Classification Problem - Search Dataset Student ID: 006322357

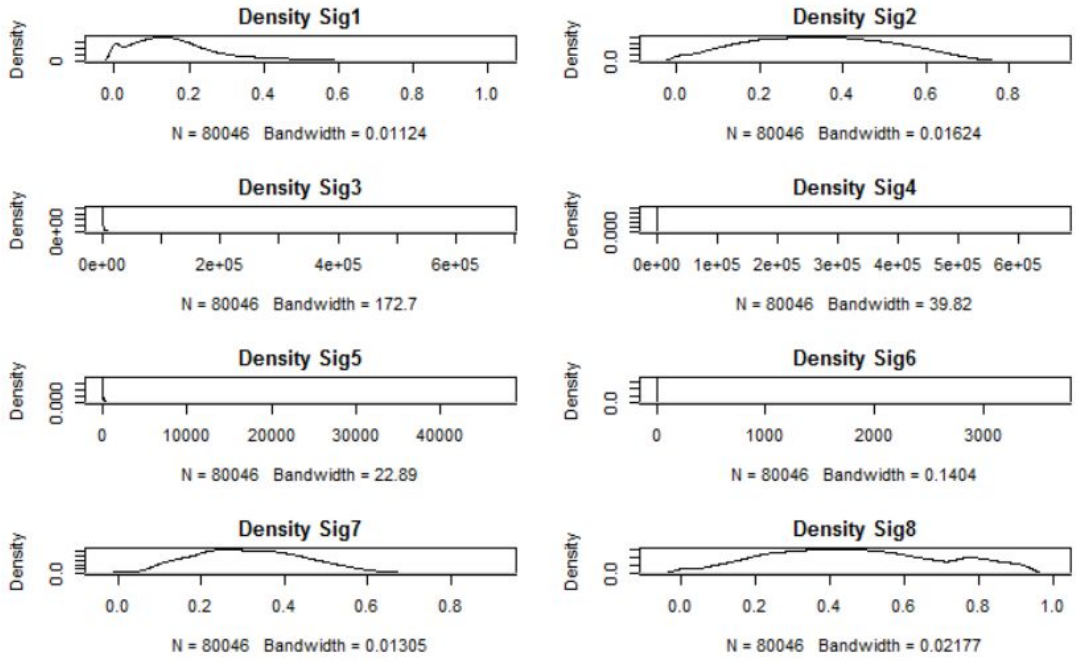
Resources Used: R3.4 and R Studio, Google Cloud Platform

Content

Data Exploration

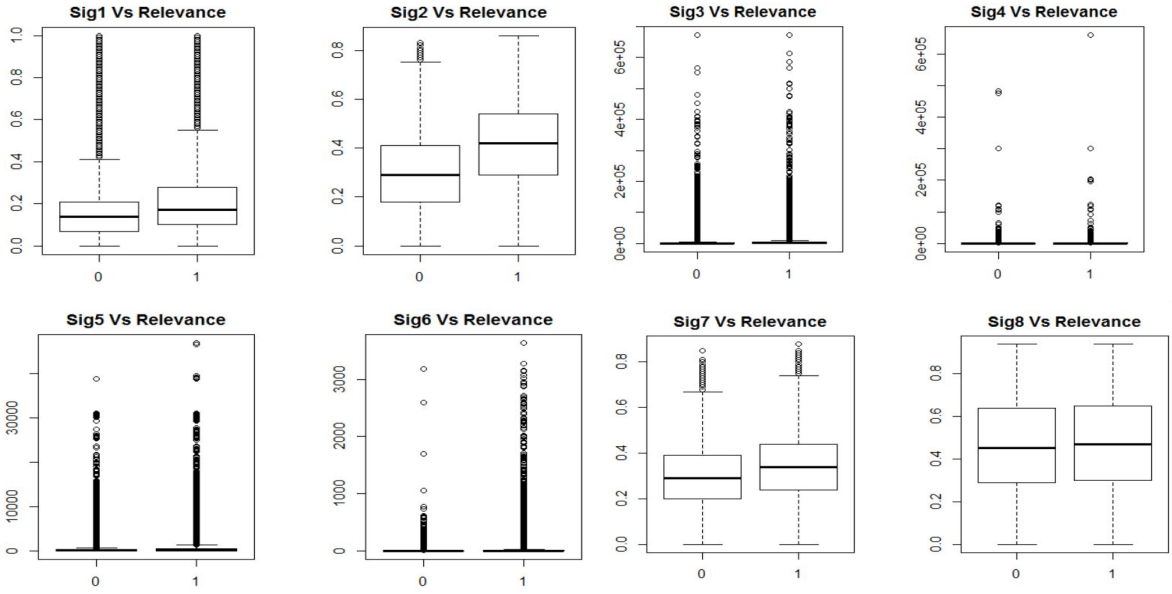
Step 1: Distribution of individual variables

- Density plots of all the variables were plotted. From these plots, it was clear that Sig1, 2, 7, 8 were almost normally distributed (skewed in some cases).
- Signals 3,4,5 and 6 were more like spikes. The “spikey” nature indicates that log() form of these signals could be explored for better results.
- Predicted variable relevance had a probability of 43.65% in the training data for level 1. This indicated that the data was unbalanced.



Step 2: Relationship with relevance

- All signals were plotted using boxplot. Signals 1,2 and 7 have clear differences in mean across relevance 0,1.
- Signal 6 shows difference of how the observations are spread.
- Other signals do not show any discernible differences across two levels of relevance. For these several nonlinear variations could be explored.



Step 3: Treatment of Query IDs

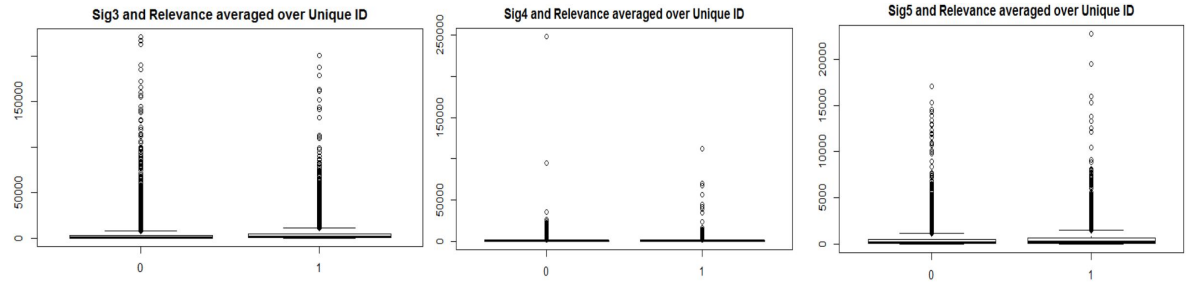
In the dataset, for a given query ID, a set of URL_IDs were checked for relevance. Certain query_IDs had more URL_IDs than others, some had higher values of signals than others. Hence it was important to factor the information that two rows in the dataset belonging to the same Query_ID were similar in some way. To ensure this, data was split by query_ID. All unique query_IDs were listed. 70% of them were randomly assigned to training set and the rest to test set.

Data separation into Test and Training Sets

```
unique_query = unique(Full_Data$query_id)
index = sample(length(unique_query), round(length(unique_query)/3), replace = FALSE)
test_QueryID = unique_query[index]
train_QueryID = unique_query[-index]

trainer = data.frame(Full_Data[Full_Data$query_id %in% train_QueryID,])
test = data.frame(Full_Data[Full_Data$query_id %in% test_QueryID,])
```

Step 4: Plots by query_ID

In step 2, it was observed that signals 3,4,5 had no discernible variation across factors 0,1 of relevance. These signals were revisited by averaging them by query_ID and plotting them again on box plot. This did not help in differentiating them by relevance.



Selection of Variables
Feature selection was initiated after data exploration and transformation. After trying/testing several combination of features, the following features were removed.

Feature Removed	Reason for removal / Alternative Course Taken
Query_ID	If two URL IDs belong to the the same query, relevance could be impacted. This has been taken care in data transformation. Hence Query_ID (being a random number) was removed.
URL_ID	Since this is just a random number, it was removed.
Sig3	On exploring these signals (Refer to data exploration section) these looked like spikes unlike other signals that were normally distributed. They were combined as one entity *Sig345* [log(1+Sig3)+ log(1+ Sig4) + log(1+Sig5)] since the combined variable was found to give best results.
Sig4	
Sig5	

Preprocessing

Step 1: All features were normalized.

Step 2: Factor variables, relevance, is_homepage converted to factors.

Step 3: Processing signals 3,4,5
From steps 2 and 4 of data exploration, it was clear that signals 3,4,5 probably had a non-linear relationship with relevance (they could also be junk variables). These were transformed and a simple logistic regression was run to understand the best fit.

a. Transforming to square/cube terms: Logistic regression did not give good p-values.
b. Transforming to log: log(1+x) was applied to these signals and it was observed that the p-values were good overall including those for features introduced later.

Step 4: Processing Sig1, 2, 6, 7, 8
These were first processed as log() values however, not all of them were significant across models. A different treatment for these has been explained in the next section.

Transforming / Feature Engineering

Step 1 : Introducing Features
It was necessary to factor that if two rows of data belonged to the same Query_ID, they could possibly be similar. To factor this, the following features were introduced.

Feature Introduced	Description	Retained in the model?
NumHP	This gives the number of homepage results given a query_ID.	Good significance across several models, Retained

Feature Introduced	Description	Retained in the model?

Popularity	This is equal to number of observations (rows) for a query_ID divided by the maximum number of observations a query_ID had in the dataset.This is based on the logic that popular queries could have more number of URL_IDs returned than queries that are otherwise.
	
	for(i in 1:length(unique_query))
	{
	D$Popularity[D$query_id == unique_query[i]] = nrow(D[D$query_id == unique_query[i],])
	}
		Not shown great significance, Removed

Sig1_Query	Signal strength adjusted to each query was determined.
	
	Sig1_Query was calculated as follows:
	Sig1 - Mean(Sig1 for the Query_ID)
	
	for(i in 1:length(unique_query))
	{
	D$Sig1_Query[D$query_id == unique_query[i]] = mean(D$sig1[D$query_id == unique_query[i]])
	}
	D$Sig1_Query = D$sig1 - D$Sig1_Query
		Good significance, retained
		
		Transformation was performed only on Sig1,2,7 and 8 since they had a normal distribution (both raw data and signals by unique ID)

Sig2_Query	Same as above but for Sig2	
Sig7_Query	Same as above but for Sig7	
Sig8_Query	Same as above but for Sig8	

Sig345	Based on the raw data, Sig3, 4, and 5 were guessed as backlinks, number of relevant words from query etc. Raw signal was spikey in nature. Hence log() was applied. Sig345 = log(1+Sig3) +log(1+Sig4) + log(1+Sig5) averaged over query_ID
	
	D$SumSig345 = log(1+D$sig3) + log(1+D$sig4) + log(1+D$sig5)
	
	#normalizing it
	D$SumSig345 = (D$SumSig345 - min(D$SumSig345))/(max(D$SumSig345) - min(D$SumSig345))
	
	#finding mean given a query_ID
	for(i in 1:length(unique_query))
	{
	D$SumSig345[D$query_id == unique_query[i]] = mean(D$SumSig345[D$query_id == unique_query[i]])
	}
		Reasonable significance in certain models, Retained

Final List of Signals after selection, processing and transformation

Original Signal (Total 12 predictors)	New signal (Total 13 predictors)	Comments
URL_ID, Query_ID	NA	Removed
Query Length	Query Length	Retained
Is_Homepage	Is_Homepage, NumHP	• Original Retained + Number of home page results for the query
Sig1, Sig2, Sig7, Sig8	Sig1, Sig1_Query, Sig2, Sig2_Query, Sig7, Sig7_Query, Sig8, Sig8_Query	• 4 signals replaced with 8 signals • Original retained + Additional features
Sig6	Sig6	Original retained
Sig3, Sig4, Sig5	log(1+Sig3) + log(1+Sig4) log(1+Sig5) averaged over query_ID	Replaced by one variable

Training Dataset (NumHp and SumSig345 would be same for a query_ID)

query_length	is_homepage	sig1	sig2	sig6	sig7	sig8	NumHp	Sig7_Query	Sig8_Query	Sig1_Query	Sig2_Query	SumSig345	relevance
2	1	-0.63453878	-1.13454024	-0.00976859	1.02232125	-0.526124833	9	0.248879073	-1.925971e-02	-1.022651e+00	-1.513016009	0.5683654	0
2	1	0.10989107	0.02436564	0.16046506	0.80555561	-0.872799686	9	0.032113429	-3.659346e-01	-2.782213e-01	-0.354110130	0.5683654	1
2	1	1.19269811	0.83559976	0.01292923	0.15525868	-1.176140183	9	-0.618183503	-6.692751e-01	8.045858e-01	0.457123986	0.5683654	1
2	1	0.17756651	0.60381858	0.00158032	0.87781082	-0.612793546	9	0.104368643	-1.059284e-01	-2.105458e-01	0.225342810	0.5683654	1
2	1	0.44826827	0.42998270	0.14911615	0.58878997	0.427231015	9	-0.184652215	9.340961e-01	6.015595e-02	0.051506928	0.5683654	1
2	1	0.31291739	0.54587329	0.28530307	1.09457647	-1.046137113	9	0.321134287	-5.392720e-01	-7.519493e-02	0.167397516	0.5683654	0
2	1	0.38059283	0.77765446	0.13776724	0.95006604	-0.656127903	9	0.176623858	-1.492628e-01	-7.519493e-03	0.399178692	0.5683654	0
2	1	0.58361915	0.83559976	0.41014108	0.66104518	0.037221805	9	-0.112397001	5.440869e-01	1.955068e-01	0.457123986	0.5683654	1
2	1	0.92199635	0.48792799	0.60307255	0.80555561	-0.136115622	9	0.032113429	3.707495e-01	5.338840e-01	0.109452222	0.5683654	0
4	1	0.04221563	-0.78686847	-0.15730442	1.96163904	0.557234085	3	0.751454232	-2.600061e-02	-1.191088e+00	-1.054604350	0.6034023	0
4	0	1.05734723	-0.78686847	-0.15730442	0.66104518	-0.136115622	3	-0.549139631	-7.193503e-01	-1.759561e-01	-1.054604350	0.6034023	0
4	1	1.05734723	0.42998270	-0.15730442	1.16683168	0.990577652	3	-0.043353129	4.073430e-01	-1.759561e-01	0.162246823	0.6034023	1
4	0	2.88458411	1.41505270	-0.15730442	1.02232125	0.860574582	3	-0.187863558	2.773399e-01	1.651281e+00	1.147316821	0.6034023	0

Data mining *(Final output obtained by weighted voting method across models)*

- K-fold cross validation used with k = 3 throughout the project. K was chosen to be 3 due to computing resource constraints.
- Methods indicated in the table were implemented.
- Decision trees were implemented in two ways.
  - Method1 - Tree was made to branch only when error reduced by a threshold value (this is fast however, suffers from the drawback that certain nodes may not be explored further since they would not have crossed the threshold rate)
  - Method2 - Tree is grown completely and then pruned

Method Used	Optimized shrinkage / control parameter	Error on cross-validation	Error on Test set	Type 1 error (1-recall)* on test set	Type 2 error* on test set	Confusion matrix*
Logistic regression	NA	34.44	34.27	51.23	21.05	Actual 1 Actual 0 Predicted 1 5752 3184 Predicted 0 6043 11939
Naive Bayes Method	NA	36.82	36.68	81.55	4.59	Actual 1 Actual 0 Predicted 1 2176 694 Predicted 0 9619 14429
Decision Tree (Method1)	Lambda = 0.000464	34.23	33.74	50.00	21.02	Actual 1 Actual 0 Predicted 1 5897 3179 Predicted 0 5898 11944
Decision Tree (Method 2)	Lambda = 0.00033	33.61	33.93	53.28	18.84	Actual 1 Actual 0 Predicted 1 5510 2850 Predicted 0 6285 12273
Random Forest	m = 2 (only 5 values tried 2,3,4,5,6)	33.20	32.90	50.07	19.01	Actual 1 Actual 0 Predicted 1 5814 2876 Predicted 0 5981 12247
Boosting	Shrinkage = 0.0501	33.47	33.02	49.55	20.12	Actual 1 Actual 0 Predicted 1 5950 3044 Predicted 0 5845 12079
SVM (Radial kernel)	Cost = 1 Gamma = 0.06390	33.92	33.71	57.91	14.83	Actual 1 Actual 0 Predicted 1 4964 2244 Predicted 0 6831 12879
Weighted voting	Logistic regression applied for obtaining a stacked model. Accuracy on test set = 32.70% Random forest, boosting, naive bayes were given higher weights.					

*The seed values were changed while calculating test error. The confusion matrix, type 1 and type 2 errors correspond to the same seed while the test error corresponds to a different seed. (while the actual value for type 1, 2 errors is slightly different, the idea that the errors are unbalanced remains the same).

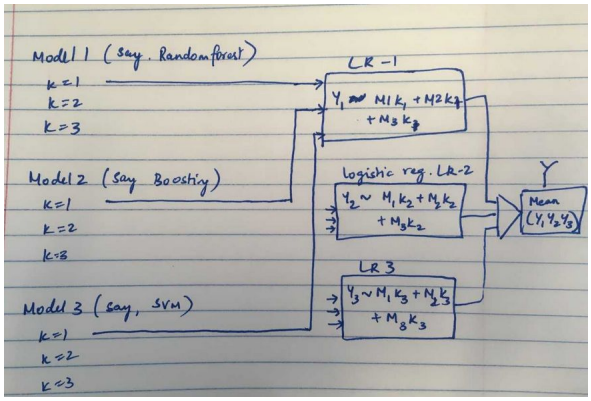
Images / Graphs on model tuning etc. have been covered in Appendix.

Interpretation / Evaluation (Stacking)

A stacked model is utilized to further optimize results. (Based on several confusion matrices, imbalanced error could be observed. For future study, ROC could be explored for this dataset and tuning could be performed differently)

Stacked Model

For every model, k-folds of cross validation is performed. 1 fold from every model is taken and a logistic regression is performed to get the response variable relevance. This is repeated for all the k-folds.




```
Call:
glm(formula = (relevance == 1) ~ ., family = binomial, data = df1)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1839  -0.9539  -0.6296   1.0691   2.1912

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.75580    0.07280 -37.855  < 2e-16 ***
lr           0.24587    0.25857   0.951  0.34166
nb          -0.24711    0.08793  -2.810  0.00495 **
dt1         -0.29720    0.21489  -1.383  0.16665
dt2         -0.19514    0.25331  -0.770  0.44107
rf           4.18997    0.26646  15.724  < 2e-16 ***
bt           1.86968    0.28145   6.643 3.07e-11 ***
svm1        -0.05770    0.05712  -1.010  0.31239
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Results from Stacking - Random forest, boosting, naive bayes have shown good significance.

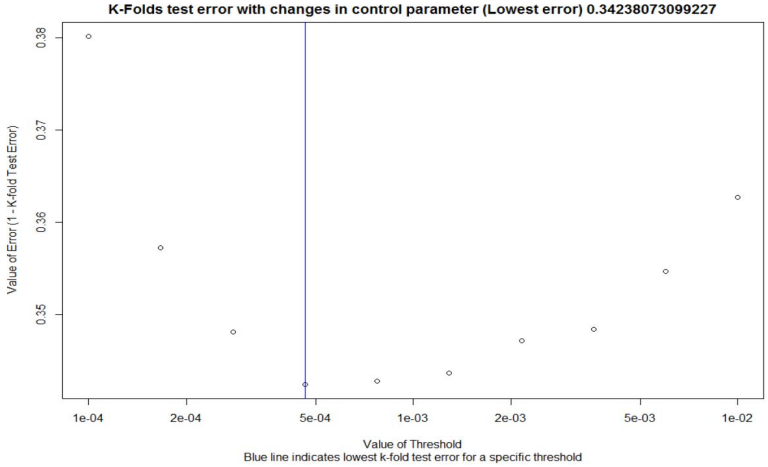
Appendix - Model wise graphs / images

Logistic Regression

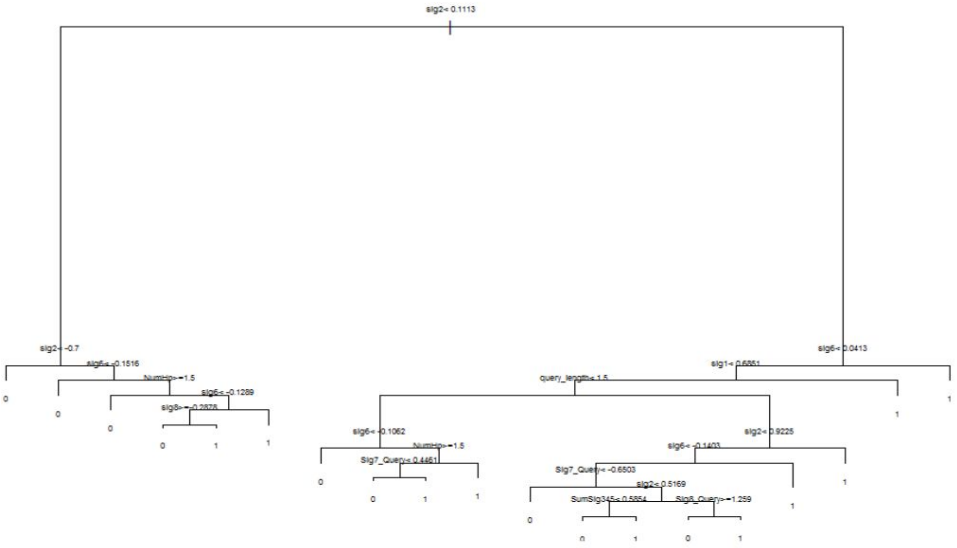
```
Call:
glm(formula = (relevance == 1) ~ ., family = binomial, data = trainer)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-7.0808  -0.9894  -0.6963   1.1348   2.1907

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.800565   0.064983 -12.320  < 2e-16 ***
query_length  0.074018   0.006704  11.041  < 2e-16 ***
is_homepage1  0.067128   0.028806   2.330   0.0198 *
sig1          0.002928   0.017446   0.168   0.8667
sig2          0.567366   0.021261  26.686  < 2e-16 ***
sig6          0.694061   0.037833  18.345  < 2e-16 ***
sig7          0.049117   0.027162   1.808   0.0706 .
sig8         -0.274860   0.022492 -12.220  < 2e-16 ***
NumHp        -0.137699   0.007277 -18.923  < 2e-16 ***
Sig7_Query    0.166549   0.030785   5.410 6.30e-08 ***
Sig8_Query    0.187867   0.025546   7.354 1.92e-13 ***
Sig1_Query    0.199415   0.021590   9.236  < 2e-16 ***
Sig2_Query   -0.022813   0.024852  -0.918   0.3587
SumSig345     1.502309   0.152926   9.824  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Decision Trees - Method 1

Method 1 : Control parameter was varied for a wide range initially. A second round of tuning for finer data gave control value of 0.000464. While tuning, k-fold cross-validation was conducted to record the errors.



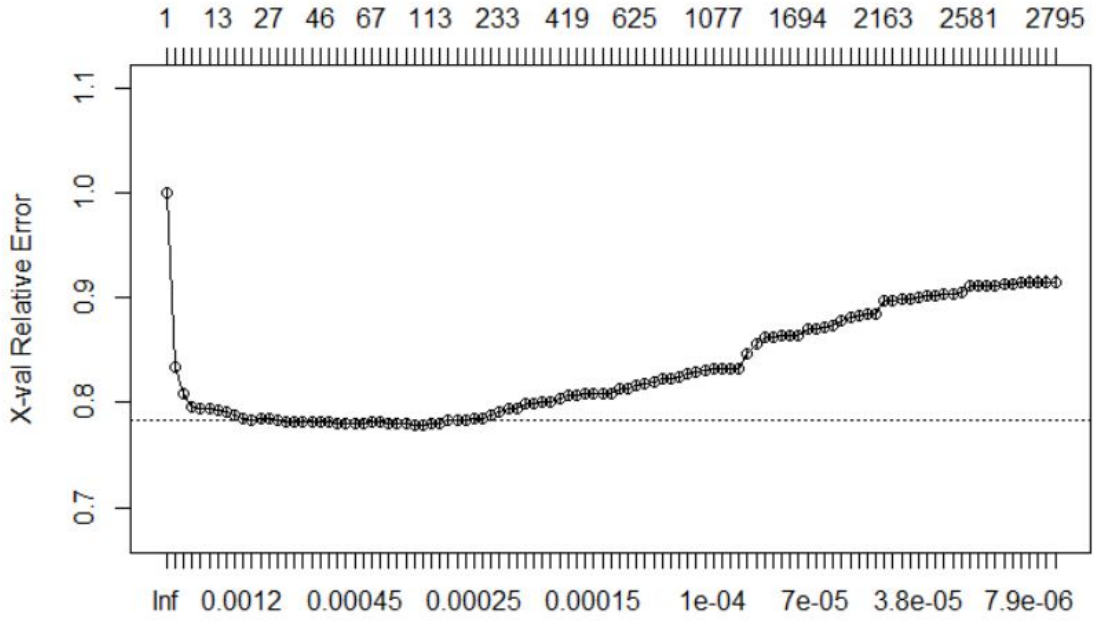
Decision tree based on optimal control parameter



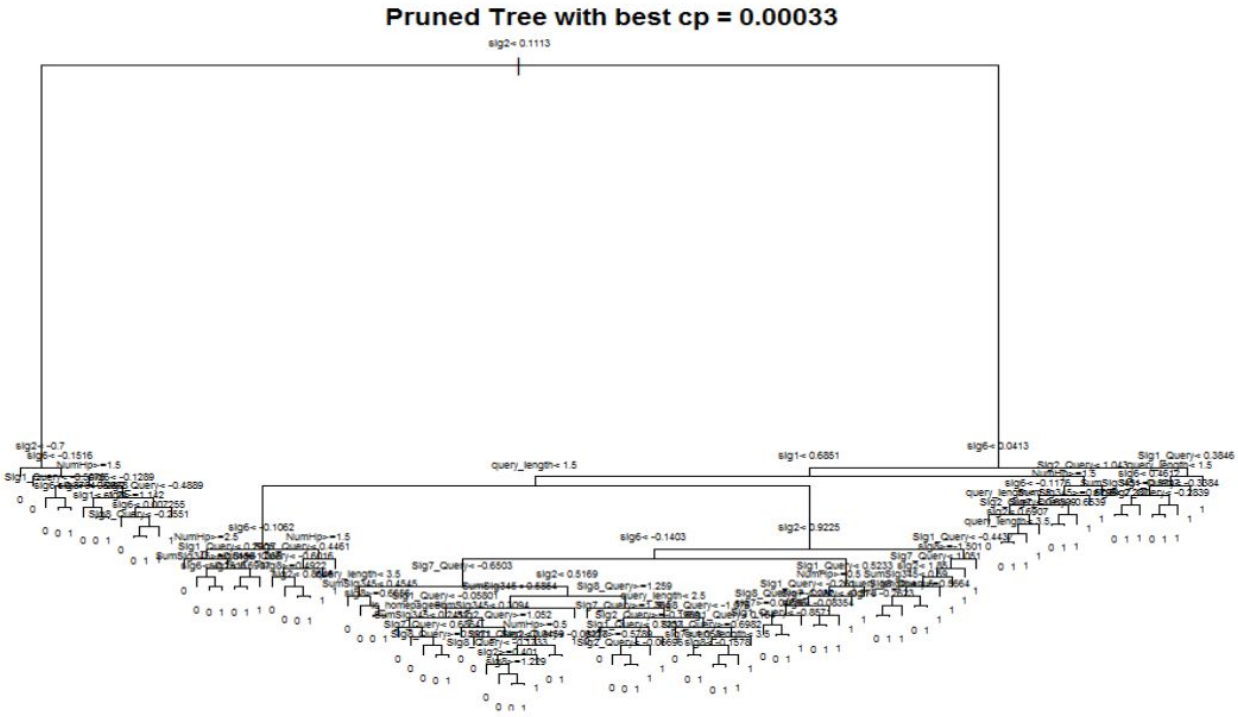
Decision Trees - Method 2

Method 2 - Growing tree completely and pruning

Best cp value from the graph was obtained as 0.00033 (y-axis is relative error and not actual error rate)

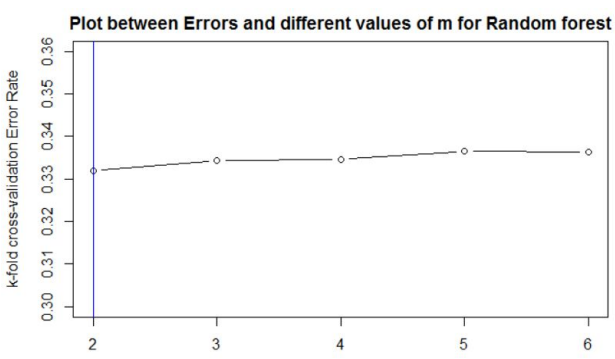


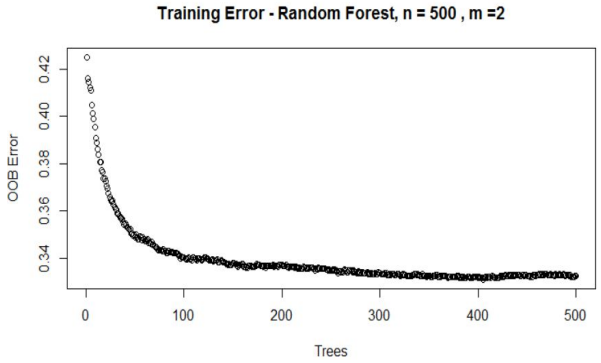
Selected tree based on best cp-value



Random Forest

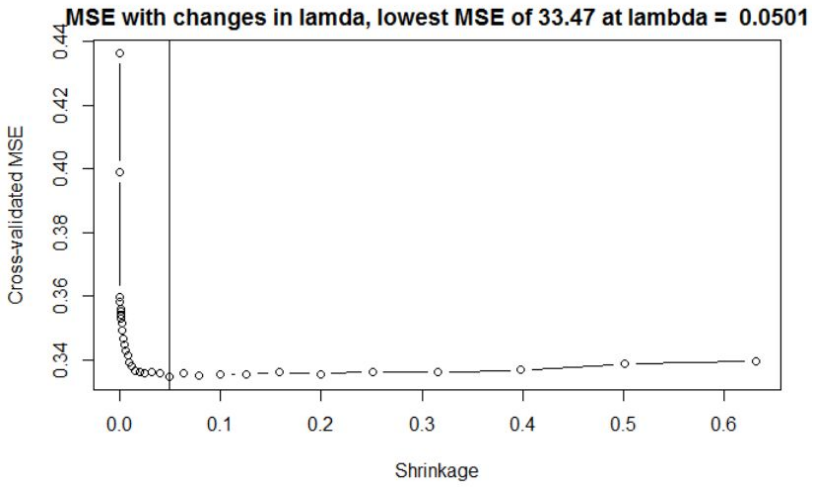
M = 2 selected for lowest cross-validated error of 33.20%





Boosting

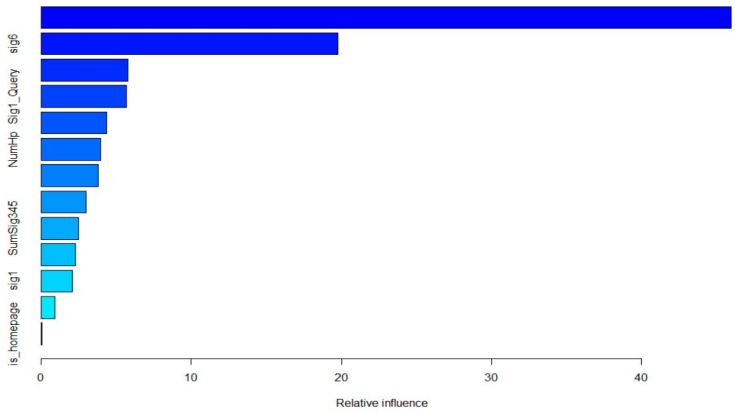
1. Boosting Error by changing shrinkage parameter



2. Boosting Parameters by importance

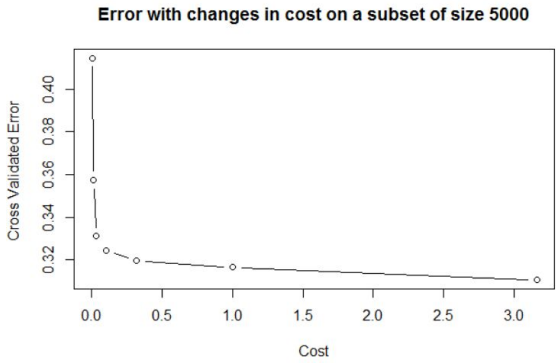
	var <fctr>	rel.inf <dbl>
sig2	sig2	46.00875720
sig6	sig6	19.78763685
query_length	query_length	5.74450721
Sig1_Query	Sig1_Query	5.64801174
Sig2_Query	Sig2_Query	4.35644247
NumHp	NumHp	3.92650482
Sig7_Query	Sig7_Query	3.78899537
Sig8_Query	Sig8_Query	2.99980171
SumSig345	SumSig345	2.48466755
sig8	sig8	2.25826687
sig1	sig1	2.09031572
sig7	sig7	0.88107285
is_homepage	is_homepage	0.02501964

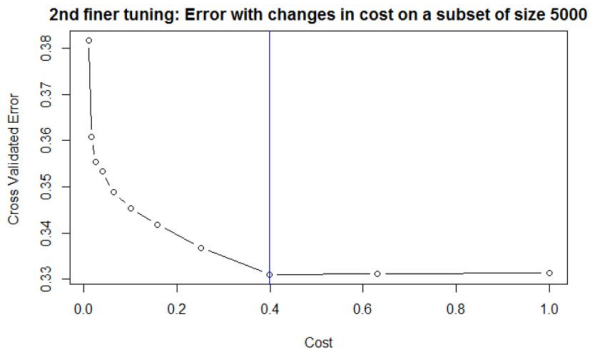
13 rows



SVM

SVM was first run with large cost range on a small dataset as shown in picture. A second tuning was performed for a cost range of 0.01 to 1, with improved least count.





However, a final tuning with gamma and cost simultaneously resulted in least error (c = 1 and gamma = 0.06390).

RStudio: Notebook Output

cost <dbl>	gamma <dbl>	error <dbl>	dispersion <dbl>
0.01000000	0.01000000	0.4040	0.02114500
0.01584893	0.01000000	0.3566	0.02141754
0.02511886	0.01000000	0.3406	0.01840411
0.03981072	0.01000000	0.3384	0.01531303
0.06309573	0.01000000	0.3356	0.01693911
0.10000000	0.01000000	0.3318	0.01797406
0.15848932	0.01000000	0.3302	0.01726782
0.25118864	0.01000000	0.3298	0.01896078
0.39810717	0.01000000	0.3290	0.02020451
0.63095734	0.01000000	0.3256	0.01863807
1.00000000	0.01000000	0.3234	0.01936893
0.01000000	0.01584893	0.3688	0.02566797
0.01584893	0.01584893	0.3436	0.01725109
0.02511886	0.01584893	0.3392	0.01738965
0.03981072	0.01584893	0.3350	0.01655295
0.06309573	0.01584893	0.3336	0.01648703
0.10000000	0.01584893	0.3308	0.01876640
0.15848932	0.01584893	0.3302	0.01917058
0.25118864	0.01584893	0.3280	0.01873796
0.39810717	0.01584893	0.3240	0.01887974
0.63095734	0.01584893	0.3236	0.02021660
1.00000000	0.01584893	0.3226	0.02093482
0.01000000	0.02511886	0.3516	0.02434566
0.01584893	0.02511886	0.3408	0.02070319
0.02511886	0.02511886	0.3360	0.01675974
0.03981072	0.02511886	0.3336	0.01613278

1-26 of 231 rows Previous 1 2 3 4 5 6 ... 9 Next