CSE 519 Data Science Fundamentals - Project Final Report
Retail Store Data Analysis

Abstract *- The purpose of this project is to analyze the retail store, Costello's hardware data, find the reason for the drop in sales and come up with solutions with evaluation, to increase future sales. In this report, we have focused on two of the most crucial business issues - Shelf Space Management and Out of Stock of products. The products arranged in the Shelf Space is crucial as it is where shoppers make impulse purchases and a good customer experience would urge them to purchase more. To handle the Out of stock problem, we have tried a novel approach by focusing on scenarios when customers don't visit their nearest Costello's store, and visit stores which are far from that of Costello's chain.*

1. INTRODUCTION

The volume, variety, and velocity of data being produced in all areas of the retail industry is growing exponentially, creating both challenges and opportunities for those diligently analyzing this data to gain a competitive advantage. We leave information about our location and also reveal hints on our lifestyles to the supermarkets through every transaction. The abundance of data encourages companies to use it to create a competitive advantage. Although retailers have been using data analytics to generate business intelligence for years, the extreme composition of today's data necessitates new approaches and tools that can be used to create amazing shopping experiences and forge tighter connections between customers, brands, and retailers.

The objectives of this research analysis report is to find appropriate suggestions to the most commonly faced and concerning questions faced by retailers and what is the best course of actions that can be taken to attain profitability in the retail sector and how to sustain in the retail business in this digital age. The most common business issues are shown in the table. Though we have given insights for most of the mentioned business issues, in this report, we have worked in detail in analysing and providing solutions for the top two business issues - Shelf Space Management and Out of Stock.

Business issues	Potential benefits from business analytics
Shelf Space Management	what products to be placed nearer in stores.
Out of stock	Improve stock levels to never run out or tie up unnecessary capital.
Planning new stores and their locations	Match the store design with surrounding preferences from its given location to improve profit.
Price optimization	Increase profit by charging the exact price a customer is willing to pay for a product.
Forecasting	Anticipate customers' demands for the future and plan ahead to increase profit.

Shelf Space Management - The shelf is a dynamic environment. It's where shoppers make both planned and impulse purchases, and where store associates stock shelves to satisfy in-store shoppers and also pick products to support their online channel. But without the consistent execution of optimized planograms at the store level that meet customer demand and expectations, even the best category management strategy can underperform in actual execution.

Out of Stock - Stockouts generally refer to a product being unavailable for purchase at retail, as opposed to elsewhere in the supply chain. They're most apparent in the fast-moving consumer goods sector. A customer walks into a local store looking for a favorite brand of a product, only to find that it's out of stock. Frustrated, this patron can consider other options: choose another brand, choose another store, choose to postpone the purchase, or choose not to buy at all. None of these scenarios are favorable to the producer. Sales, brand image, and future planning efforts are all damaged as a result of out-of-stocks.

2. RELATED WORK

Specific to the retail sector, McKinsey estimates the use of big data to increase margins by up to 60%, highlighting the importance and the relevance of big data applications. In May 2013, Rangespan, a company using big data analysis to help retailers make decisions on the ranges to carry, was taken over by Google and stopped providing services to other retailers; now has become an in-house service for Google Shopping to expand its offerings. Two months later, Boomerang Commerce, a company providing dynamic pricing services to retailers based on big data analysis, raised $8.5 million. By the time this study was carried out, most research on big data in retail has focused on how to obtain greater consumer insights to implement marketing activities more effectively. Although precise marketing can generate demand more effectively, without suitable operational support, these demands may not translate into sales. A related work[1] showcases experiments on sequential transaction dataset of a Pharma retail industry by finding complementarities and similarities between products. It learns the representation of products in low dimensional space similar to the one used in Word2Vec and make suggestions by analyzing the transactions and all context products available in the same invoices. It has also worked with another approach[2] where the product to product co-occurrence score is calculated using ratio of co-occurrence probabilities of two products to figure out the similarities of complimentarity baskets and these computed embeddings i.e. Product to Product co-occurrence score are used to predict sales per year efficiently.

3. DATASETS

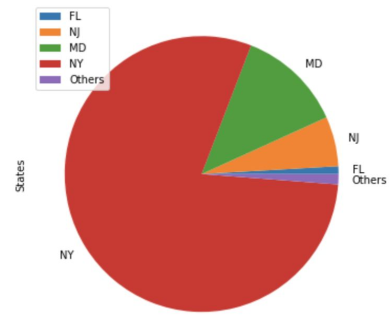
3.1 Costello Ace Data

The Costello Ace Hardware data consists of 17M records of transactions performed at 31 stores of the retail chain with 39 features (information about the item, customer, transaction, and store). The dataset contains separate entry for each of the items bought by a particular customer, because of that dataset has no unique index values. However, we have added a new features, *timestamp*, combining the date and transaction time. This allows to access each checkout and/or any other activity like exchange, return, defective individually.
Features : Item Number, Customer Number, Gross Margin%, Net Unit Sales, Store #, Timestamp

3.2 External Datasets

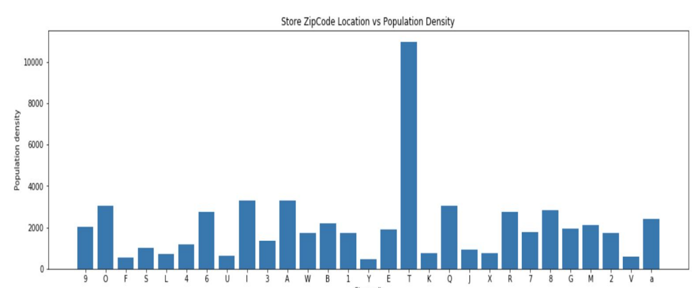
Location data
The US_Zip_Code_Dataset[1] is specific to zip codes, gives information about the Latitudes and Longitude details, geopoint, city and state of a particular zip code. We used this location data to (i) understand how many customers are located far away from the stores (ii) find the crow distance between two stores or a store and a customer location by using the latitude and longitude details of the zipcode.
Features :Latitudes, Longitudes, Zip Code

Fig 3.2.1 Percentage of customers across states of US



Demographic Data
The US_Demographic_Dataset[2] based on zip code has information about the population density, total population, city and state details specific to various zip codes across the United States of America. This data was used to predict for more openings of stores in those regions with high preference and population density.
Features :Population Density, total population, city, state

Fig 3.2.2 Population density around stores locations



Weather Data
We have used the **Weather_Dataset**[3] data to check how weather impacts the sales of the company**.** There has been a significant reduction in the Gross Margin% of sales and number of transactions during the *winter months of December-January*. *Features: Temperature*

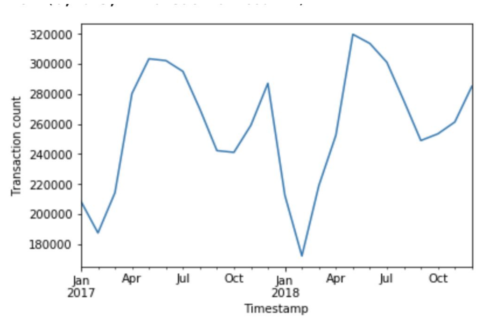


Fig3.2.3Transaction count across 2017-2018

Google Maps API

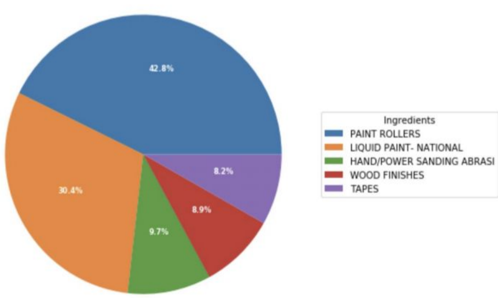
We have found the *crow_distance* using haversine function which unlike euclidean distance, considers Earth's radius while computing distance between two geographic location. We tried to find the *land_transport_distance* by using the Google_Maps_API[4].

4. EXPLORATORY DATA ANALYSIS

EDA helped us to come up with two solid hypotheses specific to our objective.

4.1 Hypotheses 1 (Shelf Space Management)

'Place similar products together in the stores so that in case of out of stock of the products, the customer can pick the closely related products without having to search for it. Products that are frequently bought together should also be placed together to improve the customer experience.'

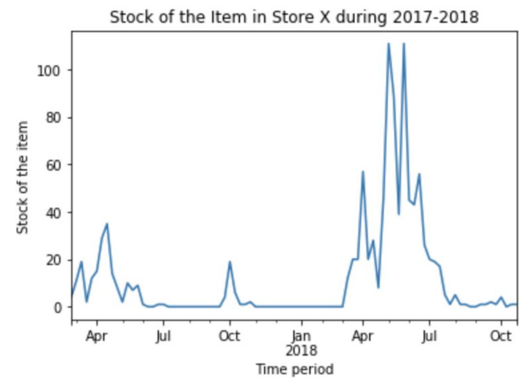


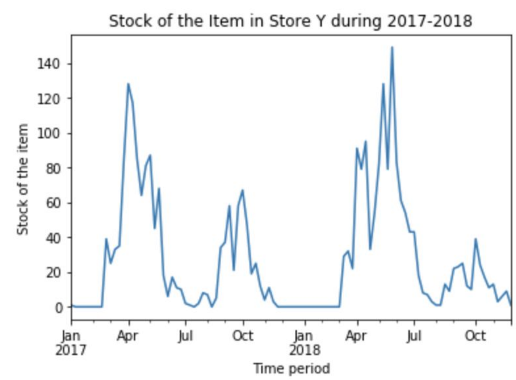
From above Fig, we can notice that if one of the PAINT ROLLERS is out of stock, then the customer can actually pick from the other paint rollers available in the store, if they are placed nearer. From Fig 5b, we can notice that PAINT ROLLERS are bought together with Liquid Paints, Tapes etc and shows a high correlation between them.

Sno	Item Number	Item Description
1	37009048	PLASTIC TRAY LINER
2	57140392	ROLLER 9X3/8 3PK PNTRS CHOICE
3	1818087	PAINT TRAY LINER 11W
4	37033003	PAINT-FORCE 3 WIRE ROLLR FRAME
5	1807312	ROLLER CVR LNTLS 1/2X7

4.2 Hypothesis 2 (Out of Stock)

'If a product is out of stock in stores nearby to a customer, then the customer will more likely prefer to postpone the purchase or even buy from another store.'





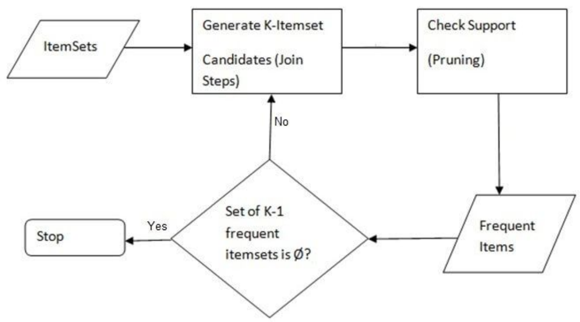
Above are the graphs stating the stock of a random product in *Store X* and *Store Y*. In *Store X*, the product is stocked very less frequently, whereas in Store Y, the product is stocked at regular intervals.

5. MODEL AND IMPLEMENTATION

5.1. Shelf space management

5.1.1 Apriori (Baseline Model)

Apriori algorithm is the way to find frequent itemsets. The Apriori algorithm needs a minimum support level as an input and a data set. The algorithm will generate a list of all candidate itemsets with one item. The transaction data set will then be scanned to see which sets meet the minimum support level. Sets that don't meet the minimum support level will get tossed out. The remaining sets will then be combined to make itemsets with two elements. Again, the transaction dataset will be scanned and itemsets not meeting the minimum support level will get tossed. This procedure will be repeated until all sets are tossed out.



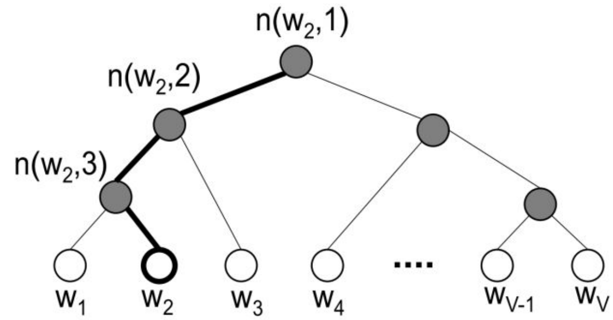
Apriori Algorithm Pseudo-code

Ck: Candidate itemset of size k
Lk : frequent itemset of size k
L1 = {frequent items};
for (k = 1; Lk !=; k++) do begin
Ck+1 = candidates generated from Lk; for each transaction t do
increment the count of all candidates in Ck+1 that are contained in t
Lk+1 = candidates in Ck+1 with min_support
end
return k Lk;

Our job is to identify the set of products that are brought together by the costello ace customers. In order to do this, we first grouped the list of items that are brought together, based on the invoice and store id. Then list of list of products that are bought together is extracted and fed as the input for the above apriori algorithm. Apriori algorithm will then generate the list of association rules that tells which product is more likely bought by the customer in combination with the other product.

5.1.2 Word to Vec (Basic Model)

Word to vec is a shallow neural network algorithm for measuring the quality of the resulting vector representations, with the expectation that not only will similar words tend to be close to each other, but that words can have multiple degrees of similarity. It is capable of capturing context of a word in a document, semantic and syntactic similarity, relation with other words, etc. The method is an unsupervised one, in the sense that it relies only on natural language corpora and doesn't require labeled data.



Word Representation by Word to Vec Model

We have used word to vec model to find the most similar items in terms of various features from Castello Ace Hardware Dataset like Class, Department, Actual Price and how frequent the products are brought together. We have trained our Word2Vec model with a vocabulary of 147066 words. The results from the word to vec model is used to place the similar products on the shelf within a departments.

Drawbacks: This basic model can be trained only upto class level and hence does not work across departments.

5.1.3 Word2Vec (Advanced Implementation)

In the given scenario Word2Vec[9] would come in handy since it's most concerned with words that come together in the same context then we can use it to find items that are usually bought together or items that are similar to each other. To do this we interpreted every purchases of a single customer as a sentence and every item in his purchase as a word.

The inputs here are one hot encoded vectors and output layer gives probability of being nearby word(item in our case) for every word (item) in the vocabulary.

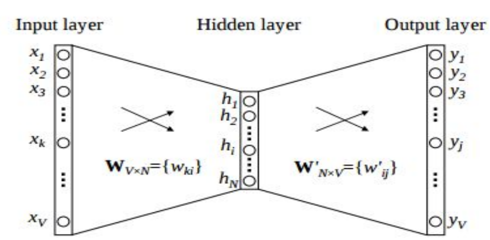


Fig : Word2Vec Model Architecture

After training the model, learned weight matrix **WV x N** is extracted, using that word vectors are extracted.

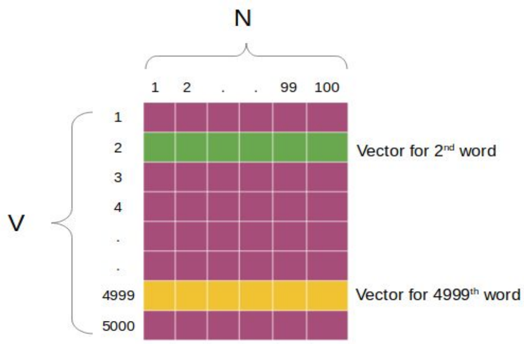


Fig : Illustration of word vectors

Here a visualization is shown using UMAP algorithm for dimensionality reduction of item embeddings to 2 dimensions.

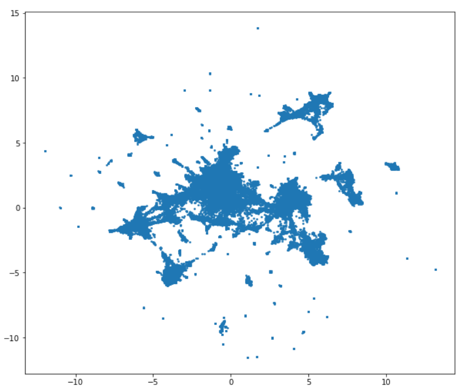


Fig : Item Embeddings in 2 Dimensions

Every dot here represents a product and together they create clusters - groups of similar products.

While using Word2Vec model trained on full dataset, there is a slight possibility that model will be biased for few items. Suppose a single customer has purchased a few specific items together in a bulk at a time or repeatedly which can behave as an outlier and disturb the performance of trained model. To verify this occurrence, we had to explore the data deeply to find such transactions and eliminate. However, in such a huge dataset with complicated features, we were not able to fully validate this situation while exploring. One more idea was to divide the dataset in few parts and compare the results to check the effect of outliers. The dataset of 2 years was broken down into 4 parts based on weather in the United States by months i.e. December to February - Winter, March to May - Spring, and etc.[10] With Word2Vec models trained on these 4 different datasets, we compared the results of few given items and got 8 out of every 10 results same across datasets by weathers with high confidence. From this, we were able to validate our model that provided accurate suggestions in most of the cases.

5.2. Out of Stock

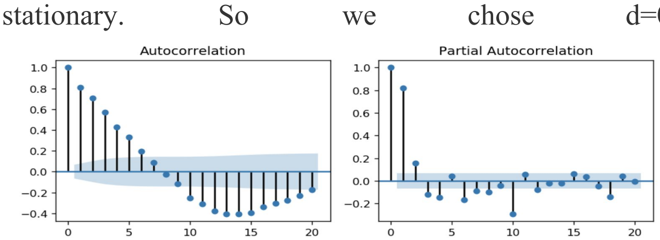
5.2.1 Arima Model

ARIMA (Auto Regressive integrated Moving Average) is one of the most popular models that can be used to forecast future values from the given time series data with the help of its own lags and lagged forecast errors in the past.

Forecasting in general is really tough. In practice, advanced models do well on in-sample forecasts but not so great out in the wild, as compared to simpler models. ARIMA models occupy that middle-range area of being simple enough to not overfit while being flexible enough to capture some of the types of relationships in the data. So, we chose ARIMA for predicting the future sales on a weekly basis.

ARIMA model has three tuning parameters P, d, and Q that can be tuned for better results. Since ARIMA is one of the linear regression model, it works fine when the series is stationary. Time series data can be made stationary by differencing previous values from the current values. Depending on the data there can be more than one differencing. The value of d represents the minimum number of differencing needed to make the time series data stationery. 'P' is the order of 'Auto Regressive' term, refers to the number of lags of Y to be used as predictors. 'Q' is the order of the 'Moving Average' (MA) term. It refers to the number of lagged forecast errors that should go into the ARIMA Model.

Augmented Dickey Fuller test (adfuller()), from the statsmodels package, confirmed that the series is stationary. So we chose d=0.



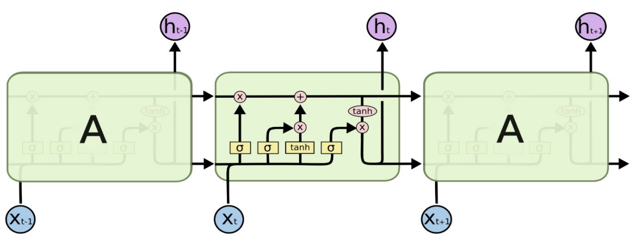
We used AutoCorrelation (ACF), & Partial

Autocorrelation Function (PACF) graphs to find the values of P and Q for Arima. We need to check, for which value in x-axis, graph line drops to 0 in y-axis for 1st time. Thus, we get P = 19 and Q=8.

From the given dataset, we have extracted only the sales data (excluding returns) and aggregated the amount of sales on a weekly basis. We have sales data for almost 104 weeks. Historical data of weeks to amount of sales is fed into the arima model for predicting the amount of sales for the future weeks. We have trained the model on the 80% of the data and predicted the net sales for the remaining weeks.

5.2.2 LSTM Model

Long Short Term Memory networks(LSTM) are a special kind of RNN, capable of learning long-term dependencies. LSTMs are explicitly designed to avoid long-term dependency problem. Our LSTM model was built to experiment on all the items across departments with 108 weeks worth of data. The model predicts the stocks in week level, using four consecutive weeks to predict the sales of the fifth week.

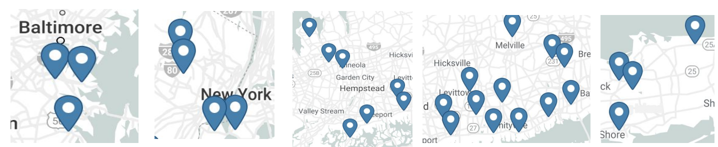


Selecting optimal parameters for a neural network architecture can often make the difference between mediocre and state-of-the-art performance. We considered 5 hyper parameters for tuning like the number of LSTM layers (nlayers), the number of hidden units in each LSTM layer (nhid), the learning rate of the optimizer (lr), backpropagation through time sequence length (bptt), dropout - applied to the layers (dropout). All experiments involved training for 100 epochs inline with available GPU resources. The training criterion was the cross-entropy loss which is the average negative log-likelihood of predicting the next word by the LM. It took approximately half an hour wall clock time to train the model for this hyper parameter configuration.

5.2.3 LSTM Model after k means clustering

K means Clustering of stores

Above two approaches doesn't provide satisfactory results. This is because we were considering all the cases when a customer moves out of his preferred store and this is causing too much of unwanted redundant results which is affecting the models performance. So now we have tweaked our hypothesis a little bit, by not considering on the basis of stores. Instead, we have grouped all the stores into different clusters based on their proximities, location and zip code from US_Zip_Code_Dataset[1]. The closer stores are in the same cluster and vice versa. After clustering, a data pipeline is created to get the number of stocks bought by customers of a particular cluster from outside their cluster.



cluster_1 cluster_2 cluster _4 cluster_5

Cluster	Stores #	State
cluster_1	X, Y, W	MD
cluster_2	U, S, a, T	NJ
cluster_3	M, I, A, B, Q, R, 6	NY
cluster_4	9, E, 8, 4, O, G , 7, V, 1, 2	NY
cluster_5	F, J, 3, K, L	NY

DataPipelines

1. In the initial step we have taken all the Stores # and Customer Data and linked it with Zip Code external Dataset to get zip codes associated with it.
2. We then grouped the stores into 5 clusters based on the location and distance between them.
3. Each store is then assigned to a cluster based on their nearest preferred store.
4. We have then considered the total number of stocks that are being purchased by customers outside their clusters. This is done for all the 5 clusters
5. Now weekly stock bought from outside clusters is computed

Model

This updated LSTM is no different from the previous model except from the input vectors and Model parameters. The input vectors are taken from the data data generated by the data pipelines described above. The model with a little tuning has shown 25% less error than the previous LSTM Model. The parameters of the Neural Network Architecture of LSTM model is as shown in the table below.

Hyper parameters	Values
number of LSTM layers(n_layers)	50
number of hidden units in each LSTM layer (n_hid)	2
backpropagation through time sequence length (bptt)	4
learning rate of the optimizer (lr)	0.001
dropout	0.5

6. RESULTS AND EVALUATION

6.1 Shell Space Management

6.1.1 Apriori

For the given dataset, Apriori algorithm generated association rules only for the possible top most high frequency itemsets and combination of items. Here is the sample results.

Items	Antecedent	Consequent
{ORANGE BURST SUET, BERRY BLAST SUET}	{ORANGE BURST SUET}	{BERRY BLAST SUET}
{ORANGE BURST SUET, BERRY BLAST SUET}	{BERRY BLAST SUET}	{ORANGE BURST SUET}
{BERRY BLAST SUET, PEANUT CRUNCH SUET}	{PEANUT CRUNCH SUET}	{BERRY BLAST SUET}
{BIRD BLEND SUET, ORANGE BURST SUET}	{ORANGE BURST SUET}	{BIRD BLEND SUET}

Products in Consequent columns are more likely to be bought by the customer when they buy the items that are listed in Antecedent.

However, we encountered a bottleneck when we wanted to find the set of products that are bought together for the given product. Also, support and confidence values are calculated manually, and tuning these parameters for a very large dataset of 17M transactions is computationally expensive and difficult to find the right value. So, we have to find better approaches to solve the problem mentioned above.

6.1.2 Word2Vec (Basic Model)

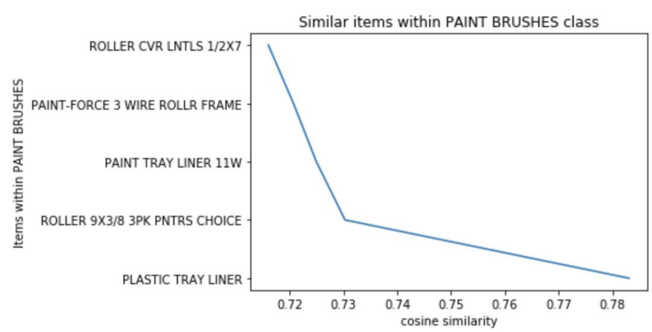
The word to vec model is trained with the items of over 1lakh words or features. The model is given with input of two items within a department and the model outputs how similar these two items are based on the feature vector of each of the two items

Evaluation metric **:** *Cosine Similarity*

Cosine similarity is a metric used to measure how similar two items are irrespective of the dimensions. The cosine similarity is advantageous because even if the two similar documents are far apart by the Euclidean distance (due to the size), chances are they may still be oriented closer together.

$$sim(u,u') = cos(\theta) = \frac{\mathbf{r}_u \cdot \mathbf{r}_{u'}}{\|\mathbf{r}_u\| \|\mathbf{r}_{u'}\|} = \sum_i \frac{r_{ui} r_{u'i}}{\sqrt{\sum_i r_{ui}^2} \sqrt{\sum_i r_{u'i}^2}}$$

Results: The results after training the wordtovec model on entire dataset and predicting how similar any two products is shown in the graph below



From the result graph, it can be noticed that the item *'Paint Roller'* of PAINT BRUSH class has items like '*Rollers', 'Paint Trays' and 'Liner*'(which are used for the same purpose) similar with it.

6.1.3 Word2Vec (Advanced Implementation)

Given few samples showcase working of the Word2Vec model trained on full dataset (containing only sales entries). As seen from tables, for a given item, 3 items are shown in results along with **confidence - probability** of coming together in cart/basket. From Item Description too, we are able to validate the performance manually.

```
Given item is =>  ['CM SLIDE LOCK KNIFE']
**********
Results
**********
  Item Number          Item Description  Confidence
0       23178   POUCH TOOL 6POCKT #W438    0.691442
1       20530  UTILITY KNIFE CLASSIC 99    0.691259
2     2199693   UTIL KNIFE AUTO LOAD ACE    0.689601
```

Suggested items for ‘CM SLIDE LOCK KNIFE’

```
Given item is =>  ['SCRAPR RAZOR5BLADES ACE']
**********
Results
**********
  Item Number          Item Description  Confidence
0     1500289  SAFE GLASS SCRAPR W/BLAD    0.811811
1     1337658     GLASS SCRAPER MINI ACE    0.810539
2     1053255       GLASS SCRAPER 4" BLD    0.795909
```

Suggested items for ‘SCRAPR RAZOR5BLADES ACE’’

```
Given item is =>  ['BRUSH SCRUB HANG-UP']
**********
Results
**********
  Item Number          Item Description  Confidence
0    AR207372   SCRUBBER GRILL & GARAGE    0.780868
1       AR203               SCRUB BRUSH    0.774784
2       10271  DIRTEX ALL PURP CLEANR1#    0.762605
```

Suggested items for ‘BRUSH SCRUB HANG-UP’

```
Given item is =>  ['STEELWOOL #0000 12PK']
**********
Results
**********
Weather => Whole year
  Item Number          Item Description  Confidence
0     1361088         STEELWOOL #00 12PK    0.833326
1       11383   REFINISHR FURN QT MINWAX    0.790331
2     1031749  OIL DANISH WATCO QT MDWT    0.772870
Weather => spring
  Item Number          Item Description  Confidence
0     1361104          STEELWOOL #0 12PK    0.729128
1     1460583   KLEAN-KUTTER REFINSHR QT    0.721853
2       11332  FINISH TUNG OIL PT MINWX    0.719852
Weather => summer
  Item Number          Item Description  Confidence
0     1054345   RESTOR-A-FNSH GLDNOAK PT    0.740860
1       12655   REFINISHR FURN QT FORMBY    0.699176
2     1093178    RESTOR-A-FNSH CHERRY PT    0.698366
Weather => fall
  Item Number          Item Description  Confidence
0       12655   REFINISHR FURN QT FORMBY    0.728157
1     1054345   RESTOR-A-FNSH GLDNOAK PT    0.713954
2     1460583   KLEAN-KUTTER REFINSHR QT    0.704885
Weather => winter
  Item Number          Item Description  Confidence
0       12655   REFINISHR FURN QT FORMBY    0.689396
1       12964          DEFTHANE SATIN QT    0.683343
2     1055276  RESTOR-A-FNSH NEUTRAL PT    0.682750
```

```
Given item is =>  ['BISSELL DAILY RENTAL']
**********
Results
**********
Weather => Whole year
    Item Number          Item Description  Confidence
0   BISSELLTOOL        BISSELL TOOL RENTAL    0.969191
1  BISSELLCLEAN      BISSELL CLEANING FEE    0.928004
2       1500693  OXY CARPT CLNR2X CON32OZ    0.919572
Weather => spring
    Item Number          Item Description  Confidence
0   BISSELLTOOL        BISSELL TOOL RENTAL    0.955958
1       1500693  OXY CARPT CLNR2X CON32OZ    0.945707
2  BISSELLCLEAN      BISSELL CLEANING FEE    0.934690
Weather => summer
   Item Number          Item Description  Confidence
0  BISSELLTOOL        BISSELL TOOL RENTAL    0.968135
1      1500727  PET CARPT CLNR2X CON32OZ    0.913730
2      1500693  OXY CARPT CLNR2X CON32OZ    0.913609
Weather => fall
    Item Number          Item Description  Confidence
0   BISSELLTOOL        BISSELL TOOL RENTAL    0.948152
1  BISSELLCLEAN      BISSELL CLEANING FEE    0.914630
2       1500693  OXY CARPT CLNR2X CON32OZ    0.885298
Weather => winter
   Item Number          Item Description  Confidence
0  BISSELLTOOL        BISSELL TOOL RENTAL    0.941509
1      1500727  PET CARPT CLNR2X CON32OZ    0.931572
2      1500693  OXY CARPT CLNR2X CON32OZ    0.918386
```

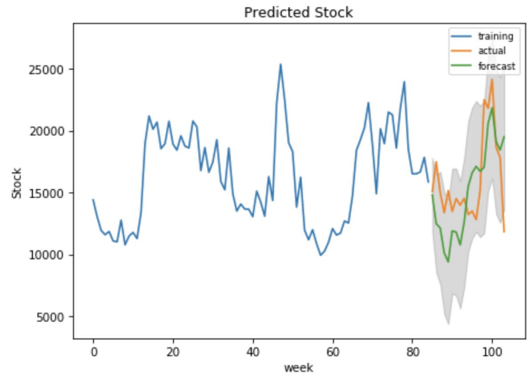
Suggested itemset for given item for varying weathers

Here, suggestions for a given item across 4 weathers i.e. Spring, Winter, Fall, and Summer are shown in these samples. An assumption that model will be biased to few items because of a single customer or bulk transactions is checked here by trying most frequent, least frequent and few other general items. From samples, it is clearly visible that suggestions stay across weathers with strong confidence.

6.2. Out of Stock

6.2.1 Arima (Base-line Model)

We trained arima model with 85 weeks data by setting 19, 0, and 8 as values for parameters p, d, and q respectively, and predicted the sales for the remaining 19 weeks. Here is a graph showing the actual number of items sold (orange) and the predicted value (green).

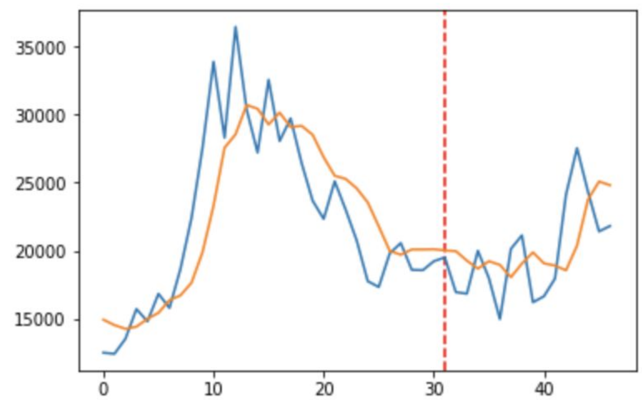


Evaluation Metric - ***RMSE 63.47***

Model is **evaluated** with the help of **RMSE** value. For a basic model, Arima is able to predict the sales per week to a certain extent with an rmse of 63.47. Though the model is performing well, we felt accuracy can still be improved with the help of other powerful models.

6.2.2 LSTM Model

The results of Arima mode is not satisfactory. So we tried using deep learning model and see how it responds to the data. The results of the LSTM Model is shown on the figure below.



Evaluation Metric - ***RMSE 59.77***

Surprisingly the rmse value is not reduced to a noticeable level. The reason for this is we have considered redundant data of the customers moving out of their preferred stores. This is rectified in the our next approach by clustering the transaction data based on stores.

6.2.2 LSTM Model after Kmeans clustering

After grouping the stores by k means clustering based on the distance, the stock data generated by data pipeline is fed to the LSTM model with little change in the network architecture. After fine tuning the model, the average rmse obtained for all the clusters was nearly 20% less than compared to the previous models.

*Evaluation Metric - **RMSE 47.69***

A detailed analysis of the results of the LSTM output is done for Cluster 1. The same applies to other clusters as well. Fig 6.2.2a shows the amount of stock that in cluster 1 during the time period 2017-2018 weekwise. Fig 6.2.2b shows the amount of stock to be added (generated by the data pipeline) to that in cluster 1. Fig 6.2.3a shows the total amount by adding the stocks in figure 6.2.2a and Fig 6.2.2b. Fig 6.2.2d shows the LSTM output of the stock to be there is Cluster 1 across the 2 year period.

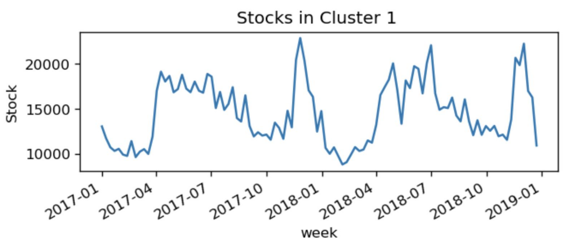


Fig 6.2.2a Stocks in Cluster 1 week wise 2017-2018

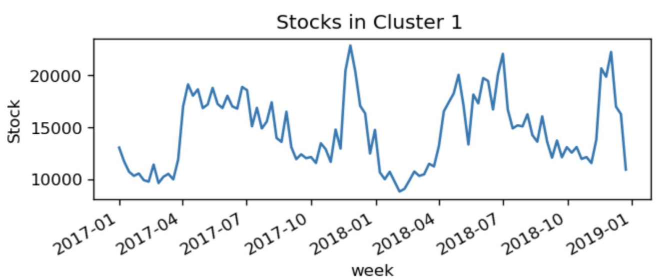


Fig 6.2.2b Stocks to be added in Cluster 1 week wise in 2017-2018

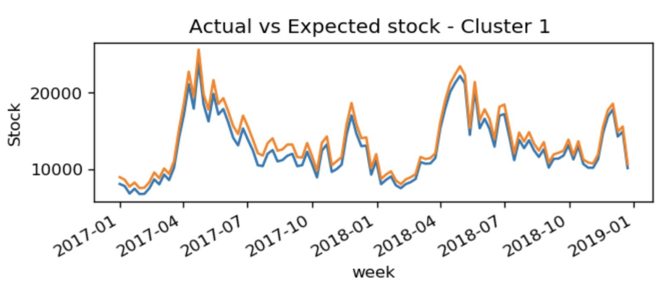


Fig 6.2.2c (Stocks in Cluster 1 + Stocks be added) vs Actual stocks in Cluster 1 week wise in 2017-2018

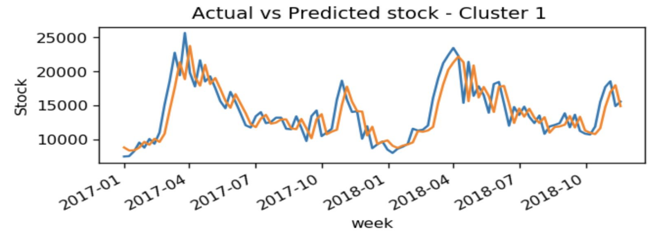


Fig 6.2.2d LSTM Predicted Stock vs Actual Stock Cluster 1 week wise in 2017-2018

There is absolutely no use in just telling the total number of stocks to be added. The distribution of the items to be stocked is the main objective. A sample of such distribution for Cluster 1 with the count of the items to be stocked in a particular week of each year 2017 and year 2018 is shown in the table below

Year - 2017 Week - 7

Total no of stocks to be added - 1312

Item Description	No of stocks to be added
FASTENERS	41
WEED & FEED 5M	28
KEY KWIKSET KW1	21
BIRDSEED WILDBIRD 20#ACE	17
PEANUT CRUNCH SUET	17

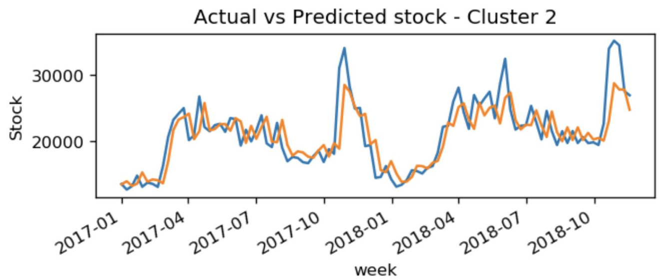
Year - 2018, Week - 45

Total no of stocks to be added - 611

Item Description	No of stocks to be added
BIRDSEED SONGBRD WB7# KT	118
FASTENERS	46
BIRDSEED WILDBIRD 20#ACE	21
PEAK WASH/DEICER	21
LAWN FOOD 5M	20

The LSTM predicted output for the other clusters is shown in the graph below.

Fig 6.2.2f LSTM Predicted Stock vs Actual Stock



Cluster 1 week wise in 2017-2018

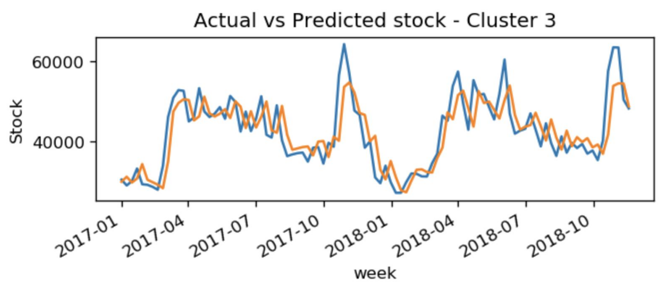


Fig 6.2.2g LSTM Predicted Stock vs Actual Stock Cluster 3 week wise in 2017-2018

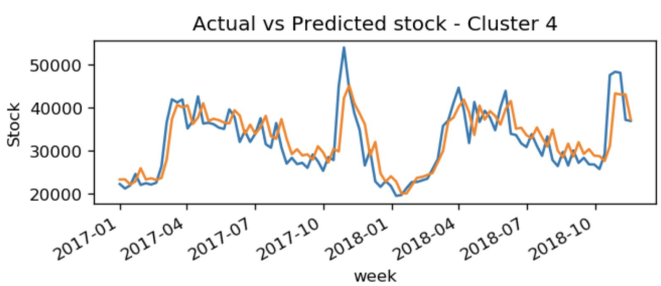


Fig 6.2.2h LSTM Predicted Stock vs Actual Stock Cluster 4 week wise in 2017-2018

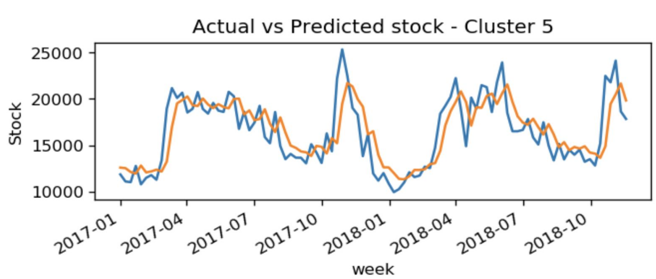


Fig 6.2.2i LSTM Predicted Stock vs Actual Stock Cluster 5 week wise in 2017-2018

7. CONCLUSION

In this report, we have successfully worked on two of the issues - *Shelf Space Management* and *Out of Stock* in the Retail Sector. We have considered a hypothesis, conducted analysis, plotted graphs, built models, got results and evaluated our results to approve our hypothesis. I hope working on solving these two issues would help Costello Ace Hardware to improve their sales in their next fiscal year.

8. REFERENCES

[1] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, "Distributed representations of words and phrases and their compositionality," in proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'13), C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.), Vol. 2. Curran Associates Inc., USA, 3111-3119, Lake Tahoe, Nevada, 2013.

[2] L. Piciu, A. damian, N. Tapus, A. Simion and B.Dumitrescu, "Deep recommender engine based on efficient product embeddings neural pipeline", in [2018 17th RoEduNet Conference: Networking in Education and Research (RoEduNet)]

[3] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. ICLR Workshop, 2013.

[4] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In EMNLP, volume 14, pages 1532–1543, 2014.

[5] Agrawal R, Srikant R., "Fast Algorithms for Mining Association Rules", VLDB. Sep 12-15 1994, Chile, 487-99, pdf, ISBN 1-55860-153-8.

[6] R. Jin and G. Agrawal, "An Efficient Implementation of Apriori Association Mining on Cluster of SMPs," Proc. Workshop High Performance Data Mining (IPDPS 2001), Apr. 2001.

[7] R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rules Between Sets of Items in Large Databases", Proceedings of the ACM SIGMOD Conference on Management of data, pp.207-216, May 1993.

[8] Hochreiter S, Schmidhuber J. Long Short-Term Memory. Neural Comput. 1997 Nov;9(8):1735–1780. Available from: http://dx.doi.org/10.1162/neco.1997.9.8.1735.

[9] https://www.arxiv.org/abs/1301.3781

[10]https://www.visittheusa.com/info/weather-seasons