Project Report

IS450 - Text Mining and Language Processing
Section G1 - Team 6
April 2024

Name	Student ID
Chiang Kheng He	01428232
Emilia Lim Jia Wei	01444825
Tan Wei Shing	01358460
Goh Ming Cheng Jeremy	01396243
Joy Chua	01398505

Abstract
With the widespread adoption of internet connectivity and digital devices, scams involving attempts to deceive individuals into giving away money or personal information have proliferated. The impacts of these schemes on victims have shown to encompass social, psychological, emotional and economic harms. Consequently, there is a strong rationale to enhance our understanding of scams in order to devise ways in which they can be disrupted. Through the use of NLP techniques, we will be analysing emails for linguistic behaviours and use this empirical analysis to select an optimum collection of behavioural features of a user's email and classifier to detect fraudulent activities.

1. Introduction

In recent years, phishing scams have become a common form of social engineering attack to obtain sensitive data. Despite Singapore's establishment of the ScamShield, Anti-Scam Centre in 2019 and ongoing police operations against both domestic and transnational syndicates (SPF, n.d.), there has been an escalating number of reported scam cases in recent years, from 9,502 in 2019 to 46,196 in 2021 (CNA, 2022). As of 2024, at least 83 victims have lost $155,000 in the DBS phishing scam in January alone. (*The Straits Times*, 2024), showing the persistence and difficulty of identifying, reporting and protecting ourselves against the threat posed by scams.

This trend shows how phishing remains a prevalent scam technique preying on human weakness and habits to steal sensitive personal data. In a business context, companies facing a growing need to protect their employees from leaking confidential company information. In 2018, Singapore's largest healthcare group, SingHealth, experienced a massive data breach after its employees fell victim to a target phishing attack, while Singapore-based e-commerce platform Lazada faced scrutiny from the Personal Data Protection Commission (CNA, 2022) after a phishing attack compromised and exposed sensitive customer data, which included names, addresses, and payment information to the public website.

These security incidents come with significant economic and reputational costs, on top of the many legal and regulatory repercussions in terms of regulatory fines, penalties, and legal liabilities. The negative publicity and reputational fallout with customers, partners, and investors, can pose risks to the organisations' long-term business continuity and growth prospects. By understanding the tactics used by scammers and building a model to detect these tactics, companies can take more proactive risk management strategies to protect their sensitive data and mitigate the risks posed by cyber threats.

2. Related work

Scams within the context of online communication and the application of NLP have received considerable academic attention. Much of the work centres on the Interpersonal Deception Theory (IDT) (Buller, 1996) and the theory that deceivers and deception leave 'footprints' that can help us identify deception. There is strong evidence supporting the case that systematic differences in language use could help predict which messages originated from deceivers and which from those telling the truth. These characteristics of deception include the reduction of specificity (use of non-immediate language) and altered lexical diversity. The altered language usage has been attributed to the stress of performing an act of deception (Witte, 2007). Many studies support the IDT's characteristics of liars and deception using different techniques. Common methods include direct in-person studies (Burgoon,1996), probability distribution, clustering (Mbaziira, 2016) and semantic analysis (Zheng, 2003). Within Burgoon's research, the study found that individuals reported providing more detailed information and clearer responses when telling the truth compared to when deceiving. However, they felt less involved and more anxious when deceiving.

The application of NLP and machine learning algorithms on unstructured text is not new. The work on spam classification has included Naive Bayes (Meyer, 2004) and other statistical methods for spam filtering (Kolcz, 2004) as well as machine learning algorithms such as Support Vector Machines and Named Entity Recognition (NER) on free text in emails to classify and detect phishing scams (Lee, 2022). In Lee's study, NER was used to extract entities like names and locations to construct meaningful crowdfunding networks and profile scammers of fraudulent campaigns. More state-of-the-art NLP approaches have also been used within the crime domain. For example, Naudé (2022) experimented with transformer models such as Bidirectional Encoder Representations from Transformers (BERT) to detect fraudulent job advertisements.

Motivations and Hypothesis

Through this project, we hope to be able to build upon existing classification techniques for predicting whether a given text is a scam. Beyond framing scam classification as a standard text categorization task, we aim to integrate work from psychology and criminal understanding into computational linguistics analysis to analyse textual data through their tone and language usage to help develop more effective scam detection models in online deception.

Our research has direct implications for businesses, enabling them to better safeguard their customers and educate their employees about common scam indicators. By leveraging improved scam detection models, businesses can mitigate losses from email fraud and potential fines. Additionally, our work can provide more concrete examples of scam symptoms, empowering employees to recognize and respond effectively to fraudulent scams they encounter. Ultimately, businesses can better protect the integrity of their customers' data and maintain business continuity.

3. Dataset

The Enron Corporation was founded in 1985 as an American natural-gas pipeline company that grew to become one of the largest energy companies globally with revenues exceeding $100 billion in 2000 (Cohn, 2021). The Enron Fraud dataset contains over 450K records of emails generated by employees of the Enron Corporation, with 32 features (Rao, n.d.). These emails present a diverse set of email information ranging from internal, marketing emails to spam and fraud attempts. After some preliminary data cleaning to remove the many corrupted, inconsistent and wrongly scrapped records, the total records come up to 433,074 emails, of which 2219 are labelled as fraud (1) and 430,855 as non-fraud (0).

Based on preliminary EDA, of the 32 Features, we believe that 9 features would be the most useful (Date, Time, From, Subject, Body, Cc, Bcc, Unique-Mails-From-Sender, Label). Many of the fields appeared highly redundant and would add unnecessary noise to the data. Thus feature selection would be necessary to ensure a cleaner and more efficient dataset for classifier accuracy and prevent overfitting.

To improve the fraud classification on the Enron dataset as well as provide more fraud examples for modelling, additional “Fraudulent Email Corpus” data set will be used. It contains a collection of more than 2,500 "Nigerian" fraud emails, dating from 1998 to 2007. Of the 8 features from Enron, 3 labels (Cc, Bcc, and Uniques-Mails-From-Sender) are absent from this dataset. We will treat them as null values, mirroring real-life scenarios of entirely new emails. By adding more fraud cases to our corpus, we believe it will improve the model accuracy towards scams by removing model bias and reducing skew for a more comprehensive topic analysis.

Main dataset: https://www.kaggle.com/datasets/advaithsrao/enron-fraud-email-dataset
Secondary dataset: https://www.kaggle.com/datasets/rtatman/fraudulent-email-corpus/code (Radev, 2008)

4. Solution Overview

Our efforts will centre on understanding the modus operandi of liars and verifying if NLP techniques are able to pick up on more advanced deceptive schemes that attempt to appear genuine. For the purpose of this project, we will be focusing on:

1. Identifying common ‘genres’ of scams to illustrate the diverse tactics employed by scammers to exploit unsuspecting victims and distinct linguistic patterns, such as a lower usage of first-person pronouns of deceptive individuals compared to truth-tellers.
2. Perform scam classification prediction and evaluate the best classifier method to be employed for detecting scams.

To achieve that, we will first need to preprocessing the data to select the relevant features and convert them into a form suitable for our models, whilst overcoming the data challenges of a large dataset with high dimensionality, class imbalance, and presence of textual data whilst maintaining data interpretability.

Preprocessing & Feature Engineering	Data Cleaning - Tokenization & Lemmatization - Removal of stopwords, punctuation, HTML & URL tags - Decontraction Feature extraction (Topic modelling, POS frequency, Emotion Sentiment Analysis, etc) Feature selection (Mutual information and Correlation analysis) Sampling to handle class imbalance
Scam classification	Usage of pre-trained models - RoBERTa - Naive Bayes - Random Forest - XGBoost, Catboost, Adaboost, LightGBM (Baseline) Hyperparameter tuning Evaluation matrix LIME interpretation
Scam Topics: Illustrating the common tactics and 'genres' of scams	Unsupervised clustering:: - K-means - LDA topic modelling (done in preprocessing step) - Dendrogram - Principal Component Analysis (PCA) Finding feature weightage to topics using SVM LIME interpretation

5. Solution Details

5.1 Data Cleaning and Preprocessing

We performed several steps to clean and prepare text data for analysis or modelling, making it suitable / easier for further analysis. Note that for certain aspects of our project (for example, SpaCy's Part-of-Speech tagging and "special character count" feature does not require all preprocessing steps)
Remove of HTML/URL and special characters: Regular expressions were used to remove non-alphanumeric characters such as commas, fullstop, exclamation marks, etc as well as strip HTML tags, URLs or hyperlinks from the text, if present.
Tokenization: Tokenization is the process of splitting a text into smaller units, such as words or punctuation marks. In this step, we use the word_tokenize() function from NLTK to tokenize the text into words. Lowercasing: We convert all the tokens to lowercase to ensure uniformity and prevent the model from treating words with different cases as different entities.
Decontracted: Contracted words like "can't" were expanded to "cannot" using the contractions.fix() library. This helped us ensure we reduced words for lemmatization and stopped word removal.
Removing Stopwords: Stopwords are common words that often do not carry significant meaning, such as 'the', 'is', 'and', etc. We remove these stop words using NLTK's English stopwords corpus. Additionally, custom stopwords are found to be excessive and useless in context. (E.g. "RE:") were removed as well.
Lemmatization: NLTK's WordNetLemmatizer is used for reducing words to their base / root form for simplification of our vocabulary. For example, 'running' and 'ran' both get reduced to 'run'.

5.2 Feature Engineering

Further features were created to provide insights into the grammatical structure and linguistic patterns within the email content.

Date / Time Features: Standardised Datetime to Coordinated Universal Time (UTC) and split into discrete categorical variables (Year, Month, Week, etc) to capture non-linear relationships.
Subject and Body Features:

- Median number of characters per word, words per sentence, length of word, etc
- Ratios of upper-case character, punctuations, typos, special characters

Part of Speech Tagging: POS tagging was extracted from the Body content using NLTK pos_tag()
Lexical Complexity: Lexical complexity refers to the richness and diversity of vocabulary used in the text. We tokenized the text into sentences and calculated the median type-token ratio (TTR) for each sentence, based on the ratio of the number of unique words (types) to the total number of words (tokens) in a sentence, representing the lexical diversity. The median TTR across all sentences in the text and returns this value as the lexical complexity.
Sentiment: Sentiment analysis using NLTK VADER (Positive, Negative, Neutral and Compound Scores)
Emotional Intensity (Classification of Emotions): Using DistilBERT with core libraries from the Hugging Face Datasets, we obtained emotion sentiment analysis classification scores. While sentiment analysis datasets often entail binary classification tasks, our dataset has six different sentiment classes for sadness, joy, love, anger, fear, and surprise. Hence, we are required to approach the problem as a multiclass classification problem.
Data Normalisation: Numerical features were normalised with SK-Learn MinMaxScaler() to improve accuracy within models like SVM, as skew is unknown and potentially non-normal.

5.3 Feature Selection

Dimensionality Reduction: Dropping of unnecessary variables based on Mutual Information, contextual knowledge and correlation analysis (dropped those >= 90-95%). We plan to keep 90-95% of the principal components that would explain a significant portion of the total cumulative variance in the data. Which brings it around roughly 36-38 components of our feature engineering dataset (90-95% Cumulative Variance).

PCA feature selection has several benefits

- Reduce curse of dimensionality
- Reduce dimensionality of data
- Identifies the most important features that explain or capture the underlying structure of the data.

5.4 Scam topic identification

K-means - agglomerative clustering	Group the emails into clusters to identify the common characteristics the clusters share as well as their features.
Latent Dirichlet Allocation (LDA) topic modelling - Word-topic distribution bar chart - pyLDAvis visualisation - Word frequency bar chart - Word Cloud map	Identify hidden topics in the emails Shows word - topic distribution identified by the LDA model to help understand what each topic is about. pyLDAvis to visualise the topics and their probabilities
Feature weightage to predict topics - SVM weights - Random forest weightage	Identify the importance of each feature in making predictions. This is often calculated as the average reduction in the Gini impurity, or the total decrease in node impurity (weighted by the probability of reaching that node (which is approximated by the proportion of samples reaching that node) averaged over all trees of the ensemble.

Latent Dirichlet Allocation (LDA) is a topic model that uses statistical machine learning to identify latent topics within unstructured text documents. Given the assumption that each document contains a mix of words that make up various topics, this technique helps us identify the word distribution of these topics.

To help uncover underlying themes, we generated 12 topics and computed the most common words present in fraud and non-fraud emails. Commonly used words were computed based on an email's subject and body. These can be found in the appendix under "Body Most Frequent Words", "Subject Most Frequent Words" and "LDA Non-Fraud/ Fraud Topics". This allowed us to see the commonality and difference in patterns between fraud and non-fraud emails.

5.5 Scam classification

Based on the dataset's golden truth label 'Label' where 0 - safe, 1 - scam, classes were highly imbalanced. To disincentive models from prioritising the majority class, we attempted both upsampling and downsampling on all our models to ensure a more balanced dataset for classification. Additionally, we will prioritise evaluation metrics like AUC-ROC and F1 to provide a more accurate measure of model performance. Multiple classification models were tested for scam classification. For the sake of conciseness, we'll cover only the top 3 performing models (with the rest in appendix).

Naive Bayes model: Chosen for its simplicity, and effectiveness in text classification tasks, the model was trained using a subset of the Enron email dataset, with labels assigned to emails based on their likelihood of being associated with scam or non-scam activities. During training, features such as word frequencies and document frequencies were extracted to build the probability model. Various training cases were used to determine the parameters for the best performing model. These cases were to address skewness in our dataset where we had a high proportion of non-scam emails compared to scam email. This included using a Count vectorizer versus a TF-IDF vectorizer, using adjusted class weights, using grid search cross validation, undersampling and oversampling.

RoBERTa (Robustly Optimised BERT Approach) model: A variant of the BERT transformer model where the masking pattern is dynamically generated for each sequence every time it is presented in a batch. This results in each sequence being masked in different ways across different training epochs, allowing the model to learn from a more diverse set of masking patterns and reducing overfitting. PyTorch was utilised for faster computation, and stratified trained test split was used to get the median evaluation metrics for a more accurate scoring of the model's performance. The utils.EarlyStopping() acted as a form of regularisation to avoid overfitting by stopping the fold's training process early if the model's performance doesn't improve for a certain number of epochs. At base, 192 max character length, and batch size of 32 with 5 epochs of training was used as the baseline. Hyperopt was used for hyperparameter tuning and 50 epochs with a shorter max_seq_len of 160, alongside a lower learning rate and weight decay yielded better results.

Random Forest: Random Forest is an ensemble learning method used in machine learning. It operates like a group decision-making team, combining the opinions of many "trees" (individual models) to improve predictions and create a more robust and accurate overall model. Ensemble learning methods consist of a set of classifiers, such as decision trees, whose predictions are aggregated to identify the most popular result. Random Forest is notable for its versatility in effectively handling continuous variables in regression tasks, categorical variables in classification tasks and other object classes (non-numerical data). After performing chosen encoding methods such as target encoding for data features, it is able perform the classification task. This flexibility makes it particularly useful for a wide range of machine learning applications.

Random forest algorithms have four main hyperparameters, which need to be set before training. These include the number of trees in the random forest, minimum number of samples, minimum number of samples and maximum depth of each tree. Random forest classifiers are commonly used to solve both regression and classification problems, and they are known for their robustness and effectiveness. Parameters used are n_estimators 100 to 300 increments by 10, min_samples_split 2, 5, 10, min_samples_leaf 1, 2, 4 max_depth 10, 20. In all three tuning, the best parameters found by the randomised search are quite similar.

The accuracy scores across the three configurations are very close to each other, with only very slight variations. The accuracy ranges from approximately 0.667 to 0.672, which does not show significant improvement. The precision, recall, and F1-score metrics for both classes (0 and 1) are also relatively consistent performance across the three configurations. There are only minor fluctuations in these metrics, indicating that the model's ability to correctly classify instances of both classes remains relatively stable. The AUC-ROC scores are also consistent across the three hyperparameters tuning, ranging from approximately 0.667 to 0.672.

Overall, based on the minimal changes in performance metrics across the different hyperparameter tuning configurations, it can be inferred that the model may have reached a plateau in terms of performance improvement. It suggests that further tuning of hyperparameters may not lead to significant enhancements in model performance, and additional strategies may need to be explored to achieve better results.

After training, the models and vectorizers were exported using the joblib Python library and torch.save(), to save the progress and maintain a consistent evaluation during our analysis. LIME, a model -agnostic interpretation technique was then used to help us understand our models predictions. The visualied values provide an estimate of the features that contributed to the individual prediction, letting us evaluate the predictions of the models.

6. Results and Analysis

6.1 Topic Modelling

Within the proportions of topics for fraud and non-fraud emails, scam emails had one dominant topic, topic 3, while non-fraud emails had a more spread out distribution, with topic 4, 6 and 11 being more common. Topic 3 was closely related to phishing, with many fraud emails containing persuasive words such as “good”, “great”, “free” and action words such as “click” and “email”. On the other hand, non-fraud emails tended to be about business operations. For instance, some emails were daily or weekly email blasts about the changing energy landscape, which were pertinent information for an energy, commodities and services company like Enron.

Using Tableau to visualise the data, we found that fraud emails tended to have altered lexical and syntactical usage. Firstly, after normalisation, certain POS tags appeared more or less frequently in scams. Scams tended to have more Coordinating Conjunctions and Adjectives, with fewer Cardinal Numbers, Prepositions and Proper Nouns. This aligns with our research that lies tend to be less specific, with more generic nouns and adjectives to create urgency or enhance persuasiveness for the user to make an action. Secondly, fraud emails tend to be longer and more likely to include complex / longer words. Within their syntactic features, Body Median Chars per word and number of sentences was higher than non-scams. Thirdly, they tend to be isolated emails, with no past interactions or correspondences, with fewer BCC’s.

6.2 Scam Classification

Using the dataset’s golden truth label ‘Label’ where 0 - safe, 1 - scam, below are the results of our analyses using various models and sampling techniques:

Upsample:

	Naive Bayes	Roberta	Catboost	Random Forest
Accuracy	0.99	0.97	1.00	0.67
Precision	0.97	0.95	0.88	0.80
Recall	1.00	0.98	0.60	0.67
F1 Score	0.99	0.97	0.71	0.63
ROC-AUC	0.99	0.99	1.00	0.67

Downsample:

	Naive Bayes	Roberta	Catboost	Random Forest
Accuracy	0.93	0.96	0.98	0.67
Precision	0.88	0.94	0.97	0.80
Recall	0.98	0.99	1.00	0.67
F1 Score	0.93	0.96	0.98	0.63
ROC-AUC	0.98	0.99	0.98	0.67

Our upsampled Naive Bayes model performed the best among our evaluation metrics consistently. While Naive Bayes may perform better than other models in this given scenario, there are other factors to consider when evaluating their feasibility for scam email classification. For example, Naive Bayes models are known for their simplicity and efficiency, making them computationally lightweight and fast to train. In scenarios where speed is crucial, like real-time email classification, Naive Bayes may outperform more complex models like RoBERTa, which require significant computational resources and longer training times. However, this may lead to inaccuracies when used on more complex relationships in emails where genre classification and word embeddings are required.

6.3 Combining Topic Modelling and Scam Classification

We found that scams do exhibit distinctive lexical and syntactical patterns compared to legitimate emails. Our findings were consistent with our research indicating that fraudulent communications often employ less specific language, relying instead on generic nouns and adjectives to create a sense of urgency or persuade recipients to take action. We believe that the longer length and attempt at more complex syntax, may be an attempt to appear more legitimate to recipients by overwhelming recipients or obscurate the lack of detailed information, in an attempt to deceive recipients into believing the message is genuine and make it harder for recipients to discern the true nature of the scam.

The topic distributions generated can serve as additional features in our classification models such as the Naive Bayes classifier. By including topic-based features alongside the text features as mentioned above, we can further improve the classifier's ability to differentiate between fraud and non-fraud emails using the underlying themes and lexical patterns in the data. Thus, by combining these 2 analytical tasks will enable users such as companies to further increase the likelihood of identifying fraudulent emails.

7. Discussions and Gap Analysis

By leveraging a scammer's subconscious leakages, we can develop more sophisticated algorithms capable of discerning fraudulent content with greater accuracy and efficiency, as well as improve general awareness by helping them recognize common scam tactics to protect themselves from scams. We were able to tackle the data challenges of class imbalance with various sampling methods and handled the presence of text with feature extraction for models that cannot handle text. Furthermore, we employed various models that were beyond the class, such as RoBERTa as well as employed LIME to better understand the relationship between the model's prediction and the features used.

For K-Means Clustering, 5 clusters were created for emails of labels 1, 0, and both 1 and 0. With each cluster, the top features associated with each cluster were also crafted to provide insights into the characteristics of the emails within each group. For example, cluster 0 may have been characterised by features related to the content of the email subjects. However, upon analysis, each cluster had varying combinations of features as seen in the appendix under "K-Means Clustering and the resultant clusters and top features using PCA". For instance, cluster 0 containing emails of both labels had a mixture of subject character ratios and email metadata. One feature included was also the body dominant topic column. Due to the lack of similarity in characteristics, it was difficult for our team to conclusively identify characteristics that would help differentiate between fraud and non-fraud emails.

Though PCA was able to form definitive clusters for the emails, it was not useful for genre identification due to its inherent "black box" nature. Nevertheless, it helped inform us of the optimal 95% variance of our variables and justify keeping of our variables.

However, there were some assumptions and limitations that we weren't able to fully overcome.
Mono topic model: We only took the most dominant topic within the Subject and Body. Given the nature of correspondence, it is likely that we have ignored most of the high document-topic distributions and discarded valuable information within the large mixture of *other* topics for the sake of simplicity.
Risk of overfitting: Additionally, resampling itself skews the data. While preferable to Borderline SMOTE, which disregards correlation leading to information loss, it's not without drawbacks as it risks overfitting. Furthermore, the data, being from the early 2000s, may not be representative of current scam tactics.
Inaccuracy in scam genre "topic" classification: Despite attempting logistic regression, SVM and KNN, we were unable to achieve a good model for predicting the scam dominant topic using our existing linguistic features. Based on our topic exploration as well as classification report, we believe this was due to both the linguistic features being insufficient to predict genres as well as there being insufficient observations per topic that we lacked the time and resources to properly remedy.
Hardware limitations: Due to the large data sizes and insufficient hardware capabilities, the team was unable to train certain models like BERT and XLNet as previously planned. In our limited time, we could not resolve the errors in time. We learned to focus on what we are able to accomplish first and came up with backup models to maintain our project vision.

8. Future and Conclusion

We believe this linguistic approach to detect fraud has many applications: identification of deceptive online profiles, catching financial fraudulent statements and preventing deceptive emails in organisations. In order to do so, we'll need to:

Expand the dataset with more diverse sources of scam messages: This can involve different languages, regions, and communication channels (emails, SMS, social media, etc). A larger and more representative dataset would enable the model to learn from a broader range of linguistic features and improve its generalisation capabilities.
Incorporate Word Embeddings and Tenses: Due to computational limitations, word embedding techniques such as Word2Vec and tenses extraction was dropped in favour of pre-trained models. Embeddings help capture semantic relationships between words and given our POS analysis showed scams used fewer VBN (Verb Past Participles), tenses could weight specific words in our model training process. These can be integrated with existing traditional machine learning algorithms like SVMs or as

input to neural network architectures to enhance the model's ability to detect subtle linguistic patterns associated with scams.
Develop an automated data pipeline for real-time deployment: To properly incorporate these improvements into enterprise applications, we'll need to develop a scalable and efficient system for scam detection that can handle large volumes of text data in real-time so as to actively track and scale to meet the business' needs. This system would likely be deployed in a cloud environment for additional scalability and accessibility, as well as provide monitoring and logging capabilities for the business to track performance and detect anomalies.

9. Project Experiences/Reflections

Chiang Kheng He: I've had the opportunity to explore a diverse range of machine learning techniques and methodologies. My contributions spanned various stages of the project lifecycle, from conducting research on related works to the implementation of RoBERTa and logistic regression evaluations, and I've gained a better understanding and appreciation for how big a text analysis project can be. Additionally, having tried Doc2Vec, justified the implementation of LDA topics and troubleshooted the RoBERTa and SVM models, this project provided me with invaluable hands-on experience across a spectrum of machine learning techniques and data analysis tools.

Tan Wei Shing: I have developed a deep appreciation for Complexity of NLP NLU Methods, complexity of computing methods, it was computationally intensive but truly interesting and eye opening in applications. I am also amazed by kaggle cloud and google collab with its TPU and GPU capabilities which help us significantly reduce computing time. My biggest take away from this module is the methods used in data preprocessing, feature selection/importance, feature engineering, NLP & NLU methods, Machine Learning for model predictions, importance of literature review to understand processes of systematically breaking down high level computation steps to perform complex analysis such as emotion sentiment analysis. When we combine these computing methods into a project we can really appreciate the overview of text mining and I can see there are many applications across many fields of work. As very much of my learning experience was practice, applications, process of doing and from knowledge derived from practice track professors whose domain knowledge in the field are very insightful and truly inspiring.

Jeremy Goh: Through this project, I have learnt about the intricacies of text models and their practical applications. I have learned more about using text models like Naive Bayes and Roberta to evaluate how well our team can make scam detection due to various personal experiences and stories from friends and family. After completing this project, I have newfound respect for those who apply these models to address practical issues.

Emilia Lim: During the course of this project, I learnt the importance of leveraging techniques such as topic modelling alongside classification models for scam detection. With the help of my teammates, I was able to explore and gain a better understanding of different techniques, especially in SVM as well as topic modelling and clustering. I was also able to experience the difficulties of working with large datasets and learnt about various sampling methods as ways to workaround this obstacle.

Joy Chua: Through this project, I have gained more insights into how the techniques taught in class can be applied to real world scenarios. I was able to explore different models taught outside of class, and tried my hand at using XLNet. Even though it was not used in the final presentation, I still learnt a lot by conducting independent research on how to utilise new modelling techniques.

10. Appendix

Packages and tools Used

- Visual Studio Code and Google Colab (for CNN GPU / Cloud utilisation)
- Kaggle Cloud (Tesla P100 Data Center Accelerator)
- NLTK (for word tokenize, stopwords, pos tagging, etc)
- NLTK VADAR Sentiment Analysis Lexicon Scores
- NLTK Tokenization
- NLTK Lemmatization
- Textstat for Flesch-Kincaid readability score (flesch reading ease())
- VADER (Valence Aware Dictionary and sEntiment Reasoner) for sentiment analysis
- SpaCy for POS Tagging
- CoreNLPParser / SpaCy Part of Speech Tagging
    - (Complexity of Separation of POS Tags & Counting)
- SpaCy Entity Recognition (Long Computation Time)
    - (Complexity due extremely large data set)
- Syntactic Complexity of Sentence
- Lexical complexity of sentence
- Frequency of Discourse Markers
- SK-Learn Feature Engineering
- SK-Learn MinMaxScaler
- SK-Learn TfidfVectorizer
- SK-Learn PCA / Mutual Info FS
- Gensim Word2Vec
- LDA Modelling
- LDA Visualisation
- Parameter Tuning
- Tableau Data Visualization
- Tableau Dashboarding
- Various ML Model Training
- K-means / Dendrogram / RF / SVM
- NB / Roberta / XLnet
- SK-Learn Hyperparameter Tuning
- SK-Learn Undersampling / Oversampling
- SK-Learn Random Search / GridSearch

Tableau Visualisation:
https://public.tableau.com/app/profile/kheng.he.chiang/viz/TMEDA/DashboardSyntactical

Future Work Drive and Google Drive
🗀 **SMU TMLP**

Features dropped

"Folder-User", "Folder-Name", "X-Folder", "X-FileName", "Mime-Version", "Content-Type", 'Suspicious-Folders', "Content-Transfer-Encoding"	System and storage details that weren't deemed irrelevant in the context of receiving an email.
'Cc',, 'From', "Attendees", "Re", "Source", "Mail-ID", "POI-Present, 'BCC', 'CC', "To", "Message-ID", 'X-From', 'X-To', 'X-Origin','X-cc', 'X-bcc'	Network details that are deemed beyond the scope of this project, based on the assumption that a new spam email can come from anyone known or unknown.

'Low-Comm'	Correlation with 'Unique-Mails-From-Sender', where >= 5 as a binary value. Doesn't provide any additional information when 'Unique-Mails-From-Sender' is >= 5.

Features engineering

Original variable: Date **New variables:** 'year', 'month', 'day', 'hour', and 'day_of_week'	Date was a date time variable with different time zones (PDT and PST). Standardised to Coordinated Universal Time (UTC).
Original variable: 'Body', 'Subject' **New variables:** 'Body_lemma', 'Subject_lemma'	● Stringified inputs with str() ● Expanded contractions using 'contraction' Python library ● Lower cases using.lower() ● Replaced newlines with whitespaces ● Lemmatisation using WordNetLemmatizer() and NLTK .pos_tag() and wordnet ● Remove of all special characters and digits ● Removal of stopwords using word_tokenize() and NLTK stopwords corpus
Original variable: 'Body_lemma' **New variables:** 'sentiment_score_compound', 'sentiment_score_positive', 'sentiment_score_neutral', 'sentiment_score_negative'	VADER to calculate emotional intensity
Original variable: 'Body_lemma', 'Subject_lemma' **New variables:** '_dominant_topic',	Topic extraction on both body and subject using LdaModel() respectively Project simplest dimensionality by using only the main topic (determined based on largest topic weight) Optimal k number of topics determined based on brute forcing ranges from 2 to 20 and evaluating based on coherence CoherenceModel() and perplexity .log_perplexity()
Original variable: 'Subject', 'Body' **New variables:** '_num_words' '_num_sentences' '_median_chars_per_word' '_median_words_per_sentence'	Normalised with MinMax Scaler()

'_uppercase_ratio' '_punctuation_ratio' '_typo_ratio'	
Original variable: 'Body' **New variables:** 'Body_lexical_complexity'	Median lexical complexity of sentence. Calculated using type-token ratio (TTR) len(set(tokens)) / len(tokens)
Original variable: 'Body' **New variables:** 'VBZ', 'VBD', 'CD', 'PRP', 'VB', 'VBN', 'RB', 'TO', 'RP', 'VBG', 'NNP', 'VBP', 'JJ', 'NNS', 'IN', 'DT', 'CC', 'JJS', 'WP', 'NN', 'JJR'	POS frequency count

```
[ ] import nltk
    import string
    from nltk.corpus import stopwords
    from nltk.tokenize import word_tokenize
    from nltk.stem import WordNetLemmatizer

    # Download necessary NLTK resources
    nltk.download('stopwords')
    nltk.download('punkt')
    nltk.download('wordnet')

    # Initialize NLTK components
    stop_words = set(stopwords.words('english'))  # Set of English stopwords
    lemmatizer = WordNetLemmatizer()  # Word lemmatizer

    # Define punctuation marks using string module
    punctuation_marks = string.punctuation

    # Define a function to preprocess text
    def preprocess_text(text):
        if isinstance(text, str):  # Check if input is a string
            # Tokenize text into words and convert to lowercase
            tokens = word_tokenize(text.lower())
            # Remove punctuation marks and non-alphanumeric tokens
            tokens = [token for token in tokens if token.isalnum() and token not in punctuation_marks]
            # Lemmatize words and remove stopwords
            tokens = [lemmatizer.lemmatize(token) for token in tokens if token not in stop_words]
            # Join the preprocessed tokens back into a string
            return " ".join(tokens)
        else:
            return ""  # Return empty string for non-string inputs

    # Apply the preprocessing function to the 'Body' column of the DataFrame
    non_fraud_df['non_fraud_body'] = non_fraud_df['Body'].apply(preprocess_text)
```

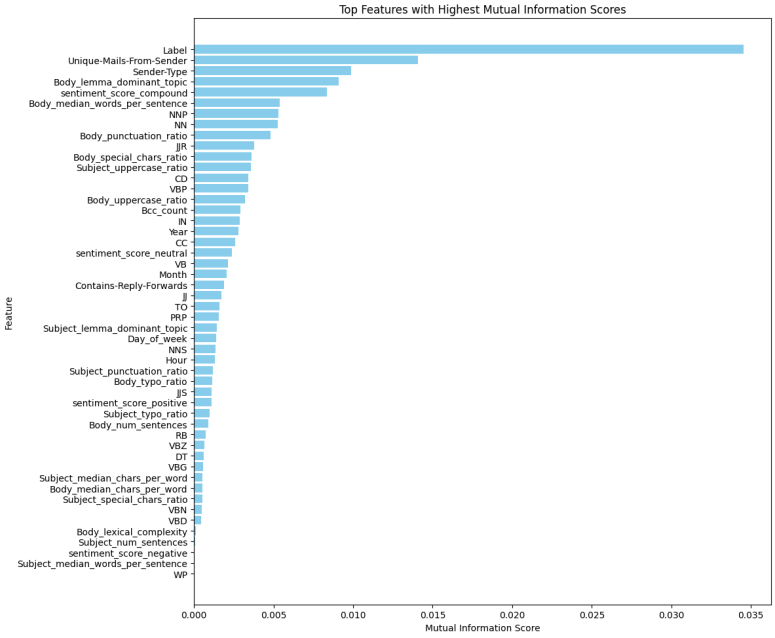
Body	Cc	Bcc	Suspicious-Folders	Sender-Type	Unique-Mails-From-Sender	Low-Comm	Contains-Reply-Forwards	Label	non_fraud_body
Status John: I'm not really sure what happened...			False	External	18	False	False	0	status john really sure happened impression vi...
re:summer inverses i suck-hope youve made more...			False	External	4	True	False	0	summer inverse youve made money natgas last 3 ...
The WTI Bullet swap contracts Hi, Following th...			False	External	3	True	False	0	wti bullet swap contract hi following received...
Fwd: NYTimes.com Article: Suspended Rabbi Quit...			False	External	9	False	True	0	fwd article suspended rabbi quits seminary pre...
daily charts and matrices as hot links 5/15 Th...			False	External	352	False	False	0	daily chart matrix hot link information contai...
...	...	...	...	...	...	...	...	...	...
Review Board Books w/Rebecca C./BillB/DaveG/Mi...			False	Internal	7	False	False	0	review board book monday september 24 2001 pm ...

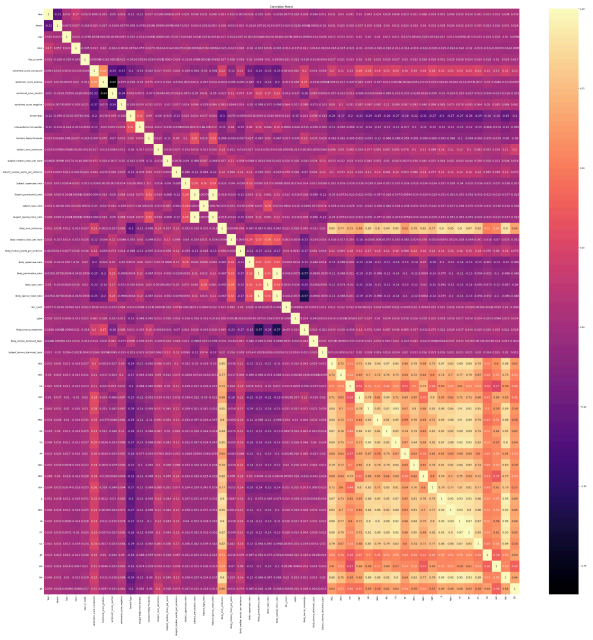
Data Cleaning & Preprocessing for LDA Modelling (Example)

```python
import nltk
import string
import re
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer

# Download necessary NLTK resources
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')
nltk.download('omw-1.4')

# Initialize NLTK components
stop_words = set(stopwords.words('english'))  # Set of English stopwords
lemmatizer = WordNetLemmatizer()  # Word lemmatizer

# Define punctuation marks using string module
punctuation_marks = string.punctuation

# Define a function to preprocess text
def preprocess_text(text):
    if isinstance(text, str):  # Check if input is a string
        # Remove numbers
        text = re.sub(r'\d+', '', text)
        # Remove HTML tags
        text = re.sub(r'<.*?>', '', text)
        # Remove URLs
        text = re.sub(r'http\S+|www\S+|https\S+', '', text)
        # Tokenize text into words and convert to lowercase
        tokens = word_tokenize(text.lower())
        # Remove punctuation marks and non-alphanumeric tokens
        tokens = [token for token in tokens if token.isalnum() and token not in punctuation_marks]
        # Lemmatize words and remove stopwords
        tokens = [lemmatizer.lemmatize(token) for token in tokens if token not in stop_words]
        # Join the preprocessed tokens back into a string
        return " ".join(tokens)
    else:
        return ""  # Return empty string for non-string inputs

# Apply the preprocessing function to the 'Body' column of the DataFrame
df3['cleaned_body'] = df3['Body'].apply(preprocess_text)
```

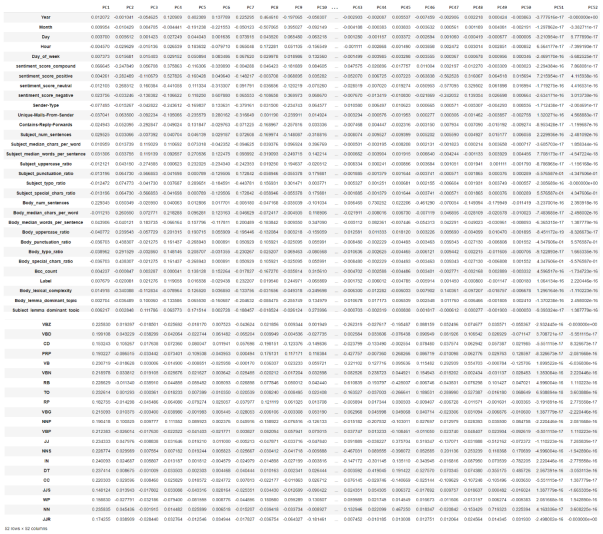
Sender-Type	...	JJ	NNS	IN	DT	CC	JJS	WP	NN	JJR	cleaned_body
0	...	8	1	12	8	2	1	1	13	0	status john really sure happen impression visi...
0	...	8	3	3	4	2	0	0	15	1	summer inverse suck hope make money natga last...
0	...	6	10	9	9	1	0	1	26	0	wti bullet swap contract hi follow e mail rece...
0	...	2	0	1	0	0	0	0	10	0	fwd article suspend rabbi quit seminary presid...
0	...	14	17	23	16	11	0	0	19	0	daily chart matrix hot link information contai...
0	...	13	7	16	14	8	0	0	37	0	resume john thank email offer route resume exp...
0	...	13	9	11	17	8	0	0	21	0	wrong address john think wrong email address b...
0	...	34	30	39	31	20	0	0	56	1	additional offering round portfolio john leave...
0	...	13	9	17	10	5	0	2	34	0	crazy mite scale little sort length sprd lengt...
0	...	3	0	4	3	1	0	0	9	0	little bird tell read time get free zdnet oneb...
0	...	16	6	40	16	8	0	1	33	0	big news sure hear fun news yet rob move bosto...
0	...	20	10	17	10	4	0	1	27	1	think rite curve flatten end overvalue part th...
0	...	26	12	12	14	7	1	0	53	0	bnp paribas commodity future ng marketwatch se...
0	...	14	16	23	17	10	0	0	19	0	daily chart matrix hot link information contai...
0	...	7	2	2	4	1	0	0	11	1	spread mkt get little bearish back winter thin...
0	...	0	0	0	3	1	0	0	8	0	ever get check send article owe click link jef...
0	...	3	1	6	3	1	0	0	8	0	service agreement john thank opportunity provi...
0	...	14	16	23	16	10	0	0	19	0	daily chart matrix hot link information contai...
0	...	8	3	12	12	2	0	0	22	0	password assistance greeting finish reset pass...
0	...	5	1	0	0	0	0	0	5	0	f g weather moderate enuf switch offset hdd ca...

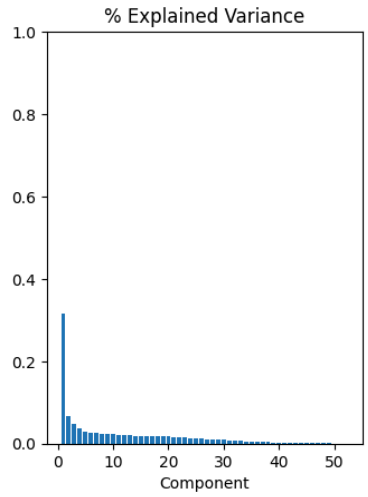
Data Cleaning & Preprocessing for Emotion Classification (Example)

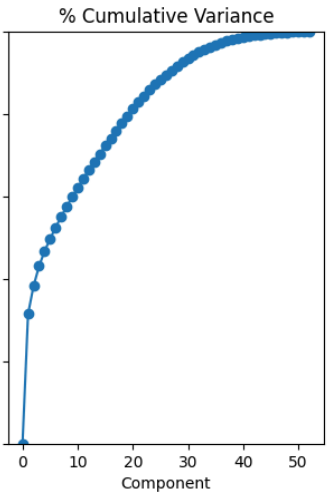




Mutual Information and Correlation analysis





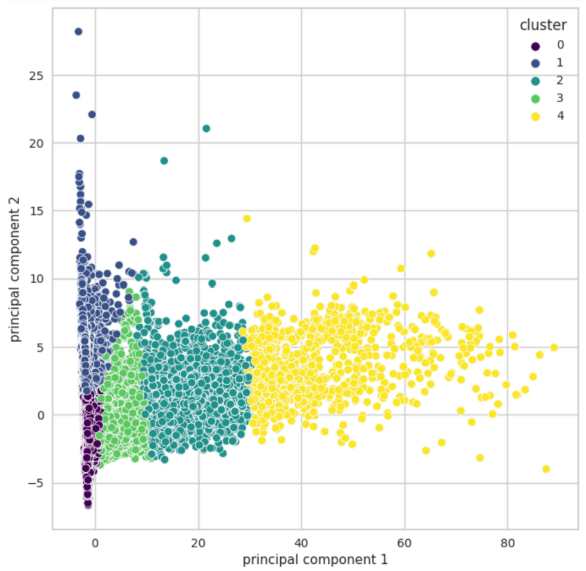


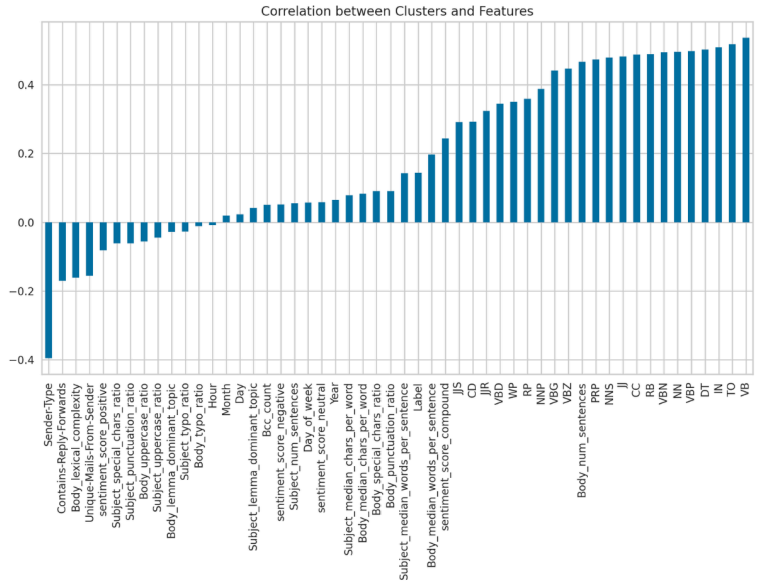
Principal componenet analysis (PCA) and K-means clustering

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from IPython.display import display

# Load dataset
df = pd.read_csv("/content/drive/MyDrive/SMU TMLP/enron_clean_V2.csv")

# Handle missing values
df.fillna(0, inplace=True)  # Replace missing values with 0 or NA

# Convert non-numeric values to numeric
df_numeric = df.select_dtypes(include=['float64', 'int64'])  # Select onl

# Standardize the data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(df_numeric)

# Create PCA object
pca = PCA()

# Fit your data
X_pca = pca.fit_transform(X_scaled)

# Create PCA dataframe
component_names = [f"PC{i+1}" for i in range(X_pca.shape[1])]
X_pca_df = pd.DataFrame(X_pca, columns=component_names)

X_pca_df
```

```
import matplotlib.pyplot as plt
import numpy as np

def plot_variance(pca, width=8, dpi=100):
    # Create figure
    fig, axs = plt.subplots(1, 2)
    n = pca.n_components_
    grid = np.arange(1, n + 1)
    # Explained variance
    evr = pca.explained_variance_ratio_
    axs[0].bar(grid, evr)
    axs[0].set(
        xlabel="Component", title="% Explained Variance", ylim=(0.0, 1.0)
    )
    # Cumulative Variance
    cv = np.cumsum(evr)
    axs[1].plot(np.r_[0, grid], np.r_[0, cv], "o-")
    axs[1].set(
        xlabel="Component", title="% Cumulative Variance", ylim=(0.0, 1.0)
    )
    # Set up figure
    fig.set(figwidth=width, dpi=dpi)
    return axs

# Assuming you have already performed PCA and stored it in the variable pca
axs = plot_variance(pca)
plt.show()
```

```
# Loadings
loadings = pd.DataFrame(
    pca.components_.T,  # transpose the matrix of loadings
    columns=component_names,  # Columns principal components
    index=df_numeric.columns,  # Rows original features
)

# Display loadings
loadings
```



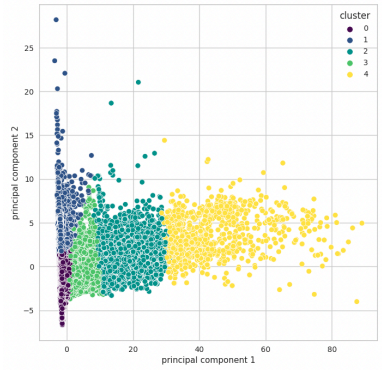
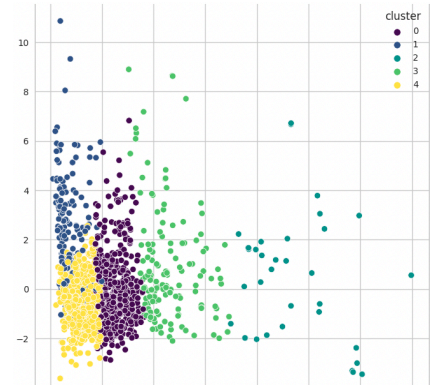
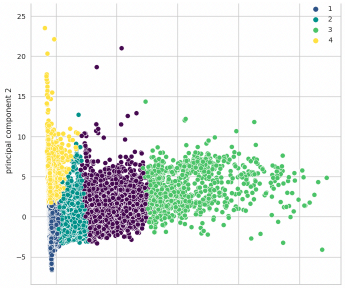


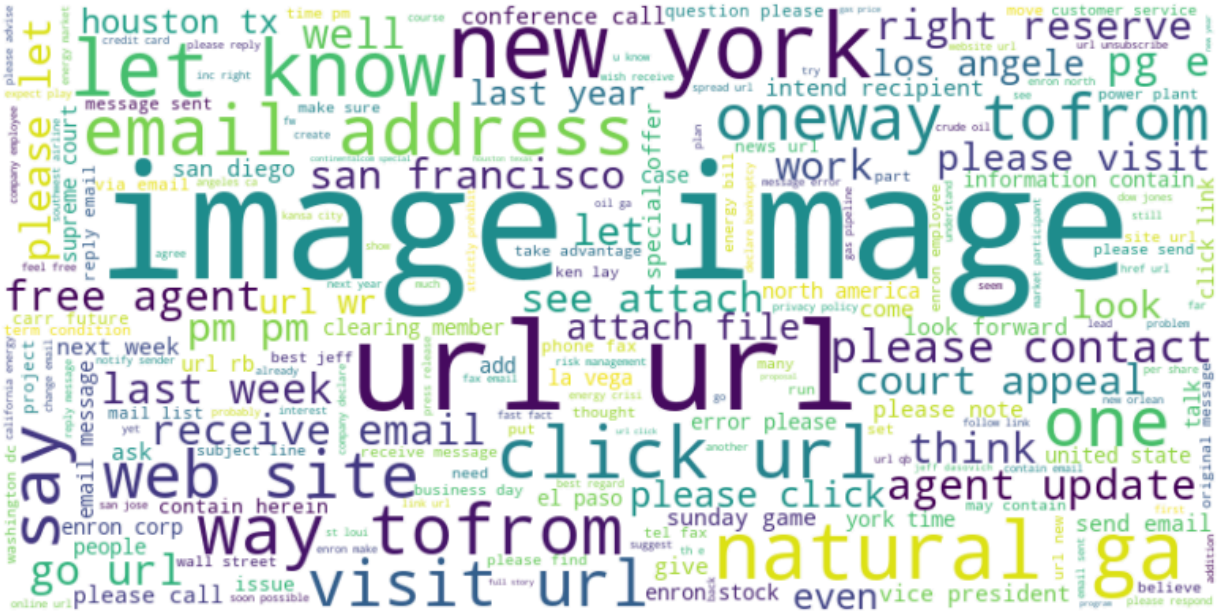
Principal Component Analysis for feature variance and clustering

K-Means Clustering and the resultant clusters and top features using PCA

Cluster No.	Label = 1 & 0's Top Features	Label = 1's Top Features	Label = 0's Top Features

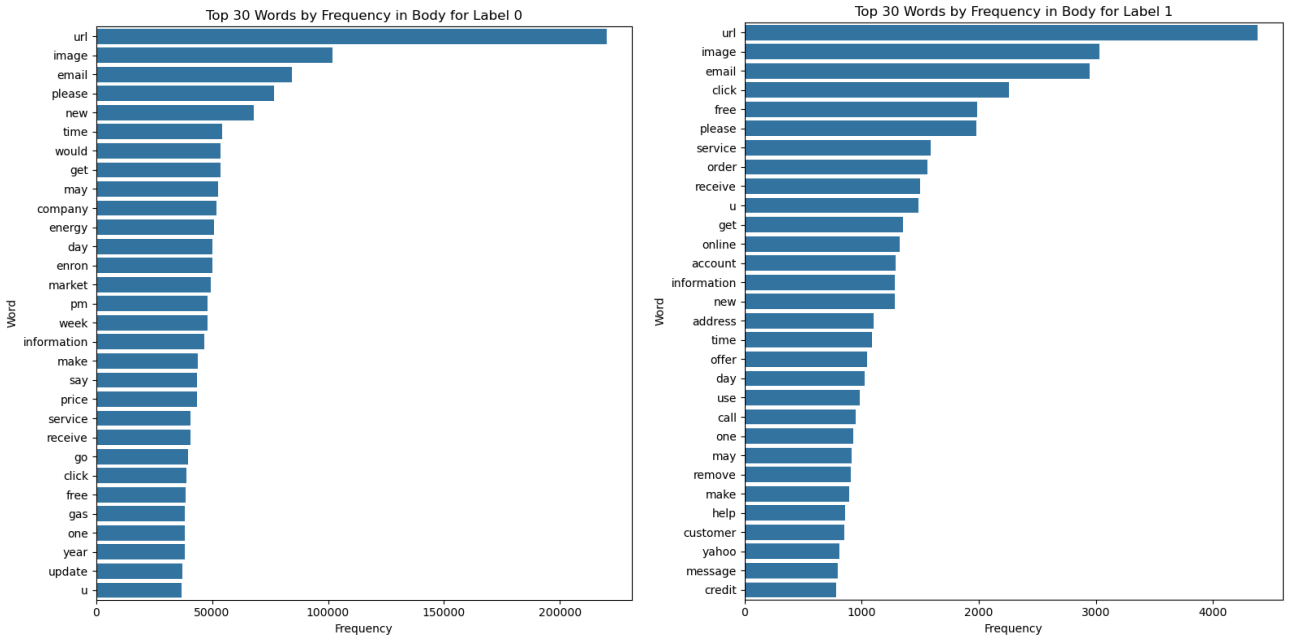
0	Subject_special_chars_ratio Subject_punctuation_ratio Subject_uppercase_ratio Hour Unique-Mails-From-Sender Body_lemma_dominant_topic Contains-Reply-Forwards sentiment_score_positive Sender-Type Body_lexical_complexity	Subject_median_words_per_sentence JJR IN PRP RB NN TO sentiment_score_compound Body_num_sentences VB	NNP Body_num_sentences CC TO DT VB NNS JJ IN NN
1	Subject_special_chars_ratio Subject_punctuation_ratio Subject_typo_ratio Unique-Mails-From-Sender sentiment_score_neutral Body_uppercase_ratio Body_median_chars_per_word Body_typo_ratio Body_punctuation_ratio Body_special_chars_ratio	Hour Body_uppercase_ratio sentiment_score_neutral Body_median_chars_per_word sentiment_score_negative Subject_special_chars_ratio Subject_punctuation_ratio Body_punctuation_ratio Body_special_chars_ratio Body_typo_ratio	Subject_special_chars_ratio Subject_punctuation_ratio Subject_uppercase_ratio Hour Unique-Mails-From-Sender Body_lemma_dominant_topic Contains-Reply-Forwards sentiment_score_positive Sender-Type Body_lexical_complexity
2	NNP Body_num_sentences CC TO DT VB NNS JJ IN NN	Body_num_sentences VBZ NN RB VBG VBP CC IN NNS DT	IN CC DT VBN RB TO sentiment_score_compound VBP VB PRP
3	IN CC DT VBN RB TO VBP sentiment_score_compound VB PRP	RB NNS JJ DT TO VB CC Body_num_sentences NN IN	RB TO NNS VBZ DT Body_num_sentences NN JJ VBD IN
4	RB TO NNS	Subject_median_chars_per_word Hour Body_median_words_per_sentence	Subject_special_chars_ratio Subject_punctuation_ratio Subject_typo_ratio Unique-Mails-From-Sender

	VBZ DT Body_num_sentences NN JJ IN VBD	Unique-Mails-From-Sender Body_lexical_complexity Subject_uppercase_ratio Body_uppercase_ratio Month sentiment_score_compound sentiment_score_positive	sentiment_score_neutral Body_uppercase_ratio Body_median_chars_per_word Body_typo_ratio Body_punctuation_ratio Body_special_chars_ratio
			

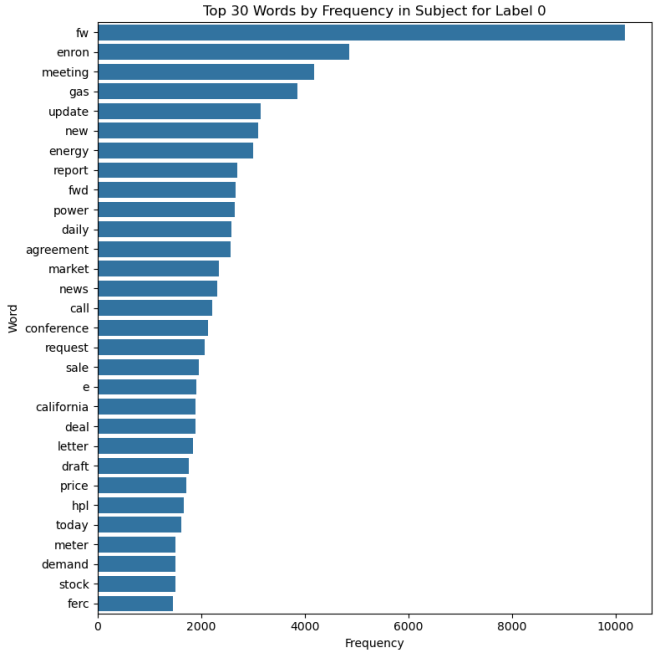


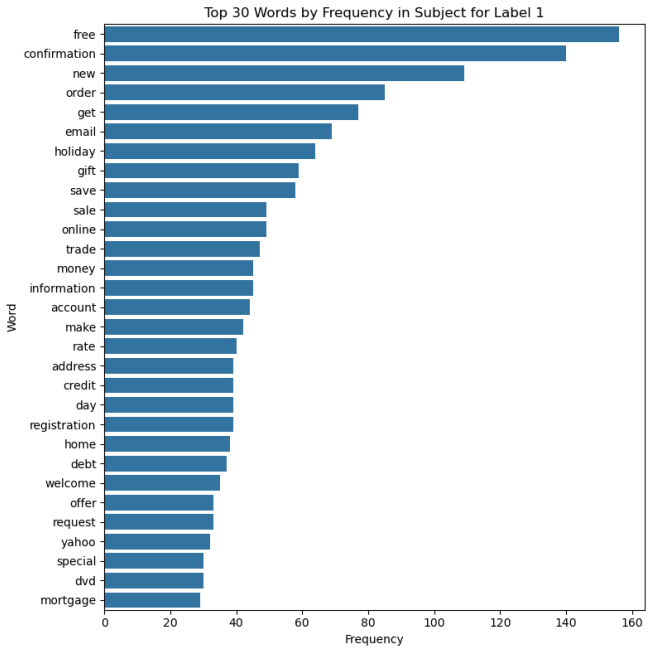
Body Word Map

Subject Word Map

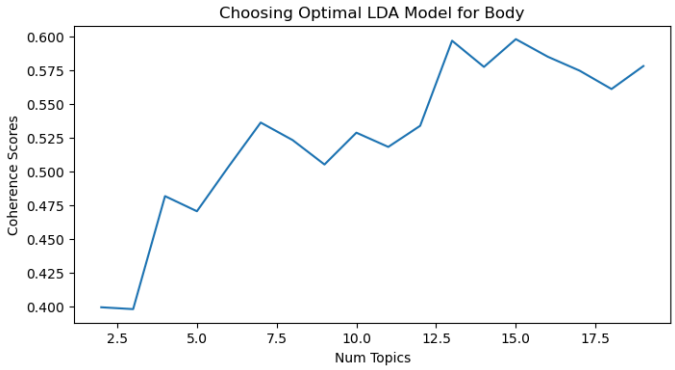


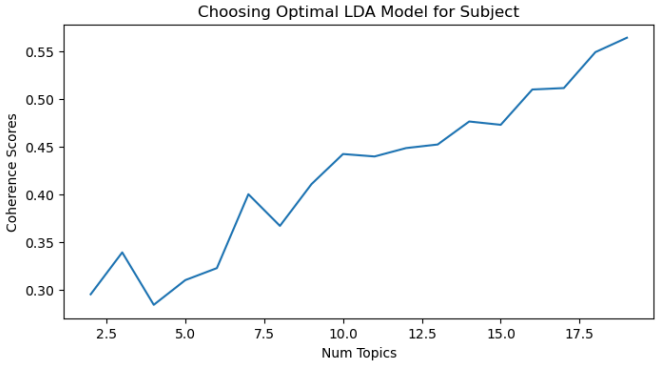
Body Most Frequent Words





Subject Most Frequent Words





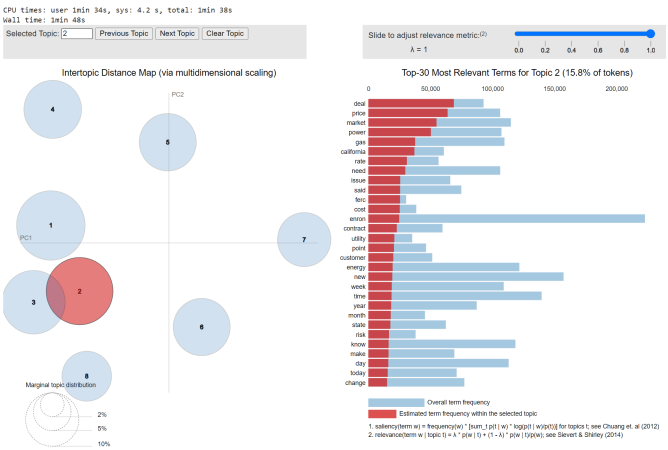
Coherence Score elbow curve for justification using 12 topics in LDA

```
Topics found via LDA non fraud:

Topic #0:
date schedule final hour gas 2001 report file day 2000 data variance transaction start trade message

Topic #1:
image url week game updated travel fare free day hotel special way new wr tx city

Topic #2:
pm forwarded enron meeting thanks error database vince jeff john mark office fyi know fw mike

Topic #3:
enron agreement attached thanks fax need know copy message phone legal information draft question let change

Topic #4:
url company new click enron image stock service free email information business million online employee year

Topic #5:
energy power company new service state url gas market business information said industry request project report

Topic #6:
deal price market power gas california rate need issue said ferc cost enron contract utility point

Topic #7:
know time like good going day think let work want week email make thing need great
```

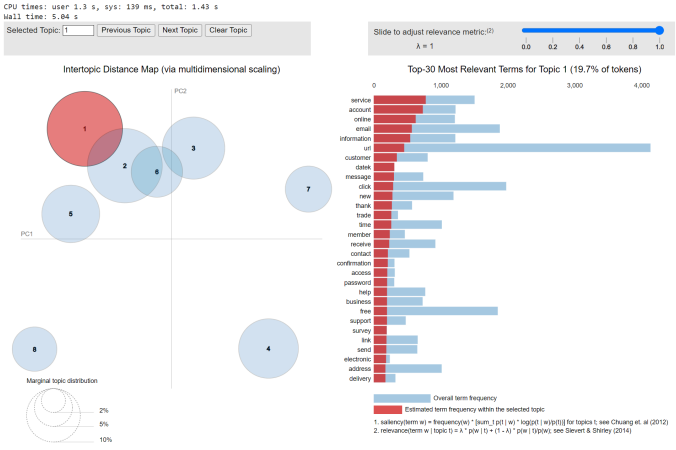
LDA Non-Fraud Topics

```
Topics found via LDA fraud:

Topic #0:
service account online email information url customer datek message click new thank trade time member receive

Topic #1:
normal gift color email url holiday arial order address image 000000 bold phone day 9pt offer

Topic #2:
yahoo order email url item account address new shipping free day click price service message information

Topic #3:
url credit order card business email click receive free new company internet information time home want

Topic #4:
dvd free click movie copy company cost wizard price game url insurance software offer life available

Topic #5:
free url click investment bank fund new money like investor time know customer investing offer business

Topic #6:
image url offer cdnow special new price free receive click sale music day email cd dvd

Topic #7:
url click free address help link web debt site support home need service payment online time
```

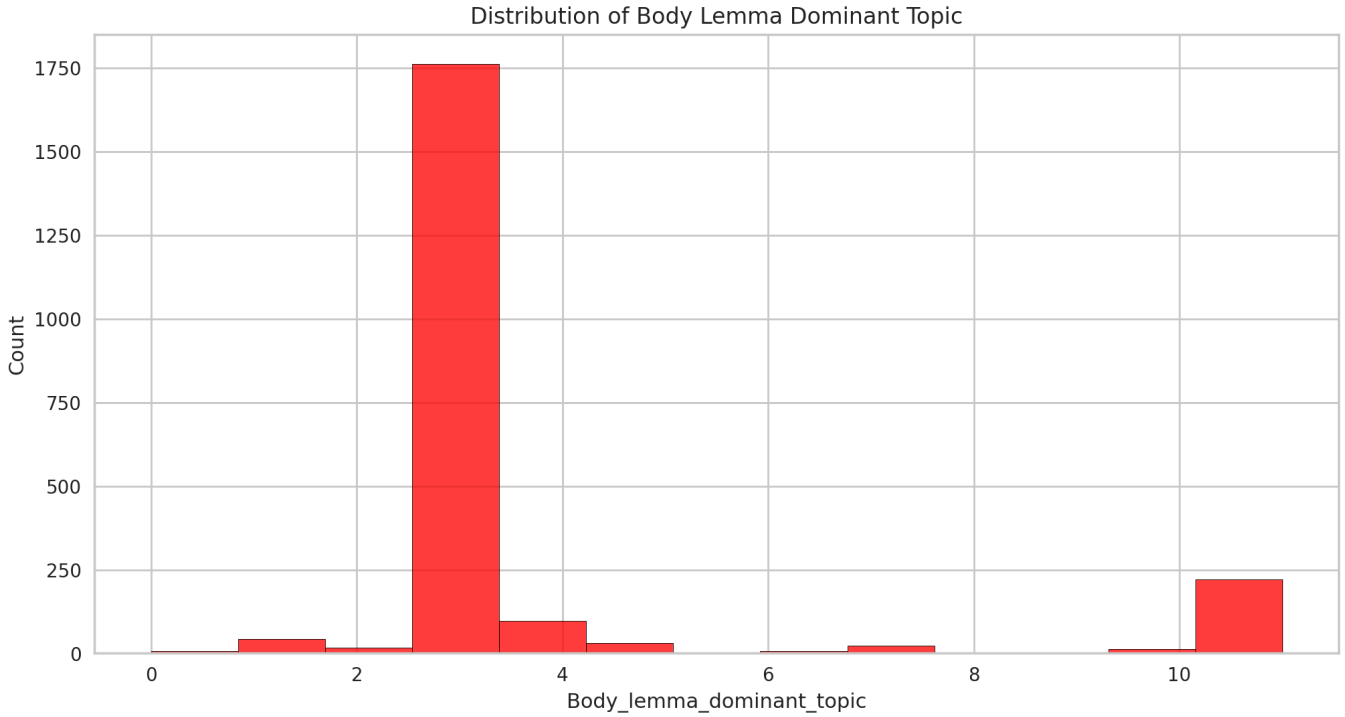
LDA Fraud Topics



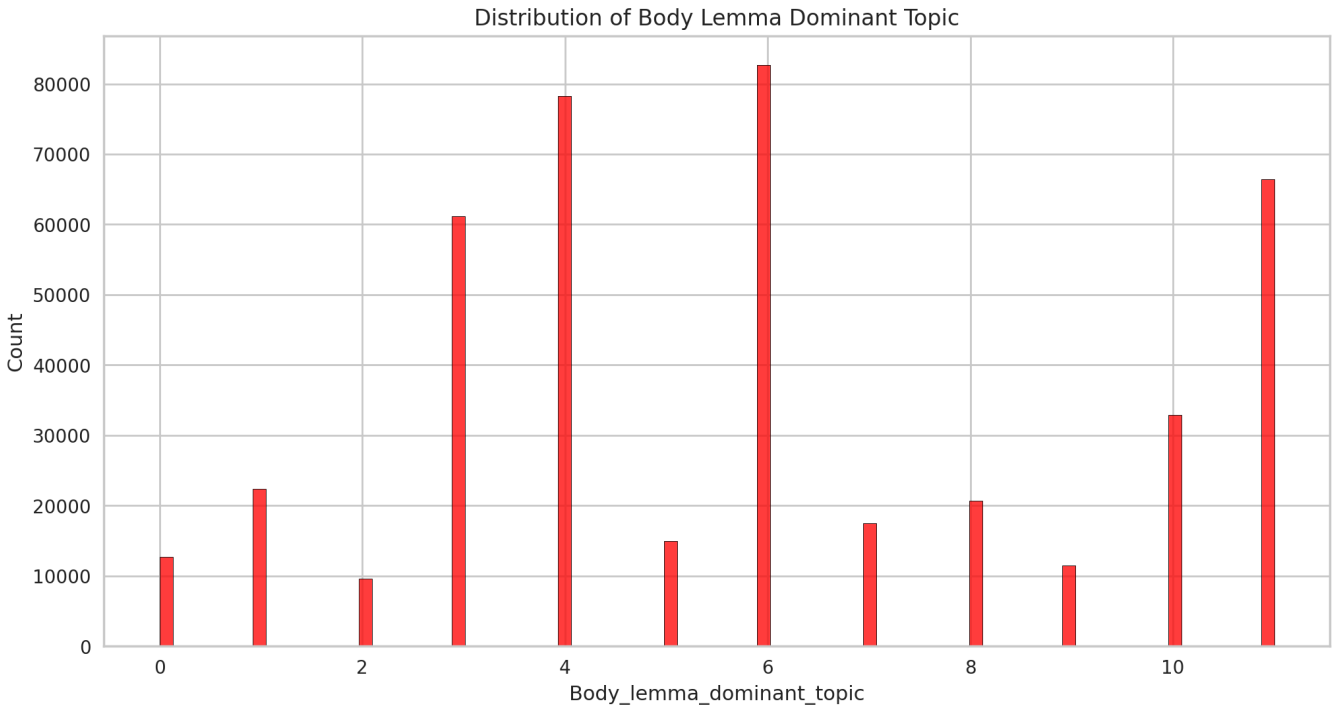
Non-Fraud LDA Topic Modelling (Probabilistic)



Fraud LDA Topic Modelling (Probabilistic)



LDA Topic distribution for scams (Label = 1)



LDA Topic distribution for non-scams (Label = 0)

Topic #3 (most frequent for scams)	**Label 0 example:** **Subject:** Big news! **Body:** Big news! I'm not sure if you have heard our fun news yet, but Rob and I are moving to Boston at the first of the year!! I will be working remotely for HBK while looking for something new up there. Rob is continuing his career with Fidelity, though in a much more convenient location. The only downside is he will actually have to go into the office every day! We would love to see everybody before we leave, though it may not be possible with the holidays. I'll be back in Dallas once a week in January and February, so we could plan to get together during one of those trips. But before that time, Rob and I will be with a group of friends from HBK at Martini Ranch on Thursday, December 14th right after work. I'm sure we'll be there late, given my love of martinis, so please stop by if you can. Regardless, I really want to stay in	**Label 1 example:** **Subject:** Amazon.com Password Assistance **Body:** Amazon.com Password Assistance Greetings from Amazon.com. To finish resetting your password jarnold@enron.com, please visit our site using one \ of the personalized links below. The following link can be used to visit the site using the secure server: <URL> The following link can be used to visit the site using the standard server: <URL> It's easy. Simply click on one of the links above to return to our Web site. If this doesn't work, you may copy and paste the link into your browser's address window, or retype it there. Once you have returned to our Web site, you will be given instructions for resetting your password. If you have any difficulty resetting your password, please feel free to contact us by responding to this e-mail. Thank you for visiting Amazon.com! Amazon.com Earth's Biggest Selection <URL>

	touch with my Texas friends, so make sure and send me your updated contact information if anything has changed in the last year. Please pass this on to people who I may have inadvertently missed or I did not have their current email address. Cheers, Heather Robertson P.S. By the way, our house is on the market right now, so if you know of anyone looking, please tell them about it. The address is 6815 Vivian Avenue, Dallas, 75223 and can be seen on <URL>	
Topic #4 (2nd most frequent combined)	**Label = 0 example:** **Subject:** The WTI Bullet swap contracts **Body:** The WTI Bullet swap contracts Hi, Following the e-mail you have received yesterday concerning the new WTI bullet swap contracts, we would like to summarize what we have done on the ICE system yesterday evening: -Deleted WTI monthly time spreads -Deleted WTI/Brent monthly diff spreads (spread with legging) -Deleted 1% NYH Harbor Fuel Oil Crack monthly (spread with legging) -Added WTI/Brent monthly diff spreads (spread with NO legging) -Added 1% NYH Harbor Fuel Oil Crack monthly (spread with NO legging) Unfortunately the WTI/Brent and the 1% NYH Fuel Oil Crack contracts (with the legging functionality) have been removed from your portfolios. You will need to add the first four nearby months' contracts to your portfolios by going to Admin Manage Portfolios Edit your portfolio.... Please do not hesitate to contact us is you have any question: Helpdesk on +1 770 738 2101 (US) Stephanie Trabia: +44 207 484 5546 (UK) Regards, Stephanie Trabia Marketing Manager IntercontinentalExchange Tel +44 207 484 5546 Fax +44 207 484 5100 Mob +44 77 33 261 268 stephanie.trabia@intcx.com	**Label = 1 example:** **Subject:** bank wire wsex **Body:** bank wire wsex Dear Mr.. Lavorato, The $1500 that you sent to us in October, has not been credited to our account.If those funds were sent through AM TRADE INTERNATIONAL, you need to have your bank send an amendment message stating that the respective funds are intended for final credit to World Sports Exchange/ ACCT 12307915. Most likely, those funds are sitting at the Antigua Overseas Bank. 2. Another suggestion would be to call back those funds since the beneficiary is not in receipt of payment, and resend it through the new instruction as per our website.This method is guaranteed more efficient. For any further questions, please call the accounts department at WSEX. REGARDS MARIA. Accounts Manager _____ _____ Get Your Private, Free E-mail from MSN Hotmail at <URL> Share information about yourself, create your own public profile at <URL>

Topic #6 (1st most frequent combined)	**Label = 0 example:** **Subject:** bnp paribas commodity future aga survey result **Body:** Forwarded by Daryl DWORKIN on 10/24/2001 12:43:01 PM 10/24/2001 11:55 AM Daryl DWORKIN BNP PARIBAS 10/24/2001 11:55 AM	**Label = 1 example:** **Subject:** your Legal NetEx password **Body:** your Legal NetEx password Hi Kay, Ed Hearn has recently signed you up to Legal NetEx, the secure extranet. To log into Legal NetEx and view Ed's documents, go to web.legalnetex.com and click "log on". Enter your user name which is kay mann (all lower case). Enter your password which is whitepaper44. If you have any questions or problems using Legal NetEx, please contact me. Matt Troncelliti InterNetEx Customer Support 610-617-1753 matt@internetex.com
Topic 11	**Label = 0** **Subject:** Status **Body:** Status John: I'm not really sure what happened between us.? I was under the impression after my visit to Houston that we were about to enter into a trial agreement for my advisory work.? Somehow,?this never occurred.? Did I say or do something wrong to screw this up??? I don't know if you've blown this whole thing off, but I still hope you are interested in trying?to create an arrangement.? As a courtesy, here is my report from this past weekend.? If you are no longer interested in my work, please tell me so.??Best wishes, Mark Sagel Psytech Analytics (410)308-0245? energy2000-1112.doc	**Label = 1** **Subject:** Do you owe the IRS money? [2a994] **Body:** Do you owe the IRS money? [2a994] Have tax problems? Do you owe the IRS money? If your debt is $10,000 US or more, we can help! Our licensed agents can help you with both past and present tax debt. We have direct contacts with the IRS, so once your application is processed we can help you immediately without further delay. Also, as our client we can offer you other services and help with other problems. Our nationally recognized tax attorneys, paralegals, legal assistants and licensed enrolled agents can help you with: Tax Preparation Audits Seizures Bank Levies Asset Protection Audit Reconsideration Trust Fund Penalty Defense Penalty Appeals Penalty Abatement Wage Garnishments and more! To receive FREE information on tax help, please fill out the form below and return it to us. There are no obligations, and supplied information is kept strictly confidential. Please note that this offer only applies to US citizens. Application processing may take up to 10 business days. Note: For debt size please also include any penalties or interest Full Name: State: Phone Number: Time to Call: Estimated Tax Debt Size: Thank you for your time. Note: If you wish to receive no further advertisements regarding this matter or any other, please reply to this e-mail with the word REMOVE in the subject. trytb910

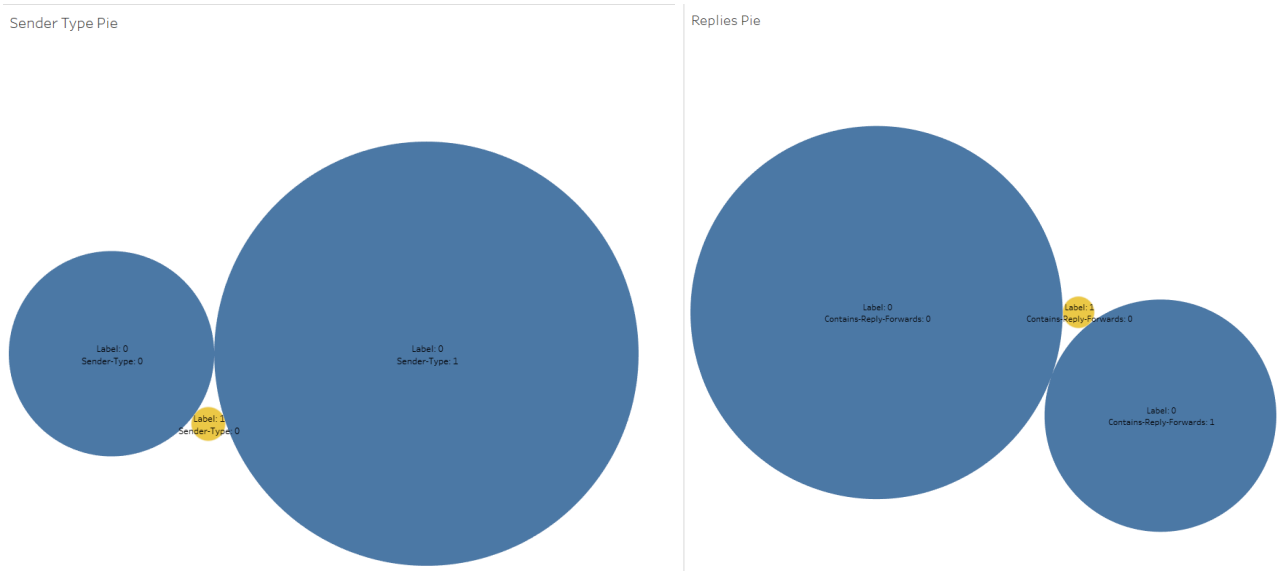


Tableau Visualisation of Email Sender-Type (0 = External or 1 = Internal) and Contains-Reply-Forwards

- Non scam (Label = 0) = 430855
- Scam (Label = 1) = 2219

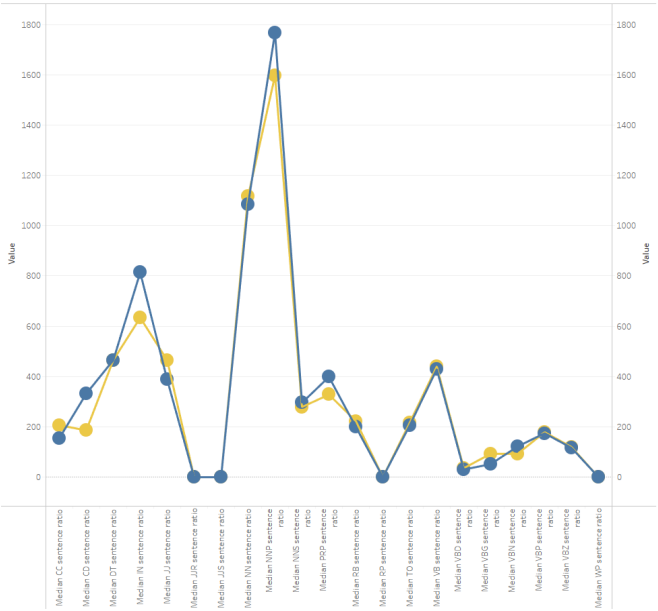


Tableau Visualisation of POS tags (normalised total count)

More Prevalent POS in Scams:	**Less Prevalent POS in Scams:**
CC (Coordinating Conjunctions): Scammers may use coordinating conjunctions to link phrases or clauses, creating a sense of urgency or persuasiveness.	CD (Cardinal Numbers): Scams often avoid using specific numbers to maintain flexibility and appeal to a broader audience.
JJ (Adjectives): Adjectives can be used to enhance emotional appeal or create a sense of urgency or trust.	IN (Prepositions): Prepositions are generally less prevalent in scam communication as they are more about relational words rather than directly persuasive language.

NN (Nouns): Nouns are used extensively in scams to describe products, services, or situations, often designed to evoke emotions or desires.	NNP (Proper Nouns): Proper nouns are less common as scammers often aim for generic appeals to reach a wider audience.
RB (Adverbs): Adverbs can modify verbs to emphasise actions or qualities, enhancing the persuasive nature of scam messages.	PRP (Personal Pronouns): Scammers may avoid personal pronouns to maintain a more generic tone, avoiding direct personalization.
VBG (Verb Gerunds): Verb gerunds can be used to describe ongoing actions or processes, adding a sense of immediacy or activity.	NNS (Plural Nouns): Similar to singular nouns, plural nouns might be less prevalent to maintain a broad appeal. Example: Instead of "Customers," they might use "People" or "Individuals."
	VBN (Verb Past Participles): Past participles might be less prevalent as they denote completed actions, which may not align with the urgency or immediacy scammers want to convey.

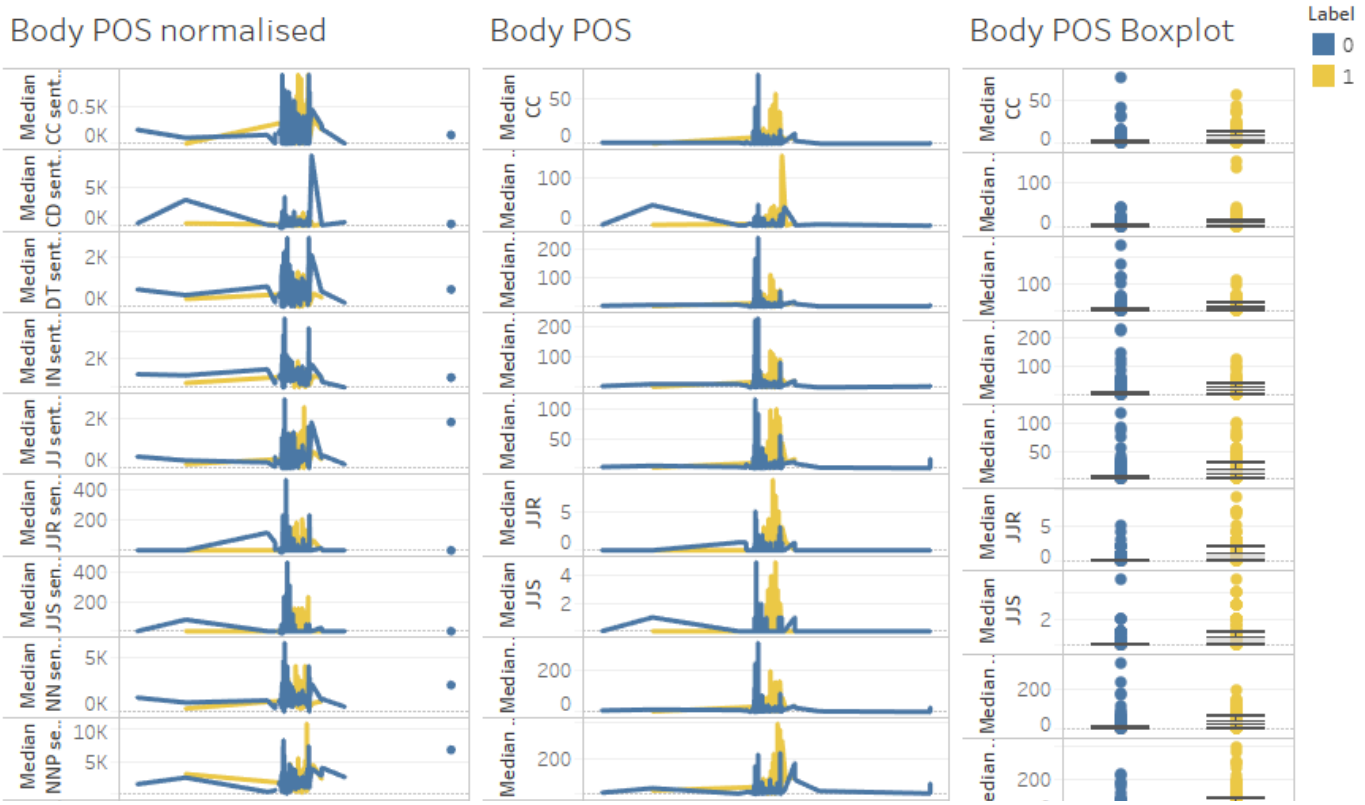


Tableau Visualisation of POS tags (raw and normalised over time)

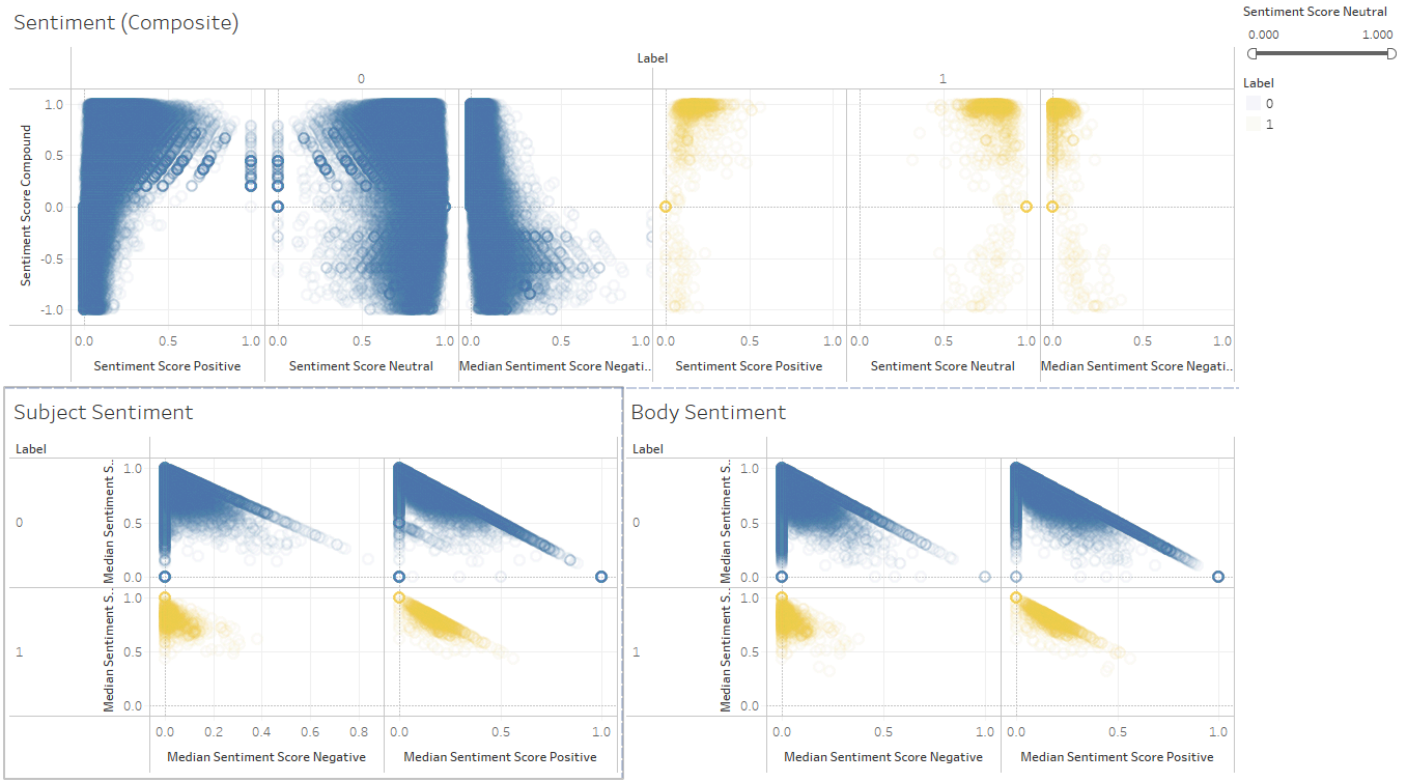
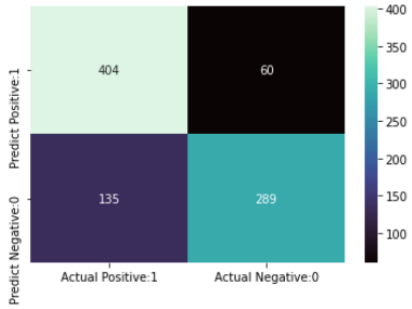
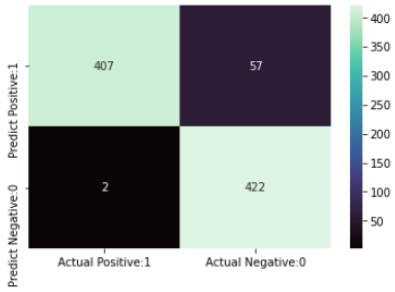
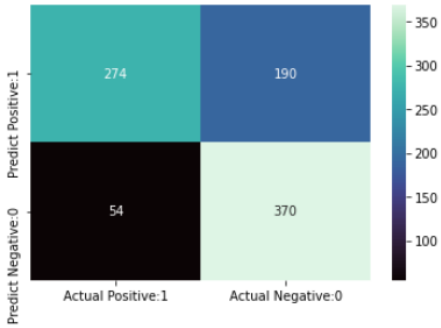
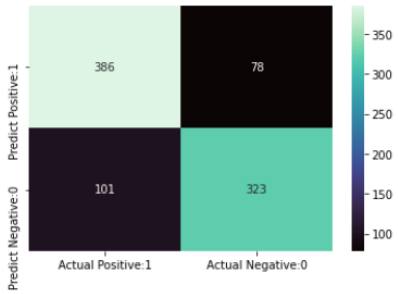
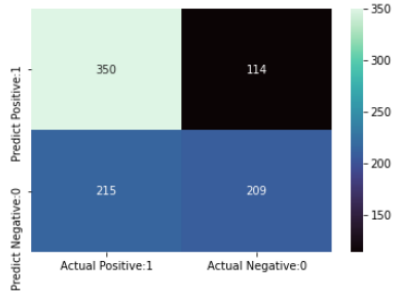
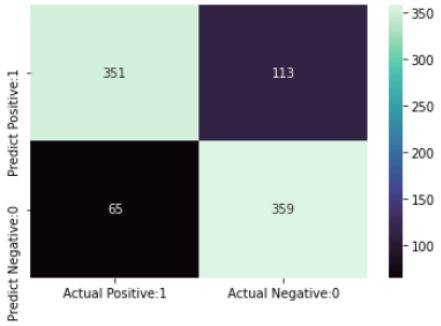


Tableau Visualisation of Sentiment Scores

Tableau Visualisation of Syntactical Features

SVM	POS	CIRCUM	TOPICS
Linear	Accuracy: 0.780405 Precision: 0.828080 Recall: 0.681603 F1 Score: 0.747736 ROC-AUC: 0.828165 	Accuracy: 0.933558 Precision: 0.881002 Recall: 0.995283 F1 Score: 0.934662 ROC-AUC: 0.984931 	Accuracy: 0.725225 Precision: 0.660714 Recall: 0.872641 F1 Score: 0.752032 ROC-AUC: 0.736540 
RBF	Accuracy: 0.798423 Precision: 0.805486 Recall: 0.761792 F1 Score: 0.783030 ROC-AUC: 0.868887 	Accuracy: 0.629504 Precision: 0.647058 Recall: 0.492924 F1 Score: 0.559571 ROC-AUC: 0.648409 	Accuracy: 0.799549 Precision: 0.760593 Recall: 0.846698 F1 Score: 0.801339 ROC-AUC: 0.831500 

#	index	Accuracy	Precision	Recall	F1-Score	AUC-ROC	AUC-PR	Training Time (s)
0	AdaBoost	0.99	0.61	0.16	0.25	0.99	0.43	61.47
1	CatBoost	1.00	0.88	0.60	0.71	1.00	0.78	109.96
2	GradientBoosting	0.99	0.77	0.15	0.25	0.99	0.46	311.87
3	LightGBM	1.00	0.67	0.46	0.55	0.97	0.55	8.12
4	Random Forest	1.00	0.98	0.51	0.67	1.00	0.82	68.23
5	SVM	0.99	0.00	0.00	0.00	0.61	0.01	4752.75
6	XGBoost	1.00	0.88	0.57	0.69	1.00	0.78	9.71
7	k-NN	0.99	0.48	0.03	0.05	0.63	0.05	0.14

Additional models attempted for Scam Classification

[0]Logistic Regression Training Accuracy: 0.7126760563380282
[0]Logistic Regression Testing Accuracy: 0.5898876404494382

	precision	recall	f1-score	support
0	1.00	0.25	0.40	4
1	0.20	0.10	0.13	10
2	0.00	0.00	0.00	4
3	0.83	0.86	0.85	81
4	0.33	0.57	0.42	21
5	0.17	0.20	0.18	5
6	0.81	0.81	0.81	21
7	0.00	0.00	0.00	3
8	0.00	0.00	0.00	4
9	0.00	0.00	0.00	2
10	0.17	0.50	0.25	2
11	0.17	0.10	0.12	21
accuracy			0.59	178
macro avg	0.31	0.28	0.26	178
weighted avg	0.57	0.59	0.57	178

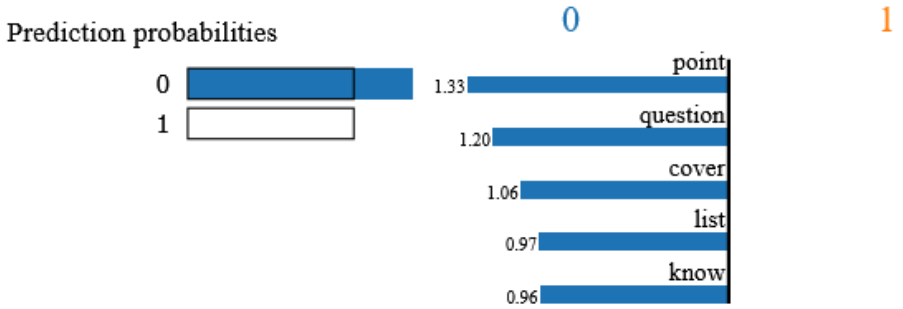
[1]KNeighborns Training Accuracy: 0.676056338028169
[1]KNeighborns Testing Accuracy: 0.5898876404494382

	precision	recall	f1-score	support
0	1.00	0.50	0.67	4
1	0.33	0.20	0.25	10
2	0.00	0.00	0.00	4
3	0.76	0.91	0.83	81
4	0.41	0.52	0.46	21
5	0.00	0.00	0.00	5
6	0.70	0.76	0.73	21
7	0.00	0.00	0.00	3
8	0.00	0.00	0.00	4
9	0.00	0.00	0.00	2
10	0.00	0.00	0.00	2
11	0.00	0.00	0.00	21
accuracy			0.59	178
macro avg	0.27	0.24	0.24	178
weighted avg	0.51	0.59	0.55	178

[0]Logistic Regression Training Accuracy: 0.7126760563380282
[0]Logistic Regression Testing Accuracy: 0.5898876404494382

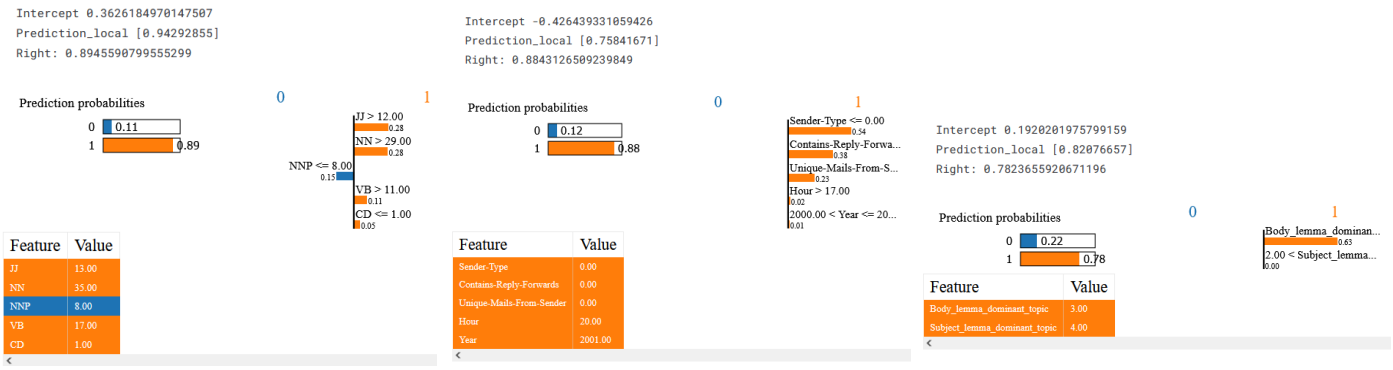
	precision	recall	f1-score	support
0	1.00	0.25	0.40	4
1	0.20	0.10	0.13	10
2	0.00	0.00	0.00	4
3	0.83	0.86	0.85	81
4	0.33	0.57	0.42	21
5	0.17	0.20	0.18	5
6	0.81	0.81	0.81	21
7	0.00	0.00	0.00	3
8	0.00	0.00	0.00	4
9	0.00	0.00	0.00	2
10	0.17	0.50	0.25	2
11	0.17	0.10	0.12	21
accuracy			0.59	178
macro avg	0.31	0.28	0.26	178
weighted avg	0.57	0.59	0.57	178

Scam Genre “Topic” Prediction using linguistics features

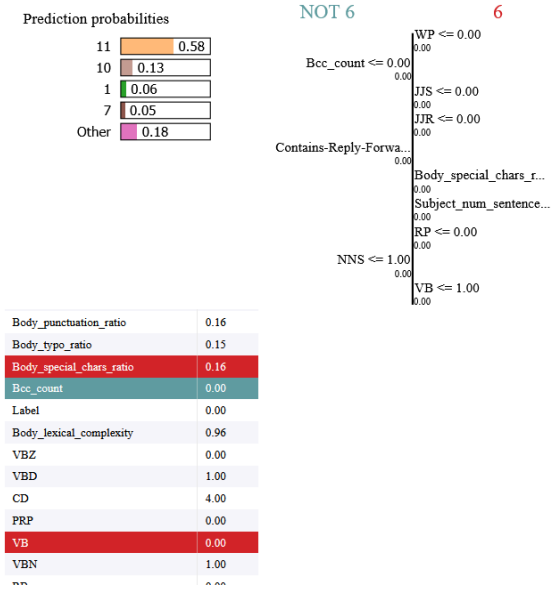


Text with highlighted words

question for legal ruth , attach be my list of question . some might be repetitive , but just want to cover all my point . if I think of any other , I will let you know . maria ps .. file be locate under our transportation folder

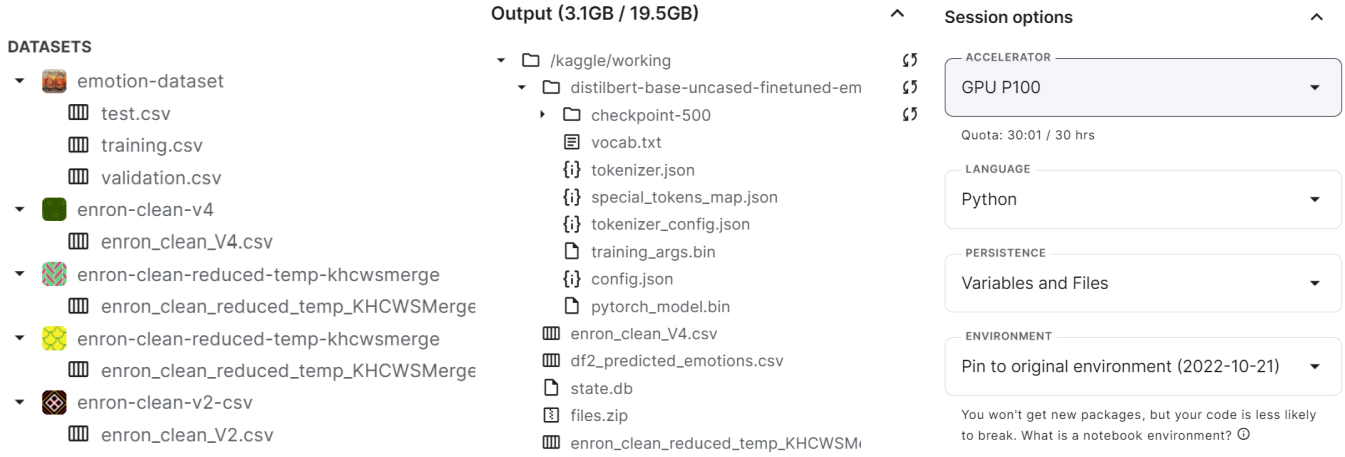


LIME interpreter for Label and LDA topic prediction



Example of inaccurate SVM topic prediction (predicted 11 when it should be topic 6)

11. Additional Current Works



Current Works - Emotion Sentiment Analysis Prediction Model

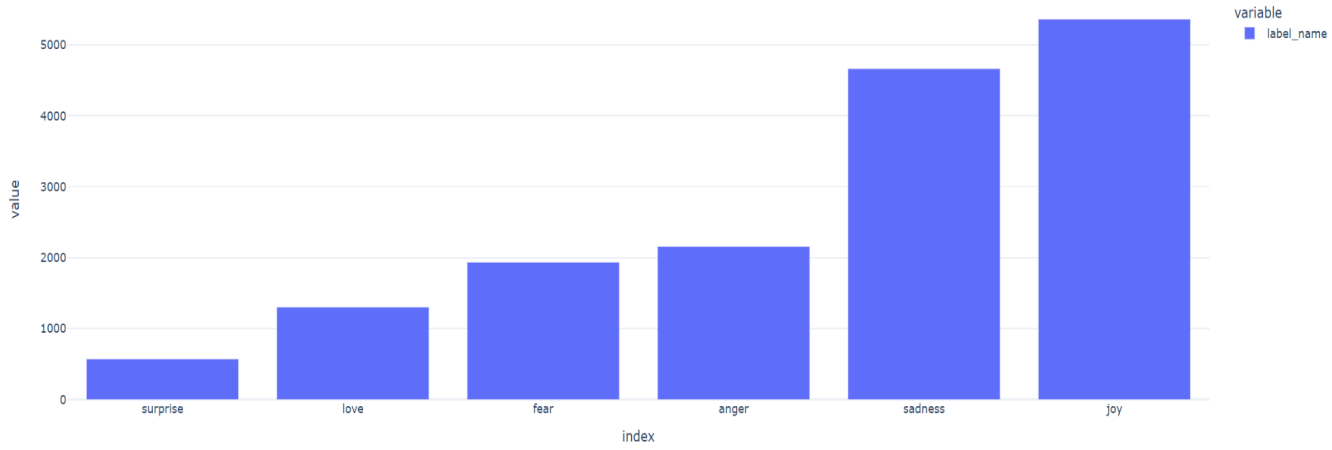
Emotion sentiment analysis, is a subfield of natural language processing (NLP), aims to automatically detect and classify emotions expressed in textual data. This research project explores the use of deep learning models, specifically DistilBERT, for emotion sentiment analysis.

The methodology involves data preprocessing, feature extraction using pre-trained language models, model training, evaluation, error analysis, and inference on new data. The results demonstrate the effectiveness of the proposed approach in accurately classifying emotions in textual data.

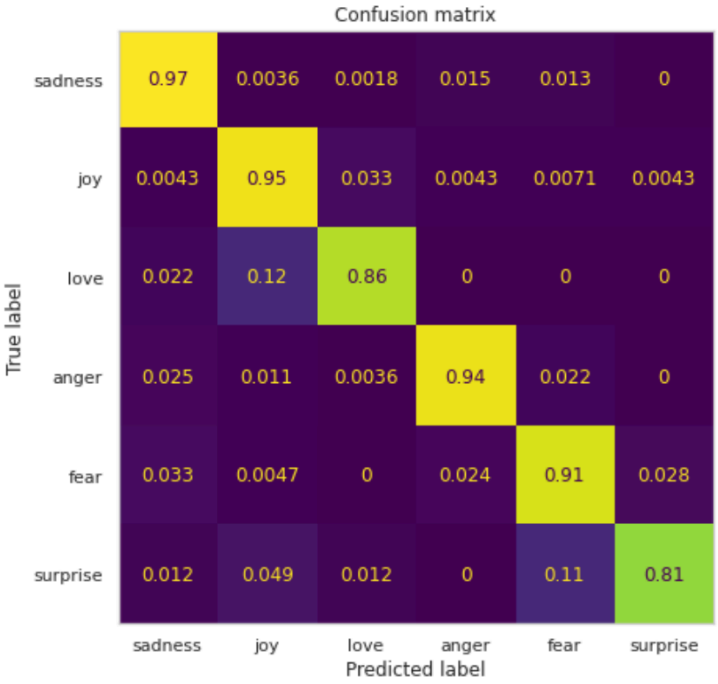
Emotion sentiment analysis plays a crucial role in understanding human emotions expressed in text, enabling applications such as sentiment analysis in social media, customer feedback analysis, and opinion mining. Traditional machine learning approaches often struggle to capture the complex semantic meaning of text. In contrast, deep learning models, particularly transformer-based architectures like DistilBERT, have shown remarkable performance in various NLP tasks.

Data Preparation Stage for Emotional Sentiment Analysis
For emotion sentiment analysis we require a labelled dataset where each text sample is associated with an emotion label (e.g., joy, sadness, anger, etc.). 6 classes joy, sadness, anger, fear, love and surprise; multiclass problem

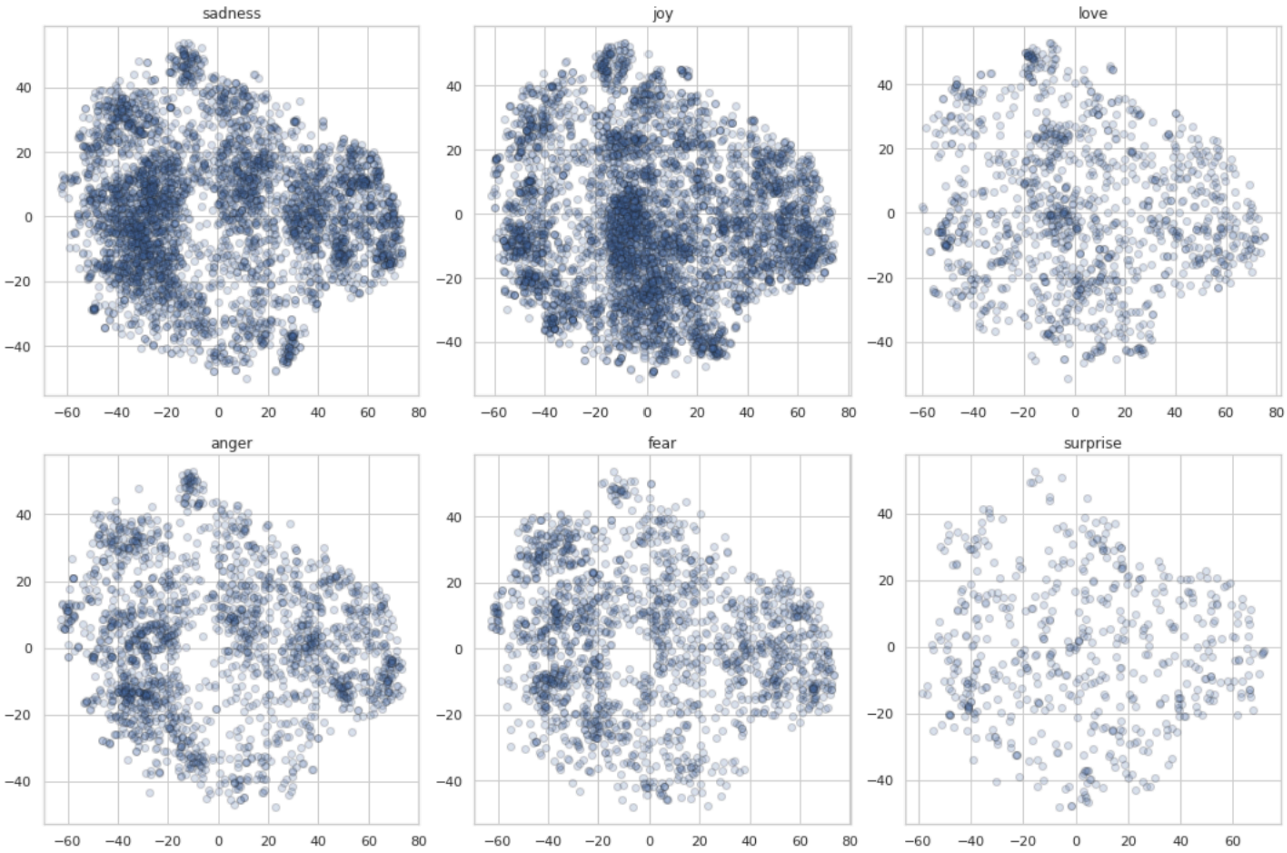
The dataset is preprocessed to remove noise, such as special characters, punctuation, and stopwords. Additionally, text normalisation techniques like lowercasing and stemming may be applied. The preprocessed dataset is split into training, validation, and test sets to evaluate the model's performance.



	text	label	label_name	Words Per Tweet
0	i didnt feel humiliated	0	sadness	4
1	i can go from feeling so hopeless to so damned...	0	sadness	21
2	im grabbing a minute to post i feel greedy wrong	3	anger	10
3	i am ever feeling nostalgic about the fireplac...	2	love	18
4	i am feeling grouchy	3	anger	4
5	ive been feeling a little burdened lately wasn...	0	sadness	12
6	ive been taking or milligrams or times recomme...	5	surprise	23
7	i feel as confused about life as a teenager or...	4	fear	17
8	i have been with petronas for years i feel tha...	1	joy	19
9	i feel romantic too	2	love	4
10	i feel like i have to make the suffering i m s...	0	sadness	14
11	i do feel that running is a divine experience ...	1	joy	21
12	i think it s the easiest time of year to feel ...	3	anger	12
13	i feel low energy i m just thirsty	0	sadness	8
14	i have immense sympathy with the general point...	1	joy	42
15	i do not feel reassured anxiety is on each side	1	joy	10
16	i didnt really feel that embarrassed	0	sadness	6
17	i feel pretty pathetic most of the time	0	sadness	8
18	i started feeling sentimental about dolls i ha...	0	sadness	23
19	i now feel compromised and skeptical of the va...	4	fear	17
20	i feel irritated and rejected without anyone d...	3	anger	12



The confusion matrix has two dimensions which shows the predicted classes and actual classes. The class matrix probabilities provide us with insights into the model confidence levels for each class label for the six emotions, sadness, joy, love, anger, fear, and surprise. Higher the value for probabilities will indicate higher confidence in the label prediction, whereas lower probabilities could potentially mean a wrong class prediction.



Visualising the Training Data
We can visualise each class distribution the model will need to separate in lower dimension space (projections onto a lower-dimensional space).

- We have a lot of categories overlapping in lower dimensional space
  - (Doesn't mean the model won't be able to classify them in higher dimensional space)
- If they are separable in the projected space, they will probably be separable in higher dimensional space
- We will utilise a manifold learning unsupervised model TSNE

Distilbert-base-uncased
Advantages

- Faster and more memory-efficient compared to BERT-base.
- Suitable for scenarios where computational resources are limited or speed is crucial.
- Trained through distillation, resulting in a smaller model size while preserving much of the performance of BERT-base.

Application

- Text classification tasks where model size and speed are important considerations.
- Applications where real-time inference is required, such as chatbots and online services.

DistilBERT, a variant of BERT known for its efficiency and comparable performance in natural language processing tasks. By using DistilBERT with core libraries from the Hugging Face Datasets, Tokenizers & Transformers we can obtain emotion sentiment analysis classification and scores. We can build a classification machine learning model that will be able to automatically identify emotional states (eg. anger, joy) from textual datasets.

The main advantage of this model is that it is much smaller than BERT which is much more efficient, still able to achieve comparable classification performance. With a limitation that DistilBERT has a maximum context size is 512 tokens.

We used libraries developed by Hugging Face, Datasets, Tokenizers, and Transformers. While sentiment analysis datasets often entail binary classification tasks, our dataset has six different sentiment classes for sadness, joy, love, anger, fear, and surprise. Hence, we are required to approach the problem as a multiclass classification problem.

For model training and evaluation, we utilise three primary datasets, training, testing, and validation. The datasets are important to improve the model performance and their generalizability in predicting the sentiment classes.

Application of Model Prediction using our Dataset

```
from torch.nn.functional import cross_entropy

def forward_pass_with_label(batch):

    # Place all input tensors on the same device as the model
    inputs = {k:v.to(device) for k,v in batch.items()
              if k in tokenizer.model_input_names}

    with torch.no_grad():
        output = model(**inputs)
        pred_label = torch.argmax(output.logits, axis=-1)
        loss = cross_entropy(output.logits, batch["label"].to(device),
                             reduction="none")

    # Place outputs on CPU for compatibility with other dataset columns
    return {"loss": loss.cpu().numpy(),
            "predicted_label": pred_label.cpu().numpy()}

# Convert our dataset back to PyTorch tensors
emotions_encoded.set_format("torch",
                            columns=["input_ids", "attention_mask", "label"])
# Compute loss values
emotions_encoded["validation"] = emotions_encoded["validation"].map(forward_pass_with_label,
                                                                    batched=True,
                                                                    batch_size=16)
```

```
from transformers import Trainer, TrainingArguments

bs = 64 # batch size
logging_steps = len(emotions_encoded["train"]) // bs
model_name = f"{model_ckpt}-finetuned-emotion"
training_args = TrainingArguments(output_dir=model_name,
                                  num_train_epochs=3,              # number of training epochs
                                  learning_rate=2e-5,              # model learning rate
                                  per_device_train_batch_size=bs,  # batch size
                                  per_device_eval_batch_size=bs,   # batch size
                                  weight_decay=0.01,
                                  evaluation_strategy="epoch",
                                  disable_tqdm=False,
                                  report_to="none",
                                  logging_steps=logging_steps,
                                  push_to_hub=False,
                                  log_level="error")
```

```python
# Load the tokenizer and model
tokenizer = AutoTokenizer.from_pretrained("distilbert-base-uncased-finetuned-emotion")
model = AutoModelForSequenceClassification.from_pretrained("distilbert-base-uncased-finetuned-emotion")

# Move the model to the appropriate device (CPU or GPU)
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model.to(device)

# New unseen by model data
new_data = 'That boy has quite a temper. He needs to learn to control his temper. She hit him in a fit of temper. He slammed the door and left in a temper.'

# Tokenize the input text
tokenized_input = tokenizer(new_data, return_tensors="pt", truncation=True, padding=True)

# Move the inputs to the appropriate device (CPU or GPU)
tokenized_input = {key: val.to(device) for key, val in tokenized_input.items()}

# Make predictions
with torch.no_grad():
    outputs = model(**tokenized_input)

# Get the predicted labels
predicted_labels = torch.argmax(outputs.logits, dim=1)

# Convert predicted labels to emotion names
emotion_names = ['sadness', 'joy', 'love', 'anger', 'fear', 'surprise']
predicted_emotions = [emotion_names[label] for label in predicted_labels]

# Display the predictions
for text, emotion in zip(new_data, predicted_emotions):
    print(f"Text: {text}\nPredicted Emotion: {emotion}\n")
```

```
Text: T
Predicted Emotion: anger
```

```python
# Load the tokenizer and model
tokenizer = AutoTokenizer.from_pretrained("distilbert-base-uncased-finetuned-emotion")
model = AutoModelForSequenceClassification.from_pretrained("distilbert-base-uncased-finetuned-emotion")

# Move the model to the appropriate device (CPU or GPU)
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model.to(device)

# New unseen by model data
new_data = 'He is feeling sad because his pet died. People were sad that he was leaving. The experience left her sadder but wiser.'

# Tokenize the input text
tokenized_input = tokenizer(new_data, return_tensors="pt", truncation=True, padding=True)

# Move the inputs to the appropriate device (CPU or GPU)
tokenized_input = {key: val.to(device) for key, val in tokenized_input.items()}

# Make predictions
with torch.no_grad():
    outputs = model(**tokenized_input)

# Get the predicted labels
predicted_labels = torch.argmax(outputs.logits, dim=1)

# Convert predicted labels to emotion names
emotion_names = ['sadness', 'joy', 'love', 'anger', 'fear', 'surprise']
predicted_emotions = [emotion_names[label] for label in predicted_labels]

# Display the predictions
for text, emotion in zip(new_data, predicted_emotions):
    print(f"Text: {text}\nPredicted Emotion: {emotion}\n")
```

```
Text: H
Predicted Emotion: sadness
```

```python
import torch
from transformers import AutoTokenizer, AutoModelForSequenceClassification

# Load the tokenizer and model
tokenizer = AutoTokenizer.from_pretrained("distilbert-base-uncased-finetuned-emotion")
model = AutoModelForSequenceClassification.from_pretrained("distilbert-base-uncased-finetuned-emotion")

# Allocate model to appropriate device (CPU or GPU)
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model.to(device)

# Function predict emotions for given text
def predict_emotion(text):
    # Tokenize the input text
    tokenized_input = tokenizer(text, return_tensors="pt", truncation=True, padding=True)
    # Move the inputs to the appropriate device (CPU or GPU)
    tokenized_input = {key: val.to(device) for key, val in tokenized_input.items()}
    # Make predictions
    with torch.no_grad():
        outputs = model(**tokenized_input)
    # Get the predicted labels
    predicted_labels = torch.argmax(outputs.logits, dim=1)
    # Convert predicted labels to emotion names
    emotion_names = ['sadness', 'joy', 'love', 'anger', 'fear', 'surprise']
    predicted_emotions = [emotion_names[label] for label in predicted_labels]
    return predicted_emotions[0]

# Testing first 100 rows df
df2_subset = df2.head(100)  # Selecting the first 100 rows for testing
df2_subset['predicted_emotion'] = df2_subset['cleaned_body'].apply(predict_emotion)

# Display df with predicted emotions
print(df2_subset[['cleaned_body', 'predicted_emotion']])
```

	cleaned_body	predicted_emotion
0	status john really sure happen impression visi...	joy
1	summer inverse suck hope make money natga last...	joy
2	wti bullet swap contract hi follow e mail rece...	joy
3	fwd article suspend rabbi quit seminary presid...	anger
4	daily chart matrix hot link information contai...	joy
5	resume john thank email offer route resume exp...	joy
6	wrong address john think wrong email address b...	anger
7	additional offering round portfolio john leave...	joy
8	crazy mite scale little sort length sprd lengt...	joy
9	little bird tell read time get free zdnet oneb...	joy
10	big news sure hear fun news yet rob move bosto...	joy
11	think rite curve flatten end overvalue part th...	joy
12	bnp paribas commodity future ng marketwatch se...	joy
13	daily chart matrix hot link information contai...	joy
14	spread mkt get little bearish back winter thin...	joy
15	ever get check send article owe click link jef...	joy
16	service agreement john thank opportunity provi...	joy
17	daily chart matrix hot link information contai...	joy
18	password assistance greeting finish reset pass...	joy
19	f g weather moderate enuf switch offset hdd ca...	joy
20	nat gas market analysis attach please find nat...	joy
21	wassup hey freak take study break final ick th...	joy
22	option candlestick information contain herein ...	joy
23	article energy industry raise production recor...	joy
24	bank wire wsex dear mr lavorato send october c...	joy
25	agreement john hope ok today say yesterday poo...	joy
26	request additional information thank request a...	joy
27	natural gas update late natural report	joy
28	mother day thank beautiful flower arrive late ...	joy
29	bnp paribas commodity future ng marketwatch se...	joy
30	defense open wonderful idea new orleans epao	joy
31	thing houston hello mr arnold hope thing calm ...	joy
32	attach please find follow today natural gas ma...	joy
33	send friend work home depot corporate office s...	joy
34	mom itinerary panama canal cruise ship sun pri...	joy
35	nat gas market analysis attach please find nat...	joy
36	fwd christmas list get cheap stuff	joy
37	free shipping amazoniversary image image dear ...	joy
38	dell online order dell enron clickathome order...	joy
39	dell order confirmation dear jennifer medcalf ...	joy
40	option candlestick hot link information contai...	joy
41	revise unleaded chart unleaded	joy
42	image log sign account mgt insight image get q...	joy
43	save big clearance event image image image exp...	joy
44	love well mkts change bit since writing netura...	joy
45	nat gas market analysis attach please find nat...	joy
46	close auction noon advertise nobody buy sell k...	joy
47	owe irs money tax problem owe irs money debt h...	joy
48	dallas world congress houston chapter receptio...	joy
49	earn bonus starpoint night earn k per day reme...	joy
50	hot lot without bid quick reminder current auc...	joy
51	daily hoil unlde cahrt information contain her...	joy
52	spec event great cabernet value click last mai...	joy
53	ola follow heffner sad carr people ny technica...	joy
54	dear mr arnold please let know call week conve...	joy
55	daily chart matrix hot link information contai...	joy

Sample output result for prediction of emotions application for the rest of the dataframe
For further feature engineering of sentiment analysis for emotions classification

- For reducing computational time and efficiency, we will not be adding the individual emotional scores for sadness, joy, love, anger, fear and surprise
- Due to the fact that by obtaining the score for each of them we are creating additional 6 columns x 500K rows which requires significant computation time to perform with the limited time for project submission

```python
preds = classifier(new_data, return_all_scores=True)
preds
```

```
[{'label': 'LABEL_0', 'score': 0.004624964203685522},
 {'label': 'LABEL_1', 'score': 0.9682039618492126},
 {'label': 'LABEL_2', 'score': 0.0035104467533528805},
 {'label': 'LABEL_3', 'score': 0.003585223574191332},
 {'label': 'LABEL_4', 'score': 0.0030851562041789293},
 {'label': 'LABEL_5', 'score': 0.016990238800644875}]
```

```python
preds1 = classifier(new_data2, return_all_scores=True)
preds1
```

```
[{'label': 'LABEL_0', 'score': 0.545443058013916},
 {'label': 'LABEL_1', 'score': 0.033940017223358154},
 {'label': 'LABEL_2', 'score': 0.006987168453633785},
 {'label': 'LABEL_3', 'score': 0.05618271231651306},
 {'label': 'LABEL_4', 'score': 0.3468289077281952},
 {'label': 'LABEL_5', 'score': 0.010618075728416443}]
```

Label0 = sadness, Label1 = joy, Label2 = love, Label3 = anger, Label4 = fear, Label5 = Surprise

To improve computation efficiency, we will only obtain emotions for sadness, joy, love, anger, fear, and surprise, for the entire dataset of 500,000 text entries. Due to time limitations we will not

include the emotional scores for each column of emotion as it will create additional six columns, resulting in a substantial increase in computational time and resources needed. It was more of a conceptual understanding of how emotion can be derived from textual data.

As we read up on current literature reviews, the only way to obtain such results is to train a machine learning model or use a large language model to perform emotion sentiment analysis. As there are no readily available libraries to perform the complex analysis as of our understanding. Most methods such as NLTK, SpaCy, TextBlob, Scikit-learn, CoreNLP only allow us to obtain sentiment polarity or scores such as negative, positive, neutral, compound and sentiment scores, such that they are unable to provide emotion sentiment analysis

It is highlighting that emotional scores could potentially provide informative features that could enhance the predictive performance of machine learning models. These scores capture emotional sentiments scores embedded in textual data. It can help to offer insights into the underlying sentiments and attitudes expressed for fraud and non-fraud text.

Moreover, the feature engineering of emotional scores can allow machine learning models to learn from the emotional context of text data, potentially leading to more accurate predictions and deeper insights into fraudulent or scam-related activities.

```
print('')
print(f'Vocab size: {tokenizer.vocab_size}')
print(f'Max length: {tokenizer.model_max_length}')
print(f'Tokeniser model input names: {tokenizer.model_input_names}')
```

```
Vocab size: 30522
Max length: 512
Tokeniser model input names: ['input_ids', 'attention_mask']
```

text	label	label_name
i didnt feel humiliated	0	sadness
i can go from feeling so hopeless to so damned hopeful just from being around someone who cares and is awake	0	sadness
im grabbing a minute to post i feel greedy wrong	3	anger
i am ever feeling nostalgic about the fireplace i will know that it is still on the property	2	love
i am feeling grouchy	3	anger
ive been feeling a little burdened lately wasnt sure why that was	0	sadness
ive been taking or milligrams or times recommended amount and ive fallen asleep a lot faster but i also feel like so funny	5	surprise
i feel as confused about life as a teenager or as jaded as a year old man	4	fear
i have been with petronas for years i feel that petronas has performed well and made a huge profit	1	joy
i feel romantic too	2	love

```
text = 'Tokenisation of text is a core task of NLP.'
tokenised_text = list(text)

# Character Tokenised list
print(f'Number of tokens: {len(tokenised_text)}')
print(tokenised_text)
```

```
Number of tokens: 43
['T', 'o', 'k', 'e', 'n', 'i', 's', 'a', 't', 'i', 'o', 'n', ' ', 'o', 'f', ' ', 't', 'e', 'x', 't', ' ', 'i', 's', ' ', 'a', ' ', 'c', 'o', 'r', 'e', ' ', 't', 'a', 's', 'k', ' ', 'o', 'f',
' ', 'N', 'L', 'P', '.']
```

Limitations

We would also like to highlight that there are limitations with the current model that it takes in a token size of 512 tokens, by tokenizing the textual data to feed into the model for classification task. The limitation poses a significant challenge as it would mean text data that are longer than 512 token size will be essentially cut off and not fed into the model.

We have attempted to expand the token size to 4096 as it was noted that the model itself has a size limit of 512 tokens, hence we believe the solution to this problem is to apply extractive summarization into shortest sentences before feeding into the machine learning model. In that case we will lose some textual information but it ensures we capture the main text information. Moreover it can also help to reduce computation time as it takes up to 7 hours just to run and obtain the emotions sentiment analysis.

Entity Recognition (NER)

- Google Natural Language AI API
- Google NER
- Refer to code uploads

Future Work (If we have time we will upload the Feature Engineering)
Methodology and Process
Entity Recognition (NER) and POS Tag for Future Tenses

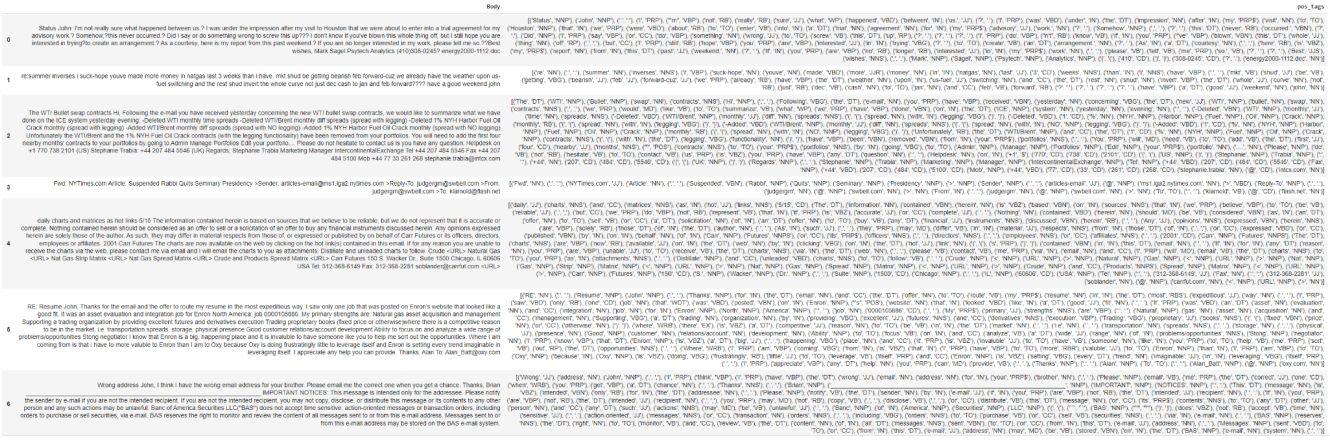
- Perform Tokenization for each sentence
- Perform Part of Speech Tagging
- For Each of the Body Text
- Creation of POS Categories
- Count Occurrences of each POS Tag Categories
- Feature Engineering for Future Tense
- To train model to detection of Fraud by identifying future Tenses
- Feed into models that will take object type, unstructured data (advantage)
- RF, LightGBM, XGBoost
- Many other models will not be able to

```python
import pandas as pd
import nltk
from nltk import word_tokenize, pos_tag

# Download NLTK resources
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')

# Function to extract parts of speech from Sentence
def extract_pos(sentence):
    tokens = word_tokenize(sentence)
    pos_tags = pos_tag(tokens)
    return pos_tags

# Convert df Body column to string type
df['Body'] = df['Body'].astype(str)

# Apply the function df Body
df['pos_tags'] = df['Body'].apply(extract_pos)

# Define POS categories
pos_categories = {
    'Adjective': 'JJ',
    'Adverb': 'RB',
    'Verb': ['VB', 'VBD', 'VBG', 'VBN', 'VBP', 'VBZ'],
    'Noun': ['NN', 'NNS', 'NNP', 'NNPS'],
    'Pronoun': ['PRP', 'PRP$'],
    'Preposition': 'IN',
    'Determiner': 'DT',
    'Modal Verb': 'MD',
    'Past Tense': 'VBD',
    'Present Tense': ['VB', 'VBP', 'VBZ'],
    'Future Tense': 'MD'  # Modal verbs as future tense
}

# Function to extract POS based on category and count occurrences
def extract_pos_category(pos_tags, category):
    if isinstance(category, list):
        words = [word for word, pos in pos_tags if pos in category]
    else:
        words = [word for word, pos in pos_tags if pos == category]
    return words, len(words)

# Create columns for each Column POS category with Counter
for full_name, abbreviation in pos_categories.items():
    df[full_name], df[f'{full_name} Count'] = zip(*df['pos_tags'].apply(lambda x: extract_pos_category(x, abbreviation)))

# Display df POS
df
```

```python
import pandas as pd
import nltk
from nltk import word_tokenize, pos_tag

# Download NLTK resources
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')

# Test Sentence Output
data = {'sentences': ["The quick brown fox will jump over the lazy dog, and then will sit down beside it."]}
df = pd.DataFrame(data)

# Function to extract parts of speech from a sentence
def extract_pos(sentence):
    # Convert to string if input is not already a string
    if not isinstance(sentence, str):
        sentence = str(sentence)
    tokens = word_tokenize(sentence)
    pos_tags = pos_tag(tokens)
    return pos_tags

# Apply the function to df
df['pos_tags'] = df['sentences'].apply(extract_pos)

# Define POS categories
pos_categories = {
    'Adjective': 'JJ',
    'Adverb': 'RB',
    'Verb': ['VB', 'VBD', 'VBG', 'VBN', 'VBP', 'VBZ'],
    'Noun': ['NN', 'NNS', 'NNP', 'NNPS'],
    'Pronoun': ['PRP', 'PRP$'],
    'Preposition': 'IN',
    'Determiner': 'DT',
    'Modal Verb': 'MD',
    'Past Tense': 'VBD',
    'Present Tense': ['VB', 'VBP', 'VBZ'],
    'Future Tense': 'MD'  # Taking Modal verbs as future tense
}

# Create columns for each Column POS category with Counter
for category, abbreviation in pos_categories.items():
    if isinstance(abbreviation, list):
        df[category + ' Count'] = df['pos_tags'].apply(lambda x: sum(1 for _, pos in x if pos in abbreviation))
        df[category] = df['pos_tags'].apply(lambda x: [word for word, pos in x if pos in abbreviation])
    else:
        df[category + ' Count'] = df['pos_tags'].apply(lambda x: sum(1 for _, pos in x if pos == abbreviation))
        df[category] = df['pos_tags'].apply(lambda x: [word for word, pos in x if pos == abbreviation])

# Drop the pos_tags column
df.drop(columns=['pos_tags'], inplace=True)

# Display the DataFrame
df
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
```

	sentences	Adjective Count	Adjective	Adverb Count	Adverb	Verb Count	Verb	Noun Count	Noun	Pronoun Count	Pronoun	Preposition Count	Preposition	Determiner Count	Determiner	Modal Verb Count	Modal Verb	Past Tense Count	Past Tense	Present Tense Count	Present Tense	Future Tense Count	Future Tense
0	The quick brown fox will jump over the lazy do...	2	[quick, lazy]	1	[then]	2	[jump, sit]	4	[brown, fox, dog, beside]	1	[it]	1	[over]	2	[The, the]	2	[will, will]	0	[]	2	[jump, sit]	2	[will, will]



	Body	pos_tags
0	Status John: I'm not really sure what happened between us ? I was under the impression after my visit to Houston that we were about to enter into a trial agreement for my advisory work ? Somehow,?this never occurred.? Did i say or do something wrong to screw this up??? i don't know if you've blown this whole thing off, but i still hope you are interested in trying?to create an arrangement.? As a courtesy, here is my report from this past weekend.? If you are no longer interested in my work, please tell me so.??Best wishes, Mark Sagel Paytech Analytics (410)308-02457 energy2000-1112.doc	[['Status', 'NNP'], ('John', 'NNP'], (',', ','), ('I', 'PRP'], ("'m", 'VBP'], ('not', 'RB'], ('really', 'RB'], ('sure', 'JJ'], ('what', 'WP'], ('happened', 'VBD'], ('between', 'IN'], ('us', 'JJ'], ('?', '.'], ('I', 'PRP'], ('was', 'VBD'], ('under', 'IN'], ('the', 'DT'], ('impression', 'NN'], ('after', 'IN'], ('my', 'PRP$'], ('visit', 'NN'], ('to', 'TO'], ('Houston', 'NNP'], ('that', 'IN'], ('we', 'PRP'], ('were', 'VBD'], ('about', 'RB'], ('to', 'TO'], ('enter', 'VB'], ('into', 'IN'], ('a', 'DT'], ('trial', 'NN'], ('agreement', 'NN'], ('for', 'IN'], ('my', 'PRP$'], ('advisory', 'JJ'], ('work', 'NN'], ('?', '.'], ('Somehow', 'NNP'], (',', ','], ('this', 'DT'], ('never', 'RB'], ('occurred', 'VBN'], ('.', '.'], ('Did', 'NNP'], ('i', 'PRP'], ('say', 'VBP'], ('or', 'CC'], ('do', 'VBP'], ('something', 'NN'], ('wrong', 'JJ'], ('to', 'TO'], ('screw', 'VB'], ('this', 'DT'], ('up', 'RP'], ('?', '.'], ('?', '.'], ('I', 'PRP'], ('do', 'VBP'], ("n't", 'RB'], ('know', 'VB'], ('if', 'IN'], ('you', 'PRP'], ('have', 'VBP'], ('just', 'RB'], ('given', 'VBN'], ('up', 'RP'], (',', ','], ('but', 'CC'], ('i', 'PRP'], ('still', 'RB'], ('hope', 'VBP'], ('you', 'PRP'], ('are', 'VBP'], ('interested', 'JJ'], ('in', 'IN'], ('trying', 'VBG'], ('?', '.'], ('to', 'TO'], ('create', 'VB'], ('an', 'DT'], ('arrangement', 'NN'], ('.?', '.'], ('As', 'IN'], ('a', 'DT'], ('courtesy', 'NN'], (',', ','], ('here', 'RB'], ('is', 'VBZ'], ('my', 'PRP$'], ('report', 'NN'], ('from', 'IN'], ('this', 'DT'], ('past', 'JJ'], ('weekend', 'NN'], ('.?', '.'], ('If', 'IN'], ('you', 'PRP'], ('are', 'VBP'], ('no', 'RB'], ('longer', 'RB'], ('interested', 'JJ'], ('in', 'IN'], ('my', 'PRP$'], ('work', 'NN'], (',', ','], ('please', 'VB'], ('tell', 'VB'], ('me', 'PRP'], ('so.??Best', 'JJS'], ('wishes', 'NNS'], (',', ','], ('Mark', 'NNP'], ('Sagel', 'NNP'], ('Paytech', 'NNP'], ('Analytics', 'NNP'], ('(', '('], ('410', 'CD'], (')', ')'], ('308-0245', 'CD'], ('?', '.'], ('energy2000-1112.doc', 'NN']]
1	re:summer inverses i suck-hope youve made more money in natgas last 3 weeks than i have. mtrl shud be getting bearish fstr forward-cuz we already have the weather upon us-fuel switching and the rest shud invert the whole curve not just doc cash to jan and fob forward???? have a good weekend john	[('re', 'NN'], (':', ':'], ('summer', 'NN'], ('inverses', 'NNS'], ('I', 'VBP'], ('suck-hope', 'NN'], ('youve', 'NN'], ('made', 'VBD'], ('more', 'JJR'], ('money', 'NN'], ('in', 'IN'], ('natgas', 'NN'], ('last', 'JJ'], ('3', 'CD'], ('weeks', 'NNS'], ('than', 'IN'], ('I', 'PRP'], ('have', 'VBP'], ('.', '.'], ('mtrl', 'VB'], ('shud', 'JJ'], ('be', 'VB'], ('getting', 'VBG'], ('bearish', 'JJ'], ('fstr', 'JJ'], ('forward-cuz', 'JJ'], ('we', 'PRP'], ('already', 'RB'], ('have', 'VBP'], ('the', 'DT'], ('weather', 'NN'], ('upon', 'IN'], ('us-fuel', 'JJ'], ('switching', 'NN'], ('and', 'CC'], ('the', 'DT'], ('rest', 'NN'], ('shud', 'NN'], ('invert', 'VBP'], ('the', 'DT'], ('whole', 'JJ'], ('curve', 'NN'], ('not', 'RB'], ('just', 'RB'], ('doc', 'VB'], ('cash', 'NN'], ('to', 'TO'], ('jan', 'NN'], ('and', 'CC'], ('fob', 'VB'], ('forward', 'RB'], ('?', '.'], ('?', '.'], ('?', '.'], ('I', 'PRP'], ('have', 'VBP'], ('a', 'DT'], ('good', 'JJ'], ('weekend', 'NN'], ('john', 'NN']]
2	The WTI Bullet swap contracts Hi, Following the e-mail you have received yesterday concerning the new WTI bullet swap contracts, we would like to summarize what we have done on the ICE system yesterday evening. -Deleted WTI monthly time spreads -Deleted WTI/Brent monthly diff spreads (spread with legging) -Deleted 1% NYH Harbor Fuel Oil Crack monthly (spread with legging) -Added WTI/Brent monthly diff spreads (spread with NO legging) -Added 1% NYH Harbor Fuel Oil Crack monthly (spread with NO legging) Unfortunately the WTI/Brent and the 1% NYH Fuel Oil Crack contracts (with the legging functionality) have been removed from your portfolios. You will need to add the first four nearby months' contracts to your portfolios by going to Admin Manage Portfolios Edit your portfo... Please do not hesitate to contact us is you have any question. Helpdesk on +1 770 738 2101 (US) Stephanie Trabka +44 207 484 5546 (UK) Regards, Stephanie Trabka Marketing Manager IntercontinentalExchange Tel +44 207 484 5546 Fax +44 207 484 5100 Mob +44 77 33 261 268 stephanie.trabka@intcx.com	[['The', 'DT'], ('WTI', 'NNP'], ('Bullet', 'NNP'], ('swap', 'NN'], ('contracts', 'NNS'], ('Hi', 'NNP'], (',', ','], ('Following', 'VBG'], ('the', 'DT'], ('e-mail', 'NN'], ('you', 'PRP'], ('have', 'VBP'], ('received', 'VBN'], ('yesterday', 'NN'], ('concerning', 'VBG'], ('the', 'DT'], ('new', 'JJ'], ('WTI', 'NNP'], ('bullet', 'NN'], ('swap', 'NN'], ('contracts', 'NNS'], (',', ','], ('we', 'PRP'], ('would', 'MD'], ('like', 'VB'], ('to', 'TO'], ('summarize', 'VB'], ('what', 'WP'], ('we', 'PRP'], ('have', 'VBP'], ('done', 'VBN'], ('on', 'IN'], ('the', 'DT'], ('ICE', 'NNP'], ('system', 'NN'], ('yesterday', 'NN'], ('evening', 'NN'], (':', ':'], ('-Deleted', 'VBD'], ('time', 'NN'], ('spreads', 'NNS'], ('-Deleted', 'VBD'], ('WTI/Brent', 'NNP'], ('monthly', 'JJ'], ('diff', 'NN'], ('spreads', 'NNS'], ('(', '('], ('spread', 'NN'], ('with', 'IN'], ('legging', 'VBG'], (')', ')'], ('-Deleted', 'VBD'], ('1', 'CD'], ('%', 'NN'], ('NYH', 'NNP'], ('Harbor', 'NNP'], ('Fuel', 'NNP'], ('Oil', 'NNP'], ('Crack', 'NNP'], ('monthly', 'JJ'], ('(', '('], ('spread', 'NN'], ('with', 'IN'], ('legging', 'VBG'], (')', ')'], ('-Added', 'VBD'], ('WTI/Brent', 'NNP'], ('monthly', 'JJ'], ('diff', 'NN'], ('spreads', 'NNS'], ('(', '('], ('spread', 'NN'], ('with', 'IN'], ('NO', 'NNP'], ('legging', 'VBG'], (')', ')'], ('-Added', 'VBD'], ('1', 'CD'], ('%', 'NN'], ('NYH', 'NNP'], ('Harbor', 'NNP'], ('Fuel', 'NNP'], ('Oil', 'NNP'], ('Crack', 'NNP'], ('monthly', 'JJ'], ('(', '('], ('spread', 'NN'], ('with', 'IN'], ('NO', 'NNP'], ('legging', 'VBG'], (')', ')'], ('Unfortunately', 'RB'], ('the', 'DT'], ('WTI/Brent', 'NNP'], ('and', 'CC'], ('the', 'DT'], ('1', 'CD'], ('%', 'NN'], ('NYH', 'NNP'], ('Fuel', 'NNP'], ('Oil', 'NNP'], ('Crack', 'NNP'], ('contracts', 'NNS'], ('(', '('], ('with', 'IN'], ('the', 'DT'], ('legging', 'VBG'], ('functionality', 'NN'], (')', ')'], ('have', 'VBP'], ('been', 'VBN'], ('removed', 'VBN'], ('from', 'IN'], ('your', 'PRP$'], ('portfolios', 'NNS'], ('.', '.'], ('You', 'PRP'], ('will', 'MD'], ('need', 'VB'], ('to', 'TO'], ('add', 'VB'], ('the', 'DT'], ('first', 'JJ'], ('four', 'CD'], ('nearby', 'JJ'], ('months', 'NNS'], ("'", 'POS'], ('contracts', 'NNS'], ('to', 'TO'], ('your', 'PRP$'], ('portfolios', 'NNS'], ('by', 'IN'], ('going', 'VBG'], ('to', 'TO'], ('Admin', 'NNP'], ('Manage', 'NNP'], ('Portfolios', 'NNP'], ('Edit', 'NNP'], ('your', 'PRP$'], ('portfolio', 'NN'], ('.', '.'], ('Please', 'NNP'], ('do', 'VB'], ('not', 'RB'], ('hesitate', 'VB'], ('to', 'TO'], ('contact', 'VB'], ('us', 'PRP'], ('is', 'VBZ'], ('you', 'PRP'], ('have', 'VBP'], ('any', 'DT'], ('question', 'NN'], ('.', '.'], ('Helpdesk', 'NNP'], ('on', 'IN'], ('+1', 'JJ'], ('770', 'CD'], ('738', 'CD'], ('2101', 'CD'], ('(', '('], ('US', 'NNP'], (')', ')'], ('Stephanie', 'NNP'], ('Trabka', 'NNP'], ('(', '('], ('+44', 'NN'], ('207', 'CD'], ('484', 'CD'], ('5546', 'CD'], (')', ')'], ('UK', 'NNP'], (')', ')'], ('Regards', 'NNP'], (',', ','], ('Stephanie', 'NNP'], ('Trabka', 'NNP'], ('Marketing', 'NNP'], ('Manager', 'NNP'], ('IntercontinentalExchange', 'NNP'], ('Tel', 'NNP'], ('+44', 'VBD'], ('207', 'CD'], ('484', 'CD'], ('5546', 'CD'], ('Fax', 'NNP'], ('+44', 'VBD'], ('207', 'CD'], ('484', 'CD'], ('5100', 'CD'], ('Mobi', 'NNP'], ('+44', 'VBD'], ('77', 'CD'], ('33', 'CD'], ('261', 'CD'], ('268', 'CD'], ('Stephanie.trabka', 'NN'], ('@', 'CD'], ('intcx.com', 'NN']]
3	Fwd: NYTimes.com Article: Suspended Rabbi Quits Seminary Presidency >Sender: articles-email@mst.tga2.nytimes.com >Reply-To: judgergm@swbell.com >From: judgergm@swbell.com >To: klarnoldi@flash.net	[['Fwd', 'NN'], (':', ':'], ('NYTimes.com', 'JJ'], ('Article', 'NN'], (':', ':'], ('Suspended', 'VBN'], ('Rabbi', 'NNP'], ('Quits', 'NNP'], ('Seminary', 'NNP'], ('Presidency', 'NNP'], ('>', 'NNP'], ('Sender', 'NNP'], (':', ':'], ('articles-email', 'JJ'], ('@', 'NNP'], ('mst.tga2.nytimes.com', 'NN'], ('>', 'VBD'], ('Reply-To', 'NNP'], (':', ':'], ('judgergm', 'NN'], ('@', 'NNP'], ('swbell.com', 'NN'], ('>', 'NN'], ('From', 'IN'], (':', ':'], ('judgergm', 'NN'], ('@', 'NNP'], ('swbell.com', 'NN'], ('>', 'NN'], ('To', 'TO'], (':', ':'], ('klarnoldi', 'VB'], ('@', 'CD'], ('flash.net', 'NN']]
4	daily charts and matrices as hot links 5/15 The information contained herein is based on sources that we believe to be reliable, but we do not represent that it is accurate or complete. Nothing contained herein should be considered as an offer to sell or a solicitation of an offer to buy any financial instruments discussed herein. Any opinions expressed herein are solely those of the author. As such, they may differ in material respects from those of, or expressed or published by on behalf of Carr Futures or its officers, directors, employees or affiliates. 2001 Carr Futures The charts are now available on the web by clicking on the hot link(s) contained in this email. If for any reason you are unable to receive the charts via the web, please contact me via email and I will email the charts to you as attachments. Editable and unloaded charts to follow. Crude <URL> Natural Gas <URL> Nat Gas Strip Matrix <URL> Nat Gas Spread Matrix <URL> Crude and Products Spread Matrix <URL> Carr Futures 150 S. Wacker Dr., Suite 1500 Chicago, IL 60606 USA Tel: 312-368-6149 Fax: 312-368-2281 seblander@carrfut.com <URL>	[['daily', 'JJ'], ('charts', 'NNS'], ('and', 'CC'], ('matrices', 'NNS'], ('as', 'IN'], ('hot', 'JJ'], ('links', 'NNS'], ('5/15', 'CD'], ('The', 'DT'], ('information', 'NN'], ('contained', 'VBN'], ('herein', 'NN'], ('is', 'VBZ'], ('based', 'VBN'], ('on', 'IN'], ('sources', 'NNS'], ('that', 'IN'], ('we', 'PRP'], ('believe', 'VBP'], ('to', 'TO'], ('be', 'VB'], ('reliable', 'JJ'], (',', ','], ('but', 'CC'], ('we', 'PRP'], ('do', 'VBP'], ('not', 'RB'], ('represent', 'VB'], ('that', 'IN'], ('it', 'PRP'], ('is', 'VBZ'], ('accurate', 'JJ'], ('or', 'CC'], ('complete', 'JJ'], ('.', '.'], ('Nothing', 'NN'], ('contained', 'VBD'], ('herein', 'NN'], ('should', 'MD'], ('be', 'VB'], ('considered', 'VBN'], ('as', 'IN'], ('an', 'DT'], ('offer', 'NN'], ('to', 'TO'], ('sell', 'VB'], ('or', 'CC'], ('a', 'DT'], ('solicitation', 'NN'], ('of', 'IN'], ('an', 'DT'], ('offer', 'NN'], ('to', 'TO'], ('buy', 'VB'], ('any', 'DT'], ('financial', 'JJ'], ('instruments', 'NNS'], ('discussed', 'VBN'], ('herein', 'RB'], (',', ','], ('Any', 'JJ'], ('opinions', 'NNS'], ('expressed', 'VBN'], ('herein', 'NNS'], ('are', 'VBP'], ('solely', 'RB'], ('those', 'DT'], ('of', 'IN'], ('the', 'DT'], ('author', 'NN'], (',', ','], ('As', 'IN'], ('such', 'JJ'], (',', ','], ('they', 'PRP'], ('may', 'MD'], ('differ', 'VB'], ('in', 'IN'], ('material', 'JJ'], ('respects', 'NNS'], ('from', 'IN'], ('those', 'DT'], ('of', 'IN'], (',', ','], ('or', 'CC'], ('expressed', 'VBD'], ('or', 'CC'], ('published', 'VBN'], ('by', 'IN'], ('on', 'IN'], ('behalf', 'NN'], ('of', 'IN'], ('Carr', 'NNP'], ('Futures', 'NNPS'], ('or', 'CC'], ('its', 'PRP$'], ('officers', 'NNS'], (',', ','], ('directors', 'NNS'], (',', ','], ('employees', 'NNS'], ('or', 'CC'], ('affiliates', 'NNS'], ('.', '.'], ('2001', 'CD'], ('Carr', 'NNP'], ('Futures', 'NNPS'], ('The', 'DT'], ('charts', 'NNS'], ('are', 'VBP'], ('now', 'RB'], ('available', 'JJ'], ('on', 'IN'], ('the', 'DT'], ('web', 'NN'], ('by', 'IN'], ('clicking', 'VBG'], ('on', 'IN'], ('the', 'DT'], ('hot', 'JJ'], ('link', 'NN'], ('(', '('], ('s', 'PRP'], (')', ')'], ('contained', 'VBN'], ('in', 'IN'], ('this', 'DT'], ('email', 'NN'], (',', ','], ('If', 'IN'], ('for', 'IN'], ('any', 'DT'], ('reason', 'NN'], ('you', 'PRP'], ('are', 'VBP'], ('unable', 'JJ'], ('to', 'TO'], ('receive', 'VB'], ('the', 'DT'], ('charts', 'NNS'], ('via', 'IN'], ('the', 'DT'], ('web', 'NN'], (',', ','], ('please', 'VB'], ('contact', 'VB'], ('me', 'PRP'], ('via', 'IN'], ('email', 'NN'], ('and', 'CC'], ('I', 'PRP'], ('will', 'MD'], ('email', 'VB'], ('the', 'DT'], ('charts', 'NNS'], ('to', 'TO'], ('you', 'PRP'], ('as', 'IN'], ('attachments', 'NNS'], (',', ','], ('Editable', 'NNP'], ('and', 'CC'], ('unloaded', 'VBD'], ('charts', 'NNS'], ('to', 'TO'], ('follow', 'VB'], (':', ':'], ('Crude', 'NNP'], ('<', 'NNP'], ('URL', 'NNP'], ('>', 'NNP'], ('Natural', 'NNP'], ('Gas', 'NNP'], ('<', 'NNP'], ('URL', 'NNP'], ('>', 'NNP'], ('Nat', 'NNP'], ('Gas', 'NNP'], ('Strip', 'NNP'], ('Matrix', 'NNP'], ('<', 'NNP'], ('URL', 'NNP'], ('>', 'NNP'], ('Nat', 'NNP'], ('Gas', 'NNP'], ('Spread', 'NNP'], ('Matrix', 'NNP'], ('<', 'NNP'], ('URL', 'NNP'], ('>', 'NNP'], ('Crude', 'NNP'], ('and', 'CC'], ('Products', 'NNPS'], ('Spread', 'NNP'], ('Matrix', 'NNP'], ('<', 'NNP'], ('URL', 'NNP'], ('>', 'NNP'], ('Carr', 'NNP'], ('Futures', 'NNP'], ('150', 'CD'], ('S.', 'NNP'], ('Wacker', 'NNP'], ('Dr.', 'NNP'], (',', ','], ('Suite', 'NNP'], ('1500', 'CD'], ('Chicago', 'NNP'], (',', ','], ('IL', 'NNP'], ('60606', 'CD'], ('USA', 'NNP'], ('Tel', 'NNP'], (':', ':'], ('312-368-6149', 'JJ'], ('Fax', 'NN'], (':', ':'], ('312-368-2281', 'JJ'], ('seblander', 'NN'], ('@', 'NNP'], ('carrfut.com', 'NN'], ('<', 'NNP'], ('URL', 'NNP'], ('>', 'NN']]
6	RE: Resume John, Thanks for the email and the offer to route my resume in the most expeditious way. I saw only one job that was posted on Enron's website that looked like a good fit. It was an asset evaluation and integration job for Enron North America. job 0000106886 My primary strengths are: Natural gas asset acquisition and management Supporting a trading organization by providing excellent futures and derivatives execution Trading proprietary books (fixed price or otherwise)where there is a competitive reason to be in the market, i.e. transportation spreads, storage, physical presence Good customer relations/account development Ability to focus on and analyze a wide range of problems/opportunities Strong negotiator I know that Enron is a big, happening place and it is invaluable to have someone like you to help me sort out the opportunities. Where I am coming from is that I have to more valuable to Enron than I am to Cey because Cey is doing frustratingly little to leverage itself and Enron is setting every trend imaginable in leveraging itself. I appreciate any help you can provide. Thanks. Alan To: Alan_Ibell@agcy.com	[['RE', 'NN'], (':', ':'], ('Resume', 'NNP'], ('John', 'NNP'], (',', ','], ('Thanks', 'NNP'], ('for', 'IN'], ('the', 'DT'], ('email', 'NN'], ('and', 'CC'], ('the', 'DT'], ('offer', 'NN'], ('to', 'TO'], ('route', 'VB'], ('my', 'PRP$'], ('resume', 'NN'], ('in', 'IN'], ('the', 'DT'], ('most', 'RBS'], ('expeditious', 'JJ'], ('way', 'NN'], ('.', '.'], ('I', 'PRP'], ('saw', 'VBD'], ('only', 'RB'], ('one', 'CD'], ('job', 'NN'], ('that', 'WDT'], ('was', 'VBD'], ('posted', 'VBN'], ('on', 'IN'], ('Enron', 'NNP'], ("'s", 'POS'], ('website', 'NN'], ('that', 'IN'], ('looked', 'VBD'], ('like', 'IN'], ('a', 'DT'], ('good', 'JJ'], ('fit', 'NN'], ('.', '.'], ('It', 'PRP'], ('was', 'VBD'], ('an', 'DT'], ('asset', 'NN'], ('evaluation', 'NN'], ('and', 'CC'], ('integration', 'NN'], ('job', 'NN'], ('for', 'IN'], ('Enron', 'NNP'], ('North', 'NNP'], ('America', 'NNP'], ('.', '.'], ('job', 'NN'], ('0000106886', 'CD'], ('My', 'PRP$'], ('primary', 'JJ'], ('strengths', 'NNS'], ('are', 'VBP'], (':', ':'], ('Natural', 'NNP'], ('gas', 'NN'], ('asset', 'NN'], ('acquisition', 'NN'], ('and', 'CC'], ('management', 'NN'], ('Supporting', 'VBG'], ('a', 'DT'], ('trading', 'NN'], ('organization', 'NN'], ('by', 'IN'], ('providing', 'VBG'], ('excellent', 'JJ'], ('futures', 'NNS'], ('and', 'CC'], ('derivatives', 'NNS'], ('execution', 'VBP'], ('Trading', 'VBG'], ('proprietary', 'JJ'], ('books', 'NNS'], ('(', '('], ('fixed', 'VBN'], ('price', 'NN'], ('or', 'CC'], ('otherwise', 'NN'], (')', ')'], ('where', 'VRB'], ('there', 'EX'], ('is', 'VBZ'], ('a', 'DT'], ('competitive', 'JJ'], ('reason', 'NN'], ('to', 'TO'], ('be', 'VB'], ('in', 'IN'], ('the', 'DT'], ('market', 'NN'], (',', ','], ('i.e', 'NN'], (',', ','], ('transportation', 'NN'], ('spreads', 'NNS'], (',', ','], ('storage', 'NN'], (',', ','], ('physical', 'JJ'], ('presence', 'NN'], ('Good', 'NNP'], ('customer', 'NN'], ('relations/account', 'NN'], ('development', 'NN'], ('Ability', 'NNP'], ('to', 'TO'], ('focus', 'VB'], ('on', 'IN'], ('and', 'CC'], ('analyze', 'VB'], ('a', 'DT'], ('wide', 'JJ'], ('range', 'NN'], ('of', 'IN'], ('problems/opportunities', 'NNS'], ('Strong', 'NNP'], ('negotiator', 'NN'], ('I', 'PRP'], ('know', 'VBP'], ('that', 'DT'], ('Enron', 'NNP'], ('is', 'VBZ'], ('a', 'DT'], ('big', 'JJ'], (',', ','], ('happening', 'VBG'], ('place', 'NN'], ('and', 'CC'], ('it', 'PRP'], ('is', 'VBZ'], ('invaluable', 'JJ'], ('to', 'TO'], ('have', 'VB'], ('someone', 'NN'], ('like', 'IN'], ('you', 'PRP'], ('to', 'TO'], ('help', 'VB'], ('me', 'PRP'], ('sort', 'VB'], ('out', 'RP'], ('the', 'DT'], ('opportunities', 'NNS'], ('.', '.'], ('Where', 'WRB'], ('I', 'PRP'], ('am', 'VBP'], ('coming', 'VBG'], ('from', 'IN'], ('is', 'VBZ'], ('that', 'IN'], ('I', 'PRP'], ('have', 'VBP'], ('to', 'TO'], ('more', 'RBR'], ('valuable', 'JJ'], ('to', 'TO'], ('Enron', 'NNP'], ('than', 'IN'], ('I', 'PRP'], ('am', 'VBP'], ('to', 'TO'], ('Cey', 'NNP'], ('because', 'IN'], ('Cey', 'NNP'], ('is', 'VBZ'], ('doing', 'VBG'], ('frustratingly', 'RB'], ('little', 'JJ'], ('to', 'TO'], ('leverage', 'VB'], ('itself', 'PRP'], ('and', 'CC'], ('Enron', 'NNP'], ('is', 'VBZ'], ('setting', 'VBG'], ('every', 'DT'], ('trend', 'NN'], ('imaginable', 'JJ'], ('in', 'IN'], ('leveraging', 'VBG'], ('itself', 'PRP'], ('.', '.'], ('I', 'PRP'], ('appreciate', 'VBP'], ('any', 'DT'], ('help', 'NN'], ('you', 'PRP'], ('can', 'MD'], ('provide', 'VB'], ('.', '.'], ('Thanks', 'NNP'], ('.', '.'], ('Alan', 'NNP'], ('To', 'TO'], (':', ':'], ('Alan_Ibell', 'NNP'], ('@', 'NNP'], ('cey.com', 'NN']]
6	Wrong address John, I think I have the wrong email address for your brother. Please email me the correct one when you get a chance. Thanks, Brian _____ IMPORTANT NOTICES: This message is intended only for the addressee. Please notify the sender by e-mail if you are not the intended recipient. If you are not the intended recipient, you may not copy, disclose, or distribute this message or its contents to any other person and any such actions may be unlawful. Banc of America Securities LLC("BAS") does not accept time sensitive, action-oriented messages or transaction orders, including orders to purchase or sell securities, via e-mail. BAS reserves the right to monitor and review the content of all messages sent to or from this e-mail address. Messages sent to or from this e-mail address may be stored on the BAS e-mail system.	[['Wrong', 'JJ'], ('address', 'NN'], ('John', 'NNP'], (',', ','], ('I', 'PRP'], ('think', 'VBP'], ('I', 'PRP'], ('have', 'VBP'], ('the', 'DT'], ('wrong', 'JJ'], ('email', 'NN'], ('address', 'NN'], ('for', 'IN'], ('your', 'PRP$'], ('brother', 'NN'], (',', ','], ('Please', 'NNP'], ('email', 'VB'], ('me', 'PRP'], ('the', 'DT'], ('correct', 'JJ'], ('one', 'CD'], ('when', 'WRB'], ('you', 'PRP'], ('get', 'VBP'], ('a', 'DT'], ('chance', 'NN'], (',', ','], ('Thanks', 'NNS'], (',', ','], ('Brian', 'NNP'], ('_____', 'NNP'], ('IMPORTANT', 'NNP'], ('NOTICES', 'NNP'], (':', ':'], ('This', 'DT'], ('message', 'NN'], ('is', 'VBZ'], ('intended', 'VBN'], ('only', 'RB'], ('for', 'IN'], ('the', 'DT'], ('addressee', 'NN'], ('.', '.'], ('Please', 'NNP'], ('notify', 'VB'], ('the', 'DT'], ('sender', 'NN'], ('by', 'IN'], ('e-mail', 'JJ'], ('if', 'IN'], ('you', 'PRP'], ('are', 'VBP'], ('not', 'RB'], ('the', 'DT'], ('intended', 'JJ'], ('recipient', 'NN'], (',', ','], ('if', 'IN'], ('you', 'PRP'], ('are', 'VBP'], ('not', 'RB'], ('the', 'DT'], ('intended', 'JJ'], ('recipient', 'NN'], (',', ','], ('you', 'PRP'], ('may', 'MD'], ('not', 'RB'], ('copy', 'VB'], (',', ','], ('disclose', 'VB'], (',', ','], ('or', 'CC'], ('distribute', 'VB'], ('this', 'DT'], ('message', 'NN'], ('or', 'CC'], ('its', 'PRP$'], ('contents', 'NNS'], ('to', 'TO'], ('any', 'DT'], ('other', 'JJ'], ('person', 'NN'], ('and', 'CC'], ('any', 'DT'], ('such', 'JJ'], ('actions', 'NNS'], ('may', 'MD'], ('be', 'VB'], ('unlawful', 'JJ'], (',', ','], ('Banc', 'NNP'], ('of', 'IN'], ('America', 'NNP'], ('Securities', 'NNP'], ('LLC', 'NNP'], ('("', '('], ('"', '"'], ('BAS', 'NNP'], ('"', '"'], (')', ')'], ('does', 'VBZ'], ('not', 'RB'], ('accept', 'VB'], ('time', 'NN'], ('sensitive', 'JJ'], (',', ','], ('action-oriented', 'JJ'], (',', ','], ('messages', 'NNS'], ('or', 'CC'], ('transaction', 'NN'], ('orders', 'NNS'], (',', ','], ('including', 'VBG'], ('orders', 'NNS'], ('to', 'TO'], ('purchase', 'VB'], ('or', 'CC'], ('sell', 'VB'], ('securities', 'NNS'], (',', ','], ('via', 'IN'], ('e-mail', 'NN'], (',', ','], ('BAS', 'NNP'], ('reserves', 'NNS'], ('the', 'DT'], ('right', 'NN'], ('to', 'TO'], ('monitor', 'VB'], ('and', 'CC'], ('review', 'VB'], ('the', 'DT'], ('content', 'NN'], ('of', 'IN'], ('all', 'DT'], ('messages', 'NNS'], ('sent', 'VBN'], ('to', 'TO'], ('or', 'CC'], ('from', 'IN'], ('this', 'DT'], ('e-mail', 'JJ'], ('address', 'NN'], (',', ','], ('Messages', 'NNP'], ('sent', 'VBD'], ('to', 'TO'], ('or', 'CC'], ('from', 'IN'], ('this', 'DT'], ('e-mail', 'JJ'], ('address', 'NN'], ('may', 'MD'], ('be', 'VB'], ('stored', 'VBN'], ('on', 'IN'], ('the', 'DT'], ('BAS', 'NNP'], ('e-mail', 'NN'], ('system', 'NN'], ('.', '.']]

12. Generative AI tools usage:

For this project, ChatGPT and Github Copilot was utilised for brainstorming and coming up with boilerplate code for our models. Zotero was used to collect and generate APA citations.

13. References

Singapore Police Force (SPF). (n.d.). Police News Release: Mid-Year Crime Statistics.Retrieved from Singapore Police Force: https://www.police.gov.sg/media-room/statistics

Channel News Asia (CNA). (2022). Crime levels, scams rise in 2021. Retrieved from https://www.channelnewsasia.com/singapore/crime-levels-scams-rise-2021-2501736

The Straits Times. (2024). At least 83 victims have lost $155,000 in DBS phishing scam since the start of January. Retrieve from https://www.straitstimes.com/singapore/at-least-83-victims-lose-155000-in-dbs-phishing-scam-since-the-start-of-january

Buller, D. B., & Burgoon, J. K. (1996). Interpersonal Deception Theory. Communication Theory, 6(3), 203–242. https://doi.org/10.1111/j.1468-2885.1996.tb00127.x

Witte, E. H. (2007). Toward a Group Facilitation Technique for Project Teams. Group Processes & Intergroup Relations, 10(3), 299-309. https://doi-org.libproxy.smu.edu.sg/10.1177/1368430207078694

BURGOON, J. K., BULLER, D. B., FLOYD, K., & GRANDPRE, J. (1996). Deceptive Realities: Sender, Receiver, and Observer Perspectives in Deceptive Conversations. Communication Research, 23(6), 724-748. https://doi.org/10.1177/009365096023006005

Mbaziira, A., & Jones, J. (2016). A Text-Based Deception Detection Model for Cybercrime. International Conference on Technology and Management

Zheng, R., Qin, Y., Huang, Z., & Chen, H. (2003). Authorship analysis in cybercrime investigation. Hong Kong University of Science and Technology, 2665, 59–73. https://doi.org/10.1007/3-540-44853-5_5

Meyer, T. A., & Whateley, B. (2004). SpamBayes: Effective open-source, Bayesian based, email classification system. In CEAS.

Kolcz, A., Chowdhury, A., & Alspector, J. (2004). The impact of feature selection on signature-driven spam detection. In Proceedings of the 1st Conference on Email and Anti-Spam (CEAS-2004).

Lee, S., Shafqat, W., & Kim, H.C. (2022). Backers Beware: Characteristics and Detection of Fraudulent Crowdfunding Campaigns. Sensors (Basel, Switzerland), 22(19), 7677-. https://doi.org/10.3390/s22197677

Naudé, M., Adebayo, K. J., & Nanda, R. (2023). A machine learning approach to detecting fraudulent job types. AI & SOCIETY, 38(2), 1013-1024.

Rao, A. S. (n.d.). Enron Fraud Email Dataset. Retrieved from https://www.kaggle.com/datasets/advaithsrao/enron-fraud-email-dataset

Radev, D. (2008), CLAIR collection of fraud email, ACL Data and Code Repository, ADCR2008T001, http://aclweb.org/aclwiki