STA5076Z: SUPERVISED LEARNING - PROJECT

Corne Oosthuizen - OSTAND005

Due: 30 July 2017

Table of Contents

1 - Handwritten Digit Classification

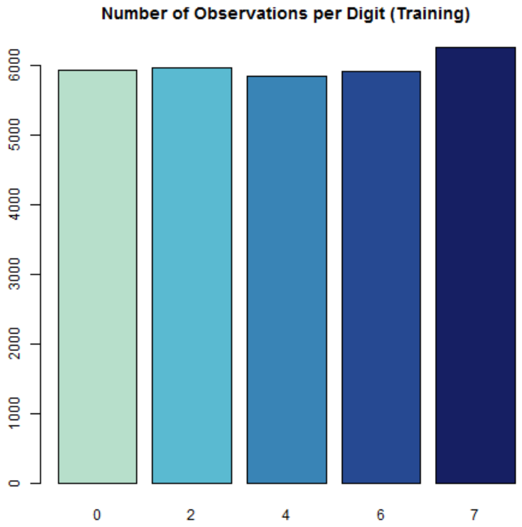
Problem Statement

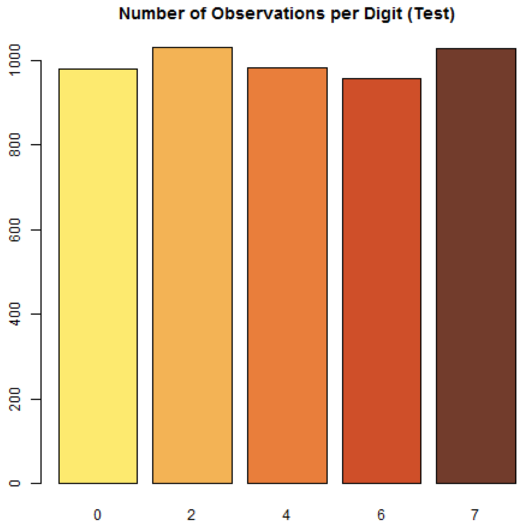
The MNIST (Modified National Institute of Standards and Technology) dataset comprises tens of thousands of grey-scale, handwritten digits that have been size-normalised and centred in a fixed-size image. Each image is 28 pixels in height and width, with 784 pixels in total. Each pixel has an integer value ranging from 0 to 255 indicating the darkness of that pixel, where darker pixels have larger numbers. Each image has also been labelled with the digit that it represents. Your goal is to develop an accurate classifier that can be used to identify the digit in a new handwritten image[1].

The dataset as provided on Vula resources contained a training (60,000 observations) and test (10,000 observations) set. Each observation describes a single digit with the label defining what the pixels represent, followed by the 784 pixels comprising of the **28x28** image (the pixels are described in columns named pix_, where is the row and is the column of the pixel).

label | pix1_1 | pix1_2 | ... | pix28_27 | pix28_28

The following digits where assigned to me for this project:





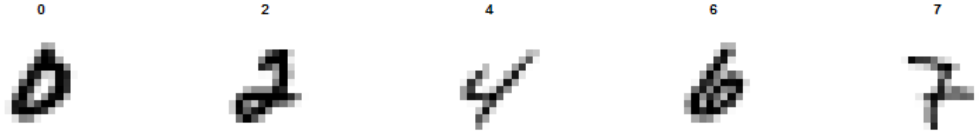
DIGIT	TRAIN	TEST
0	5923	980
2	5958	1032
4	5842	982
6	5918	958
7	6265	1028
TOTAL:	29906	4980

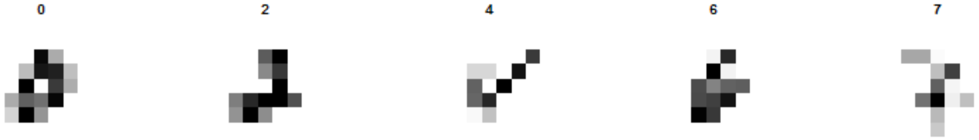
Data Trasformations

The inital dataset contains images comprising of **28x28** pixels, as shown below:

Initial Digits

My initial thought was that it might be beneficial to reduce the images (scale) to a smaller size to therefore also reduce the number of variables (pixels) that would have to be used in the learning models. The scalled versions (**14x14** and **7x7**) of the above images are shown below:

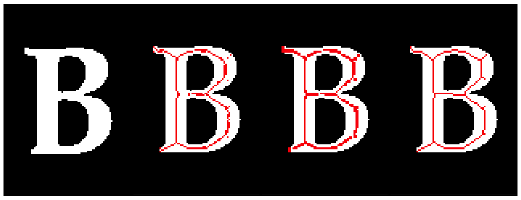




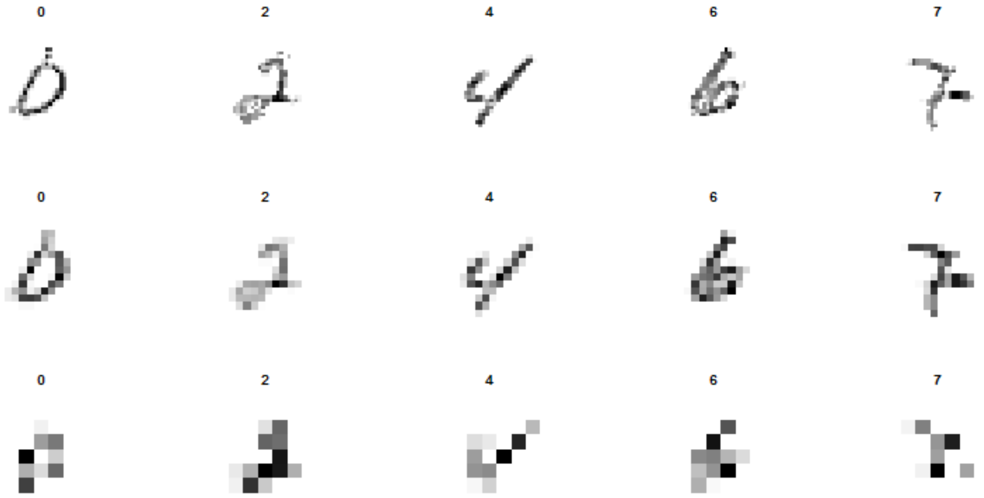
It is clear that with scaling there will be some loss of data but the digits are still recognisable at **14x14**, so that is the size we can use for the analysis. The other sizes are still generated for all the subsequent transformations which will allow comparison at a later stage.

Some of the given digits have some undersirable charactericstics if they are to be recoginised by a digital process, one of these is that the line thickness of the digits are not equal and because we don't really care about the thickness but only the shape it would be good to apply a shape analysis process called topological skeletonisation to each digit [2], the process involves the thinning of a shpae to the medial line that aproximates the path of the original.The skeletonised version of the character is shown in red.

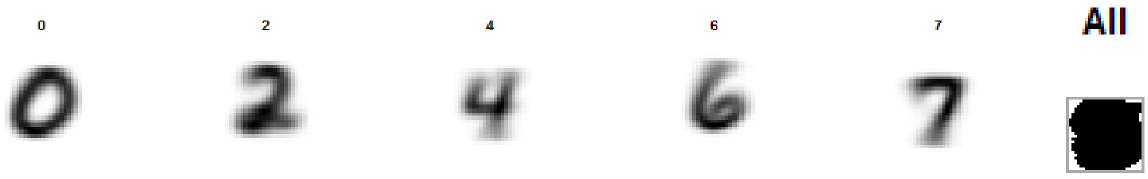
Skeletonisation Example



To apply this filter to the digits the best result was obtained by first smoothing the digit and then applying the skeletonise method, thus resulting in the following:



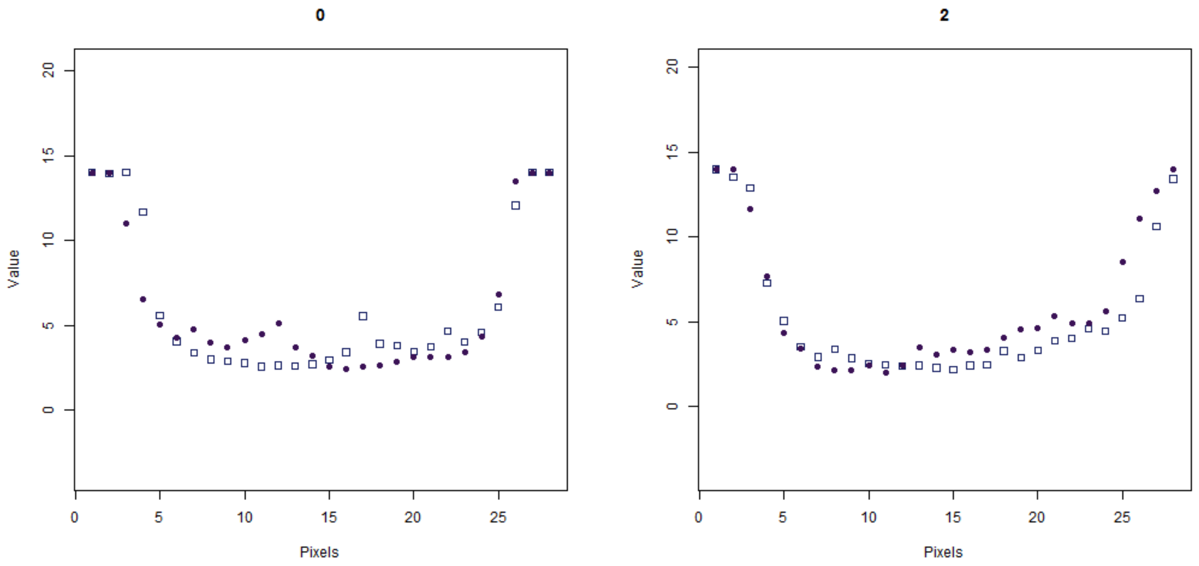
The next step was traying to look at reducing the number of variables by looking at cropping the images. To see if this was possible the average pixel density for all the digits in the dataset would give us the following:

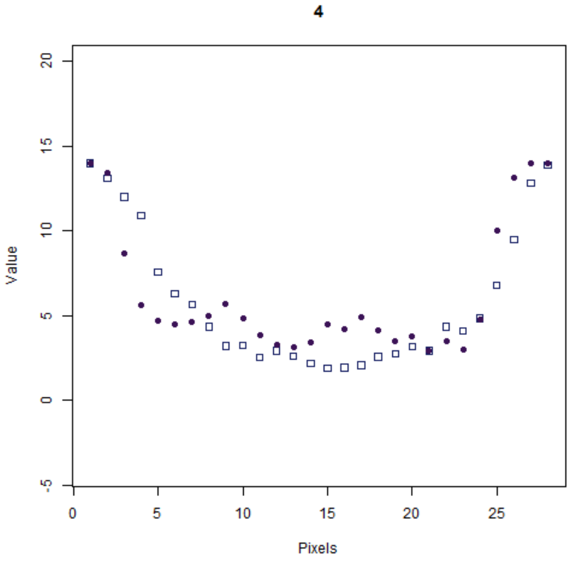


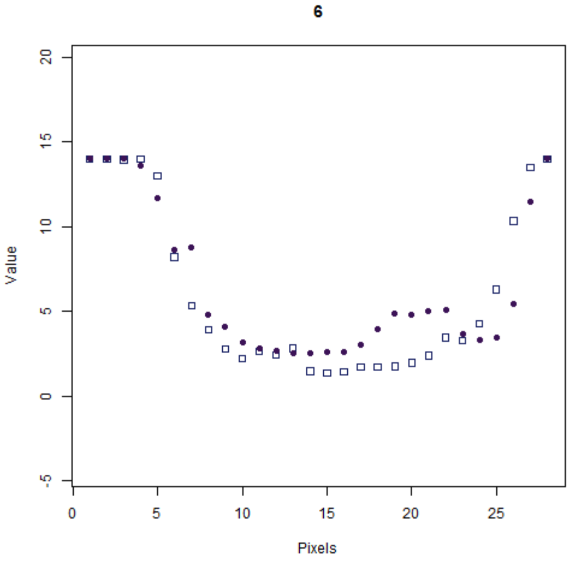
Digits - Average

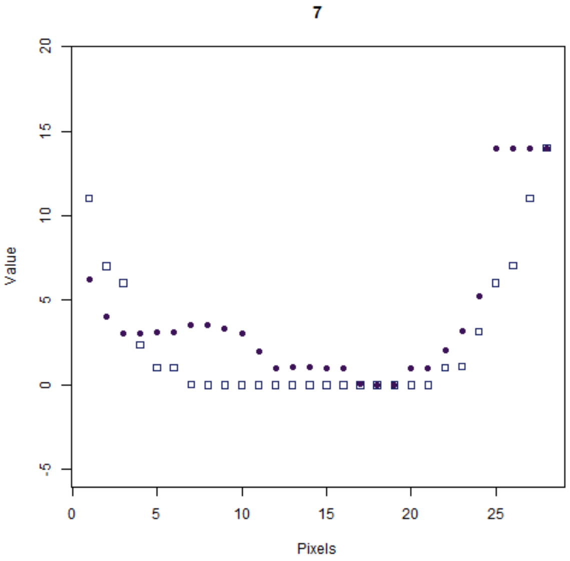
The last image (on the right) shows the combined average pixel density (with added exageration of the colour depth to be able to see if cropping can be used), unfortunately the image indicates that cropping specific columns might skew the pixel data. This then would not work for reducing the dataset.

Through some research refering to additional image transformations and digit recognition techniques[4] it might be possible to increase the accuracy of the models by adding in the symmetry analysis of a digit to improve the description of the shape[5]. The graph showing the symmetry analysis of our chosen 5 digits are as follows:

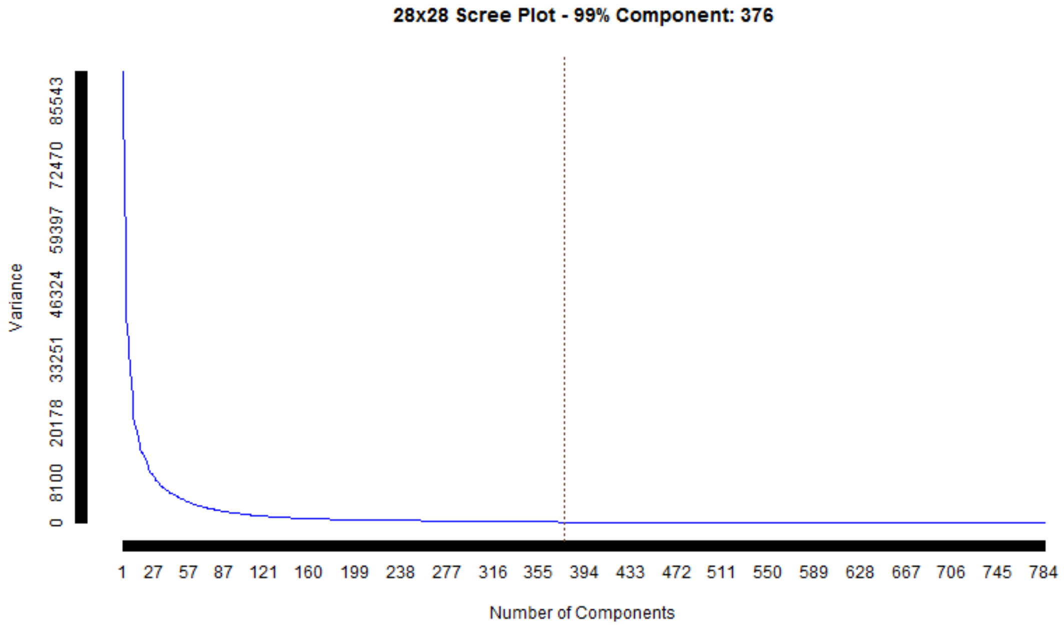


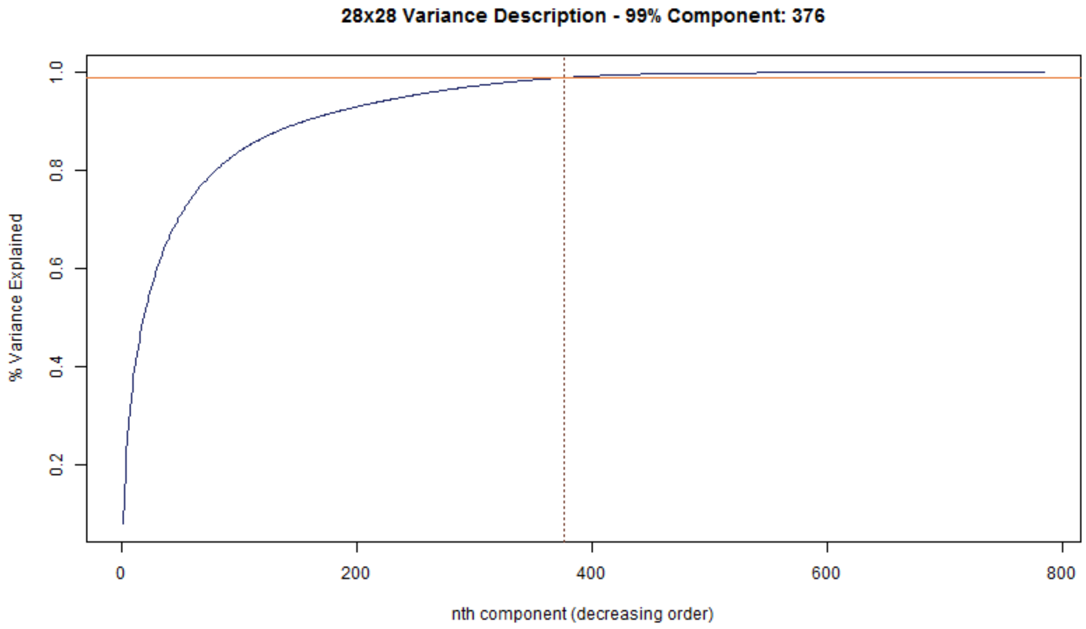


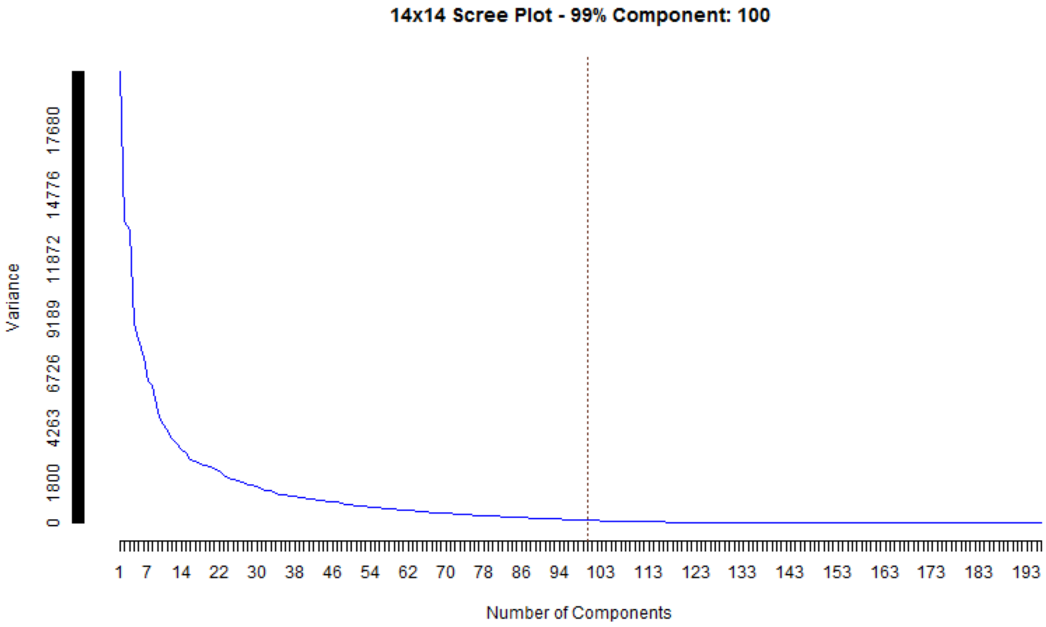


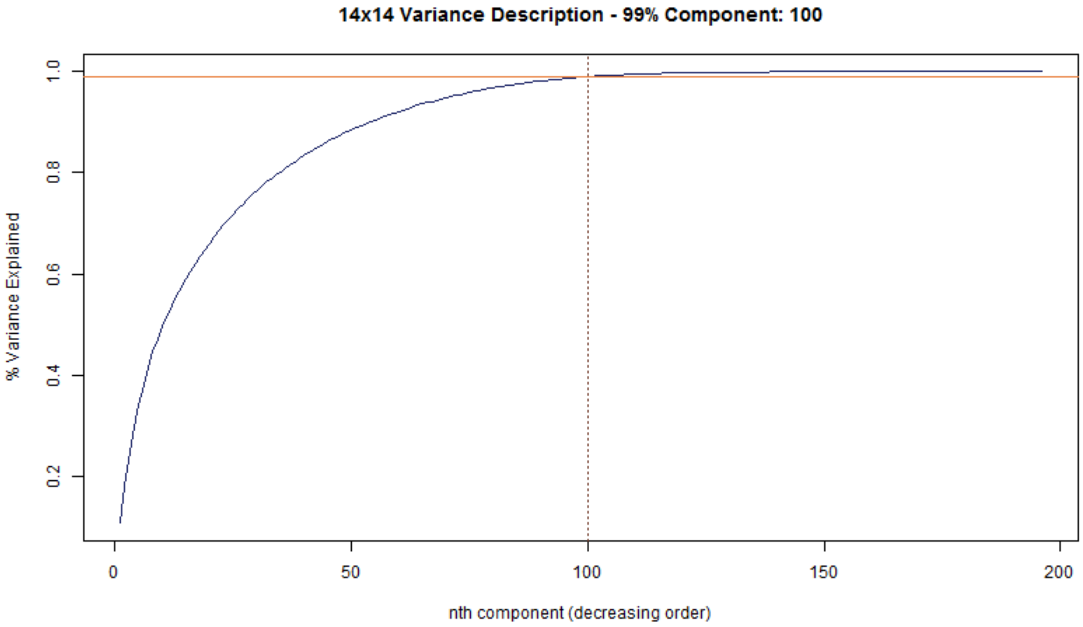


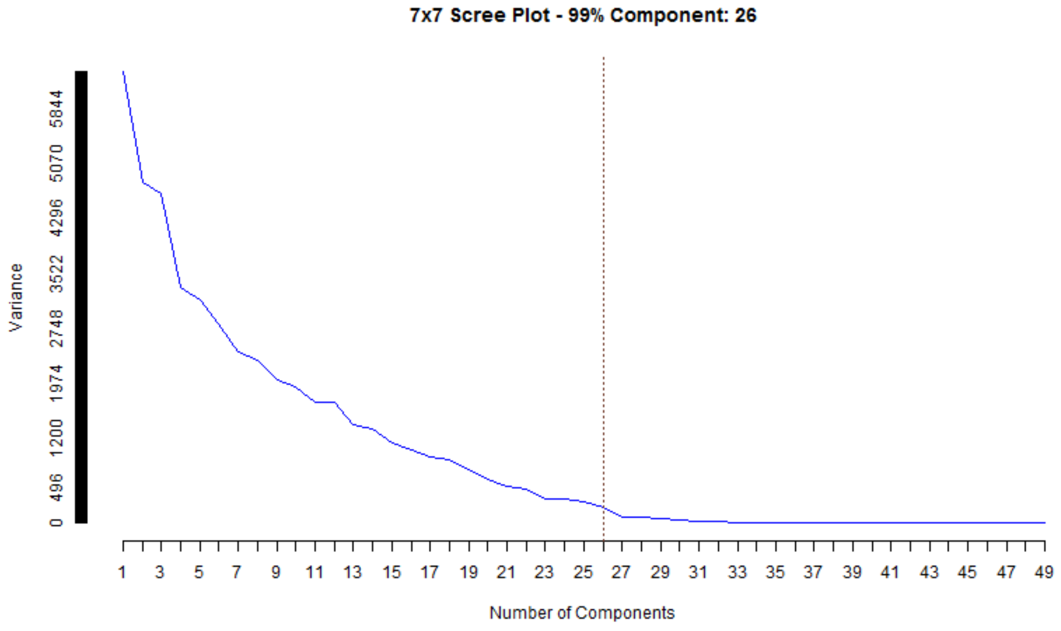
Now that there is a reasonable set of data for the digits we can run a Principal Component Analysis (PCA) on each one as some of the supervised techniques to follow require the variables to be uncorrelated. The selection for the PCA variance is set to use the components that describe 99% of the variance.

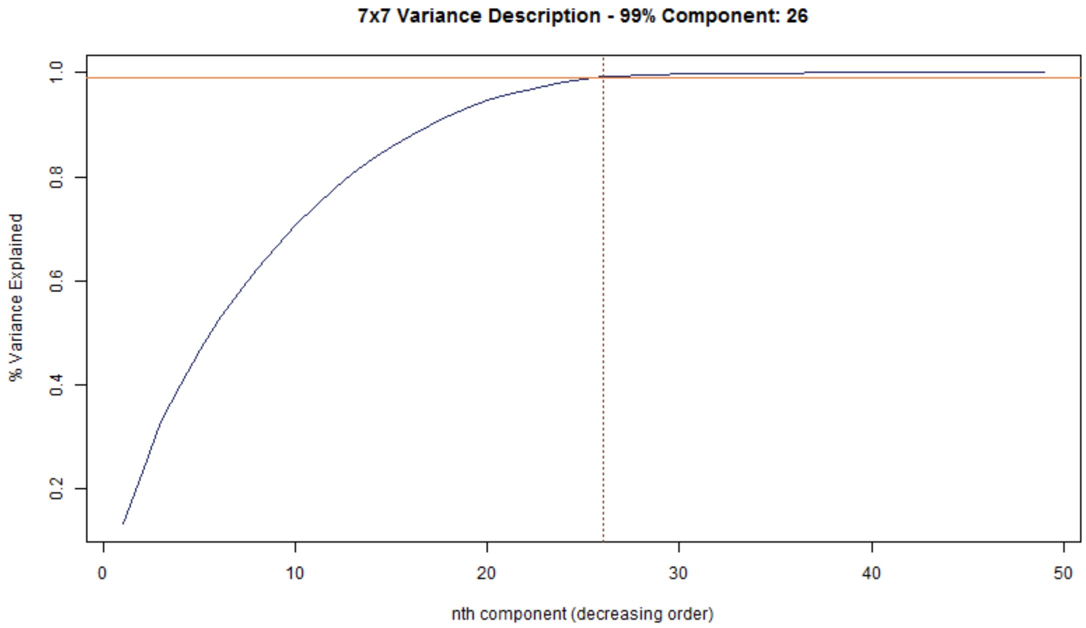

28x28 Scree Plot - 99% Component: 376
Variance
0
8100
20178
33251
46324
59397
72470
85543
1
27
57
87
121
160
199
238
277
316
355
394
433
472
511
550
589
628
667
706
745
784
Number of Components


28x28 Variance Description - 99% Component: 376
% Variance Explained
0.2
0.4
0.6
0.8
1.0
0
200
400
600
800
nth component (decreasing order)


14x14 Scree Plot - 99% Component: 100
Variance
0
1800
4263
6726
9189
11872
14776
17680
1
7
14
22
30
38
46
54
62
70
78
86
94
103
113
123
133
143
153
163
173
183
193
Number of Components


14x14 Variance Description - 99% Component: 100
% Variance Explained
0.2
0.4
0.6
0.8
1.0
0
50
100
150
200
nth component (decreasing order)



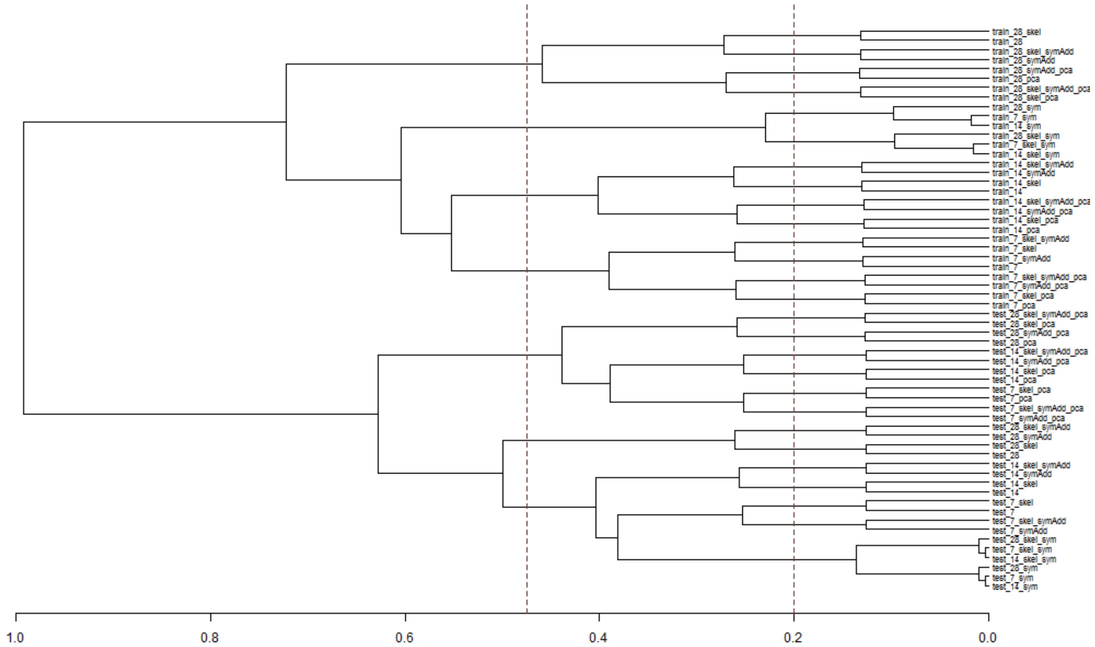


The dataset generation takes quite a while to run as it transforms the raw training and test data with the described transformations while document the execution time and the resulting size of the data objects. The final number of files is **140** as each dataset is stored as a comma delimited file (csv) and a R object store (rds), the PCA vector is also stored for later re-use.

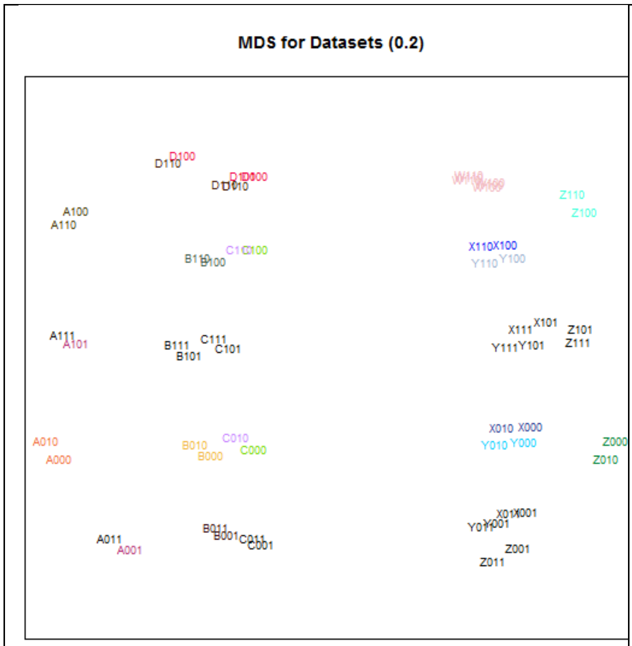
The final set of possible data sets to run our analysis on looks like:

The table describes the name, type, transformation actions performed on that dataset, the relevant size (dimensions and bytes), and the relative time it took to produce the set.

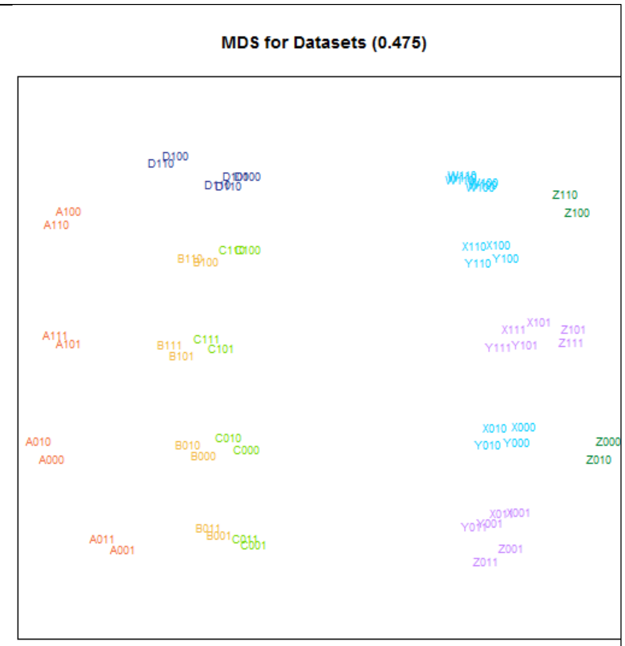
Running a cluster analysis and visualising the distribution with Multidimentional scaling gives:



Dendogram Describing the various datasets, lines at h=0.2 and h=0.475



Multi-Dimensional Scaling at h=0.2



Multi-Dimensional Scaling at h=0.475

The label is a composition of the the type and size (table below), symmetry (binary), skeleton (binary), and PCA (binary). So For example the cluster that can be seen on the top left, A100 and A110 describes a training with size 28x28, symmetry and A110 has skeleton applied to the digit. Below that we can see A111 and A101 where the same kind of grouping exists but with PCA applied. We can see this kind of grouping continues with most of the other images sizes and dataset types. Intresting is that on images sizes of 14x14 and 7x7 the clusters seem to be closer together than their bigger versions.

SIZE	TRAIN	TEST
28	A	Z
14	B	Y
7	C	X
0	D	W

Model Selection

This problem has been explored many times before [10] and looking at the leader board directions it is most likely that because the data cannot be described in a linear model a higher predeictive accuracy will be obtained by using models that use higher dimensional mappings, but losing out on interprability. The models selected can then be used to run the various tranformed datasets and see if the transformations improve accuracy in each case.

KNN

Implemented using the `caret train` method and using a repeated cross validation, a cross validation method repeated 4 times each with 5 folds. The best fitted model returns:

```
k-Nearest Neighbors

29906 samples
   98 predictor
    5 classes: '0', '2', '4', '6', '7'

No pre-processing
Resampling: Cross-Validated (5 fold, repeated 4 times)
Summary of sample sizes: 23926, 23925, 23925, 23924, 23924, 23924, ...
Resampling results across tuning parameters:

  k  Accuracy  Kappa
  5  0.991     0.989
  7  0.990     0.988
  9  0.990     0.987

Accuracy was used to select the optimal model using  the largest value.
The final value used for the model was k = 5.
```

Based on this running the algorithm with k = 5 as the selected hyperparameter, run that against the un-transformed dataset (A000/Z000) and then on (B111/Y111)

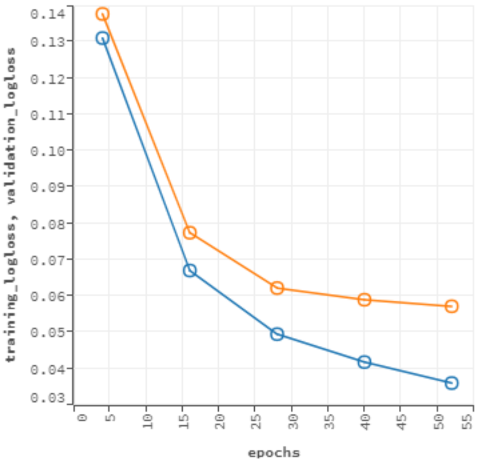
```
Confusion Matrix and Statistics (A000/Z000) - RunTime:  143.619828939438 (2min 24sec)

          Reference
Prediction    0    2    4    6    7
         0  974   11    3    6    0
         2    2  999    1    0    6
         4    0    2  972    3    7
         6    3    1    4  949    0
         7    1   19    2    0 1015

Overall Statistics

               Accuracy : 0.986
                 95% CI : (0.982, 0.989)
    No Information Rate : 0.207
    P-Value [Acc > NIR] : <0.0000000000000002

                  Kappa : 0.982
 Mcnemar's Test P-Value : NA
```

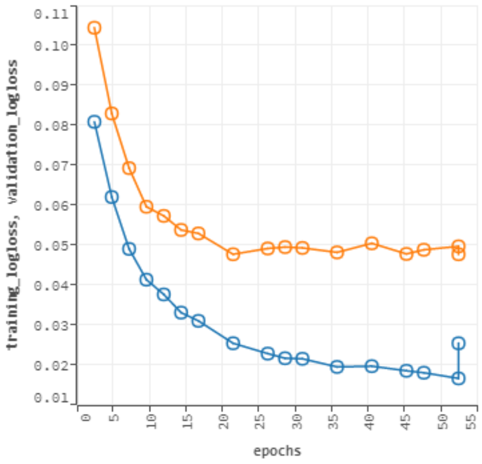
```
Confusion Matrix and Statistics (B111/Y111) - RunTime:  10.498685836792 (10sec)

          Reference
Prediction    0    2    4    6    7
         0  974    8    0    4    0
         2    2 1004    0    0    5
         4    0    1  974    2    6
         6    3    1    6  952    0
         7    1   18    2    0 1017

Overall Statistics

               Accuracy : 0.9882
                 95% CI : (0.9847, 0.991)
    No Information Rate : 0.2072
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9852
 Mcnemar's Test P-Value : NA
```

In this case th transformed dataset improved accuracy by *0.22%* and the training time was reduced by *133.12* seconds.

Neural Network

The Neural network was trained with the same datasets the training data was split into a training and validation set (80/20) and the test set was used for predictions and reporting.





The Graphs show the training grid for BY111 and AZ000 indicating the drop in error rate between the training and validation sets.

```
Confusion Matrix and Statistics (AZ000) - RunTime:  126.3698 (2min 6sec)

          Reference
Prediction    0    2    4    6    7
         0  965    5    1    6    1
         2    1 1007    6    3   16
         4    2    4  964    5    3
         6    9    7    3  944    0
         7    3    9    8    0 1008

Overall Statistics

               Accuracy : 0.982
                 95% CI : (0.977, 0.985)
    No Information Rate : 0.207
    P-Value [Acc > NIR] : <0.0000000000000002
```

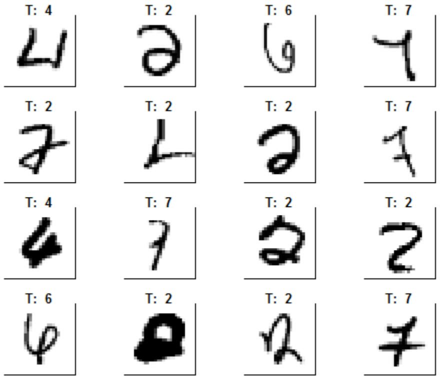
```
Confusion Matrix and Statistics - (BY111) - RunTime:  25.94692 (26sec)

          Reference
Prediction    0    2    4    6    7
         0  969    6    1    9    2
         2    4 1012    9    1   25
         4    0    1  961    4    6
         6    6    5    7  943    0
         7    1    8    4    1  995

Overall Statistics

               Accuracy : 0.98
                 95% CI : (0.976, 0.984)
    No Information Rate : 0.207
    P-Value [Acc > NIR] : <0.0000000000000002

                  Kappa : 0.975
 Mcnemar's Test P-Value : 0.0133
```

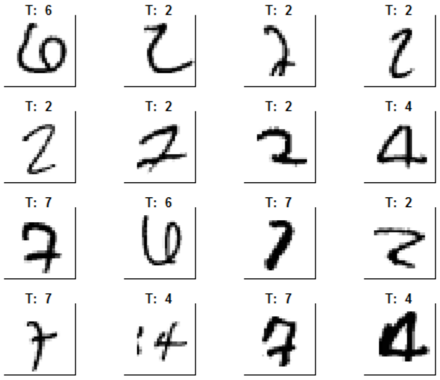
The accuracy did not improve at all and seems to have slightly worstened, although the training time was reduced by *100.43* seconds.

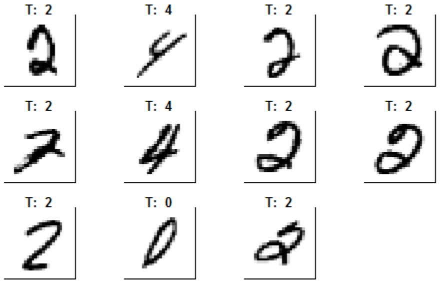
KNN (AZ000) **KNN (AZ000)** **NN (Z000)** **NN (BY111)**

71	59	92	100

This still results in several classification errors which looked at the following digits 43 are the ones that is generally wrongly identified:







Conclusion

The models used in this example gives a quite reasonable accuracy (around *98.41*%) which can be improved by using an ensemble method that will take the most "voted" for digit and return that result, this will push the accuracy up to *99.1*4%.

The selection of models, which can be expanded to include Random Forests, K-means, and Support Vector Maschine (SVM); and the breatdh of the dataset searching could notbe fully explored before the due date of this report.

2 - Student Performance Data Set

Problem Statement

The student performance[6][7] for a set of students are described in the dataset with demographic information pertaining to each one. There are two subjects recorded Math and Portugese. Given the demographic information, access to study information, willingness to study and available hours (free and social) what is the likely score for their first and second semester and then their final expected grade.

Data Description

Variable	Description
school	student’s school (binary: ‘GP’ - Gabriel Pereira or ‘MS’ - Mousinho da Silveira)
sex	student’s sex (binary: ‘F’ - female or ‘M’ - male)
age	student’s age (numeric: from 15 to 22)
address	student’s home address type (binary: ‘U’ - urban or ‘R’ - rural)
famsize	family size (binary: ‘LE3’ - less or equal to 3 or ‘GT3’ - greater than 3)
Pstatus	parent’s cohabitation status (binary: ‘T’ - living together or ‘A’ - apart)
Medu	mother’s education (numeric: 0 - none, 1 - primary education (4th grade), 2 – 5th to 9th grade, 3 – secondary education or 4 – higher education)
Fedu	father’s education (numeric: 0 - none, 1 - primary education (4th grade), 2 – 5th to 9th grade, 3 – secondary education or 4 – higher education)
Mjob	mother’s job (nominal: ‘teacher’, ‘health’ care related, civil ‘services’ (e.g. administrative or police), ‘at_home’ or ‘other’)
Fjob	father’s job (nominal: ‘teacher’, ‘health’ care related, civil ‘services’ (e.g. administrative or police), ‘at_home’ or ‘other’)
reason	reason to choose this school (nominal: close to ‘home’, school ‘reputation’, ‘course’ preference or ‘other’)
guardian	student’s guardian (nominal: ‘mother’, ‘father’ or ‘other’)
traveltime	home to school travel time (numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour)
studytime	weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours)
failures	number of past class failures (numeric: n if 1<=n<3, else 4)
schoolsup	extra educational support (binary: yes or no)
famsup	family educational support (binary: yes or no)
paid	extra paid classes within the course subject (Math or Portuguese) (binary: yes or no)
activities	extra-curricular activities (binary: yes or no)
nursery	attended nursery school (binary: yes or no)
higher	wants to take higher education (binary: yes or no)
internet	Internet access at home (binary: yes or no)
romantic	with a romantic relationship (binary: yes or no)
famrel	quality of family relationships (numeric: from 1 - very bad to 5 - excellent)
freetime	free time after school (numeric: from 1 - very low to 5 - very high)
goout	going out with friends (numeric: from 1 - very low to 5 - very high)
Dalc	workday alcohol consumption (numeric: from 1 - very low to 5 - very high)
Walc	weekend alcohol consumption (numeric: from 1 - very low to 5 - very high)
health	current health status (numeric: from 1 - very bad to 5 - very good)
absences	number of school absences (numeric: from 0 to 93)

Variable	Description
G1	first period grade (numeric: from 0 to 20)
G2	second period grade (numeric: from 0 to 20)
G3	final grade (numeric: from 0 to 20, output target)

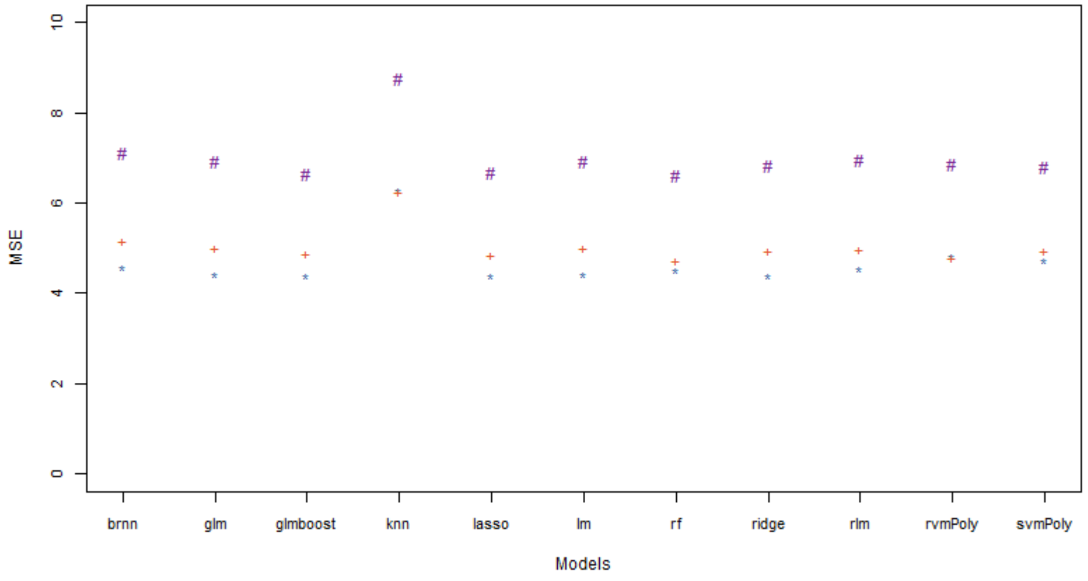
The data variables span a range of demographic data, some of which might be applicable to the results that we would like to predict.

Using a list of supervised methods

The Math dataset only contains 396 observations whereas the Portuguese dataset contains 649 observations, there are students that have participated in both courses. For the training and test set I've limited it to just the Portugese set which is then subdivided into a training and test set on a random selected 75% split. This results in a training set of 488 observations and test set of 161 observations. The training set is then run in a repeated cross fold validation, for this configuration that was set to repeat 4 times and each time do 4 folds. The library used was the `caret` package which contains a training wrapper function for a comprehensive list of learning models, the model selection[8] was based on the algorithms covered in class and additionally intrest. The training was repeated to predict the first semester (G1 - *), second semester (G2 - +) and final mark (G3 - #). The tale also includes the time it took to train each model in seconds and is sorted on lowest Mean Square Error (MSE) of the final mark (lowest to highest). The MSE for each predictor is calculated only on the testing set that was set aside in the beginning.

Name	Method	G1.MSE	G2.MSE	G3.MSE	G1.RMSE	G2.RMSE	G3.RMSE	G3.Accurracy	G1.Time	G2.Time	G3.Time	Size
rf	Random Forest	4.52	4.74	6.64	2.13	2.18	2.58	87.1	52.614	52.274	51.90	5.312
glmboost	Boosted Generalized Linear Model	4.40	4.90	6.66	2.10	2.21	2.58	87.1	1.855	1.973	2.08	0.840
lasso	The lasso	4.42	4.86	6.70	2.10	2.20	2.59	87.1	1.323	1.283	1.48	0.238
svmPoly	Support Vector Machines with Polynomial Kernel	4.73	4.94	6.82	2.17	2.22	2.61	86.9	18.632	17.978	17.48	0.416
ridge	Ridge Regression	4.41	4.95	6.85	2.10	2.23	2.62	86.9	2.275	2.337	2.72	0.226
rvmPoly	Relevance Vector Machines with Polynomial Kernel	4.83	4.79	6.88	2.20	2.19	2.62	86.9	133.056	129.924	129.56	0.408
glm	Generalized Linear Model	4.44	5.02	6.94	2.11	2.24	2.63	86.8	1.010	1.028	1.35	1.069
lm	Linear Regression	4.44	5.02	6.94	2.11	2.24	2.63	86.8	0.803	0.833	1.23	0.696
rlm	Robust Linear Model	4.54	4.97	6.96	2.13	2.23	2.64	86.8	2.971	3.327	3.77	0.921
brnn	Bayesian Regularized Neural Networks	4.60	5.16	7.13	2.15	2.27	2.67	86.6	19.522	19.086	21.66	0.391
knn	k-Nearest Neighbors	6.29	6.26	8.77	2.51	2.50	2.96	85.2	0.724	0.764	1.14	0.394

All the models selected seem to converge to a similar Root Mean Square Error (RMSE). This might be the case because of the small number of observation in our training set. The Random Forest [rf] model results in the best result for the final predictor G3 and is generally quite good accross G2 and G1. The Boosted Generalized Linear Model [glmboost] (fitting generalized linear models by likelihood based boosting) performs similarly well and slightly better than the Generalized Linear Model [glm]. By far the worst model looks to be k-Nearest Neighbors, it has the largest error on all of the predictors although being one of the fastest models to train.



Student Model Comparison

General output:

```
488 samples
 30 predictor

No pre-processing
Resampling: Cross-Validated (4 fold, repeated 4 times)
Summary of sample sizes: 366, 366, 367, 365, 366, 367, ...
```

Looking at the **`rf`**, **`glmboost`**, **`lasso`** more closely:

```
Random Forest

Resampling results across tuning parameters:

  mtry  RMSE  Rsquared
   2    2.82  0.308
  20    2.72  0.309
  39    2.75  0.299

RMSE was used to select the optimal model using  the smallest value.
The final value used for the model was mtry = 20.

Boosted Generalized Linear Model

Resampling results across tuning parameters:

  mstop  RMSE  Rsquared
   50    2.84  0.255
  100    2.80  0.271
  150    2.79  0.272

Tuning parameter 'prune' was held constant at a value of no
RMSE was used to select the optimal model using  the smallest value.
The final values used for the model were mstop = 150 and prune = no.

The lasso

Resampling results across tuning parameters:

  fraction  RMSE  Rsquared
  0.1       3.01  0.206
  0.5       2.78  0.272
  0.9       2.83  0.261

RMSE was used to select the optimal model using  the smallest value.
The final value used for the model was fraction = 0.5
```

The **`caret train`** method which sets up a grid of tuning parameters for a few classification and regression routines, fits each model and calculates a resampling based performance measure; can with some time be used to further explore the hyperparameters for the top models.

Conclusion

The models did all converge on similar error rate and accuracy, all of which could be improved if more observations where obtainable. Generally the results for G1 (*) and G2 (+) are a lot better than for G3 (#) and it might be better to create an ensemble between the top 3 or 4 models for each predictor. The best model to select for this problem is likely to be either the Boosted Generalized Linear Model [glmboost] or the Lasso [lasso] method, both display a resonable level of accuracy but significantly faster training times than the Random Forest [rf]. In terms of final model size the Lasso [lasso] method produces a much smaller object that would be easily reproducable.

3 - Notes

The project was helpfull to understand the Supervised Learning methods available and how they can be practically applied to regression and classification problems. The time spend on changing and transforming the MNIST didigt data was far more than I expected and that impacted the approach to the number and scale of the models that could be run on that dataset.

The selection of the student record dataset was a little smaller than expected but would be good start for approaching the "Open University Learning Analytics dataset Data Set"[9] which contains a much more comprehensive set of variables and a much larger set of observations.

GitHub: **https://github.com/TurRil/STA5076Z-Supervised-Learning**

4 - References

- Project Description (SupLearnProject2017.pdf), Author: Miguel Lacerda. Last Accessed: 19-May-2017. Vula Resources.
- Skeletonisation, Author: Jon Clayden. Last Accessed: 28-June-2017. URL: **https://www.rdocumentation.org/packages/mmand/versions/1.5.0**
- Image: Skel.png. Last Accessed: 28-June-2017. URL: **https://commons.wikimedia.org/wiki/File:Skel.png**
- Classification and Image Processing on MNSIT Data Set, Author: Andrew Halsaver, Brian Becker, & Farshad Chitchian. Last Accessed: 28-June-2017. URL: **http://bribecker.github.io/img/handWrittenDigitClassification.pdf**
- MNIST Dataset: Classifying Images, Author: Brian Becker. Last Accessed: 28-June-2017. URL: **http://bribecker.github.io/blog/MNIST-3/**
- Student Performance Data Set, Author: Paulo Cortez, University of Minho, GuimarÃ£es, Portugal, **http://www3.dsi.uminho.pt/pcortez**, URL: **https://archive.ics.uci.edu/ml/datasets/Student+Performance**
- P. Cortez and A. Silva. Using Data Mining to Predict Secondary School Student Performance. In A. Brito and J. Teixeira Eds., Proceedings of 5th FUture BUsiness TEChnology Conference

(FUBUTEC 2008) pp. 5-12, Porto, Portugal, April, 2008, EUROSIS, ISBN 978-9077381-39-7. URL: http://www3.dsi.uminho.pt/pcortez/student.pdf

- Caret, Available Models. Last Accessed: 30-June-2017. URL: https://topepo.github.io/caret/available-models.html
- Open University Learning Analytics dataset Data Set. Released under CC-BY 4.0. Please use this paper for your citation: Kuzilek, J., Hlosta, M., Herrmannova, D., Zdrahal, Z. and Wolff, A. OU Analyse: Analysing At-Risk Students at The Open University. Learning Analytics Review, no. LAK15-1, March 2015, ISSN: 2057-7494. URL: https://archive.ics.uci.edu/ml/datasets/Open+University+Learning+Analytics+dataset
- Digit Recognizer, Learn computer vision fundamentals with the famous MNIST data. Last Accessed: 30 June 2017. URL: https://www.kaggle.com/c/digit-recognizer
- R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL: https://www.R-project.org/
- Jon Clayden (2017). mmand: Mathematical Morphology in Any Number of Dimensions. R package version 1.5.0. https://CRAN.R-project.org/package=mmand
- Joern Schulz Peter Toft pto@imm.dtu.dk Jesper James Jensen jjj@oedan.dk Peter Philipsen pap@imm.dtu.dk (2010). PET: Simulation and Reconstruction of PET Images. R package version 0.4.9. https://CRAN.R-project.org/package=PET
- Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., Hornik, K.(2016). cluster: Cluster Analysis Basics and Extensions. R package version 2.0.5.
- Venables, W. N. & Ripley, B. D. (2002) Modern Applied Statistics with S. Fourth Edition. Springer, New York. ISBN 0-387-95457-0
- Jan de Leeuw, Patrick Mair (2009). Multidimensional Scaling Using Majorization: SMACOF in R. Journal of Statistical Software, 31(3), 1-30. URL http://www.jstatsoft.org/v31/i03/.

- Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., Hornik, K.(2016). cluster: Cluster Analysis Basics and Extensions. R package version 2.0.5.
- Venables, W. N. & Ripley, B. D. (2002) Modern Applied Statistics with S. Fourth Edition. Springer, New York. ISBN 0-387-95457-0
- Jan de Leeuw, Patrick Mair (2009). Multidimensional Scaling Using Majorization: SMACOF in R. Journal of Statistical Software, 31(3), 1-30. URL http://www.jstatsoft.org/v31/i03/

- Max Kuhn. Contributions from Jed Wing, Steve Weston, Andre Williams, Chris Keefer, Allan Engelhardt, Tony Cooper, Zachary Mayer, Brenton Kenkel, the R Core Team, Michael Benesty, Reynald Lescarbeau, Andrew Ziem, Luca Scrucca, Yuan Tang, Can Candan and Tyler Hunt. (2017). caret: Classification and Regression Training. R package version 6.0-76. https://CRAN.R-project.org/package=caret
- Ben Hamner (2017). Metrics: Evaluation Metrics for Machine Learning. R package version 0.1.2. https://CRAN.R-project.org/package=Metrics