Can Sentiment Scores Improve the Prediction of Smartphone Sales?

Group Report FINA4350: SalesEQ

Authors:
Jason Li 3035704758
Ricky Choi 3035684075
Zixian Fan 30355771610
Maximilian Droschl 3036275326
Mahir Faiaz 3035918696

GitRepo:
Github Repo

Submitted on 3rd May 2024

1 Research Objective and Methodology

The smartphone market within the United States (U.S.) became increasingly competitive in recent years, which puts extra pressure on manufacturers and forces them to adapt their strategies accordingly Fan and other researchers (Fan and Yang, 2020). The rapid pace of product development, increasing differentiation among smartphones, and relatively short life cycles of smartphones contribute to unpredictable sales patterns and increased volatility, exacerbating the challenge. One of the strategies employed by manufacturers is forecasting industry sales. If this process is not managed effectively, it can have significant consequences. The traditional models used to predict smartphone sales are primarily based on publicly available data, which mostly includes past values or consumer sentiment indices. Consequently, it is relatively difficult to make more accurate predictions than competitors. The models are only as good and informative as the underlying data used. Therefore, our team motivated to extend these traditional techniques to a hybrid forecasting model that aims to incorporate sentiment indices estimated using past text data related to phone sales.

Thereby, we test the hypothesis whether an own sentiment score derived from news articles adds predictive information to traditional model specifications. However, as monthly phone sales data, both for the entire industry and for individual market suppliers, is not publicly available, our analysis focuses on predicting the monthly log returns of a constructed equity portfolio, which reflects the monthly performance of the overall phone selling industry within the U.S. The rationale for this assertion is that, according to Fama (Fama, 1970), financial markets are considered the most efficient markets, reflecting all available information. In contrast to other performance measures, such as sales figures, financial markets provide a more comprehensive view of a firm's performance. However, Roberts (Roberts, 1967) demonstrated that there are various forms of efficiency that may lead to a pure approximation of future prices. This is because fundamental analysis relies on publicly available information and cannot generate excess returns. As the literature reports semi-strong form efficiency and occasional violations of it, the aim of this study is to provide more accurate forecasts by adding a privately constructed sentiment score. If it can be demonstrated that our own sentiment score contains more information than publicly available sentiment scores, it can be reasonably assumed that our sentiment score may assist in predicting phone sales.

In order to test the above hypothesis in an appropriate manner, the research employs the following methodology. First, text data from news articles related to smartphone sales published between January 2012 and December 2022 will be scraped, cleaned, and preprocessed. Next, simple text data analysis will be performed. The sentiment score will be constructed using a Sentiment-LDA model, which was introduced by researchers (Li et al., 2010). This model considers the sentiments and topics of each word within a text document simultaneously. This allows the model to derive document polarities for each article scrapped, which will then contribute to monthly average scores. In order to predict one-period log returns, four ARMAX (1,1) models are fit on a rolling basis, utilising both our own sentiment score and the Michigan Consumer Sentiment Index (MCSI) from the University of Michigan (University of Michigan, 2024) as exogenous variables. Consequently, the performance of each model specification can be evaluated based on MAPE, RMSE, and the accuracy in predicting the correct return sign.

2 Data Collection

For textual data, Factiva is used to scrap articles to ensure a standardized format of articles. The headline,

body, date and source of the articles are kept for further analysis and model building. To scrap articles which would be relevant to forecasting smartphone sales, the article query on Factiva is devised to query articles which are related to different aspects of smartphone sales market. These aspects include:

Aspects	Sales	Model Features	Labor Issues
Keywords	Smartphone Sale Market	Smartphone Model Feature	Smartphone Worker Union Device Company

Table 1: Data Query Details

The following is some summary of the data collected:

Quantity 100 articles per month (12,568 in total with some month not having 100)
Timeframe 2012.01 - 2022.12
Data Source - New York Times - Wall Street Journal - Reuters - Associated Press
Query (((smartphone or smartphones) and (sale or sales) and (market or markets)) or ((smartphone or smartphones) and (model or models) and (feature or features)) or ((smartphone or smartphones) and (worker or workers) and (union or unions) and (device or devices) and (company or companies))) and fmt=article

Table 2: Summary of Data Collection Parameters

3 Data Cleaning

Data Cleaning is an important part in text preprocessing as it determines the quality of data and hence the performance of the model. The goal of text preprocessing is to remove redundant text that does not help with explaining the sentiment of the articles, and to unify the formats of the texts. Text pre-processing was applied to the headline and content of each article.

We first converted all words into lowercases to eliminate the influence of word cases on their meanings. Next, we performed word tokenization on the texts. Then, we removed redundant data such as stop words, special characters, numbers, and synonyms. These elements appeared in various texts and therefore did not provide information for sentiment analysis. Removal of irrelevant data also speeds up the training process and improves the quality of data. After that, lemmatization was performed to convert words into their meaningful root forms. Although the execution time for lemmatization would be longer than that of stemming, lemmatization provided better results by performing analysis that depends on the word's part-of-speech and producing real, dictionary words. Text pre-processing was performed in "TextPreprocessing.py".

There are a few limitations we encountered during data cleaning. First, spelling mistakes cannot be eliminated. There are no patterns for spelling mistakes, and it is impossible to do eyeball check on all words and correct them. Given the sources of text are reputable newspapers such as the New York Times and the Washington Post, error due to spelling mistakes can be neglected. Second, the original meaning of some words may be lost after transformation. For instance, "U.S." becomes "us" after lowercasing and removal of punctuation. It may be misrecognized as the pronoun "us" instead of a country. We believe this problem does not occur frequently and hence it can be neglected.

4 Data Analysis

Once the data cleaning part finished, the very first thing to do is to verify the completeness of the data, such as the number of rows and the amount of text per sample, so that we can make sure that the subsequent model can provide robust results with the cleaned data. We have provided details in the Rudimentary analysis, for example, a total of 132 months of data corresponds to exactly 11 years of dependent variable. It is worth mentioning that the number of texts obtained in some months did not reach 100-target, which may be since the news in these months did not focus on mobile phone

related topics. However, even though the number of texts obtained may be low during these months, there are at least 50 texts articles per month, so the amount of data is still sufficient for model training of sentiment analysis later.

In addition to this, some text analytics have been applied, such as making a summary of the cleaned text, showing Word Cloud and Word Co-occurrence Network Graphs. Detailed instructions have been provided in the Blog and will not be repeated here.

5 Sentiment Analysis

5.1 Sentiment-LDA Model

Sentiment-LDA (Latent Dirichlet Allocation) is a probabilistic model that extends the classical LDA model to incorporate sentiment analysis. The idea is to model not only the themes present in a collection of documents, but also the sentiment expressed by these themes. In our study, by fitting the Sentiment-LDA model to all the articles of each month, we want to identify the most critical words that can express positive and negative sentiments in these articles. This means that we ultimately set up the categorisation of emotions as a binary classification problem between positive and negative. By basing the words we get, we can further construct a sentiment score for each month, and thus parse out what the average sentiment for mobile phones is for all the articles in that month.

Combining the code of Li and other researchers (Li et al., 2010), we customised the sentiment LDA model. Our final goal is based on two initial helper functions. The first function, 'sampleFromDirichlet' allows us to take samples from the Dirichlet distribution, which is used to generate a probability distribution of topics for each document and a sentiment distribution for each topic in each document. Notice that our final goal is a sentiment distribution for all documents, which is not the same as a single document. Then, the function 'sampleFromCategorical' will sample an index from a categorical distribution defined by the parameter theta. This is used to randomly assign topics and sentiments to each word in the documents based on their respective distributions.

After that, we will be based on Gibbs Sampler to keep sampling close to the corresponding intrinsic distribution of the vocabulary through the MCMC process. With the function 'processSingleReview' that processes the text, we first use the function 'processReviews' to convert the reviews into bag-of-words matrices 'CountVectorizer'. We then use the 'conditionalDistribution' function to compute the conditional distribution of themes and sentiments for specific words in a given document, based on the current state of the model. It can be noticed that we use the default generation of the '_initialize_' function for the initial state generalization, and after that we keep on deciding the posterior distribution through the effect of priority and likelihood, which is the core of Gibbs Sampler. In the 'run' function, we keep iterating the prior and posterior distributions to approximate the sentiment that these words really represent. Finally, we use the 'getTopKWordsByLikelihood' function to extract the words that best represent each theme and sentiment. They help to explain the themes after the model has been trained and are central to our calculation of the Sentiment Score.

Ultimately, with the hyperparameters set, we fitted the Sentiment-LDA model separately for a total of 132 months, and ultimately, for each month, we retained 25 of the words representing positive and negative sentiment, respectively.

5.2 Sentiment Score

The next step is to calculate an average sentiment score for each month based on the 25 most positive and negative words. The monthly sentiment score is calculated based on the average article score. For each article,

a negative and a positive score are calculated, whose difference builds the final sentiment score for that specific article. The positive and negative scores are calculated as the sum of weighted occurrences of each of the monthly top 25 positive and negative words. The weights are calculated as the inverse of the rank of the word within the list of the top 25 words. Calculation is done in the SentimentScore.ipynb file.

6 Time Series Modelling

This section is divided into four subsections. The first section presents all variables used for time series modelling and forecasting, accompanied by a brief description of pre-processing. The next section examines data exploration and necessary adaptations for the series under consideration to be fit to ARMAX models. The third section presents the results of model selection.

First, the dependent variable must be created. To keep the focus on the American phone market, our dependent variable will be the monthly log returns of an equity portfolio consisting of the five largest phone companies according to their market share within the American telecommunications market. According to Bloomberg, the companies with the largest market shares in the U.S. smartphone industry are Apple (68.23%), Samsung (25.20%), Motorola (3.90%), and Google (2.67%). The percentages provided here correspond to the portfolio weights. Using the yfinance and pandas_datareader package, US price data for each of these assets is fetched. The final log return series is than calculated according to

$$return_t = log(\frac{p_t}{p_{t-1}}).$$

Secondly, a series for the second exogenous variable, the MCSI from the University of Michigan (University of Michigan, 2024), is created. In order to ensure consistent chronological ordering, a unified date column is created, and the dataset is filtered to only include entries from December 2011 to December 2022. One additional month in 2011 is included since the data is differenced once later in the analysis. Additionally, the MSCI is scaled to the [-1,1] interval. Finally, all three series, namely the log return series, the MSCI series, and our own sentiment score series, are combined into a single data frame, called preprocessed_data.csv.

As a next step, briefly preliminary analysis of the time series data is performed. After visual inspection and conducting a series of Augmented Dickey Fuller tests, where the null hypothesis asserts non-stationarity, i.e. the presence of a unit root, it was found that both our own sentiment score and the log return series are stationary. This was not the case for the MCSI series (ADF statistic: -0.131; p-value = 0.946). After differencing, the results indicated stationarity. A final visual inspection of the three variables revealed that no trend or seasonality component could be identified, in accordance with the test results. The resulting data is stored in the modeldata.csv file. For further details, the reader is referred to the TimeseriesExploration.ipynb file.

Subsequently, an appropriate order for the ARMAX (p, q) model is selected. This is accomplished in the TimeseriesModelling.ipynb file. It is important to note that this script primarily serves the purpose of selecting the right ARMAX order. Utilising the data from the modeldta.csv file, lagged values of the exogenous variables are generated. Furthermore, training and testing data sets are created, with the training data spanning from January 2012 to January 2021 and the testing data extending from February 2021 to December 2021. Furthermore, data sets are created for each of the four models. The Base model includes no exogenous variables, the MCSI model includes lagged values of the MCSI, the Sentiment model includes lagged values of our own sentiment score, and the Both model includes lagged values of both the MCSI and our own sentiment score variables. Based on the lowest BIC and AIC, the

ARMAX (1,1) models were selected (see Table 3).

The model selection is further supported through applying simple tests (such as the Ljung-Box test) for checking the hypothesis that the residuals are observed values of iid random variables. In every case, the null of i.i.d. data (i.e., there are no autocorrelations in the series at any of the tested lags) was confirmed. Details of the code can be found in our sentimentLDA.py and SentimentAnalysis.ipynb. It is worth noting that the code in SentimentAnalysis.ipynb involves two duplicates, due to the fact that the query missed certain cases when we first fetched the data, and thus the addition was made in the second code. If you re-run the code you can simply run

Table 3: Time Series Model Specifications

Model	Specification
Base	$return_t = \phi return_{t-1} + \theta_{t-1}\epsilon_{t-1} + \epsilon_t$
MCSI	$return_t = \phi return_{t-1} + \theta_{t-1}\epsilon_{t-1} + \epsilon_t + \beta_1 MCSI_{t-1}$
Sentiment	$return_t = \phi return_{t-1} + \theta_{t-1}\epsilon_{t-1} + \epsilon_t + \beta_1 Sent_{t-1}$
Both	$return_t = \phi return_{t-1} + \theta_{t-1}\epsilon_{t-1} + \epsilon_t + \beta_1 MCSI_{t-1} + \beta_2 Sent_{t-1}$

7 Results and Discussion

Finally, in order to assess and compare the predictive power of each of these models, they are fit on a rolling basis, using the past 36 months to predict the return and its sign for the following month. This will enable us to calculate average performance measures such as RMSE, MAPE, and the accuracy in predicting the correct return sign for each of the above models.

Table 4: Performance Metrics of Models

Model	**Average MAPE**	**Average RMSE**	**Average Accuracy**
Base	**138.0830**	0.0597	0.5638
MCSI	132.1203	0.059778	0.5532
Sentiment	142.3472	**0.059677**	**0.6277**
Both	139.9262	0.059814	0.5957

The results in Table 4 indicate that models with lagged values of our own sentiment score exhibit the highest performance in terms of RMSE and accuracy in predicting the correct return sign. When an LSTM was fitted in a similar framework, the results differed slightly from those reported above (see TimeseriesLSTM.ipynb). The models incorporating our own sentiment score demonstrated a slight reduction in performance compared to the model utilising the MSCI, although the inclusion of both sentiment scores resulted in an increase in accuracy to 100%, in comparison to 54% observed in the Base LSTM. This implies that the custom sentiment score is potentially retrieving information/consumer attitudes that are not fully captured by the broader, publicly available sentiment indices. Given that financial markets are efficient and integrate available information, as proposed by Fama (Fama, 1970), any additional, relevant information that can improve prediction accuracy is valuable. In this case, our sentiment score appears to represent such information. In particular, even in an efficient market, there is a lag before all information is reflected in prices. This aligns with the semi-strong form of market efficiency proposed by Roberts (Roberts, 1967), where prices reflect all publicly available information but not necessarily immediately. The lagged sentiment scores capture information that takes time to be fully incorporated into market prices. By utilising sentiment scores from the previous month, the model acknowledges that investor sentiment to news can have a delayed effect. To conclude, it can be assumed that our own sentiment score contains more information than conventional, publicly available sentiment indices, such as the MCSI. Furthermore, it can be hypothesised that the sentiment score may enhance the accuracy of phone sales prediction.

References

Eugene F. Fama. 1970. Efficient capital markets: A review of theory and empirical work. *Journal of Finance*, 25(2):383–417.

Yiwei Fan and Chongkui Yang. 2020. Competition, product proliferation, and welfare: A study of the us smartphone market. *American Economic Journal: Microeconomics*, 12(2):99–134.

Fang Li, Minlie Huang, and Xiaoyan Zhu. 2010. Sentiment analysis with global topics and local dependency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, pages 1371–1376.

Harry V. Roberts. 1967. Statistical versus clinical prediction of the stock market. Technical report, University of Chicago, Chicago.

University of Michigan. 2024. Surveys of consumers - data. https://data.sca.isr.umich.edu/data-archive/mine.php. Accessed: 2024-01-01.