Northeastern University
College of Professional Studies

Does Tenure Impact Malicious Alerts?

Team 3

Ana Paniagua

Diana Puerta

Minyi Chen

Instructor: Jamie Warner

ALY 6980 Capstone

Spring 2019

June 22, 2019

Does Tenure Impact Malicious Alerts?

Data breaches seems to be a problem that affects most companies these days. The major reasons that cause data leak is due to human error, malware, insider misuser or theft of data carrying device. Companies such as Yahoo, Equifax, Marriot International, eBay have been victims of data breaches. The article Corporate Governance, Social Responsibility and Data Breaches narrates a study regarding the corporate governance and social responsibility related to data breaches. The study states that smaller companies with greater financial expertise are less likely to be breached.

Nevertheless, all companies can become victims of intentional or unintentional release of information (C. Lending, Kristina Minnick & P. Schorno, 2018, pg. 413). An example of data breach occurred in 2014 when Sonny announced a data breach caused by a hacking group, thus security incident contained the salaries of 6.000 employees and top executives as well as their social security number and credit card numbers (Silverman and Fritz, 2014). The company's reputation was damaged and affected the profits on films to be released at the time. In fact, PricewaterhouseCoopers states that there was a 73% increase in the number of data breaches from 2013 to 2014 (C. Lending, Kristina Minnick & P. Schorno, 2018, pg. 414).

The occurrence of data breaches is becoming more common and it leads to bad publicity, negative stock returns, bad reputation and long-term operational losses. Therefore, we will explore further the likelihood of tenure employees caused malicious alerts.

Exploratory Data Analysis

Features Created

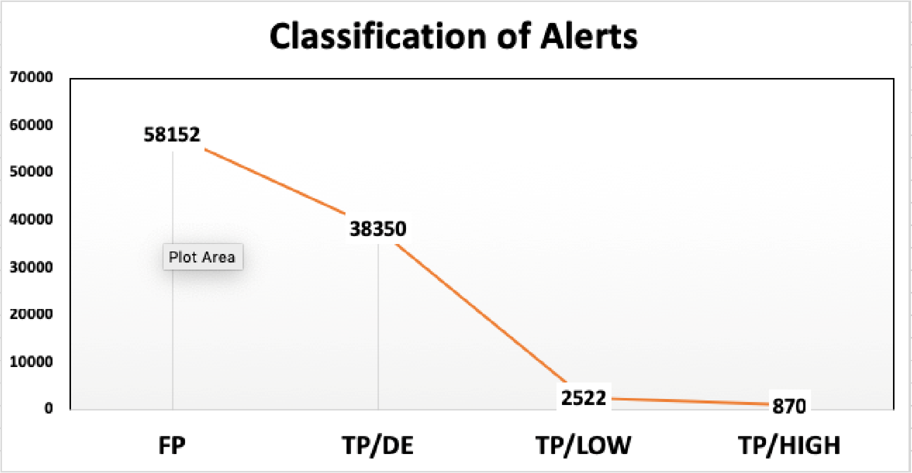
Days: how many days took an alert to be escalated from the time it was inserted into the system to the date it was escalated. Seniority years: years of experience working for GE**.** Seniority Level: whether the employee is an entry level, associate, senior or executive**.** The date column was divided into escalation date and escalation time to join the two datasets

Data Cleaning & Preparation

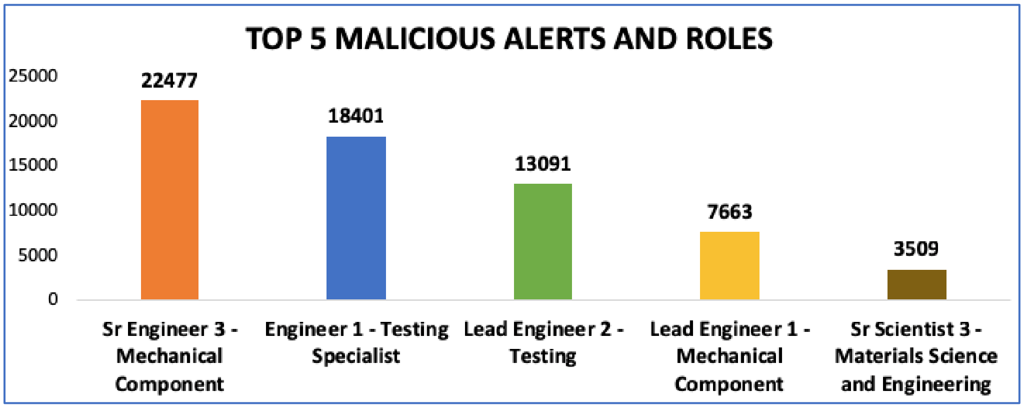
Missing values were handled by adding the average of the column to the missing values. The column date and time where missing values were added 0. Duplicates were removed in Excel and a join between the two datasets were done using the employee_id.

Exploratory Data Analysis Findings

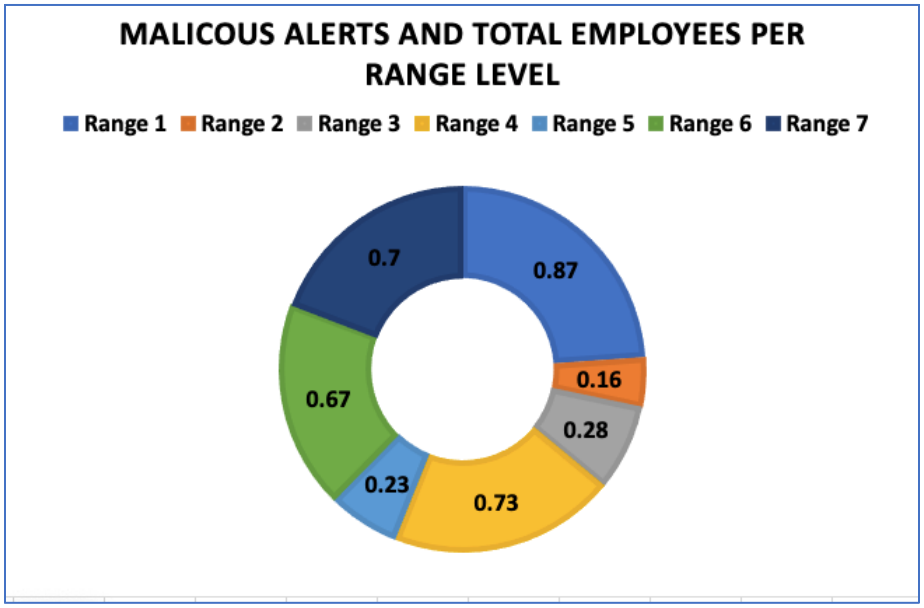
When looking at the classification we were able to notice that FP misfired alerts are the highest total classification alert with 58,152, TP/DE little to no risk 38,350 alerts, TP/High means high risk 870, TP/Low which means risk involved 2,522 alerts. It can be seen that the majority of alerts are indicators misfired or false alerts.



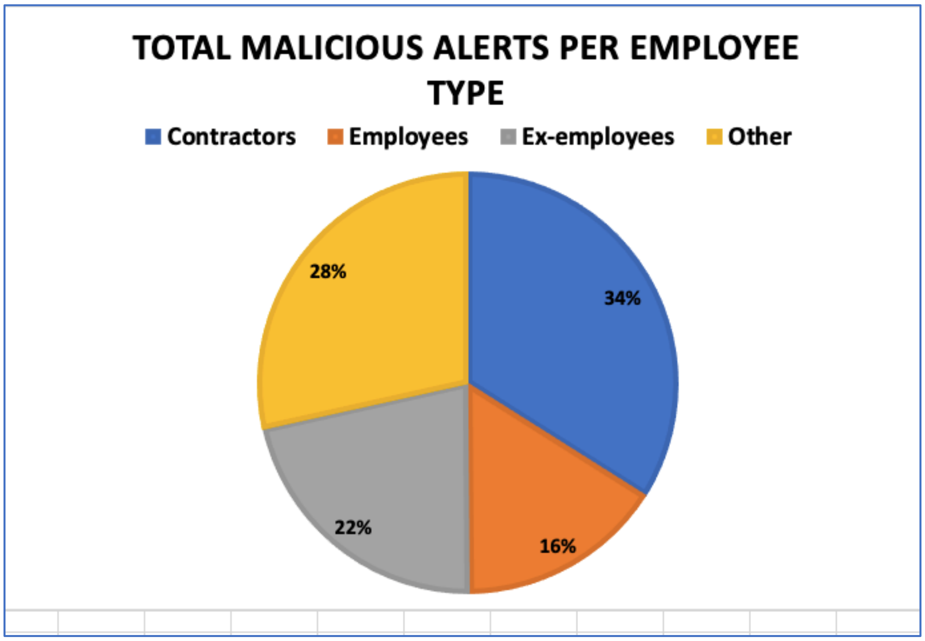
We wanted to explore further the malicious alerts based on the roles of those employees who caused the top 5 highest number of alerts. Employees working as Sr. Engineer 3 Mechanical Component have caused 22,477 malicious alerts. Engineer 1 Testing Specialist have caused 18,401 malicious alerts. Lead Engineer 2 Testing have caused 13,091. Lead Engineer 1 Mechanical Component 7,663 malicious alerts and Sr. Scientist 2 Material Science and Engineering has 3,509 malicious alerts.



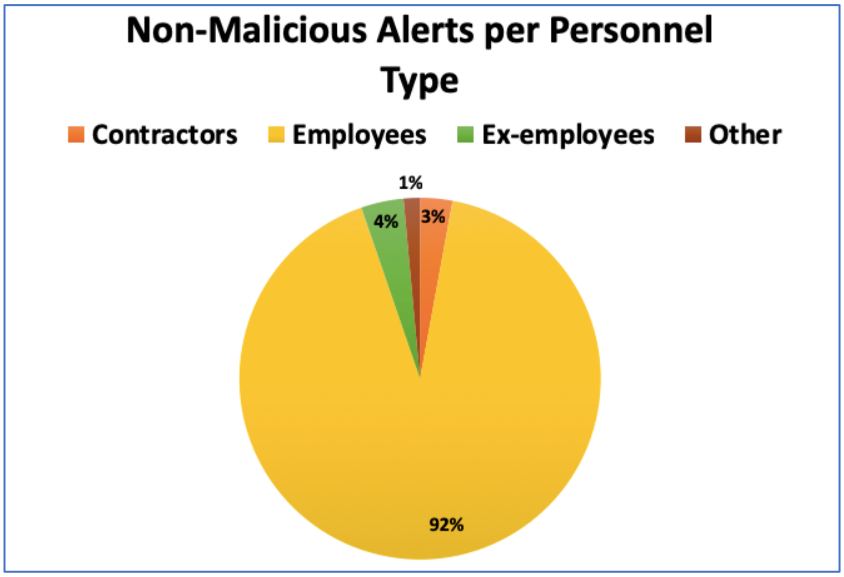
The largest number of malicious alerts based on employees including all types of alerts except for FP as this is considered indicators misfired. We found out that Range 1 which includes those employees who have been working from 0 to 5 years are the ones who have triggered the highest number of malicious alerts when we normalized the data, there are 0.87 or 87%. Followed by Range 4 which includes 19 to 23 years of seniority equals to 73%. The third one is Range 7 which includes employees that have been working from 36 to 41 years for GE with a 70%. These are the three major ranges we should pay close attentions to as they have caused the majority of the malicious alerts.



We also wanted to find out which type of employee have caused the highest number of malicous alerts. We have out that it was Contractors, followed by Other which might include active employees the person type is unknown and employees.



In terms of non-malicious alerts we found out that employees caused the highest number of alerts 92%, 4% former employees and 3% contractors and other refers to those we are not sue what person_type fall under.



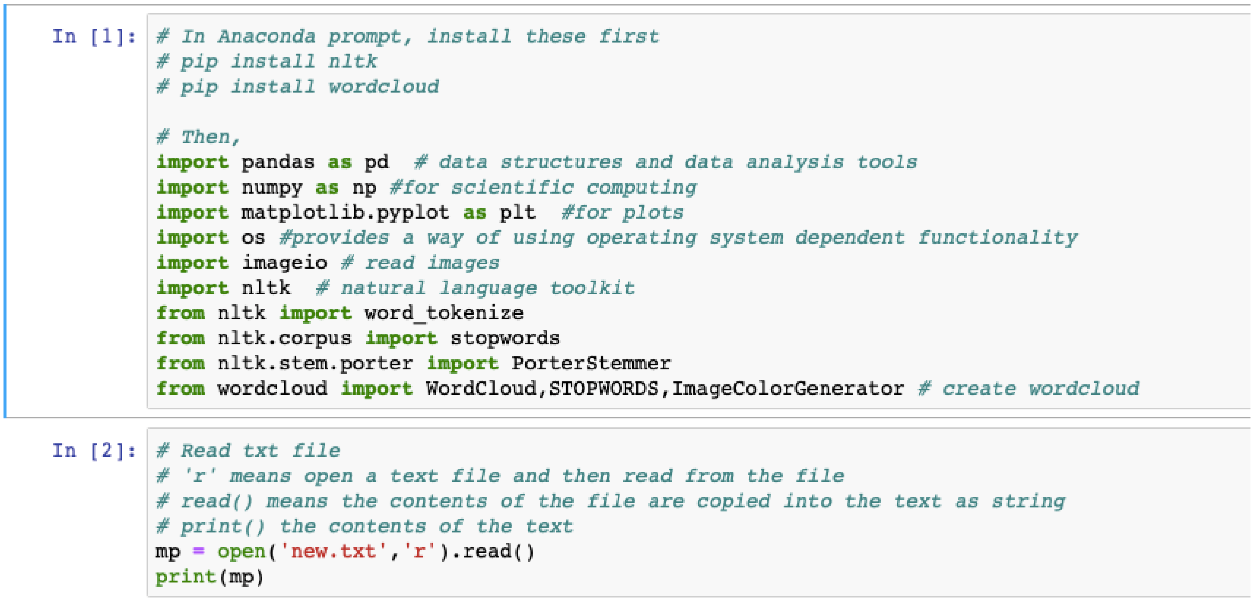
The data exploratory was done in Excel and Tableau.

Text Mining

According to studies classifying textual data is beneficial for companies to understand a massive information and leverage value. The process includes accessing, analyzing, and annotated information. However, to identify patterns and extract valuable information data is analyzed. However, this approach can be challenging since, some digital documents consist of unstructured text containing data, rather than structure data. During this project, our group utilized ***text mining*** techniques to automate the text processing and derive useful insights from the unstructured GE dataset.by following the following steps we used the ***alert_type*** and ***Indicators*** columns to visualize the most frequent words according to the dataset.

Text Cleaning Step Process

1. Remove stop words, since this helps in sentence construction.
2. Convert to lower, to maintain a standardization across all text and we got rid of case differences and convert the entire text to lower case.
3. Remove punctuation.
4. Remove numbers from the text.
5. Remove _ and whitespaces in the text.
6. Stemming and Lemmatization to convert the terms into their root form.

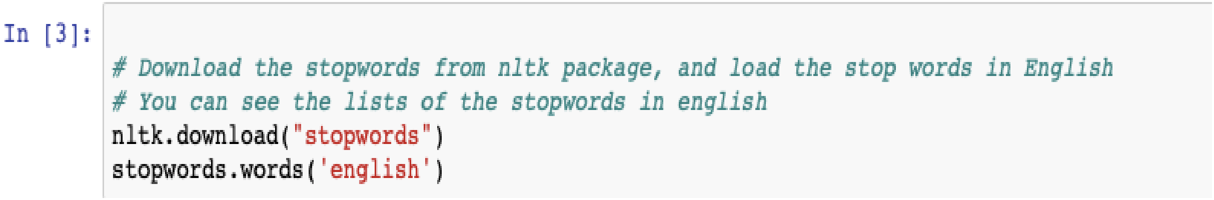


In [1]:

```
# In Anaconda prompt, install these first
# pip install nltk
# pip install wordcloud

# Then,
import pandas as pd  # data structures and data analysis tools
import numpy as np #for scientific computing
import matplotlib.pyplot as plt  #for plots
import os #provides a way of using operating system dependent functionality
import imageio # read images
import nltk  # natural language toolkit
from nltk import word_tokenize
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from wordcloud import WordCloud,STOPWORDS,ImageColorGenerator # create wordcloud
```

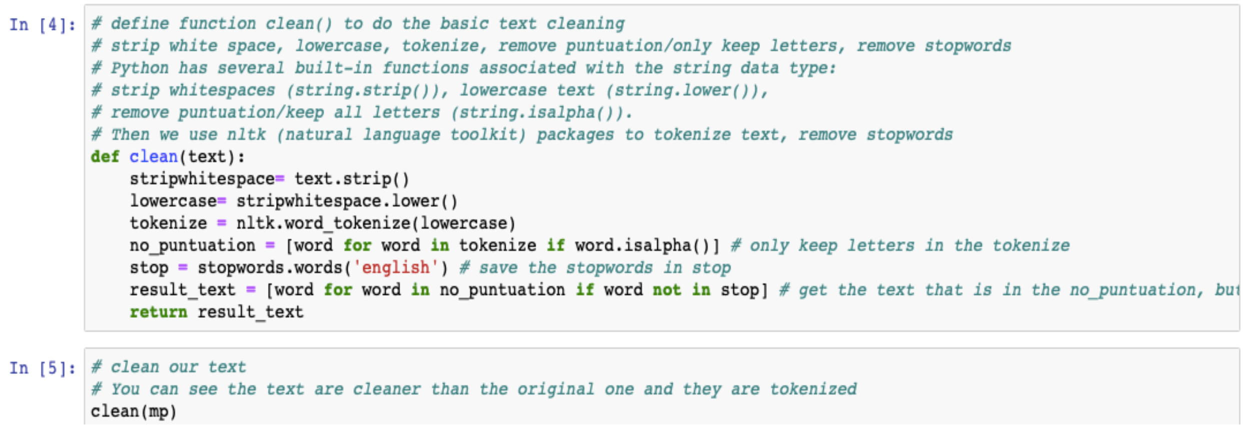
In [2]:

```
# Read txt file
# 'r' means open a text file and then read from the file
# read() means the contents of the file are copied into the text as string
# print() the contents of the text
mp = open('new.txt','r').read()
print(mp)
```

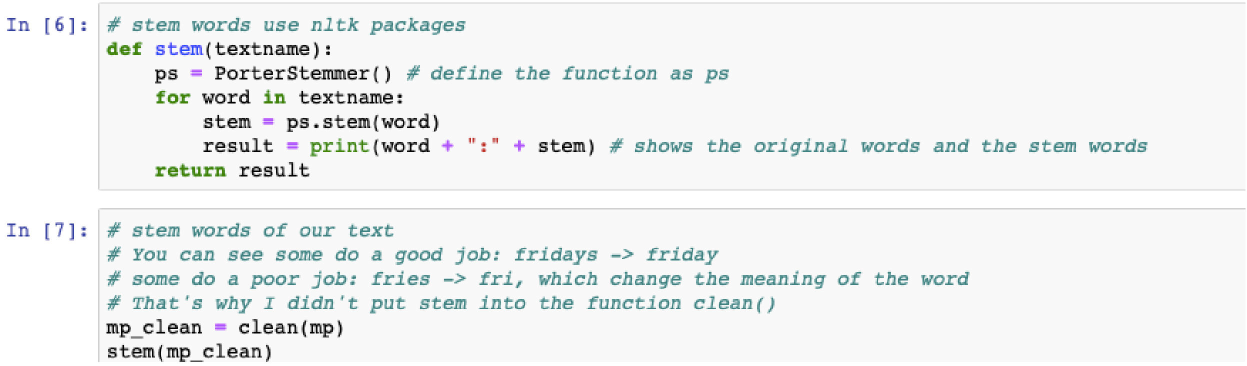


In [3]:

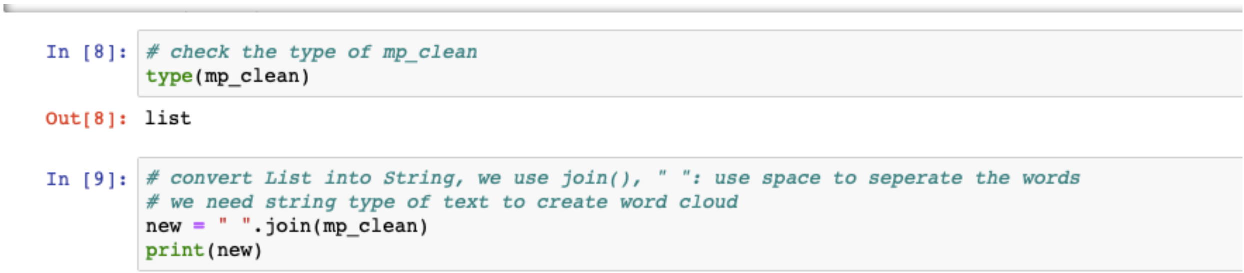
```

# Download the stopwords from nltk package, and load the stop words in English
# You can see the lists of the stopwords in english
nltk.download("stopwords")
stopwords.words('english')
```



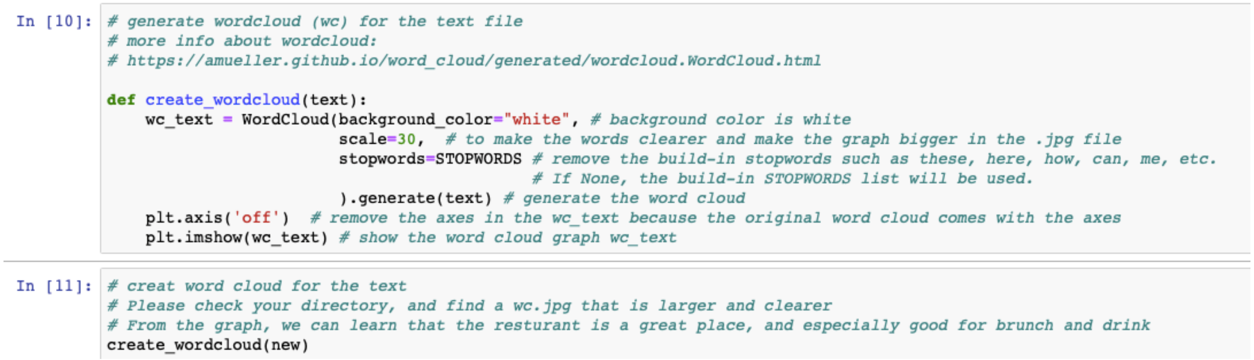
```
In [4]: # define function clean() to do the basic text cleaning
        # strip white space, lowercase, tokenize, remove puntuation/only keep letters, remove stopwords
        # Python has several built-in functions associated with the string data type:
        # strip whitespaces (string.strip()), lowercase text (string.lower()),
        # remove puntuation/keep all letters (string.isalpha()).
        # Then we use nltk (natural language toolkit) packages to tokenize text, remove stopwords
        def clean(text):
            stripwhitespace= text.strip()
            lowercase= stripwhitespace.lower()
            tokenize = nltk.word_tokenize(lowercase)
            no_puntuation = [word for word in tokenize if word.isalpha()] # only keep letters in the tokenize
            stop = stopwords.words('english') # save the stopwords in stop
            result_text = [word for word in no_puntuation if word not in stop] # get the text that is in the no_puntuation, but
            return result_text

In [5]: # clean our text
        # You can see the text are cleaner than the original one and they are tokenized
        clean(mp)
```



```
In [6]: # stem words use nltk packages
        def stem(textname):
            ps = PorterStemmer() # define the function as ps
            for word in textname:
                stem = ps.stem(word)
                result = print(word + ":" + stem) # shows the original words and the stem words
            return result

In [7]: # stem words of our text
        # You can see some do a good job: fridays -> friday
        # some do a poor job: fries -> fri, which change the meaning of the word
        # That's why I didn't put stem into the function clean()
        mp_clean = clean(mp)
        stem(mp_clean)
```



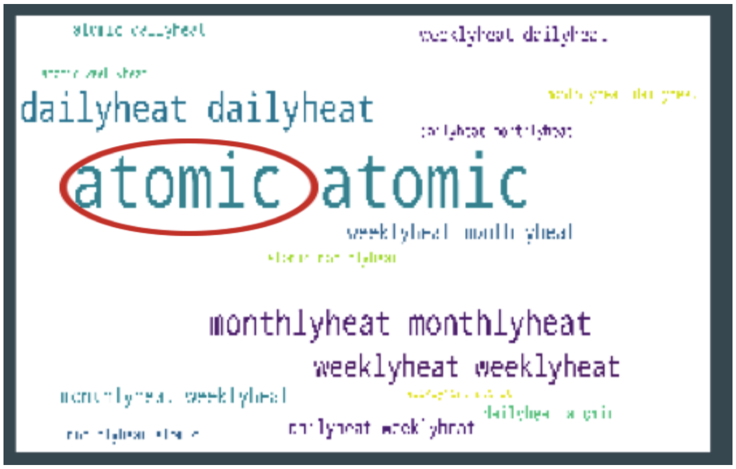
```
In [8]: # check the type of mp_clean
        type(mp_clean)

Out[8]: list

In [9]: # convert List into String, we use join(), " ": use space to seperate the words
        # we need string type of text to create word cloud
        new = " ".join(mp_clean)
        print(new)
```



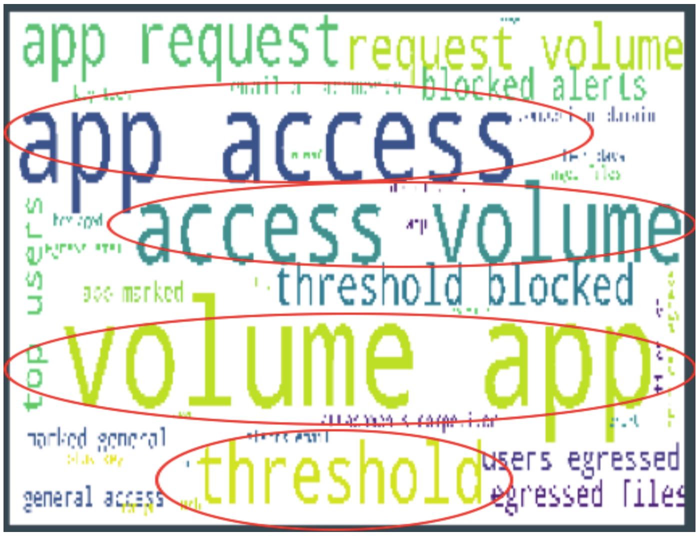
```
In [10]: # generate wordcloud (wc) for the text file
         # more info about wordcloud:
         # https://amueller.github.io/word_cloud/generated/wordcloud.WordCloud.html

         def create_wordcloud(text):
             wc_text = WordCloud(background_color="white", # background color is white
                                 scale=30,  # to make the words clearer and make the graph bigger in the .jpg file
                                 stopwords=STOPWORDS # remove the build-in stopwords such as these, here, how, can, me, etc.
                                                     # If None, the build-in STOPWORDS list will be used.
                                 ).generate(text) # generate the word cloud
             plt.axis('off')  # remove the axes in the wc_text because the original word cloud comes with the axes
             plt.imshow(wc_text) # show the word cloud graph wc_text

In [11]: # creat word cloud for the text
         # Please check your directory, and find a wc.jpg that is larger and clearer
         # From the graph, we can learn that the resturant is a great place, and especially good for brunch and drink
         create_wordcloud(new)
```

Subsequently, we created a word cloud by using the text mining package (*tm*) and the word cloud generator package (*word cloud*) available in python. This step helped us to analyze texts and to visualize the keywords as a word cloud.

According to the **word cloud** visual representation of text data and the most frequent words of the alert_typefield were the following: Atomic indicator, fires an alert. In addition, other common heat indicators that have a score assigned to the alerts that are also broken down into Daily, Weekly, and Monthly heat Alerts.



Following, we analyzed the Indicator column **word cloud** to have a clear visual.



As a result, the most frequent words were the following: Volume app, app access, access volume, threshold. Additionally, other indicators were: app request, request Volume, blocked alerts, threshold blocked, and top users.

Therefore, this information can be helpful since, the indicators that appear more frequently are the ones that are fired more often as the text mining analysis shows. Until here, we've chopped the description variable and extracted the most features out of it. However, Considering the massive volume of content being generated by General Electric aviation daily heat alerts, this analysis it’s going to be a surge in demand for people who are well versed with text mining and the natural language processing. As a result, this analysis shows that there needs to be a deeper investigation of the reasons why these alerts are fired more frequently

Classification Models

It will be helpful if GE can use a model to predict whether an alert has malicious intent, so that GE could conduct different solutions for alerts with and without malicious intents. We use different classification models to predict whether the alert involve malicious intent. From the results of the models, we are interested in finding the effect of employees’ tenure on malicious intent. Our hypothetical thought is that the longer the employees’ tenure is, the less likely they will have malicious intent. In this modeling section, there are four main steps. First, data cleaning and preparation. Second, modeling. Third, results and analysis. Fourth, discussion.

Data Preparation

The original dataset processed_alerts_obfuscated_v2.csv from sponsor GE includes redundant data so that we first need to get rid of the redundant data. GE sponsor says that the column “indicators” is derived by the column “incidcator_pairs”. Since our model doesn’t need the “indicators” column, we are going to make the dataset back to normal without redundant data rows. In order to do this, I first create a new column named “alert_id” which give the rows with the same “alert_escalation_date”, “alert_id_fk”, and “insert_date” the same alert_id. Then, I only keep one row of each unique alert_id. My sample size reduced from the original 127710 rows to 100467 rows. The reason I created this new column alert_id is that I don’t want to delete more than I should. I realized that even the same alert_id_fk could have different alert_escalation_date, such as the alert_id_fk: 2120096302. Many of them have this similar situation. Next, I checked the unique values of each column in this dataset and decide to keep score, owner_name, alert_type, classification as part of my input features in the baseline predictive models.

For the second dataset demographics.csv. I create new columns “Today”, “days”, “work_years”, and “tenure_level” in the Excel. Today column has the same value as “4/10/2019 12:00:00 AM”, which is the same format as the GE_Hire_Date. Days columns is the difference between Today and GE_Hire_Date, which means the days the employees have worked. Then, I coverted the days into work_years by dividing them into 365. I only keep the rows that are employees. We are not interested in ex-employees and contractors since they don’t have either the hired date or end date. Tenure_level is 1 (work years=[0,6]), 2 (work years=(6,11]), 3 (work years = (11,17]), 4 (work years=(17,23]), 5 (work years = (23,29]), 6 (work years=(29,36]), 7 (work years is greater than 36). I used this level based on our EDA analysis. In Python, I keep columns work_years, tenure_level and function_group in the predictive model. Work_years is our interested tenure input feature. Function_groups shows that whether the employee works in Commercial, Enabling, or Production. The reason I keep the input features is they might help explain the malicious intent.

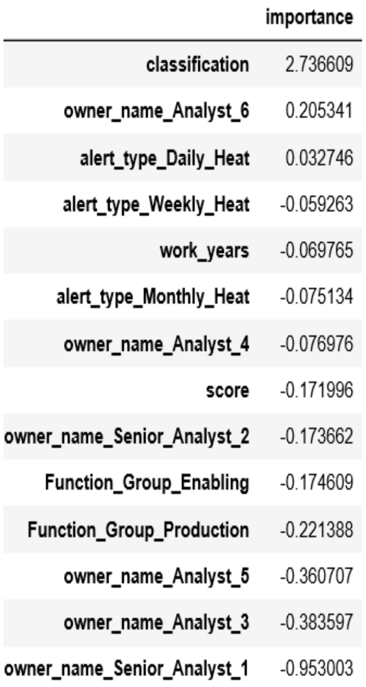
After deciding the input features, I started feature engineering and organize all the input features and output label that I keep into a new dataset. First, I changed the output label malicious from yes/no to 1/0 in the data frame. Second, I merged the two datasets by their employee ID. Third, I checked the missing values of all the data. Fourth, I treated the classification feature as ordinal categorical features and label each category from FP, TP/DE, TP/LOW, TP/HIGH to 0,1,2,3 respectively. Fifth, I use one-hot encoding method to clean the nominal categorical features. That means, I will create a new column for each unique value in each column of the nominal categorical features. At the end, I have a dataset with sample size of 92519 rows and 15 columns. The prevalence of the positive class is 38.8% which means 38.8% of the dataset has malicious intent.

The last step of preparing data is to split the sample into 70% training data, 15% validation data, and 15% test data. Training data set is used to train our model. Validation data set is used to see how we can improve our model. Testing data set is used to see how well the model is.

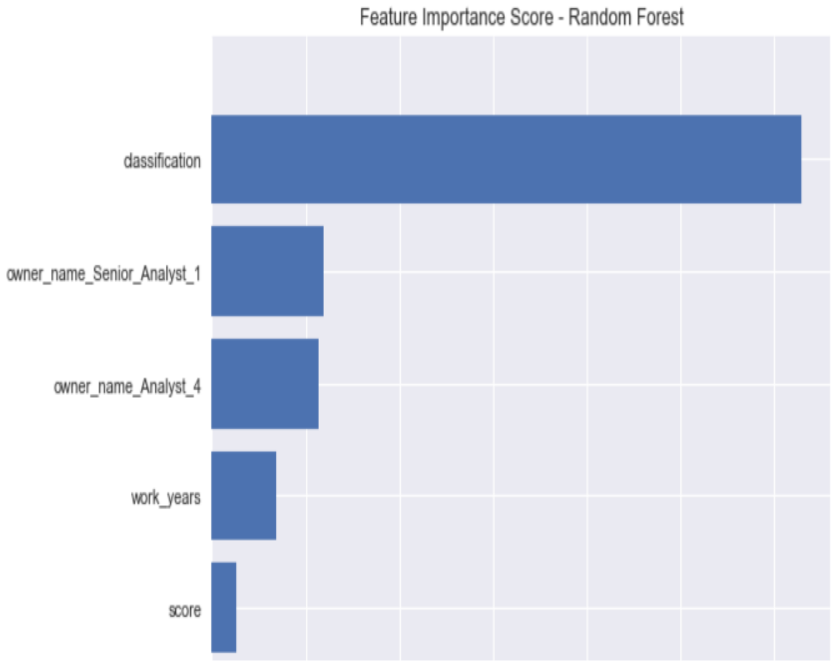
Methods and Results

Models using variable work_years. In this project, I use five classification models: K-nearest neighbors (KNN), logistic regression (LR), decision tree (DT), random forest (RF), and gradient boosting classifier (GBC). Then I will use AUC function to evaluate the performance of the models using the validation dataset. Surprisingly, the models all seem to be perfect since their AUC of the validation data set are all 1 expect for the logistic regression which is AUC=0.979, and random forest which is AUC=0.999. The proportion of positives that are correctly identified is high in all models. All models are 100% identified expect for KNN classifier is 99.7% identified.

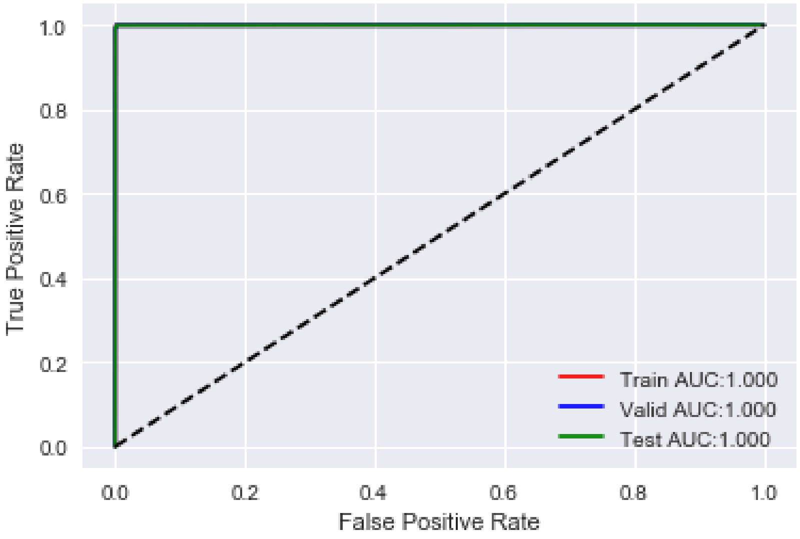
Next, I examined the input feature importance of logistic regression and random forest model. The logistic regression suggests us that the tenure has negative effect on malicious intent. That means, the longer the people work, the less chance they have malicious intent. The logistic regression also suggests that classification has positive effect on malicious intent. That means, the higher the classification type, the more likely the alert involves malicious intent. Random forest model (Figure 1b) suggests that classification is very importance feature in deciding the malicious intent, and tenure (work_years) is the fourth important feature.



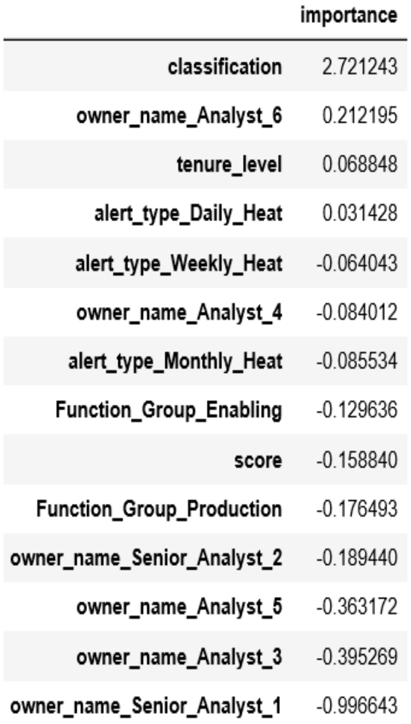
	importance
classification	2.736609
owner_name_Analyst_6	0.205341
alert_type_Daily_Heat	0.032746
alert_type_Weekly_Heat	-0.059263
work_years	-0.069765
alert_type_Monthly_Heat	-0.075134
owner_name_Analyst_4	-0.076976
score	-0.171996
owner_name_Senior_Analyst_2	-0.173662
Function_Group_Enabling	-0.174609
Function_Group_Production	-0.221388
owner_name_Analyst_5	-0.360707
owner_name_Analyst_3	-0.383597
owner_name_Senior_Analyst_1	-0.953003



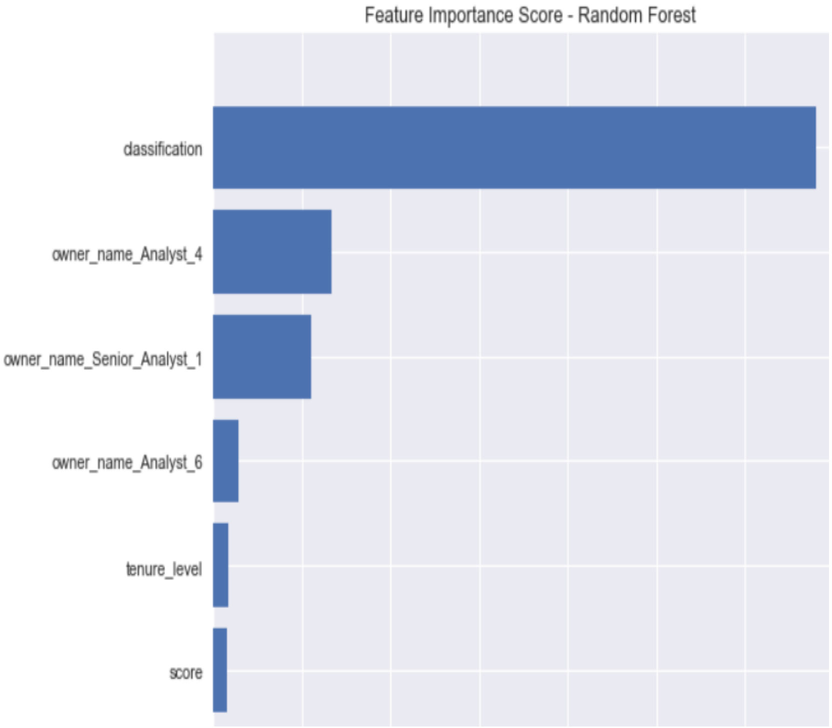
The last step is to evaluate the performance of the model. We pick the gradient boosting classifier (GBC) as my best classifier, and then we test the GBC model performance using the testing data set. The results turn out that the model is very good.



Models using variable tenure_level. We also use tenure_level instead of work_years in our predictive model to see what results will change. Everything else the same, we replace the work_years variable as tenure_level as described before. Interestingly, we found contradict results. See graphs below. The new model using tenure_level suggests that the higher the tenure_level is, the higher chance of malicious intent.



	importance
classification	2.721243
owner_name_Analyst_6	0.212195
tenure_level	0.068848
alert_type_Daily_Heat	0.031428
alert_type_Weekly_Heat	-0.064043
owner_name_Analyst_4	-0.084012
alert_type_Monthly_Heat	-0.085534
Function_Group_Enabling	-0.129636
score	-0.158840
Function_Group_Production	-0.176493
owner_name_Senior_Analyst_2	-0.189440
owner_name_Analyst_5	-0.363172
owner_name_Analyst_3	-0.395269
owner_name_Senior_Analyst_1	-0.996643



Analysis

Our predictive models using variable “work_years” suggests that longer tenure will reduce the chance of having malicious intent. It is interesting if we pick the people with very long tenure out of the dataset and look at their demographic information. We select the rows that the work_years is greater than 40 and malicious is 1. We then count the sample by employee_id. We can see employee “3930955969” has the most malicious count. His title is “Sr Product Management Manager 3 - Hardware Owner”. He works in the commercial group in the US. Other employees in the list also have senior titles and work in the US but in different departments and function groups.

We also investigate the rows that the work_years is greater than 40 and malicious is 0. Interestingly, we found that some employees overlap with the previous group. This group of employees also have senior titles, work in different function groups, states, and departments. It doesn’t seem to have a pattern for this group of employees who have worked more than 40 years besides they all have senior titles. This also suggests that malicious intent might depends on other factors that are not in our data set, such as their quality of work, their relationship with their co-workers, their personalities, their incomes, etc.

Employees that have worked more than 40 years		
employee_id	# of alerts with malicious	# of alerts without malicious
3225522111	7	14
3318674302	6	2
3930955969	134	67
5568769466	22	6
14164700621	0	2
17540620514	1	6
6619074875	2	0
16455521187	1	0

Discussion

There are a few things we need to be understand when we interpret these findings. Malicious intent is recorded as an analyst's believe. Analysts could have bias on the malicious based on their ethical believes. Assuming analysts are correctly label all the malicious intent. The ethical problems could be affected by not only tenure, but also other factors such as their quality of work, the relationship between co-workers, their value in the companies, and their personality traits, etc.

In the literature, the findings of how tenure affect the business ethics is mixed as well. Ardichvili, Jondle, and Kowske (2012) find that executives with longer tenure tend to have better business ethics while mid-level managers and non-managerial employees have less business ethics as they work longer. Neswiswi (2014) use a non-parametric Kruskal Wallis Test to find that there is positive relationship between ethics and tenure. Ng and Feldman (2013) find no significant relationship between tenure and job performance. One of their job performances includes organizational citizenship behavior. These contradict findings in the literature make sense. There are two effects here. First, people who work longer could perform better because they have better skills. Second, people work longer could also be bored about the work and gain less interested in the company. It depends on which effect is dominated that the tenure will lead to which result.

Conclusion

In our project, we try to understand the relationship between tenure and malicious intents. We first use EDA to explore the dataset and find that tenure level at range 1, 4, and 7 cause most of the malicious intent. Second, we use text mining to see what indicators happens most in the dataset. It turns out that volume app, app access, access volume, threshold are the most frequent indicators that trigger the alerts. Third, we use predict models to predict whether an alert has malicious intent. From the results, we find that long tenure will lower the malicious intent while higher tenure_level will increase the malicious intent. These findings are consistent with the findings in the literature. We suggest that there are other factors that affect the malicious intent and they are not in our data set. Those factors could be quality of work, the relationship between co-workers, their value in the companies, and their personality traits, their income, etc.

References

Ardichvili, A., Jondle, D., & Kowske, B. (2012). Minding the gap: Exploring differences in perceptions of ethical business cultures among executives, mid-level managers and non-managers. *Human Resource Development International*, 15(3), 337-352.

Lending C, Minnick K, Schorno PJ. Corporate Governance, Social Responsibility, and Data Breaches. *Financial Review*. 2018;53(2):413-455. doi:10.1111/fire.12160.

Ng, T. W., & Feldman, D. C. (2013). Does longer job tenure help or hinder job performance?. *Journal of Vocational Behavior*, 83(3), 305-314.

Neswiswi, H. (2014). *Employee attitude towards business ethics in the motor industry* (Doctoral dissertation, University of Pretoria).

Moro, S., Cortez, P., & Rita, P. (2015). Business intelligence in banking: A literature analysis from 2002 to 2013 using text mining and latent Dirichlet allocation. *Expert Systems with Applications*, *42*(3), 1314-1324.

Seibel, J. C., Feng, Y., & Foster, R. L. (2008). *U.S. Patent No. 7,315,861*. Washington, DC: U.S. Patent and Trademark Office.

Ur-Rahman, N., & Harding, J. A. (2012). Textual data mining for industrial knowledge management and text classification: A business oriented approach. *Expert Systems with Applications*, *39*(5), 4729-4739.