Final Project

Applied Data Science in Finance

The similarity of ECB's communication
(Diego Amaya, Jean-Yves Filbien)
Replication & Extension

Bruno Sciascia, Thomas Grangaud, Titouan Guesdon, Nicolas Blache

Contents

Introduction

This report explains step by step how we replicated and extended (up to October 2024) the paper titled *"The Similarity of ECB's Communication"* by Diego Amaya and Jean-Yves Filbien. We began by preparing our data, which involved web scrapping, data processing and cleaning, and tokenization. We then performed the sentiment and similarity analyses, and gathered additional data (e.g., MRO, STOXX), requiring further processing, to reproduce and construct the graphs and tables displayed in the original study. Beyond replicating empirical results, we extended the paper by incorporating a measure of subjectivity and employing a light transformer model for text classification.

Get the data ready

Web scraping

The first step of the project is to get the data from the European Central bank website. Instead of working with data from January 1999 to December 2013, we extend the analysis to statements disclosed until October 2024. Web scraping the website was a difficult part since it is coded with Java script dynamic content. Thus, we use a specific Selenium framework for automating web browsers and the web driver tool. We also need to use a "scroll by" method to scroll down the whole webpage. Dates were extracted from the URLs. Indeed, we struggled to scrape the dates directly from the website because we could not find the dates directly int the pages, either because they were not here or because they were always written with the location of the announcement. To counter this we took it from the URLs' string. Then we were able to get the titles, dates, and contents that we stored into an Excel database.

Data processing and cleaning

We preprocessed the dataset of ECB to prepare it for further calculations and analysis. We convert dates from a two-digit year format to a standardized four-digit format using a function that distinguishes between 20th and 21st-century dates. The dates are reformatted to YYYY-MM-DD and set as the index, enabling chronological filtering. The dataset is then filtered by the title column, retaining only entries with specific keywords (e.g., "ECB Press Conference" and "Introductory Statement") to remove rows where the document was not a monetary statement.

Duplicate entries are identified by their index and grouped by month-year to isolate potential overlaps, and a few specific erroneous dates are manually excluded. The content column undergoes targeted cleaning: the first sentence of each entry is removed if present, as containing positive words with no meaning such as "good" in "good afternoon". Common Q&A sections, identified by predefined phrases (such as for instance: "We are now ready to take your questions"), are systematically stripped. Additional columns track the length of the content before and after cleaning to verify modifications, and rows where the Q&A section was not removed are flagged for review. This process ensures the dataset is thoroughly refined, retaining only relevant and clean textual data for accurate analysis.

Tokenization

With our fully cleaned dataset, we started processing the data to add columns we need for the analysis. We tokenized the contents into lists, by removing English stop words, using the Porter stemmer like in the paper, and converting everything to lowercase words.

Compute sentiment

The sentiment analysis that we performed follows the computation of pessimism proposed by Ferguson et al. (2013) and Garcia (2013). We computed the sentiment analysis twice: the first time we removed the words added after 2013, and the second time we used the whole dictionary.

Compute similarity

We computed the Jaccard similarity function with the bigrams and iterated over the data frame to find the amount of redundant information in two successive ECB statements. $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$

We want to identify if there is a linear trend in the similarity of ECB communication statement over time, thanks to the following regression. We add control vectors such as the quarterly output gap, the change in the MRO rate, and the Harmonised Index Consumer Prices that correspond to the open market operations in the euro zone.

$$logSimilarity_i = \alpha_0 + \alpha_1 logTime_i + \beta' Controle_i + \varepsilon_i$$

Get data for other variables

Main Refinancing Operations (MRO)

Data from the European Central Bank (ECB) was extracted to compute changes in the Minimum Reserve Operations (MRO) rate. The dataset was cleaned and processed to calculate the difference in MRO rates across consecutive dates, isolating non-zero changes. These changes were then aligned with the nearest ECB announcement dates in the main data frame, ensuring temporal accuracy. Finally, the results were added as a new column, mro_change, to facilitate further analysis.

GDP data and output gap

The Hodrick-Prescott (HP) filter is a mathematical tool used in time series analysis to separate a time series into two components:

- Trend Component: Represents the long-term trend of the series.
- Cyclical Component: Represents short-term fluctuations or deviations from the trend.

The HP filter is commonly applied in macroeconomics to analyze economic data such as GDP, where the goal is to distinguish between the underlying trend (potential output) and business cycle variations.

The HP filter minimizes the following objective function:

$$\min_{\tau_t} \sum_{t=1}^{T} (y_t - \tau_t)^2 + \lambda \sum_{t=2}^{T-1} [(\tau_{t+1} - \tau_t) - (\tau_t - \tau_{t-1})]^2$$

Where:

- y_t : The observed time series (e.g., GDP).
- τ_t: The trend component of the series.
- $y_t - \tau_t$: The cyclical component.
- λ : A smoothing parameter controlling the trade-off between the smoothness of the trend and the closeness of the fit.

1. $(y_t - \tau_t)^2$: Ensures the trend closely follows the data
2. $\lambda \sum_{t=2}^{T-1} [(\tau_{t+1} - \tau_t) - (\tau_t - \tau_{t-1})]^2$: Penalizes large changes in the growth rate of the trend, enforcing smoothness.

So we applied the Hodrick-Prescott (HP) filter to decompose the time series of observed GDP into its trend (potential GDP) and cyclical (deviations from the trend) components. We started by extracting the GDP data and using the HP filter with a smoothing parameter ($\lambda = 1600$) suited for quarterly data. The filter separates the series into a smooth long-term trend, representing the potential GDP, and short-term fluctuations, representing the cyclical component. These components are then added back into the dataset as new columns. Finally, we computed the output gap—the percentage deviation of actual GDP from potential GDP—indicating whether the economy is overperforming (positive gap) or underperforming (negative gap).

HCPI data for inflation

Daily data was downloaded from FRED, specifically monthly HICP data, and preprocessed by converting the dates to a standard datetime format and filtering for entries from 1999 onward. The data was then aligned with the nearest ECB announcement dates in the main dataframe using an as-of merge. A new column, hcpi_change, was created to calculate the percentage change in HICP values, facilitating analysis of inflation dynamics relative to announcement dates.

EStoxxx 50 index for absolute CAR

Like the paper we use the daily closing values of the Euro Stoxx 50 index and use the constant mean return model to estimate the abnormal component of stock market index returns associated with the announcements. We then calculate the cumulative abnormal returns 11-day window controls for effects such as news leaks. We added the data to our main data frame by merging to the nearest date to align with the corresponding dates of announcement.

Our goal is to estimate the influence of communication similarity on market's reaction to announcements with the following regression:

$$|CAR_i| = \gamma_0 + \gamma_1 logSimilarity_i * Pessimism_i + \alpha' Controle_i + n_i$$

Empirical results

Descriptive statistics

We started looking at the evolution of similarity over time in ECB communications as shown in Fig. 1. The overall trend of the graph increased. And we have the same graph that the paper. But we have a drop in 2997. We tried to fix it but we were unable to find a reason that could match the result of the paper. The announcements made on those dates, and which make the similarity drops, are relevant and should not be deleted or modified.

We notice a significant drop in similarity starting in early 2020. We assume that the Covid-19 crisis accounts for this phenomenon, because it led to very high uncertainty about the economy, and it may be reflected in the speeches. After 2022, similarity started to rise again.

Table 1 shows all MRO changes from January1, 1999, to October 17, 2024. We found that ECB announced no change 216 times, announced and increase 28 times and announced a decrease 21 times. Among the 49 changes, 31 were of 25 basis points. We see that in 2022 and 2023, 10 out of 16 announcements were MRO rises, which could explain the rise in similarity that we found.

In table 2 we present our descriptive statistics for our variables. Over the 25 years period of study, the absolute cumulative abnormal returns mean is close to zero. The average level of pessimism is positive but also close to zero indicating a neutral tone. The output gap mean is almost zero indicating overall stability in the euro zone over the whole period.

Fig.1. Similarity measure since ECB's creation.

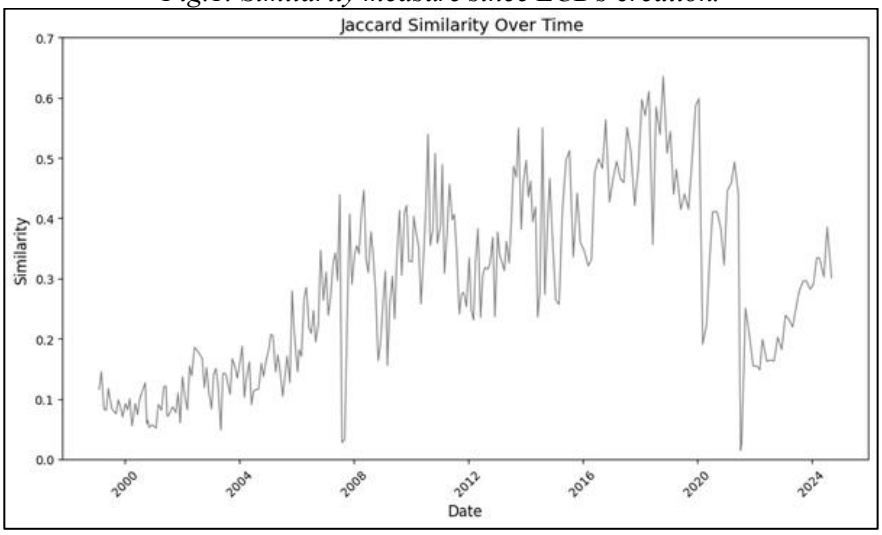
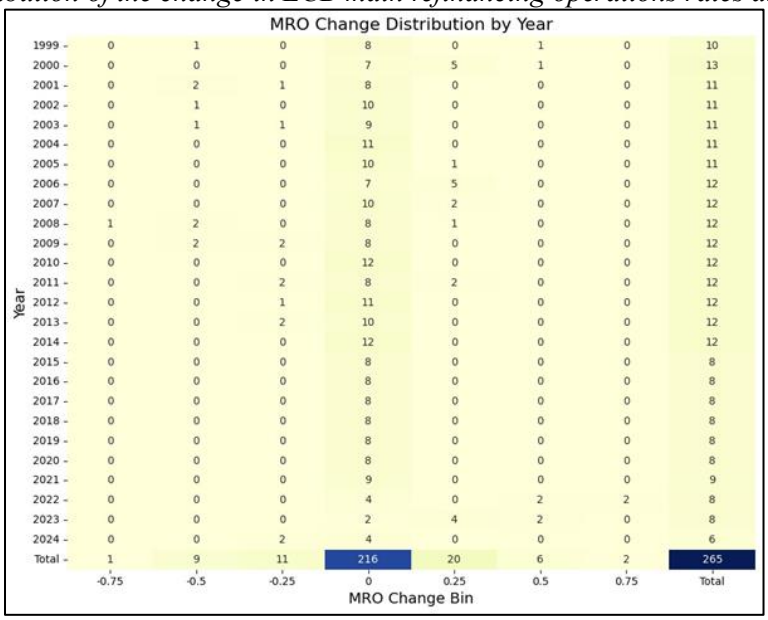


Table 1. Distribution of the change in ECB main refinancing operations rates announcements



Year	-0.75	-0.5	-0.25	0	0.25	0.5	0.75	Total
1999	0	1	0	8	0	1	0	10
2000	0	0	0	7	5	1	0	13
2001	0	2	1	8	0	0	0	11
2002	0	1	0	10	0	0	0	11
2003	0	1	1	9	0	0	0	11
2004	0	0	0	11	0	0	0	11
2005	0	0	0	10	1	0	0	11
2006	0	0	0	7	5	0	0	12
2007	0	0	0	10	2	0	0	12
2008	1	2	0	8	1	0	0	12
2009	0	2	2	8	0	0	0	12
2010	0	0	0	12	0	0	0	12
2011	0	0	2	8	2	0	0	12
2012	0	0	1	11	0	0	0	12
2013	0	0	2	10	0	0	0	12
2014	0	0	0	12	0	0	0	12
2015	0	0	0	8	0	0	0	8
2016	0	0	0	8	0	0	0	8
2017	0	0	0	8	0	0	0	8
2018	0	0	0	8	0	0	0	8
2019	0	0	0	8	0	0	0	8
2020	0	0	0	8	0	0	0	8
2021	0	0	0	9	0	0	0	9
2022	0	0	0	4	0	2	2	8
2023	0	0	0	2	4	2	0	8
2024	0	0	2	4	0	0	0	6
Total	1	9	11	216	20	6	2	265

Table 2. Summary statistics

	Mean	Std. Dev.	Min.	Quartile 1	Median	Quartile 3	Max.
CAR	-0.0087	0.0781	-0.5512	-0.0303	-0.0003	0.0285	0.2552
abs_CAR	0.0476	0.0624	0.0001	0.0155	0.0297	0.0574	0.5512
pessimism	0.2306	0.9457	-2.445	-0.2931	0.2076	0.7434	3.1384
jaccard_similarity	0.2803	0.1472	0.0147	0.1535	0.2794	0.3885	0.6356
output_gap	-0.0049	1.7661	-12.7444	-0.9054	-0.0961	1.0057	3.3627
hcpi_change	0.2067	0.4725	-1.2996	-0.0599	0.2249	0.4163	2.4342

The similarity of ECB communication across time

In table 3 we report the results of our OLS regression of the model described before to explain the effects of several variables on similarity over time. The table shows that the trend variable has a positive and significant value at a 1% level indicating that central bankers' speeches are more and more similar over time. This is consistent with the results of Amaya & Filbien (2015) who analysed this phenomenon only

until 2013. However, we find lower trend coefficients that are very close to zero (0.465 vs 0.493), because of the big drop in similarity after 2020. This may be explained by the fact that in times of a pandemic like Covid-19, we expect central bankers' speeches to be changing due to the high uncertainty. We also find that inflation has a negative impact on similarity (-0.170), with significant value at a 5% level indicating that in times of inflation the similarity between speeches decreases. We find the same value as in the other paper indicating that inflation has always had the same effect on similarity, and the pandemic crisis has not changed anything about that. We note that our adjusted R^2 are lower than in the paper meaning that a smaller proportion of variation in the dependent variable that is explained by the independent variables. In other words, other factors may influence our model, and this may be explained by high uncertainty during the Covid-19 crisis.

Table 3. Explaining similarity with time: OLS regression. The following table presents the results of OLS regressions that explain the similarity. The dependent variable is the measure of similarity of ECB speech at press conferences. The sample consists of 265 ECB monetary policy announcements made between January1, 1999, and October 17, 2024.

Variable	(1)	(2)	(3)	(4)
Intercept	-1.4300***	-5.1577***	-5.1758***	-2.1765***
logTime		0.4587***	0.4654***	
Time (count)				0.0057***
output gap	0.0099	0.4587***	0.0147	0.0257
HCPI change	-0.1231		-0.1696**	-0.2037***
MRO change	-0.0500		-0.2021	-0.3739*
Adjusted R^2	-0.0034	0.4359	0.4478	0.4086

** Statistical significance at the 10% level.*
*** Statistical significance at the 5% level.*
**** Statistical significance at the 1% level.*

ECB communication similarity and market learning

In table 4 we report the results of our OLS regression of the model described before to explain the effects of several variables on market reaction over time measured as the absolute cumulative abnormal returns. The first column presents the result of the event study regressing pessimism on absolute cumulative abnormal returns to see if the informational content of announcements has an effect on investors' reaction. We find that pessimism has a positive effect on market reaction with significant value at a 5% level. However, our coefficient is very close to zero and lower than in the paper (0.008 vs 0.530). Indeed, speeches' overall sentiment has had almost no effect on markets. We can infer that investors rely less and less on the polarity of announcements. We may think that in times of crisis, invertor behavior on the markets was driven by other factors. The second column shows the effects of conventional open market operations on market reaction. But we don't consider this model as reliable because variables are not statistically significant, like in the paper. In the third column, we show that the effect of the announcement on the market is negative with a 5% significance level when we factor similarity with pessimism. However, in the last column we find different results than in the paper because this factoring is not significant anymore when we add open market operations variables. Note that we could not find data for surprise MRO and didn't include it in the regression. That is, our results say that communication's similarity doesn't affect the dispersion of stock market's reaction with the informational content of announcements as a factor. We cannot say that similarity helps investors to adjust their forecasts precisely. Additionally, we cannot say that ECB's communication policy is successful in maintaining stock prices. Again, this difference with the original paper might be linked to the fact that the very high volatility encountered on stock prices during the pandemic disrupted any kind of predictability.

Table 4. Explaining absolute cumulative abnormal returns with time and similarity: OLS regression. This table presents the results of OLS regressions that explain the Estoxxx 50 index absolute cumulative

abnormal returns. The sample consists of 265 ECB monetary policy announcements made between January1, 1999, and October 17, 2024.

Variable	(1)	(2)	(3)	(4)
Intercept	0.0457***	0.0473***	0.0455***	0.0461***
Pessimism	0.0083**			
Pessimism x Similarity			-0.0063**	-0.0050*
Output gap		-0.0038*		-0.0031
HCPI change		0.0026		0.0001
MRO change		-0.0295		-0.0149
Adjusted R^2	0.0122	0.0098	0.0208	0.0194

** Statistical significance at the 10% level.*
*** Statistical significance at the 5% level.*
**** Statistical significance at the 1% level.*

Extension

Introducing subjectivity

TextBlob is a Python library designed for Natural Language Processing (NLP) tasks. It utilizes parts of Pattern's natural language processing tools (e.g., sentiment analysis, inflection, conjugation) and integrates them with the capabilities of NLTK (Natural Language Toolkit). This enables TextBlob to serve as a simpler tool for text analysis. It analyzes sentences to return two key metrics: polarity and subjectivity. The subjectivity is a float within the range [0.0, 1.0] where 0.0 is very objective and 1.0 is very subjective, indicating that the sentence contains more personal opinions rather than objective facts.

To do so, it calculates average subjectivity over each word in each text using a dictionary and tagged scores. It uses pattern library for that, which takes the individual word scores from sentiwordnet.

So TextBlob first tokenized the text, breaking it down into individual words and phrases. Then it uses the Natural Language Toolkit (NLTK) to tag each word with its part of speech (e.g., noun, verb, adjective). After this step, TextBlob employed the pre-trained classifier to analyze the sentiment of the text. The subjectivity score is calculated based on the presence of subjective words and phrases. Subjective words are those that convey personal opinions, emotions, or judgments (e.g., "wonderful," "terrible"). Objective words, on the other hand, are factual and neutral (e.g., "table," "city")

After computing subjectivity we regressed on CAR. The coefficient for "Subjectivity" is -0.0774, suggesting that higher levels of subjectivity are associated with a reduction in CAR. However, it is not significant, then indicates that this result is not statistically significant at conventional levels. With the second regression involving only textual variables, the coefficient for "Subjectivity" is -0.1035, also suggesting that higher levels of subjectivity are associated with lower CAR. But still, it is not statistically significant, meaning there is insufficient evidence to conclude a meaningful impact of subjectivity on CAR in this model.

Table 5. Explaining absolute cumulative abnormal returns introducing subjectivity: OLS regression

Variable	(1)
Intercept	0.0778
Subjectivity	-0.0774
Output gap	-0.0040*
HCPI change	0.0037
MRO change	-0.0321
Adjusted R^2	0.0084

Variable	(1)
Intercept	0.0866*
Subjectivity	-0.1035
Pessimism	-0.0083
Pessimism x Similarity	-0.0115
Adjusted R^2	0.0165

Using a light Transformer model to classify text

DistilBERT is a compact Transformer model derived from BERT base through distillation. With 40% fewer parameters than google-bert, it operates 60% faster while retaining over 95% of BERT's performance on the GLUE language understanding benchmark.

We used a fine-tuned version of DistillBERT specifically tailored for sentiment analysis in the financial domain. It has been trained on the Financial PhraseBank dataset to classify financial texts into three sentiment categories: Positive, Neutral, Negative

For information, BERT (Bidirectional Encoder Representations from Transformers) is a natural language processing (NLP) model introduced by Google in 2018. It is based on the Transformer architecture and is designed to pre-train deep bidirectional representations of text. Unlike earlier models that processed text in a unidirectional manner (either left-to-right or right-to-left), BERT captures the full context of a word by considering the words on both sides simultaneously (bidirectional processing), thus understanding language in a way that accounts for the meaning of a word within its specific context, which is crucial for tasks like text classification. With this model we find that 75% of the ECB announcements were classified positive.

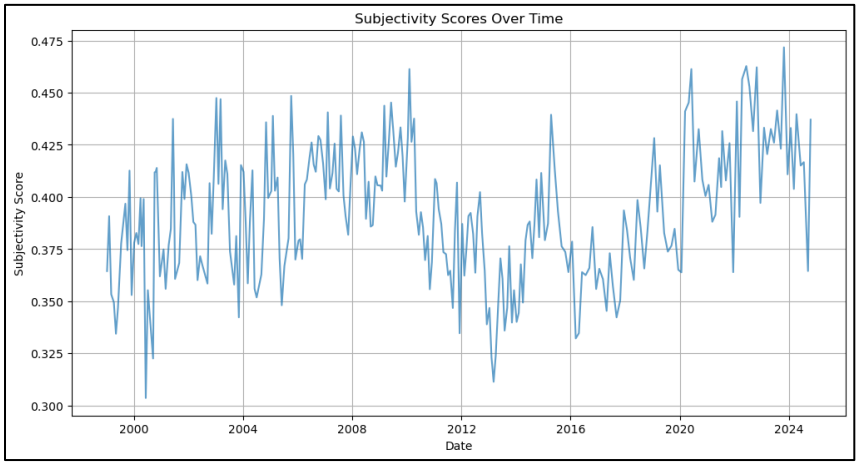
Conclusion

In summary, our results show some valuable insights for the analysis of similarity of ECB announcements. We found a significant linear trend and discovered that inflation has negative impact on similarity, in accordance with the original paper. However, our model doesn't allow us to predict stock market returns because most of the variables are not statistically significant. Moreover, our adjusted R^2 are very low indicating that our independent variables are not sufficient to explain the absolute CAR. In any case, the original paper also found a weak model with low adjusted R^2 values. Finally, we tried to include a new metric, subjectivity, into our regression on absolute CAR to see if it can have an impact. The results were not significant and didn't really improve the model.

To make a final comment, we believe that the high volatility and uncertainty during and after the Covid-19 crisis disrupted the predictability of stock returns. Moreover, the original paper didn't clearly mention how all variables and data were computed and processed, especially the output gap and the content of the texts, which may explain some differences between our results.

Appendix

Appendix 1: Subjectivity using Textblob



Appendix 2: Regression 1 Python output

```
Regression: Intercept, Output Gap, Inflation, MRO Change
Coefficients:
const          -1.4300***
output_gap         0.0099
hcpi_change       -0.1231
mro_change        -0.0500
dtype: object
Adjusted R-squared: -0.0034
--------------------------------------------------
Regression: Intercept and Time
Coefficients:
const      -5.1577***
logTime     0.4587***
dtype: object
Adjusted R-squared: 0.4359
--------------------------------------------------
Regression: Intercept, Time, Output Gap, Inflation, MRO Change
Coefficients:
const          -5.1758***
logTime         0.4654***
output_gap         0.0147
hcpi_change     -0.1696**
mro_change        -0.2021
dtype: object
Adjusted R-squared: 0.4478
--------------------------------------------------
Regression: Intercept, Time (count), Output Gap, Inflation, MRO Change
Coefficients:
const          -2.1765***
time_count      0.0057***
output_gap         0.0257
hcpi_change    -0.2037***
mro_change       -0.3739*
dtype: object
Adjusted R-squared: 0.4086
```

Appendix 3: Regression 2 Python output

```
Regression: Intercept, Pessimism and R²
Coefficients:
const       0.0457***
pessimism    0.0083**
dtype: object
Adjusted R-squared: 0.0122
--------------------------------------------------
Regression: Intercept, Output Gap, Inflation, MRO Change and R²
Coefficients:
const         0.0473***
output_gap     -0.0038*
hcpi_change     0.0026
mro_change      -0.0295
dtype: object
Adjusted R-squared: 0.0098
--------------------------------------------------
Regression: Intercept, Pessimism * Similarity and R²
Coefficients:
const        0.0455***
pess_x_sim   -0.0063**
dtype: object
Adjusted R-squared: 0.0208
--------------------------------------------------
Regression: Intercept, Pessimism * Similarity, Output Gap, Inflation, MRO Change and R²
Coefficients:
const         0.0461***
pess_x_sim     -0.0050*
output_gap      -0.0031
hcpi_change     0.0001
mro_change      -0.0149
dtype: object
Adjusted R-squared: 0.0194
```

Appendix 4: First extension Regression Python output

```
Regression: Intercept, Subjectivity, Output Gap, Inflation, MRO Change and R²
Coefficients:
const             0.0778
Subjectivity     -0.0774
output_gap      -0.0040*
hcpi_change       0.0037
mro_change       -0.0321
dtype: object
Adjusted R-squared: 0.0084
--------------------------------------------------
```

Appendix 5: Second extension Regression Python output

```
Regression: Intercept, Subjectivity, Pessimism, Pessimism * Similarity  and R²
Coefficients:
const           0.0866*
Subjectivity    -0.1035
pessimism       -0.0083
pess_x_sim      -0.0115
dtype: object
Adjusted R-squared: 0.0165
--------------------------------------------------
```

Appendix 6: DistilBERT model Python code

```python
#model DistilBERT Fine-Tuned for Financial Sentiment Analysis

import torch
from transformers import DistilBertTokenizer, DistilBertForSequenceClassification

tokenizer = DistilBertTokenizer.from_pretrained("AnkitAI/distilbert-base-uncased-financial-news-sentiment-analysis")
model = DistilBertForSequenceClassification.from_pretrained("AnkitAI/distilbert-base-uncased-financial-news-sentiment-analysis")

inputs = tokenizer(df["content"].tolist(), return_tensors="pt", padding=True, truncation=True)
with torch.no_grad():
    logits = model(**inputs).logits

predicted_class_id = logits.argmax(dim=1).tolist()

label_mapping = {0: "Negative", 1: "Neutral", 2: "Positive"}
predicted_labels_financial = [label_mapping[class_id] for class_id in predicted_class_id]

print(f"Predicted Sentiments: {predicted_labels_financial}")
```