Department of Sociology

Master's Degree

Data Science

Computational Social Sciences Project

The importance of sentiment in scam filtering

A study proposal on statistical and machine learning models that also include sentiment analysis

Student

Maurizio P. De Marchi (230654)

Academic Year 2021/2022

INTRODUCTION

"*Phishing is the act of electronically and deceitfully contacting a person, with the aim of making the person electronically perform an act that is beneficial to the deceiver and harmful to the deceived. Phishing incidents typically occur by email*" [4]. In the current era, phishing is becoming more of a problem than ever. Phishing and scam attempts seem to have been growing [6], to the point that also companies famous for security awareness can fall victims [5]. Finding new ways to efficiently block them will be a key-point to avoid scammers' social engineering acts and save web users from economical demise. Given the context, detecting and taking into consideration the social engineering attempt might be the key to improve filtering techniques. This proposal will focus on the emotional aspect of phishing social engineering, by attempting to incorporate it in predictive models.

STATE OF THE ART

Previous Studies

Phishing susceptibility

The main reference for this section is "*A meta-analysis of field experiments on phishing susceptibility*" [4].

Phishing susceptibility is regarded as "*the probability that a recipient performs an action requested in a fraudulent message, is a widespread problem*". Studying the psychological processes that lead a victim to succumb to a scam could be of great help to further augment existing email phishing prevention tools, however the literature about susceptibility appears to be scarce compared to the one about technical prevention.
In addition, no established model seems to exist to predict and understand phishing susceptibility.

Some studies seem to confirm existing general theories about deception and persuasion, where other evidences seem to contradict it.

Concluding this section, the results of the meta-analysis:

1. The characteristics of the person receiving the message (e.g.: personality, propensity to trust, etc.) seems to be of little significance.
2. The use of established deceptive tactics is confirmed to have an effect.
3. It seems that it is not important if the email comes from the right email server.
4. It was not explicitly tested if the content of the scam made a difference.
5. Circumstances are relevant, especially the timing: "*Data showed a lot of traffic due to the phishing emails just before closing hours and almost no traffic after closing hours*". However, time of day, week and year were not controlled for.
6. Adaptation matters.

Other findings were more concerned about the methodological issues of the previous studies. In the hypothetical research project, dedicating some time to further research on this aspect could be of great importance, however it won't be treated further here due to the scope of this assignment.

Emotionality

Despite being already investigated by the previously cited meta-analysis, the two papers used for this section were deemed particularly useful due to the investigation of the textual content and anatomy of scams, thus shifting the perspective of the analysis.

According to "*You can trust me: a multimethod analysis of the Nigerian email scam*" [3] the concept of trust seems to be very common in the classic Nigerian email phishing template (also commonly referred as "419 scam" or "advance fee scam").

Other important emotional manipulations that seem to be recurrent (according to "*The anatomy of written scam communications: An empirical analysis.*" [1]) are:

- Credibility, shown trough care, concern and prioritizing the needs of others.
- Showing vulnerability, to give the impression of honesty.
- Compliments (social solidarity) and flattery, to lower the recipients' guard.

Spam filtering

The main reference for this section is "*A Systematic Review on Spam Filtering Techniques based on Natural Language Processing Framework*".

With respect to the spam filtering techniques, the following can be said (keeping in mind the Spam vs Ham dataset [7] context of the analysis):

- Despite not impacting greatly in susceptibility, the email address is regarded as a significant factor for detection.
- The email subject consistency tries to appeal to urgency.
- Given that redirecting the user to visit a link could be an important goal of the scammer, checking for link presence is deemed useful.
- Concerning Natural Language Processing (NLP), the analysis suggests the following:
  - Spam classification is done by identifying a list of repetitive words previously identified in scams
  - Other techniques such as Sentiment analysis and Topic Modeling were mentioned only with respect to spam comments. In addition, Polarity (positive or negative sentiment) seems to be the only aspect considered for spam, in the context of Sentiment Analysis.

Research Question

Given the literature so far, is seems obvious that the emotional characteristic of text is an important part of scamming, however there is very little research on how emotionality of text contributes to the filtering process. Moreover, the studies claiming the importance of emotional manipulation (like [3]) have been conducted on a small corpus of spam texts. This context leads to the following research question: **what impact does text emotional connotation have in spam detection?** To answer the question, a big corpus of emails has been analyzed, including parameters such as sentiments, by an ensemble of statistical models. The whole process will be described in the following section.

METHODOLOGY

Description of the dataset

The making of

Given the nature of the problem, various data gathering techniques have been used.

- To build the non-scam portion of the dataset, a portion of the famous enron email corpus [10] has been used. Given that having non-spam email was essential, only emails from the folders "_sent_mail" and "personal" were parsed into the dataset.
- To build the first part of the scam part of the dataset, all emails present on each page of 419scam.org [9] (2004 → 2022) have been scraped.
- To build the final, and most conspicuous part of the scam dataset, emails from untroubled.org [8] have been used. Given the great amount of data present, only emails from 1998, 2020 and 2022 have been parsed into the dataset.

After parsing and cleaning, the dataset was formed by 359.552 emails (289.337 scam, 70.215 not-scam).

Data cleaning

Some cleaning operations have been performed:

- Emails with empty text, empty date or empty subject have been discarded.
- Non-UTF8 characters have been removed from the text.
- For sentiment analysis, email body texts have been converted to lowercase, stop-words have been removed, numbers and non-letters have been removed too. Finally, all words have been lemmatized.
- HTMLs tags have been removed.
- emails have been removed.
- common non-stop-words have been removed (enron, com, etc, …);
- single characters have been removed

Predictors

Given that text alone is not enough to build a model, predictors have been extracted from the fields *date*, *subject* and *text.*
Some however have been proven *problematic*:

- The **domain** of the email address is often treated with a whitelist/blacklist approach. Due to how the dataset was constructed, it is obvious that all emails coming from the enron domain are not-scam, so the domain could not be included as a predictor. To account for this, in the study one would have to find more non-scam emails. However, due to time constraints, in this proposal all emails have been assumed to have passed the domain check and needs to be identified after that.
- Email **urgency in the subject** is clearly an important aspect. Unfortunately, urgency detection in text is a hard problem of its own, so some surrogate predictors have been used.

Finally, the used predictors are:

- **Hour of the day** (approximated to the next after 45), *factor;*
- **Day of the week**, *factor;*
- **Day of the month**, *factor;*
- **Month**, *factor* ;
- **Percentage of capital letter in the subject text** *(integer from 0 to 100)*, in order to partially account from urgency.
- **Non-letter characters in subject** (spaces excluded), also to partially account for urgency. *integer.*
- **Length of the subject text**, *integer*.
- **Presence of links** in email text, factor.
- **Sentiment score in subject and text email**, integer. In particular:
  - anger;
  - anticipation;
  - disgust;
  - fear;
  - joy;
  - sadness;
  - surprise;
  - trust;
  - overall positiveness;
  - overall negativeness;

Figure 1: predictors of the dataset

Small exploratory analysis

As can be seen in figure 2 and 3, the most common words in scam emails are those regarding external resources, so one could expect the presence of links to be a very important predictor.

Figure 2: most common words on scam emails

Figure 3: most common words on non-scam emails

Another interesting thing to check preliminarily would be to look for a significant difference in sentiments. As can be seen in figure 4, the significative difference in mean between trust, positive, negative and overall sentiment in general would make one think of sentiment as a significant predictor.

```
    Welch Two Sample t-test

data:  df$Text_trust[df$y == "scam"] and df$Text_trust[df$y != "scam"]
t = 97.838, df = 260476, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 3.674837 3.825882
sample estimates:
mean of x mean of y
 7.928643  4.178684


    Welch Two Sample t-test

data:  df$Text_negative[df$y == "scam"] and df$Text_negative[df$y != "scam"]
t = 88.69, df = 243790, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 2.652339 2.772217
sample estimates:
mean of x mean of y
 5.764868  3.051798


    Welch Two Sample t-test

data:  df$Text_positive[df$y == "scam"] and df$Text_positive[df$y != "scam"]
t = 101.33, df = 250875, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 5.785710 6.013935
sample estimates:
mean of x mean of y
12.738035  6.838212


    Welch Two Sample t-test

data:  unlist(df[df$y == "scam", seq(from = 18, to = 37)]) and unlist(df[df$y != "scam", seq(from = 18, to = 37)])
t = 234.68, df = 4677966, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1.126860 1.145841
sample estimates:
mean of x mean of y
 2.406328  1.269978
```

Figure 4: Significancy t-test for average sentiments between scam and non-scam

Analytical Strategy

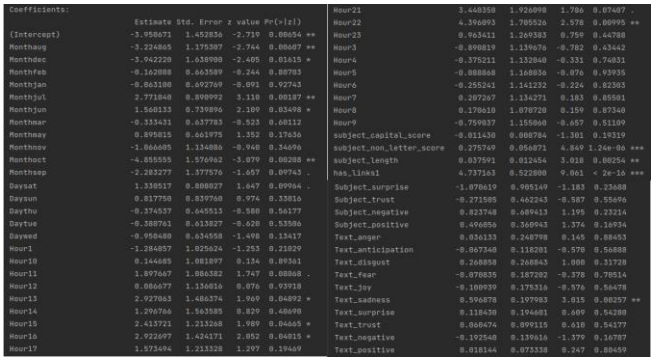
The analytical strategy consists in training the models with the identified predictors. To grasp whether sentiments have an effect, three methods can be used:

1. For the parametric approaches, the use of the p-value.
2. For the non-parametric approaches, the use of variable importance.
3. Testing a model trained in different ways, in particular:
   - without sentiment predictors (control condition).
   - with sentiment predictors
   - only with predictors deemed significant by the p-value. Logistic regression has been used as the base to get those predictors, also given that variable importance had comparable results.

Used models

1. **Logistic regression [18]**: due to the binary response, logistic regression was deemed as the most suitable linear model. This model identified as significant predictors (less than 0.01, see figure 2):
   1. some months
   2. the 22nd hour of the day
   3. non-letter scores
   4. the subject length
   5. if the email contains links
   6. text sadness
2. **Linear Discriminant Analysis** [15]
3. **Naive Bayes** [13]
4. **K-nearest Neighbors** [14]
5. **Tree methods**, in particular:
   1. Simple classification trees [19] *[see figure 6,7]*
   2. Pruned trees [19] [see figure 8,9].
   3. Bagged Trees [16].
   4. Random Forests [16] [see figure 10].
   5. Boosted Trees [12]
6. SVMs [13], in particular:
   1. Maximal Margin Classifier [see figure 11]
   2. Support Vector Classifier
   3. Support Vector Machine
      - polynomial kernel
      - sigmoid kernel
      - radial kernel

Figures



	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	-3.956671	1.452836	-2.719	0.00654	**
Monthaug	-3.224865	1.175387	-2.744	0.00607	**
Monthdec	-3.942220	1.638980	-2.405	0.01615	*
Monthfeb	-0.162088	0.663589	-0.244	0.80703	
Monthjan	-0.663180	0.692769	-0.893	0.92743	
Monthjul	2.771848	0.898992	3.118	0.00187	**
Monthjun	1.560133	0.739896	2.109	0.03498	*
Monthmar	-0.333431	0.637783	-0.523	0.60112	
Monthmay	0.895815	0.661975	1.352	0.17636	
Monthnov	-1.066605	1.134886	-0.940	0.34696	
Monthoct	-4.855555	1.570962	-3.079	0.00208	**
Monthsep	-2.283277	1.377576	-1.657	0.09743	.
Daysat	1.330517	0.808027	1.647	0.09964	.
Daysun	0.817750	0.839760	0.974	0.33016	
Daythu	-0.374537	0.645513	-0.580	0.56177	
Daytue	-0.380761	0.613027	-0.620	0.53506	
Daywed	-0.950488	0.634558	-1.498	0.13417	
Hour1	-1.284857	1.025024	-1.253	0.21029	
Hour10	0.144685	1.081897	0.134	0.89361	
Hour11	1.897667	1.086582	1.747	0.08068	.
Hour12	0.086677	1.136816	0.076	0.93918	
Hour13	2.927863	1.486374	1.969	0.04892	*
Hour14	1.296766	1.563585	0.829	0.40698	
Hour15	2.413721	1.213268	1.989	0.04665	*
Hour16	2.922697	1.424171	2.052	0.04015	*
Hour17	1.573494	1.213328	1.297	0.19469	
Hour21	3.448358	1.926098	1.786	0.07407	.
Hour22	4.396093	1.705526	2.578	0.00995	**
Hour23	0.963411	1.269383	0.759	0.44788	
Hour3	-0.890819	1.139676	-0.782	0.43442	
Hour4	-0.375211	1.132040	-0.331	0.74031	
Hour5	-0.088868	1.160036	-0.076	0.93935	
Hour6	-0.255241	1.141232	-0.224	0.82303	
Hour7	0.397267	1.134271	0.183	0.85501	
Hour8	0.178618	1.078728	0.159	0.87340	
Hour9	-0.759037	1.155060	-0.657	0.51109	
subject_capital_score	-0.011430	0.008784	-1.301	0.19319	
subject_non_letter_score	0.275749	0.056871	4.849	1.24e-06	***
subject_length	0.037591	0.012454	3.018	0.00254	**
has_links1	4.737163	0.522800	9.061	< 2e-16	***
Subject_surprise	-1.070619	0.905149	-1.183	0.23688	
Subject_trust	-0.271585	0.462243	-0.587	0.55696	
Subject_negative	0.823748	0.689413	1.195	0.23214	
Subject_positive	0.496856	0.360963	1.376	0.16934	
Text_anger	0.836133	0.248798	0.149	0.88453	
Text_anticipation	-0.867348	0.118201	-0.578	0.56888	
Text_disgust	0.268858	0.268843	1.000	0.31720	
Text_fear	-0.870835	0.187202	-0.378	0.70514	
Text_joy	-0.100939	0.175316	-0.576	0.56478	
Text_sadness	0.596878	0.197983	3.015	0.00257	**
Text_surprise	0.118430	0.194681	0.609	0.54288	
Text_trust	0.868474	0.099115	0.618	0.54177	
Text_negative	-0.192540	0.139616	-1.379	0.16787	
Text_positive	0.018144	0.075358	0.247	0.80459	

Figure 5: Logistic Regression p-values. Some non-significant have been cut due to space

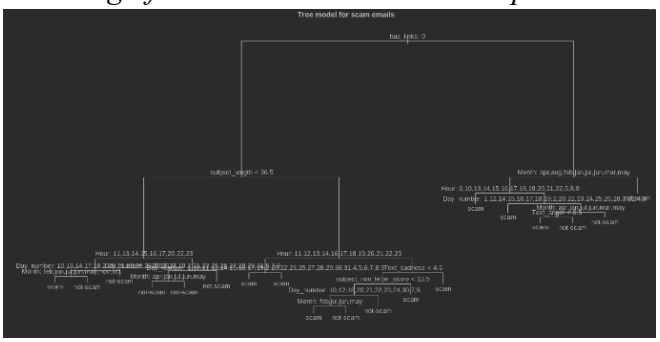
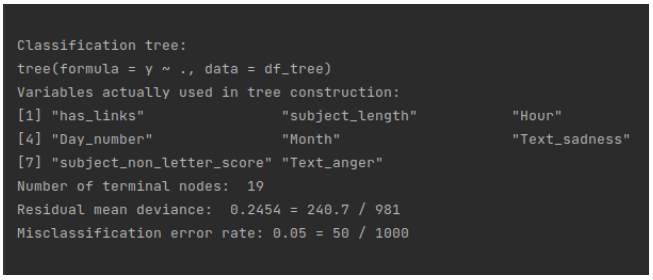


Figure 6: Simple classification tree



```
Classification tree:
tree(formula = y ~ ., data = df_tree)
Variables actually used in tree construction:
[1] "has_links"                "subject_length"           "Hour"
[4] "Day_number"               "Month"                    "Text_sadness"
[7] "subject_non_letter_score" "Text_anger"
Number of terminal nodes:  19
Residual mean deviance:  0.2454 = 240.7 / 981
Misclassification error rate: 0.05 = 50 / 1000
```

Figure 7: predictors simple trees actually used

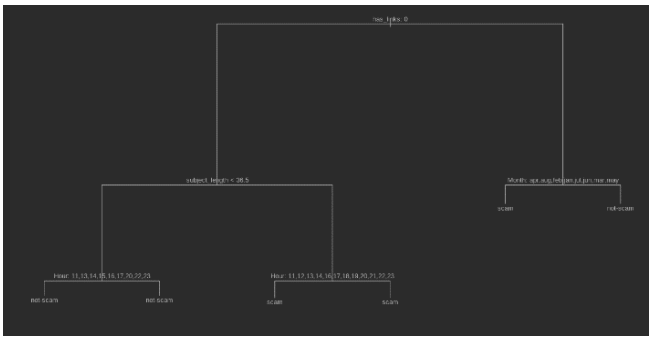


Figure 8: Pruned tree

```
Classification tree:
snip.tree(tree = model_tree, nodes = c(8L, 10L, 9L, 6L, 11L))
Variables actually used in tree construction:
[1] "has_links"      "subject_length" "Hour"           "Month"
Number of terminal nodes:  6
Residual mean deviance:  0.5256 = 522.5 / 994
Misclassification error rate: 0.104 = 104 / 1000
```

Figure 9: predictors pruned tree actually used

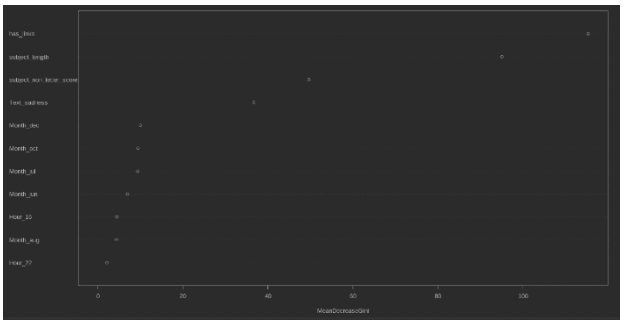


Figure 10: Random Forest variable importance

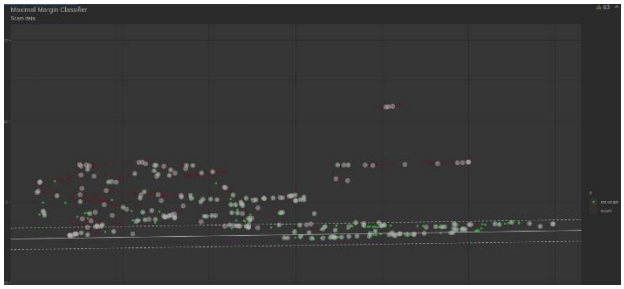


Figure 11: Maximal margin classifier (training)

Example of analysis

Due to computational constraints, the example analysis has been conducted on a small sample of the dataset: 1000 emails for tuning and training, 1000 emails for testing (with a disproportion of 70/30 in favor of scam). The analysis was divided in model selection, where hyper-parameters were chosen trough k-fold cross validation, and model assessment, where a test set was constructed by sampling new emails (never seen by the models) from the dataset. The process was done for each of the 3 settings (everything, no sentiment, only significant). Due to data uneven representation, the Area Under the Receiving operation curve (AUC) will be used as main metric reference, however other metrics *(Sensitivity, Specificity, Negative predicted value, Balanced Accuracy, Simple accuracy, Precision and F1 Score)* have also been computed for transparency.

As can be seen in tables 1 and 2, including sentiment did not significantly improve the predictive power of most models, with a difference of only ~1% of accuracy. Including only significant predictors also didn't improve the predictive power, in most cases in worsened it.

Table 1: All predictors (green=best, yellow=second, red=worst)

Method	AUC	Sensitivity	Specificity	Negative Predictive Value	Balanced Accuracy	Simple Accuracy	Precision	F1 score
Logistic Regression	95.0%	92.3%	85.4%	81.7%	88.8%	90.3%	94.0%	93.1%
KNN	91.9%	90.2%	78.6%	77.0%	84.4%	86.8%	91.0%	90.6%
LDA	95.5%	94.3%	82.1%	87.0%	88.2%	90.4%	91.9%	93.1%
Naive Bayes	83.3%	88.5%	42.4%	84.7%	65.5%	60.9%	50.7%	64.5%
Simple Tree	88.8%	92.9%	74.6%	84.3%	83.8%	86.7%	87.7%	90.2%
Pruned Tree	90.7%	89.9%	74.9%	76.7%	82.4%	85.3%	89.0%	89.4%
Bagging	95.2%	94.7%	76.8%	88.3%	85.7%	88.5%	88.6%	91.5%
Random Forest	97.9%	95.9%	84.7%	90.7%	90.3%	92.3%	93.0%	94.4%
Boosting	97.3%	86.0%	93.7%	94.0%	89.9%	91.4%	85.4%	85.7%
Maximal Margin Classifier	89.8%	72.0%	81.6%	10.3%	76.8%	72.4%	99.0%	83.4%
Support Vector Classifier	95.1%	92.4%	84.8%	82.0%	88.6%	90.2%	93.7%	93.0%
SVM radial	92.3%	83.5%	82.0%	56.3%	82.8%	83.2%	94.7%	88.8%
SVM Sigmoid	76.6%	73.1%	49.6%	22.3%	61.3%	69.9%	90.3%	80.8%
SVM Polynomial	94.9%	93.4%	84.7%	84.7%	89.0%	90.8%	93.4%	93.4%

Table 2: No sentiment (green=best, yellow=second, red=worst)

Method	AUC	Sensitivity	Specificity	Negative Predictive Value	Balanced Accuracy	Simple Accuracy	Precision	F1
Logistic Regression	94.7%	92.4%	84.8%	82%	88.6%	90.2%	93.7%	93%
KNN	92.7%	93.3%	81.2%	84.7%	87.2%	89.5%	91.6%	92.4%
LDA	95.1%	93.1%	82.1%	84%	87.6%	89.7%	92.1%	92.6%
Naive Bayes	94.7%	94.5%	80.7%	87.7%	87.6%	90%	91%	92.7%
Simple Tree	88.7%	90.6%	73.5%	78.7%	82%	85.1%	87.9%	89.2%
Pruned Tree	88.8%	89.9%	74.9%	76.7%	82.4%	85.3%	89%	89.4%
Bagging	94.2%	89.2%	84.6%	73.3%	86.9%	88%	94.3%	91.7%
Random Forest	96.9%	89.7%	91%	74%	90.3%	90%	96.9%	93.1%
Boosting	96.2%	86.7%	94%	94.3%	90.3%	91.8%	86.1%	86.4%
Maximal Margin Classifier	70.6%	59.6%	15.4%	21.3%	37.5%	41.2%	49.7%	54.2%
Support Vector Classifier	95.1%	92.5%	83.2%	82.3%	87.8%	89.7%	92.9%	92.7%
SVM radial	95.7%	92.2%	85.3%	81.3%	88.7%	90.2%	94%	93.1%

Method	AUC	Sensitivity	Specificity	Negative Predictive Value	Balanced Accuracy	Simple Accuracy	Precision	F1
SVM Sigmoid	86.8%	86.6%	78.1%	66.7%	82.3%	84.4%	92%	89.2%
SVM Polynomial	94.85%	93.4%	80.7%	85%	87.1%	89.4%	91.3%	92.3%

Table 3: Only significant (green=best, yellow=second, red=worst)

Method	AUC	Sensitivity	Specificity	Negative Predictive Value	Balanced Accuracy	Simple Accuracy	Precision	F1
Logistic Regression	94.5%	92.4%	81.5%	82.3%	87.0%	89.1%	92.0%	92.2%
KNN	93.0%	92.3%	82.6%	82.0%	87.4%	89.4%	92.6%	92.4%
LDA	94.5%	94.5%	77.4%	88.0%	86.0%	88.7%	89.0%	91.7%
Naive Bayes	92.4%	93.5%	73.1%	86.0%	83.3%	86.3%	86.4%	89.8%
Simple Tree	88.6%	92.7%	77.2%	83.7%	85.0%	87.7%	89.4%	91.1%
Pruned Tree	88.6%	92.7%	77.2%	83.7%	85.0%	87.7%	89.4%	91.1%
Bagging	93.9%	90.3%	79.7%	77.0%	85.0%	87.2%	91.6%	90.9%
Random Forest	95.9%	93.0%	80.3%	84.0%	86.6%	89.0%	91.1%	92.1%
Boosting	95.1%	81.3%	92.4%	92.0%	86.9%	89.1%	82.2%	81.7%
Maximal Margin Classifier	74.4%	70.0%	NA	NA	NA	70.0%	100.0%	82.4%
Support Vector Classifier	94.2%	93.0%	82.6%	83.7%	87.8%	89.8%	92.4%	92.7%
SVM radial	93.7%	93.1%	81.6%	84.0%	87.3%	89.5%	91.9%	92.5%
SVM Sigmoid	81.3%	83.0%	64.1%	59.0%	73.6%	77.8%	85.9%	84.4%
SVM Polynomial	94.3%	92.9%	81.8%	83.7%	87.3%	89.5%	92.0%	92.5%

DISCUSSION

As seen so far, despite being an important aspect in the literature, text sentiment seems to be of little importance in a predictive context. This proposal's approach however has some weak points that needs to be fixed in the full study:

- Given that emails metadata is also important, more variety should be included. Taking data only from the enron dataset is a limitation from this point of view (e.g., only one domain for non-scam).
- Another limitation has been given by the available computational power: scam emails were downsampled at the beginning to calculate sentiment in only few hours. Additionally, another downsample was done to tune models in a reasonable time. Finally, R [11] has been used as the main software to carry out computations, which can also be a limit because of the processor-centric logic (a package for GPU-based computations of SVMs has been tried, but R crashed without any error message).
- Temporal predictors that were not checked in previous studies seemed to be of little importance and cranks up computational times due to the high number of levels.

- A very big number of emails were not parsed due to the high variety in format. In the full study, more time should be dedicated to refining data parsing.
- As said in the methodology section, urgency detection is an aspect that was only partially accounted for.
- Different models and performance measures were used to avoid methodological biases. Tree based methods appears to be the most suited tool in the box for this problem, however there is still space for adding new models to test in the ensemble.
- P-value and variable importance were used to get the most significant predictors; however, the best-subset selection method could also be used with more computational power available.
- The library Syuzhet [17] was used to compute different kinds of sentiments. This is a double-edged sword: more sentiment variety to check is a clear advantage (e.g., text sadness was an important predictor), but makes dimensionality skyrocket, and so computational times. For reference, computing the sentiment of the dataset took ~ 2 hours.

In conclusion, sentiment analysis seems not to be so promising, but fixing the limitations could yield a different result.
All the code used for the assignment is available on Git Hub [20].

SOURCES

Papers and Meta-analysis

[1] Carter E. The anatomy of written scam communications: An empirical analysis. *Crime, Media, Culture*. 2015;11(2):89-103. doi:<10.1177/1741659015572310>

[2] P. Garg and N. Girdhar, "A Systematic Review on Spam Filtering Techniques based on Natural Language Processing Framework," 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2021, pp. 30-35, doi: 10.1109/Confluence51648.2021.9377042. <https://ieeexplore.ieee.org/document/9377042>

[3] Rich, T. You can trust me: a multimethod analysis of the Nigerian email scam. *Secur J* 31, 208–225 (2018). <https://doi.org/10.1057/s41284-017-0095-0>

[4] T. Sommestad and H. Karlzén, "A meta-analysis of field experiments on phishing susceptibility", 2019 APWG Symposium on Electronic Crime Research (eCrime), 2019, pp. 1-14, doi: 10.1109/eCrime47957.2019.9037502. <https://ieeexplore.ieee.org/document/9037502>

Articles

[5] Proofpoint phishing report https://www.proofpoint.com/us/blog/security-awareness-training/2022-state-phish-explores-increasingly-active-threat-landscape

[6] Signal phishing attack: https://support.signal.org/hc/en-us/articles/4850133017242-Twilio-Incident-What-Signal-Users-Need-to-Know-

Cited datasets and data sources

[7] Spam vs Ham dataset <https://www.kaggle.com/code/balakishan77/spam-or-ham-email-classification/data> last visited: 04/08/2022

[8] Spam dataset <http://untroubled.org/spam/>

[9] Scraped emails <https://www.419scam.org/emails/index.htm>

[10] Enron dataset <https://www.cs.cmu.edu/~enron/>

R

[11] R software

Libraries (NB: just the one cited in the report, not all libraries used for the project):

[12] caret https://cran.r-project.org/web/packages/caret/index.html

[13] e1071 https://cran.r-project.org/web/packages/e1071/index.html

[14] kknn https://cran.r-project.org/web/packages/kknn/index.html

[15] MASS
https://cran.r-project.org/web/packages/MASS/index.html
[16] randomForest
https://cran.r-project.org/web/packages/randomForest/index.html
[17] syuzhet
https://cran.r-project.org/web/packages/syuzhet/index.html
[18] tidymodels
https://cran.r-project.org/web/packages/tidymodels/index.html
[19] tree
https://cran.r-project.org/web/packages/tree/index.html

Code

[20] Git Hub repository
https://github.com/DMMP0/Computational-Social-Sciences-project