Final project

Norbert Eke

David Rowswell

Aalisha Lakdawala

Intro: Data set

- Spam emails classification
- Objective: minimise the false positive classifications.
- Data set: 4601 emails with content frequencies
- Data source: UCI Machine Learning repository

Methodology & results

1	Hierarchical clustering (single linkage)	39.38 %
2	Hierarchical clustering (average linkage)	39.38 %
3	Hierarchical clustering (complete linkage)	39.38 %
4	knn Classification	18.6 %
5	K-means clustering	36.4 %
6	LDA cross validation	11.8 %
7	LDA	11.0 %
8	QDA	17.12671 %
9	QDA cross validation	16.90937 %
10	Logistic regression	[8.1; 10.5; 7.7] %
11	Artificial Neural Network	[7.7; 6.6; 7.0; 7.6; 6.4; 6.7] %
12	Principle Component Analysis	14.9098 %
13	Bagging	5.23 %
14	Random forrest	4.75 %
15	Classification Trees	9.90%
16	Multiple Linear Regression wo/ interactions	12.10 %
17	Multiple Linear Regression w/ interactions	22.00 %

Logistic regression

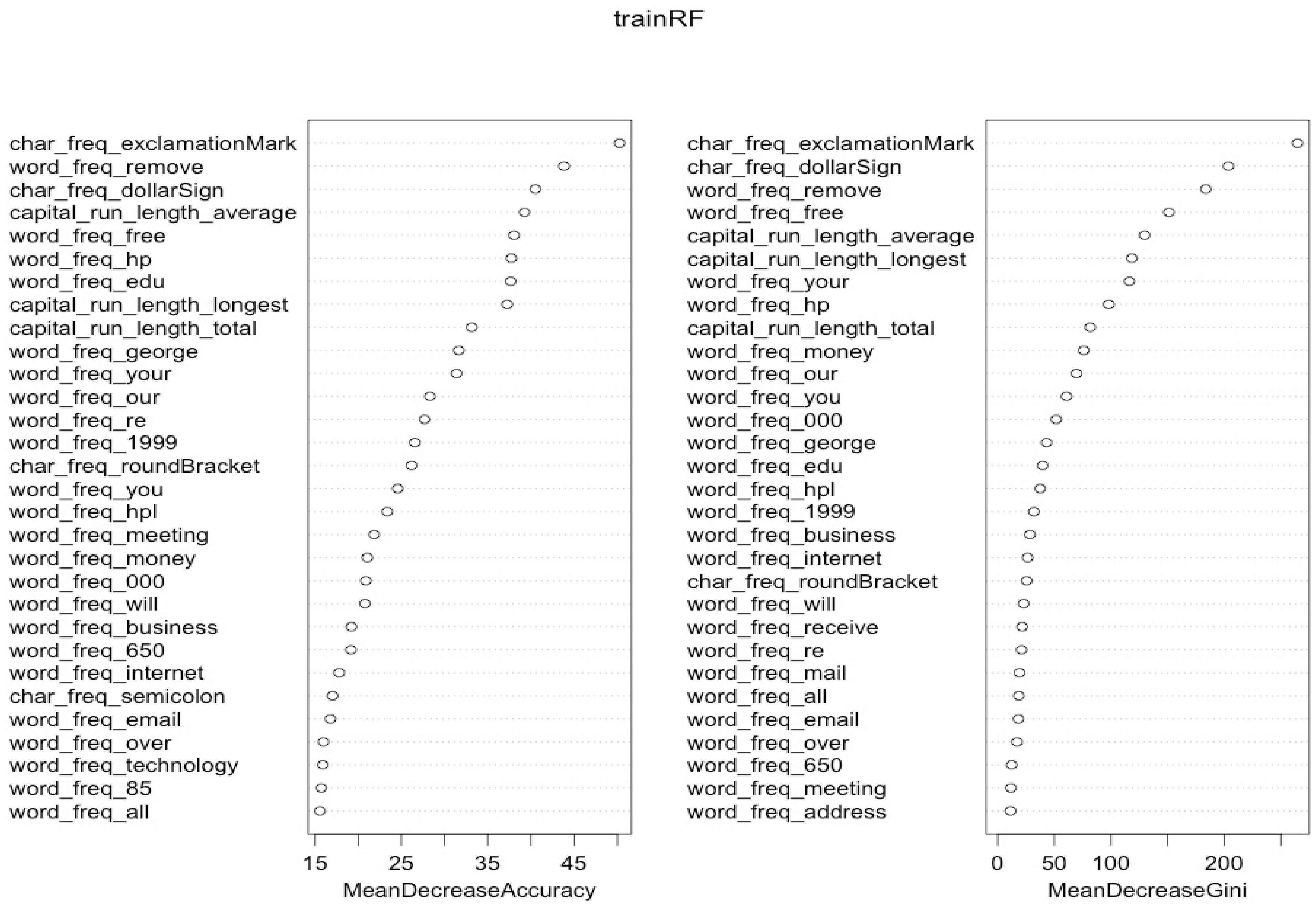
- Classification into groups instead of the continuous scale from 0-N
- Misclassification rate: 6.8%
- Results are too big to be shown here but are attached in the appendix

Bagging

- Mtry = 57
- Misclassification = 5.24%
- Since bagging is an average of models, it loses its interpretability but it does have good end result classification.

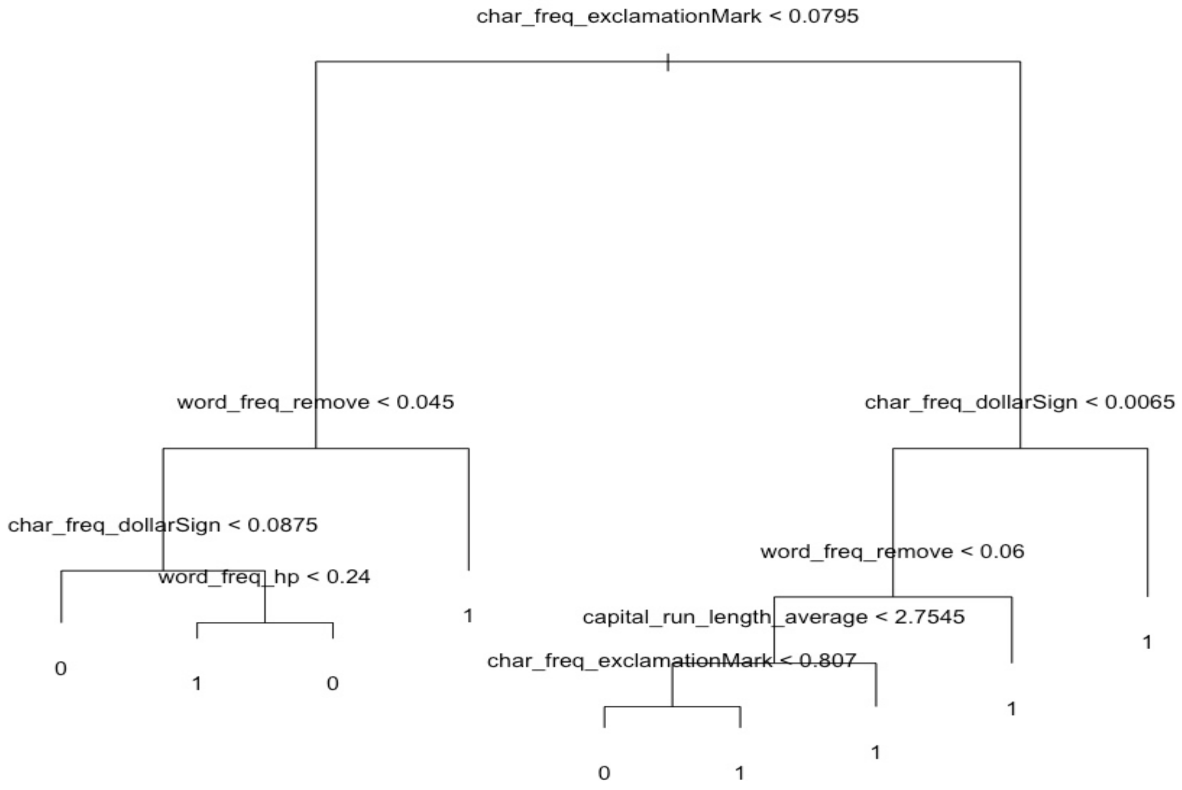
Random Forests

- Set mtry to default
- Decreases variance and increases stability
- Misclassification rate: 4.75%


trainRF
char_freq_exclamationMark
word_freq_remove
char_freq_dollarSign
capital_run_length_average
word_freq_free
word_freq_hp
word_freq_edu
capital_run_length_longest
capital_run_length_total
word_freq_george
word_freq_your
word_freq_our
word_freq_re
word_freq_1999
char_freq_roundBracket
word_freq_you
word_freq_hpl
word_freq_meeting
word_freq_money
word_freq_000
word_freq_will
word_freq_business
word_freq_650
word_freq_internet
char_freq_semicolon
word_freq_email
word_freq_over
word_freq_technology
word_freq_85
word_freq_all
15 25 35 45
MeanDecreaseAccuracy
char_freq_exclamationMark
char_freq_dollarSign
word_freq_remove
word_freq_free
capital_run_length_average
capital_run_length_longest
word_freq_your
word_freq_hp
capital_run_length_total
word_freq_money
word_freq_our
word_freq_you
word_freq_000
word_freq_george
word_freq_edu
word_freq_hpl
word_freq_1999
word_freq_business
word_freq_internet
char_freq_roundBracket
word_freq_will
word_freq_receive
word_freq_re
word_freq_mail
word_freq_all
word_freq_email
word_freq_over
word_freq_650
word_freq_meeting
word_freq_address
0 50 100 200
MeanDecreaseGini

Classification Tree splits

Misclassification % = 9.90 %



0 = non-spam e-mails and 1 = spam emails

Principle Component Analysis

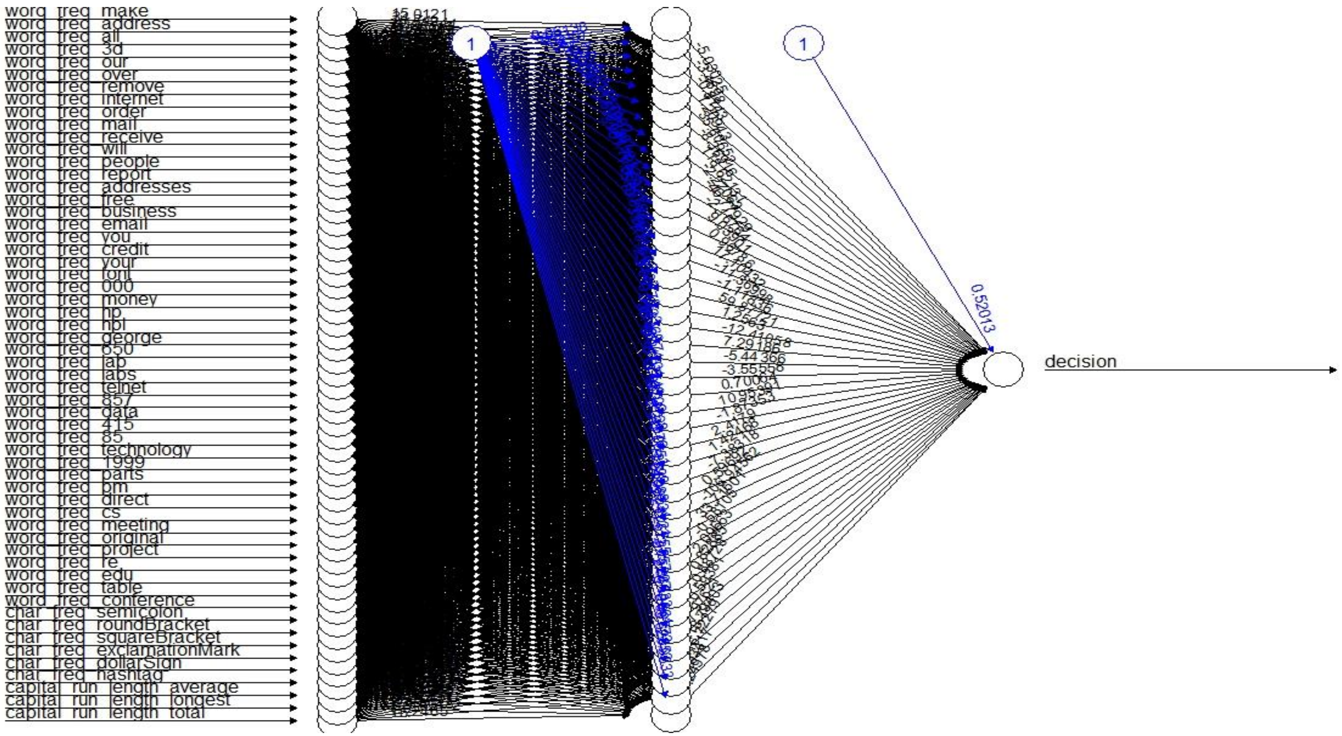
- First principle component loadings:

```
> pcaSpam <- prcomp(spam[,-58], scale.=TRUE)
> round(pcaSpam$rotation[,1], 2)
            word_freq_make          word_freq_address              word_freq_all
                     -0.04                      -0.01                      -0.05
              word_freq_3d              word_freq_our             word_freq_over
                     -0.01                      -0.04                      -0.05
          word_freq_remove         word_freq_internet            word_freq_order
                     -0.05                      -0.03                      -0.05
            word_freq_mail          word_freq_receive             word_freq_will
                     -0.02                      -0.05                      -0.02
          word_freq_people           word_freq_report        word_freq_addresses
                     -0.04                      -0.02                      -0.03
            word_freq_free         word_freq_business            word_freq_email
                     -0.04                      -0.05                      -0.02
             word_freq_you           word_freq_credit             word_freq_your
                     -0.08                      -0.03                      -0.08
            word_freq_font              word_freq_000            word_freq_money
                     -0.01                      -0.05                      -0.04
              word_freq_hp              word_freq_hpl           word_freq_george
                      0.21                       0.21                       0.04
             word_freq_650              word_freq_lab             word_freq_labs
                      0.28                       0.22                       0.30
          word_freq_telnet              word_freq_857             word_freq_data
                      0.31                       0.35                       0.01
             word_freq_415               word_freq_85       word_freq_technology
                      0.35                       0.27                       0.32
            word_freq_1999            word_freq_parts               word_freq_pm
                      0.05                       0.00                       0.04
          word_freq_direct               word_freq_cs          word_freq_meeting
                      0.32                       0.01                       0.02
        word_freq_original          word_freq_project               word_freq_re
                      0.07                       0.01                       0.01
             word_freq_edu            word_freq_table       word_freq_conference
                      0.00                       0.00                       0.00
       char_freq_semicolon     char_freq_roundBracket     char_freq_squareBracket
                      0.00                       0.14                       0.02
 char_freq_exclamationMark        char_freq_dollarSign           char_freq_hashtag
                     -0.04                      -0.05                       0.00
capital_run_length_average capital_run_length_longest    capital_run_length_total
                     -0.02                      -0.03                      -0.04
```

The word frequencies make sense, since the emails are Hewlett-Packard Internal-only Technical Reports.

Artificial Neural Networks

- nnSpam <- neuralnet(formula, data=train, linear.output = FALSE, hidden=40)
- nnresults <- compute(nnSpam, test[,-58])
- table(testDecision, nnresults$net.result>0.5)



Neutral Network Result

Misclassification %	**Hidden = ...**	**Misclassification %**	**Hidden = ...**
7.7	**5**	**6.7**	**c(12,11)**
6.6	**10**	**8.9**	**c(18,17)**
7.0	**20**	**8.7**	**c(20,10)**
7.6	**30**	**7.6**	**c(20,15)**
7.0	**35**	**7.5**	**c(20,18)**
6.9	**36**	**6.9**	**c(25,13)**
6.4	**37**	**7.6**	**c(25,15)**
8.1	**38**	**8.3**	**c(20,20)**
6.7	**39**	**8.5**	**c(25,20)**
6.4	**40**	**8.2**	**c(10,10,5)**
7.1	**45**	**8.1**	**c(10,10,10)**
6.7	**50**	**8.5**	**c(15,10,5)**
6.8	**c(7,7)**	**7.9**	**c(15,10,10)**
7.3	**c(10,10)**	**7.7**	**c(15,15,10)**
7.4	**c(15,10)**	**7.0**	**c(15,15,15)**
6.7	**c(15,15)**	**9.0**	**c(20,10,10)**
7.8	**c(13,13)**	**7.6**	**c(20,15,10)**
7.4	**c(11,11)**	**9.9**	**c(20,15,15)**

Conclusion

- Bagging, random Forests, Neural networks & Logistic regression worked the best
- Misclassification error can be brought down to 4.75%