Forecasting Bag Shipments Using Social Media Trends

By

Robert Knox, Adetola Adedeji, Xiaolei Zhang

Supervisor: Arnab Bose

A Capstone Project

Submitted to the University of Chicago in partial fulfillment
of the requirements for the degree of

Master of Science in Analytics

Graham School of Continuing Liberal and Professional Studies

June, 2019

The Capstone Project committee for Robert Knox, Adetola Adedeji, Xiaolei Zhang
Certifies that this is the approved version of the following capstone project report:

Forecasting Bag Shipments Using Social Media Trends

Approved by Supervising Committee:

Arnab Bose	Dr. Sema Barlas

Abstract

Abstract

Our research was to find exogenous variables derived from social media data in order to improve the forecasts of Scholle IPN's tomato bag shipments on a monthly basis. We leveraged two data sources to help forecast shipments, Google Trends and Twitter. Google Trends data collected the relative frequency of certain keyword searches over the research period of 2007 to 2018. Twitter data collected tweets containing keywords over the research period of 2007 to 2018. These tweets were then processed using open-source natural language processing libraries to determine the tweet's sentimentality of positive or negative. Several models were created, validated and ensembled to produce the best model to forecast tomato bag shipments.

Keywords: Forecast, Shipment, Social Media, Regression, Time Series, Random Forest, XGBoost,Ensemble Model.

Executive Summary

OBJECTIVE: Our research was to find exogenous variables derived from social media data in order to improve the forecasts of Scholle IPN's tomato bag shipments on a monthly basis.

METHODS: We used natural language processing and time-series analysis to forecast tomato bag shipments.

RESULTS: We found models that leveraged...

CONCLUSIONS: We were able to ...

Table of Contents

List of Figures

List of Tables

Preface

A preface is OPTIONAL. Use a preface if you want to explain your interest in the report topic and include anything about your experience that readers should keep in mind. If you would rather not include a preface, comment it out or delete it from the YAML header of the index.Rmd file.

Introduction

Founded in 1945, Scholle IPN Corporation (Scholle) is a pioneer in the bag-in-box technology. The company uses to package and dispense a variety of consumer products, from food, beverages to non-food markets such as agricultural chemicals and liquid cleaning products. Scholle mainly produces bag-in-box, pouch packaging, and packaging components like caps, pumps and connectors. Some of Scholle's competitors in the industry are DuPont Liquid Packaging Systems, Sonoco Products Company and Survitec Group Limited. As a industry-leader, Scholle is constantly providing products in a safe, economic and sustainable way to customers globally.

One of the company's primary markets is bags for processed-tomato products, such as salsa, tomato paste and ketchup. Every year, the company forecasts the demand for bags for these products, but the accuracy of the prediction is low because the current forecasting methods only considers the quantity of bags shipped in the previous year.

Problem Statement

Scholle needs a more accurate method to forecast its bag shipments. One of the company's primary markets is packages for processed-tomato products, such as salsa, tomato paste and ketchup. Every year, the company forecasts the demand for bags for these products, but the accuracy of the prediction is low because the current forecasting method only considers the quantity of bags shipped in the previous year. Social media is changing the world. It provides companies with feedback on customer conversations about their products and services, and can affect demand for their products. Currently, this potentially rich data set is not factored in the forecasting of tomato bag shipments. Scholle Management believes there is a relationship between social media feedback and the quantity of tomato bags shipped. Scholle hopes to improve their tomato bag inventory management, level of stock turnover and fulfill the demand more accurately by incorporating these data into their predictive models.

Research Purpose

The purpose of this research is to forecast tomato bag shipments using social media data about certain keywords associated to process tomato products.

Our research objectives include: * Develop a methodology to determine the positive or negative sentiment associated to a specific product. * Investigate several data sets to find significant cross-correlation with Scholle data. *

Variables & Scope

The key dependent variable we will investigate will be the monthly aggregated shipment quantity of tomato bags. These will be limited to the bags sold in the United States. Our independent variables will be derived from Google Trends and Twitter data generated by users in the United States.

Background

Social Media is changing the world. It provides companies feedback on customer conversations of their products and services, and can affect demand for their products. Currently, this important metric is not factored in the forecasting of tomato bag sales.

Scholle Management believes there is a relationship between social media feedback and quantity of tomato bags sold year. Scholle hopes to improve their tomato bag inventory management, level of stock turnover and fulfill the demand more accurately.

This project collected, analyzed and generated solutions on how social media conversation correlate with the demand for processed tomato products and factors those findings into the prediction of tomato bag shipments.

Methodology

Data

Scholle Data

Scholle provided a data file in the form of an excel document that included sales record data from 2010-2018. The following table describes the relevant columns from the data file.

INSERT TABLE OF SCHOLLE COLUMNS HERE.

Based on discussions with Scholle's management, we removed any sales transactions that were marked with a Quantity less than or equal to zero. This was due to the data being extracted from their sales system in which negative or zero quantity records were used to true up returns or modify existing order pricing which were not relevant to our analysis.

Descriptive Analyses

Exploratory Data Analysis

We began with exploratory data analysis to help understand the Scholle data and decide how to approach the problem of forecasting bag sales. Quantity is the primary focus as it describes the number of bag shipments.

Distribution of Quantity

Order quantity is typically less than 50,000 bags, with a few orders significantly higher. Specifically, most order quantities are less than 30,000 bags. See 1 below for histogram of Quantity.

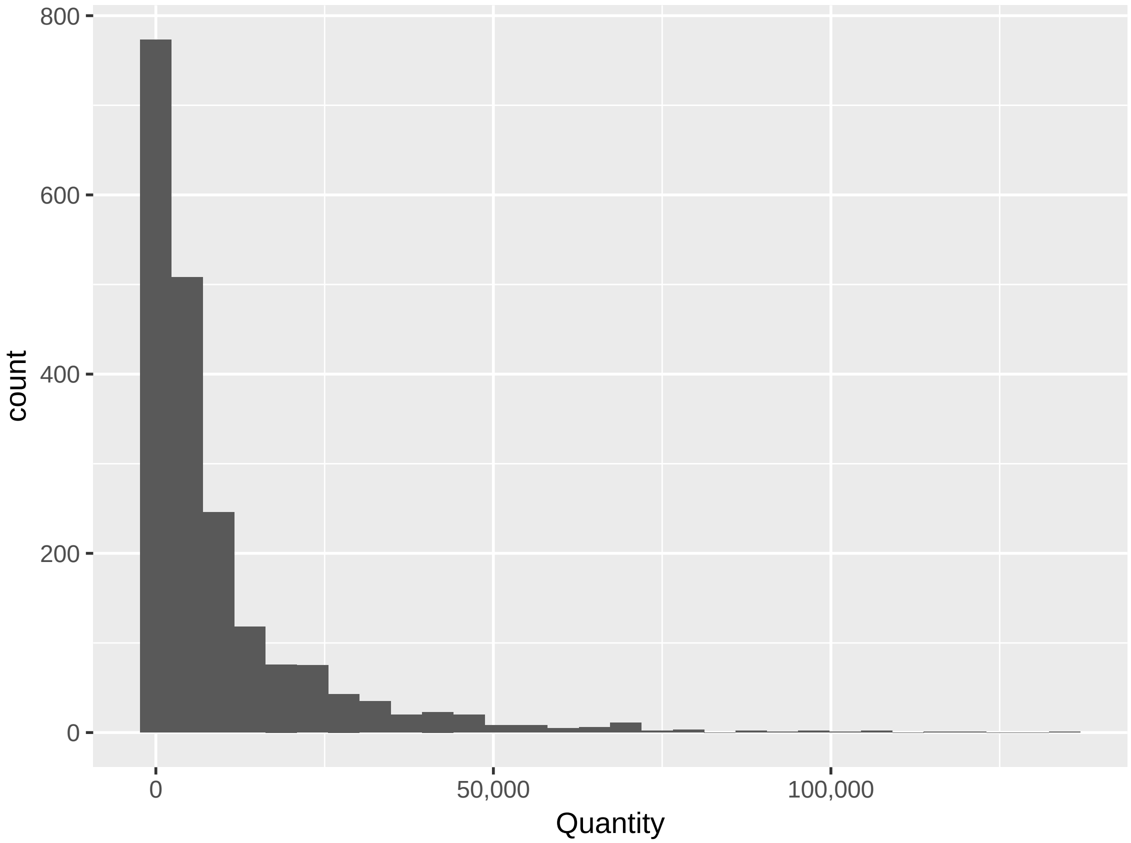


Figure 1: Histogram of Quantity

Quantity is not normally distributed but is instead heavily skewed left. Large quantity orders are atypical and could have significant impact on models.

Distribution of Quantity over time {methodology-QvsT .unnumbered}

The figure below plots the aggregated Quantity over time.

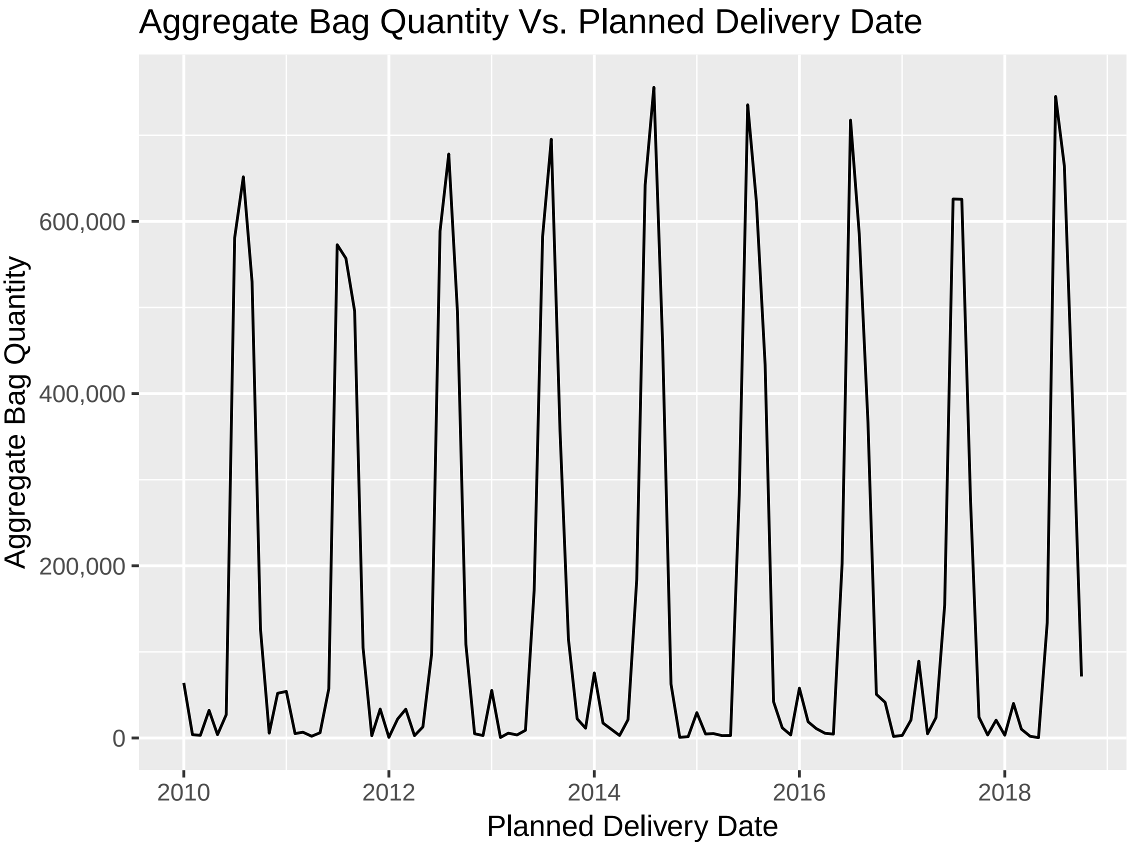


Figure 2: Quantity over Time

There is a large degree of seasonal behavior. Tomato bag shipments increase during summer time from June to August and begin to drop off in September and October. Quantity is very small during other months of the year. This seasonality is easier to observe in the chart below which aggregates bag shipments for each month of the year.

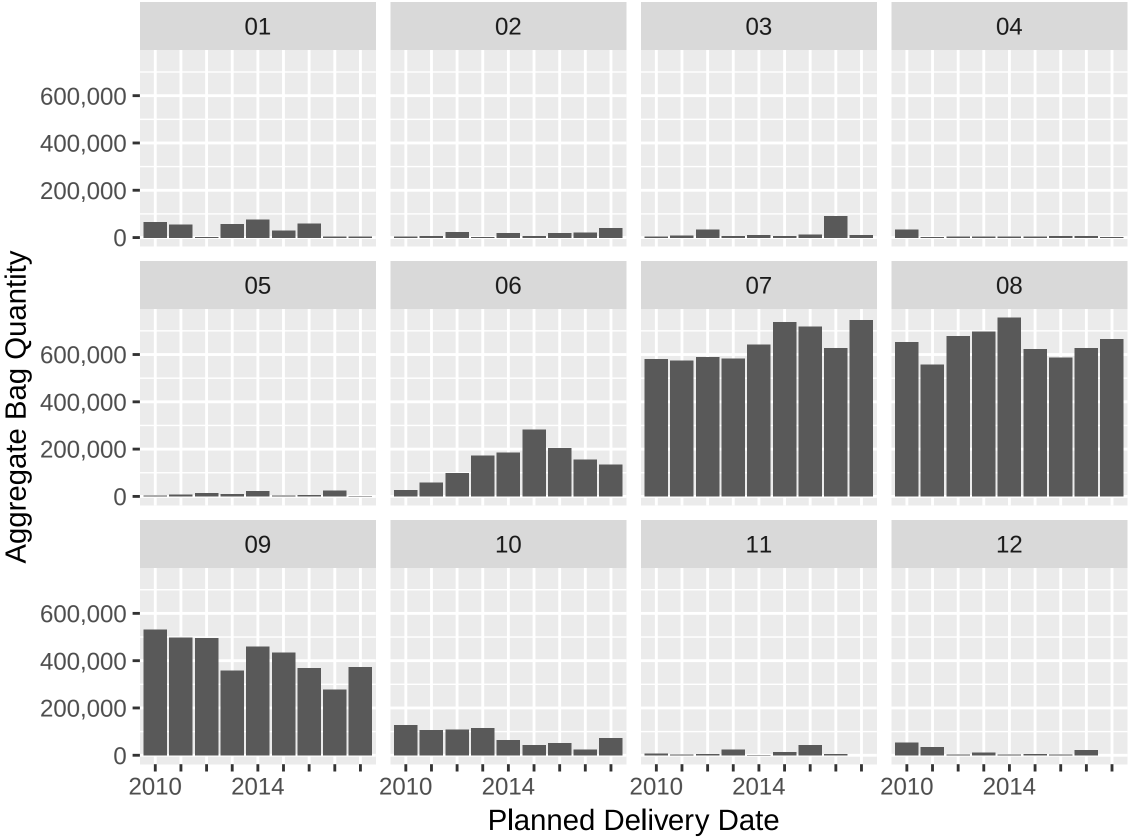


Figure 3: Monthy Quantity by Year

There is little year to year variation exhibited in each month, with the possible exception of June which increased from 2010 to 2015 and then decreased from 2016-2018.

Quantity by Item Number

In addition to exploring the shipment quantity by time, we also observed the effect that the Item Number on Quantity. Over 86% of shipments are driven by the top 10 out of a total of 60 item numbers.

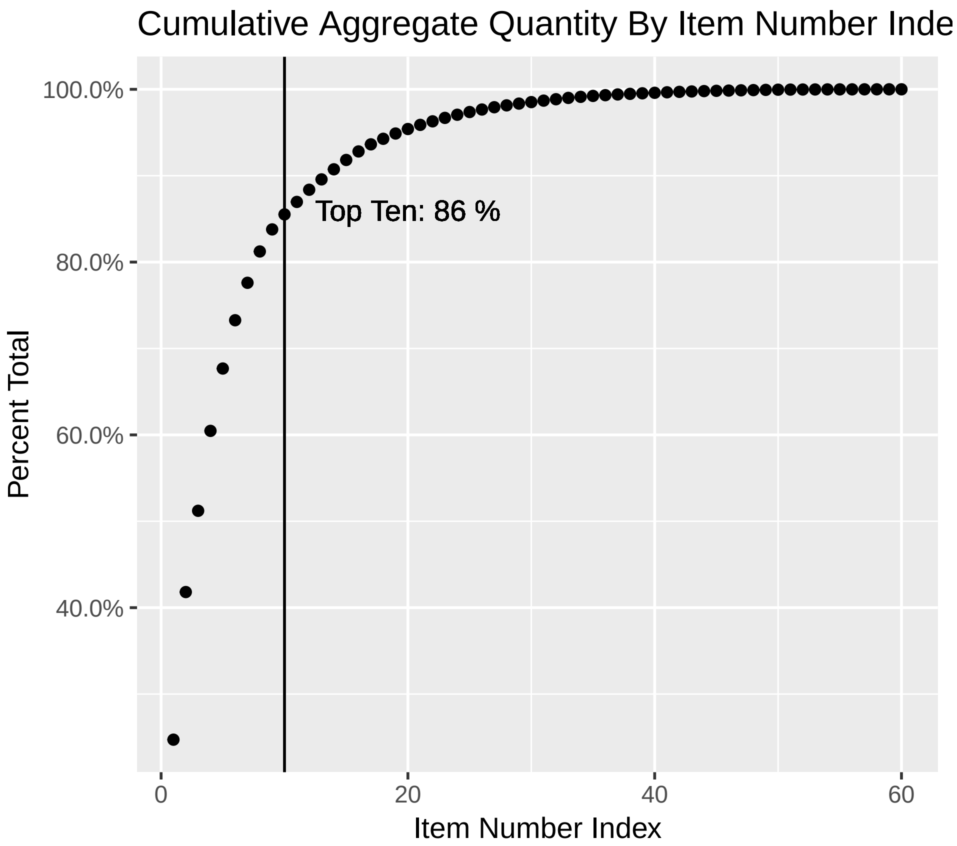


Figure 4: Cumulative Sales by Item Number

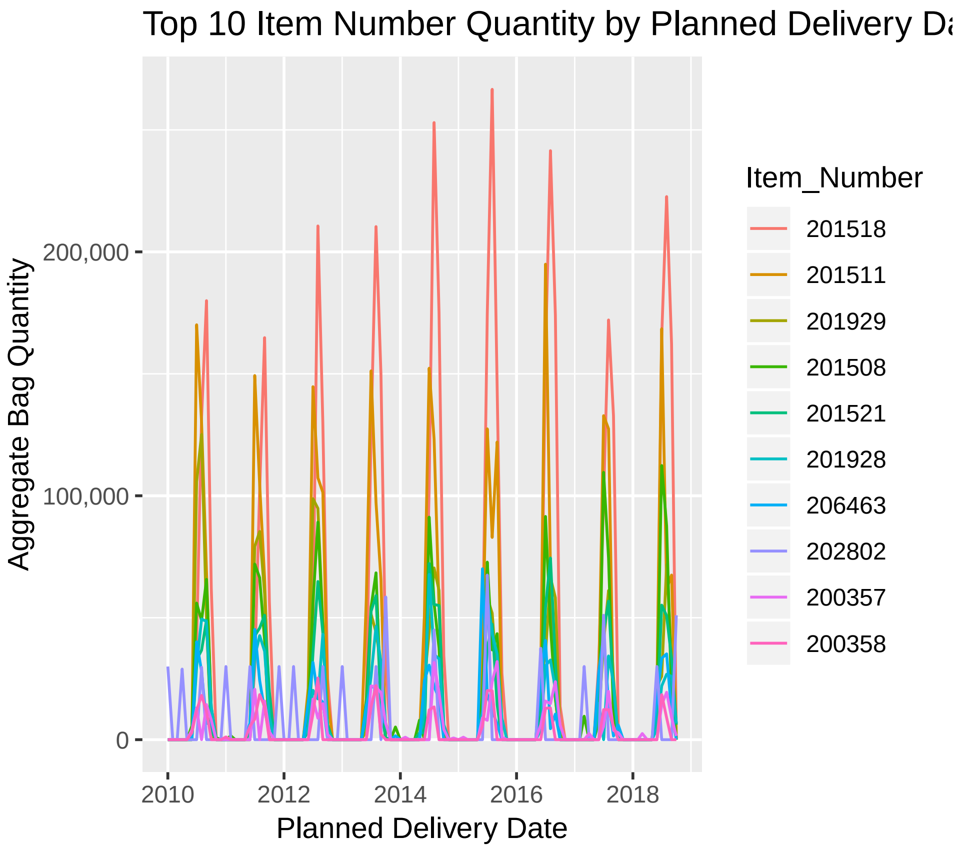


Figure 5: Top 10 Item Number Quantity over Time

Quantity by Item Number does not greatly vary greatly from the aggregated Quantity. Because the seasonal nature of the data is so strong, we will consider the shipment quantity aggregated to the monthly level.

Descriptive analyses

External Data Collection

Google engine search and twitter are important and representative social media sources whose data is accessible. We collected social media data from two sources: Twitter and the Google Trends. Google Trends summarized US monthly search statistics from 2004 to 2019. The twitter data collected are of US users from 2007 to 2018.

Google Trends returns a single value showing how frequently a given search term (e.g. 'Tomato') goes in Google's search engine relative to its total search frequency for a given period.

Twitter data were gathered by collecting relevant tweets from Twitter using key search words as shown in Table tab below. We can grab a tweet based on defined keyword from Twitter by calling the Twitter API function. Subsequently, we can categorize opinions expressed in a piece of text, in order to determine opinion on our research (i.e, positive, negative, or neutral).

External Data Source	Frequency Granularity	Range	Size	Data Type
Google Trend	Monthly	2004-2019	182	Numeric
Twitter	Monthly	2007-2019		Text, Numeric

Table **??** above shows the description of the social media dataset. The google trend dataset based on relative search frequency is on a monthly basis. The monthly dataset at the time of reporting has 182 rows.

The Twitter data are sampled via a quasi-random approach that grabs data monthly over the entire period of 2007 - 2018. We had to employ this method of querying due to the nature of the Twitter API and its restrictions on total tweets returned. The Twitter API returns tweets in reverse chronological order. The total number of tweets that would mention one of our keywords would be vastly larger than the number we could collect over the course of our data collection period (January 2019 - March 2019). Limited in this way, we decided to strategically collect tweets from each month of the year between January 2007 and March 2019.

Assumptions & Limitations

Google Trends Assumptions:

- We limited the Google Trends search to the United States.
- The keywords are independent of each other.

Limitation:

- The actual number of searches for the term being queried is not made available. Instead, the data are reported from 0-100, where 100 represents the maximum relative search frequency.

Twitter Datasets Assumptions:

- We limited Twitter data to the United States.
- The frequency of tweets we collected for each keyword will be independent of the time period in which we collected them.

Limitations: * The demographic information of the twitter account user cannot be determined.

- With the limitation of our premium account API activity, we can only submit 100 requests to collect tweet per month. Per each request, we can get a maximum of 500 tweets back.

- The language associated with the Twitter account returned from the API does not guarantee the language of the Tweet. While we specified English language tweets, we received many tweets that were not in English. We subsequently dropped these tweets.

Plan for use

Working with Scholle, we developed a list of keywords to target in our social media data collection. The keywords are listed in Table **??** below:

```
Warning in kable_markdown(x = structure(c("bbq", "chili", "ketchup",
"pasta", : The table should have a header (column names)
```

bbq	chili	ketchup	pasta	pizza	salsa	spaghetti	tomato

Twitter data We developed a sentiment index for all tweets using natural language processing techniques tilizing the textblob Library to analyze how similar or discrepant the meaning of tweets among each keyword

Google Trends Correlation analysis of frequency of individual search terms compared to quantity Subsequent seasonality & trend analysis to identify meaningful predictors

Twitter Data

We have collected tweets based on the keywords in Table below. Tweets were aggregated on a monthly basis. The figure below shows how frequently the keywords appears in the returned tweets over time. Pizza is more frequently mentioned in Twitter than other keywords.

Google Trend Analysis

We collected Google Trend data consisting the same keywords as we did in Tweet collection. The Google Trends data was standardized and subsequently the cross correlation was found with respect to the Scholle bag sales. Figure 6 displays the results of the cross correlation analysis.

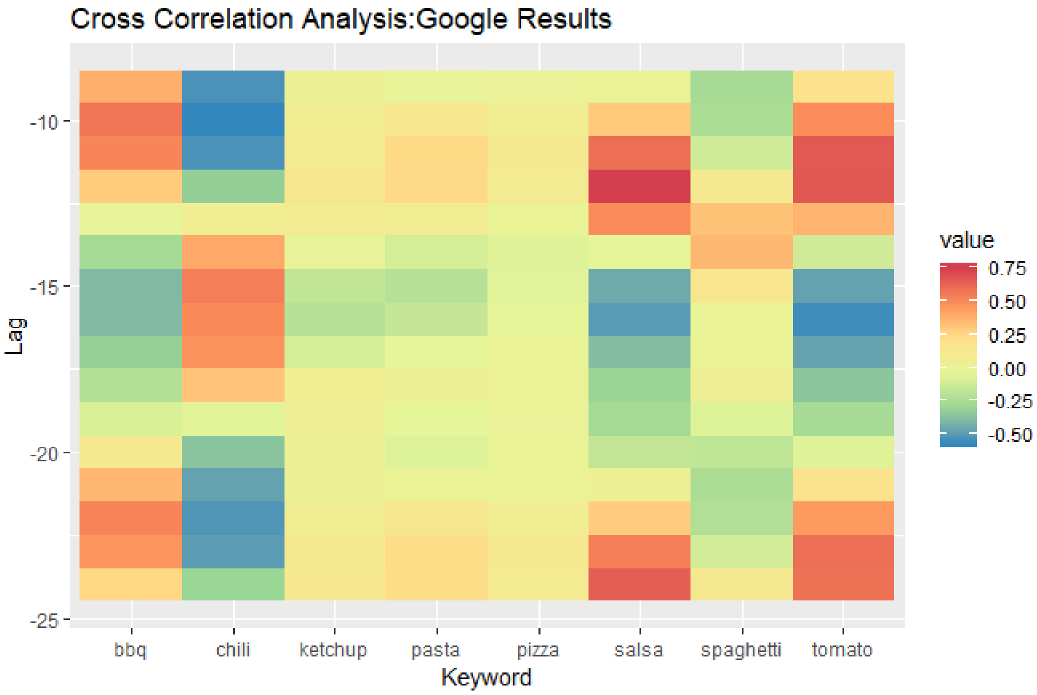


Figure 6: CCF Google

bbq	chili	ketchup	pasta	pizza	salsa	spaghetti	tomato
Lag 10	Lag 15	None	None	None	Lag 12	None	Lag 12

In Table **??** above, we highlight cross-correlations greater than 0.5 and with a time period of greater than 8 months. Since the Scholle bag sales data are records of the demand are placed well in advance of the desired delivery date, we would expect a long lag window in order for the trends of social media to drive market forces that would affect demand. Red indicates positive relations and blue indicates negative correlations. The deeper the color, the greater its correlation with Scholle's data.

Tweet Sentiment Analysis

We have conducted the following steps to conduct the sentiment analysis of the tweets.

Text Cleanup Pipeline:

1. Remove RT, URLs and non-text characters (except @ and punctuation symbols)
2. Handle mentions by replacing with upper case letters.
3. Remove all remaining non-text characters (including @ symbol and excluding punctuation symbols).

4. Check the language of the cleaned up tweet and drop any tweets that are not in English.

SpaCy Pipeline:

We used the spaCy English Core Web Large model to analyze each tweet to process into three data sets:

1. Tokenization - each tweet was broken into its component elements of words, punctuation, etc.

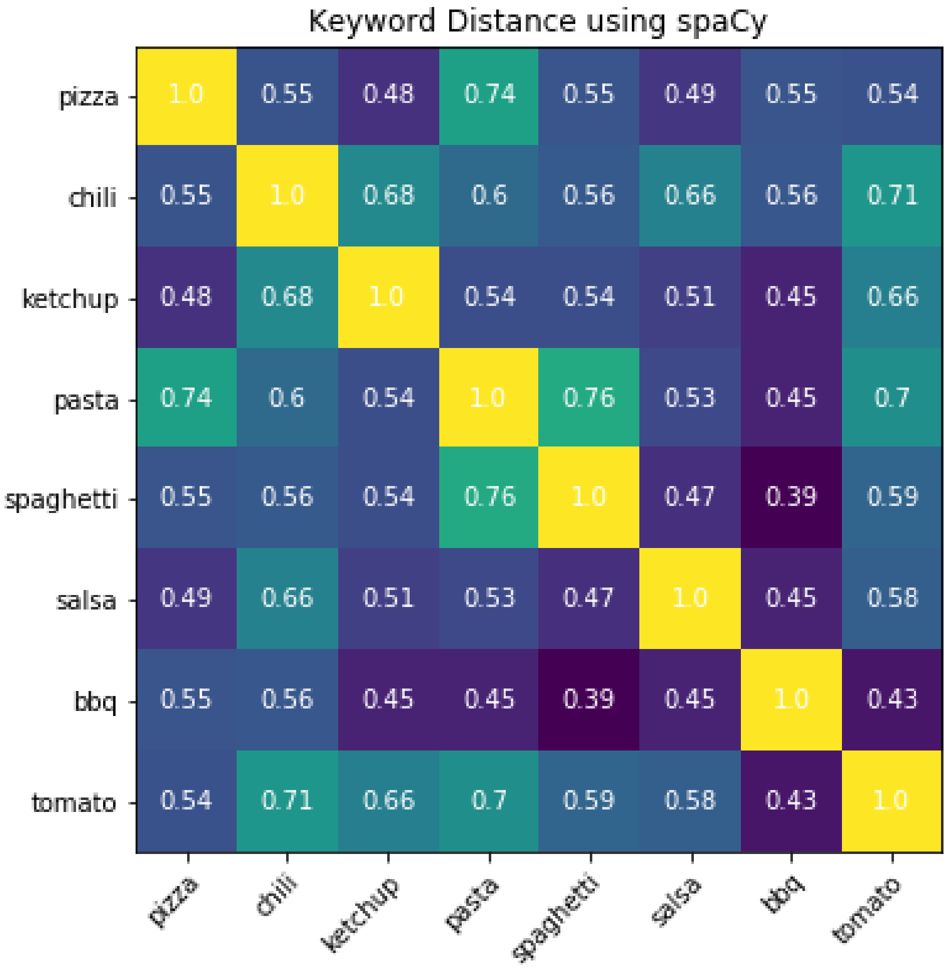
2. Dependency Parsing - Annotate the tweet to add the syntactic dependency within the tweet i.e. compare link verbs to their respective nouns.

3. Named Entities - Each tweet was analyzed to identify the named entities in the tweet. These entities will include the mentions because they were capitalized in the Text Cleanup Pipeline.

4. Removal of stop words - all stop words identified were removed.

Vector Extraction:

Spacy includes vector representations for individual words as well as entire entire sentences. See Figure below for the Keyword Distance using Spacy.These are represented as 300 dimension Numpy arrays. To begin, we confirmed that were was a reasonable cosine distance measure between each of the keywords.



	pizza	chili	ketchup	pasta	spaghetti	salsa	bbq	tomato
pizza	1.0	0.55	0.48	0.74	0.55	0.49	0.55	0.54
chili	0.55	1.0	0.68	0.6	0.56	0.66	0.56	0.71
ketchup	0.48	0.68	1.0	0.54	0.54	0.51	0.45	0.66
pasta	0.74	0.6	0.54	1.0	0.76	0.53	0.45	0.7
spaghetti	0.55	0.56	0.54	0.76	1.0	0.47	0.39	0.59
salsa	0.49	0.66	0.51	0.53	0.47	1.0	0.45	0.58
bbq	0.55	0.56	0.45	0.45	0.39	0.45	1.0	0.43
tomato	0.54	0.71	0.66	0.7	0.59	0.58	0.43	1.0

Figure 7: Spacy Keyword Vector Distance

There is a reasonable distance between each of the keywords with the exception of bbq and barbecue but this is to be expected since they reference the same thing.

In addition, vectors representations of each tweet can be extracted. For each tweet keyword we summarised all vectors by finding their mean values for each dimension. We then found the pairwise distance measures for these ‘average’ tweets. See in the figure below

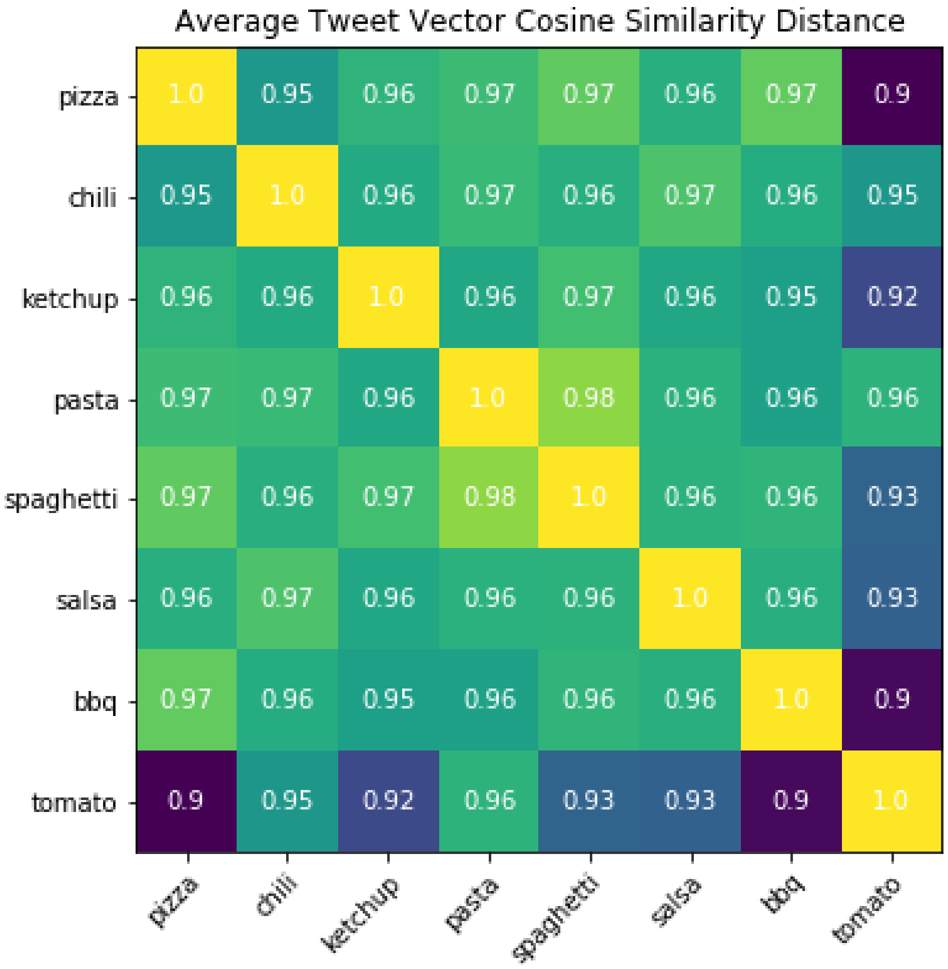


Figure 8: Spacy Average Tweet Vector Distance

Here, the "average" tweets are rather similar to each other with the greatest distance from tomato to salsa.

In future work we will leverage these vector representations of the tweets to conduct transfer learning to identify tweet sentiment.

Sentiment Analysis using TextBlob

TextBlob is an open source Python library for conducting natural language processing. It has a built in sentiment analyzer that utilizes two axes of analysis Polarity and Subjectivity. Polarity refers to a positive or negative sentiment and ranges from positive one to negative one respectively. Subjective expresses the subjectivity or objectivity of the text. See Figure 24 below for the Tweet Polarity. The subjectivity axis ranges from zero to positive one where 0 is very objective and 1 is very subjective. See Figure 25 below for the Tweet Subjectivity .

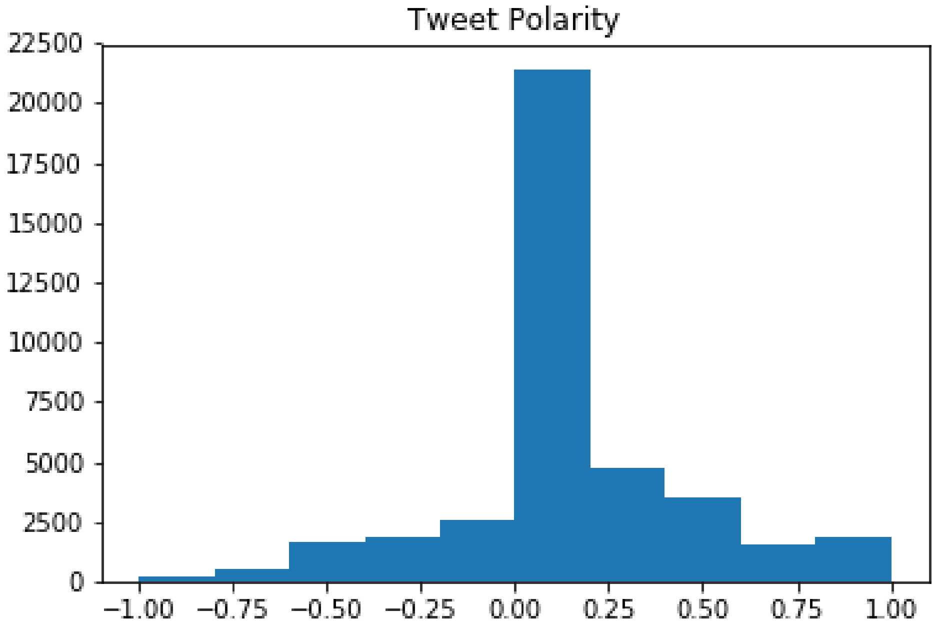


Figure 9: Textblob Tweet Polarity

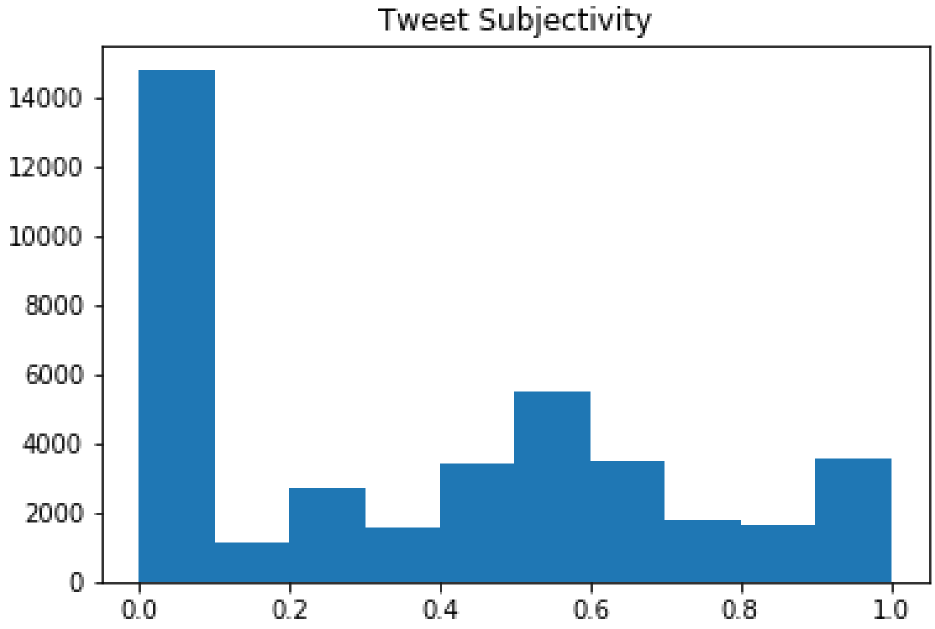


Figure 10: Textblob Tweet Subjectivity

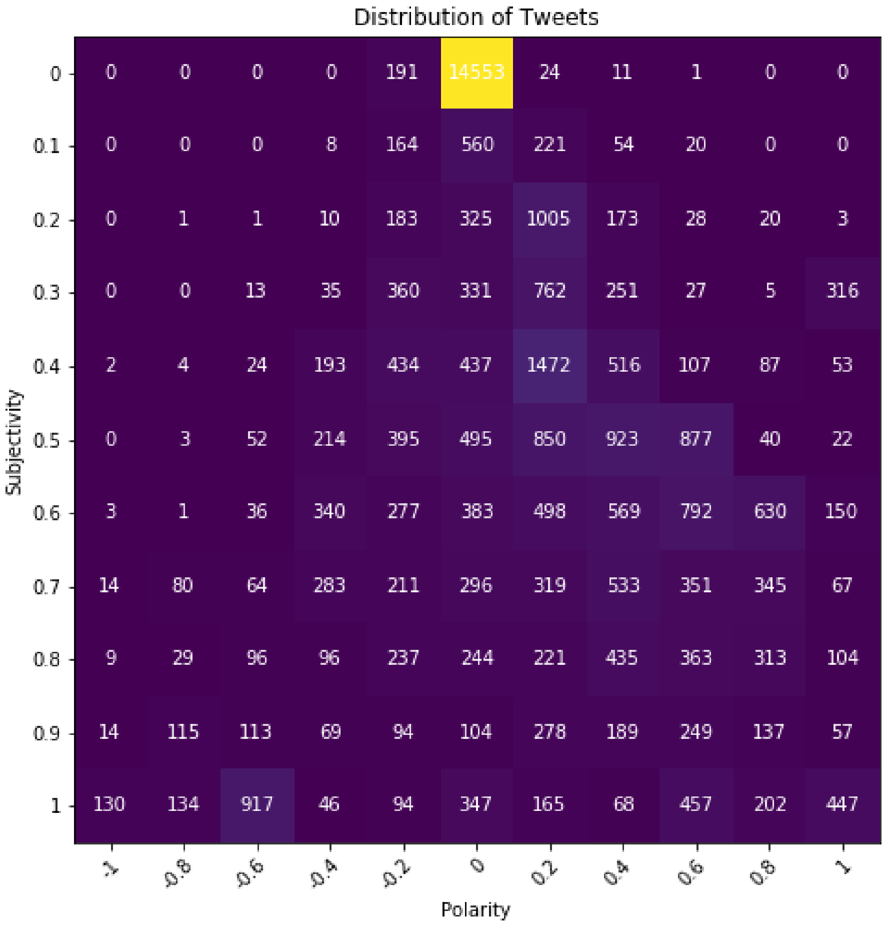


Figure 11: Textblob Tweet Distribution

TextBlob categorizes the vast majority of tweets as non-subjective non-polar.

With the tweets collected, we generated a sentiment index by taking each tweet's subjectivity & polarity and multiplied them by the retweet count for that tweet. We then aggregated the sentiment index at the monthly level for each keyword. The cross correlation between the Scholle data and the sentiment monthly index for the overall sentiment and for each keyword was calculated and the results displayed in the figure below.

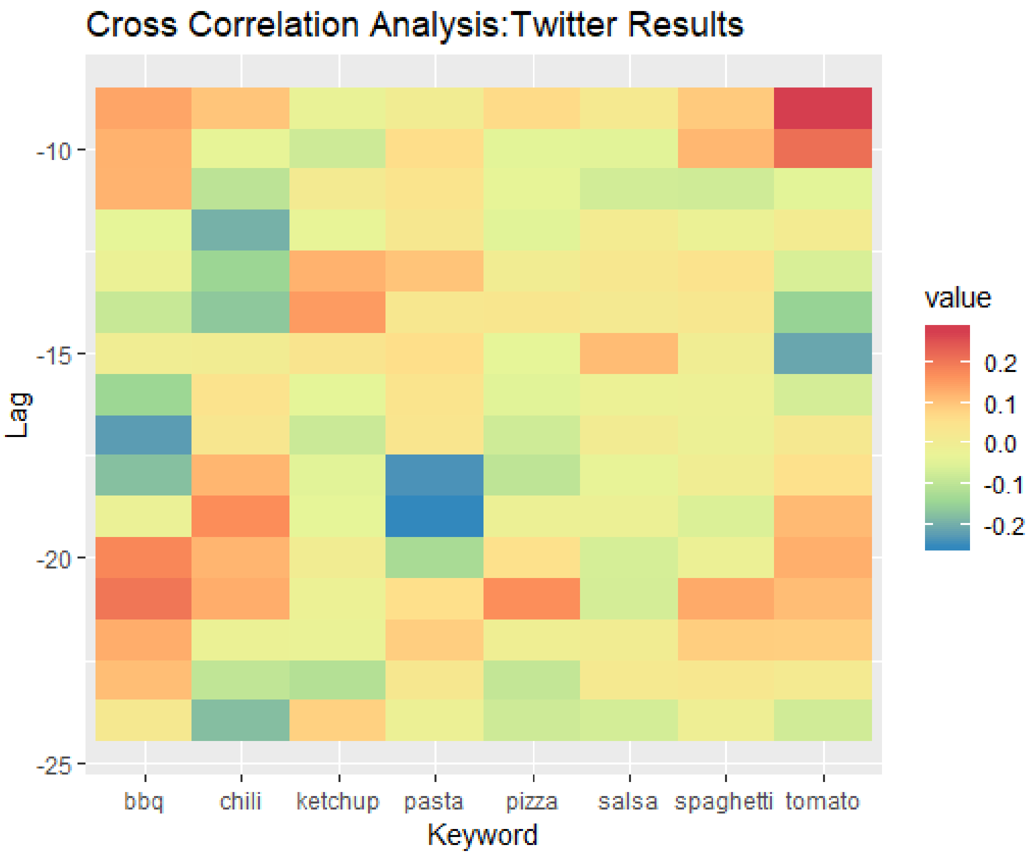


Figure 12: Twitter CCF Results

It is important to note the difference in scale relative to the Google Trends results; the correlations to the Twitter sentiment index are much weaker.

The table below compiles the list of lagged values used in our analysis.

bbq	chili	ketchup	pasta	pizza	salsa	spaghetti	tomato
Lag 17	None	None	Lag 18	None	None	None	Lag 15

Modeling Framework

Model Selection Metrics

In order to determine the best model for predicting bag shipments, we began by choosing selection metrics to test each model. The following metrics outlined below will be used to measure the performance of each model. 1. SMAPE 2. RMSE 3. % bias – no of time above forecast vs below. 4. Accuracy

sMAPE -Symmetric Mean Absolute Percentage Error

Symmetric mean absolute percentage error (sMAPE) is an accuracy measure based on percentage (or relative) errors. It allows us to understand error independent of scale.

$$sMAPE = \frac{100\%}{n}\frac{|A_t - F_t|}{(|A_t| + |F_t|)/2}$$

sMAPE has a lower bound of 0% and an upper bound of 100%. The best value is 0% while the worst value is 100%.

The major limitation with sMAPE is that it gives more penalties to underestimates than overestimates.

RMSE - Root Mean Squared Error

The root mean squared error (RMSE) of an estimator measures the average of the squares of the errors—that is, the average squared difference between the estimated values and what is estimated. The square root of this value is then taken to produce the RMSE.

$$RMSE = \sqrt{\frac{\sum^n (A_t - F_t)^2}{n}}$$

RMSE measures the variation between predicted and measured values and it provides us a basis for measuring the model variance on the same scale as the data.

The RMSE is a measure of the quality of an estimator. RMSE is always non-negative, as the process of squaring removes any negative signs. It also penalizes larger differences. The best value is zero while the worst value is unbounded.

In sum, the lower the RMSE, the smaller the error, the better the estimator.

Percent Bias – ratio of model high or low.

Bias refers to the propensity of a measurement process to over- or under-estimate the value of a population parameter. Percent bias (PBIAS) measures the average tendency of the predicted values to be larger or smaller than the actual values.

Percent Bias is calculated by taking the sign of the residual for each data point and setting positive values to 1 and negatives values to zero. These are then averaged and multiplied by 100 to produce a percentage. The best percentage bias is 50% - there are just as many over predictions as under predictions. The worst percentage bias is either 0% or 100% as the model is regularly over- or under-estimating the actual data.

Accuracy - Mean Accuracy between Model & Actual

Accuracy measures the closeness of a model prediction to the actual value. In our case we are measuring the mean accuracy for all model predictions versus actual values. The best accuracy measure is 1; the prediction and the actual are the same so their ratio is 1. Accuracies that diverge from 1 are bad; a value greater than 1 means the prediction was higher than the actual while a value below 1 is means the predictions was lower than the actual. We aggregate the accuracy measure for each data point and find the overall mean. One precaution to consider by using this procedure is it may mask an underlying trend in prediction of the accuracy in which the model could overestimate early and then underestimate late or other non-linear behavior.

Additional Modeling Considerations

Training and Forecasting Windows

In order to evaluate our model to avoid either overfitting or underfitting , we split our dataset into a training set and a test set. Based on our discussions with Scholle, the test set will be the subsequent 18 months. The length of the training period was chosen based on the evaluation of the model stability of our baseline model using a rolling window cross validation.

Rolling Window Cross Validation

Time series data present a unique challenge in analysis in that the data are not independent - they are collected at regular intervals over the course of time. We employed rolling window cross validation to generate multiple train and test sets from the overall data set. Initially we explored the model stability of our baseline model and used the best criteria from that in order to decide the length of the cross validation window.

Findings

Results of descriptive analyses

Because of the highly seasonal nature of the data, we created a baseline model using only Scholle data. We used periods of 2-year, 4-year and 6-year windows, to identify the forecast window period with the greatest stability.

Baseline Model - Prophet

Prophet is an open-source tool developed by Facebook to conduct time series modeling. Prophet models data using a decomposable time series model with three main components: trend, seasonality, and holidays. The focus is to model the time series via regression instead of as a generative model like ARIMA would. This is done for flexibility, the ability to handle irregularly spaced data, speed, and interpretability.

Cross-Validation Window Selection

The figures below show results of the cross-validation analysis at 2, 4 and 6 year training windows using the Prophet Model.

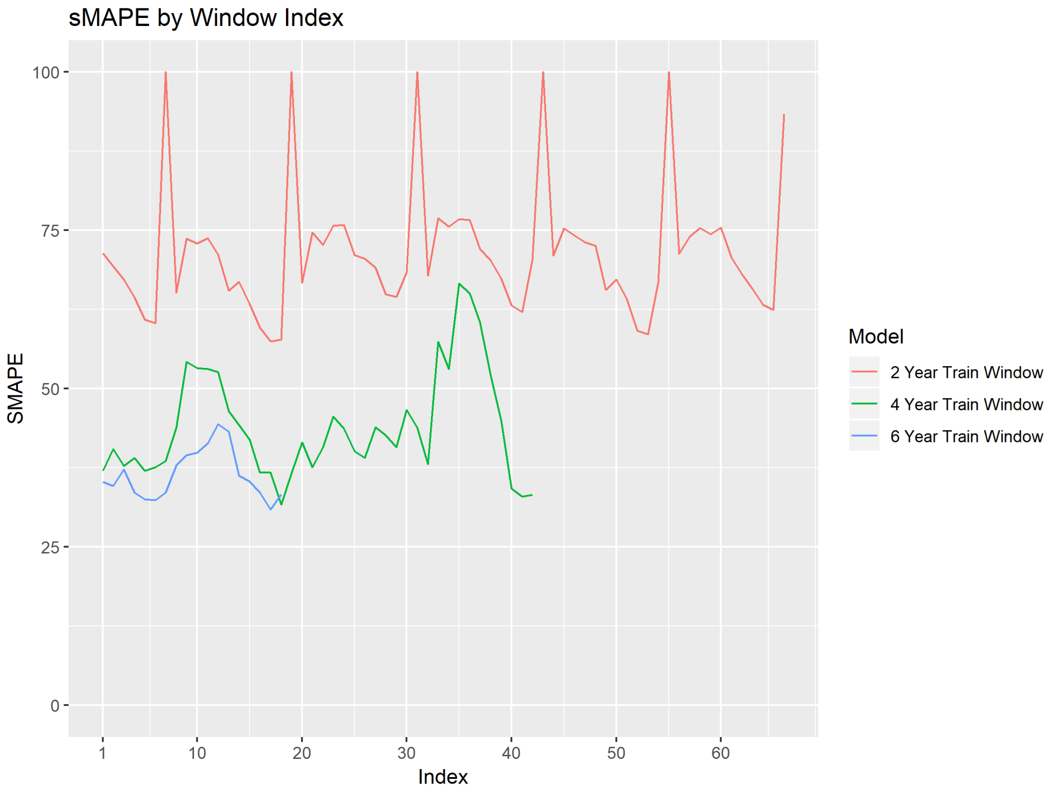


Figure 13: 2 4 and 6 year cross validation of Prophet models - sMAPE

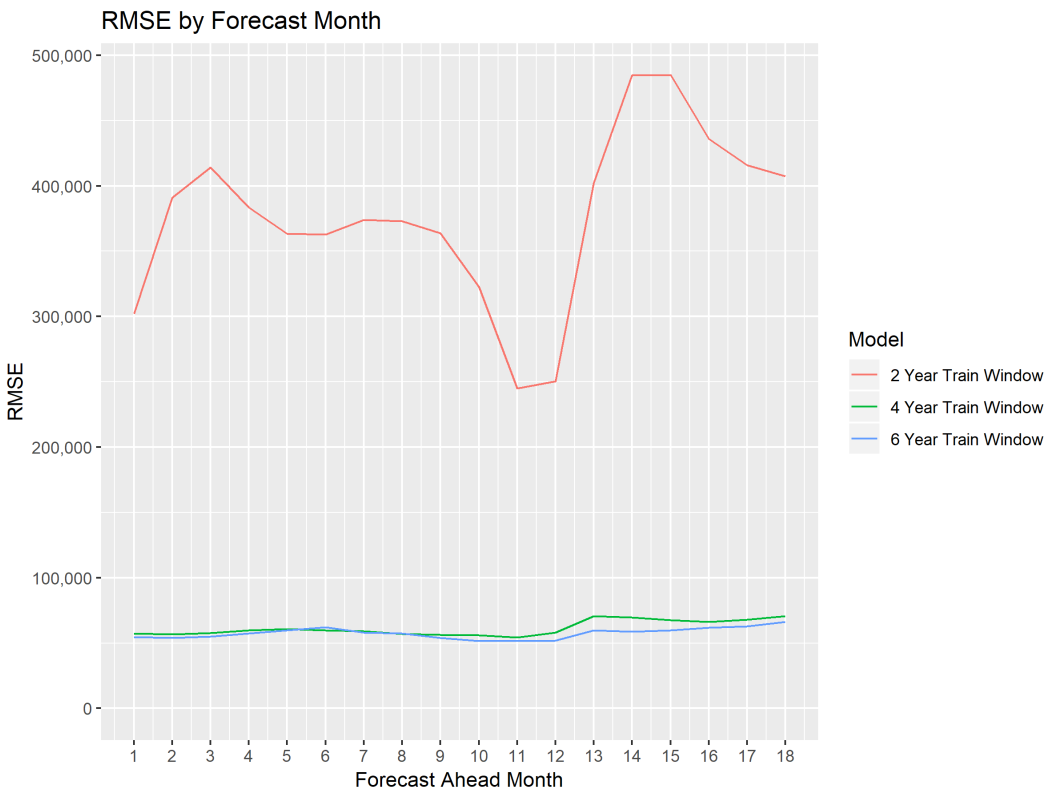


Figure 14: 2 4 and 6 year cross validation of Prophet models - RMSE

For both sMAPE and RMSE, the 2 year window shows much higher variability than the 4 or 6 year windows. Given the similarity of the 4 and 6 year windows RMSE, we chose the 4 year window to both reduce the data requirements of the model and allow for additional cross-validation of each other model. A 4 year cross validation allows us to generate 42 complete training and testing windows whereas A 6 year cross validation window only allows us to generate 18.

Baseline Model Results

The reported model metrics are the mean values for each of the cross validation periods collected.

X	sMAPE	RMSE	X..Bias	Accuracy
Train	30%	29896.68	22%	0.86
Test	43%	62625.69	67%	4.04

Challenger Model sARIMA

Seasonality is a key feature of the dataset, as it was observed that the Tomato bags sales increase significantly during summer time from June to August and drop in September and

October while repeating this cycle annually.This key attribute in the dataset meant we deploy a model that uses differencing at a lag equalling the number of seasons to remove additive seasonal effect.

For this challenger, we split the data into two groupings; harvest months (June -October) and all months. In the Table 7 & 8 below, we summarise the result for the model:

X	sMAPE	RMSE	X..Bias	Accuracy
Train	9%	125522.0	19%	1.76
Test	68%	338725.3	100%	0.43

In the figure below, the mean forecast value is highlighted in blue and the actual value is captured in red, and the confidence interval ranging between 80%-95%. The actual values are represented in black.

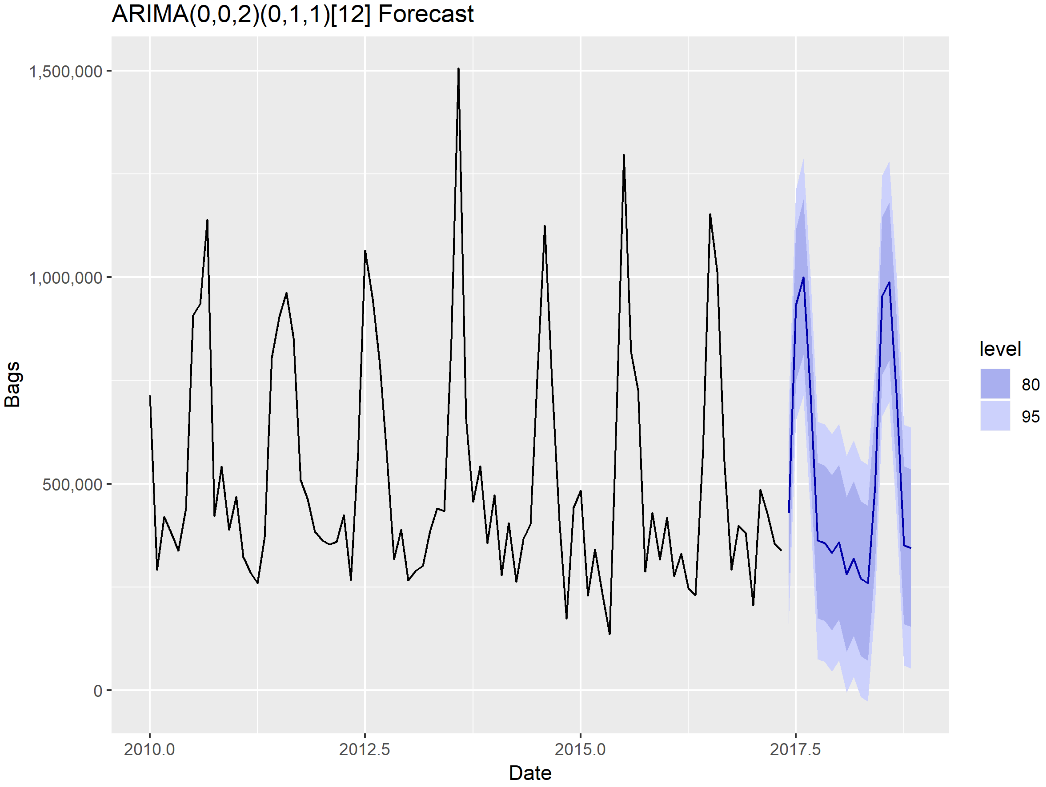


Figure 15: 12 Month Arima

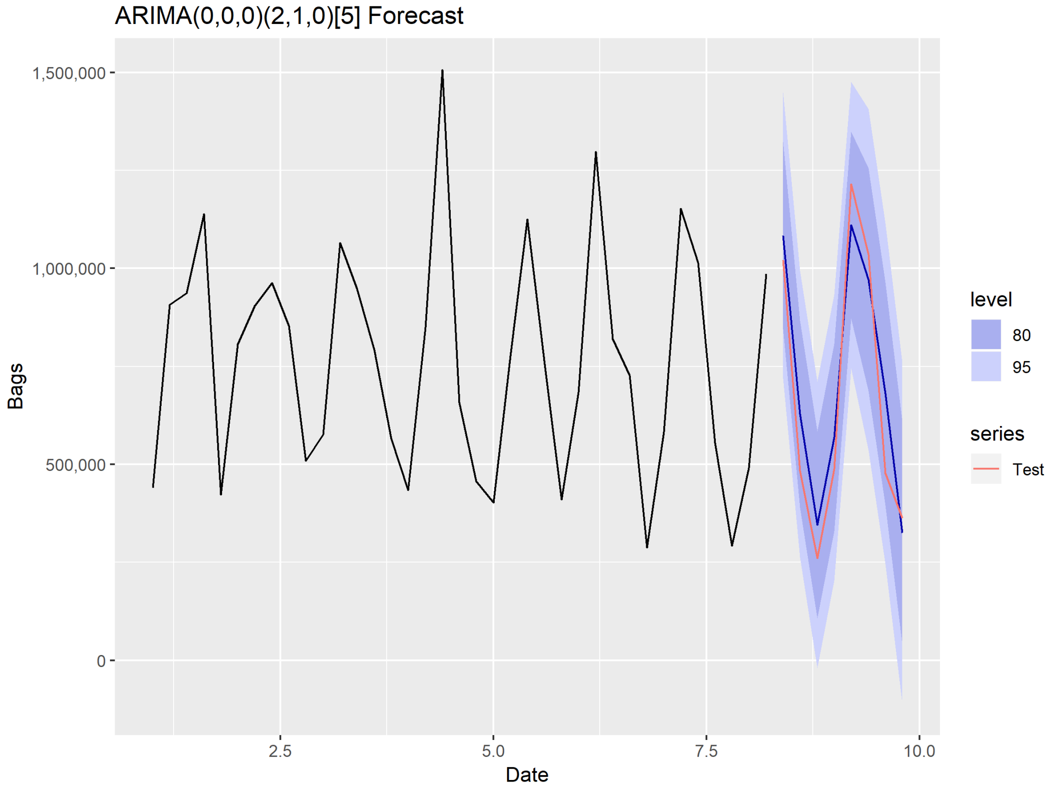


Figure 16: Harvest (June-October) Month Arima

The Arima model, especially for the harvest months, provides a compelling challenge due to its simplicity.

Challenger Model - Regression with ARIMA Errors

The next challenger model we built is a linear regression model with ARIMA errors. While the regression model allows for the inclusion of predictor variables, it does not allow for the subtle time series dynamics that can be handled by ARIMA models. The regression with ARIMA errors model solves this problem by fitting regression models with all the relevant variables first, and then applying ARIMA to the residuals of the regression to detect time series elements in the residuals.

We explored the correlations of twitter data and Google trend Scholle tomato bag sales and found the keywords and lags in Table 4 and Table 6 tend to strongly correlated with tomato bag sales. Since the regression with ARIMA errors is based on linear regression, we first build a linear regression model these keywords and lags. Results are shown in below Table _ . Significant variables are highlighted in green.

Based on the variables significant in the linear model, we built the regression model with ARIMA errors. The parameters and accuracy are shown below in Table and Table .

Keyword...Lag	Estimate	P.Value
Intercept	114883	0.00
bbq_twitter_lag17	-7771	0.48
pasta_twitter_lag18	7946	0.74
tomato_twitter_lag15	2645	0.80
Google bbq lag 10	-39516	0.14
Google chili lag 15	-42963	0.02
Google salsa lag 12	92606	0.00
Google tomato lag 12	122522	0.00

Based on the variables significant in the linear model, we built the regression model with ARIMA errors. The parameters and accuracy are shown below in Table and Table .

Keyword...Lag	Estimate	P.Value
Google bbq lag 10	-39516	0.137173
Google chili lag 15	-42963	0.017008
Google salsa lag 12	92606	0.000366
Google tomato lag 12	122522	0.000145

Positive coefficients imply that for a unit increase in the variable, there is a corresponding positive increase in the Scholle bag sales. Negative coefficients imply that for a unit increase in the variable, there is a corresponding negative decrease in the Scholle bag sales.The negative coefficient shows there is an inverse relationship between the variable and Scholle bag sales.

X	sMAPE	RMSE	X..Bias	Accuracy
Train	25%	93857.76	20%	0.38
Test	44%	68342.19	58%	-0.93

The above results indicate that Google trend data tend to have a greater influence in the predictions than twitter data, because the count in google searches is more direct in measuring the importance of the keywords and lags, compared to twitter data which might lose some information both due to the limitations in gathering tweet data and due to the complicated natural language preprocessing process. The ARIMA error is (1,0,0),(0,1,2)[12], indicating that the regression data was not sufficient to capture the time series elements of the data.

Challenger Model - Random Forest Regression

The second challenger model we built was a Random Forest Regression Model. Random Forest leverages many regression trees to build a consensus model in addition to bootstrap aggregation or bagging to generate additional augmented data. Bagging simply builds additional training sets by sampling with replacement from provided training data. Each individual tree uses the bagged training data and selects a random subset of features at each branching point rather than all features available to build the regression. This restriction forces the model to create a more robust estimator.

One useful feature when using tree-based approaches for regression is the ability to use categorical or ordinal predictors without the need for one-hot-encoding. In our model we represented the date as a pair of categorical variables, one for year and a second for month. We chose to do this because of the seasonal nature of the data.

In addition, we decided to challenge the model by only using lags that we thought to be have a reasonable explanation for their effect. To this end we chose to use lags greater than 9 months. Our logic was that it would take time for an increase or decrease in the social media presence of one of the keywords selected to go from an uptick in interest of consumer products to be captured by Scholle's bag sales.

The table below displays the results of our Random Forest model.

X	sMAPE	RMSE	X..Bias	Accuracy
Train	38%	66560.6	64%	2.94
Test	37%	61758.7	28%	5.54

An important feature of Random Forest modelling is its ability to generate rank-ordered summaries of variable importance. The model's feature importance is displayed in the figure below.

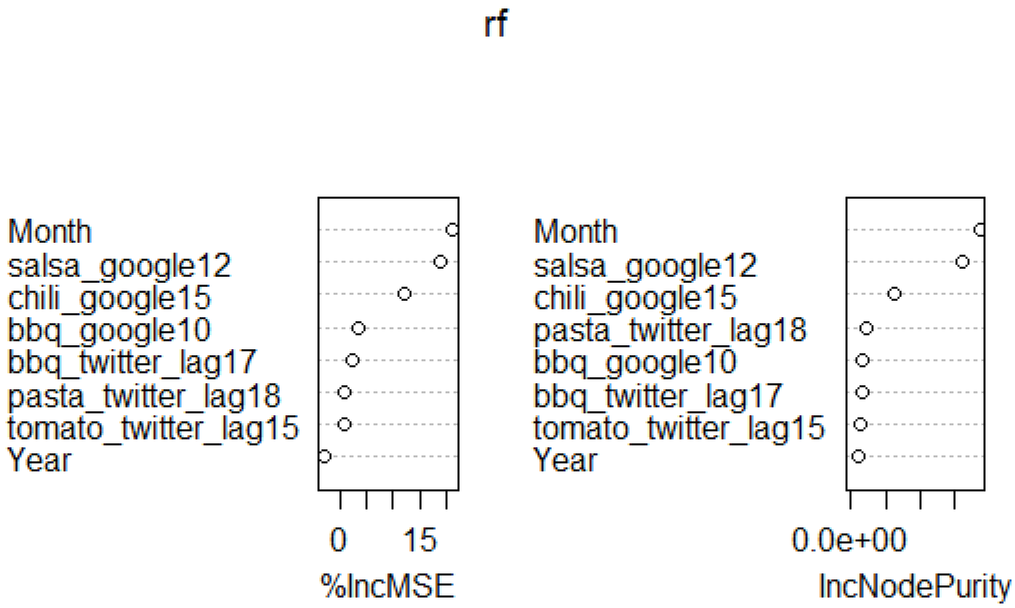


Figure 17: Random Forest Importance

The most important features are displayed at the top of this chart, in this the month was the most important predictor followed closely by google salsa data at lag 12. This agrees with the importance of monthly seasonality that we observed in the other models.

Challenger Model - XGBoost

XGBoost is an implementation of gradient boosted decision trees designed for speed and performance. It build trees one at a time, where each new tree helps to correct errors made by previously trained tree.

X	sMAPE	RMSE	X..Bias	Accuracy
Train	15%	55311.64	11%	0.52
Test	35%	88097.99	44%	2.00

Stacking Forecasts

Our project focuses on shipment quantity. There will inherently be some lag between the time someone has interaction on social media and its effect on the tomato bag shipments. Given that the vast majority of shipments occur during June through October, we limited the evaluation of model results to those months. These months coincide with when tomatoes become ripe in California, we thus refer to these months as the harvest months.

The stacking method is used to create an additional consensus model by using the results of trained models. We used two approaches to stacking, simple averaging and a linear model.

We then combined our forecast results from all models mentioned prior in this report, including the ARIMA model built only on the more stationary harvest months, the regression with ARIMA errors model using on strong correlated keywords and lags, random forest model, XGBoost model and Prophet model. We averaged the forecasts of all combinations of the models. We found that we were able to improve the model predictions by combining the Random Forest and Regression with Arima Error models predictions using a simple average.

The second approach to stacking we attempted was to build a linear model using the results of all our other models. This model was trained on all the harvest month data for 2014-2018. An important facet of a linear model is that it optimizes the weighting of each variable. Our case this is the model result. The fitted values produced a better estimate of the actual than the simple average of Random Forest and Regression with Arima Error.

	x
Intercept	-6428.4630100
Arima	0.7431722
Arima_reg	-0.1683120
RF	0.6933001
xgb	0.3950322
pro	-0.5076684

However the best fitting linear model does not consider if each of the inputs adds to the total information that is represented in each of the variables added. To address this, we applied the step function to find the model that produces the lowest AIC.

Overall the linear model blend performed the best for both sMAPE and RMSE leading us to choose it as our champion model.

key: * arima - * arima_reg - * xgb - * pro -

Model	sMAPE	RMSE	X..Bias	Accuracy
arima	12.76369	369493.5	0.48	1.0930270
arima_reg	11.59555	378343.2	0.52	1.1734880
RF	12.75160	427591.7	0.36	1.0892416
xgb	19.91717	611248.7	0.32	0.9974092
pro	16.30583	393135.5	0.56	1.1628570
tot_avg	12.82756	362912.5	0.44	1.1032046
arima+arima_reg	11.99662	366844.9	0.52	1.1332575
arima+RF	11.06201	339078.2	0.44	1.0911343
arima+xgb	14.08614	412555.8	0.36	1.0452181

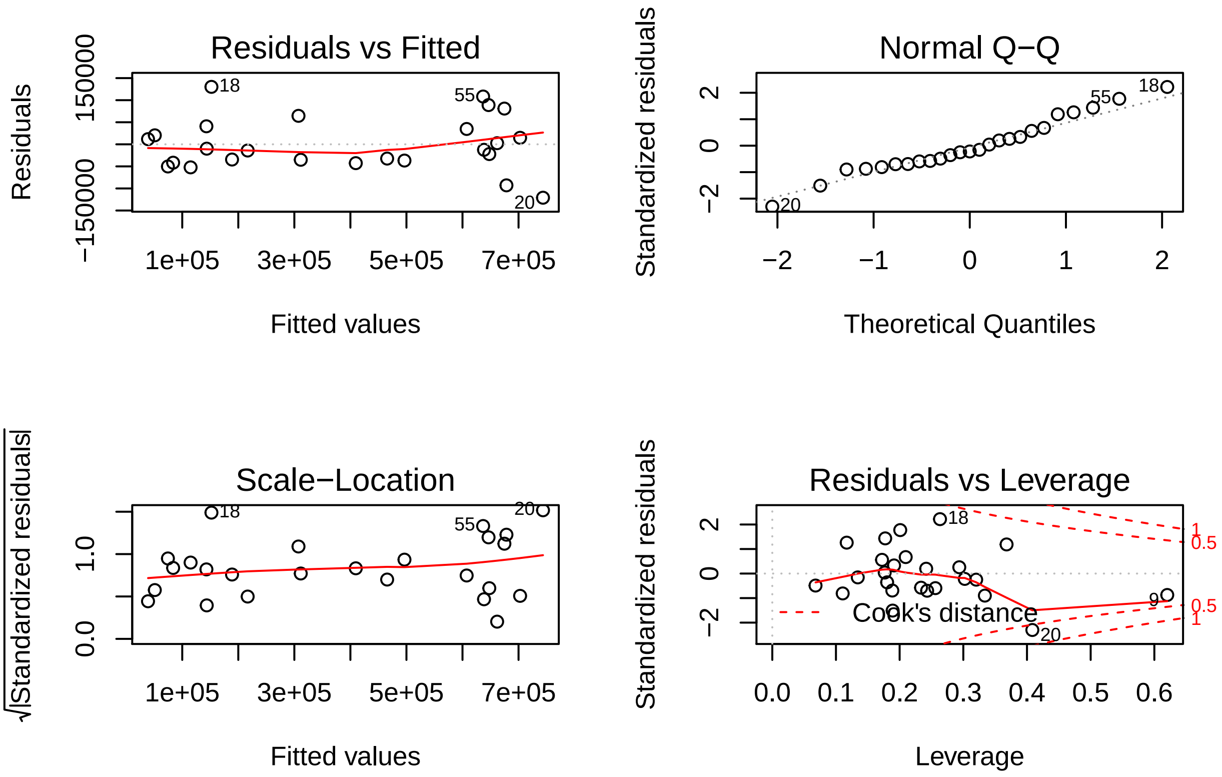
Model	sMAPE	RMSE	X..Bias	Accuracy
arima+pro	13.66767	345109.6	0.48	1.1279420
arima_reg+RF	11.02261	345499.1	0.48	1.1313648
arima_reg+xgb	13.47914	417879.0	0.36	1.0854486
arima_reg+pro	12.96459	348679.2	0.56	1.1681725
RF+xgb	15.06955	494505.8	0.36	1.0433254
RF+pro	13.95099	378280.1	0.52	1.1260493
xgb+pro	17.60323	473411.7	0.48	1.0801331
arima+arima_reg+RF	11.33500	335521.4	0.52	1.1185856
arima+arima_reg+xgb	12.76800	374804.7	0.52	1.0879748
arima+arima_reg+pro	12.82931	344284.7	0.52	1.1431240
arima_reg+RF+xgb	12.61945	402010.3	0.40	1.0867130
arima_reg+RF+pro	12.19553	342480.6	0.52	1.1418622
RF+xgb+pro	12.61945	402010.3	0.40	1.0867130
Linear Model Blend	10.95078	297958.3	0.56	1.1296389

Residual Analysis

```r
#plot each of the forecasts & the blend & actual
```

Residual analysis is an important final step because it helps us understand if our model has any systemic biases that we should be aware of going forward.

```r
lm_blender <- readRDS('./data/LM_Blender.rds')
par(mfrow=c(2,2))
plot(lm_blender)
```



```
par(mfrow=c(1,1))
```

```
facts = c(mean = mean(lm_blender$residuals),
          median = median(lm_blender$residuals),
          variance = var(lm_blender$residuals),
          skewness = e1071::skewness(lm_blender$residuals),
          kurtosis = e1071::kurtosis(lm_blender$residuals))

kableExtra::kable(facts, digit = 2, align = "r", caption = "Model Blend Summary",
      format = "markdown", longtable = FALSE)
```

	x
mean	0.00
median	-12461.53
variance	3699131576.47
skewness	0.34
kurtosis	-0.47

Breusch-Pagan test for heteroscedasticity

```
	studentized Breusch-Pagan test

data:  lm_blender
BP = 10.402, df = 5, p-value = 0.06462
```

NCV test for heteroscedasticity

Both the Breusch-Pagan and NCV tests detect heteroscedasticity.

Conclusion

We were able to use social media data to better forecast tomato bag shipment data beyond the ability Scholle's existing process of only incorporating its prior year data.

Recommendations

We recommend continuing collecting social media data and incorporating the results into the overall ensemble model for harvest month data.

Appendix A

The First Appendix

This first appendix includes all of the R chunks of code that were hidden throughout the document (using the `include = FALSE` chunk tag) to help with readibility and/or setup.

In section ??:

In section ??:

Appendix B

A Second Appendix, for example

References

There are a variety of tools available for creating a bibliography database (stored with the .bib extension). In addition to BibTeX suggested below, you may want to consider using the free and easy-to-use tool called Zotero.

R Markdown uses *pandoc* (`http://pandoc.org/`) to build its bibliographies. To cite references in your thesis (after creating your bibliography database), place the reference name inside square brackets and precede it by the "at" symbol. For example, here's a reference to a book about worrying: (Molina & Borkovec, 1994). This `Molina1994` entry appears in a file called `thesis.bib` in the `bib` folder. This bibliography database file was created by a program called BibTeX. You can call this file something else if you like (look at the YAML header in the main .Rmd file) and, by default, is to placed in the `bib` folder.

Additional Tips

- The sooner you start compiling your bibliography for something as large as a capstone, the better. Typing in source after source is mind-numbing enough; do you really want to do it for hours on end at the last minute?
- The cite key (a citation's label) needs to be unique from the other entries.
- When you have more than one author or editor, you need to separate each author's name by the word "and" e.g. `Author = {Noble, Sam and Youngberg, Jessica},`

Example output generated from bib file

Angel, E. (2000). *Interactive computer graphics : A top-down approach with opengl*. Boston, MA: Addison Wesley Longman.

Angel, E. (2001a). *Batch-file computer graphics : A bottom-up approach with quicktime*. Boston, MA: Wesley Addison Longman.

Angel, E. (2001b). *Test second book by angel*. Boston, MA: Wesley Addison Longman.

Molina, S. T., & Borkovec, T. D. (1994). The Penn State worry questionnaire: Psychometric properties and associated characteristics. In G. C. L. Davey & F. Tallis (Eds.), *Worrying: Perspectives on theory, assessment and treatment* (pp. 265–283). New York: Wiley.