How Sentiment Analysis of Reddit Can Improve the Prediction of Number of Sales for iPhones

Group 9 – Marketing
Gary Buranasampatanon, Sabrina Li, Joey Gu, Xiaokai Liu

I. Business Understanding

In this project, we aim to predict the future sales of iPhones using sentiment analysis results generated from Reddit posts. IPhone is the general name of smartphones produced by Apple and it is one of the most popular smartphones in the market. Until May of 2019, there are 21 different models of iPhones and eight of them are on sale.

Prediction of number of iPhone sales is meaningful in several ways. First, prediction of iPhone sales impacts the whole smartphone market deeply. Smartphone plays an important role in people's daily lives. The number of active smartphones across the globe reached 3.3 billion by the end of 2018. Apple is one of those most popular smartphone manufacturers. IPhone holds 13.2 percent of global smartphone market shares. According to Counterpoint Research's Market Monitor, iPhone continued to reign supreme in the higher-end smartphone market until the third quarter of 2018. IPhone holds a share of 79 percent of smartphone over $800. For the premium smartphone market over $400, iPhone still holds the lion's share, making up 47 percent of the total share. The prediction of sales of iPhone is meaningful for the prediction of future smartphone market, especially the high-end smartphones. Also, prediction of iPhone sales will contribute to the prediction of Apple stock price. Sales of iPhone play an important role in the market value of Apple, even in the whole market. IPhone is the most profitable product of Apple. According to Statista, more than 60% of its revenue comes from iPhone. Hence, studying and forecasting on the sales of iPhones are really essential in estimating Apple stock price. Apple has the second highest market value in the world until May 1st, 2019 and was the top company with largest market capitalization from 2012 to 2018. As we can see, Apple stock (AAPL) price highly influences the whole market. Therefore, prediction of iPhone sales will help us make a more prediction of Apple stock price, even the global economy environment. Last but not least, the prediction of iPhone sales is also important for Apple to maximize the profit. More precise prediction about iPhone sales will help Apple make a better decision on the amount of orders of machine elements from production factories. Both overstock and understock will cause a large amount of loss of profits to a company. As I mentioned before, iPhone earns the largest portion of profit for Apple. Hence, accurate prediction of iPhone sales is really crucial in controlling the amount of iPhone orders from factories and in maximizing the profit from iPhone. In conclusion, an accurate prediction of iPhone sales will help Apple to maximize the revenue and minimize the loss caused by incorrect amount of stock.

To make a more precise prediction, we want to incorporate people's overall attitudes towards iPhone into the prediction of iPhone sales using time series methods.

We believe that people's attitudes to iPhone can affect the sales. Nowadays, with the rapid development of internet and social media, the information exchanges happen in every second of our lives. People are exposed to a large amount of information every day and people are getting used to studying more before making a decision. For instance, before they decide to pay for a product, they may want to do more researches about this product including reading other consumers' feedbacks or reviews. The feedbacks may influence people’s decisions on their consumptions. For example, someone expected feature A of one product. However, if he/she finds some bad feedbacks on feature A of one product, he/she may exclude this product as a choice. Consumer feedback on online web board is one of the most approachable way for people to learn the information. It reflects a subset of the whole customers’ opinions about each product and it can be viewed as a “sample” of all the reviews. Therefore, we think these opinions may provide valuable information about the number of sales in each period of time.

Reddit (https://www.reddit.com) is one of the largest open online community with a large amount of discussions and opinions on various topics. Reddit has 330 million users and 26.4 million monthly active users until April of 2019. There are 2.8 million comments left on Reddit daily. It is ranked as the #6 most visited website in the US. It is a large platform for people to exchange their ideas with each other. Reddit divided the discussions into some sub-community which gathers the posts with same topic together. The sub-community is called subreddit. We extracted the posts about iPhone on Reddit as a reflection of people’s overall attitudes.

As mentioned before, we would like to study the sales of iPhones and incorporate people’s rating into the time series analysis of iPhone sales. We would use the posts in subreddit of iPhone, a sub-community which contains iPhone related posts, as a representative of people’s overall opinions and feedbacks. It has 1.5 million subscribers in total. People are actively expressing their ideas or sharing their experiences about iPhones. In order to quantize these opinions’ effects and extract the trend, we run experiments on sentiment analysis of the posts on the Reddit and explore how sentiment analysis of posts on Reddit can improve the prediction of number of sales for iPhones.

We employed three time series models to forecast future sales of iPhone using the information from Reddit. With all the comments from 2009 to 2018 crawled from the Reddit, we implement necessary preprocessing steps including lemmatization and remove of stopping words using NLTK package. Then we implement VADER algorithm to calculate the sentiment score of each post. For time series part, we implement four different time series models: Autoregressive moving average (ARMA) model, Autoregressive moving average model with exogenous inputs model (ARMAX), Vector Autoregression (VAR) model, and Gaussian Process Regression (GPR). Our baseline model is the Autoregressive and Moving Average model (ARMA). It is used to describe weakly stationary stochastic time series in terms of two polynomials. The first of these polynomials is for autoregression and the second for the moving average. We would like to figure out whether VAR model and ARX models outperform simple ARMA model when forecasting the sale of iPhones accompanying with overall sentimental score of posts on Reddit. VAR is a stochastic process model used to capture the linear interdependencies among multiple time series. VAR models generalize the univariate autoregressive model by allowing for more than one evolving variable and ARX model is a linear representation of a dynamic system in discrete time. We believe that VAR model and ARX model would provide a more accurate prediction than ARMA model.

After implementing these models, we find that the VAR model has the best performance with the smallest error, and both VAR and GPR model are able to beat the baseline model. Overall, our best model improves the result by nearly 30%. This improvement shows that, with the proper analysis on Reddit posts, we are able to increase the accuracy in predicting number of sales of iPhone in the near future.

II. Data Understanding

Since our goal is to predict the future sales of iPhones, we should use the past number of sales data of iPhone to conduct time series modelling. We got the iPhone quarterly sales data from Apple. Inc. Quarterly Income statement, where Apple announce their revenue, operating expenses, income, etc. Also, since we want to quantify people's attitudes towards iPhone, we need all the posts from Reddit discussion group of iPhone to conduct sentiment analysis. Hence, we decided to scrape the posts and the comments from Reddit using Python web page scraping techniques. Reddit was founded in 2006, but it started to have a competitively large number of users since 2009. Because of that, and the fact that Apple decided not to report the number of sales for iPhones, Macs and iPads after the last quarter of 2018, we decided to scrape all comments from 2009 to 2018. As there are various discussion groups on Reddit, we selected the iPhone subgroup to scrape the data as it is the most representative group that discusses all models of iPhone.

Apple Inc. has announced the financial results for each quarter from 2009, including their sales, income, balance sheet, and cash flows. One thing that people are interested in their summary data is the product summary, which introduces the unit of sales and the revenue for each product. This document divides products to several categories: iPhone, iPad, Mac, Services, and Other Products. The number that we are looking at is the number of sales for iPhone in each quarter. As can be seen from Figure 1, the number of sales for iPhone generally has a trend of increase, but bounced between 40 million to 80 million from 2014 to 2018. This is reasonable because the increase of sales can be caused by the release of a new model of iPhone, and the units of sales would drop for the next quarter when the model is no longer new to the customer.

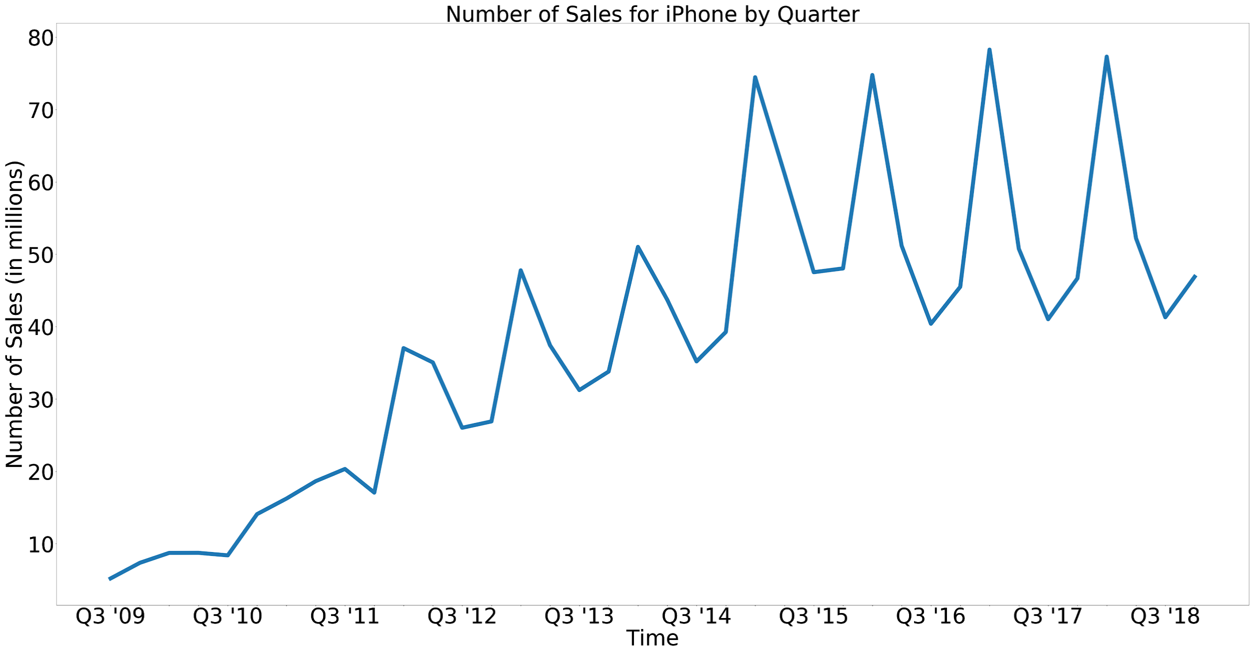

Figure 1: Number of sales for iPhone by Quarter

The other important factor in our analysis is the subreddit of users on Reddit. Reddit is a social media platform with various information regarding different subjects, and the posts that are organized by subgroups are called “subreddits”. These subreddits can be created by all users, and people can upvote and downvote other people’s subreddits to show their attitude. There are several different communities on Reddit where people are having various discussion about iPhone products. This includes the subgroups for different models, the iOS system, and applications for iPhones. However, to capture the most useful and representative information that are related to iPhone, we used the data from the big iPhone discussion group, which contains users’ posts regarding many different iPhone models. In this way, we could gather people’s attitudes toward iPhone as a general product.

In the original dataset of posts on Reddit, we have 5 variables and get 1048609 samples in total from Reddit. Each row of our data contains the information of one post. The first column of our dataset indicates the index of each post. The second column is the score of posts that calculated by the up-votes and down-votes under each comment. One up-votes counts as +1 and one down-votes counts as -1. Higher scores indicate that the posts are more popular and more people shared the similar ideas. The third column is ‘id’ which contains the unique id for each post. The column “body” contains the text of each post. The last column is ‘comment_created’, which is the timestamp when each post was created. For the data from Apple, we have the number of sales of each quarter from July 2009 to December 2018, which means that we have

38 time points in total for sales data. Therefore, we would have 4 variables in total before we merge the two datasets, including three variables that are continuous: scores of posts and the number of sales, a text variable, which is the content of the posts, and a variable for timestamp. The main idea of merging two datasets that contain different information is to first calculate the sentiment score for each post and comment, then group the posts by quarter and calculate the average mean of all scores in each group as the quarterly sentiment scores. In this way, we could merge two datasets by quarter and get the final variables that would be used in the modeling part. The variables that we analyzed eventually are the number of sales per quarter, the average sentiment scores aggregated by quarter, and the quarters from the third quarter of 2009 to the last quarter of 2018. To first calculate the sentiment score of each post, we used VADER algorithm and generated the compound sentiment score based on the content of the posts. The range for each score is from -1 to 1, where -1 means extremely negative and 1 means extremely positive.

Having the sentiment score of each post, we discovered that the range for these scores is small. Therefore, we decided to standardize the values so that it is easier to distinguish the differences. After the transformation, the sentiment score varies from -1.5 to 3, and the trend of change in score is clearer. Figure 2 shows that, from 2009 to 2018, the score actually went down from positive to negative in general, which implies that people are having more negative comments for iPhone products.

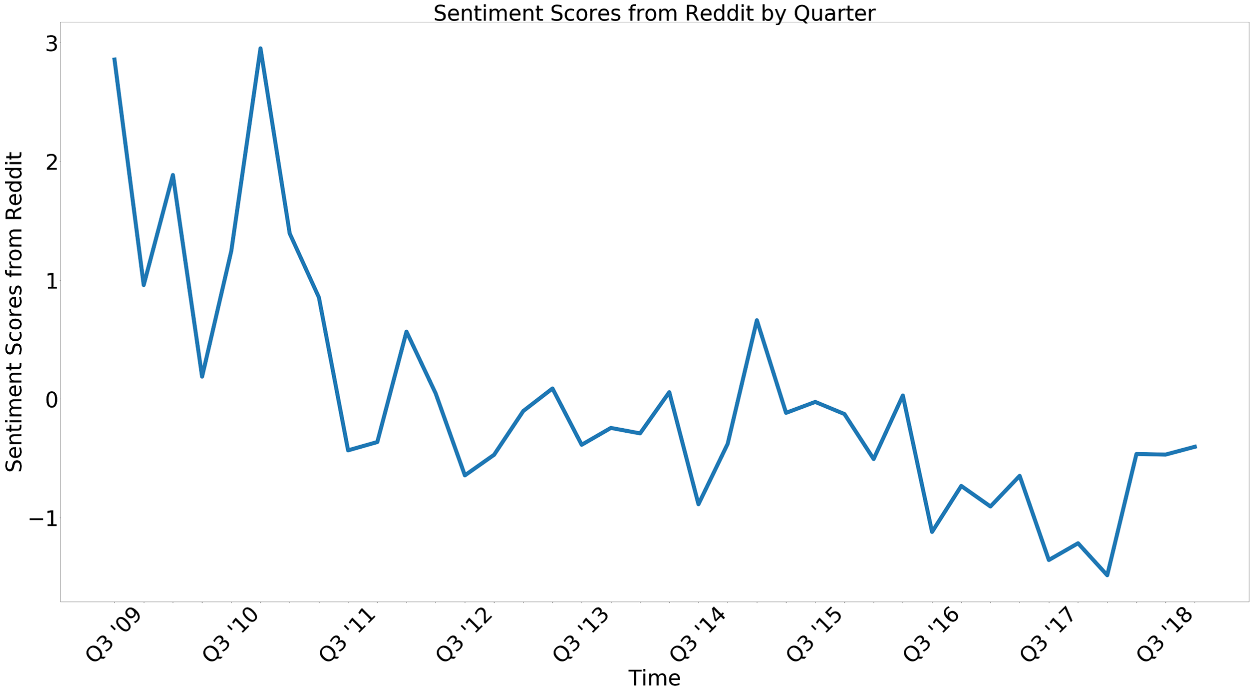

Figure 2: Sentiment Scores from Reddit by Quarter

Since we are using all data points in the dataset, we did not apply and sampling technique as it could lead to loss of information. However, as our analysis is based on information from Reddit, there could exist bias in the generated dataset, as the population for Reddit users is not equivalent to the population of all people or all customers who bought iPhone products. In this case, it can be more representative if we can include data from other social media. Our study requires a long period of time so that we can include as much data points as possible, but other social media platforms like Facebook or Twitter would not allow us to access these data. Therefore, the sample we choose eventually is the population from Reddit discussion group, and we will not conclude the result as the analysis based on all social media in general. The dataset we are having after the aggregation and merging manipulations are all in time series data structure for the convenience of modeling. As we have all the information for each quarter, there is no need for imputation for missing values either. Also, because that the original number of variables is small enough to fit in models, no dimension reduction is needed to downsize the number of features. With no categorical variables in the dataset, we do not need to create dummy variables either.

Our final dataset contains the index of each quarter, the aggregate sentiment score, the number of posts, and the number of sales for iPhone. Table 1 provides a detailed description about the features.

	quarter_index	sentiment_score	number_posts	normed_value	sales (millions)
mean	N/A	0.184	64844.658	0.000	38.078
std	N/A	0.016	50279.960	1.013	19.982
min	7.000	0.160	2390.000	-1.485	5.210
25%	N/A	0.176	36785.250	-0.497	21.762
50%	N/A	0.180	53689.500	-0.267	39.835
75%	N/A	0.187	90026.750	0.162	47.985
max	44.000	0.231	219803.000	2.951	78.290

Table 1: Statistics for the Model Inputs

As shown in Table 1, original sentiment score has a range of 0.16 to 0.23, which has been improved after the normalization. The normalized score varies from -1.485 to 2.951, which a mean of 0 and a standard deviation of 1.013. The range for the number of posts is approximately 2390 to 219803, with a mean of 64844, which indicates that we have around 65000 posts for each quarter on average.

III. Model Understanding

A. Types of Models and Goal

This project comprises two modelling parts about sentiment analysis and time-series analysis. The sentiment analysis part deals with calculating sentiment values of all Reddit posts, and the time-analysis part deals with finding the relationships between the sentiment values and the number of iPhone sales.

For the sentiment-analysis part, we use the VADER algorithm, which is available through the NLTK package on Python, and evaluate sentiment value of every Reddit post. Before running VADER on all posts, we have to preprocess the data in order to generate more accurate sentiment values with the three following steps. First, we remove all unwanted symbols e.g. commas and hyphens from every sentence. Second, we lemmatize every word in order to retrieve every word's root form. For example, the algorithm will transform a word 'went' to its root form as 'go.' Third, we remove every English stop word from our consideration. Both the lemmatization and stop-word-removal parts were conducted with the help of NLTK package.

After having preprocessed every post, we ran the VADER algorithm to obtain the preprocessed posts' sentiment scores. The output from the sentiment analysis will have a value between -1 for most negative sentiment to 1 for most positive sentiment. As we already generate sentiment scores for all posts, we create a quarterly index of sentiment values by calculating an equally weighted average value of all sentiment values in every quarter. Surprisingly, we find that the index output is positive in almost every quarter, and most of the values are in between 0.15 and 0.20. These values may indicate that, for most of the time, iPhone users are satisfied with their products, and their general opinion about iPhone have stayed positive throughout a number of quarters. Hence, we decided to use the standardized sentiment values instead of the raw sentiment values to reflect the degree of deviation from public expectation in every quarter. The standardized values are calculated by subtracting every sentiment score with its first moment and dividing the difference by its standard deviation.

For the time-series modelling part, we have tried running 4 models so far to test our assumption about the number of iPhone sales. The first model is ARMA model. This model captures lagged effects impacted by the endogenous variable and shocks from earlier periods altogether. According to a classical assumption about time-series data, we assume that the movement of the endogenous variable from one period to another follows a linear trend.

As a univariate model, ARMA may only include information about a trend created by the endogenous variable alone. However, it does not include any additional information contained in any other variable that may have some predictive power over the endogenous variable. In order to incorporate some information about the public opinion to our model, we deploy two multivariate time-series models including ARMAX and VAR with different assumptions about the interaction between the two variables. ARMAX assumes that the sentiment values affect the number of sales, but the other direction does not apply. In contrast, VAR assumes that there exist bidirectional relations between the two variables, and both of them should provide some forecasting power for one another. Again, as both of the models still assume that the data are in time-series form, the described relations between these variables are all linear.

Before running these models, we conducted a few tests on the data. First, we ran Augmented Dickey-Fuller test on both variables. We found that the number of sales is I(1), and the sentiment values are I(0). Hence, we would consider logarithmic differences of the numbers of sales, or sales growth, in the model, whereas the original form of the standardized sentiment values is preserved. Second, we ran exponential smoothing state space model with Box-Cox transformation on both time series to detect seasonality in the data. According to the seasonality test, we found that there exists seasonality in the sales growth, but not in the sentiment value. Thus, we removed the seasonal component of the sales growth and preserve only the trend component; from this point onwards in this report, the term sales growth will only cover the trend of sales growth.

The linearity in these time-series models raises a question to us about whether a non-linear model may outperform them. An advantage of using a simple linear model is its smaller number observations needed for model's convergence. With our dataset having just about 40 data points in total, using more complex model will be more challenging to make it converged. Because of the concern on the small number of observations, we decided to run an experiment using Gaussian Process Regression (GPR), which is known for its low requirement on the number of data observations.

Despite GPR's small requirement on the number of observations, similar to other kernel-based supervised learning methods, it requires developers to specify about what kind of kernel to use. Mathematically, we can prove that kernels have additive properties. Hence, we add the four following kernels altogether in order to explain different components of the sales growth.

- Squared Exponential Kernel: The squared exponential kernel has lower variation than other kernels. Hence, this kernel should capture the long-term trend about the sales growth.
- Matern Kernel (nu = 2.5): The kernel is used to capture medium-run variation in the model as the degree of correlation plunges very quickly after some threshold.
- Radial Basis Kernel: The radial basis kernel usually has greater degree of variation than both Matern kernel and squared exponential kernel. Hence, we will use this kernel to capture changes in medium-to-short-run trend.
- White Noise Kernel: This kernel will behave similarly to a noise component in time series models do. The added noise will prevent the generated function of GPR from overfitting to the training data.

An additional benefit of using GPR is its output as a set of functions rather than just a pointwise prediction. As a Bayesian method, GPR can also demonstrate prediction uncertainty at anywhere in its function's support. Therefore, if the performance from running this experiment is adequate or superior to performances from the traditional time-series models, we will also be able to add the degree of predictive variation to our result, which may be more suitable to forecasting task in a number of cases.

B. Model Selection

Choosing the best-performing model for time-series data is different from the case of longitudinal data. Instead of cross-validating the dataset, we will have to use information criteria to choose the best performing time-series model. In our case, we choose a model specification that generates the most optimal value of Akaike Information Criteria (AIC) for every iteration. This model selection method allows us to use a parsimonious model for every iteration, while not undermining the model's performance. Still, we may not be able to keep track of every model's coefficients in all iterations and compare all of their performances. We believe that we can accept the trade-off. This matches with general practice, where only the best and parsimonious model is generally the only model that is kept after running each experiment.

Specifically, we set the following range of values of parameter on which we will run our experiment. For the ARMA model, we set both the number of lags for our endogenous variable and the number of lags for noises or shocks to be from 0 to 4.

Both the ARX and the VAR models also have the same range of number of lags for their linear systems.

The GPR model has a different model selection method. As it does not follow the time-series assumptions, it treats every observation in the model as iid data. Even though the iid properties allow us to run k-fold cross-validation as usual, we will use the same walk-forward validation method as used in the time-series model in order to directly compare their performances. In order to ensure that the model's initialization will be sufficiently good, for each step of iteration, we compare the results from using five different initializations together and choose the best performing setting as the output.

C. Descriptive Statistics

For all of these models, we assume that the data are normally distributed. Some of the forms of equations for all of these models can be found as the following.

1. ARMA Model

$x_{t+1} = \alpha_0 + \epsilon_{t+1} + \sum_{k=1}^{k=p} \alpha_{t+1-k} x_{t-k+1} + \sum_{l=1}^{l=p} \beta_{t+1-k} \epsilon_{t-k+1}$, with all of the ϵ following Normal distribution.

The similar setting applies to the cases of both VAR and ARX. All of these techniques assume that the predicted values will be the mean of some estimated Normal distributions, depending on the given input. For all of these time-series models we can check whether each variable in the model is significant by running the t-test on it. For this experiment, we will run the test at 10%, 5%, and 1% confidence levels.

2. Gaussian Process Regression

Let y be observations included with noise, $f(x)$ a function value over the domain of x, and ϵ an independent identically distributed Gaussian noise. Then, the noisy version of observation has the assumption that $y = f(x) + \epsilon$. The prior on the noisy observations for covariance becomes $cov(y_p, y_q) = k(x_p, x_q) = k(x_p, x_q) + \sigma_n^2 \delta_{pq}$ or $cov(y) = K(X, X) + \sigma_n^2 I$ where δ_{pq} being a Kronecker delta with $\delta_{pq} = 1$ when $p = q$ and 0 otherwise, K being a kernel matrix and X being a matrix of training dataset. Hence, we can write the joint distribution of the observations and the function value at the test location, X_* under the prior as

$$\begin{bmatrix} y \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m \\ m_* \end{bmatrix}, \begin{pmatrix} K(X,X) + \sigma_n^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{pmatrix} \right).$$

Therefore, in this model, there does not exist any explicit way of describing the descriptive statistic. The only modifications that we can make to the model will be only for the kernel K. All of the parameters in this model will be optimized under a gradient method. In this experiment, we use the L-BFGS method implemented in Scikit-learn to perform parameter optimization.

3. Variable Relationships

The model coefficients for the time-series models are as the following.

AR Model - Coefficients

	Dependent variable:
	y
AR - 1	-0.955***
	(0.084)
AR - 2	-0.995***
	(0.053)
AR - 3	-0.939***
	(0.078)
MA - 1	0.552***
	(0.173)
MA - 2	0.799***
	(0.168)
MA - 3	0.802***
	(0.208)
const	0.058***
	(0.022)
Observations	37
Log Likelihood	10.651
sigma-squared	0.026
Akaike Inf. Crit.	-5.302

Note: *p**p***p<0.01

Table 2: Coefficients of AR Model

According to the table, the ARMA model chose the number of lags for the dependent variable the number of lags for the noise as 3. The result is quite surprising for the coefficients of all dependent variable lags having all negative values, which are very close to -1. The coefficients denote the situation that the number of sales varies considerably from the negative value to the positive value and vice versa from one period to another. The lagged noises also affect the forecasting result vehemently with the all three lags having pretty positive high values. The given coefficients imply how extremely fluctuating the target variable is. Thus, adding another variable, for example a sentiment score, may assist the original model by providing better signal to it.

ARX Model - Coefficients

	Dependent variable:
	y
AR - 1	-0.739***
	(0.123)
AR - 2	-0.862***
	(0.092)
AR - 3	-0.642***
	(0.120)
normed_sentiment	0.075***
	(0.012)
const	0.065***
	(0.009)
Observations	37
Log Likelihood	10.862
sigma-squared	0.030
Akaike Inf. Crit.	-9.724
Note:	*p**p***p<0.01

Table 3: Coefficients of ARX Model

According to the table, the ARX model also chooses 3 lags of the dependent variables as the best model specification. In this model, the magnitudes of the lagged variables’ coefficients decrease with the help of the normalized sentiment score added to the model. Still, they have the same negative patterns as before. This pattern is predictable as, again, the original data fluctuate vehemently across different periods. Apart from the lagged target variables’ coefficients, the positive coefficient from the normalized sentiment score illustrates a hopeful signal that the generated sentiment score may reflect the reality that if the sentiment about the product is positive, there should be a greater chance that the number of sales will increase. We will be able to expand the idea about an effect of public sentiment on the sales growth in the next part after running the VAR model.

VAR Model - Coefficients

	Dependent variable:
	y
adj_sales.l1	-0.715***
	(0.140)
sentiment_val.l1	0.124**
	(0.053)
adj_sales.l2	-0.867***
	(0.124)
sentiment_val.l2	-0.026
	(0.057)
adj_sales.l3	-0.669***
	(0.146)
sentiment_val.l3	0.157***
	(0.047)
const	0.221***
	(0.041)
Observations	34
R^2	0.749
Adjusted R^2	0.693

Residual Std. Error	0.185 (df = 27)
F Statistic	13.408*** (df = 6; 27)
Note:	*p**p***p<0.01

Table 4: Coefficients of VAR Model

According to the table, the VAR model also chooses the number of target variable lags and the number of sentiment value lags as 3. Similar to the former cases, the coefficients for the lagged variables are still negative. The additional coefficients on sentiment values are now positive for the first lag and the third lag, but negative in the second lag. The negative value in the second lag is quite counterintuitive. However, as this coefficient value is not statistically significant at 10% level, we cannot guarantee that this finding will hold for the case when more observations are collected. Excluding this variable, we can conclude that, according to this dataset, the number of iPhone sales positively correlate with the lagged sentiment scores.

For the Gaussian Process Regression model, as it does not generate any explicit functional form as the time series models do, the preliminary result obtained from running the model will be only the R^2 value. Unfortunately, it seems that the model encounters an overfitting problem because the obtained R^2 value is very close to 1 even after several experiments on modifying the kernels. Therefore, we should not expect the great performance from this model in the forecasting result either.

IV. Model Performance

To evaluate the performance of the model, we would like to recall the goal of our project. Our final goal is to explore whether overall sentiment on the Reddit will help with predicting the sales of iPhones. That is to say, we are trying to explore the interactions between people's attitude on the internet and actual sales of a product in the real life. From our intuition, we can think of that if people's attitude towards a product drops, which means there exists an overall dissatisfaction of a product, the sales will have a corresponding drop. And the converse also applies. This project is on the application aspect, as it mainly concentrates on the combination of real comments on the internet and the real sales of products.

With all the comments from 2009 to 2018 crawled from the Reddit, we implement necessary preprocessing steps including lemmatization and remove of stopping words using NLTK package. Then we implement VADER algorithm to calculate the sentiment score of each post, and normalize all the score using standard normalization. As the sales data is on the basis of quarter, we calculate the average quarterly sentiment scores accordingly.

For the time series part, we firstly conducted Augmented Dickey-Fuller test on both variables. The results of the normalized sentiment score are: Dickey-Fuller = -1.996, Lag order = 3, p-value = 0.575, alternative hypothesis: stationary. As the p-value is 0.575 and alternative hypothesis is stationary, we cannot reject the null hypothesis that a unit root is present in the normalized sentiment scores. For sales data, the results are: Dickey-Fuller = 0.070, Lag order = 3, p-value = 0.99, alternative hypothesis: stationary. And we can see that the p-value does not allow us to refuse the null hypothesis that a unit root is present. Hence, we will consider logarithmic differences of the numbers of sales. After the logarithmic differences, we get the following results: Dickey-Fuller = -3.850, Lag order = 3, p-value = 0.028, alternative hypothesis: stationary, in which situation we can confidently refuse the null hypothesis of present of unit root, and take the alternative hypothesis that it is stationary.

Secondly, we conducted exponential smoothing state space model with Box-Cox transformation on both time series to detect seasonality in the data. Exponential smoothing is a rule of thumb technique for smoothing time series data using the exponential window function. Whereas in the simple moving average the past observations are weighted equally, exponential functions are used to assign exponentially decreasing weights over time. From the seasonality test results, we come up with the conclusions that there exists seasonality in the sales growth, but not in the sentiment value. Thus, we removed the seasonality in the sales growth.
Next, after all these preprocessing and tests of data, we fit different time-series model on the dataset. For the specialty of time-series data, we could not implement traditional train test split methods on our data, which can cause data leakage if used. In addition, it is a multi-variable time series model, which requires more special treatment. For our model, we implemented forward chaining method to fit the model. Using this method, we successively considered each quarter as the test set and all previous quarters as the training set. For example, if our dataset has five days, then we would produce four different training and test splits.

As for dealing with multiple time series, there are two types of methods: regular and population informed. The basic idea of regular nested train test split is the same as stated above. For population informed nested train test split, we broke the strict temporal ordering for the independence between different participants' data. For our model, we implemented the regular nested train test split.

For our metrics, since it is a regression problem, we consider Root Mean Square Error (RMSE) as the accuracy metric to measure the model performance.
Then we implement four different time series models: Autoregressive moving average model (ARMA model), Autoregressive moving average model with exogenous inputs model (ARMAX model), Vector Autoregression model (VAR model), and Gaussian Process Regression (GPR). Our baseline model is ARMA, which is simply the merger between AR and MA models. It attempts to capture the shock effects observed in the white noise terms and explain the momentum and mean reversion effects observed in trading markets. In the baseline, we only take account of the time series structure in the number of sales alone, and implement forward chain method to get the best estimation. When considering model selection, we use the Akaike information criterion (AIC) to measure the quality of model, and select the model ARMA (p, q) with the lowest AIC. The range of p and q is [0,4]. For the best ARMA model, we got an RMSE of 0.264, where p=3 and q=3.

ARMAX is the improved version of ARMA. It takes account of exogenous inputs terms. The notation ARMAX (p, q, b) refers to the model with p autoregressive terms, q moving average terms and b exogenous inputs terms. In this model, we take account of the impacts of sentiment scores on the variation of number of sales, which means setting sentiment scores as xreg in the ARIMA function, and implement forward chain method to get the best estimation. When considering model selection, we use the Akaike information criterion (AIC) to measure the quality of model as previous, and select the model ARMAX (p, q) with the lowest AIC. The range of p and q is [0,4]. For the best ARMA model, we got the RMSE of ARMAX equal to 0.409, where p=3 and q=0. As we can see, the ARMAX model does not beat baseline model. The main reason maybe that we enforce a restriction that the number of sales has to rely on the sentiment scores, but in fact there might not exist a strong relation between these 2 variables. They can affect each other in some way, but might not in this specific way.

As for VAR model, it can capture the linear interdependencies among multiple time series. It generalizes the univariate autoregressive model (AR model) by allowing for more than one evolving variable. All variables in a VAR enter the model in the same way: each variable has an equation explaining its evolution based on its own lagged values, the lagged values of the other model variables, and an error term. Hence, in the VAR model, we assume that there exist bidirectional relations between the two variables, and both of them should provide some forecasting power for one another. When considering model selection, we use the Akaike information criterion (AIC) to measure the quality of model as stated above, and select the model VAR (p) with the lowest AIC. The range of p is [1,4]. For the best VAR model, we got the RMSE of VAR model equal to 0.189 when p is 3. Compared to the baseline model, there is an improvement of 28.4% in RMSE for VAR model. Since in VAR we consider the bidirectional relations between the two variables, it agrees with our intuition that the results should be better than the baseline model which only takes number of sales into account. When comparing VAR with ARMAX, we can find that the results are better for VAR, and it agrees with our assumption that both variables can affect each other, and that we cannot enforce the restrictions that only the number of sales relies on sentiment analysis.

GPR model is a nonparametric kernel-based probabilistic model. As stated above, it has a small requirement on the number of observations, and we can use four kernels: Squared Exponential Kernel, Matern Kernel, Radial Basis Kernel, and White Noise Kernel altogether to explain different components of the sales growth. In such condition, they have additive properties. In addition, we find that including variable the number of posts of each quarter helps with improving RMSE, approximately from 0.28 to 0.24.

To reach the best results, we use different combination of four kernels. We firstly tried each kernel individually and compare the results.

Kernels	RMSE (num_posts included)	RMSE (num_posts not included)
Squared Exponential Kernel (k1)	0.243	0.306
Matern Kernel (k2)	0.258	0.344
Radial Basis Kernel (k3)	0.277	0.331
White Noise Kernel (k4)	0.332	0.332

Table 5: RMSE for individual kernels

From this part, we can find that Squared Exponential Kernel has the best performance of four, and in this case, including the number of posts as another variable improves the RMSE by 20.7%. Next, we tried different combinations of two kernels among these four. To make it clear, we abbreviate each kernels to k1, k2, k3 and k4 (as shown above in the table).

Kernels	RMSE (num_posts included)	RMSE (num_posts not included)
k1+k2	0.242	0.330
k1+k3	0.243	0.308
k1+k4	0.243	0.287
k2+k3	0.271	0.348
k2+k4	0.272	0.288
k3+k4	0.294	0.290

Table 6: RMSE for kernel combinations

From the table, we can find that when k1, Squared Exponential Kernel is part of the kernels, RMSE outperforms other combinations a lot with the number of posts included. And for the situation the number of posts not included, k4, White Noise Kernel helps a lot with improving the RMSE. Then we try 3 and 4 combinations of kernels.

Kernels	RMSE (num_posts included)	RMSE (num_posts not included)
k1+k2+k3	0.260	0.334
k1+k2+k4	0.243	0.287
k1+k3+k4	0.243	0.287
k2+k3+k4	0.268	0.288
k1+k2+k3+k4	0.243	0.287

Table 7: RMSE for combinations of 3 and 4 kernels

From all the information above, we can see that k1+k2, Squared Exponential Kernel and Matern Kernel with the number of posts included has the best RMSE, which is 0.242.

With all the models performed, we can find that VAR has the best RMSE of 0.189. This gives us an inspiration that sentiment scores of comments on the internet can help with predicting the number of sales in real markets. Since ARMAX does not outperform baseline, unidirectional relation between sentiment scores and the number of sales is a bad assumption. And due to the small size of the dataset, GPR has a better performance than ARMA. The accuracy of the analysis can be improved in the future if we apply techniques that can fix misspellings and slangs in the posts. Overall, these findings allow us to explore more using the information from other social media platforms such as Facebook and Twitter. Therefore, sentiment analysis on Tweets or Facebook posts can be useful as well to predict the number of sales in iPhone or other popular smart phone models that people tend to discuss frequently online.