independentemente das restantes variáveis Y. Cada modelo de previsão incluirá os valores das variáveis Y previamente calculadas. Com isto, o modelo irá ter em conta a codependência entre as variáveis alvo levando a resultados mais precisos. O trabalho feito nesta tese e baseado num artigo existente sobre métodos de classificação multi-label para regressão multi-alvo. Irei demonstrar a eficácia do algoritmo base usando 12 datasets e provar a influência da ordem do qual as variáveis alvo que são previstas têm sobre o erro médio final de todas as variáveis alvo.

Acknowledgements

Firstly I would like to my supervisor Carlos Soares for providing me guidance during this process and for having the patience of enduring me for a semester explaining to me everything I needed to know to make sure I did a good work.

I also thank my parents, my brother and my sister for the support they gave me and for giving me whatever I needed to finish my work.

I would like also like to leave a special thanks to professor Augusto Sousa for his advice and words of encouragement when I was in need of them. To all my friends and colleagues who gave me support and encouragement I thank you. I would like to leave a final heartfelt thanks to Rita Robalo and Tânia Trindade whom if not for their support this thesis would have not been completed.

Rodrigo da Fonte

"Quidquid latine dictum sit, altum videtur"

Ancient roman proverb

Contents

List of Figures

List of Tables

Abbreviations

MTR	Multi Target Regression
ERC	Ensemble Regression Chain
ST	Single Target
MT	Multi Target
MRT	Multiple Regression Tree
PCT	Predictive Clustering Tree
RMSE	Root Mean Squared Error
RRMSE	Relative Root Mean Squared Error
ARMSE	Average Root Mean Squared Error
ARRMSE	Average Relative Root Mean Squared Error
MSE	Mean Squared Error
MAD	Mean Absolute Deviation
MAE	Mean Absolute Error
MTCR	Multi Target Chained Regression

Chapter 1

Introduction

Multiple regression is an algorithm to predict multiple variables from the previously known values of other variables from which there is a correlation between these and the variables that we want to predict.

Multi target regression has multiple applications in the real world. Many of these applications are found in ecological studies [KDW^{+}09] and chemometrics to infer concentrations of several analytes from multi-variate calibration using multi-variate spectral data (e.g.[[Bur99]]). There are many more applications in the fields of finance, medicine, administration, physics. Basically any field that needs predictions for several interdependent numerical at the same time in the same environment potentially need multi-target regression.

The main advantage of multi-target regression over single target regression is that it the dependence between target variables is taken into account rather than predicting one variable at a time. Further we can describe more of the advantages as the following:

- They resemble closely how the researcher thinks about the data
- They allow easier visualization and interpretation of the data
- More data can be analysed simultaneously
- Better insight into the variables relationships

Multiple regression however has its limitations and issues. In some multiple regression tasks we have the problem of multicollinearity between variables. In some cases we have overfitting of predictive models to a specific task. More importantly we have the subject of the codependence between our target variables.

A related task to multi-target regression is multi-label classification, the difference being that multi-target regression deals with real numerical values and multi-label classification deals with discrete values. There has been recent discussion on utilizing chaining classifiers on multi-label classification [SXTGV12], [RPHF14]. Chaining classifiers consist in linking the predictive models in such a way that the value predicted by one model is influenced by the values predicted by the previously generated models. There is an article [SXTGV12] that has already studied the use of chaining models. The objective of this thesis is to recreate this experiment using a different base algorithm and run some experiments in other factors that may alter the performance of the algorithm.

This thesis has four additional chapters. In chapter 2 I will discuss the state of the art on multiple regression and chaining models and explain what we can learn from them. In chapter 3 I formally define the problem and explain the methodologies used to implement a solution, with details about the algorithms and datasets. In chapter 4 I present the evaluation method chosen and the results obtained from my experimental setup. Finally, chapter 5 concludes this dissertation summarizing the work done and talk about future perspectives.

Chapter 2

State of the Art

In this chapter I will discuss the state of art and related articles about multiple regression and chaining classifiers. The papers analyzed are organized into different dimensions that characterize them. Multi-target regression is an algorithm with the objective of predicting multiple numerical variables from the previously known values of other numerical values from which there is correlation between them. A more formal definition of multiple regression tasks is provided in chapter 3.

2.1 Multi-target Regression

Existing work on multi-target regression can be divided into the following dimensions:

- Algorithm
- Data type
- Approach
- Evaluation method
- Application

The analysis is summarized in Table 1. This table aims to give the reader a better understanding of what is the focus of papers written about the subject of multiple regression.

Table 2.1: Articles researched categorized by different scopes

Article	Algorithm	Data type	Approach	Evaluation measures	Application
[HSS15]	Ensemble; Neural network	Numeric	Adaptation	ARRMSE	Empirical study
[SXTGV12]	Stacking;chaining ST models	Numeric	Adaptation	ARRMSE & RRMSE	Empirical study
[TSX14]	Linear multiple regression	Numeric	Combination	ARRMSE	Empirical study
[AZD09]	Rule Ensembles	Numeric	Adaptation	RRMSE	Empirical study
[AKR14]	Genetic Programming	Numeric	Combination	MAD	Empirical study
[BPN07]	Multiple linear regression	Fuzzy sets	Adaptation	RMSE	Empirical study
[HSH15]	Mixture of experts	Numeric	Combination	MSE and MAE	Empirical study
[KDW+09]	Ensemble of MRT	Numeric	Combination	RMSE	Ecological modelling
[Lev15]	Ensemble of PCT	Numeric	Combination	RMSE	Empirical study

2.1.1 Algorithm

Multiple regression problems are present in a great deal of real life problems and can fit into, so it should come with little surprise that almost all of the studied articles deal with different algorithms. A majority of these articles [BPN07] [HSS15] [AKR14] [TSX14] [AZD09] [Lev15] [HSH15] explain how the algorithm they are using for their problem can be improved by using multiple regression analysis in some or all aspects of the algorithm. It is also observable that there is a wide array of different types of algorithms that can be adapted to use multiple regression.

2.1.2 Data type

All articles discuss problems relating to MTR, for that reason it's independent variables are real numeric. One article [BPN07] has numerical fuzzy sets as its main focus. Other works [AZD09] [HSH15] are suited for data types other than numeric, however their focus is to improve their algorithm in dealing with multiple numerical variables.

The dependent variables in the studied articles are also real numerical variables.

2.1.3 Approach

All of the discussed articles discuss how to adapt their algorithm for multiple regression problems or how they introduce multiple linear regression to improve its effectiveness. The need for combination of algorithms comes from the fact that neither algorithm has a silver bullet solution to all problems. What happens is that the problems discussed in some articles [TSX14] [AKR14] [HSS15] [Lev15] are not well suited for multiple regression problems and thus choose to try to incorporate some of the solutions used for these problems into their own solution. This adaptation is done in different ways, like for example making small changes to steps on the original algorithm to make it better suited for MTR problems. Another example of adaptation is the processing of data using MTR separately from the original algorithm.

2.1.4 Evaluation measures

For most of the articles, the authors choose to use either RRMSE [SXTGV12] [AZD09], ARRMSE [HSS15] [SXTGV12] [TSX14] or RMSE [KDW+09] [BPN07] [Lev15]. The fact that they use similar evaluation measures simplifies the task of comparing the results of each paper so that we can learn which method is more suited for each of the studied problems.

2.1.5 Application

The research done on the state of the art in the field of multiple regression was focused on scientific papers and thus most of articles here are empirical studies with the purpose of comparing their respective modified algorithm with the baseline algorithm of their choice using public domain datasets. The only exception to this is an article [KDW+09] that studies a specific problem regarding the modeling of vegetation condition.

It should also be noted that many articles express difficulty in finding datasets for testing and state that reason is the fact that most applications are for industrial use [AZD09] [HSH15] [HSS15] [SXTGV12]

2.2 Chaining Classifiers

Chaining classifiers are a set of predictive models in which the prediction of each model is influenced by the prediction of the previous model. Most recently, chaining classifiers have been used in multi-label classification problems [RPHF14], [KD10].

2.3 Chaining models for multiple regression

On the subject of using chained models for multiple regression problems there is one article in my research [SXTGV12] that deals with it. This article introduces two new algorithms for MTR, a multi-target stacking algorithm and an ensemble of regressor chains (ERC), both of which are based in multi-label classification methods.

Of these two algorithms, I am more focused on the ERC algorithm which deals adapts chaining classifiers used in multi-label classification to an ensemble of predictive models for multi-target regression. The article suggests instead of training a regression models for each target separately each model is trained based on the values of the previously trained models.

2.4 Summary

From the research done it can be concluded that there are many problems that deal with multiple numerical variables simultaneously and many algorithms have difficulties in dealing with these problems. The necessity to cover this issue makes work about multiple regression problems even more valuable as it also indirectly helps these same articles in achieving more accurate results.

Chapter 3

Methodology and Problem Formulation

A multiple regression problem consists in the prediction of multiple numerical values, which we call target variables, from the numerical values other numerical variables which we call input variables. Firstly, I will formally describe multiple regression and then chaining models.

3.1 Multi-Target Regression

There are two vectors, X and Y. X contains d input variables and Y contains m target variables. We extract both vectors from a joint probability distribution $P(X,Y)$, and obtain the sample (x,y) which are part of the domain of X and Y and subsequently categorized into the independent input variables $\{x_1,...,x_d\}$ and the target variables $\{y_1,...,y_m\}$ that we wish to predict and are dependent on the input variables. Given a set of samples S we extract a sample s with n instances we can create and train a model $h{:}X \rightarrow Y$ based on the values of the X and Y vectors from our set of samples S and with a new sample$\{(x_1,y_1),...,(x_q,y_q)\}$ of data we can input the new vector X_q extracted from sample and predict a vector $\hat{Y}_q$ such that $\hat{Y}_q$=$h(X_q)$.

3.2 Chaining models

Let C be a set of classifier models $h_1,...,h_m$ which each model in C receives an input vector X with d numerical variables and outputs a predicted numerical value $\hat{y}_n$ such that $h_n(\mathrm{X}) \rightarrow \hat{y}_n$. Firstly we will train the first model h_1 with the values provided by the input variables in vector X so that $h_1(\mathrm{X}) \rightarrow \hat{y}_1$. With this first model we can predict the first target variable y_1. From here on we iteratively use the predicted values as a new variable for our current target y_n that $\hat{y}_n \rightarrow h_n(\mathrm{X},\hat{y}_1,\hat{y}_2,...,\hat{y}_{n-1})$.

3.3 Problem

It has been previously suggested to adapt multi-label classification methods for multiple regression problems [SXTGV12] which uses chaining models for MTR problems. An important aspect to have in mind in MTR problems is the dependence between the target variables which are not taken into account in a normal MTR problem. By linking the prediction models during the multi-target regression we can pass the target values that we predicted to the next models so that it can have some insight into the values of the targets and model the dependence between the targets.

The purpose of this thesis is to recreate the existing approach in order to further investigate it and to identify opportunities for improvement. In summary, we can define the goal as the following:

- Recreate the implementation done by [SXTGV12]
- Study and explore the implementation to identify possible modifications that can improve it's performance
- Make an empirical study on the algorithm

3.4 Methodology

We use an algorithm that we call Multi Target Chained Regression (MTCR) that is based on the ERC algorithm used by [SXTGV12]. In this version used for the study however we do not use an ensemble of regression models like [SXTGV12] does. Formally, the algorithm can be defined as following:

```
Ensure: targets has no non-numerical attributes
  targets ← variables selected for prediction
  df ← input dataset
  folds ← create 10 folds for validation
  for $i = 0$ to 10 do
    training_set ← folds[subset≠i]
    test_set ← folds[subset=i]
    models ← regrChain(training_set,targets)
    preds ← regrChain.predict(models,test_set,targets)
    errors[i] ← rrmse(preds,mean(training_set[,targets]),test_set[,targets])
  end for
  error ← mean(errors)
```

Algorithm 1: Multi Target Chained Regression

The regrChain function in the above algorithm which is our training function is defined as following:

```
Ensure: targets has no non-numerical attributes
  targets ← variables selected for prediction
  df ← input dataset
  folds ← create 10 folds for validation
  for $i = 0$ to 10 do
    training_set ← folds[subset≠i]
    test_set ← folds[subset=i]
    models ← regrChain(training_set,targets)
    preds ← regrChain.predict(models,test_set,targets)
    errors[i] ← rrmse(preds,mean(training_set[,targets]),test_set[,targets])
  end for
  error ← mean(errors)
```

Algorithm 2: Multi Target Chained Regression

```
Input: workset, targets
Output: h
  h[] ← list of models
  attrs ← workset[,!targets]

  h[1] ← generate(targets,workset[-targets.indexes≠1]){Generate first model of the chain}

  {Generate remaining models}
  for $i = 2$ to length(targets) do
    workset[,targets[i-1]] ← predict(h[i],workset(,attrs + targets[1..i-1]))
    h[i] ← generate(targets,workset(,attrs + targets[1..i]))
  end for
```

Algorithm 3: regrChain

And the regrChain.predict:

The modelling step starts by generating the first model based on the values of the input vector X for the first target variable y_1. From here on it is an iterative cycle to generate the remaining models based on the values of the input vector X complemented with all the target variables that have modelled so far plus the current target.

Having generated the models, we can start to predict new Y values. The prediction is done in the same fashion as the model generation. We iteratively predict for the current target based on the input vector X and all the targets that where already predicted. After predicting the value for the current target, we then replace the true value for that target in the workset with the value we have predicted to be later used in the prediction of the next target variable.

From the implementation of this algorithm we came up with an idea for an experiment to perform on the MTCR algorithm. When we are predicting each target independently it does not really matter the order in which we are predicting the targets as the outcome of each model does not influence the others. In chained regression however each prediction model depends on the values from the input variables and the values of previously predicted models with the exception

```
Input: models,workset,targets
Output: p

  attrs ← workset[,!targets]
  for $i = 1$ to length(targets) do
    prevs ← predict(models[i],workset[,attrs + targets[1..i]])
    workset[,targets[i]] ← prevs
  end for
  p ← workset[,targets[i]]
```

Algorithm 4: regrChain.predict

of the first model. This first model predicts the value for the first target variable without any values of other target variables. A good candidate for the first target to be predicted would be the target that has the lowest correlation with the other target variables (ideally none at all). Using the reasoning for the last target, a good candidate would be the target that has the highest correlation with the other target variables. Based on this reflection we include here a version of the MTCR that shuffles the order of the vector containing the target variables.

```
Output: targets has no non-numerical attributes
  targets ← variables selected for prediction
  df ← input dataset
  folds ← create 10 folds for validation
  permute(targets)
  for $i = 0$ to 10 do
    training_set ← folds[subset≠i]
    test_set ← folds[subset=i]
    models ← regrChain(training_set,targets)
    preds ← regrChain.predict(models,test_set,targets)
    errors[i] ← rrmse(preds,mean(training_set[,targets]),test_set[,targets])
  end for
  error ← mean(errors)
```

Algorithm 5: Multi Target Chained Regression with target permutation

3.5 Datasets

In this section I will explain each of the datasets used to obtain our results. Since we are doing an empirical study based on the work of [SXTGV12] we are going to used the same datasets used by them.

3.5.1 Airline Ticket Price

These datasets are related to the prediction of airline ticket prices. Each sample represents a series of observations from a specific observation time to a departure time. The input variables are values that may or may not be useful to predict the price of a ticket for a certain departure date. We have two datasets available, both with 6 target variables: ATP1d which has as target variables the ticket price for the next day and ATP7d has as target variables the minimal price observed for the next seven days. Both predict the prices for 6 airline companies. ATP1d has 337 instances with 417 variables and ATP7d has 296 instances with 417 variables. The number of variables on both datasets exceed the number of instances. There is a higher chance that we will overfit our models during training.

3.5.2 Electrical Discharge Machining

The task of this dataset is to predict two variables that are normally controlled by a human operator. The target variables can only take the values -1,0 and 1. This dataset has 154 instances and 18 variables.

3.5.3 River Flow

This dataset concerns the prediction of river flows in a 48 hour period at specific locations at 8 sites in the Mississippi river in the United States. Each sample contains the most recent observation and past observations (6, 12, 18, 24, 36, 48 and 60 hours). This dataset is divided into two. The first dataset RF1 has a total of 72 variables with 8 target variables and 9125 instances. The RF2 has additional attributes regarding precipitation forecast for the sites and has 584 variables with 8 target variables and 9125 instances.

3.5.4 Supply Chain Management

The Supply Chain Management datasets are derived from the Trading Agent Competition in Supply Chain Management tournament in 2010. Each instance concerns one day of the tournament and the input variables are the prices for a specific tournament day. There are 4 more additional time-delayed observations to assist in anticipation of prices. Each dataset contains 16 targets that correspond to the next day mean price (SCM1d) or the mean price for 20-days in the future (SCM20d). The SCM1d has 9803 instances with 296 variables and the SCM20d has 8966 instances with 77 variables.

3.5.5 Water Quality

This dataset concerns the representation of plant and animal species in in Slovenian rivers. It has 16 input variables referring to physical and chemical water parameters and 14 target variables with 1060 instances.

3.6 Summary

In this section I explained in detail what I propose to do and how I did it. I explained the concepts of multiple regression and chaining models that we use on this work. I also presented the pseudo-code of the algorithm used to accomplish the goal of this work and finally explained the datasets used for testing.

Chapter 4

Evaluation and Results

This section has the purpose of explaining in detail the experimental setup and the analysis of the results obtain from the experimental setup.

4.1 Experimental Setup

To evaluate the effectiveness of our adapted algorithm we choose to use the Relative Root Mean Squared Root (RRMSE). For better accuracy and to prevent overfitting of the models we choose to use a ten fold cross-validation.

Assuming h as one of our predictive models and D_{test} as our test set we can formally define RRMSE as:

$$RRMSE(h, D_{test}) = \sqrt{\frac{\sum_{x,y_j \in D_{test}} (\hat{y}_j - y_j)^2}{\sum_{x,y_j \in D_{test}} (\bar{Y}_j - y_j)^2}}$$

where $\bar{Y}_j$ is the mean value of the target variable Y_j and $\hat{y}_j$ is the prediction we obtained from h for Y_j. However the relative error is the absolute error divided by the absolute value, so we can also formulate it like this:

$$\text{RRMSE(pred,vals,obs)} = \frac{rmse(P,O)}{rmse(M(V)[length(V)][nrow(O)],O)}$$

And our RMSE function is defined as:

$$\text{RMSE(P,O)} = \sqrt{\sum (P-O)^2}$$

Where P are our predicted values for our targets, V are the mean values of the target variables in our training set, O are the values of our target variables in our test set. As a last step, I calculate the ARRMSE for all RRMSE values obtained from our ten fold cross-validation.

To compare our results we are going to need baseline values to judge if the results are an improvement or not. For that purpose I ran three sets of experiments: The first set of experiments use the MTCR algorithm previously described in Chapter 3 with our 8 datasets using a ten fold cross-validation. To obtain more solid results we run this experiment 50 times so that later we can analyze the results and make sure that none of them deviate much from the average result. The second set of experiments are our baseline experiments, the one which will be use to compare our results. These experiments follow the same algorithm on our first set of experiments except that it does have chaining models to aid in the prediction of the target variables. Our third set of experiments tests the idea proposed in Chapter 3 of shuffling the order in which the target variables are predicted in the MTCR algorithm so that we can analyze whether or not it has a significant change to the final results.

4.2 Results

For each set of the 50 experiments, I have obtained these ARRMSE values presented in Table 4.1.

Table 4.1: Results obtained from the sets of experiments. The first column refers to dataset which the experiment was ran, the second column has the values obtained from the baseline algorithm, the third column has the values from the adapted algorithm with chaining models and the fourth column are the values from the algorithm with chaining modes and with a randomized order for the prediction of each target variable

Dataset	Baseline	w/ Chaining models	w/ CM & target permutation
ATP1d	0.1131121	0.1156379	0.1144487
ATP7d	0.1726954	0.1703362	0.1729925
EDM	0.7621011	0.7885278	0.7737469
SCM1d	0.3937462	0.3934722	0.3936193
SCM20d	0.6486522	0.6489777	0.6490212
WQ	0.9582973	0.957545	0.9579445
RF1	0.5031137	0.2292605	0.5082242
RF2	0.2305416	0.228815	0.2296425

Aside from a small outlier of 0.3 on the CM algorithm for the RF1 dataset there seems to be little difference between the results obtained. A possible explanation may be that the datasets that we used for this experimental setup present a very low correlation between the target variables thus making little difference when using CM and CM with target permutation.

To get a better reading of the actual results, I present Table 4.2 with the median of the ARRSME instead of the average:

Table 4.2: Medians of the ARRMSE for all experiments

Dataset	Baseline	w/ Chaining models	w/ CM & target permutation
ATP1d	0.1112288	0.1147338	0.114577
ATP7d	0.16922	0.1644026	0.1698244
EDM	0.7753854	0.8887811	0.7755602
SCM1d	0.3937175	0.3934049	0.3935307
SCM20d	0.6490938	0.649066	0.6490869
WQ	0.9582089	0.9567444	0.9574621
RF1	0.5025652	0.2308819	0.5085379
RF2	0.2333964	0.2280929	0.2321634

Again, the only value of note is the value obtained for the CM algorithm with the RF1. For the remaining values there is little difference.

4.3 Summary

In this chapter I presented the experimental setup used and justified its use with the need to have comparable results with the results of others works. I then proceeded to show the results obtained and drew my conclusions about them.

Chapter 5

Conclusions and Future work

This work presented an empirical study on a existing MTR method using chaining classifiers based on an existing work [SXTGV12] that does an empirical study of an ensemble of regression models using classifier chains. Its objective was to further analyze the use of chaining models in MTR in order to identify opportunities for enhancement of the algorithm.

Three set of experiments, each with 50 experiments, were run and from the results calculated the ARRMSE. The experiments were done on 8 public domain datasets, with different number of instances, target variables and input variables. The datasets where chosen based on the datasets used by other works on the same field. However, as there are few public datasets for this task, the results should be interpreted carefully.

For each dataset I conducted three experiments. The first experiment is the baseline of this thesis. We run a multi-target regression algorithm for all of our target variables using linear regression as our base algorithm. In this setup we assume independence between the target variables. The second experiment is the adapted algorithm that incorporates a chaining model concept, meaning that for each target that the model is predicting will have the values of all previously predicted target variables to aid in its prediction in order to take advantage of the dependence between targets. The third experiment is similar to the second experiment with the difference being that the order in which the target variables are predicted is random. The reasoning behind this experiment is that while the target variables are dependent between them some of the targets are more dependent than others. This means that there is an optimal order in which to predict, starting by the least dependent target to the most dependent. This third and final experiment has as its objective to see if the order in which the targets are predicted has any influence in the final error outcome of the algorithm.

To measure the performance of all algorithms I have chosen to use the ARRMSE (Average Relative Root Mean Squared Error). The reason for using this metric is to have comparable results with other works in the same field as many of them use the ARRMSE or similar as the metric of their choice. After the analysis of the results we can conclude that there is no significant change in using chaining models in MTR and by using target permutation. The results have been analyzed and I can conclude that there seems to be little change in using an algorithm with chaining models incorporated to the baseline algorithm that we use. When comparing our results with the results in [SXTGV12] we can see that there are differences between the values obtained. However it should be noted that there are differences between the the algorithms presented in this dissertation and the algorithms in [SXTGV12]. We do not use an ensemble of models, we use different base algorithms and the pre-processing of the datasets for training and testing are also not the same.

5.1 Future work

For future work into this thesis I suggest first a consolidation work. For example the use of datasets in which there is a strong dependence between targets would be relevant to further analyze the target permutation experiment. I also suggest using different base algorithms as for this dissertation I only used the linear regression algorithm. It would be interesting to repeat these same experiments using other algorithms such as regression trees.

Secondly, I suggest an enhancement of the algorithm using a set of models that predict the error for each prediction model. The idea is that for each prediction we have a model that has a prediction for the error that model will have so that we can adjust the value predicted by adding the error predicted.

References

[AKR14] Ignacio Arnaldo, Krzysztof Krawiec, and Una-may O Reilly. Multiple regression genetic programming. *GECCO '14 Proceedings of the 2014 conference on Genetic and evolutionary computation*, pages 879–886, July 2014.

[AZD09] Timo Aho, Bernard ZEnko, and Saso Dzeroski. Rule ensembles for multi-target regression. *2009 Ninth IEEE International Conference on Data Mining*, pages 21–30, 2009.

[BPN07] Andrzej Bargiela, Witold Pedrycz, and Tomoharu Nakashima. Multiple regression with fuzzy data. *Fuzzy Sets and Systems*, pages 2169–2188, April 2007.

[Bur99] John F.Viveros Román Burnham, Alison J.MacGregor. Latent variable multivariate regression modeling. *Chemometrics and Intelligent Laboratory Systems*, pages 167–180, March 1999.

[HSH15] Esmaeil Hadavandi, Jamal Shahrabi, and Yoichi Hayashi. Spmoe: a novel subspace-projected mixture of experts model for multi-target regression problems. *Proceedings of the Third Conference on Object-Oriented Technologies and Systems*, March 2015.

[HSS15] Esmaeil Hadavandi, Jamal Shahrabi, and Shahaboddin Shamshirband. A novel boosted-neural network ensemble for modeling multi-target regression problems. *Engineering Applications of Artificial Intelligence*, pages 204–219, June 2015.

[KD10] Eyke Hüllermeier Krzysztof Dembczyński, Weiwei Cheng. Bayes optimal multilabel classification via probabilistic classifier chains. *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 279–286, 2010.

[KDW^{+}09] Dragi Kocev, Saso Dzeroski, Matt D. White, Graeme R. Newell, and Peter Griffioen. Using single- and multi-target regression trees and ensembles to model a compound index of vegetation condition. *Ecological Modelling*, pages 1159–1168, January 2009.

[Lev15] Jurica Levati. Semi-supervised learning for multi-target regression. April 2015.

[RPHF14] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. 2014.

[SXTGV12] Eleftherios Spyromitros-Xioufis, Grigorios Tsoumakas, William Groves, and Ioannis Vlahavas. Multi-label classification methods for multi-target regression. pages 1–21, June 2012.

[TSX14] Grigorios Tsoumakas and E Spyromitros-Xioufis. Multi-target regression via random linear target combinations. *arXiv preprint arXiv:*, pages 1–16, 2014.

Appendix A

This section presents all the results obtained from the 3 set of experiments on 8 datasets containing a total of 1200 numerical values. The data is organized first by dataset and then by the experiment. We present all individual values, the average value and the median value. We also present a histogram and a plot with these same values.

A.1 ATP1d Baseline

Values:

0.118481637555109	0.129736434556252	0.120114320791078	0.144548052047007	0.112242706575596
0.111337479373168	0.120533500619108	0.109330151812616	0.11381147495039	0.106441140596352
0.146338956579982	0.0906112171550271	0.109926157258124	0.100449997142077	0.106241309856617
0.127775751722505	0.104460649970145	0.105254383622658	0.103382508410278	0.111321884461235
0.11263597966168	0.114130336761317	0.121597963159678	0.103105613747508	0.11776963713373
0.116793153970259	0.104014364102419	0.109624735721653	0.102220152900271	0.117260809702982
0.124411256881815	0.10987262935852	0.130299422746941	0.0913216609234036	0.117020104218718
0.120007965246976	0.0969902639376351	0.103113707887365	0.116253697401627	0.107779068714462
0.12564097427216	0.107553336067268	0.135841798175693	0.111135684187306	0.107159262610319
0.107937532456312	0.107668350724487	0.103101405113003	0.109104060952624	0.111902750887533

Average: 0.1131121 Median: 0.1112288

Figure A.1: Histogram of results for the baseline experiment on the ATP1d dataset

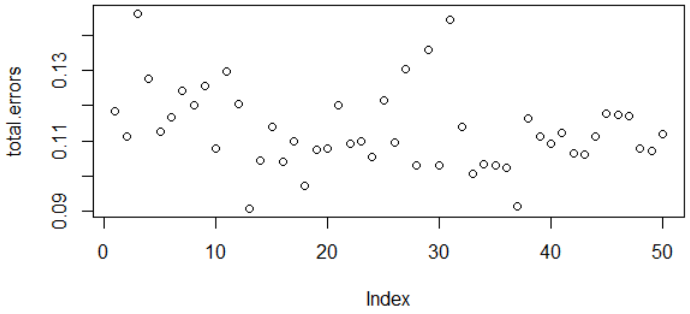

Figure A.2: Plot of results for the baseline experiment on the ATP1d dataset

A.2 ATP1d CM

Values:

0.106872211127578	0.109352548223152	0.108564231800875	0.103636452974495	0.114630355809278
0.144831151177413	0.133465175114422	0.116781008925507	0.120424533191331	0.107857910116275
0.112894813201679	0.105837629036267	0.126681273693317	0.114837276964064	0.11557793290613
0.11898784498463	0.11074751481213	0.13021290665899	0.124655157507969	0.109831867476148
0.101149278725634	0.115976094621893	0.124228626765734	0.117493610418594	0.111627935349243
0.129595224868442	0.11068254379321	0.113814783794361	0.103628671319001	0.096245534960169
0.127518905159472	0.118329758386556	0.121826671797564	0.118082629284618	0.113169047133308
0.114527492807437	0.11485283007649	0.114157788256524	0.119738031586106	0.112218609869748
0.0977071845882733	0.0973406414584453	0.110497882996482	0.123326068955579	0.1223141560535
0.11844686603911	0.106192986197227	0.130674816366026	0.113211354765095	0.1266368955363

Average: 0.1156379 Median: 0.1147338

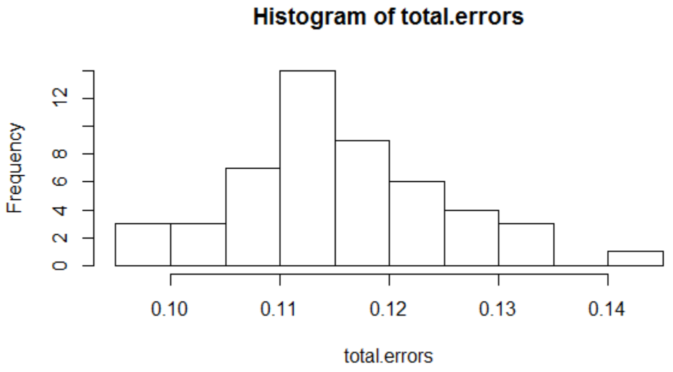

Figure A.3: Histogram of results for the CM experiment on the ATP1d dataset

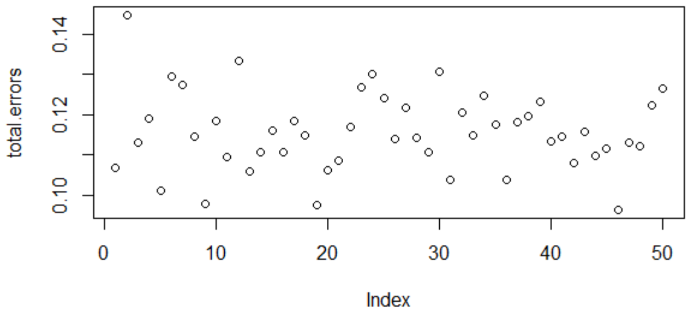

Figure A.4: Plot of results for the CM experiment on the ATP1d dataset

A.3 ATP1d CM w/ perm

Values:

0.10577280981654	0.114996872292714	0.117502436172499	0.116298960697091	0.0916724013636712
0.12217082889078	0.11063442343645	0.115490108107344	0.128218466131878	0.113160980782818
0.131901123960522	0.121851469991586	0.113457675463891	0.111330469399189	0.119676039451162
0.130282812374255	0.0972730856583667	0.116026239899016	0.100774744302264	0.111746108755936
0.129070736328536	0.113840087169977	0.0925037079935399	0.116844389864942	0.111963800284479
0.108338195411094	0.0995615502288418	0.0988478288752333	0.0978376025248055	0.110419673257148
0.140775488598726	0.114157208477259	0.119553577114203	0.0984564002672852	0.114048209719853
0.0833249059300946	0.128995623556342	0.121696069244634	0.110091670225053	0.110995685379511
0.134542259925629	0.128077467563433	0.122277343536413	0.110713886932471	0.123767693269194
0.118574088680479	0.126568418843311	0.100033355194389	0.130432728603812	0.115888601433347

Average: 0.1144487 Median: 0.114577

Figure A.5: Histogram of results for the CM w/ permutation experiment on the ATP1d dataset

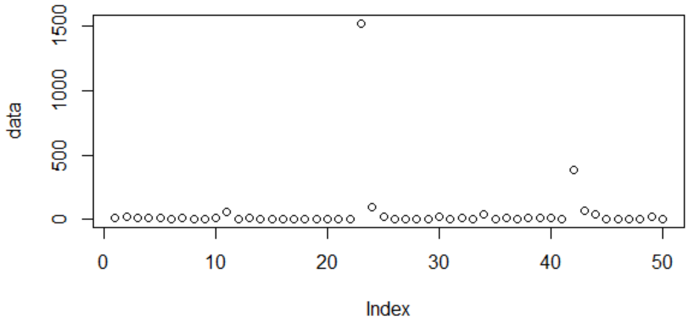

Figure A.6: Plot of results for the CM w/ permutation experiment on the ATP1d dataset

A.4 ATP7d Baseline

Values:

0.16284978381446	0.178054221370214	0.174186688196792	0.149412387596404	0.14534804245461
0.187666912689082	0.198325787928813	0.185965698359066	0.177410898206449	0.163035916836853
0.20164937184298	0.150248763958752	0.178631238220065	0.194844030780692	0.163072999536385
0.138904366088956	0.156900571202076	0.191599376090496	0.155621775308463	0.175533521320878
0.185689695476254	0.16075207914506	0.168333247761486	0.222589854606112	0.153368960674307
0.215338182400281	0.147327996503436	0.165769418992138	0.185245928504205	0.158961638788111
0.226557483484046	0.139020434870395	0.175285126531046	0.152109393843001	0.175909291371221
0.173868350480308	0.174786520637348	0.162122225547869	0.211923202150968	0.166745065649118
0.149351240632163	0.202545366513049	0.169550244861142	0.166520178705518	0.150697596488342
0.155044453877727	0.149403556634789	0.196756637066153	0.168889743983368	0.175043998573505

Average: 0.1726954 Median: 0.16922

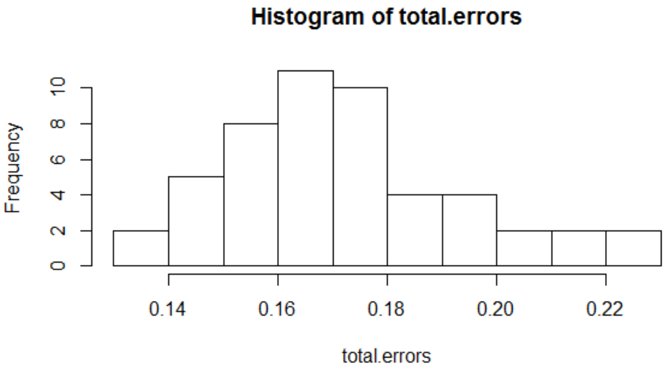

Figure A.7: Histogram of results for the baseline experiment on the ATP7d dataset

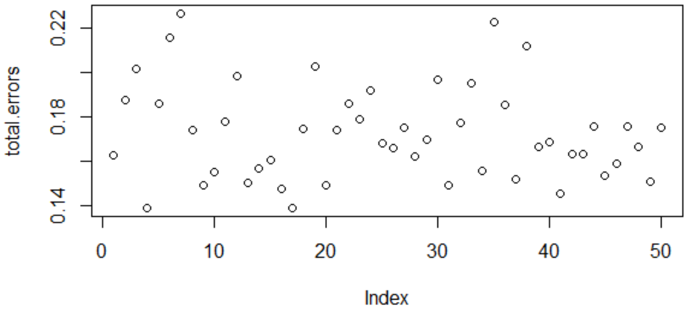

Figure A.8: Plot of results for the baseline experiment on the ATP7d dataset

A.5 ATP7d CM

Values:

0.197281307128731	0.167539364569838	0.203035535874155	0.15348910018736	0.210158291103609
0.190519936944364	0.156224786526133	0.164032166316052	0.159052308311795	0.155051203125352
0.167587929416691	0.178357640692891	0.160601998302529	0.177593325516328	0.228720678329048
0.152213088640469	0.184867205802714	0.154694742234806	0.15183014305104	0.163010120709685
0.193006967701942	0.155774801438172	0.185685646145916	0.164454355688514	0.175456057916255
0.166002189135867	0.1384745215697	0.164350856021947	0.196335235799116	0.191426428938295
0.157135925755486	0.140801518795898	0.167461229195575	0.158595148200379	0.160578379350218
0.153777236628027	0.204560143633772	0.16160292651907	0.14055871511699	0.191578260112947
0.156947614313754	0.177851202855893	0.165669395427633	0.209828842990673	0.172629835025024
0.155029904297866	0.151418713272249	0.158761171988933	0.144041362863668	0.181153427525782

Average: 0.1703362 Median: 0.1644026

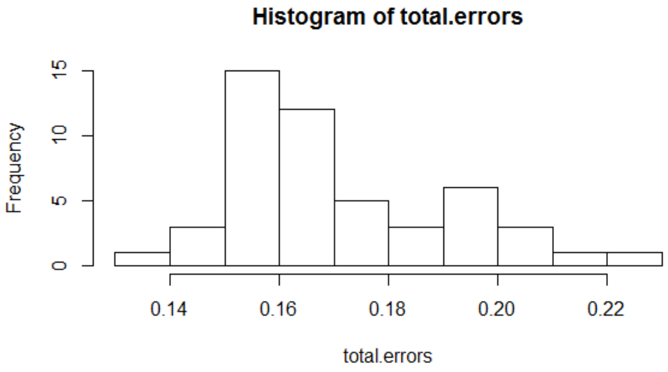

Figure A.9: Histogram of results for the CM experiment on the ATP7d dataset

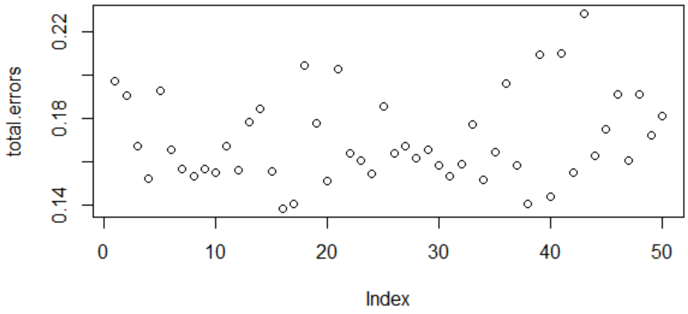

Figure A.10: Plot of results for the CM experiment on the ATP7d dataset

A.6 ATP7d CM w/ perm

Values:

0.203432342134204	0.207882548243152	0.189786087733965	0.164447170393143	0.166669201715582
0.197818465244803	0.152403189350059	0.15453030511601	0.17243305287615	0.152496832028417
0.131151271031621	0.169475524311796	0.219854456457083	0.183293766618847	0.191330241777375
0.183901426944509	0.144642474443871	0.157612997162771	0.169995201969443	0.151524282603202
0.163479154087932	0.149683365730487	0.169653605097602	0.154442454860081	0.177165169933489
0.204174072280352	0.177203849200655	0.186586173233559	0.163300785635832	0.188839687652385
0.160519150868124	0.164902733983117	0.175293954276725	0.2007299419866	0.17923922199259
0.165795691769887	0.137594043806192	0.220050066968066	0.178814363022563	0.140473276047554
0.187348727414181	0.163573383006598	0.16590363665157	0.157148974447152	0.179428901826814
0.199264182021355	0.175767477106387	0.184210369547048	0.160921921895494	0.153437220127092

Average: 0.1729925 Median: 0.1698244

Figure A.11: Histogram of results for the CM w/ permutation experiment on the ATP7d dataset

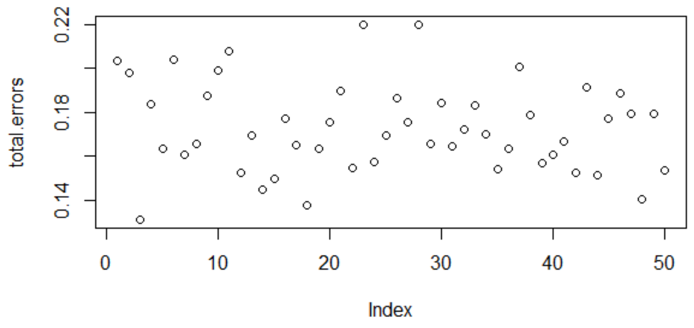

Figure A.12: Plot of results for the CM w/ permutation experiment on the ATP7d dataset

A.7 EDM Baseline

Values:

0.809912284526321	0.71981818844822	0.787022649136193	0.759622279256502	0.775385424765607
0.71981818844822	0.71981818844822	0.787022649136193	0.759622279256502	0.775385424765607
0.71981818844822	0.71981818844822	0.787022649136193	0.759622279256502	0.775385424765607
0.71981818844822	0.787022649136193	0.787022649136193	0.759622279256502	0.775385424765607
0.71981818844822	0.787022649136193	0.787022649136193	0.759622279256502	0.775385424765607
0.71981818844822	0.787022649136193	0.759622279256502	0.759622279256502	0.775385424765607
0.71981818844822	0.787022649136193	0.759622279256502	0.759622279256502	0.779778723131036
0.71981818844822	0.787022649136193	0.759622279256502	0.775385424765607	0.779778723131036
0.71981818844822	0.787022649136193	0.759622279256502	0.775385424765607	0.779778723131036
0.71981818844822	0.787022649136193	0.759622279256502	0.775385424765607	0.779778723131036

Average: 0.7621011 Median: 0.7753854

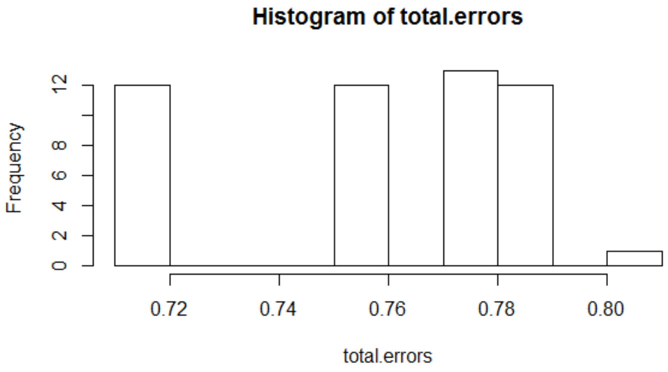

Figure A.13: Histogram of results for the baseline experiment on the EDM dataset

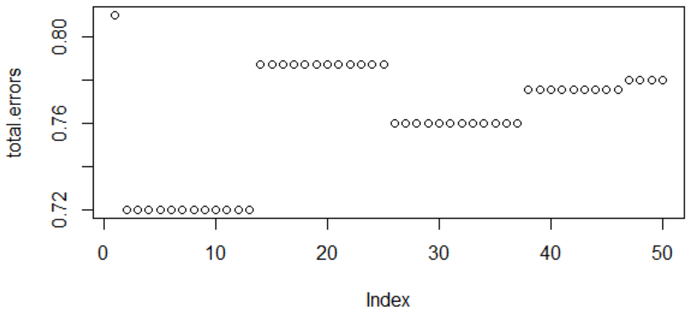

Figure A.14: Plot of results for the baseline experiment on the EDM dataset

A.8 EDM CM

Values:

0.964703005021748	0.956041485055016	0.956272518785221	0.961816643880931	0.957012163630915
0.960007560350917	0.954407198925163	0.955727980496405	0.960469113429564	0.960493524529302
0.954916021320433	0.95666820750204	0.950737847454312	0.960099318161223	0.960370929362055
0.95466477625074	0.962155394789868	0.95663494641413	0.953725698794597	0.95635986341514
0.966054424193374	0.956664510109079	0.954434245709309	0.951336337997848	0.953675137876985
0.960849981485938	0.956291534850853	0.955310397940014	0.954401387465839	0.962135041164629
0.956972217189972	0.955758172946164	0.958727855033737	0.951628731116538	0.959283452383813
0.958012293195973	0.951895590046623	0.956820548585938	0.959541499813187	0.958871590526512
0.957578549747174	0.955931170409493	0.962337846945408	0.961067402643209	0.962089481365633
0.956659744149982	0.961358988716025	0.955958325996449	0.951625632142837	0.960692741809616

Average: 0.7885278 Median: 0.8041579

Figure A.15: Histogram of results for the CM experiment on the EDM dataset

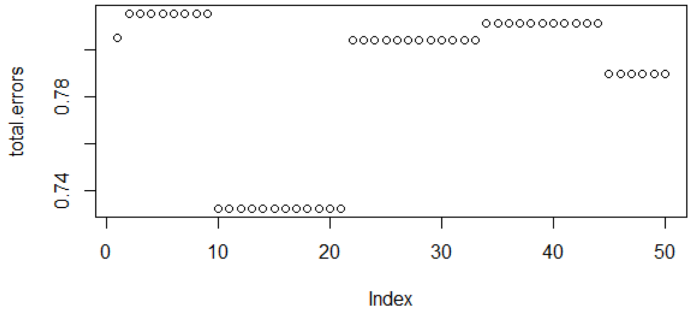

Figure A.16: Plot of results for the CM experiment on the EDM dataset

A.9 EDM CM w/perm

Values:

0.823768810514044	0.751280688753552	0.743680670038329	0.81927910359248	0.775560242271949
0.796633457614159	0.751280688753552	0.743680670038329	0.81927910359248	0.775560242271949
0.751280688753552	0.751280688753552	0.743680670038329	0.81927910359248	0.775560242271949
0.751280688753552	0.743680670038329	0.743680670038329	0.81927910359248	0.775560242271949
0.751280688753552	0.743680670038329	0.743680670038329	0.81927910359248	0.775560242271949
0.751280688753552	0.743680670038329	0.81927910359248	0.81927910359248	0.775560242271949
0.751280688753552	0.743680670038329	0.81927910359248	0.81927910359248	0.775560242271949
0.751280688753552	0.743680670038329	0.81927910359248	0.775560242271949	0.775560242271949
0.751280688753552	0.743680670038329	0.81927910359248	0.775560242271949	0.775560242271949
0.751280688753552	0.743680670038329	0.81927910359248	0.775560242271949	0.74061481369631

Average: 0.7737469 Median: 0.7755602

Figure A.17: Histogram of results for the CM w/ perm experiment on the EDM dataset

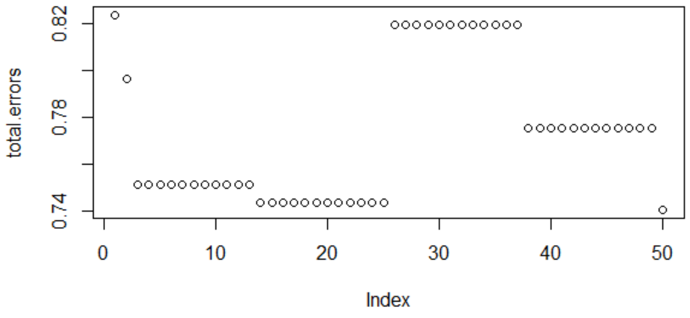

Figure A.18: Plot of results for the CM w/ permutation experiment on the EDM dataset

A.10 SCM1d Baseline

Values:

0.393966376447381	0.394485687647654	0.393152964535779	0.393281901635723	0.394154293378681
0.39303222710673	0.393765778664993	0.393333539742969	0.393586041200029	0.394131273411669
0.393824440448467	0.393441481445026	0.393177578353577	0.393962804207604	0.394019605960969
0.393946542520062	0.393804911252416	0.394134525215923	0.393624542347856	0.393769213682856
0.393447512261215	0.393989817286543	0.393574906781605	0.3940144193374	0.394518339177065
0.393555315945583	0.39366921211658	0.393192602262694	0.393398058348286	0.394187642480634
0.393590631977574	0.393368004288849	0.393272753730019	0.393772439523323	0.393634298317033
0.393529452710973	0.394116418505474	0.393408095533731	0.394755961739798	0.394087801038135
0.393576224729531	0.39321669386515	0.393401949629465	0.393154953962061	0.394140860041651
0.393942627420928	0.393912553807058	0.394523694112523	0.394191027980714	0.393572447185425

Average: 0.3937462 Median: 0.3937175

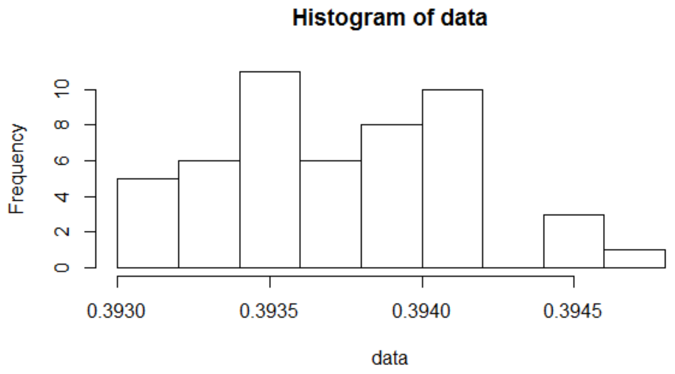

Figure A.19: Histogram of results for the baseline experiment on the SCM1d dataset

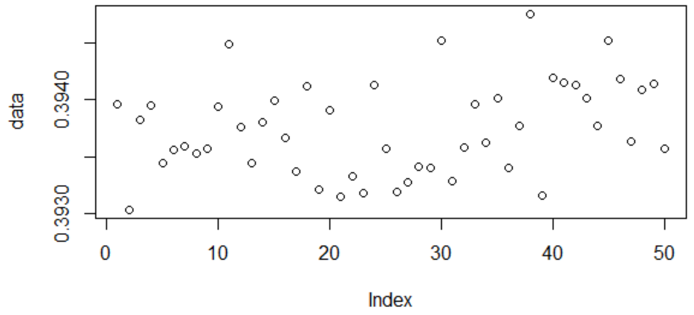

Figure A.20: Plot of results for the baseline experiment on the SCM1d dataset

A.11 SCM1d CM

Values:

0.392842673159828	0.393501851547821	0.393188237817661	0.394144016867385	0.393317604516381
0.393327196315766	0.393517020422876	0.394704727529823	0.393247439160274	0.392862014689363
0.393123405838178	0.392704248359052	0.393563847487171	0.393065378409447	0.394139757724793
0.392938194130948	0.393573389992898	0.392738839273251	0.392955383171922	0.39443439645638
0.393319504706741	0.393716233282877	0.392938493040388	0.392472861698114	0.393640536840196
0.39401985263999	0.392994080288141	0.394059681682758	0.392609564578612	0.394234286297937
0.392537504471126	0.393283164431539	0.39374117124053	0.39323298746344	0.393533066522545
0.393649123140358	0.393442284996332	0.394277474059191	0.393745211738228	0.393203678301094
0.393200919423045	0.394296498118948	0.394206853706862	0.393405538202012	0.393197079217835
0.393404304335557	0.394217040084306	0.39341267440439	0.393314100581925	0.394416970651027

Average: 0.3934722 Median: 0.3934049

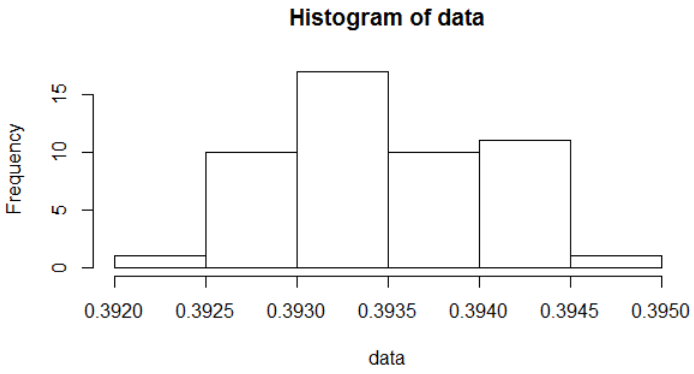

Figure A.21: Histogram of results for the CM experiment on the SCM1d dataset

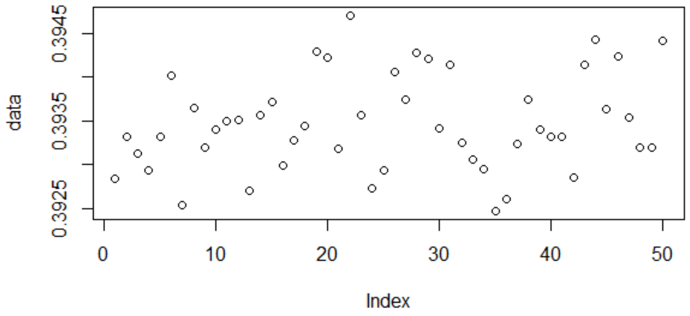

Figure A.22: Plot of results for the CM experiment on the SCM1d dataset

A.12 SCM1d CM w/ perm

Values:

0.393032565590101	0.39414977615439	0.393431115848162	0.393296770896923	0.393531958428196
0.393355009157819	0.393786138213008	0.393129329499365	0.392244943319524	0.393476822755711
0.394231146860164	0.393508735161025	0.393356360251412	0.393669267013179	0.393896982375798
0.39449981083531	0.393824980925637	0.393203982858456	0.394359055450567	0.393655874180242
0.392037591581742	0.392946227070749	0.39385489634725	0.393524366401199	0.394327186797157
0.393142563614035	0.394793041446489	0.394617964262688	0.394555457892759	0.394474834286541
0.394128498930381	0.393052044300064	0.39322096908133	0.393857083741204	0.393871141628007
0.393917478996534	0.393001352587326	0.393800054363138	0.393130744904604	0.39406401103965
0.393698076934633	0.39322340232311	0.393404929444944	0.393295445078677	0.394414651732778
0.393125813004273	0.393529479358144	0.39428067444716	0.393203622599437	0.392830037860348

Average: 0.3936193 Median: 0.3935307

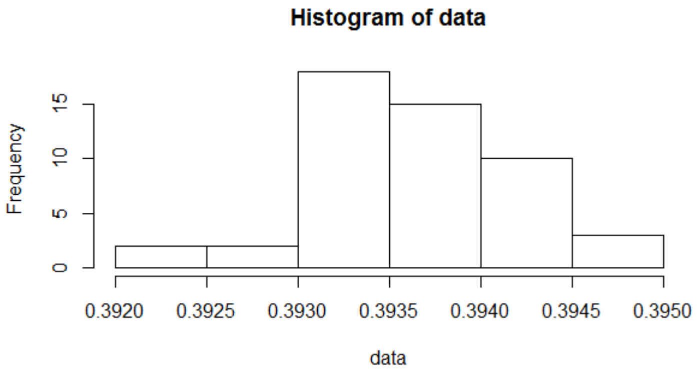

Figure A.23: Histogram of results for the CM w/ permutation experiment on the SCM1d dataset

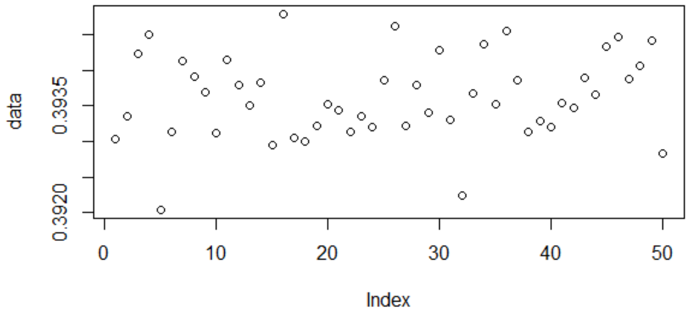

Figure A.24: Plot of results for the CM w/ permutation experiment on the SCM1d dataset

A.13 SCM20d Baseline

Values:

0.644965302532856	0.653548662803597	0.65123023429726	0.646356031209507	0.648918272014566
0.651671352267859	0.644305651797862	0.649592734351364	0.647815627578701	0.648597978733168
0.645507548532171	0.649473157172052	0.646606111103581	0.647103209404903	0.648444959804298
0.646678729546071	0.650234230949851	0.647411867902046	0.651581607632126	0.648572497925823
0.648752458487499	0.653439894325942	0.648065814157238	0.64935048548614	0.651243211338784
0.641949594970681	0.65134228439785	0.643228299684113	0.652334905655899	0.647241350134633
0.641269978899835	0.644209187060307	0.651355700650641	0.649989449156017	0.649655005431902
0.652323655697641	0.645567760279082	0.64924472244886	0.649074486888471	0.652733835451712
0.650451653208012	0.649181879944312	0.648718772891555	0.649113198173481	0.649445628237142
0.651543095506765	0.652380988311045	0.650397831180663	0.647171414670751	0.643219906541493

Average: 0.6486522 Median: 0.6490938

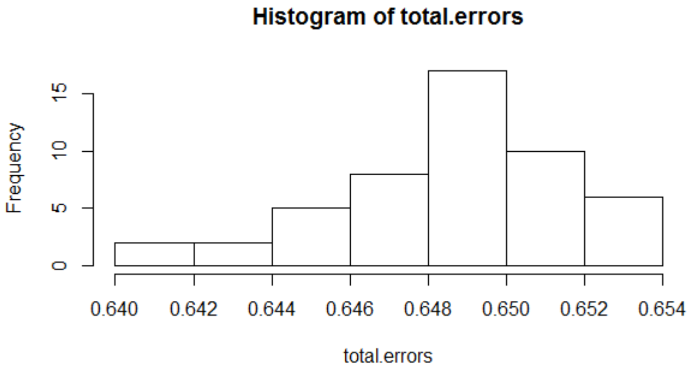

Figure A.25: Histogram of results for the baseline experiment on the SCM20d dataset

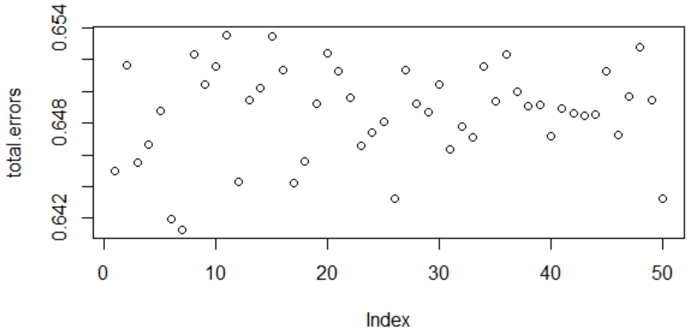

Figure A.26: Plot of results for the baseline experiment on the SCM20d dataset

A.14 SCM20d CM

Values:

0.651448547084851	0.648172579685131	0.651897457961529	0.651829266686917	0.648280883593783
0.650396245686613	0.650578185722933	0.644792575402866	0.647002731897358	0.647517882103838
0.649962291282401	0.650224591087031	0.652747486131777	0.649900206790643	0.651074691983144
0.648634735780298	0.648909101008726	0.647871990779151	0.653798878241248	0.652741458398081
0.643832113893592	0.645193575741175	0.646680209286413	0.652646861787755	0.644605078171228
0.646300567785216	0.651212975213704	0.648529921666387	0.650359318239594	0.64852951966702
0.649543248679811	0.653790983830554	0.639838443346642	0.648087726767313	0.647672503704664
0.652721052502871	0.648742462054785	0.645410694911353	0.649222904245125	0.646812958188061
0.649913832851577	0.651836333793279	0.652074072398783	0.646377202667164	0.645243226749917
0.647001554776387	0.651358380031872	0.649477695283006	0.651732811181579	0.646355697482167

Average: 0.6489777 Median: 0.649066

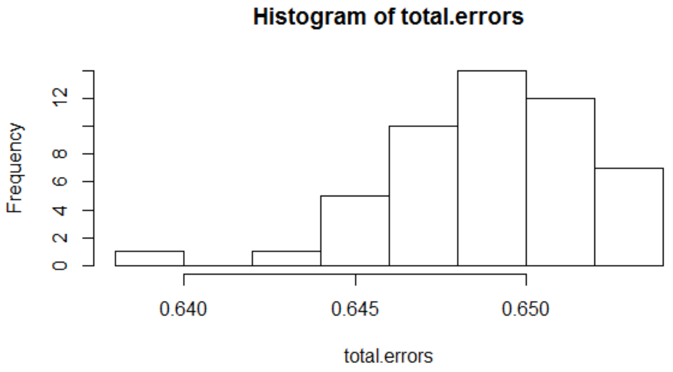

Figure A.27: Histogram of results for the CM experiment on the SCM20d dataset

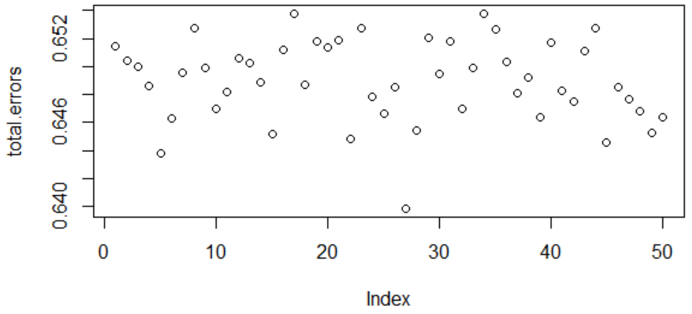

Figure A.28: Plot of results for the CM experiment on the SCM20d dataset

A.15 SCM20d CM w/ perm

Values:

0.649604624763842	0.649658014693305	0.650361072391616	0.647930315621177	0.646627130253374
0.650809333719355	0.648051212204312	0.641906001618606	0.65151631276401	0.647267486653274
0.642202740981521	0.651928639017597	0.647418077333873	0.650199135241616	0.654846982626592
0.651370374906207	0.652321397650345	0.649894519914001	0.651990334593992	0.645827536207657
0.647330753219652	0.647761586629803	0.651563845586518	0.647104097277514	0.653007137029525
0.644572506321966	0.649248164413735	0.6483714178688	0.652897418388787	0.649103012388669
0.647811271058297	0.650680124977018	0.65327999117806	0.644721411281317	0.646854802020814
0.647391142994258	0.65453287294759	0.646515473941817	0.650434835539072	0.64838941287279
0.644994501449078	0.654954278259139	0.649578932915669	0.649251688828074	0.644007050049676
0.648096481336085	0.649070803777339	0.645992114568034	0.647415680535628	0.654395740250897

Average: 0.6490212 Median: 0.6490869

Figure A.29: Histogram of results for the CM / permutation experiment on the SCM20d dataset

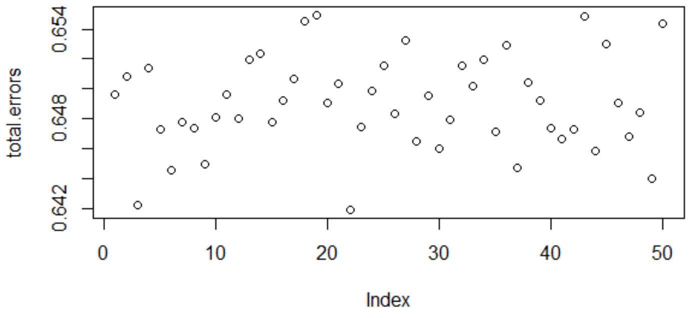

Figure A.30: Plot of results for the CM w/ permutation experiment on the SCM20d dataset

A.16 WQ Baseline

Values:

0.956294961617982	0.957767988512175	0.962402935385027	0.958592824192477	0.962191899763689
0.954388654693089	0.953502491375195	0.951326341431122	0.954763284942171	0.95625148884702
0.961351791221026	0.951175793723259	0.958347022865008	0.956934132238408	0.963905635963547
0.957359444807064	0.957805566498293	0.9597767633723	0.956984477907494	0.956930145410541
0.964855088301531	0.956564719399691	0.963849976323505	0.962205646684101	0.958385819388454
0.960181148807286	0.95873964412247	0.970427040493065	0.956242102559379	0.958529939778871
0.954854527642803	0.956909377764653	0.944402235207839	0.960566426991389	0.951371349036462
0.960242039032906	0.952885324788767	0.956296965228696	0.967821899252801	0.962010385481028
0.956283810843287	0.956849544994479	0.958895102507678	0.958961695729678	0.95807068232644
0.960257946812401	0.960286974261683	0.958618436667211	0.956941833015245	0.96430476352567

Average: 0.9582973 Median: 0.9582089

Figure A.31: Histogram of results for the baseline experiment on the WQ dataset

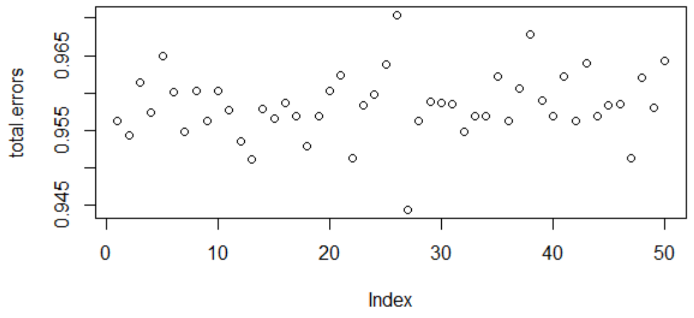

Figure A.32: Plot of results for the baseline experiment on the WQ dataset

A.17 WQ CM

Values:

0.964703005021748	0.956041485055016	0.956272518785221	0.961816643880931	0.957012163630915
0.960007560350917	0.954407198925163	0.955727980496405	0.960469113429564	0.960493524529302
0.954916021320433	0.95666820750204	0.950737847454312	0.960099318161223	0.960370929362055
0.95466477625074	0.962155394789868	0.95663494641413	0.953725698794597	0.95635986341514
0.966054424193374	0.956664510109079	0.954434245709309	0.951336337997848	0.953675137876985
0.960849981485938	0.956291534850853	0.955310397940014	0.954401387465839	0.962135041164629
0.956972217189972	0.955758172946164	0.958727855033737	0.951628731116538	0.959283452383813
0.958012293195973	0.951895590046623	0.956820548585938	0.959541499813187	0.958871590526512
0.957578549747174	0.955931170409493	0.962337846945408	0.961067402643209	0.962089481365633
0.956659744149982	0.961358988716025	0.955958325996449	0.951625632142837	0.960692741809616

Average: 0.957545 Median: 0.9567444

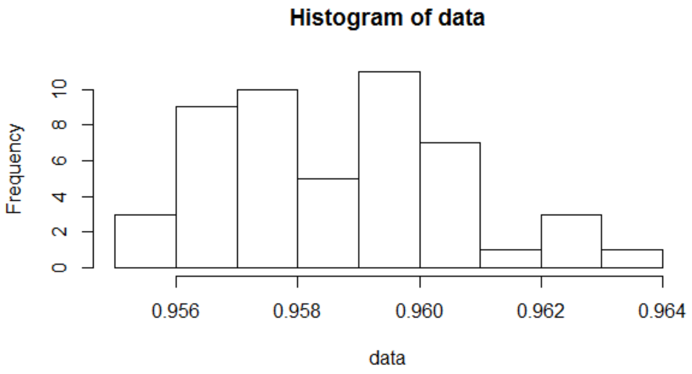

Figure A.33: Histogram of results for the CM experiment on the WQ dataset

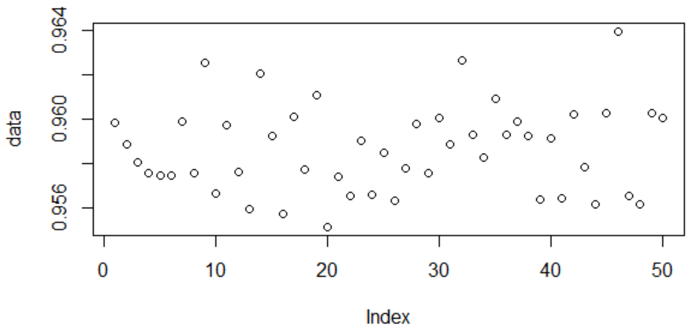

Figure A.34: Plot of results for the CM experiment on the WQ dataset

A.18 WQ CM w/ perm

Values:

0.961833763192779	0.951153929584165	0.95976623535602	0.966012021708669	0.959715576673443
0.957612046935545	0.965814685629295	0.951220004876298	0.964647627993177	0.956493546614955
0.953402779572309	0.955635180896787	0.95381314118927	0.959219292147262	0.963350026711101
0.954084207466969	0.956339758526992	0.958546932449753	0.961926009889363	0.957401219220273
0.953230594313621	0.9658844521868	0.957306776335971	0.949099111581499	0.95949181355997
0.957522926024079	0.964148393830279	0.952757190316483	0.959343514819682	0.959579331599689
0.959267274249541	0.955689277791365	0.955796465817888	0.956384276490186	0.954443980503347
0.956067670255343	0.954232420491844	0.956841872364205	0.962994513796277	0.952507845374743
0.959927665545337	0.956491346965495	0.962066974534295	0.961208637533786	0.964917284015106
0.961017833930434	0.95448364169147	0.959061017451945	0.950393964697652	0.957081319168351

Average: 0.9579445 Median: 0.9574621

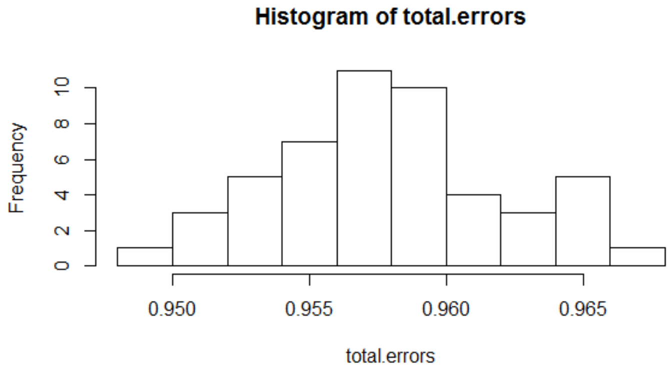

Figure A.35: Histogram of results for the CM w/ permutation experiment on the WQ dataset

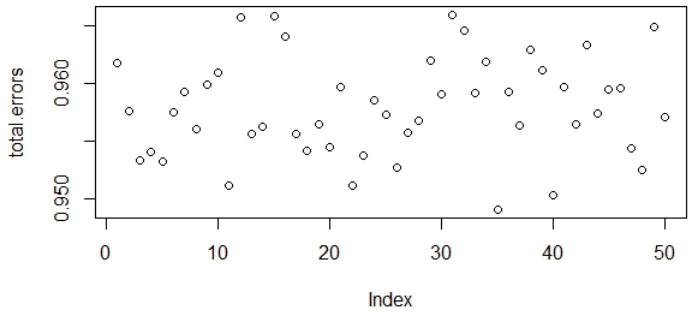

Figure A.36: Plot of results for the CM w/ permutation experiment on the WQ dataset

A.19 RF1 Baseline

Values:

0.537732983193795	0.503531130941061	0.499879063638946	0.523073533384514	0.531001826757023
0.52592608401494	0.534514986088976	0.495388472731722	0.494939740282595	0.485103256681815
0.479262674629944	0.517717455710077	0.458840110466186	0.467491053345681	0.467299858958906
0.494402145141521	0.544830275110345	0.509351473892928	0.524865796617643	0.46912047586164
0.48222614994436	0.501278826971875	0.512979641095606	0.537991166296496	0.487111616158782
0.524558220091234	0.509250805902511	0.528753049122137	0.457973862027165	0.507697450313276
0.510002559521939	0.526828939401086	0.496597118922541	0.501599358213418	0.47751849206664
0.48938492018495	0.48401041086476	0.459372945105474	0.454357242547534	0.501028717509562
0.488368221036994	0.527537779105526	0.517097266089451	0.508225843117569	0.486686758909896
0.486650704584051	0.510559681681218	0.526503526793092	0.570702551989687	0.518559063551491

Average: 0.5031137 Median: 0.5025652

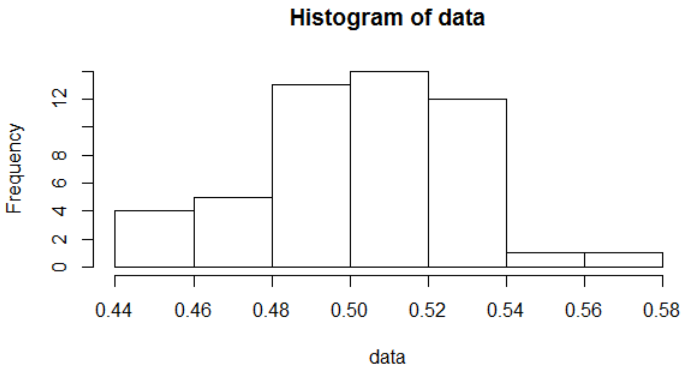

Figure A.37: Histogram of results for the baseline experiment on the RF1 dataset

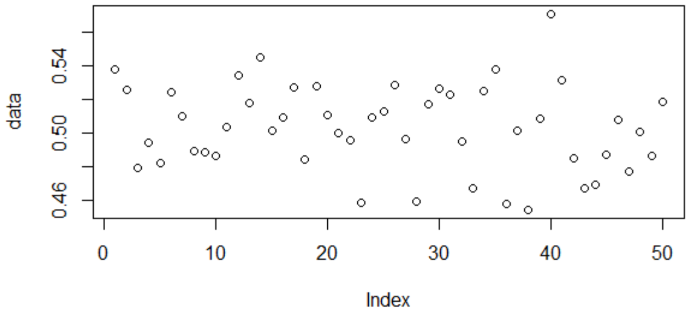

Figure A.38: Plot of results for the baseline experiment on the RF1 dataset

A.20 RF1 CM

Values:

0.569929675955124	0.503686489334862	0.558032982708572	0.491678876566137	0.469427949451189
0.536308467323294	0.51966396300848	0.521690972253921	0.464214823752428	0.535024786957986
0.510199569993838	0.50508956328228	0.542844640592024	0.474483702476933	0.465300415222441
0.538869694975628	0.523296152894997	0.506876130584043	0.52502133256458	0.538265847553607
0.489906078979755	0.532491137097232	0.46682798553689	0.502846366322948	0.497277418842679
0.467429034317346	0.528134564990463	0.50108291421682	0.478619976071453	0.512269461627138
0.475358932681789	0.529276078252845	0.492677868584699	0.570539451549706	0.525150388898654
0.543149742740693	0.467076100432199	0.521307568633181	0.534885972986514	0.488563304292504
0.476108230759504	0.486923742789522	0.485680562387189	0.493597259086235	0.447996567735239
0.524454611278145	0.480858752510629	0.523149231449518	0.552177015023454	0.515488640963094

Average: 0.2292605 Median:0.2308819

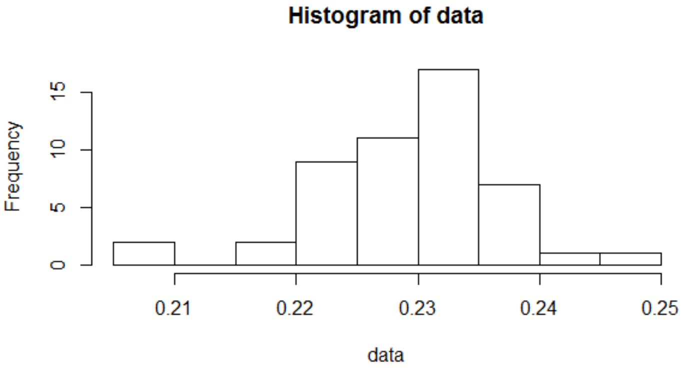

Figure A.39: Histogram of results for the CM experiment on the RF1 dataset

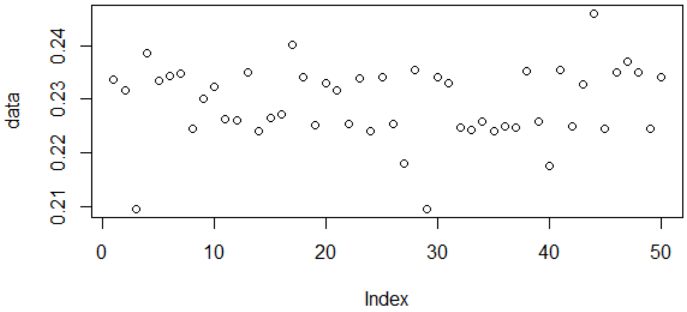

Figure A.40: Plot of results for the CM experiment on the RF1 dataset

A.21 RF1 CM w/ perm

Values:

0.569929675955124	0.503686489334862	0.558032982708572	0.491678876566137	0.469427949451189
0.536308467323294	0.51966396300848	0.521690972253921	0.464214823752428	0.535024786957986
0.510199569993838	0.50508956328228	0.542844640592024	0.474483702476933	0.465300415222441
0.538869694975628	0.523296152894997	0.506876130584043	0.52502133256458	0.538265847553607
0.489906078979755	0.532491137097232	0.46682798553689	0.502846366322948	0.497277418842679
0.467429034317346	0.528134564990463	0.50108291421682	0.478619976071453	0.512269461627138
0.475358932681789	0.529276078252845	0.492677868584699	0.570539451549706	0.525150388898654
0.543149742740693	0.467076100432199	0.521307568633181	0.534885972986514	0.488563304292504
0.476108230759504	0.486923742789522	0.485680562387189	0.493597259086235	0.447996567735239
0.524454611278145	0.480858752510629	0.523149231449518	0.552177015023454	0.515488640963094

Average: 0.5082242 Median: 0.5085379

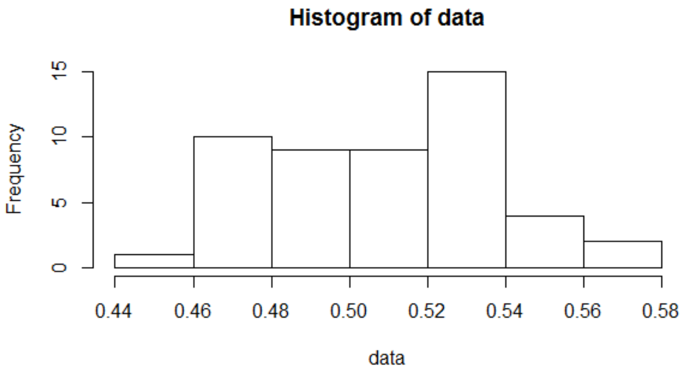

Figure A.41: Histogram of results for the CM w/ permutation experiment on the RF1 dataset

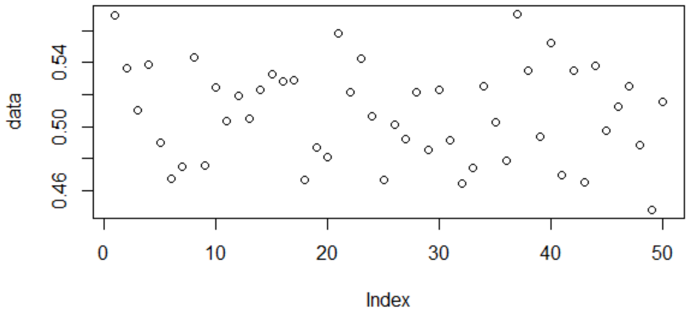

Figure A.42: Plot of results for the CM w/ permutation experiment on the RF1 dataset

A.22 RF2 Baseline

Values:

0.237494754324039	0.230996001175867	0.232903337893892	0.234610437760869	0.234012119078333
0.218817075471922	0.233615708626302	0.233812268131301	0.217114690929183	0.218354147855057
0.233182294462292	0.235321296008601	0.233376334945488	0.224208263908769	0.233277070212033
0.234716071387923	0.217670021437173	0.23372374911808	0.234831385534083	0.23372894677264
0.225606529396129	0.235811913932164	0.235623331093342	0.238223397377412	0.22542463140224
0.232960013277212	0.235879129337119	0.22465519177052	0.226612978030248	0.218201953862042
0.235743045100363	0.233416440471803	0.233581257054085	0.231597645900285	0.227460316435944
0.234952763199435	0.232504579034379	0.225934558174311	0.235756928483429	0.224877369087393
0.233621739296542	0.224475739290561	0.226944896254201	0.233843209569446	0.2406885394866
0.236430865131883	0.233973759546305	0.216522600658627	0.225058551121297	0.234930118257156

Average: 0.2305416 Median: 0.2333964

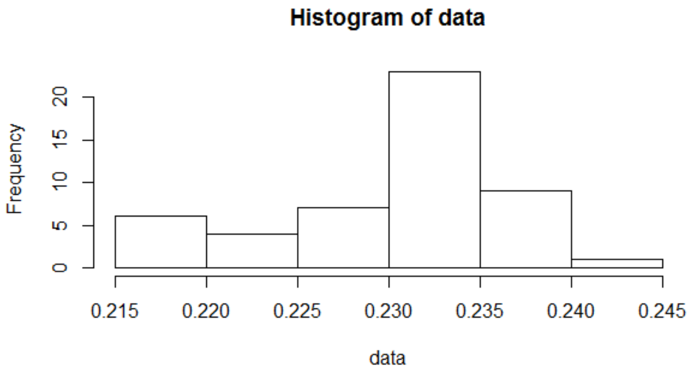

Figure A.43: Histogram of results for the baseline experiment on the RF2 dataset

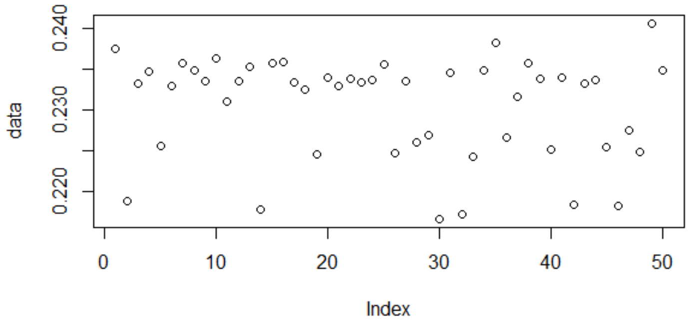

Figure A.44: Plot of results for the baseline experiment on the RF2 dataset

A.23 RF2 CM

Values:

0.239903891474493	0.225362545666868	0.234282125628001	0.231124271184871	0.225432564566004
0.234417532815061	0.22606381459763	0.235085803530263	0.228133606330362	0.226120352247019
0.225398872179802	0.216807291406707	0.228025140252155	0.234311838049458	0.216417699697481
0.225724627960876	0.236180087265443	0.225994814885624	0.218624212954599	0.23407240801394
0.227442737245754	0.237959265954285	0.233087629009639	0.226217164423267	0.232711883554209
0.226011390378813	0.226144424461956	0.235405985575642	0.23334890343006	0.234947703975953
0.23287924872852	0.242398292557939	0.224992177619993	0.223998741544628	0.232770225855714
0.228053233607526	0.21704637042779	0.228706483857434	0.225680728423079	0.232843892131374
0.233894924491539	0.216830177531201	0.228132573220742	0.234880790828997	0.225654517060202
0.2275225986765	0.23430416180814	0.232240764187837	0.211763438444631	0.225396549931305

Average: 0.228815 Median: 0.2280929

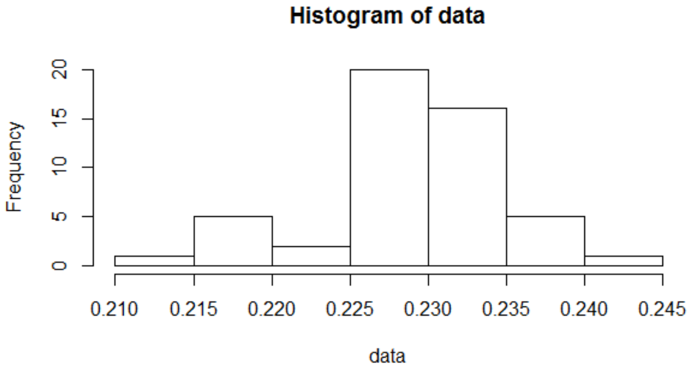

Figure A.45: Histogram of results for the CM experiment on the RF2 dataset

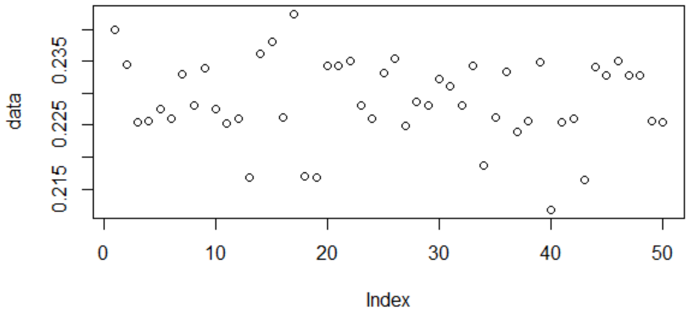

Figure A.46: Plot of results for the CM experiment on the RF2 dataset

A.24 RF2 CM w/ perm

Values:

0.224932601064087	0.225242252263537	0.233408450597302	0.226480881390326	0.233621221009131
0.225875736614233	0.231767261992016	0.22496240421588	0.227034761577026	0.234247462575141
0.225330101011183	0.233176775806898	0.233815984025656	0.232395701998952	0.231931119957271
0.235817632529384	0.218440160304762	0.21713652316741	0.233750996781229	0.232805351965844
0.232928901665712	0.226446018331118	0.234948573768106	0.235695713166413	0.235732754299236
0.23186853806809	0.225829093842361	0.225849482070777	0.216705852405661	0.23269258423939
0.227150330394512	0.22656188871866	0.233458726993886	0.234422678819928	0.227258598960971
0.237127529929324	0.232604551140931	0.225792030068414	0.216684342615172	0.23273888939105
0.234304669733901	0.226017981457417	0.23583587443594	0.234091488524296	0.224551838426369
0.228074141527139	0.234354011629881	0.233342712963118	0.232791464593311	0.224090027655083

Average: 0.2296425 Median: 0.2321634

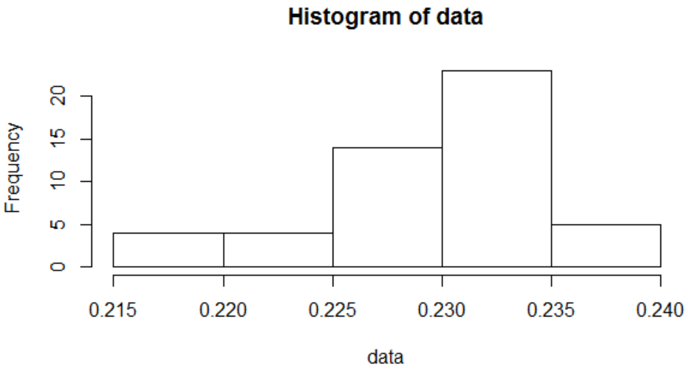

Figure A.47: Histogram of results for the CM w/ permutation experiment on the RF2 dataset

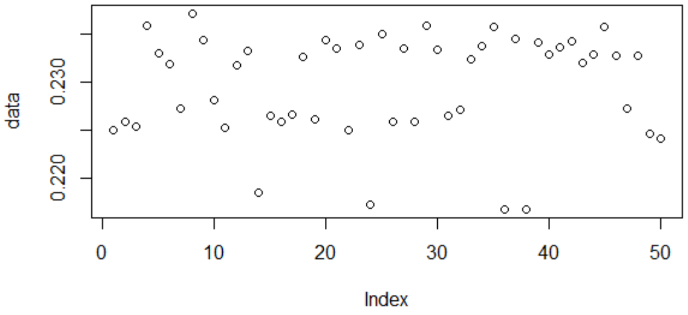

Figure A.48: Plot of results for the CM w/ permutation experiment on the RF2 dataset