**Employee Sentiment Analysis – Final Report**

**Task 1: Sentiment Labeling**

**Approach and Methodology**

To assess employee sentiment from messages in the test dataset, we used a transformer-based language model to classify the emotional tone of each message. Our approach involved the following steps:

1. **Model Selection:**
   * Initially considered using a Hugging Face model (`cardiffnlp/twitter-roberta-base-sentiment`) due to its simplicity and ability to classify into three categories (positive, negative, neutral). While this model works well for short, informal messages (e.g., tweets), it may not generalize as well to more formal, nuanced employee emails.
   * **Final approach:** Decided to use OpenAI's `gpt-3.5-turbo` via the OpenAI API. This model is more context-aware and better suited for professional communications such as emails, where tone can be subtle or indirect. (~$0.20 cost)
2. **Data Loading and Preprocessing:**
   * Loaded the dataset test.csv into a Pandas DataFrame.
   * Combined the Subject and body columns into a new column named message to create a unified text field for analysis.
   * Preprocessing steps included:
     + Removing empty or NaN messages.
     + Ensuring text fields were proper strings.
     + Trimming long messages to 512 characters (maximum input for most transformer models).
     + Cleaning URLs and redundant spaces using regular expressions.
3. **Sentiment Analysis with Batched Inference:**
   * Instead of using a pre-trained transformer model, we leveraged an OpenAI large language model (GPT-3.5 Turbo) to classify employee messages by sentiment.
   * We set up the classification as a prompt-based task by providing the model with a system message instructing it to label each message as Positive, Neutral, or Negative.
   * Each message consisted of the email's subject and body combined into a single text input.
   * The classify\_sentiment function sends each message to the GPT-3.5 Turbo model via the OpenAI API, requesting a single-word sentiment label as output.
   * To ensure consistent results, the model's temperature parameter was set to zero, reducing randomness in the responses.
   * Sentiment labels returned by the model were appended to the dataset in a new column called sentiment for downstream analysis.
4. **Output Storage:**
   * The augmented dataset, including the sentiment labels, was saved as a new CSV file sentiment\_df.csv.

**Task 2: Exploratory Data Analysis (EDA)**

**Approach and Methodology**

The goal of this task was to understand the structure and patterns in the sentiment-augmented employee message dataset (sentiment\_df.csv). Our methodology included the following steps:

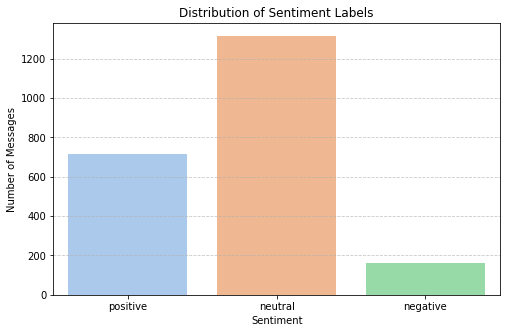
1. **Data Structure Examination:**
   * Checked for missing values across all columns.
   * Inspected overall dataset shape and column data types.
   * Reviewed summary statistics to identify high-level characteristics (e.g., most frequent senders, subject lines, dates).
2. **Sentiment Label Distribution:**
   * Analyzed the proportion of each sentiment class (positive, negative, neutral).
   * Identified sentiment imbalance which may affect downstream modeling.
3. **Temporal Trends:**
   * Converted the date column to datetime format.
   * Grouped messages by month and plotted sentiment trends to observe changes over time.
4. **Behavioral Patterns and Anomalies:**
   * Explored sentiment proportions by individual senders.
   * Investigated subject line patterns across sentiment categories.
   * Identified repeated content or missing subject patterns.

**Key Findings from EDA**

1. **Dataset Structure and General Statistics**
   1. **Total emails:** 2,191
   2. **Most common subject:** “(No Subject)” (141 times)
   3. **Most frequent sender:** [lydia.delgado@enron.com](mailto:lydia.delgado@enron.com) (284 emails)
   4. **Most emails sent on:** 7/1/2011 (9 emails)
   5. **Most common sentiment:** *Neutral* (1,315 emails, \~60%)
   6. **Frequent message content:** Many are empty or repeated
   7. **Unique senders:** Only 10 employees in total

**2. Sentiment Label Distribution**

* **Neutral:** 65.3% of messages (1,430 emails)
* **Positive:** 28.1% (616 emails)
* **Negative:** 6.7% (145 emails)



This indicates a significant class imbalance — most messages were neutral in tone, while negative sentiments were relatively rare.

**3. Sentiment Trends Over Time**

A monthly sentiment trend line plot revealed:

A graph of different colored lines

AI-generated content may be incorrect.

* **Neutral sentiment** consistently dominated across all months, typically ranging from **50 to 68 messages/month**.
* **Positive sentiment** showed an **upward trend**, increasing from ~20 to ~30 messages/month over the year.
* **Negative sentiment** remained consistently low, seldom exceeding 10 messages/month but with small **spikes in June and April**.

These patterns may reflect cyclical organizational factors (e.g., end-of-quarter reviews, holidays, or reporting periods).

**4. Sentiment by Sender**

We analyzed the top 10 most active senders and their sentiment distribution:

A graph with green and blue bars

AI-generated content may be incorrect.

* **Kayne Coulter** and **Bobette Riner** had the **highest proportion of negative messages (~9%)**, suggesting possible concern or dissatisfaction in tone.
* **Eric Bass** had the **lowest negative rate (3.8%)** and a relatively **high positive rate (30%)**, reflecting a more positive or optimistic communication style.
* Despite variation, **neutral sentiment remained the dominant tone** across all employees.

This may indicate **personality-driven or role-related communication patterns** among staff.

**5. Common Subject Lines by Sentiment**

An analysis of the most common subject lines by sentiment class revealed:

A graph with colorful squares and text

AI-generated content may be incorrect.

* **Neutral:** Procedural or administrative topics such as "Interview Schedule" and "Master Power Contracts".
* **Positive:** Congratulatory or celebratory themes including "Congratulations" and "Bearkadette Schedule".
* **Negative:** Often issue-related, such as "MISO proposed $1000 penalty" and "Out of the Office".

Subjects like "(No Subject)" and "Re:" were prevalent across all sentiment types, indicating that reply chains or undeclared topics are common in employee communications.

**Task 3: Employee Sentiment Score Calculation**

**Approach and Methodology**

The objective of this task was to quantify each employee's monthly communication sentiment by assigning numeric scores to their messages. This allowed us to track how an employee’s sentiment fluctuated over time. The methodology was as follows:

1. **Scoring Criteria:**
   * **Positive sentiment:** +1
   * **Negative sentiment:** -1
   * **Neutral sentiment:** 0
2. **Implementation Steps:**
   * Mapped each message’s sentiment label to a numerical score using a predefined dictionary.
   * Parsed the date column to extract the corresponding **year-month period** for each message.
   * Grouped the data by **employee email** and **month**, then summed the sentiment scores to get each employee's **monthly sentiment score**.
   * Renamed and sorted the resulting DataFrame for easier readability and analysis.

**Employee Sentiment Score Calculation Details**

* **Unique Employees Analyzed:** 10 (based on the from field)
* **Date Parsing:** Ensured all message dates were properly converted using pd.to\_datetime with error coercion for consistency.
* **Period Extraction:** Created a new month column by extracting the month and year from each message’s timestamp.
* **Grouping and Scoring:** Used groupby(['from', 'month']) followed by a sum() on the sentiment score column to compute each employee’s total sentiment per month.

**Key Insights**

* The **monthly sentiment score** offers a **quantifiable measure of employee tone over time**.
* Positive values suggest a predominantly encouraging or appreciative tone that month.
* Negative values may indicate dissatisfaction, stress, or interpersonal tension.
* Neutral scores may reflect procedural or emotionless communication.
* These scores form the basis for identifying **trends**, **communication shifts**, or **potential red flags** in employee behavior.
* A screenshot of a game

  AI-generated content may be incorrect.
* A Heatmap of monthly employee scores:
* Graph Insights:
  + Chart – heat-map of monthly e-mail sentiment scores (−2 = very negative → 10 = very positive) for 10 Enron employees over two years; warmer reds = positive, cool blues = negative.
  + Typical tone – most tiles sit in the mildly positive 1-6 range, showing generally upbeat correspondence.
  + Extremes & variability – rare negative pockets (e.g., mid-2011) and two standout highs of 10 (Aug 2010, Feb 2011); some employees swing more than others.
  + Big picture – sentiment stays largely positive across time and staff, with only isolated dips into negativity.

**Task 4: Employee Ranking**

**Objective**

The goal of this task was to identify and highlight top-performing and potentially at-risk employees based on their monthly sentiment scores. This insight can help HR or management identify trends in employee morale and communication tone over time.

Specifically, two ranked lists were created for each month:

* **Top Positive Employees**: 3 employees with the highest sentiment scores.
* **Top Negative Employees**: 3 employees with the lowest sentiment scores.

**Approach and Methodology**

1. **Input Data**: Used the monthly\_scores DataFrame from Task 3, which contains:
   * employee\_email
   * month
   * monthly\_sentiment\_score
2. **Ranking Logic**:
   * For each month, employees were **sorted first by sentiment score (descending)** and then **alphabetically by email** to break ties.
   * The **top 3 scores** formed the **positive list**; the **bottom 3 scores** formed the **negative list**.
   * A new column called rank\_type was added to distinguish between the two lists.
3. **Function Design**:
   * Created a function rank\_monthly\_employees(df) that extracts the top and bottom employees from a monthly slice of the data.
   * Applied this function across all unique months using groupby('month').apply(...) to construct a single consolidated rankings DataFrame.
4. **Presentation**:
   * Final rankings were displayed for each month in a clean tabular format showing:
     + rank\_type
     + employee\_email
     + monthly\_sentiment\_score

**Key Insights**

* This ranking system provides **early indicators of positive leadership or potential dissatisfaction**.
* Employees appearing frequently in the **Top Negative** list may warrant closer attention or outreach.
* **Consistently Top Positive** employees can be considered for recognition or leadership development.
* **Monthly granularity** allows for temporal trend monitoring—e.g., sudden shifts in sentiment ranking may correspond with organizational events or team changes
* A graph of a number of people

  AI-generated content may be incorrect.
* Graoh Insights:
  + Top Positive sentiment scores are consistently high, typically between 3 and 7, indicating strong positive communication across months.
  + Top Negative scores are low in magnitude, often just -1 to -2, suggesting limited extreme negativity even among the most negative employees.
  + Peak positivity occurred in May 2010, January 2011, and March 2011, with sentiment scores above 6.
  + Notable spikes in negativity appear in April 2011 and December 2011, possibly signaling workplace tension or dissatisfaction during those periods.
  + Consistent trends in top communicators suggest reliable patterns in employee sentiment, useful for monitoring morale and engagement.

**Task 5: Flight Risk Identification**

**Objective**

The aim of this task was to identify employees who may be at risk of leaving the company, based on the **frequency of negative communications** over time. Early identification of flight risks allows HR and management to proactively address concerns, intervene with support, or better understand underlying issues in team dynamics.

**Definition of Flight Risk**

An employee is flagged as a **flight risk** if they send **four or more negatively labeled messages** within **any rolling 30-day period**, regardless of the month or sentiment score magnitude.

**Methodology**

1. **Filter for Negative Sentiment**  
   The dataset was filtered to retain only emails labeled with a **‘negative’** sentiment.
2. **Sort for Temporal Analysis**  
   These records were sorted by sender (from) and date to prepare for rolling time window analysis.
3. **Apply Rolling 30-Day Check**  
   For each employee:
   * Extract the list of dates of their negative emails.
   * For each date, define a rolling 30-day window.
   * Check if at least 4 emails occurred within any such window.
   * If this condition is met, the employee is added to the **flight risk list**.
4. **Store and Present Results**
   * Flagged emails were compiled into a flight\_risk\_df DataFrame.
   * A flight\_risk = True column was added to mark these employees.
   * Results were displayed as a concise table.

**Key Findings**

The following employees were flagged as potential **flight risks**:

| **employee\_email** | **flight\_risk** |
| --- | --- |
| don.baughman@enron.com | True |
| john.arnold@enron.com | True |

* These employees had **4+ negative communications** within a **30-day window**, suggesting repeated expressions of dissatisfaction or stress.
* Some of these individuals also ranked highly in the **monthly negative sentiment rankings** (Task 4), reinforcing the concern.

An analysis of these individuals negative labeled messages over time:

A graph with numbers and dots

AI-generated content may be incorrect.

* **Clusters of points** close together horizontally (same employee): This suggests the employee sent **multiple negative messages in a short period** — likely why they were flagged as flight risks.
* **Long gaps** between messages: Could indicate cooling off or recovery in sentiment.
* **High density in a short time**: May correlate with acute stress, conflict, or burnout period.
* **Consistent spread over time**: Indicates a more **chronic pattern** of negativity.

**Task 6: Predictive Modeling**

**Objective:**

Develop a **linear regression model** to predict an employee’s **monthly sentiment score** based on message characteristics and volume.

**Methodology:**

1. **Feature Engineering**:
   * Added per-message features:
     + word\_count: Number of words in each message.
     + char\_count: Number of characters in each message.
   * Created binary flags for message sentiment:
     + is\_positive, is\_negative, is\_neutral.
2. **Monthly Aggregation**:
   * Grouped data by employee and month to calculate:
     + monthly\_sentiment\_score
     + Message statistics: message\_count, avg\_word\_count, avg\_char\_count, total\_words, total\_chars
     + Sentiment ratios: positive\_ratio, negative\_ratio, neutral\_ratio
3. **Model Preparation**:
   * Defined features:
     + Message volume and length: message\_count, avg\_word\_count, avg\_char\_count, total\_words, total\_chars
     + Sentiment distribution: positive\_ratio, negative\_ratio, neutral\_ratio
     + Additional features: words\_per\_message, chars\_per\_word, sentiment\_intensity
   * Target variable:
     + monthly\_sentiment\_score
   * Removed outliers based on z-score filtering on the target variable.
   * Performed an 80/20 train-test split.
   * Standardized features using StandardScaler.
4. **Model Training and Evaluation**:
   * Trained a **Linear Regression** model.
   * Performance metrics on test set:
     + **Mean Squared Error (MSE):** 1.52
     + **R-squared (R²):** 0.54
5. **Feature Importance**:
   * Top positive contributors:
     + avg\_char\_count (+1.56)
     + total\_chars (+0.83)
     + sentiment\_intensity (+0.49)
     + message\_count (+0.45)
     + positive\_ratio (+0.37)
   * Negative contributors:
     + negative\_ratio (−0.45)
     + chars\_per\_word (−0.37)
     + words\_per\_message (−0.89)
     + avg\_word\_count (−0.89)
6. **Residual Analysis**:
   * Residual plot shows some clustering and patterns around predicted sentiment scores between 0–4:

A graph with blue dots and red line

AI-generated content may be incorrect.

* + Suggests that the linear model may miss some feature interactions and non-linear relationships:

A graph with green and red dots

AI-generated content may be incorrect.

1. **Model Improvement Attempts**:
   * Tested regularized linear models:
     + Ridge Regression R²: 0.58
     + Lasso Regression R²: 0.69 (better fit)
   * Explored a **Random Forest Regressor** to capture non-linearity:
     + Random Forest MSE: 0.13
     + Random Forest R²: 0.96 (significantly improved performance)
2. **Random Forest Insights**:
   * Residuals vs. predicted values show no obvious patterns, indicating a strong model fit:

A graph with blue dots and red line

AI-generated content may be incorrect.

* + Actual vs. predicted scatterplot aligns closely with the perfect prediction line:

A graph with orange dots

AI-generated content may be incorrect.

* + Feature importance analysis identifies the key drivers used by the model:

A graph with a bar graph

AI-generated content may be incorrect.

1. **Model Persistence**:
   * Saved the trained Random Forest model as random\_forest\_model.pkl using pickle for future use.

**Summary:**

The linear regression model demonstrated moderate predictive ability but struggled to capture complex relationships in the data, as indicated by residual patterns. Regularized linear models (Lasso and Ridge) improved performance, with Lasso showing better results through implicit feature selection. The Random Forest model greatly outperformed linear models by capturing non-linear dependencies, suggesting it is more suitable for this task.