## Problem Statement:

Time-series analyses are a common tool among data scientists. They provide insight into underlying trends within the business. They also allow the business to create forecasts to plan for future needs such as inventory. This project aims to build off these concepts by forecasting for various segments of the business. The online retailer I am working with currently does not differentiate between the ~4000 items sold. Using natural language processing (NLP) techniques, the business might be able to segment their business meaningfully and use that information to provide more fine-grained business forecasts.

## Goal:

The goal of this project is two-fold. The first goal is to explore possible business segments. By applying NLP techniques to the item descriptions and then feeding them into clustering models, I hope to discover meaningful ways to break down the business into constituent parts. The second goal of the project is to use the insights from part one to create a four-week forecast for the business to prepare for the upcoming holiday season. The data contains approximately one year of data and stops just short of the holiday season. By creating time-series analyses for each of the business segments created in the prior section, I aim to provide information that will allow the business to adequately stock and prepare for the holiday season.

## Data Wrangling:

The greatest challenge of getting this data ready to be processed was figuring out what to do with cancelled orders. The refunds were designated by an Invoice Number that started with 'C'. While these were easy to isolate and remove from the data frame, there was no single marker that tied them to the original order that ended up being cancelled. There were 9,288 cancelled orders. This meant that between the cancellations themselves – i.e. the invoices starting with 'C' – and their original counterparts, I was looking to take out 18,576 of the approximately 514,00 total orders.

To try to identify the original order, I created an "identifier". This identifier combined all the columns that would match between both a cancellation and its corresponding order. I multiplied UnitPrice, Quantity, and CustomerID, then multiplied that number by -1 to match it with the negative value created by the refund in the cancelled order. I then concatenated this value with the StockCode. I created a list of all the identifiers from the cancelled orders and subset the data frame down to only rows that contained an identifier in that list.

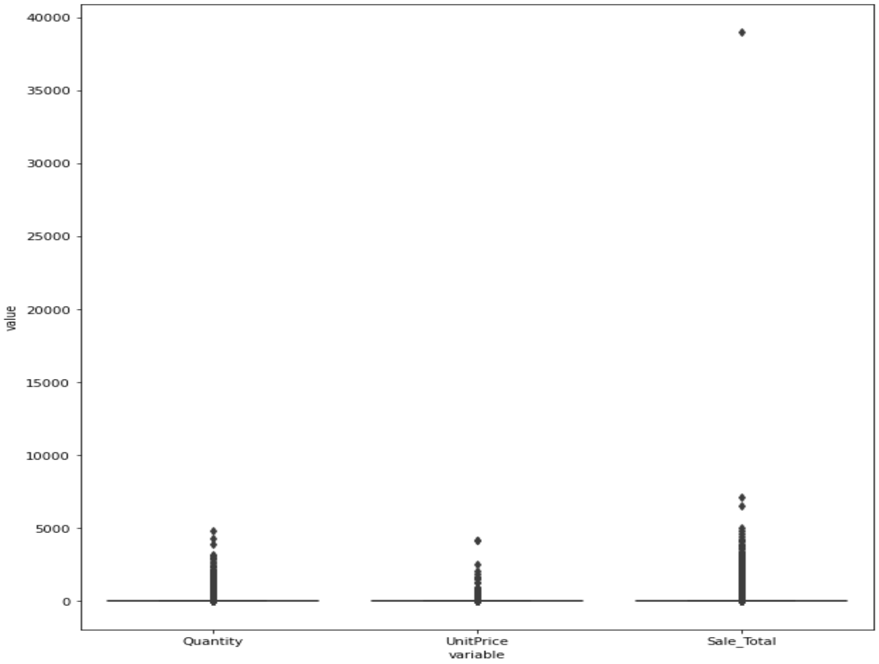
This subset contained 18,239 sales (The cancellations themselves had been removed, so I was hoping to find close to 9,000 sales). A customer could place the same order five times and only cancel one of them, so finding extra matches was not unexpected. When I narrowed down this subset to include only unique values, it contained 6,632 sales. When this still left a disparity of a couple thousand sales, it was still pretty good. I later used box plots to look for outliers and ensure that I did not miss any huge orders that were later cancelled. As I noted, this does put limitations on the accuracy of any forecasting model I later created. However, in the absence of a clear way to identify and isolate sales that were later cancelled, getting down to 2,500 unwanted records out of 514,000 was not a terrible outcome.

Finally, I did some additional clean up, such as excluding UnitPrice and Quantity values that were less than or equal to zero. The data was then in a state that was ready to be manipulated and modeled with.

## EDA:

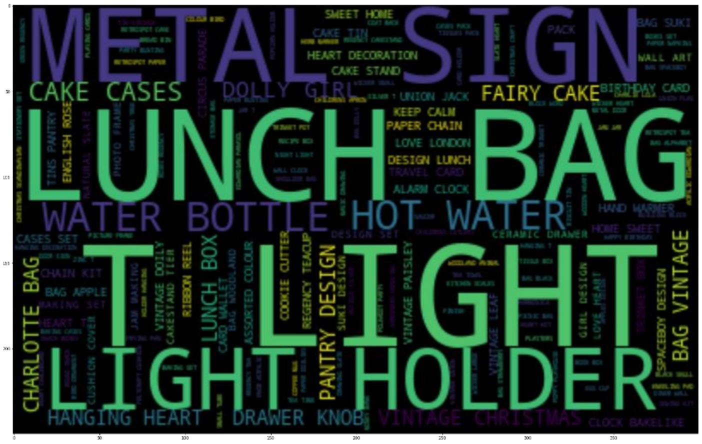
In order to explore the data, I performed three main tasks. The first was to create a box plot of the 'UnitPrice' and 'Quantity' columns. This revealed that there were many values at or below zero in both columns. Upon closer examination, the descriptions of these items were either manual adjustments or comments such as missing, smashed, or damaged. While these are important records for the business to have, like the cancelled orders, they weren't explicitly tied to particular sales. For forecasting purposes, therefore, I excluded them.

After filtering non-positive values, I created a column representing the sales total by multiplying the unit price by the quantity. I then once again created a box plot, this time with all three columns.

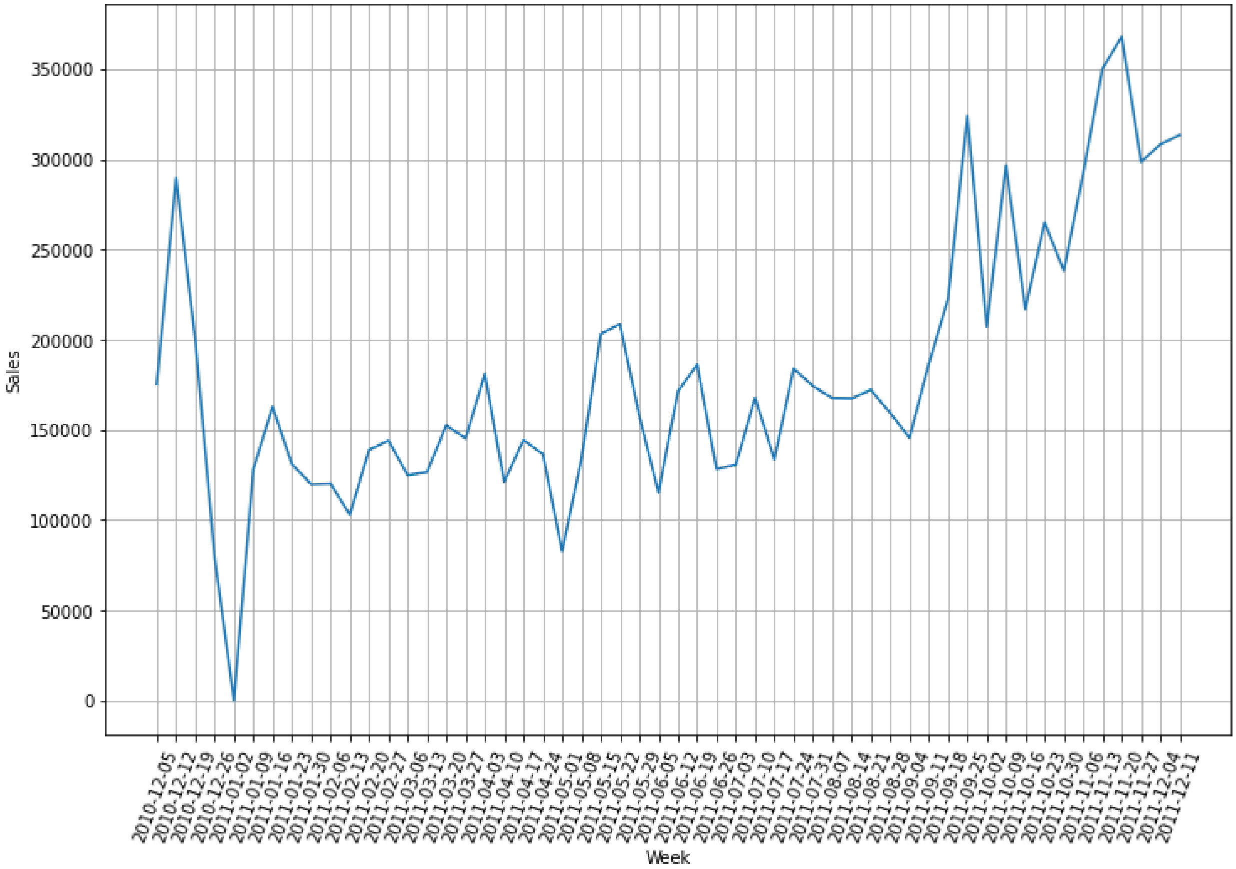


As we can see above, there is one value in the 'Sale_Total' column that is an outlier. This turned out to be a cancelled order that was manually adjusted in the sales records. I removed this value and examined the next two highest values to ensure they were not also cancelled orders that had slipped by my initial filtering.

The second step in my EDA was to start to get an idea of the descriptions in the sales data. I first created a Word Cloud to visualize what words were showing up most frequently in the descriptions. This was useful in getting a quick, initial idea of the business as a place that sells primarily home décor and crafting materials. However, it also alerted me to the potential difficulty in creating business segments, since commonly occurring words were things like "fairy cake", "charlotte bag", and "t light". The word cloud can be viewed below.

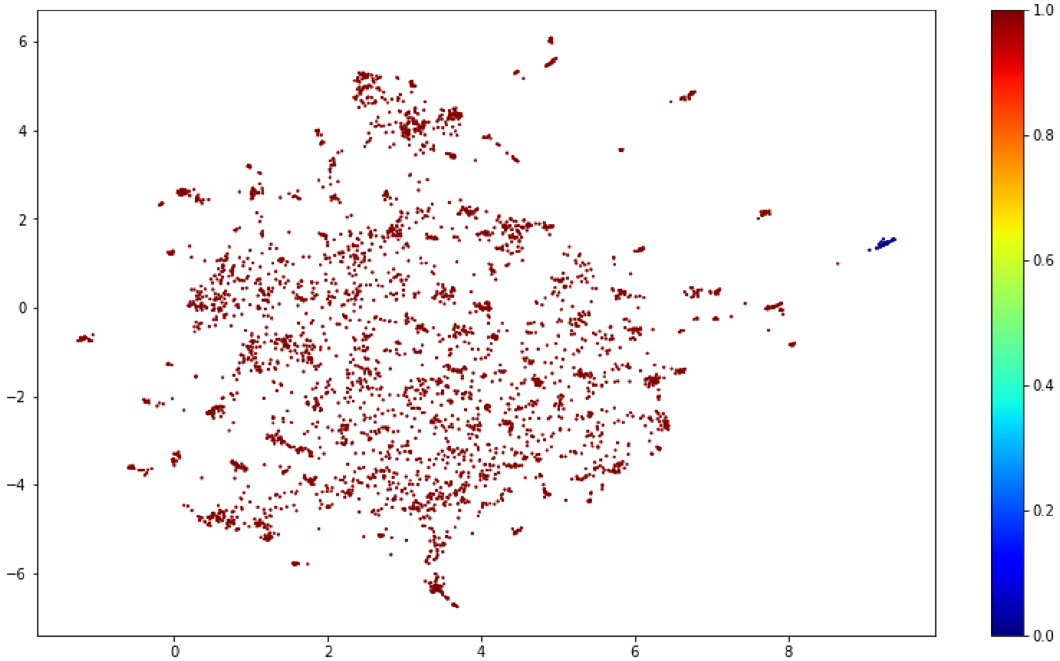


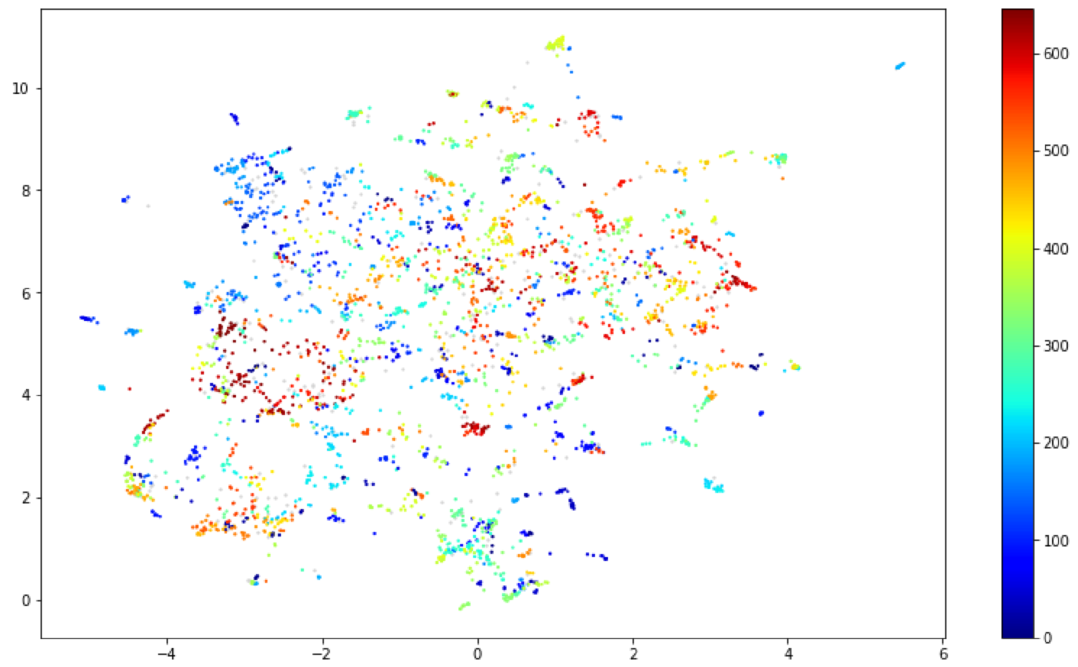
Finally, I created line plots of the monthly and weekly sales. I originally thought the data covered almost two years. I realized however, that this was only a single year of data. This allowed me to shift my focus from a monthly forecast to a forecast that makes a prediction about sales in the weeks leading up to the Christmas holiday. The weekly sales line plot can be seen below.



## Modeling – Part I

For the first step in my modeling process, I wanted to use unsupervised clustering to attempt to segment the business. The least effective method was using UMAP to reduce the dimensionality of the data and then using HDBSCAN. Given how the data was represented by UMAP, there were essentially no natural clusters. The data was more or less evenly and randomly distributed. As a result, when I ran HDBSCAN with multiple different sentence embedding models, it gave me one of two results. Either three clusters were created with almost all the data points in one of them and only a few outliers in the other two, or hundreds of groups were created. An instance of each can be seen below.





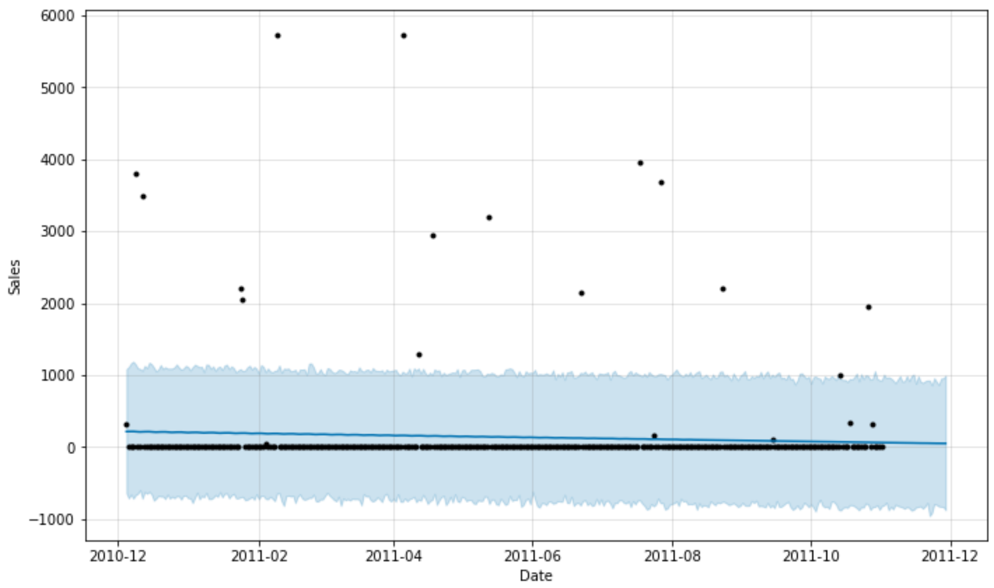
I also attempted clustering using a gensim and a latent dirichlet allocation(LDA) model. After clustering with 3 and 8 topics with gensim and 5 topics with LDA, it became clear that the items sold by the company cannot be cleanly segmented into meaningful groups. The best that could be done was by fine-tuning a 5 segment LDA model. Five segments was in the sweet spot in terms of finding various potential aspects of the business, while not breaking it into so many sections that it would be cumbersome to model and monitor all of them. While the nature of the data meant that grouping was inherently going to be blurry, after tuning the batch size and maximum iterations the groupings made some sense (though a little imagination and leeway is needed).

## Modeling Part II – Forecasting

The second modeling goal was to create a four-week forecast for the business. My initial intention had been to utilize the segments identified by the clustering model to give an analysis of how various aspects of the business were doing. Given the challenges I faced dividing the business into meaningful segments, I first decided to see if I could forecast sales for various regions within the business.

The original data contained countries so in an attempt to bin the data into a few larger groups, I created a data frame with only the countries and their respective regions and then merged the two data frames. However, when I created to forecast with Asia and the South Pacific, the region with the third most sales(out of five regions I intended on using), it became clear that domestic sales accounted for too great a percentage of total sales.

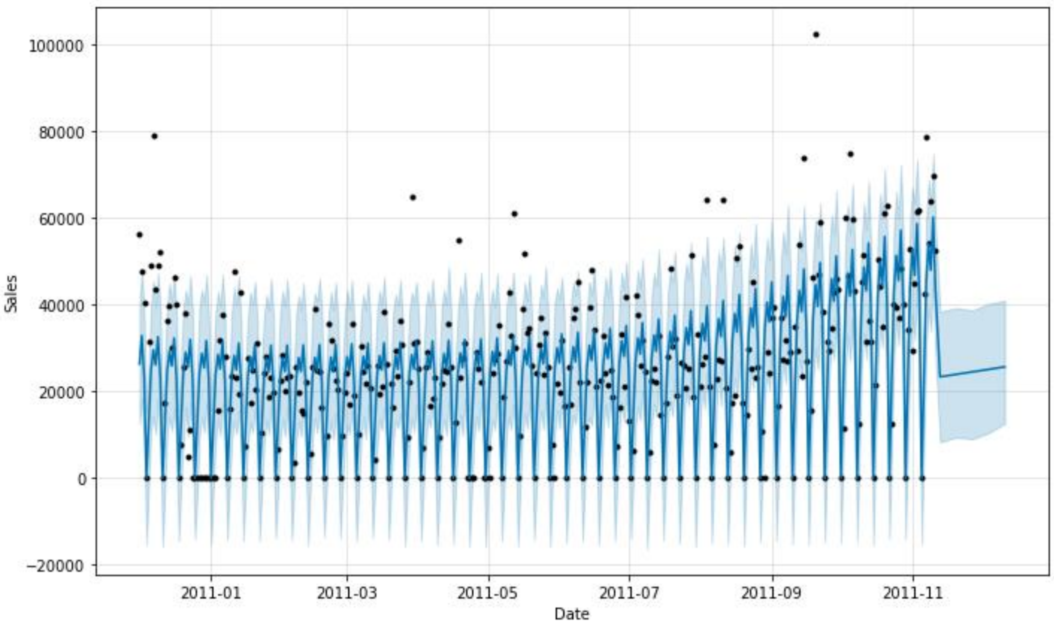
### Sales for the Asia/South Pacific



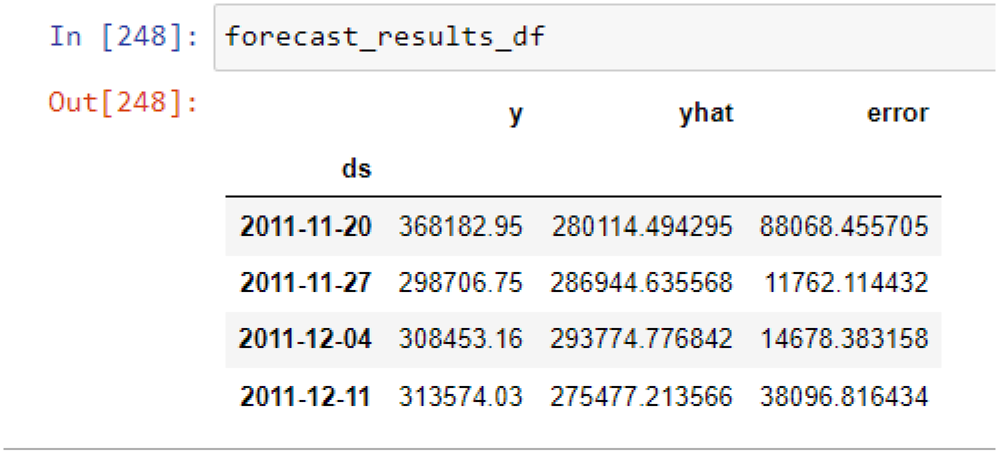
A vast majority of days saw no sales in this region. Because some weeks had zero sales, attempting to calculate mean absolute percentage error (MAPE) threw an error because of zeroes in the denominator. I modeled with only the domestic data as well to see if foreign sales were accounting for the noise in the data. However, this returned a greater MAPE than modeling with all the data.

In my final model I utilized all the data to minimize error. My business initiative was to give a four-week forecast that would aid in ordering and staffing through the holiday season. To test model accuracy, I removed the last four weeks of data from the training set. This allowed me to compare four weeks of data generated by the model against actual sales.

### Daily Modeling of Full Dataset with Weekly Average Predictions



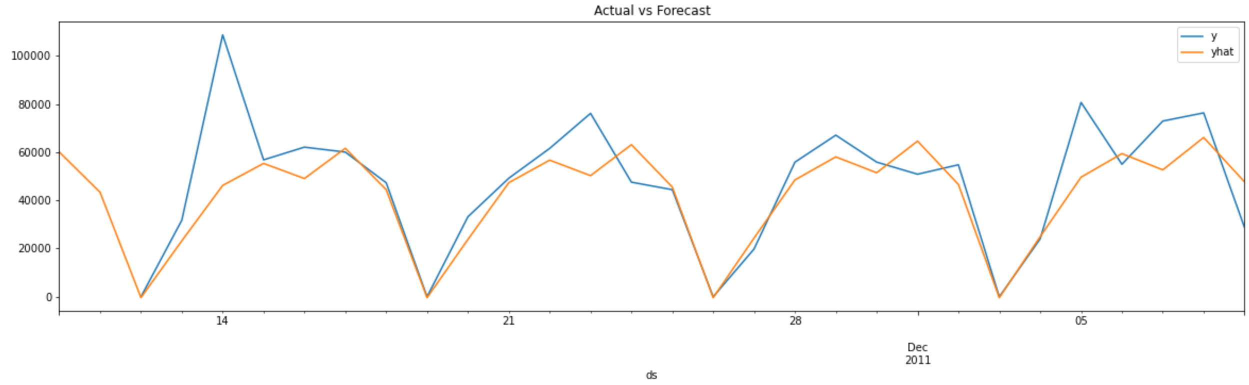
Below is a data frame containing the predicted values, actual vales, and the error. Since my goal was to create a weekly forecast, I disregarded daily error in favor of aggregating the data and calculating it on a weekly basis.



```
In [248]: forecast_results_df
```

Out[248]:

| ds | y | yhat | error |
|---|---|---|---|
| 2011-11-20 | 368182.95 | 280114.494295 | 88068.455705 |
| 2011-11-27 | 298706.75 | 286944.635568 | 11762.114432 |
| 2011-12-04 | 308453.16 | 293774.776842 | 14678.383158 |
| 2011-12-11 | 313574.03 | 275477.213566 | 38096.816434 |



We can see from both the table and the graph that our model consistently underpredicted, with a majority of the error coming in the first week as a result of one unusually high sales day.

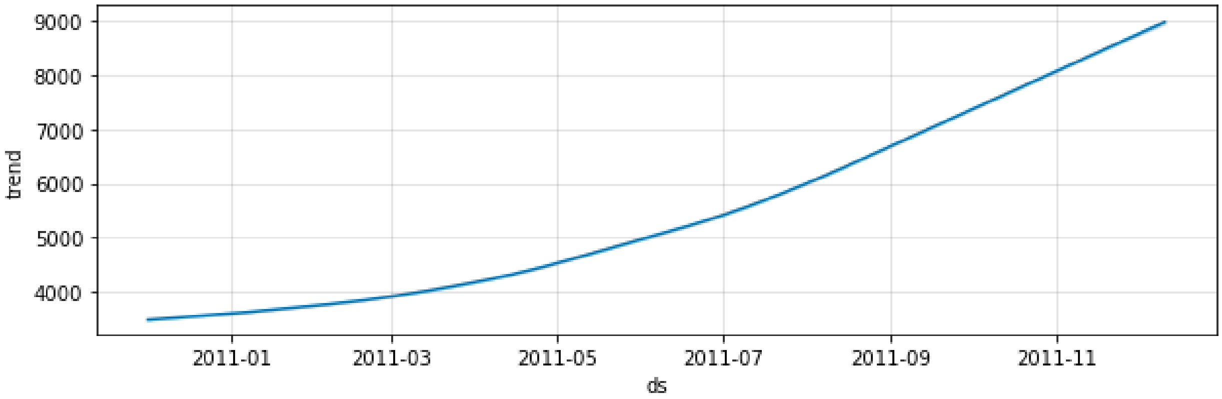
The forecast had a mean absolute percentage error of 11.2% and a root mean square error of 48,891.

## Conclusion

Ultimately, I would have to work with decision makers in the company to decide whether this model is accurate enough to make it worth using. It may be worth adding 10% to the forecasted numbers and then preparing for weekly sales with some additional buffer. I would also look to gain more information on the day with nearly double the normal sales total. Was this a surprise or had the company been working on a big sales deal? Are there any other inputs we might be able to add to the model that would allow us to predict unusually high sales day. If not, because the model consistently underpredicted, there could be legitimate concern that the model is simply not identifying a trend and the predictions could get less accurate week over week.

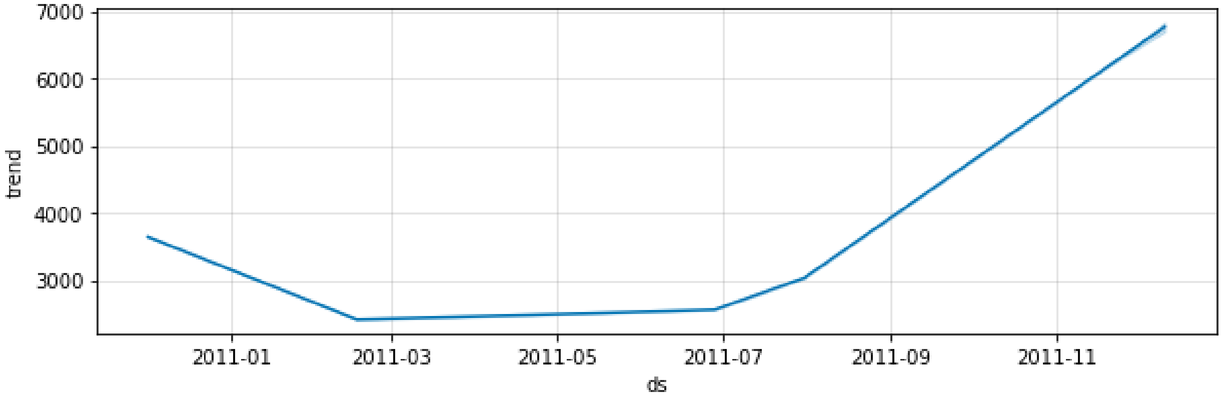
In addition, while the segments I created using the clustering model did not prove useful, by looking at the trend lines created by forecasting Segment_0 vs that of Segment_1, the underlying potential of the exercise becomes clear.

**Segment 0 Trendline**



Above we see Segment_0's trendline starts below 4,000 and increases throughout the year. In contrast, below we see that Segment_1’s trendline starts at nearly 40,000 and continues to decrease until around March when it begins its upward trend.

**Segment 1 Trendline**



If we imagine that the segmentation model had identified groupings that could be made sense of, it is easy to imagine gaining quick, meaningful insight into how the various segments of the business have performed compared to one another throughout the year. Additionally, the business may be able to finetune their ordering and staffing according. Some trends might be obvious, like selling Christmas products in December, but it is also possible that some unforeseen trends could be uncovered that could lead to the development of new sales and marketing efforts.