Contents

[Approach and Methodologies 1](#_Toc203368112)

[1. Data Loading and Initial Inspection 1](#_Toc203368113)

[2. Sentiment Labeling using Pretrained Transformer Model 1](#_Toc203368114)

[3. EDA: 2](#_Toc203368115)

[4. Employee Sentiment Score: 2](#_Toc203368116)

[5. Flight Risk Identification: 3](#_Toc203368117)

[6. Feature Engineering 3](#_Toc203368118)

[7. Text Preprocessing 3](#_Toc203368119)

[8. Feature Selection and TF-IDF Vectorization 3](#_Toc203368120)

[9. Outlier Detection and Skewness 4](#_Toc203368121)

[10. Feature Scaling 4](#_Toc203368122)

[11. Model Development 4](#_Toc203368123)

[12. Model Evaluation 4](#_Toc203368124)

## Approach and Methodologies

### 1. Data Loading and Initial Inspection

* Loaded data from an Excel file containing columns: Subject, Body, Date, and From.
* Checked sender frequencies and basic data statistics.

### 2. Sentiment Labeling using Pretrained Transformer Model

* Used HuggingFace's cardiffnlp/twitter-roberta-base-sentiment model.

**Reason to choose Bert:**

Data is mostly formal, mild in tone, and context-sensitive.

VADER and TextBlob would both struggle with subtle emotions.

* Tokenized and classified each message body using a custom function.
* Assigned sentiment labels:
  + **Positive** → 2
  + **Neutral** → 1
  + **Negative** → 0

### 3. EDA:

* Convert date column to datetime
* Created a new year\_month feature

Created following visuals:

* Email Count per Month Plot – bar plot
* Top 3 Email Senders Plot – bar plot
* Email Sentiment Distribution – pie chart
* Sentiment Distribution per Sender Plot – bar plot
* Message Lengths by Sentiment – bar plot
* Message Count – histogram

**Key findings:**

1. Almost same number of emails sent per month.
2. Max Email Count was in 2011-04
3. Top 3 Email senders : 1. [patti.thompson@enron.com, 2](mailto:patti.thompson@enron.com,%202). [john.arnold@enron.com](mailto:john.arnold@enron.com), 3.[lydia.delgado@enron.com](mailto:lydia.delgado@enron.com)
4. Highest Sentiment: neutral
5. Least Sentiment: negative
6. Most neutral email sender: lydia.delgado@enron.com
7. Most positive email sender: johnny.palmer@enron.com
8. Most negative email sender: [bobette.riner@ipgdirect.com](mailto:bobette.riner@ipgdirect.com)
9. Positive sentiment has highest message length compared to other. Can help in predicting sentiment score.
10. Neutral sentiment has highest message count. This dominant trend can help in predicting sentiment score.

### 4. Employee Sentiment Score:

* Emails were grouped by [from, year\_month], and their sentiment scores were summed to calculate a monthly\_sentiment\_score.
* Created top3\_positive and top3\_negative employees per month based on sentiment score:
* Sorts by month first, then descending score for top\_positive sentiment and ascending for top\_negative
* Groupby year\_month and keep top 3 rows for each month
* Keeps only the top 3 (or bottom 3) rows for each month.
* Alphabetical sorting after filtering.

Plotted barplots to visualize Top 3 Positive Employees per Month and Top 3 Negative Employees per Month.

### 5. Flight Risk Identification:

* Filtered negative sentiment and group by from
* Sort by date in ascending order.
* For each message, count how many other negative messages (including the current one) fall within the next 30 days by subtracting the current date from all other dates and checking if the difference is between 0 and 30 days.
* If any employee has 4 or more negative emails within a 30-day period, they are added to the flight risk list.

### 6. Feature Engineering

* Extracted new features:
  + message\_length: Character length of the body.
  + word\_count: Number of words in the body.
  + message\_count: Number of messages per employee per month.

### 7. Text Preprocessing

* Applied preprocessing using NLTK:
  + Lowercasing
  + Tokenization
  + Stopword removal
  + Punctuation removal
  + Lemmatization
* Stored cleaned text in a new column body\_cleaned.

### 8. Feature Selection and TF-IDF Vectorization

* Selected Features “message\_length,message\_count,word\_count” to train model
* Applied TfidfVectorizer on body\_cleaned to extract top 500 features and converted it to DataFrame
* Concatenated tfidf with selected features.

### 9. Outlier Detection and Skewness

* Detected outliers in message\_length and word\_count using **boxplots**.
* Checked skewness of numeric features.
* Applied log1p to handle right-skewed
  + message\_length
  + word\_count

### 10. Feature Scaling

* Applied **RobustScaler** to handle skewed and outlier-heavy features:
  + message\_length
  + word\_count
* Used **StandardScaler** for mild skewed message\_count.

### 11. Model Development

* **Linear Regression** was used to predict the sentiment score (-1, 0, 1).
* Dataset was split into **train-test sets** (using train\_test\_split).

### 12. Model Evaluation

* Evaluation metrics used:
  + MAE (Mean Absolute Error): 0.316

Used MAE due to outliers.

On average, model’s prediction error is around 0.316 sentiment points.

* + R² Score: 0.296

29.7% variance explained by model

* Actual vs Predicted Sentiment Scores Plot

Observation:

* The model struggles most with negative sentiments (predictions are farther from -1).
* It performs better for positive sentiments, predicting scores closer to +1.
* Neutral predictions are somewhat accurate.