Project Midpoint
Term: Winter 2024
Team: Clark Morin, Madison Trimble, Trey Pezzetti, Julie Athanasiadis

Executive Summary:

In our group project, we will focus on leveraging machine learning and deep learning techniques to develop a categorical predictor that flags potential financial fraud in email communications. This project is designed to aid compliance officers by automatically sorting emails that may contain signs of financial fraud, requiring manual review. Our approach will involve significant preprocessing of a dataset to highlight the presence of common financial fraud keywords in the emails' subject and body.

We plan to manually label approximately 1000 emails to determine their potential for financial fraud, with an additional set labeled with the assistance of LLMs. The project will explore various machine learning models, including logistic regression, random forest classifier, gradient boosting classifier, and support vector classifier. Our objective is to evaluate these models' effectiveness in identifying potentially fraudulent communications and discuss the applicability of our findings to other companies.

By focusing on this specific application of data science, our project will provide insights into the practical use of machine learning for enhancing corporate compliance processes. We aim to demonstrate a clear, impactful application of data science skills, supported by thorough analysis and evaluation of our models' performance.

Problem Statement:

In the context of increasing incidents of financial fraud, identifying potential fraud in corporate communications, such as emails, is a critical challenge for compliance officers. Our project aims to develop a machine learning model that can flag emails potentially related to financial fraud, thereby facilitating a more efficient review process.

Research Objectives:

1. To preprocess the email dataset by removing irrelevant columns and creating a new column that highlights the presence of financial fraud keywords in the subject and body of the emails
2. To manually label approximately 1000 emails for the presence of potential financial fraud, supplemented by an additional 1000 emails labeled with the assistance of ChatGPT
3. To train, test, and hyper-tune various machine learning models, including KNN, Decision Trees, and a deep learning model, on the labeled dataset
4. To evaluate the performance of these models in correctly identifying emails that may contain financial fraud

5. To discuss the potential of our model as a tool for compliance officers in various companies, contributing to the broader application of machine learning in corporate governance and ethics.

Approach and Methodology:

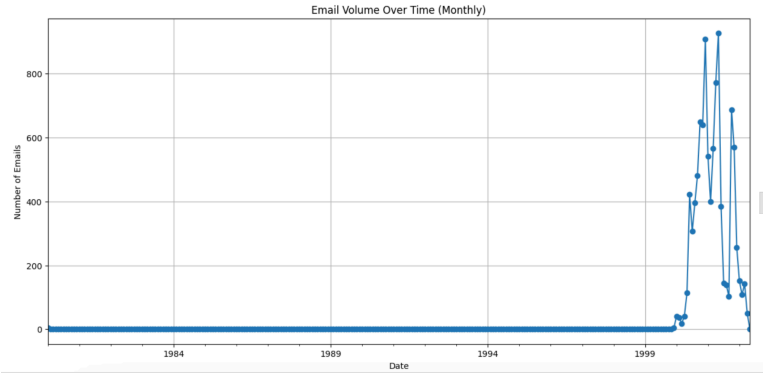
Our project will now incorporate an end-to-end machine learning pipeline and cross-validation design specifically tailored to detect potential financial fraud in email communications. This includes data cleaning, feature engineering to identify keywords related to financial fraud, and the development of machine learning models suited for categorical prediction. We will focus on the practical application of these models in a real-world setting, providing compliance officers with a valuable tool for email review processes. The project's success will be measured by the models' accuracy in flagging relevant communications and their potential applicability in a corporate context.

Data Preparation:

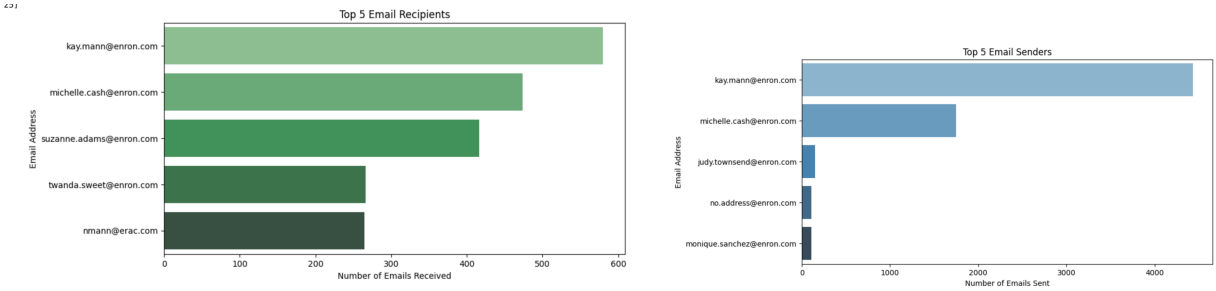
- Preprocess and Analyze Data: The script processes a subset of emails (up to 10,000) and applies preprocessing steps. This includes converting email bodies to a readable format and handling encoding issues
- Keyword Search: The script implements a search for specific keywords within the email bodies to identify potential indicators of financial fraud. A comprehensive list of financial fraud-related keywords is defined, and emails are scanned for the presence of these keywords
- Flag Potential Fraud Emails: A new column, Fraud_Flag, is created in the dataset to flag emails potentially related to financial fraud based on the presence of the defined keywords in the email's subject or body
- Email Domain and Length Analysis: Additional features such as the email domain, email length, and normalized email length are calculated and included in the dataset for further analysis
- Sentiment Analysis: The script performs sentiment analysis on the email bodies to evaluate the sentiment polarity, adding another layer of analysis to potentially identify suspicious emails
- Final Dataset Preparation: The final dataset is prepared by selecting relevant columns and calculating the number of emails flagged for potential fraud. The dataset is then ready for the machine learning model training phase

Exploratory Data Analysis (EDA):

1. Plotting Email Volume by Month Over Time: This involved aggregating the email data to calculate the volume of emails sent each month. A time series plot was created to visualize how email communication trends changed over time
    a. Findings: Our findings below aligned with the times Enron were at the peak of its existence. We were not able to find a ton of patterns as we only are looking at 1000 emails



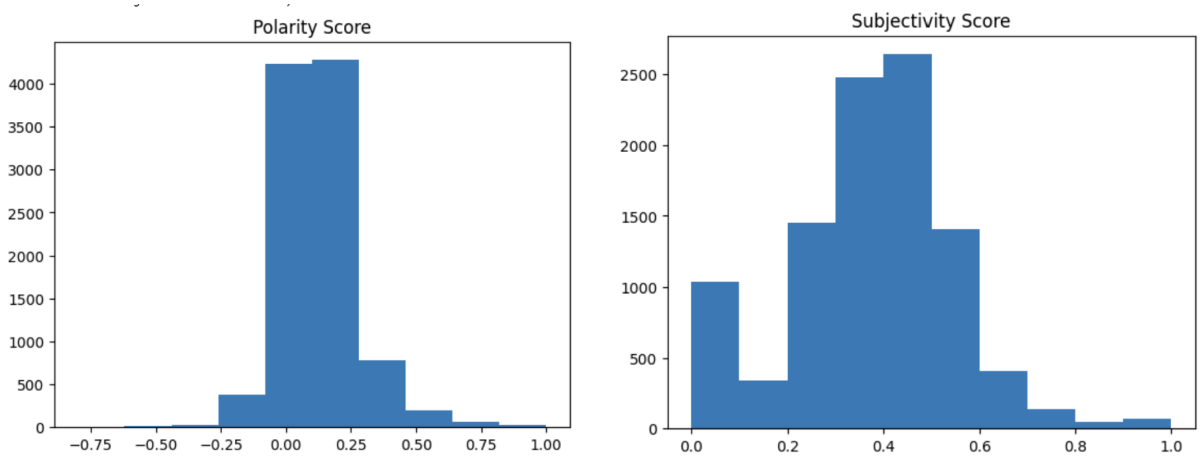
2. Top 5 Senders and Receivers of Emails: We identified the top 5 senders and receivers in the dataset by counting the number of emails sent and received by each participant. This analysis involved calculating the frequency of emails for each unique email address in the "From" and "To" fields and then ranking them to identify the most active participants.
    a. Findings: We found that kay.mann@enron.com sent and received the most email. Additionally, we found out that 'suzanne.adams@enron.com <--> kay.mann@enron.com' exchanged 435 emails which was the most by any pairin our dataset. Suzanne worked in the legal department and we were unable to find Kay Mann's role.



3. Sentiment Analysis via TextBlob: The project utilized TextBlob to conduct sentiment analysis on the emails, generating both polarity and subjectivity scores. This analysis helped find potential trends within the text that may indicate underlying sentiments affecting the organization's internal culture
    a. Polarity Score Analysis:
        i. Definition and Range: Polarity scores are floating-point numbers ranging from -1.0 to 1.0, where scores below -0.5 are deemed negative, scores above 0.5 are considered positive, and scores in between are viewed as neutral
        ii. Findings: Initial analysis showed that most emails have polarity scores between -0.25 and 0.5, suggesting a predominantly neutral sentiment. However, emails with scores below -0.5, indicating negative sentiment,

could be critical for identifying communications that might negatively impact internal culture

b. Subjectivity Score Analysis:
    i. Definition and Range: Subjectivity scores also float between 0.0 and 1.0. Scores closer to 0.0 are more objective, while scores nearing 1.0 are more subjective
    ii. Findings: The initial subjectivity analysis indicated that most emails fall between 0.2 and 0.8 in subjectivity, pointing to a moderate level of subjectivity in communications. Notably, around 1000 emails were found to be completely objective. Emails with subjectivity scores above 0.8, marking them as highly subjective, present an interesting subset for further review due to their potential implications for organizational culture



Feature Engineering/Variable Transformations:

- Extraction of Key Email Attributes: The script defines a function to parse each email and extract essential attributes such as the "From" (sender), "To" (recipient), "Subject", "Date", and "Body". This step transforms email data into a structured format suitable for analysis
- Creation of Fraud_Flag Feature:
    - A critical feature engineering step is the creation of the Fraud_Flag column. This binary feature indicates whether an email potentially relates to financial fraud, determined by the presence of specific keywords related to financial fraud in the email's subject or body
    - The list of keywords includes terms like "Fraud", "Embezzlement", "Bribery", etc.
    - The script checks each email for these keywords and flags those containing any of them, facilitating targeted analysis and modeling

- Email Domain Extraction: The script performs a transformation to extract the domain from each email address in the dataset. This categorical feature can provide insights into the communication patterns and identify frequent domains that may warrant closer examination.
- Email Length Calculation: Another feature engineered is the calculation of email length, determined by the number of characters in the email's body. This numeric feature could help identify outliers, such as exceptionally long or short emails that might be of interest in the context of fraud detection.
- Normalization of Email Length: To standardize the email length feature for model training, the script applies normalization to this variable. Normalization ensures that the email length is on a similar scale as other numeric features in the dataset, improving model performance and interpretability.
- Sentiment Polarity Score: Sentiment analysis is conducted on the email bodies, resulting in a sentiment polarity score for each email. This score is a continuous variable that measures the sentiment tone of the email, ranging from negative to positive. It's used to potentially highlight emails with extreme sentiments that could be related to fraudulent activities.

Methodology

- Tools and Environment:
    - Python was the primary programming language chosen for coding and executing our models. Python offers extensive libraries, versatility, and readability necessary for coding algorithms and collaboration. The decision to utilize a cloud-based coding environment encouraged the team to collaborate on code development in real-time and allowed for simultaneous access to the updated code. Google Colab provided access to GPU resources for faster model training and execution. The default GPU for Colab is the NVIDIA Tesla K80 which provides 12GB of VRAM.
- Data Preprocessing and Feature Engineering:
    - Pandas and Numpy were utilized for preprocessing of the contents of the data set. Parsing the contents of the emails allowed for the extraction of relevant elements such as sender, recipient, and body. The email body text was cleaned by removing whitespaces and stop words like “and” and “the”. These processes allowed us to effectively handle encoding issues and convert emails to a more readable format, preparing them for analysis.
    - Feature engineering involved extracting the email domain from each email address to get insight into communication patterns within and outside of Enron. Calculation and normalization of the length of emails was also employed to optimize model performance analysis. NLTK and TextBlob libraries were employed to enable natural language processing through sentiment analysis.
- Model Selection and Implementation:
    - The Scikit-Learn library was employed to implement the machine learning models; Logistic Regression, Random Forest Classifier, Gradient Boosting

Classifier, and Support Vector Classifier. These models have their unique advantages; Logistic Regression is well suited for binary classification, Random Forest and Gradient Boosting both excel in capturing more complex data patterns, and the Support Vector Classifier's flexibility allows for the handling of more intricate patterns.

- Model Evaluation:
    - Model performance was measured using a variety of metrics including accuracy, precision, recall, confusion matrix, F1 score, and ROC AUC. The combination of these gave us a holistic view of how the various models handled the data sets.
        - Accuracy provided us with the ratio of correctly predicted cases amongst all predicted cases.
        - Precision identifies the ratio of correctly predicted positive results to all positive predictions.
        - Recall provides the ratio of correctly predicted positive observations to all observations in the positive class.
        - The confusion matrix takes into account how many true positive, false positive, true negative, and false negative predictions occur giving us a view of model performance across classes.
        - The F1 score is especially useful in a case where the two classes are imbalanced as it gives us a weighted average of the precision and recall for the models.
        - Receiver Operating Characteristic Area Under the Curve, or ROC AUC, provides a graphical representation of the models' ability to discriminate between the fraudulent and non fraudulent classes.

Findings and Conclusions

- Each of the models exhibited a very high level of accuracy in their performance. The linear regression model achieved 99.6%across the different cross-validation folds. The random forest model had an average accuracy of 99.8%. The highest performing model in terms of accuracy was the gradient boosting model with an astonishing 99.9%. Finally, the support vector classifier model achieves 99.7% percent accuracy. On top of the impressive accuracy results, each model has a standard deviation of 0.0 meaning that they all exhibited minimal variability.
- The confusion matrix using the Random Forest Classifier returned 6 true positives, 0 false positives, 1987 true negatives, and 7 false negatives. The precision ratio of 1.0 indicates that all predicted positive cases were indeed positive. Recall of 0.4615 shows that the model was able to determine 46.15% of actual fraudulent emails. An F1 score of 0.6316 indicates a relatively balanced measure of precision and recall. FInally the ROC AUC score of 0.9966 suggests that the model has high discriminatory power to distinguish between positive (fraudulent) and negative (not fraudulent) classes.
- The scores listed above indicate that the models can effectively generalize from the dataset. The decision regarding which model to deploy could come down to the combined value of accuracy, interpretability, and computational efficiency.

Lessons Learned and Recommendations

- Labeling the data more effectively could unlock additional opportunities for exploration such as enhanced model understanding, fine-tuning performance, and addressing imbalances between classes.
- This approach is very effectively utilized in email datasets. The use of the methods in this particular study do not extend to telephone conversations, in-person communication, or physical documents.
- Third party datasets that can add value to the existing analysis would ideally provide additional context or complementary information to the problem. This could include data concerning compliance, customer feedback, legal proceedings, etc.

Literature Review

In the article by E.G. Dada and S.B. Joseph entitled *Random Forests Machine Learning Technique for Email Spam Filtering* the authors discuss the process by which they could effectively identify and predict spam emails within the Enron dataset. The study employs a content-based filtering approach that is used to identify certain keywords that may be frequently associated with phishing or spam within emails within the dataset. The frequency at which these keywords, or features, appeared in an email was measured against a predetermined threshold value and determined the likelihood that each of them contained spam (Dada and Joseph 2018). The authors chose the Random Forests algorithm to achieve feature extraction and classification. This model's performance was evaluated using 10-fold cross validation and achieved an accuracy of 99.92% and an error rate of 0.0296% (Dada and Joseph 2018).

In a 2016 study by Shrawan Kumar Trivedi, multiple machine learning tools were employed to determine which performed best in classifying emails based on the presence of certain features. Email files were extracted from the database and transformed by removing HTML tags, stop words, and converting words to their base form, also known as lemmatization, to create a feature dictionary (Trivedi 2016). The author selected the machine learning classifiers Bayesian, Naive Bayes, Support Vector, and Decision Tree. Each model's performance was measured by accuracy, false positive rate, and training time. Support Vector achieved the lowest false positive rate at 6.5% and the highest accuracy at 93.3% (Trivedi 2016).

References

Dada, E.G., and S.B. Joseph. *Random Forests Machine Learning Technique for Email Spam Filtering* 9, no. 1 (July 2018). https://www.researchgate.net/profile/Emmanuel-Dada/publication/327720311_Random_Forests_Machine_Learning_Technique_for_Email_Spam_Filtering/links/5ba0b0c0299bf13e6038ecc1/Random-Forests-Machine-Learning-Technique-for-Email-Spam-Filtering.pdf.

Trivedi, Shrawan Kumar. “A Study of Machine Learning Classifiers for Spam Detection.” *2016 4th International Symposium on Computational and Business Intelligence (ISCBI)*, September 2016. https://doi.org/10.1109/iscbi.2016.7743279.