Enron Email Analysis:

What happened inside Enron?

IND5003 – Group: Beef Don
Group Member:
Li Jia A0309977H
Lei Ruoer A0309989B
Su WanxiA0309960X
Zhao Haiyao A0309952W
Li Yanfeng A0309986H

Abstract

During Enron scandal, most emails from senior managers revealed many of them have already know the act of the company is already against law. By conducting an unsupervised learning, we wish to figure out who have a higher chance to be a criminal and who have strong connection with them. By first analyzing the data, we can discover what business the company is running, how often do they use email. In this paper, we first introduce the Enron dataset and the company background. Secondly, we share review existing code for analysis Enron dataset and compared to them. Later, we conduct our own topic modeling using machine learning technology through K-Means clustering and sentiment analysis. Then, we predict the crime based on our previous methodologies and compare with the real legal consequences. Finally, we draw the conclusion for our analysis on the dataset and discuss limits by using our purposed methods.

Key Words: Enron dataset, K-Means, data analysis, sentiment analysis, machine learning

Introduction

The Enron email dataset contains 500,000+ emails from 150 employees of the Enron Corporation [1]. Enron Corporation was once an energy company located in Houston, Texas, USA. The company was selected as "America's Most Innovative Company" by Fortune magazine for six consecutive years [2]. However, what really made Enron famous around the world was the bankruptcy of this company with assets of hundreds of billions in a few weeks in 2002, and the financial fraud scandal that had been carefully planned, institutionalized and systematic for many years.

Current dataset has lots of words containing financial characteristics, such as "‘salary’, ‘deferral_payments’, ‘bonus’". Those words also fit what company have done previously: providing high salary to its employees, earning money that are against law, loaning money and calculating long-term project profit into one year to make the annual financial statements look nice. However, Andrew Fastow (CFO of the company) has destroyed, tamper, or fabricate financial records for a long time which finally leads to company breakdown. Some of the original data have been destroyed, the best way to figure out who involves in this criminal is to view the email, analyzing connections and potential high-risk people to make sure who is quality and penalize them.

In this report, we will first contrast with others' papers and go through their methods review. Then we clean and process the data, analyzing the data characteristics. Afterwards, we share the overview of code structure by demonstrating our machine learning models and do the sentiment analysis based on this NLP task. By combining all the features we have discovered, we can predict the criminal from this Enron scandal. Based on our model's limitations, we will share our insights and hope to shed light on future directions for using unsupervised learning technique dealing with Enron dataset. Finally, we attach with conclusion. The source code can be seen in https://github.com/Lijiateenie/ind5003---group-project---enron-email .

Methods Review

To detect who is the criminal, there are a few choices of machine learning algorithms (e.g., Gaussian Naïve-Bayes, Decision Tree Classifier, Support Vector Machine and so on). [3] have already discovered the best parameters for the Gaussian Naïve-Bayes.

```
SELECTOR__K = [15], REDUCER__N_COMPONENTS = [6]
```

[4] have also tried Support Vector Machine technique, using the parameters as bellows:

```
SELECTOR__K = [18], REDUCER__N_COMPONENTS = [10], C_PARAM = [100], GAMMA_PARAM = [.01]
CLASS_WEIGHT = ['balanced'], KERNEL = ['sigmoid']
```

The best features they have been selected using NB are as follows:

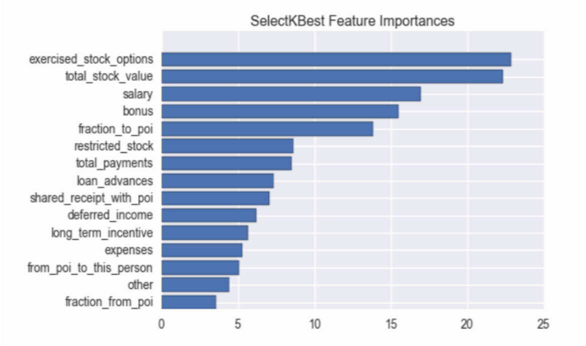


Fig 1: select K best features using Naive Bayes Methods

The top four features are: exercised_stock_options, total_stock_value, salary as well as bonus. According to the news report, Enron company is known as the high salary and high bonus company, advocating "wolf culture", eliminating people who are the last 10% each year. People tend to be work harder and only cares of the profit and benefits to the company, instead of right or wrong. One thing that is interesting regarding to the bonus is that CFO (by the age of 25) for annual bonus only, already got $5,000,000. Around 2001, the company's performance is declining, but the boss still expects everyone to buy stocks, deceiving everyone that the stocks will appreciate in the future, to obtain a large amount of cash and escape. Therefore, by simply selecting features already can verify the facts.

According to the [5], they did the search from social network aspects and discovered that employees who have already disconnected suddenly have mutual communication through emails with Enron company during crisis period. Employees who have not been in contact with the company for a long time use private communication tools, bypassing formal chains of communication. In Fig 2, job title hierarchy has been discovered by analyzing back and forth emails.

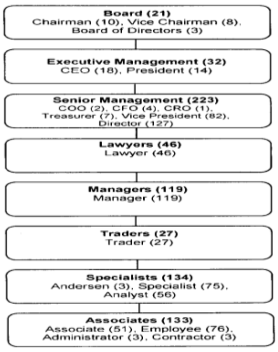
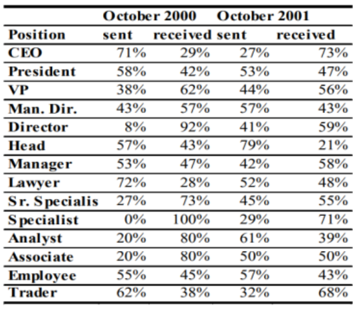

Board (21)
Chairman (10), Vice Chairman (8), Board of Directors (3)
Executive Management (32)
CEO (18), President (14)
Senior Management (223)
COO (2), CFO (4), CRO (1), Treasurer (7), Vice President (82), Director (127)
Lawyers (46)
Lawyer (46)
Managers (119)
Manager (119)
Traders (27)
Trader (27)
Specialists (134)
Andersen (3), Specialist (75), Analyst (56)
Associates (133)
Associate (51), Employee (76), Administrator (3), Contractor (3)

Fig 2. Hierarchy of the company based on emails

In Jana's work [6], they conclude emails exchange per month from 2000 to 2001 (during crisis), calculating and summarizing for each position, how many percent has been sent and received. CEO are sending most of the emails for informing the nearest or stakeholders about crisis in 2000, while in 2001 he sends small volume of emails and most of the time only receive emails. The major shift from 2000 to 2001 is that in 2000 higher rank positions tended to be directive (send more than receive) whereas by 2001 they became consumers (receive more than send).



	October 2000		October 2001	
Position	**sent**	**received**	**sent**	**received**
CEO	71%	29%	27%	73%
President	58%	42%	53%	47%
VP	38%	62%	44%	56%
Man. Dir.	43%	57%	57%	43%
Director	8%	92%	41%	59%
Head	57%	43%	79%	21%
Manager	53%	47%	42%	58%
Lawyer	72%	28%	52%	48%
Sr. Specialis	27%	73%	45%	55%
Specialist	0%	100%	29%	71%
Analyst	20%	80%	61%	39%
Associate	20%	80%	50%	50%
Employee	55%	45%	57%	43%
Trader	62%	38%	32%	68%

Fig 3. sent and received percentage for 2000 and 2001

[7] did a case study on Enron Corpus, testing the formality being affected by social distance, relative power and other factors. The formality classifier they used can be also transferred to other dataset to test other linguistic or social theories. Personal and business email can be differentiated via analyzing social distance. However, additional features can be added (layout and zoning) to better identify social linkage.

Statement of Questions of Interest

In this project, we aim to investigate key issues related to the Enron scandal using unsupervised learning techniques. The following are the main research questions:

1. Can we identify **potential criminals** or suspicious individuals from the Enron email dataset?

By constructing a network graph based on the sender and receiver relationships in emails, we will explore individuals with close communication ties within the company, and analyze whether these communications are normal business interactions or abnormal activities. Using topic modeling (such as LDA or K-means) and clustering analysis, we will extract groups of emails with similar themes or content from a large volume of emails. By examining the distribution of keywords and topics, we hope to identify abnormal behaviors related to illegal activities.

2. Can we uncover **changes in the main content and emotional trends** of emails before and after **negative news emerging**?

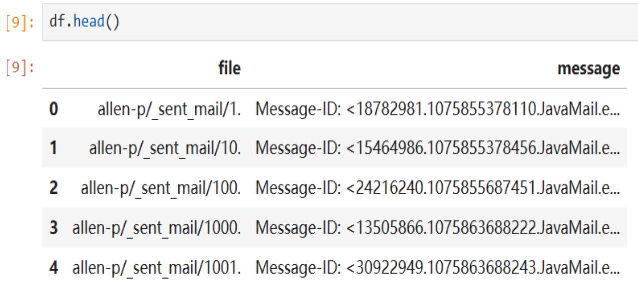
We will analyze the trends in email traffic, themes, and emotional tendencies over time, exploring whether executives took measures to soothe employee emotions during the crisis, and whether the content and emotions in employee emails changed during the crisis. We will pay particular attention to the fluctuations in the emotional tone of internal company emails before and after the scandal was exposed.

3. How did the contents or behaviors of **managers who want to reduce risk differ from aggressive managers** in the emails?

By comparing the communication patterns of employees before and after the crisis or find the abnormal point through the content, we will focus on the email activity characteristics of senior management such as CEO&CFO, including their frequency of sending and receiving emails, email themes, word frequency and emotional tendencies. Through these analyses, we hope to reveal the behavioral characteristics of aggressive managers during crisis management and compare them with those who want to reduce risk.

Data Overview

Before conducting data analysis on email dataset, we should first do data cleaning as well as pre-processing (Fig 4). The email in raw formats contains few useless information which might cause negative impacts on topic modeling analysis, therefore leads to failure in criminal prediction. Ideally, what we need are: date, from, to, subject, body and so on and so forth (Fig 5.).

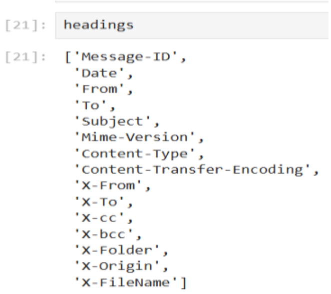


```
[9]: df.head()
```

[9]:

	file	message
0	allen-p/_sent_mail/1.	Message-ID: <18782981.1075855378110.JavaMail.e...
1	allen-p/_sent_mail/10.	Message-ID: <15464986.1075855378456.JavaMail.e...
2	allen-p/_sent_mail/100.	Message-ID: <24216240.1075855687451.JavaMail.e...
3	allen-p/_sent_mail/1000.	Message-ID: <13505866.1075863688222.JavaMail.e...
4	allen-p/_sent_mail/1001.	Message-ID: <30922949.1075863688243.JavaMail.e...

Fig 4. five first messages from datasets

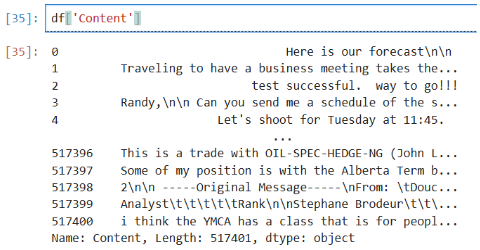


```
[21]: headings

[21]: ['Message-ID',
       'Date',
       'From',
       'To',
       'Subject',
       'Mime-Version',
       'Content-Type',
       'Content-Transfer-Encoding',
       'X-From',
       'X-To',
       'X-cc',
       'X-bcc',
       'X-Folder',
       'X-Origin',
       'X-FileName']
```

Fig 5. Separate parts from messages

We removed the data which has 'null' value, the final size of the dataset is (517401 rows × 18 columns). When we print the content from the message (from Fig 6), it consists of unwanted strings such as '\n' or '-----' or '\t'. So, we needs to remove those noise from data and get it cleaned.



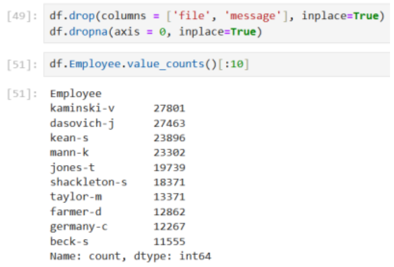
```
[35]: df['Content']

[35]: 0                                 Here is our forecast\n\n
      1         Traveling to have a business meeting takes the...
      2                              test successful.  way to go!!!
      3         Randy,\n\n Can you send me a schedule of the s...
      4                         Let's shoot for Tuesday at 11:45.
                                      ...
      517396    This is a trade with OIL-SPEC-HEDGE-NG (John L...
      517397    Some of my position is with the Alberta Term b...
      517398    2\n\n -----Original Message-----\nFrom: \tDouc...
      517399    Analyst\t\t\t\t\tRank\n\nStephane Brodeur\t\t\...
      517400    i think the YMCA has a class that is for peopl...
      Name: Content, Length: 517401, dtype: object
```

Fig 6. Print out content to check the raw data

Data Analysis

We first count how many times are those names appears. Here we print out top 10 employees with most emails rolling back and forth, in Fig 7. We also count how many emails back and forth for each year or month. The result shows the most volume of using email is around year 2001 for around 256402 emails. For the most frequent date is 1980-01-01 with around 316 emails (the start of the company).



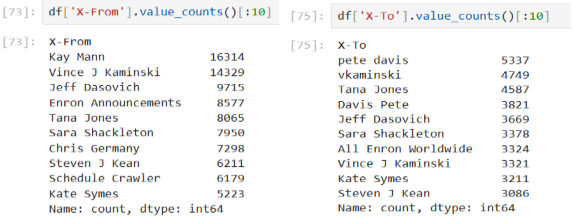
```
[49]: df.drop(columns = ['file', 'message'], inplace=True)
      df.dropna(axis = 0, inplace=True)

[51]: df.Employee.value_counts()[:10]

[51]: Employee
      kaminski-v      27801
      dasovich-j      27463
      kean-s          23896
      mann-k          23302
      jones-t         19739
      shackleton-s    18371
      taylor-m        13371
      farmer-d        12862
      germany-c       12267
      beck-s          11555
      Name: count, dtype: int64
```

Fig 7. count numbers of each employees appeared

We count through the variable of 'x-from' and 'x-to'. X-From means who sent the email, X-To means who receive the email. We ignored the 'cc' part when doing analysis, because 'cc' only means noticing, but not who are acting.



```
[73]: df['X-From'].value_counts()[:10]

[73]: X-From
      Kay Mann                  16314
      Vince J Kaminski          14329
      Jeff Dasovich              9715
      Enron Announcements        8577
      Tana Jones                 8065
      Sara Shackleton            7950
      Chris Germany              7298
      Steven J Kean              6211
      Schedule Crawler           6179
      Kate Symes                 5223
      Name: count, dtype: int64
```

```
[75]: df['X-To'].value_counts()[:10]

[75]: X-To
      pete davis                5337
      vkaminski                 4749
      Tana Jones                4587
      Davis Pete                3821
      Jeff Dasovich             3669
      Sara Shackleton           3378
      All Enron Worldwide       3324
      Vince J Kaminski          3321
      Kate Symes                3211
      Steven J Kean             3086
      Name: count, dtype: int64
```

Fig 8. count the values for 'x-from' and 'x-to'

Word Frequency is another dimension that we take into consideration. We first print out work frequency, get the result of most frequent word "('the', 4996338), ('to', 3505646), ('and', 2504108), ('of', 2322559), ('a', 1844691), ('in', 1594883)" which contains lots of stop-words. Therefore, we remove stopwords and print in wordcloud to check which words usually appears in emails (Fig 9).

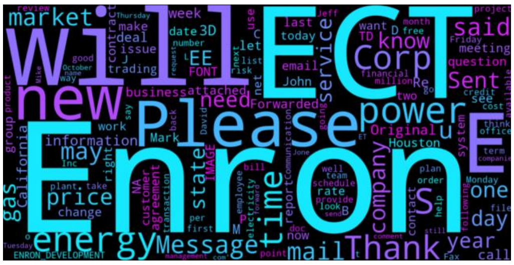


Fig 9. Word Cloud for emails

Overview of Code Structure

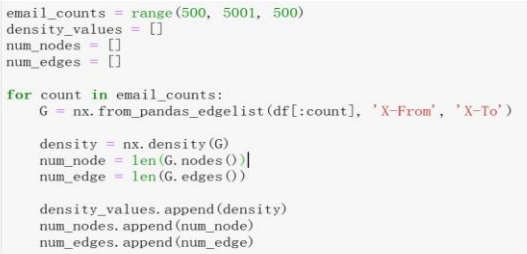
Network of Key Users

To investigate the relationships between email senders and receivers, and to identify individuals who frequently sent or received emails or exhibited other abnormal email behaviors in this case, network graph was constructed.

Data Extraction and Processing

First, read the email data from the CSV file and extracted the sender column (X-From) and receiver column (X-To) to build an undirected graph. Additionally, calculate the density, number of nodes, and number of edges in the network to help understand the network's density, changes in the number of nodes, and the impact of adding a node on the overall network structure.

Due to the large volume of data, it was not feasible to construct network graphs for all entries. Considering that the email files are random, I extracted the first 5,000 emails and incrementally added them in batches of 500 to create relevant charts.



```
email_counts = range(500, 5001, 500)
density_values = []
num_nodes = []
num_edges = []

for count in email_counts:
    G = nx.from_pandas_edgelist(df[:count], 'X-From', 'X-To')

    density = nx.density(G)
    num_node = len(G.nodes())
    num_edge = len(G.edges())

    density_values.append(density)
    num_nodes.append(num_node)
    num_edges.append(num_edge)
```

Analysis of Relationship Graph Features

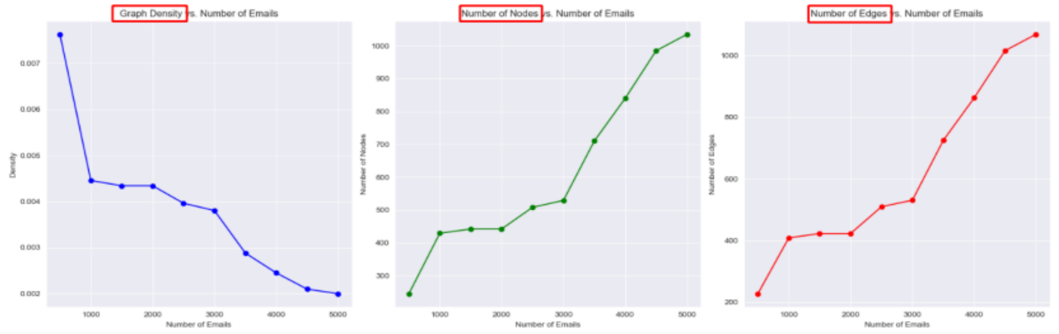
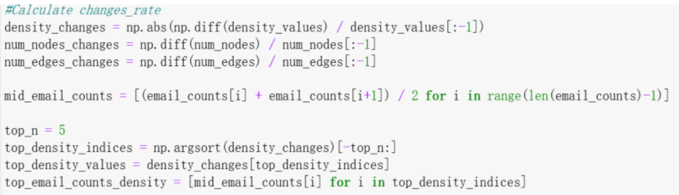


Fig10 Changes of density, number of nodes & edges

From the graph, we can observe the following:

- As the number of emails increases, the density exhibits a steady downward trend, while the number of nodes and edges shows a consistent upward trend.
- When the number of emails increases from 750 to 1,000, there is a notable change in all metrics.
- The trends in the number of nodes and edges are very similar, with almost identical curve shapes. This suggests that no single outlier has caused a significant alteration in the relationships within the analyzed dataset.

To further investigate the trends of these three metrics, their rates of change have been calculated and visualized. This provides a clearer insight into the correlation between the rates of change of these metrics.

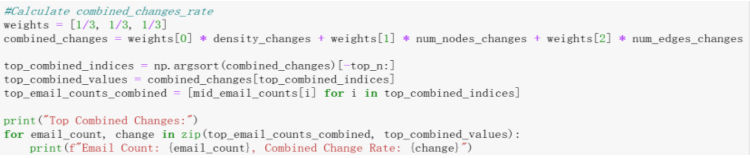


```
#Calculate changes_rate
density_changes = np.abs(np.diff(density_values) / density_values[:-1])
num_nodes_changes = np.diff(num_nodes) / num_nodes[:-1]
num_edges_changes = np.diff(num_edges) / num_edges[:-1]

mid_email_counts = [(email_counts[i] + email_counts[i+1]) / 2 for i in range(len(email_counts)-1)]

top_n = 5
top_density_indices = np.argsort(density_changes)[-top_n:]
top_density_values = density_changes[top_density_indices]
top_email_counts_density = [mid_email_counts[i] for i in top_density_indices]
```

In addition, We assigned weights to the three metrics and calculated a composite change rate index to comprehensively demonstrate the overall impact of the changes in these metrics on the relationship graph. And We chose to assign equal weights here.

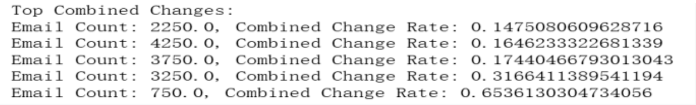


```
#Calculate combined_changes_rate
weights = [1/3, 1/3, 1/3]
combined_changes = weights[0] * density_changes + weights[1] * num_nodes_changes + weights[2] * num_edges_changes

top_combined_indices = np.argsort(combined_changes)[-top_n:]
top_combined_values = combined_changes[top_combined_indices]
top_email_counts_combined = [mid_email_counts[i] for i in top_combined_indices]

print("Top Combined Changes:")
for email_count, change in zip(top_email_counts_combined, top_combined_values):
    print(f"Email Count: {email_count}, Combined Change Rate: {change}")
```

The change rate curves and composite change rates are shown in the following figure:



```
Top Combined Changes:
Email Count: 2250.0, Combined Change Rate: 0.1475080609628716
Email Count: 4250.0, Combined Change Rate: 0.1646233322681339
Email Count: 3750.0, Combined Change Rate: 0.17440466793013043
Email Count: 3250.0, Combined Change Rate: 0.3166411389541194
Email Count: 750.0, Combined Change Rate: 0.6536130304734056
```

Fig11 composite change rates of different points

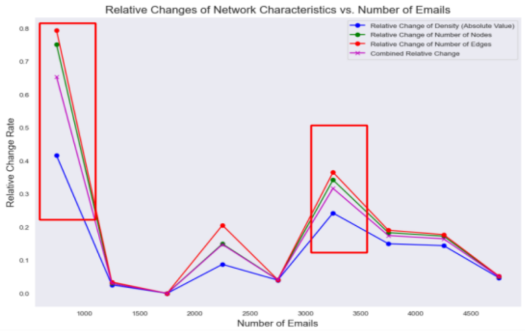


Fig12 Relative Changes of Network Characteristics

From the graph, we can observe the following:

- When the number of emails is 750, the composite change rate index is the highest, indicating that the increase in email volume has a significant impact on the relationships between nodes at this point. However, I believe this is due to the small sample size, which can easily show strong correlations and is not suitable for investigating the relationships across the entire dataset. Therefore, the data set with 750 emails is discarded.
- When the number of emails is 3250, the composite change rate index is the second highest. At this point, the change rates for density, number of nodes, and number of edges are all substantial. Therefore, the data set with 3250 emails is selected for network construction.

Network Graph

Based on the final selection results, a relationship graph was constructed depicting the senders and receivers of the first 3250 emails.

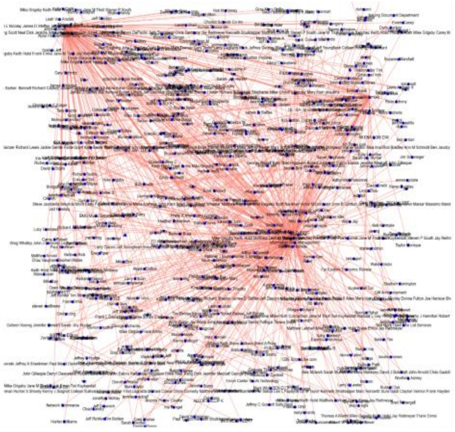
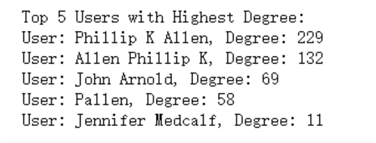


Fig13 Network Graph of first 3250 emails

The above network graph illustrates that there are relatively clear focal points among the senders and receivers of the first 3250 emails.

However, it is difficult to identify specific entities. Therefore, we calculated the degree of each node and **printed out the top 5 users** with the highest node degrees.



```
Top 5 Users with Highest Degree:
User: Phillip K Allen, Degree: 229
User: Allen Phillip K, Degree: 132
User: John Arnold, Degree: 69
User: Pallen, Degree: 58
User: Jennifer Medcalf, Degree: 11
```

Fig14 Top 5 users with highest degree

- "Phillip K Allen" and "Allen Phillip K" in the output results are actually the same person, but they were identified as different users in the network graph. This discrepancy is likely due to variations in how employees addressed each other in their emails.

As shown in the figure, the degree of "Phillip K Allen" is particularly high, significantly exceeding that of the third-ranked user which suggests that he may be one of the key figures in the case.

Network Graph of 5 main users

To make the relationship graph more concise and to highlight the importance of the five identified individuals and their email interactions, we removed all email relationships unrelated to these five individuals and only plotted their relevant connections, as shown in the following figure:

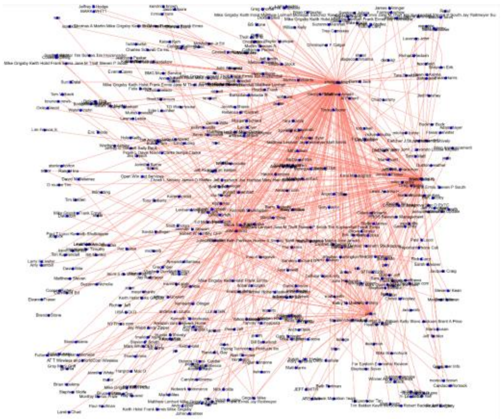


Fig15 Network Graph of 5 main users

The figure demonstrates that among the first 3250 emails, there are clear central nodes, primarily centered around Phillip K Allen. This suggests that these individuals may have significant ties to the truth of the case. Further analysis should be conducted to explore these connections, which could provide valuable insights for the ultimate determination of guilt.

Analyze Conclusion of network

After verification, we found that **Phillip K. Allen served as Enron's Chief Risk Officer,** responsible for managing and assessing various risks, including financial, market, and operational risks.

He was a major participant in the company's complex financial transactions, including many derivative trades and the establishment of special purpose entities (SPEs). These entities were used to hide the company's debt and losses, **further validating his role as one of the perpetrators in the Enron scandal.**

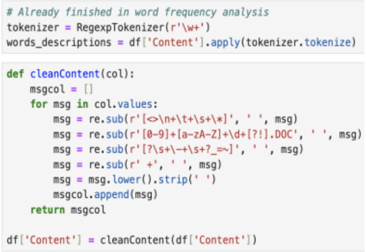
As for why other employees also have relatively high degrees, though significantly lower than Phillip K. Allen, I believe this is likely because their job responsibilities require frequent communication with internal personnel. However, they are not directly involved in the criminal activities related to the case. After further investigation, it was confirmed that the remaining three employees among the five identified individuals are not associated with the criminal activities.

Topic Analysis

We analyzed the content of these emails for cluster analysis to explore internal key communication topics within Enron.

Data Processing

For simplicity, we primarily focused on 'Content', which holds the main body of the email, to analyze the topics of internal communication. To prepare the data for clustering, we carried out several data processing steps. In the previous word frequency analysis, tokenization was already completed. Then we cleaned the 'Content' by removing extra symbols, numbers, and case differences, standardizing the text format.



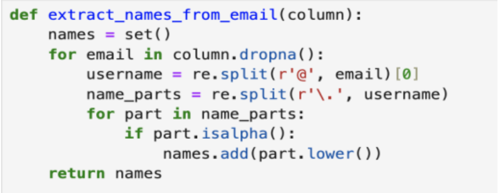
```
# Already finished in word frequency analysis
tokenizer = RegexpTokenizer(r'\w+')
words_descriptions = df['Content'].apply(tokenizer.tokenize)
```

```
def cleanContent(col):
    msgcol = []
    for msg in col.values:
        msg = re.sub(r'[<>\n+\t+\s+\*]', ' ', msg)
        msg = re.sub(r'[0-9]+[a-zA-Z]+\d+[?!].DOC', ' ', msg)
        msg = re.sub(r'[?\s+\-+\s+?_=~]', ' ', msg)
        msg = re.sub(r' +', ' ', msg)
        msg = msg.lower().strip(' ')
        msgcol.append(msg)
    return msgcol

df['Content'] = cleanContent(df['Content'])
```

Fig16. Clean the 'Content'

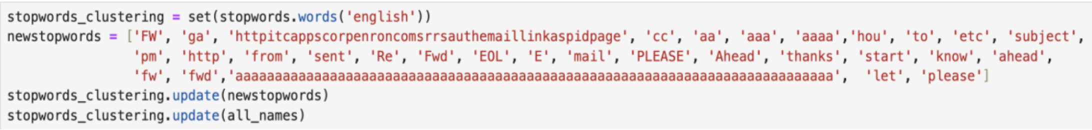
On this basis, we extracted sender and recipient names from 'From' and 'To' in the dataframe and removed them from 'Content' to prevent personal names from influencing the clustering topics.



```
def extract_names_from_email(column):
    names = set()
    for email in column.dropna():
        username = re.split(r'@', email)[0]
        name_parts = re.split(r'\.', username)
        for part in name_parts:
            if part.isalpha():
                names.add(part.lower())
    return names
```

Fig17. Extract Names from Email Addresses

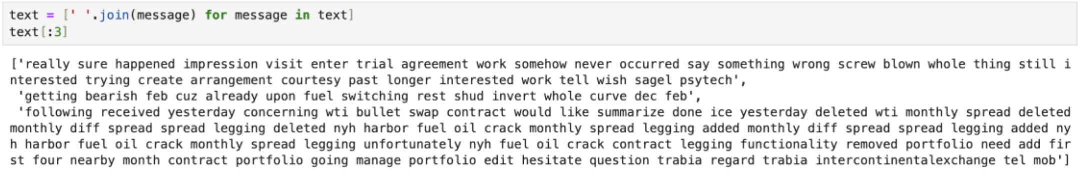
After the initial cleaning, we tokenized the text, converted it to lowercase, removed stop words, and applied lemmatization to standardize words to their root form. This process also reduces noise for clearer analysis.



```
stopwords_clustering = set(stopwords.words('english'))
newstopwords = ['FW', 'ga', 'httpitcappscorpenroncomsrrsauthemaillinkaspidpage', 'cc', 'aa', 'aaa', 'aaaa','hou', 'to', 'etc', 'subject',
                'pm', 'http', 'from', 'sent', 'Re', 'Fwd', 'EOL', 'E', 'mail', 'PLEASE', 'Ahead', 'thanks', 'start', 'know', 'ahead',
                'fw', 'fwd','aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa',  'let', 'please']
stopwords_clustering.update(newstopwords)
stopwords_clustering.update(all_names)
```

Fig18. Create stopwords for cluster analysis

The following shows the first 3 rows of the 'text' list.

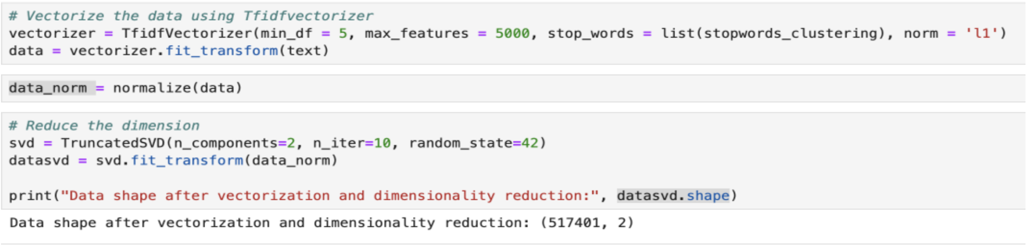


```
text = [' '.join(message) for message in text]
text[:3]
```

```
['really sure happened impression visit enter trial agreement work somehow never occurred say something wrong screw blown whole thing still i
nterested trying create arrangement courtesy past longer interested work tell wish sagel psytech',
 'getting bearish feb cuz already upon fuel switching rest shud invert whole curve dec feb',
 'following received yesterday concerning wti bullet swap contract would like summarize done ice yesterday deleted wti monthly spread deleted
monthly diff spread spread legging deleted nyh harbor fuel oil crack monthly spread legging added monthly diff spread spread legging added ny
h harbor fuel oil crack monthly spread legging unfortunately nyh fuel oil crack contract legging functionality removed portfolio need add fir
st four nearby month contract portfolio going manage portfolio edit hesitate question trabia regard trabia intercontinentalexchange tel mob']
```

Fig19. First 3 rows of 'text'

We used TfidfVectorizer to vectorize 'text' using, then normalized it, and finally reduced the dimensionality to two principal components using TruncatedSVD.



```
# Vectorize the data using Tfidfvectorizer
vectorizer = TfidfVectorizer(min_df = 5, max_features = 5000, stop_words = list(stopwords_clustering), norm = 'l1')
data = vectorizer.fit_transform(text)
```

```
data_norm = normalize(data)
```

```
# Reduce the dimension
svd = TruncatedSVD(n_components=2, n_iter=10, random_state=42)
datasvd = svd.fit_transform(data_norm)

print("Data shape after vectorization and dimensionality reduction:", datasvd.shape)
```

```
Data shape after vectorization and dimensionality reduction: (517401, 2)
```

Fig20. Text Vectorization, Normalization, and Dimensionality Reduction Process

Clustering

To determine the best number of clusters, we used the elbow method, which calculates the sum of squared distances (inertia) across different cluster numbers. The point at which the decrease in inertia slows down significantly, known as the “elbow point,” suggests an appropriate number of clusters. In our case, this helped us select a cluster number that balances detail and interpretability.

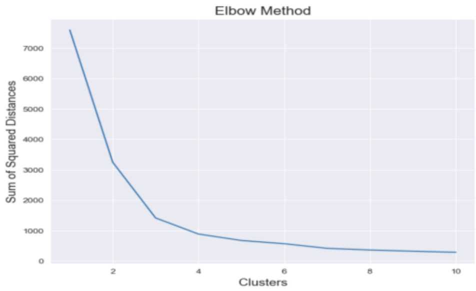
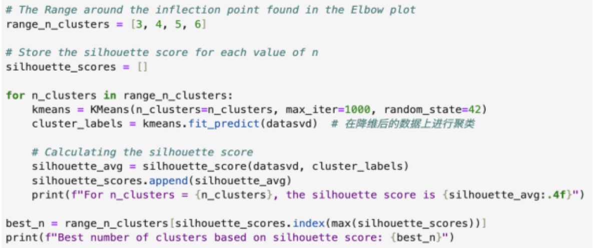


Fig21. Elbow Plot

Additionally, we applied the silhouette coefficient as a secondary validation, further confirming the optimal cluster count for our dataset.Based on the elbow plot and silhouette score, the optimal number of clusters is 3. The scatter plot of the clusters is shown below.



```
# The Range around the inflection point found in the Elbow plot
range_n_clusters = [3, 4, 5, 6]

# Store the silhouette score for each value of n
silhouette_scores = []

for n_clusters in range_n_clusters:
    kmeans = KMeans(n_clusters=n_clusters, max_iter=1000, random_state=42)
    cluster_labels = kmeans.fit_predict(datasvd)  # 在降维后的数据上进行聚类

    # Calculating the silhouette score
    silhouette_avg = silhouette_score(datasvd, cluster_labels)
    silhouette_scores.append(silhouette_avg)
    print(f"For n_clusters = {n_clusters}, the silhouette score is {silhouette_avg:.4f}")

best_n = range_n_clusters[silhouette_scores.index(max(silhouette_scores))]
print(f"Best number of clusters based on silhouette score: {best_n}")
```

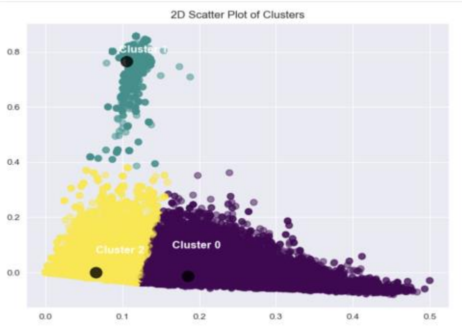
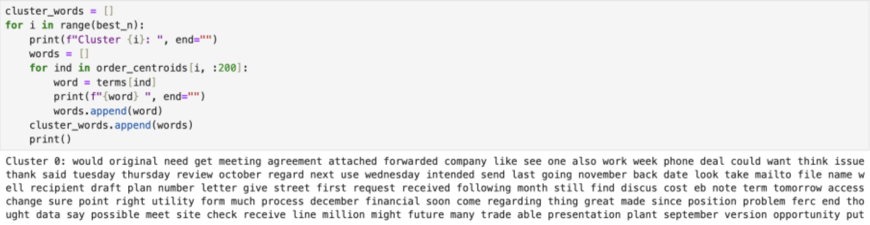


Fig22. Calculate Best Number of Clusters

Fig23. 2D Scatter Plot of Clustered Data Points

Results and Analysis

After clustering, we output the high-frequency topic words for each cluster to identify the main topics within each cluster.

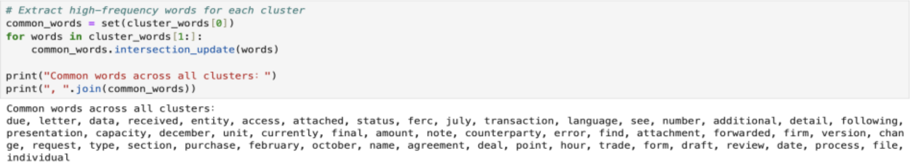


```
cluster_words = []
for i in range(best_n):
    print(f"Cluster {i}: ", end="")
    words = []
    for ind in order_centroids[i, :200]:
        word = terms[ind]
        print(f"{word} ", end="")
        words.append(word)
    cluster_words.append(words)
    print()
```

```
Cluster 0: would original need get meeting agreement attached forwarded company like see one also work week phone deal could want think issue
thank said tuesday thursday review october regard next use wednesday intended send last going november back date look take mailto file name w
ell recipient draft plan number letter give street first request received following month still find discus cost eb note term tomorrow access
change sure point right utility form much process december financial soon come regarding thing great made since position problem ferc end tho
ught data say possible meet site check receive line million might future many trade able presentation plant september version opportunity put
```

Fig24. Cluster the Topic Words

We extracted and removed common words across 3 clusters from each cluster's topic words to allow each cluster's unique characteristics to stand out more clearly. These common words, like "transaction," "data," and "status," reflect standard business processes in Enron emails, such as transactions, document management, and status updates. Their presence across all clusters shows their broad use across various departments.



```
# Extract high-frequency words for each cluster
common_words = set(cluster_words[0])
for words in cluster_words[1:]:
    common_words.intersection_update(words)

print("Common words across all clusters: ")
print(", ".join(common_words))
```

```
Common words across all clusters:
due, letter, data, received, entity, access, attached, status, ferc, july, transaction, language, see, number, additional, detail, following,
presentation, capacity, december, unit, currently, final, amount, note, counterparty, error, find, attachment, forwarded, firm, version, chan
ge, request, type, section, purchase, february, october, name, agreement, deal, point, hour, trade, form, draft, review, date, process, file,
individual
```

Fig25. Extract common words for each cluster

We used the WordCloud library to generate word clouds for each cluster and display the high-frequency words within each cluster.

Cluster 0 reflects routine communication between Enron's management and staff, such as meeting arrangements and business planning, potentially involving financial operations and decisions. In the Enron scandal, these communications likely includes discussions between executives and departments on financial maneuvers and strategies. For evidence of complex financial structures that concealed losses and inflated profits, Cluster 0 emails are worth exploring.

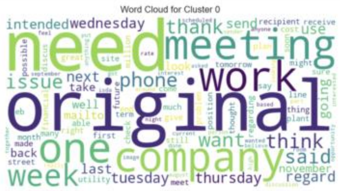


Fig26. Word Cloud for Cluster 0

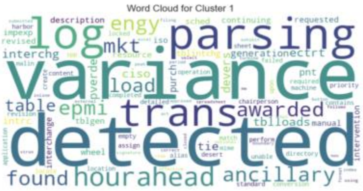


Fig27. Word Cloud for Cluster 1

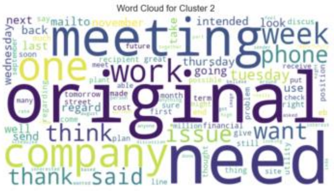


Fig28. Word Cloud for Cluster 2

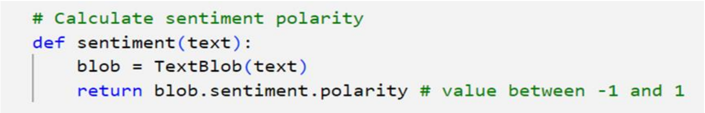
Cluster 1 involves Enron's technical operations and data analysis for energy market trading, reflecting activities by technical staff and data analysts. For evidence of price manipulation and inflated profits, Cluster 1 emails should be reviewed.

Cluster 2 primarily reflects routine management communication within the company, involving work requirements, planning, and phone communication, showing daily coordination and task scheduling among employees. Unlike the high-level financial decisions in Cluster 0, Cluster 2 focuses more on daily tasks and progress updates to ensure employees follow company directives. For insights into how Enron used daily management to conceal the true financial state, Cluster 2 emails are relevant.

Sentiment Analysis

To further analyze the content of each email, our group conducted sentiment analysis to classify the Enron emails based on the emotion it conveys.

We used TextBlob to calculate each email's sentiment polarity.



```python
# Calculate sentiment polarity
def sentiment(text):
    blob = TextBlob(text)
    return blob.sentiment.polarity # value between -1 and 1
```

Fig29. Sentiment Polarity Calculation

Polarity values range from -1 to 1, where negative values indicate negative sentiment, positive values indicate positive sentiment, and a value of 0 indicates a neutral sentiment.

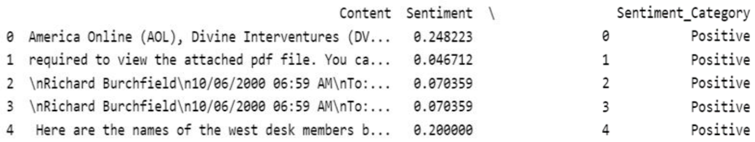
Based on the results, each email could be classified into one of these 3 categories:

Posit3ve: Sentiment score > 0

Negative: Sentiment score < 0

Neutral: Sentiment score = 0

The results of the first 5 rows are as follows:



```
                                        Content  Sentiment  \            Sentiment_Category
0  America Online (AOL), Divine Interventures (DV...   0.248223           0          Positive
1  required to view the attached pdf file. You ca...   0.046712           1          Positive
2  \nRichard Burchfield\n10/06/2000 06:59 AM\nTo:...   0.070359           2          Positive
3  \nRichard Burchfield\n10/06/2000 06:59 AM\nTo:...   0.070359           3          Positive
4   Here are the names of the west desk members b...   0.200000           4          Positive
```

Fig30. Sentiment Category of first 5 rows

To visualize the overall sentiment distribution in this dataset, we calculated the number of emails in each sentiment category and generate a bar plot to visually display this distribution.

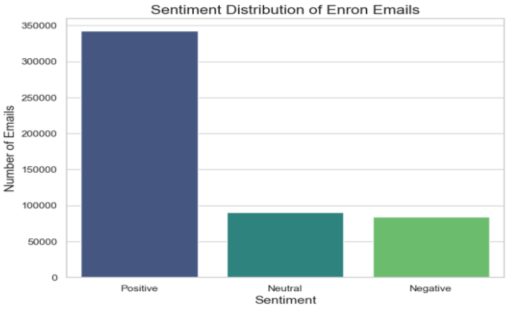
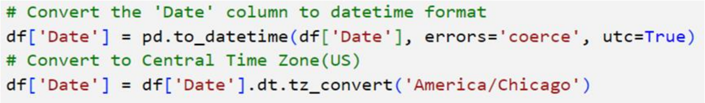


Fig31. Sentiment Distribution

Sentiment Trends Over Time

After categorizing sentiment in emails, we wanted to analyze sentiment trends to identify some patterns in different sentiments over time, particularly focusing on fluctuations around key events within the specified time frame.

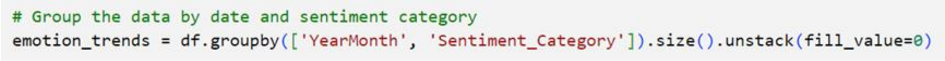
To accomplish this, we first converted the date column into a datetime format. In order to ensure accuracy in time-based analysis, we converted the time zones to the Central Time Zone (America/Chicago), where Enron was headquartered.



```
# Convert the 'Date' column to datetime format
df['Date'] = pd.to_datetime(df['Date'], errors='coerce', utc=True)
# Convert to Central Time Zone(US)
df['Date'] = df['Date'].dt.tz_convert('America/Chicago')
```

Fig32. Date format conversion

The data was then grouped by date and sentiment categories.



```
# Group the data by date and sentiment category
emotion_trends = df.groupby(['YearMonth', 'Sentiment_Category']).size().unstack(fill_value=0)
```

Fig33. Group data by date and sentiment categories

We calculated the number of emails in each category per month, creating a time series that shows how sentiment in Enron emails changed over time. The sentiment trends were visualized with a line chart, where each line represents a different sentiment category.

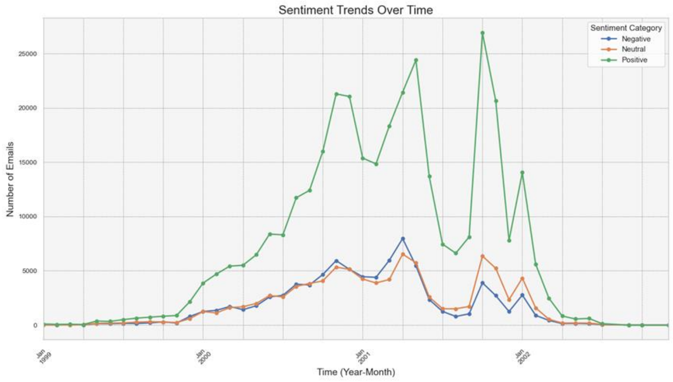


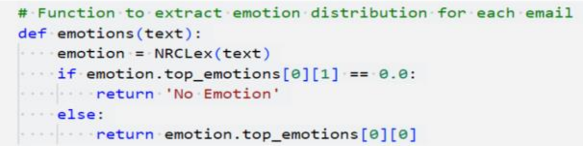
Fig34. Sentiment Trends Over Time

This chart clearly illustrates the sentiment fluctuations among Enron employees from 1999 to 2002. Notably, there were some significant shifts in employee sentiment around the time of major events, particularly the Enron scandal.

Emotion Analysis

To specify the sentiments in emails, we then conducted emotion analysis to understand the emotional tone conveyed in each email.

For each text, we applied NRCLex to extract its primary emotion and categorize each email more specifically into several emotion types, such as trust, joy, surprise, fear, sadness, anger, etc.

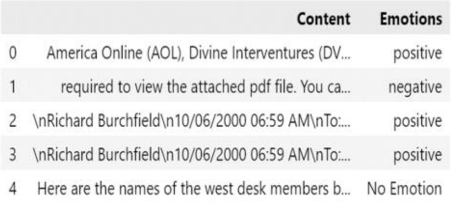


```
# Function to extract emotion distribution for each email
def emotions(text):
    emotion = NRCLex(text)
    if emotion.top_emotions[0][1] == 0.0:
        return 'No Emotion'
    else:
        return emotion.top_emotions[0][0]
```

Fig35. Emotion Extraction

If no strong emotion was detected (score of 0), the email was classified under "No Emotion." Otherwise, the highest-scoring emotion was assigned to that email.

For example, the first five emails were categorized as:



	Content	Emotions
0	America Online (AOL), Divine Interventures (DV...	positive
1	required to view the attached pdf file. You ca...	negative
2	\nRichard Burchfield\n10/06/2000 06:59 AM\nTo:...	positive
3	\nRichard Burchfield\n10/06/2000 06:59 AM\nTo:...	positive
4	Here are the names of the west desk members b...	No Emotion

Fig36. Emotion Categories of first 5 emails

After that, we calculated the distribution of these emotions and visualized it via a pie chart.

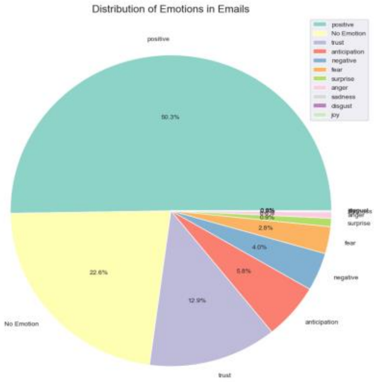


Fig37. Emotion Distribution

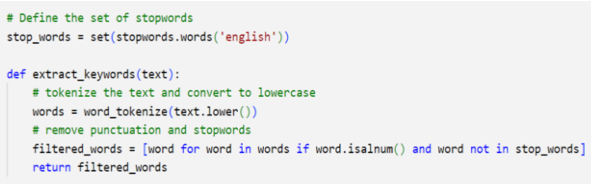
Each slice of the pie chart indicates the percentage of emails belonging to a particular emotion category, making it easy to see which emotions are most prevalent.

Through the chart, we can see that while positive and neutral emotions dominate, the presence of fear, anticipation and negative could correlate with periods of uncertainty, especially leading up to the company's crisis.

Keywords Associated with Each Emotion Category

We also extract the keywords for each emotion category in emails to better understand employees' emotions.

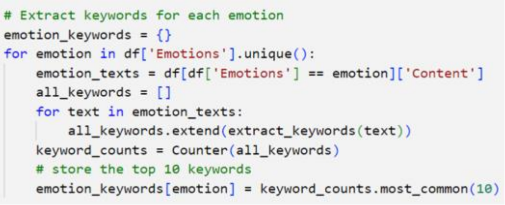
To accurately identify meaningful keywords, we first remove common stopwords (e.g., "the," "and") and punctuation in text using NLTK's stopwords module. We then categorize emails by their assigned emotion label and extract keywords separately for each emotion category.



```
# Define the set of stopwords
stop_words = set(stopwords.words('english'))

def extract_keywords(text):
    # tokenize the text and convert to lowercase
    words = word_tokenize(text.lower())
    # remove punctuation and stopwords
    filtered_words = [word for word in words if word.isalnum() and word not in stop_words]
    return filtered_words
```

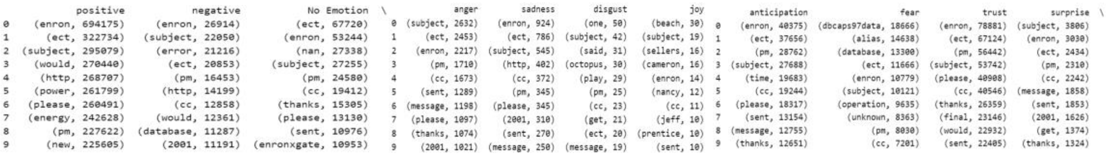
Fig38. Data Processing



```
# Extract keywords for each emotion
emotion_keywords = {}
for emotion in df['Emotions'].unique():
    emotion_texts = df[df['Emotions'] == emotion]['Content']
    all_keywords = []
    for text in emotion_texts:
        all_keywords.extend(extract_keywords(text))
    keyword_counts = Counter(all_keywords)
    # store the top 10 keywords
    emotion_keywords[emotion] = keyword_counts.most_common(10)
```

Fig39. Keywords Extraction

After collecting all words for each emotion, we counted the frequency of each word and stored the top 10 keywords. The results compiled into a DataFrame are as follows:



```
           positive          negative          No Emotion  \
0   (enron, 694175)   (enron, 26914)        (ect, 67720)
1     (ect, 322734) (subject, 22050)      (enron, 53244)
2 (subject, 295079)   (error, 21216)        (nan, 27338)
3   (would, 270440)     (ect, 20853)    (subject, 27255)
4    (http, 268707)      (pm, 16453)         (pm, 24580)
5   (power, 261799)    (http, 14199)         (cc, 19412)
6  (please, 260491)      (cc, 12858)     (thanks, 15305)
7  (energy, 242628)   (would, 12361)     (please, 13130)
8      (pm, 227622) (database, 11287)      (sent, 10976)
9     (new, 225605)     (2001, 11191) (enronxgate, 10953)

              anger          sadness          disgust             joy
0  (subject, 2632)     (enron, 924)        (one, 50)     (beach, 30)
1      (ect, 2453)       (ect, 786)    (subject, 42)   (subject, 19)
2    (enron, 2217)   (subject, 545)       (said, 31)   (sellers, 16)
3       (pm, 1710)      (http, 402)    (octopus, 30)   (cameron, 16)
4       (cc, 1673)        (cc, 372)       (play, 29)     (enron, 14)
5     (sent, 1289)        (pm, 345)         (pm, 25)     (nancy, 12)
6  (message, 1198)    (please, 345)         (cc, 23)        (cc, 11)
7   (please, 1097)      (2001, 310)        (get, 21)      (jeff, 10)
8   (thanks, 1074)      (sent, 270)        (ect, 20)  (prentice, 10)
9     (2001, 1021)   (message, 250)    (message, 19)      (sent, 10)

        anticipation                  fear             trust          surprise  \
0    (enron, 40375)  (dbcaps97data, 18666)    (enron, 78881)  (subject, 3806)
1      (ect, 37656)         (alias, 14638)      (ect, 67124)    (enron, 3030)
2       (pm, 28762)      (database, 13300)       (pm, 56442)      (ect, 2434)
3  (subject, 27688)           (ect, 11666)  (subject, 53742)       (pm, 2310)
4     (time, 19683)         (enron, 10779)   (please, 40908)       (cc, 2242)
5       (cc, 19244)       (subject, 10121)       (cc, 40546)  (message, 1858)
6   (please, 18317)      (operation, 9635)   (thanks, 26359)     (sent, 1853)
7     (sent, 13154)        (unknown, 8363)    (final, 23146)     (2001, 1626)
8  (message, 12755)             (pm, 8030)    (would, 22932)      (get, 1374)
9   (thanks, 12651)             (cc, 7201)     (sent, 22405)   (thanks, 1324)
```

Fig40. Top 10 Keywords for Each Emotion

We can see that keywords related to negative emotions like "error" and "2001" highlight the stressful feelings among people in the company, especially around the crisis period.

Predict Criminal

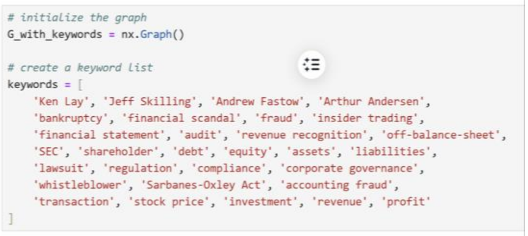
Through early method reviews we found several machine learning algorithms available (e.g., Gaussian Plain Bayes, Decision Tree Classifiers, Support Vector Machines, etc.). These methods all use the 'final_project_dataset.pkl' data source, however, this data source integrates some structured data about employees, such as salary, position, bonus, etc., which are from other publicly available information rather than the original email message 'enron_mail_20150507.tar.gz'.

Social Network Analysis

Considering that other data sources are not within the scope of the project requirements, plus our group wanted to use unsupervised learning methods to find the key people in the Enron incident. We used **Social Network Analysis** to find Point of Interests (POIs) for the Enron event to visualize the key players of the Enron event in a simple way.

We were going to build the social network using X-From, X-To and Content columns. In order to reduce our computational effort and improve accuracy, we need to preprocess the Content columns before building the social network, including removing irrelevant stop words, converting all the Content to lowercase, word shape reduction, etc.

Furthermore, we wanted to construct a social network that would focus on Enron's financial fraud itself, rather than just show people relationships, we constructed a keywords list (Fig 42), requiring that when 'Processed_Content' contains at least one of the keywords to consider constructing relationships based on sender and recipient. Then, we used bar charts to show the KOIs after building the network and calculating the degree centrality.



```
# initialize the graph
G_with_keywords = nx.Graph()

# create a keyword list
keywords = [
    'Ken Lay', 'Jeff Skilling', 'Andrew Fastow', 'Arthur Andersen',
    'bankruptcy', 'financial scandal', 'fraud', 'insider trading',
    'financial statement', 'audit', 'revenue recognition', 'off-balance-sheet',
    'SEC', 'shareholder', 'debt', 'equity', 'assets', 'liabilities',
    'lawsuit', 'regulation', 'compliance', 'corporate governance',
    'whistleblower', 'Sarbanes-Oxley Act', 'accounting fraud',
    'transaction', 'stock price', 'investment', 'revenue', 'profit'
]
```

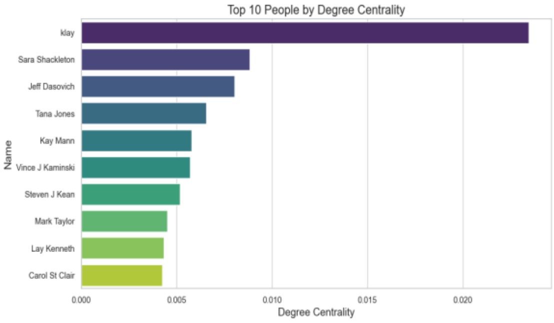


Fig41. Keywords List

Fig42. Top 10 People by Degree Centrality

Through the output, we found that some employees or executives had a high association with the Enron event. These include Vince J. Kaminski, Sara Shackleton, Tana Jones, Ken Lay, Jeff Dasovich, Mark Taylor, Kay Mann, Carol St. Clair, and Steven J. Kean.

Vince J. Kaminski, as CRO, has repeatedly warned the company's top executives about financial reporting and wind management. Sara Shackleton and Tana Jones have also warned the top executives as legal counsel and treasurer, respectively, but none of the three have had sufficient impact.

The remaining six people in the table, including Ken Lay (klay and Lay Kenneth are the same person), are directly or indirectly led to Enron's bankruptcy. Ken Lay as the CEO, because of the connivance of the financial operation was eventually sentenced by the court at last.

With the information from Wikipedia, it is easy to know that executives such as Jeff Skilling (COO & CEO), Richard Causey and Andrew Fastow (CFO) are also liable. However, Richard Causey and Andrew Fastow do not have relevant email records, probably due to personal privacy reasons. Meanwhile, Jeff Skilling's relevant emails are only about 3,000 in the 50w dataset. The above reasons can explain why social network analytics were unable to uncover these three individuals who had significant influence in Enron scandal.

In summary, social network analysis can find the potential POIs in the Enron scandal. This approach is concise and clear, which helps us to find the key people quickly.

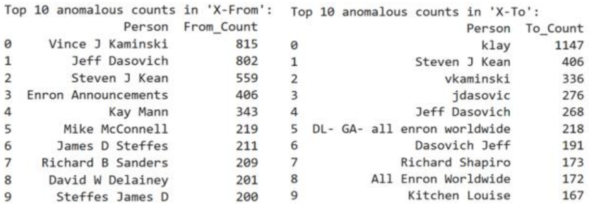
Isolation Forest to Predict the Abnormal

While social network analysis can be helpful for understanding the connections in a given event, it can only determine which people in the Enron case are more closely related to other people. However, it can't produce evidence that this person might be a potential criminal. The simplest example is that Ken Lay, as the CEO of the company, should rightfully have a higher Degree Centrality.

Compared to SNA, Isolation Forest focuses on isolating data points and identifies anomalous data by analyzing the distribution of features. It is more suitable for detection of people or emails that are 'different', such as potential fraud. Moreover, Isolation Forest can process high-dimensional data such as TF-IDF, sentiment scores and other features of email content into a feature vector. Based on this, we try to use Isolation Forest to find potential criminals.

First, we fill the 'Processed_Content' column with missing strings and convert it into a sparse matrix with 'max_features = 1000'. In addition, considering the fluctuation of the overall sentiment of the email during the fitting process, we combine the text data with the sentiment scores.

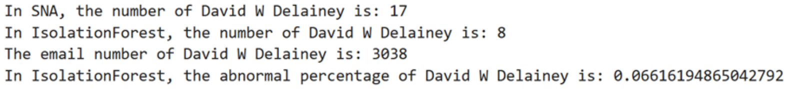
After normalization, we set the contamination to 0.05 and fit the predicted abnormal emails, meaning that abnormal emails are assumed to be 5% of the overall. The predicted results will be stored in the 'anomaly' column, with values of -1 representing anomalous emails and values of 1 representing normal emails. Now that the anomalous results are available (Fig 44), we would like to show the people corresponding to the top 10 anomalous email counts in terms of sender and recipient dimensions, to find those key people who may be associated with fraudulent behavior.



```
Top 10 anomalous counts in 'X-From':   Top 10 anomalous counts in 'X-To':
               Person  From_Count                            Person  To_Count
0     Vince J Kaminski        815      0                          klay      1147
1        Jeff Dasovich        802      1                 Steven J Kean       406
2        Steven J Kean        559      2                     vkaminski       336
3  Enron Announcements        406      3                      jdasovic       276
4             Kay Mann        343      4                 Jeff Dasovich       268
5       Mike McConnell        219      5  DL- GA- all enron worldwide       218
6      James D Steffes        211      6                 Dasovich Jeff       191
7     Richard B Sanders       209      7               Richard Shapiro       173
8      David W Delainey       201      8            All Enron Worldwide       172
9      Steffes James D        200      9                Kitchen Louise       167
```

Fig 43. Result of Isolation Forest

Compared to the SNA results, we found some different results. For example, David W Delainey was ranked 8th in IsolationForest for the number of anomalous emails found, yet only 17th in SNA. This is very interesting because David W. Delainey was the CEO of Enron's North American energy trading division, played a key role in the company's financial manipulation and actively cooperated with the prosecution. Such a key player social network does not reflect its importance. Through Fig 45, it was found that David W. Delainey had only 3,038 emails, ranking him 21st among all senders, and that his email anomaly rate of 6.6% was higher than the average.



```
In SNA, the number of David W Delainey is: 17
In IsolationForest, the number of David W Delainey is: 8
The email number of David W Delainey is: 3038
In IsolationForest, the abnormal percentage of David W Delainey is: 0.06616194865042792
```

Fig 44. Some Describe of David W Delainey

Finally, we visualize the results (Fig 46). With the above example, it's easy to see the limitations of SNA. Since SNA uses a relationship between senders and recipients to build the network, it leads to results that are affected by the number of emails. And IsolationForest can well bypass this problem by calculating the anomaly rate. Moreover, it can freely combine sparse matrices, which means that using high-frequency words in the fitting process and considering influence of emotions becomes possible.

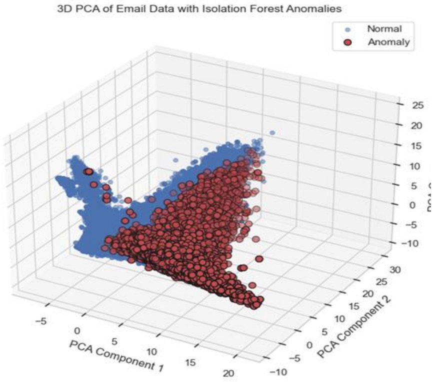


Fig 45. 3D PCA of Email Data with Isolation Forest

Anomaly Visualization with Time Series

Based on the identification of outliers, we added time series and finally visualised them to see at what points in time the key players might have influenced Enron. Figures 47 and 48 are the time series plots of Ken Lay (CEO) and Vince J Kaminski (CRO) respectively, showing the number of their outlier emails and the trend of their sentiments over the months. It can be seen that Ken Lay's mood fluctuated in December 2000, when Enron was facing some significant financial and operational challenges, and in December 2001, when Enron declared bankruptcy, Ken Lay received a sharp increase in abnormal emails and his mood dropped to a low point. Vince J. Kaminski, as CRO, had identified Enron's potential risks in advance and had warned the company's executives several times in March 2000, September 2000 and May 2001, coinciding with the peaks in the time series graph. Ultimately, however, the company did not heed his advice and went bankrupt.

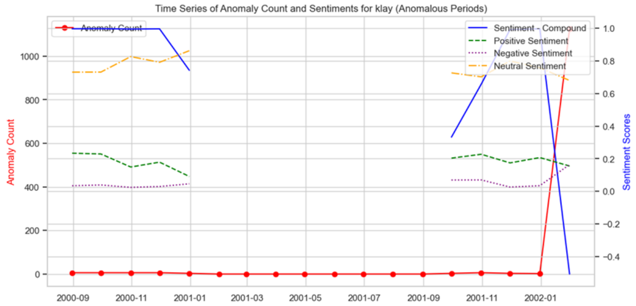


Fig 46. Time Series of Anomaly Count and Sentiments forKen Lay

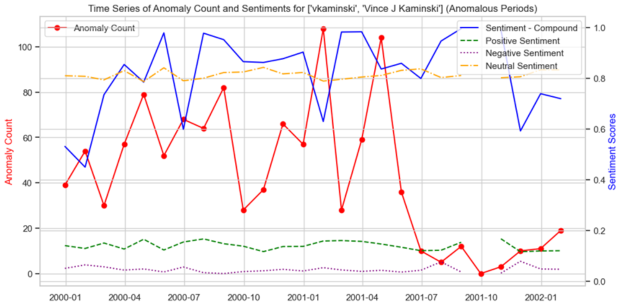


Fig47. Time Series of Anomaly Count and Sentiments for Vince J Kaminski

Limitation with Current Methods

In the process of creating the network graph, due to the large volume of email data and its randomness, we only verified the composite change rate of the first 5000 emails for the relationship graph analysis, to make a reasonable decision regarding the data volume.

If computational resources permit, it would be advisable to sort and analyze all over 40,000 emails in chronological order, and to conduct a segmented analysis of the data from before and after the emergence of negative news until the company's bankruptcy. By performing a detailed analysis of the relationship graph for each stage, we could better investigate the key individuals and main content of internal emails at different stages, thereby providing a more detailed analysis of the case.

We used the SNA and Isolation Forest methods to predict criminals. Although both methods successfully predicted some criminals in the Enron incident, both methods tend to find POIs. Which means that these two methods will find people related to the incident, but not necessarily the guilty ones. For example, Vince J Kaminski, who repeatedly warned Enron executives of risks, was also identified by these two methods because of the large proportion of abnormal emails. This shows that the method we used is generalized and still lacks the ability to identify specific tendencies.

Due to the limited CPU computing power of team members, some feasible methods cannot be used. For example, using One-Class SVM to imitate Isolation Forest to label abnormal emails.

The original email dataset lacks structured data about employees (such as salary, bonus, position, etc.), and it would be a very large task to use NER to identify the main body of all the text in 500,000 emails. Therefore, we did not try to incorporate the employee information published on the Internet and use machine learning to make predictions which might be a better way to predict the criminals.

Conclusion

By applying methodology about NLP, we got some interesting findings:

1. Can we identify potential criminals or suspicious individuals from the Enron email dataset?

Through **Network Graph**、 **Social Network Analysis (SNA)** and **Isolation Forest**, we can identify potential criminals.

The network graph demonstrates that among the first 3250 emails, there are clear central nodes, primarily centered around Phillip K. Allen which means the manager was likely the potential criminal. After verification, it has been confirmed that Phillip K. Allen, Enron's Chief Risk Officer, was indeed a major participant in the company's complex financial transactions, including numerous derivative trades and the establishment of special purpose entities (SPEs) used to hide the company's debt and losses. This further validates his role as one of the perpetrators in the Enron scandal.

Both SNA and Isolation Forest can also find POIs of the Enron scandal. SNA is simpler and more intuitive, while Isolation Forest is more effective. However, SNA and Isolation Forest also found some interesting conclusions, which will be discussed at the end

2. Can we uncover changes in the main content and emotional trends of emails before and after negative news emerging?

To uncover changes in the main content and emotional trends of emails before and after negative news emerging, we used TextBlob and NRCLex to classify the sentiment (positive, negative, or neutral) in each email and extract dominant emotions (e.g., joy, fear, trust). We also analyzed sentiment trends to identify some patterns in different sentiments over time.

We found that positive sentiment rises slightly before August 2001, likely as the company tried to maintain morale during internal concern. Sharp increases in negative sentiment around October-November 2001 reflect reactions to financial disclosures and Dynegy's failed acquisition. After Enron's bankruptcy in December 2001, positive sentiment declines sharply, and overall communication decreases. This shows that the emotions in Enron's emails did change before and after the negative news appeared, and they were successfully captured by NLP.

3. How did the contents or behaviors of managers who want to reduce risk differ from aggressive managers in the emails?

Before we start to analyze the Enron scandal in depth, we want to know what happened inside Enron, which will help us conduct further research.

The cluster analysis reveals 3 main communication themes within Enron. Cluster 0 centers on high-level financial and strategic discussions, potentially containing evidence of complex financial structures that concealed losses and inflated profits. Cluster 1 focuses on technical operations and data analysis for energy market trading, likely providing insights into price manipulation and inflated earnings. Cluster 2 reflects routine management communications that ensured employees followed company directives and maintained the appearance of normal operations. These clusters offer key directions for further investigation into the scandal's financial and managerial practices.

Based on the results of cluster analysis, we can conclude that the Enron scandal is highly related to the topic of financial fraud. Based on this, we can divide the people found through SNA or Isolation Forest into two major camps.

Vince J Kaminski, Jeff Dasovich, Kay Mann, Richard Shapiro and Sara Shackleton belong to the prevention and warning camp. Most of them are responsible for risk, legal compliance or government communication, and they had raised objections before the Enron scandal broke out.

Steven J Kean, Mike McConnell, James D Steffes, Richard B Sanders, David W Delainey, Ken Lay, Kitchen Louise, Tana Jones, Mark Taylor, Carol St Clair belong to the aggressive camp. Most of them participated in the decision-making that led to Enron's bankruptcy, such as document approval, energy trading, and corporate decision-making. Among them, Ken Lay and David W Delainey need to bear legal responsibility due to their high degree of participation.

What's interesting, Vince, as CRO, often raised objections to senior management, and his abnormal peaks and emotional peaks were positively correlated with historical events. In contrast, Ken Lay, as CEO, had no abnormal performance for most of the time, until the company went bankrupt, when he generated many abnormal emails and experienced drastic emotional fluctuations.

Reference

[1] https://www.cs.cmu.edu/~./enron/

[2] https://en.wikipedia.org/wiki/Enron

[3] http://luizschiller.com/enron/

[4] haown1992/Enron: 机器学习建模分析 - 从安然公司邮件中发现欺诈证据

[5] Diesner J, Frantz T L, Carley K M. Communication networks from the Enron email corpus “It's always about the people. Enron is no different”[J]. Computational & Mathematical Organization Theory, 2005, 11: 201-228.

[6] Diesner, J., & Carley, K. M. (2005, April). Exploration of communication networks from the Enron email corpus. In *SIAM International Conference on Data Mining: Workshop on Link Analysis, Counterterrorism and Security, Newport Beach, CA* (pp. 3-14).

[7] Peterson, K., Hohensee, M., & Xia, F. (2011, June). Email formality in the workplace: A case study on the Enron corpus. In *Proceedings of the Workshop on Language in Social Media (LSM 2011)* (pp. 86-95).