

Employee Sentiment Analysis

LLM Final Assessment Report

Jason Clibanoff

Overview

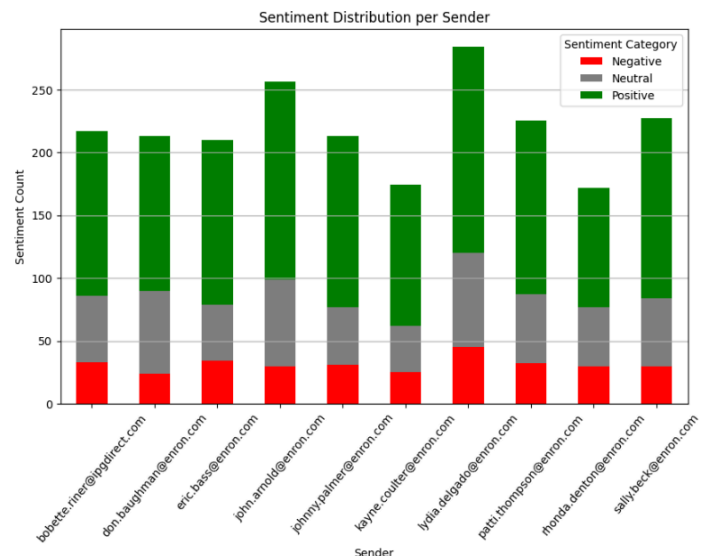
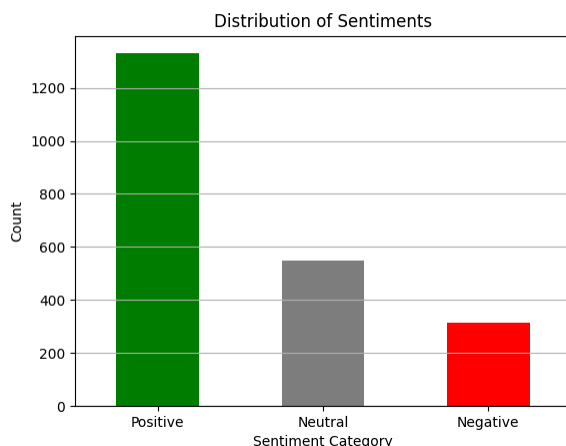
In this project a dataset of 2191 emails from ten unique employee email addresses is analyzed to determine employee sentiment. The project is broken up into six tasks: sentiment labelling, exploratory data analysis, employee score calculations, employee ranking, flight risk evaluation, and predictive modelling.

Task 1. Sentiment Labelling

For this task, I utilized TextBlob's sentiment analysis feature, specifically the polarity score. I made a function to extract the sentiment from the subject and body of each email, and averaged the two values overall sentiment of the email. Based on the result, messages were tagged as positive (if above zero), negative (if below zero), or neutral (if exactly zero). These values were then added to the DataFrame in a new "sentiment" column.

Task 2. Exploratory Data Analysis (EDA)

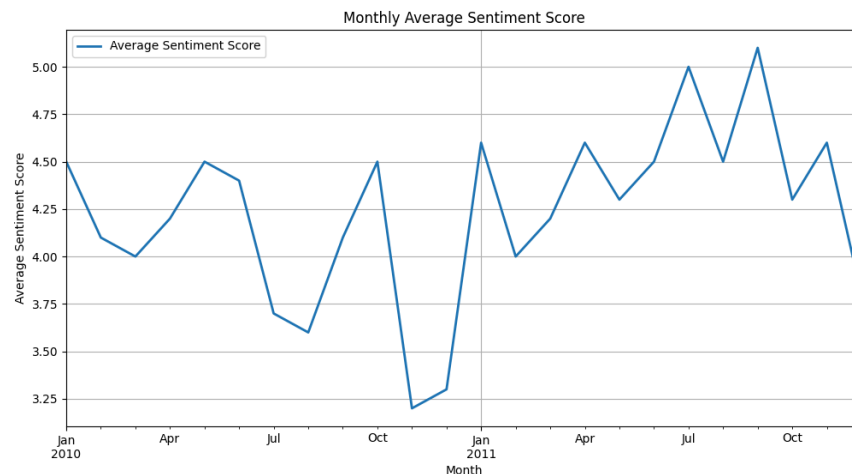
From just using the describe() function, it can be seen that the most frequent sentiment is positive, and the most emails were sent on July first 2011, and lydia.delgado@enron.com sent the most emails overall. The overall distribution of sentiment shows the vast majority of emails sent had a positive sentiment, and breaking it down to look at each individual reflects this as well.



I grouped by email address and sentiment, and created a pivot table called sentiments_pivot to see a count of how many emails of each sentiment category each employee sent. Next I grouped the data by month, email address, and sentiment rating and made a pivot table called monthly_pivot. With this monthly_pivot I was able to look at trends for average and individual sentiments over time.

Task 3. Employee Score Calculation

To apply a sentiment score to each message, I developed a function to assign values of 1, 0, or -1 based on each row's corresponding sentiment label, and applied this to my DataFrame. I was then also able to apply this logic to my monthly_pivot, and calculated each employee's monthly sentiment score. The plot below shows the average monthly sentiment score over time.



Looking at the overall sentiment trends for all employees, sentiment tends to drop towards the end of each year, reaching an all time low in November 2010, and trending in the same direction at the end of 2011. Average sentiment score reached an all time high in September 2011.

Task 4. Employee Ranking

To determine the most positive and negative employees, I referred back to my sentiments_pivot, which was grouped by employee and sentiment. I created a new column showing the total sentiment score for each employee, then sorted by highest score to find the most positive employees, and the lowest score to find the most negative employees. The results for all time most positive and negative employees can be seen in the table below.

Most Positive Employees (Highest All-time Sentiment Scores)

Email Address	Sentiment Score
john.arnold@enron.com	127
lydia.delgado@enron.com	119
sally.beck@enron.com	113

Most Negative Employees (Lowest All-time Sentiment Scores)

Email Address	Sentiment Score
rhonda.denton@enron.com	65
kayne.coulter@enron.com	87
eric.bass@enron.com	97

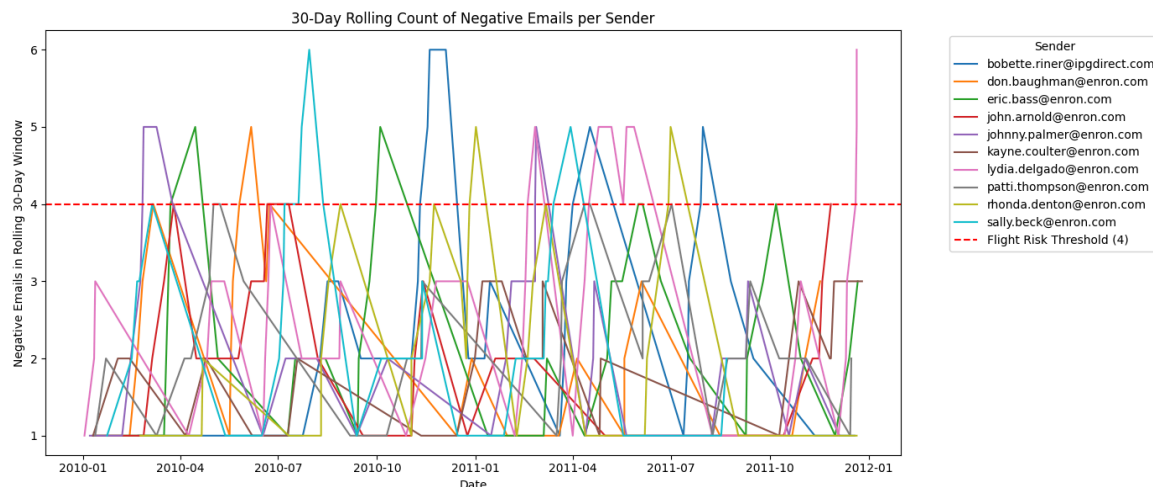
Next, I developed two functions to identify the highest and lowest sentiment scores for any given month. The first extracts data for a specified month from `monthly_pivot`, selects the top three senders based on the highest sentiment scores (breaking ties alphabetically), and returns a ranked DataFrame with email addresses and scores, using a dynamic column name for the selected month. The second function mirrors this process but retrieves the bottom three senders.

Task 5. Flight Risk Identification

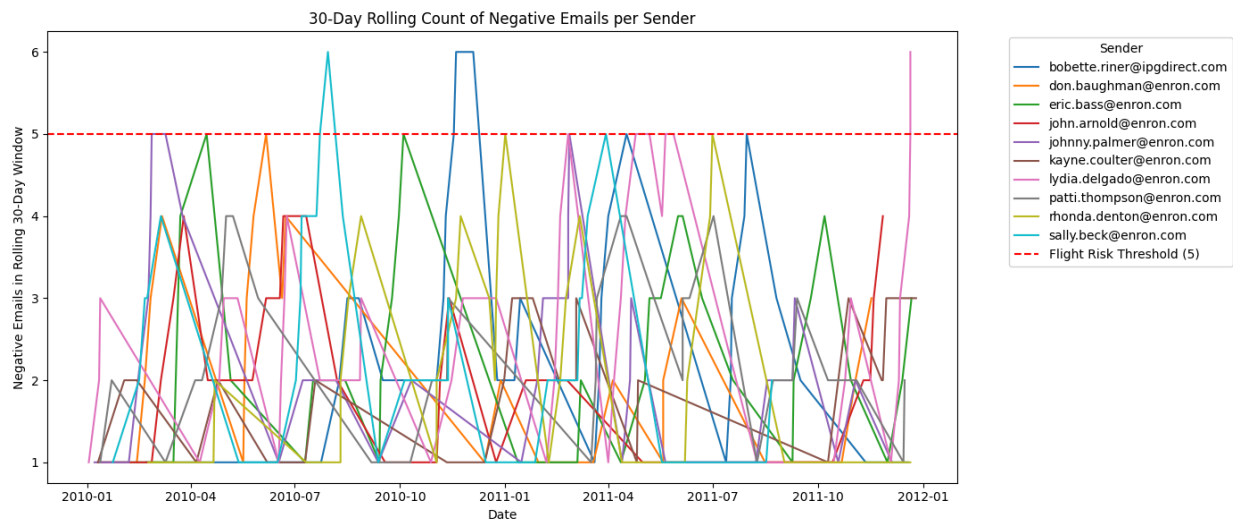
To identify employees likely to leave, I developed code that filters the dataset for messages with a sentiment score of -1 and groups them by sender to track the volume of negative messages each employee sends. For each sender, it computes a 30-day rolling count of negative messages, using message timestamps as the index. Any sender with at least four negative messages within such a window is flagged as suspicious.

Checking for flight risk with the original criteria of anyone having four negative emails within any 30 day span was not very effective, as nearly everyone met that criteria.

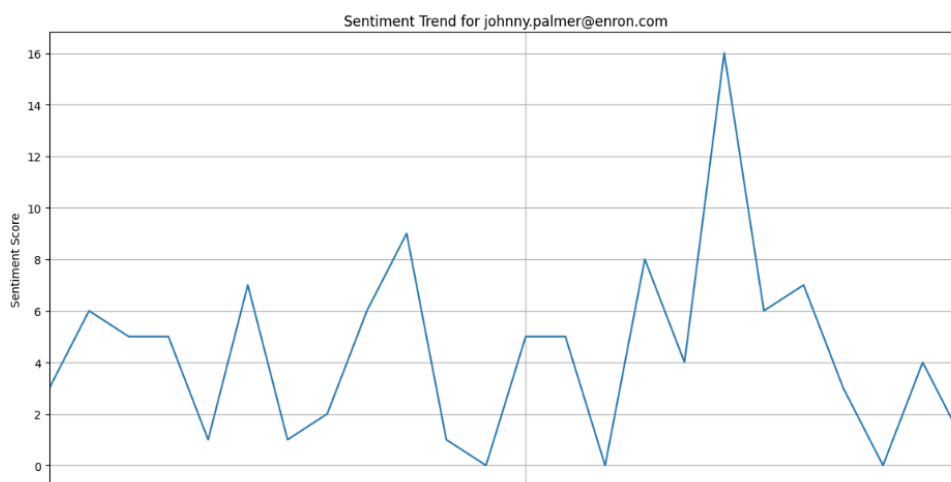
Kayne.coulter@enron.com was the only employee initially not ruled a flight risk using this metric. Bobette.riner@ipgdirect.com, don.baughman@enron.com, eric.bass@enron.com, john.arnold@enron.com, johnny.palmer@enron.com, lydia.delgado@enron.com, patti.thompson@enron.com, rhonda.denton@enron.com, and sally.beck@enron.com had at least four negative emails within a rolling 30 day period at some point.



Raising the threshold to more than five negative messages in a rolling 30 day span, only sally.beck@enron.com, bobette.riner@ipgdirect.com, and lydia.delgado@enron.com are flagged.



This metric might also be flawed, as people who just send the most emails are more likely to have a few more negative ones mixed in. Lydia.delgado@enron.com is flagged even though she ranks as top two in all time sentiment score, just because she sent more emails than everyone else. It might be more accurate to instead look at who is trending downwards in recent months. For example, looking at a plot of johnny.palmer@enron.com's monthly sentiment scores over time, we can see that in recent months his sentiment ratings have been plummeting.



Task 6. Predictive Modelling

Next I made a model predicting monthly sentiment scores, but before creating my model, I added some more features to my DataFrame. I used nltk's `opinion_lexicon` and `word_tokenizer` to find the number of positive and negative words in each email, as well as the percentage of words in each email that were positive or negative. I also added a total word count for the subject and body of each email, and a monthly count of emails for each sender. I made my date columns numeric, replacing it with a count of how many months from the start of the data, and dropped my non-numeric columns. Then I used the sklearn library for splitting my data into training and testing data, and for scaling my data. I created a LinearRegression model, trained it on my `X_train_scaled`, and used my `X_test_scaled` to make predictions.

The resulting model ended up with a mean squared error of 4.5, and an R^2 score of 0.53, meaning the model is able to explain about 53% of the sentiment variation. Plotted below is the predicted vs actual monthly sentiment scores. The model still has room for improvement, and perhaps another deep learning model could bring stronger results. This model struggles to accurately predict some of the lowest sentiment scores, but generally is able to show the linear progression of monthly sentiment as it relates to the other features in the data set.

