Replication of Loughran & McDonald Figure 1

Team Members:
Srushti Shinde (ss17454)
Huilin Zhang (hz3455)

1. Introduction

This report details the replication of Loughran & McDonald's Figure 1 by analyzing the correlation between excess returns and sentiment metrics from S&P 500 companies' quarterly report

2. Data Collection

For this analysis, we collected excess returns data and sentiment metrics from publicly traded S&P 500 companies using the CRSP database via WRDS and 10-Q filings from the SEC's EDGAR system. This comprehensive dataset spans five years, covering the period from 2018 to 2022, and includes a total of 10,000 filings

3. Methodology

Following are the files which we used for sentiment analysis of the 10Qs over a 5 year period and using Loughran & McDonald's dictionary.

a. EDGAR_Pac.py
It generates MASTERINDEX objects which act as metadata holders for SEC filings. This script includes modifications to specifically include S&P 500 companies' 10-Q forms by filtering based on CIKs retrieved from a predefined ticker list.

b. EDGAR_DownloadForms.py
The script is designed to download 10-Q filings from the SEC's EDGAR database for S&P 500 companies, identified using CIK numbers. It adapts the timing constraints and sleep intervals to expedite the download process, fetching the required forms within the specified range of dates.

c. Mda.py
Aimed at extracting Management's Discussion and Analysis (MDA) sections from 10-Q filings, this script uses text parsing techniques to isolate and analyze narrative sections that are rich in sentiment and forward-looking statements.

d. Load_MasterDictionary.py
This utility script loads a predefined linguistic dictionary, which is crucial for sentiment analysis in the financial context. The dictionary categorizes words

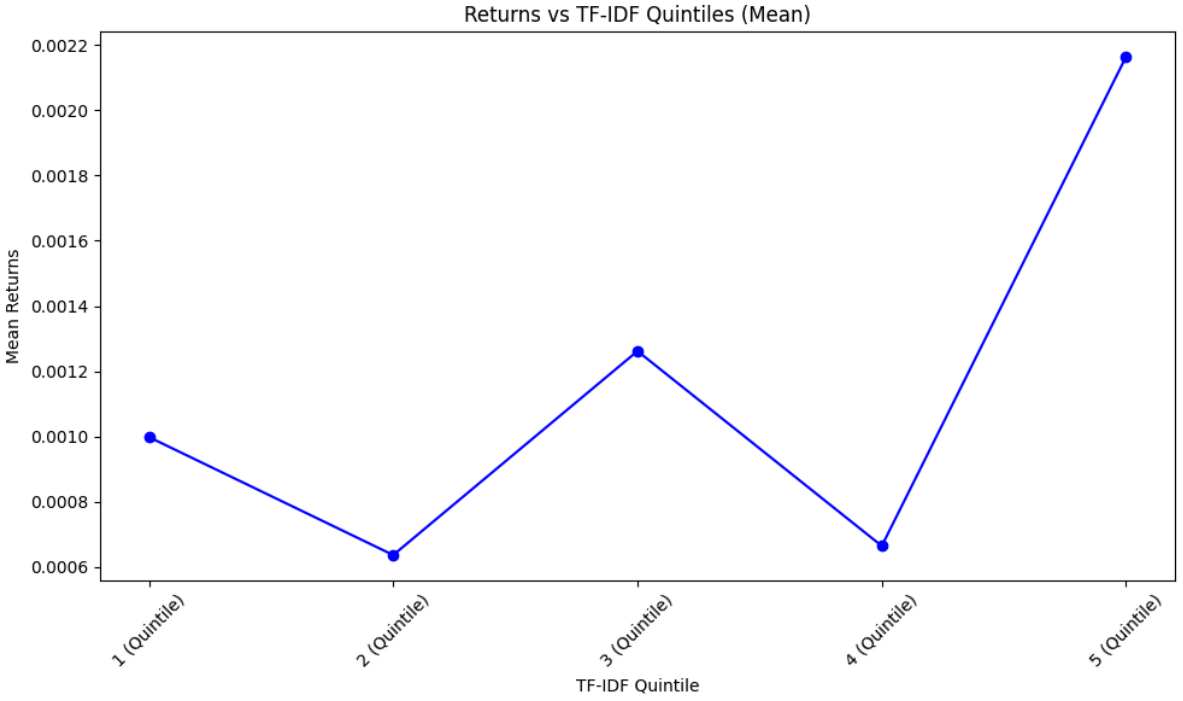
into various sentiment groups, facilitating more nuanced sentiment extraction from financial texts.

e. Generic_Parser.py
This script is enhanced to not only count negative words but also calculate the tf-idf weight of each word, by assigning a unique number to each negative word in the loaded dictionary. It then constructs matrixes to analyze document word frequencies and their inverse document frequency for further sentiment analysis.

f. Excess_return_data.py
This script calculates the excess returns from the daily stock prices retrieved through the CRSP database. It aligns the returns data with the release dates of 10-Q reports to analyze the impact of report sentiments on stock performance.

g. Return.py & Plot_result.py
This script processes data to analyze the relationship between TF-IDF (a metric used to evaluate text importance) and financial returns. Then, it groups the data into quintiles based on TF-IDF values and calculates the mean return for each quintile. Finally, the script generates a line plot to visualize how mean returns vary across TF-IDF quintiles.

4. Results



Mean returns for each quintile:

	tfidf_quintile	returns
0	1 (Quintile)	0.000998
1	2 (Quintile)	0.000637
2	3 (Quintile)	0.001262
3	4 (Quintile)	0.000665
4	5 (Quintile)	0.002163

5. Conclusion and Further improvements

1. The results indicate a general positive relationship between TF-IDF quintiles and mean returns, but it's not perfectly consistent.

2. The second and fourth quintiles show dips in returns, which could be due to various factors.
    a. One possible reason is that the dictionary used to calculate TF-IDF may not have been updated to reflect recent changes in market sentiment, especially during the COVID-19 pandemic.
    b. The pandemic introduced significant volatility and new economic conditions, which may not have been fully captured by an older dictionary, leading to less accurate assessments of text relevance and its connection to returns.

3. Additionally, the success rate of extracting MDA (Management Discussion and Analysis) sections from the text files is around 80%. This is largely due to inconsistencies in the format of the documents and occasional encoding errors.

4. To achieve a more accurate analysis, the remaining 20% of the documents will require manual adjustments to correct formatting issues and handle any garbled text. This would help ensure more precise extraction and improve the overall accuracy of the report.