Contents

Table of Figures

1. Introduction – Background

Data breaches along with dump of sensitive information including passwords, social security and other business critical information is becoming norm in today`s digital world.
This stolen information is often sold in underground community forums. This information can be further abused by the buyers in several forms such as buying goods with stolen credit card information, holding social media accounts data for ransom etc. A database of 1.4 billion clear text credentials which was compilation of several data breaches including sites such as yahoo.com, linkedin etc was discovered and caught media attention due to the sheer size and organized compilation in cleartext. The data in the dump is organized alphabetically in several files making it easier to search for passwords and also associated email addresses. The breaches are more than year old and several sites have taken preventive measures post headlines to let users know and forcing mandatory passwords reset in some cases.

This database especially passwords and associated Email domains offers an interesting opportunity to understand the password behavior and characteristics across various geographical regions. By understanding this dataset, various common trends irrespective of geographical regions or private companies can be highlighted along with the need of best practices in selecting strong password and measuring the strength of it getting compromised either by automated ways or simply guessing the passwords if it is very common. Due to the vast size of the dataset and diverse records aggregate statistics were derived so that characteristics can be associated to Email Domain which can then be categorized into geographical or business sectors. Following to the data analysis, predictive model was designed to predict geographical region based on statistical characteristics of passwords. In the example implementation, UK and Russia passwords for the Gmail domain have been trained, analyzed and predictions were made for it, but this model can be further extended to predict various geographies of the same Email domain or business sectors in the similar fashion by training the respective dataset. Business use case of this model outcome can also be to understand how different regions or business sectors take password security, compare password strength and based on the outcome increase complexity across different regions/ business sectors. This will reduce predictability, common password trends and increase time to crack it by automated password brute force tools or dictionary wordlists of common trends.

2. Dataset

The links to original archive of 41 GB having 1981 files were found on various social media blogs with common Pastebin link containing both compressed
and uncompressed version of the archive. The data is structured in an alphabetic directory tree fragmented in 1,981 pieces to allow fast searches.

The dump includes search tools and insert scripts explained in a README file.

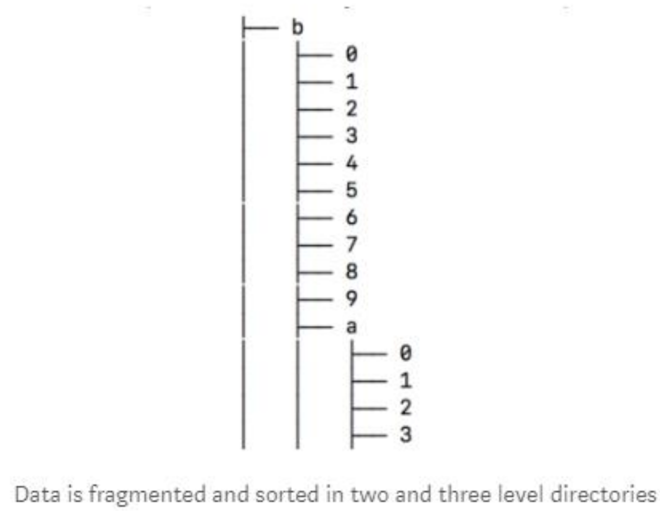

Figure 1: Directory Structure of the original Dataset.

3. Data preparation and Cleaning

All the files were without file extensions hence added txt extension to all the 1986 files.
Below Powershell one-liner used to append. R code could also be used to achieve the same.

```
#To add extension to files
Get-ChildItem -Path "C:\Users\ashwin\Documents\Training\Springboard\capstone" -
File | Rename-Item -NewName { $PSItem.Name + ".txt" }
```

Figure 2 Powershell one-liner to rename and add extensions to all files

In order to get familiar with the dataset of all the files. R code was written to read all the files and generate metadata about all the files.
The metadata was file name along with the file path and no of records within files including the no of duplicates in each files.
Since each file is alphabetically sorted, all the dataset was processed with lapply and generating summary stats each line representing single files within files.

Out of the total 1986 files and the total records- 1400553869, no of duplicates - 202281 were found.

Base dataset was read via read_delim to parse below columns removing de-dupolicates as a first step in the data preparation.
FileSchema parsed from all the files after deduplication:
The dataset primarily contains email id and password an does not have any other details indicating if it is about actual email domain or used at any other sites.
After taking a closer look at the datasets several data quality issues were identified including duplicate records and separated from the original dataset.

Below criteria was used to find invalid patterns of the dataset and kept out of the scope for the analysis.

- Duplicates records
- Files with invalid data or null characters.
- Email Id with no password present.
- Passwords with less than 6 char length.

Apart from this, there were other characteristics were observed but kept in the analysis as those are valid password on the sites.

- Similar Email ID and password.

4. Data Wrangling

Summary files aggregated by EmailDomain as well as password length were generated to use it for further data wrangling as well as data visualization purposes.

Consolidated Summary:

```
> names(charfreqtable)
[1] "EMailDomain"       "TotalRecords"      "CharacterCount"    "LowercaseCount"
"UppercaseCount"    "AlphaNumericCount" "NumericCount"      "CyrillicCount"
"PunctCount"
```

Figure 3: Schema for Summary Dataset Aggregated by EmailDomain

```
> colnames(pwdltable)
[1] "PasswordLength"      "DistinctPasswords"   "DistinctEmailDomain"
"DistinctEmails"      "TotalCount"
```

Figure 4: Schema for Summary Dataset Aggregated by PasswordLength

For each password associated with EmailDomain , additional measurement variables were generated describing password characteristics. The passwords are aggregated and associated with EmailDomain and not individual EmailID. Dataset was also further enriched by splitting the EMaildomain into domain , subdomain and suffix by using Urltools library.

```
> colnames(final.data)
 [1] "EMailDomain"       "domain"            "suffix"
"AlphaNumericCount" "CharacterCount"    "CyrillicCount"     "LowercaseCount"
"NumericCount"      "PunctCount"
[10] "subdomain"         "TotalRecords"      "UppercaseCount"
```

Figure 5: Schema for Dataset with measurement variables for password by EmailDomain

This dataset was further extended to populate the sectors of each email domain associated with S& P 500 companies joining against the website column of the S & P 500 dataset acquired from data.world. was used to segregate domain, subdomain from Emaildomain values.
Out of the 500 sites, 201 companies data were matched which are from 21 different sectors.

```
> colnames(sector.average.data)
 [1] "EMailDomain"       "domain"            "suffix"            "TotalRecords"
"AlphaNumericCount" "CharacterCount"    "CyrillicCount"     "LowercaseCount"
"NumericCount"
[10] "PunctCount"        "subdomain"         "UppercaseCount"    "title"
"website"           "sector"
```

Figure 6: Schema for Dataset with joining against S & P 500 Domains (FileSize: 497 MB)

5. Data Analysis

Before doing data analysis on the entire dataset, few statistics were generated about the quality of the entire dataset to filter invalid data.

- **No of Total Records:** 1400553869
- **No of duplicate Records:** 202281
- **No of Records with no passwords/no EmailIds :** 2546790
- **No of records with password length less than 6 :** 40140152
- **No of Files with bad data (NULL chars) :** 105
- **No of domains with above 1000 passwords:** 8904
- **No of domains involved – from S& P 500 Companies :** 201
- **No of S &P 500 Sectors involved:** 21

6. Data visualization

- **Top 15 Email domains were found and plotted the percentage distribution of the entire dataset.**

```
ggplot(charfreqtabletop15, aes(x=reorder(EMailDomain, Percentage), y=Percentage))
+ geom_bar(stat = "identity")
+ coord_flip()
+ labs(y = "Percentage of Total Dataset", x = "Top 15 EmailDomains")
```

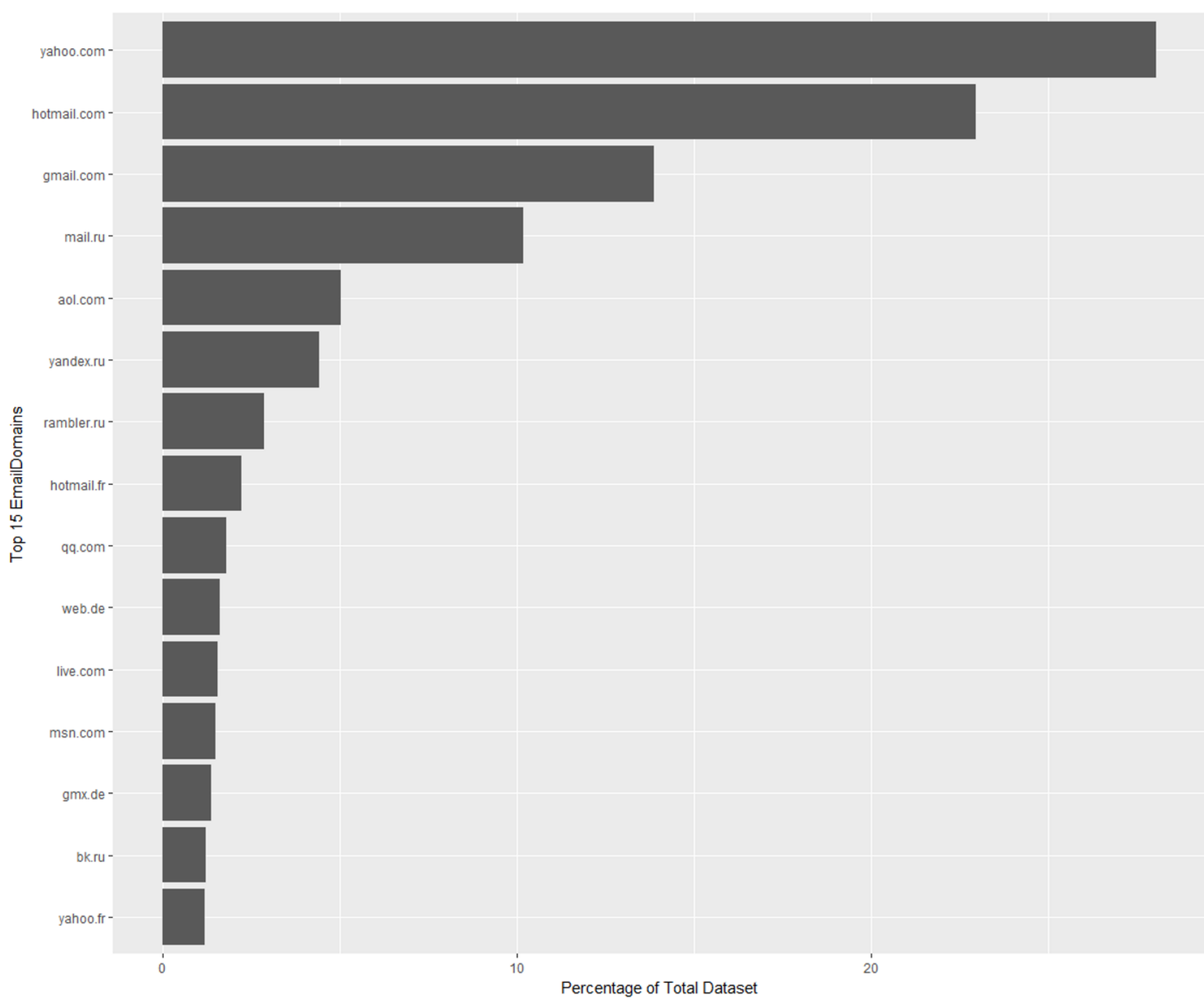

Figure 7: Percentage Distribution of Top 15 Email Domain involved

- **Histograms on Password Length are plotted with Password length in numeric on x axis and Total Count of records in Millions on y axis.**

```
options(scipen=10000)
ggplot((pwdltable %>% filter(PasswordLength < 40)), aes(x=factor(PasswordLength),
y=TotalCount/1e06))
+ geom_bar(stat = "identity")
+  labs(x = "PasswordLength", y = "Total Count (in millions)n")
```

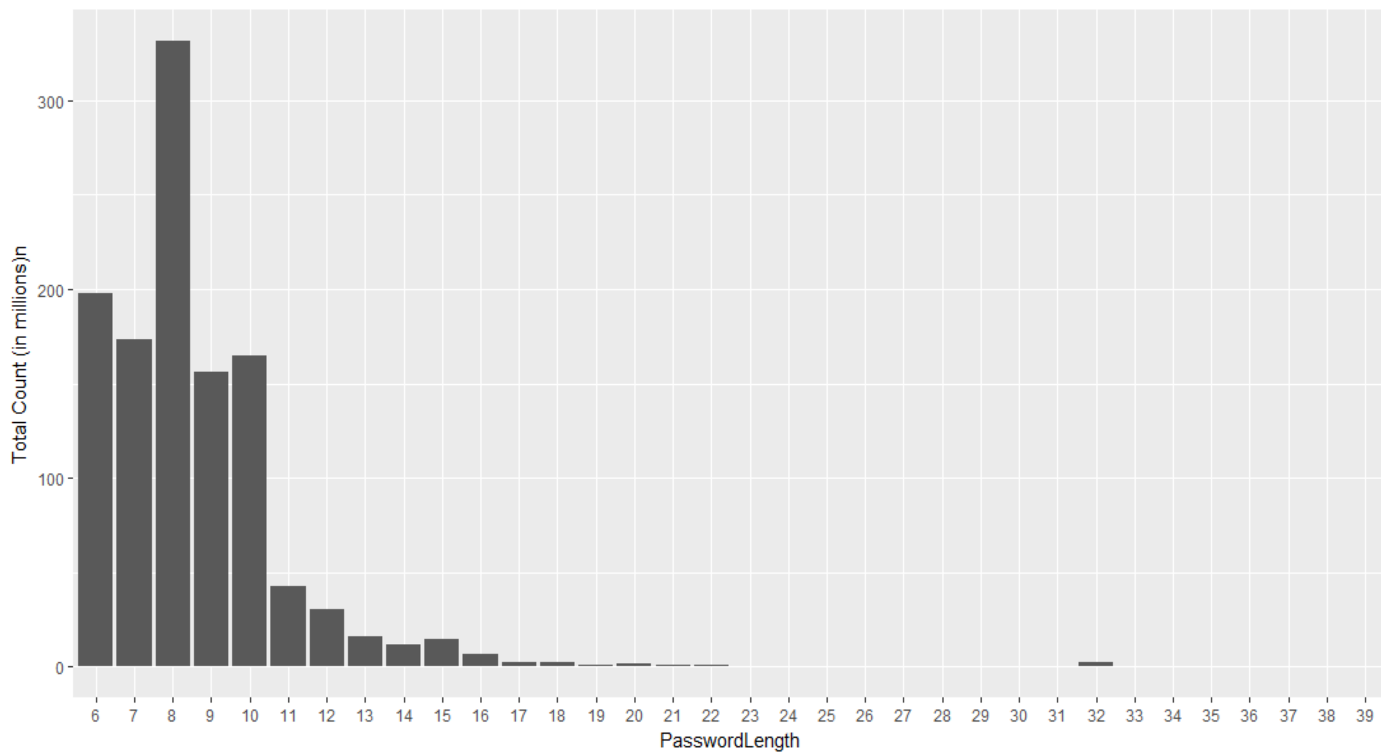

Figure 8: Histogram on Password Length with TotalCount in Millions

- **Bar chart showing the no of domains per sector present in the dataset is plotted with sector on y-axis and no of companies on x axis.**

```
ggplot(sectorbysite, aes(x=reorder(sector,NoofCompanies), y=NoofCompanies))
+ geom_bar(stat = "identity") + coord_flip()
```

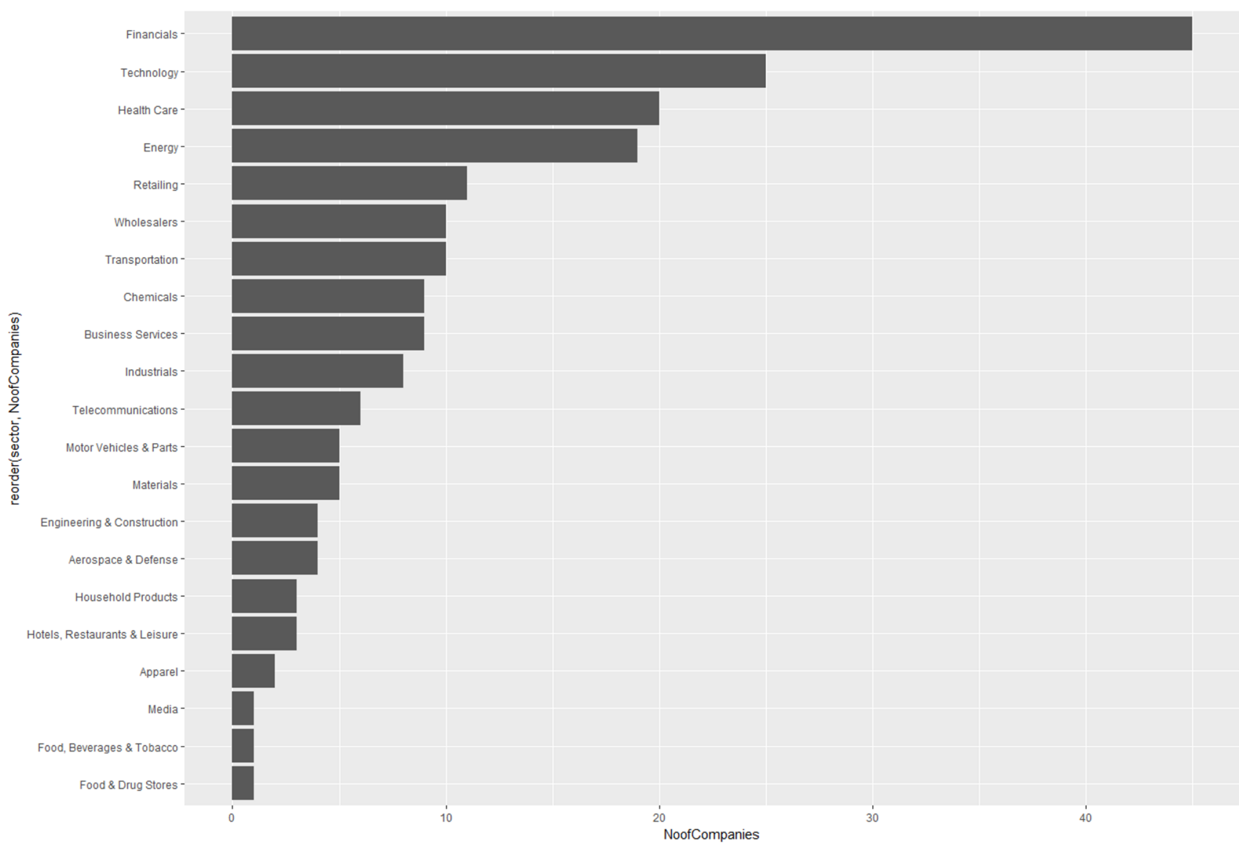

Figure 9: No of companies present per S&P 500 Sector present in the dataset.

- **Boxplots is plotted to display distribution of avg password length across various sectors.**

```
ggplot(sector.average.data, aes(x=sector, y=CharacterCount))
+ geom_boxplot()
+ coord_flip()
```

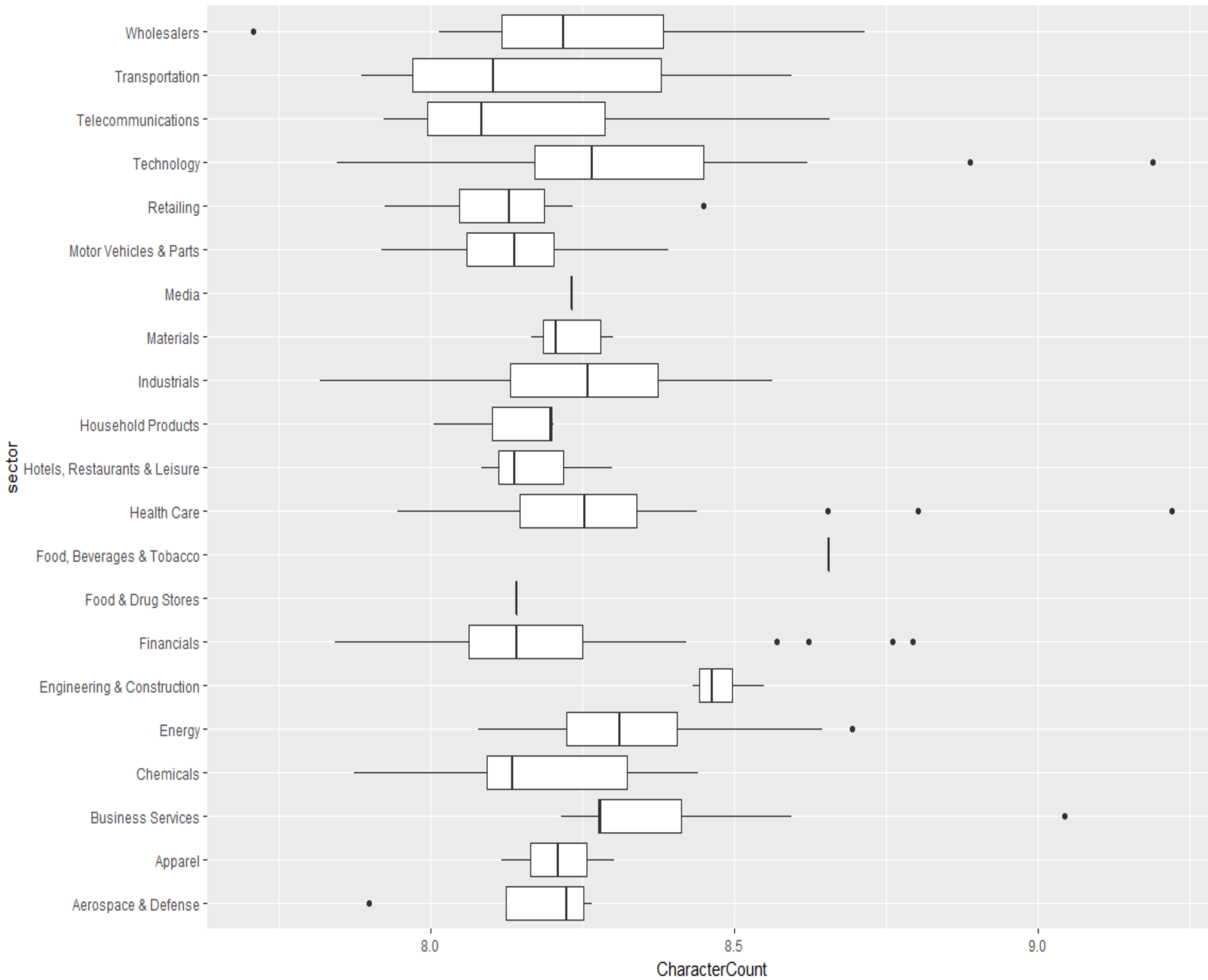

Figure 10: Box plot to display the avg password length across various S&P 500 sectors

7. Data Modelling

Goal of analyzing this dataset was to determine whether there was a relationship between passwords characteristics among people from 2 different continents in this case UK and Russia was taken as an example. Since both of these countries have different keyboard layout, another goal would be to understand if any language specific keywords such as Cyrillic characters are used which can help in predicting if the passwords are from a specific region. From the initial dataset, filtering was applied to separate data associated with 2 domains (gmail.co.uk and gmail.ru). Both of them have been sampled to the same size.

Additional variables were created from the password. Below is quick description on each of them.

- totalRecords = no of total records in the entire dataset.
- CharacterCount = No of total chars in passwords.
- LowercaseCount = No of small case letter in passwords.
- UppercaseCount= No of Uppercase letters in passwords.
- AlphaNumericCount = No of total alphanumeric characters in passwords.
- NumericCount = No of Numeric characters in passwords.
- CyrillicCount = No of Cyrillic characters (foreign chars/non-US keyboard) in passwords.
- PunctCount = No of Special characters in passwords.

Below is a quick representation of the base data associated with each email domain and relationship between them:

Aggregated Stats comparison between the 2 domains.

```
> ruvsukdomains
```

	EMailDomain	TotalRecords	CharacterCount	LowercaseCount	UppercaseCount	AlphaNumericCount	NumericCount	CyrillicCount	PunctCount
1	gmail.co.uk	114294	911194	642294	19205	909621	248122	0	1541
2	gmail.ru	217391	1901800	738458	33481	1870619	1001797	95859	26806

Figure 11: Aggregated Stats comparison between UK and RU domains

```
ggplot(fildetails_df, aes(x=EMailDomain, y=lower.alpha.count)) + geom_boxplot()
```

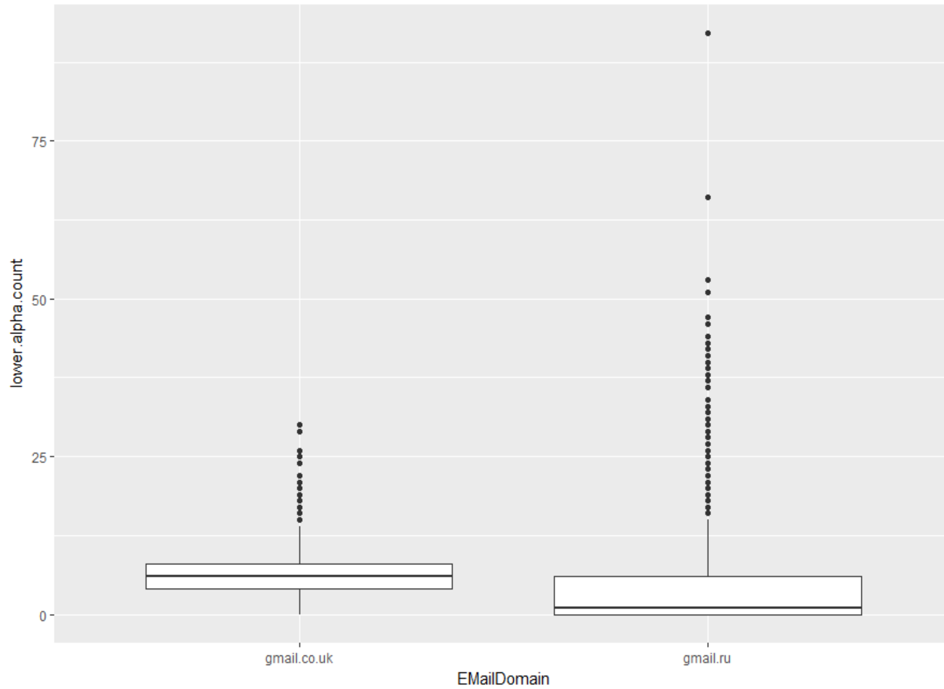

Figure 12: Boxplot of UK&RU domains for lower.alpha.count

```
ggplot(fildetails_df, aes(x=EMailDomain, y=upper.alpha.count)) + geom_boxplot()
```

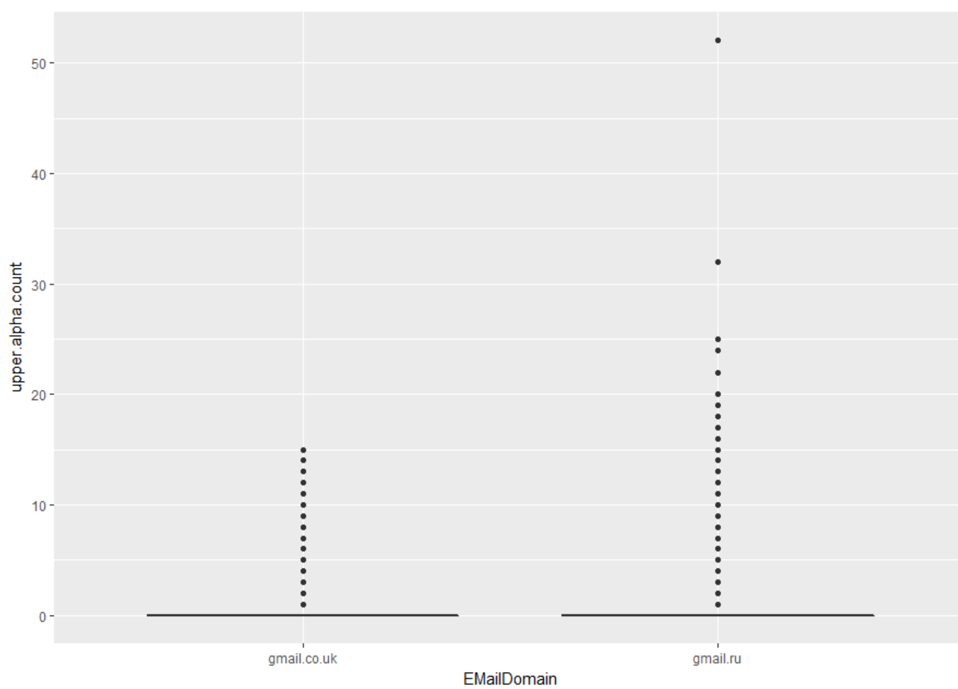

Figure 13: Boxplot of UK&RU domains for upper.alpha.count

```
ggplot(fildetails_df, aes(x=EMailDomain, y=numeric.count)) + geom_boxplot()
```

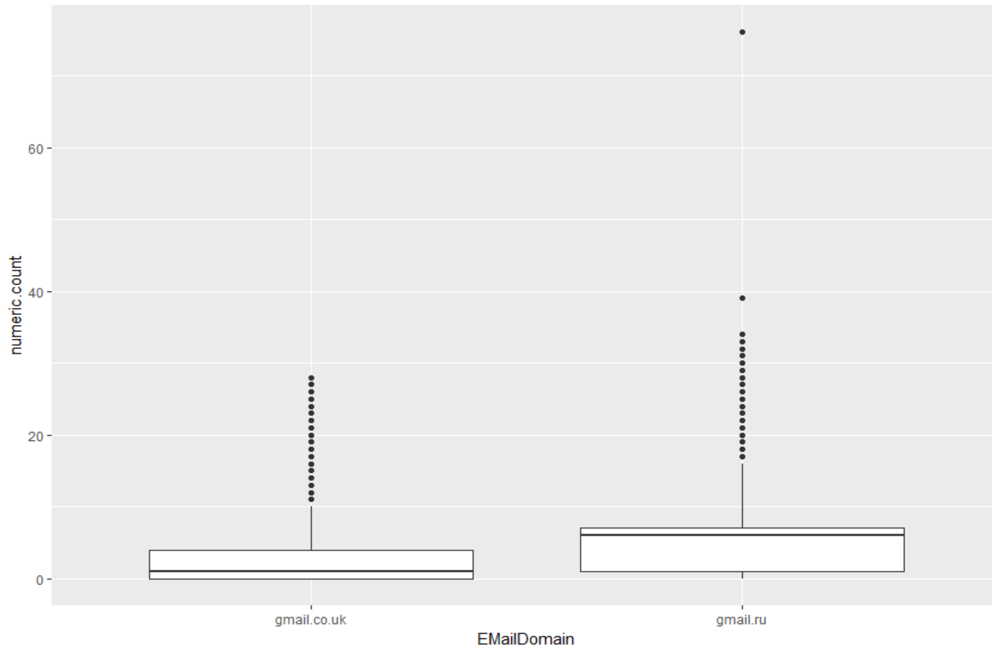

Figure 14: Boxplot of UK&RU domains for numeric.count

```
ggplot(fildetails_df, aes(x=EMailDomain, y=alphanumeric.count)) + geom_boxplot()
```

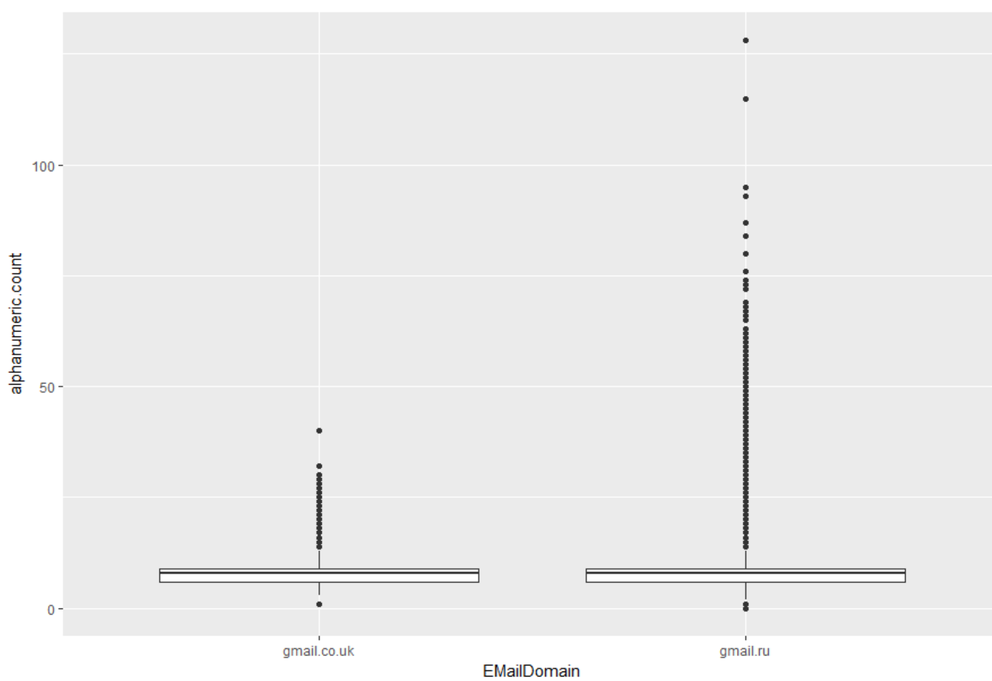

Figure 15: Boxplot of UK&RU domains for alphanumeric.count

```
ggplot(fildetails_df, aes(x=EMailDomain, y=punct.count)) + geom_boxplot()
```

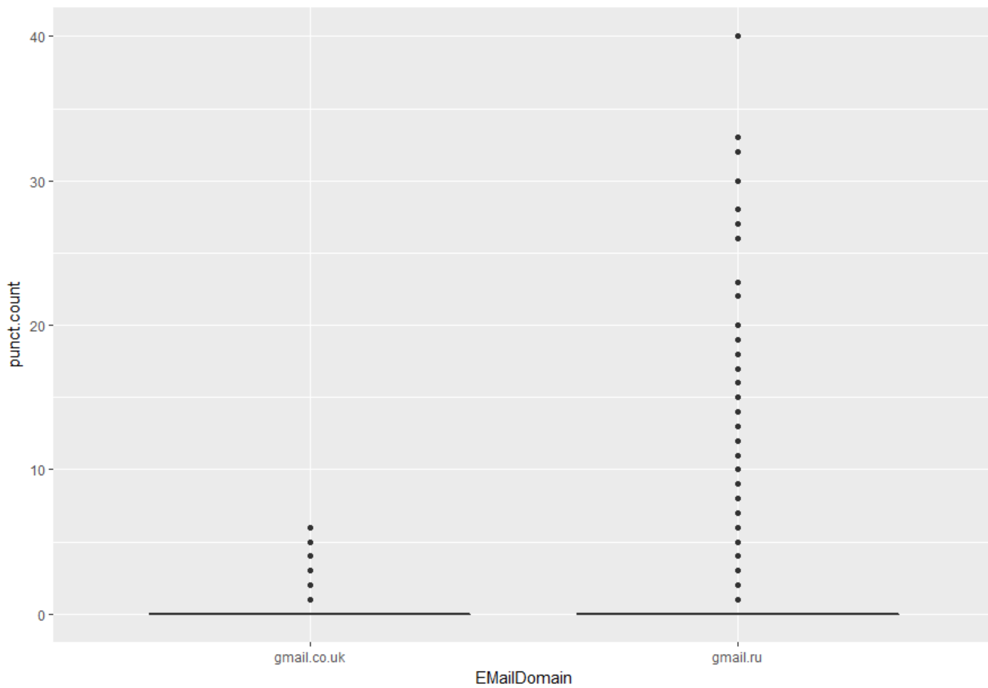

Figure 16: Boxplot of UK&RU domains for punct.count

```
ggplot(fildetails_df, aes(x=EMailDomain, y=cyrillic.count)) + geom_boxplot()
```

There are 0 cyrillic characters in uk domain email accounts as compared to ru domains.

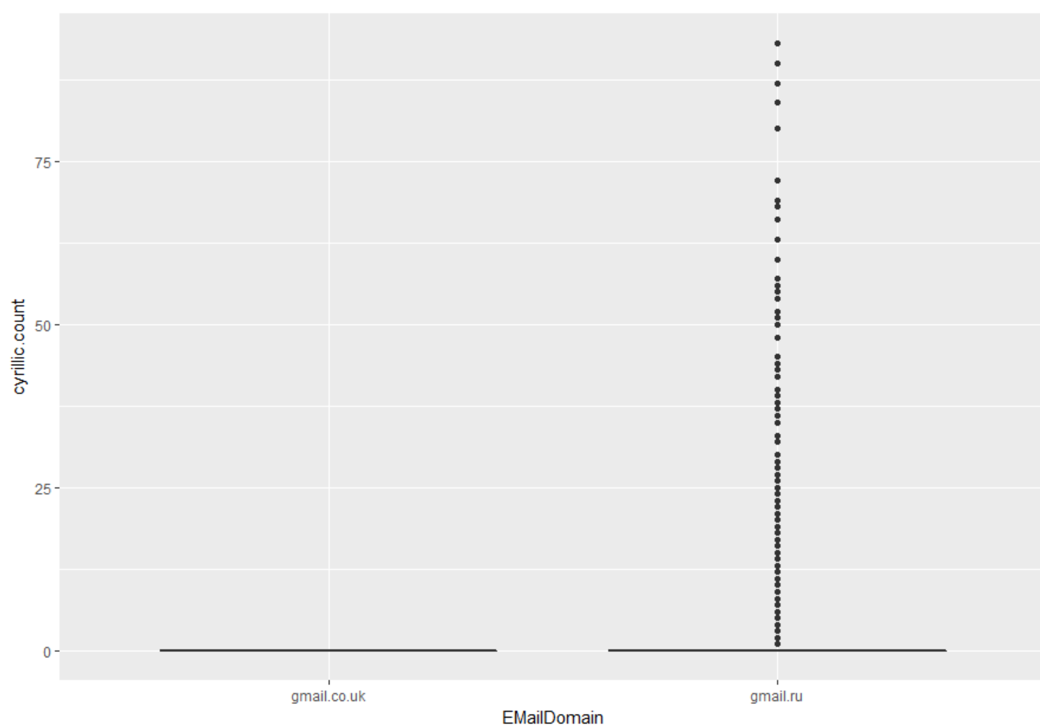

Figure 17: Boxplot of UK&RU domains for cyrillic.count

```
ggplot(fildetails_df, aes(x=EMailDomain, y=total.count)) + geom_boxplot()
```

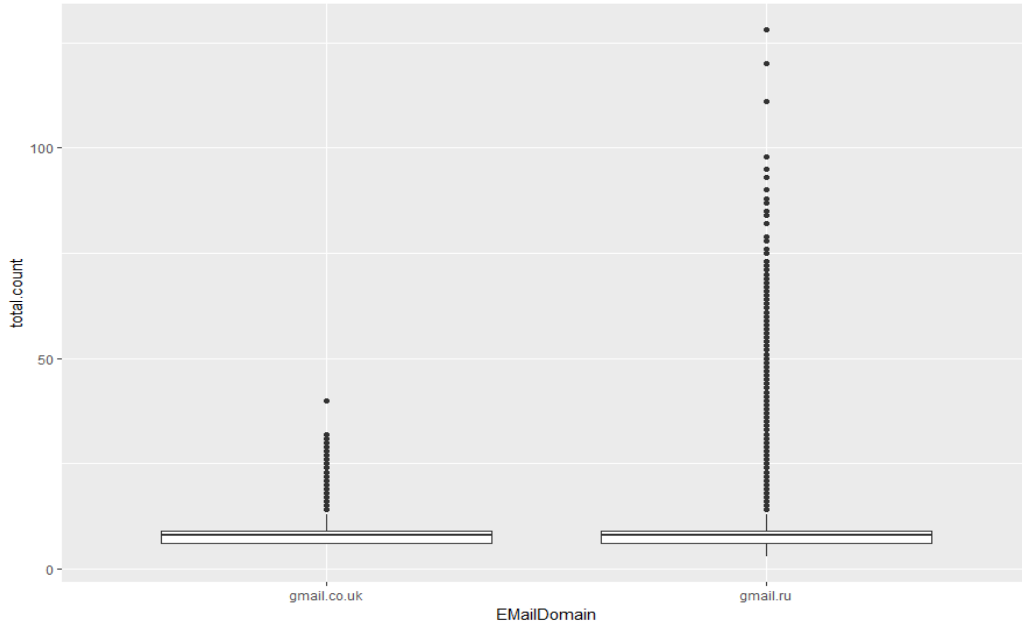

Figure 18: Boxplot of UK&RU domains for total.count

1. Logistic Regression

Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is the cumulative logistic distribution**.** In our dataset for modelling, we have binary dependent variable as Emaildomain and multiple independent variables around password characteristics.

2. Model fitting

We split the data into two chunks with 75 % as training and remaining 25% test dataset. The training set will be used to fit our model which we will be testing over the testing set.

```
datSplit <-  split(fildetails_df, fildetails_df$EMailDomain)

sampSize <- min(nrow(datSplit$gmail.co.uk), nrow(datSplit$gmail.ru))

datSplit[[1]] <- datSplit[[1]] %>% sample_n(sampSize)
datSplit[[2]] <- datSplit[[2]] %>% sample_n(sampSize)

datspli_sample<- bind_rows(datSplit)

set.seed(8675309)
datspli_sample <- datspli_sample %>%
mutate(TestTrain =  sample(c(0,1), size = nrow(datspli_sample),
      prob c(.75,.25), replace = T)
      , EMailDomain = as.factor(EMailDomain))

df_train <- datspli_sample %>% filter(TestTrain==0)
df_test <- datspli_sample %>% filter(TestTrain==1)
```

Compiling the formula which is a symbolic description of the model to be fitted with all numeric measurement variables from the summary dataset.

Emaildomain is predicted by these variables.

```
RegFormula <- as.formula("EMailDomain ~ lower.alpha.count + upper.alpha.count +
numeric.count + punct.count + cyrillic.count + total.count")
```

Figure 19: Logistic function of Linear Regression

Model object is created by passing the formula , training dataset with family set to binomial

```
LM1 <- glm(RegFormula,df_train ,family = "binomial")
LM1

Call:  glm(formula = RegFormula, family = "binomial", data = df_train)

Coefficients:
  (Intercept)  lower.alpha.count  upper.alpha.count      numeric.count        punct.count     cyrillic.count        total.count
      -1.1865            -1.5375            -1.4855            -1.2835            -0.4354            10.6879             1.5686

Degrees of Freedom: 176228 Total (i.e. Null);  176222 Residual
Null Deviance:      244300
Residual Deviance: 210500   AIC: 210500
```

Figure 20: Output of the model object after applying the formula

3. Interpreting the results of logistic regression model

Now in this section, we can analyze the fitting results and interpret model outputs. We have obtained the results of the model by running the summary function on the model object.

```
      lmSummary <- summary(LM1)
    > lmSummary

Call:
glm(formula = RegFormula, family = "binomial", data = df_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.8391  -0.9039  -0.7914   0.9643   1.6628

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)       -1.18647    0.02156 -55.025  < 2e-16 ***
lower.alpha.count -1.53755    0.19947  -7.708 1.28e-14 ***
upper.alpha.count -1.48552    0.19960  -7.443 9.88e-14 ***
numeric.count     -1.28345    0.19948  -6.434 1.24e-10 ***
punct.count       -0.43538    0.20270  -2.148   0.0317 *
cyrillic.count    10.68789   34.87605   0.306   0.7593
total.count        1.56858    0.19947   7.864 3.73e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 244305  on 176228  degrees of freedom
Residual deviance: 210453  on 176222  degrees of freedom
AIC: 210467

Number of Fisher Scoring iterations: 19
```

Figure 21: Summary output of the Model object

First of all, we can see that cyrillic count variable has no statistical significance and punct.count has also less significance as compared to other measurement variables.

There is a strong correlation between lower.alpha.count, upper.aplha.count and numeric.count which are inclined towards predicting towards gmail.co.uk domain where as total.count has strong correlation towards the other side predicting gmail.ru domains.

```
> df_test <- df_test %>%
+   mutate(Preds = ifelse(Predicted > Threshold, 0, 1)) %>% mutate(Email-
True=as.factor(EmailTrue), Preds=as.factor(Preds))

> confusionMatrix(df_test$EmailTrue, df_test$Preds)
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 21865  7435
         1 10932 18145

               Accuracy : 0.6854
                 95% CI : (0.6816, 0.6891)
    No Information Rate : 0.5618
    P-Value [Acc > NIR] : < 2.2e-16
    Kappa : 0.3704
     Mcnemar's' Test P-Value : < 2.2e-16
    Sensitivity : 0.6667
    Specificity : 0.7093
    Pos Pred Value : 0.7462
    Neg Pred Value : 0.6240
    Prevalence : 0.5618
    Detection Rate : 0.3745
    Detection Prevalence : 0.5019
    Balanced Accuracy : 0.6880
    'Positive' Class : 0
```

Figure 22: Confusion Matrix for Model Performance

Above confusion matrix for the model performance shows, the model is accurately predicting 70% of the time the email domain.

AUC value is calculated as 0.74

```
> auc <- performance(ROCRpredict, measure = "auc")
> auc <- auc@y.values[[1]]
> auc
[1] 0.7430779
```

Figure 23: Area Under the Curve Calculation of the Model

As a last step, we are going to plot the ROC curve and calculate the AUC (area under the curve) which are typical performance measurements for a binary classifier. The ROC is a curve generated by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. From the below ROC curve optimum threshold for the model is 0.4

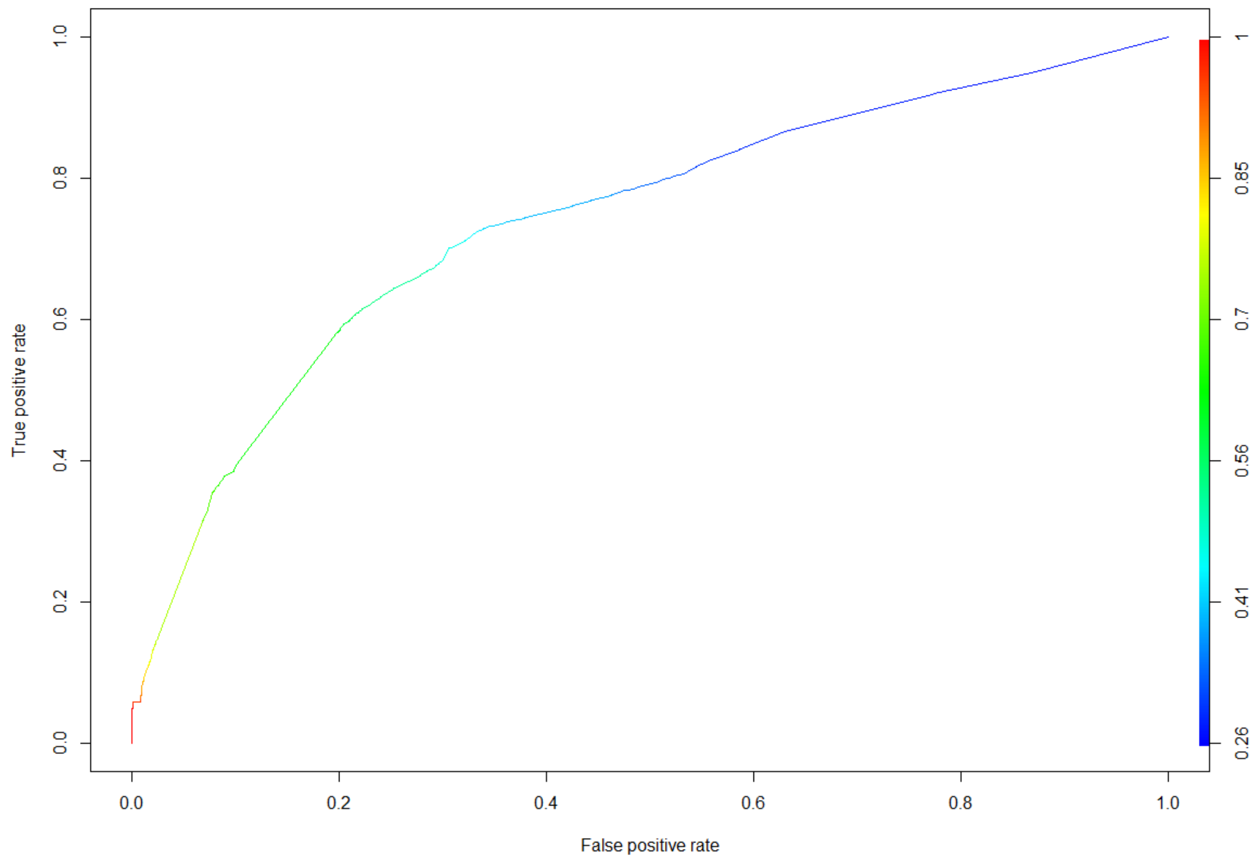

Figure 24: ROC Curve - Diagnostic capability of the model as its threshold is varied

To further compare the results, results were generated without the punct.count and Cyrillic.count which are statistically insignificant as previously generated lm formula.

```
newRegFormula <- as.formula("EMailDomain ~ lower.alpha.count + upper.alpha.count
+ numeric.count + total.count")
```

Figure 25: Logistic function of Linear Regression without the Statistically insignificant variables

```
newLM1 <- glm(newRegFormula,df_train ,family = "binomial")
newlmSummary <- summary(newLM1)

> newlmSummary

Call:
glm(formula = newRegFormula, family = "binomial", data = df_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.7836  -0.9033  -0.7913   0.9634   1.6620

Coefficients:
                  Estimate Std. Error z value            Pr(>|z|)
(Intercept)       -1.18242    0.02156  -54.85 <0.0000000000000002 ***
lower.alpha.count -1.17246    0.02966  -39.53 <0.0000000000000002 ***
upper.alpha.count -1.12025    0.03013  -37.17 <0.0000000000000002 ***
numeric.count     -0.91787    0.02960  -31.01 <0.0000000000000002 ***
total.count        1.20273    0.02943   40.86 <0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 244305  on 176228  degrees of freedom
Residual deviance: 210522  on 176224  degrees of freedom
AIC: 210532

Number of Fisher Scoring iterations: 8
```

Figure 26: Summary output of the Model object without the statistically insignificant variables

```
> Threshold <- 0.4
>
> df_test <- df_test %>%
+   mutate(Preds = ifelse(Predicted > Threshold, 0, 1)) %>%
mutate(EmailTrue=as.factor(EmailTrue), Preds=as.factor(Preds))
> library(caret)
> confusionMatrix(df_test$EmailTrue, df_test$Preds)
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 21864  7436
         1 10930 18147

               Accuracy : 0.6854
                 95% CI : (0.6816, 0.6892)
    No Information Rate : 0.5618
    P-Value [Acc > NIR] : < 0.00000000000000022

                  Kappa : 0.3705
 Mcnemar's' Test P-Value : < 0.00000000000000022

            Sensitivity : 0.6667
            Specificity : 0.7093
         Pos Pred Value : 0.7462
         Neg Pred Value : 0.6241
             Prevalence : 0.5618
         Detection Rate : 0.3745
   Detection Prevalence : 0.5019
      Balanced Accuracy : 0.6880

       'Positive' Class : 0

> auc <- performance(ROCRpredict, measure = "auc")
> auc <- auc@y.values[[1]]
> auc
[1] 0.7430366
```

Figure 27: Confusion Matrix and AUC values of New model object

8. Conclusion and Final Thoughts:

We started the data analysis of the large compilation of various data breaches with size over 41 GB including 1981 files. In the first section, we identified various data quality issues to reduce the scope of the analysis and input data. Rather than focusing on emailID to password combination, we aggregated dataset to the email domain along with the password values and generated several measurement variables associated with password. Also, with respect to emaildomain, with the help of data wrangling additional features were generated along with S&P 500 company and sector associated with it.

In the data modelling section, we have taken Emaildomain as dependent variables with multiple password characteristics as independent variables by estimating probabilities using a logistic function. After training dataset, Emaildomain was predicted on the test dataset for which model was able to correctly predict emaildomain 68 % of the time as per the confusion matrix results. New calculations were also generated without the statistically insignificant variables (Punct.count and Cyrillic.count) and nearly similar model accuracy rate generated with slight variation in the AUC values.

This model can be used to predict UK vs RU domains based on the passwords.The dataset of passwords can be compiled with the data wrangling to generate additional numeric variables such as Uppercase, lowercase, alphanumeric, numeric counts per password and then domain predictions can be generated with the help of the model and can achieve 68% accuracy rate.

While model is not perfect, it has still lot of room for improvement to improve accuracy and reduce false positive rate. Apart from existing numeric measurement variables, additional variables/features with more regional contextual information may have helped in improving the accuracy of the model and predicting the country. The dataset also does not give any confidence apart from the Emaildomain to associate it to specific region/ business sector. However one can still utilize this existing model to predict the region by training respective regional email domains dataset based on the given input password value.

Public Email Services/Corporate sectors does not usually handle or store passwords in cleartext formats but this data analysis can be used in strengthening existing security controls/ regional attributions in case of any investigations where only password is known.

9. Project Code References

- Project Code Repository: https://github.com/ashwin-patil/springboard-intro-to-datascience/tree/master/capstone_project
- Data Preparation : https://github.com/ashwin-patil/springboard-intro-to-datascience/blob/master/capstone_project/00_Data_Preparation/DataPreparation-FileStats.R
- Data Cleaning : https://github.com/ashwin-patil/springboard-intro-to-datascience/blob/master/capstone_project/01_Data_Cleaning/DataCleaning-Dups_NA_lessthan6.R
- Data Wrangling: https://github.com/ashwin-patil/springboard-intro-to-datascience/blob/master/capstone_project/02_Data_Wrangling/DataWrangling.R
- Data Visualization: https://github.com/ashwin-patil/springboard-intro-to-datascience/blob/master/capstone_project/04_Data_Visualization/DataVisualization.R
- Data Modelling: https://github.com/ashwin-patil/springboard-intro-to-datascience/blob/master/capstone_project/05_Data_Modelling/DataModeling-LogisticRegression.R