Report on
Future drug sales prediction

Aravind Senguttuvan[1], Arun Allamsetty[2], and Prashant Waghulde[3]

[1]School of Computing, University of Utah

April 17, 2014

Contents

1 Introduction

We are trying to develop a system which predicts drug sales for the upcoming year based on past archives of data by mining and visualizing it to draw valid conclusions. Sales are the lifeblood of a business. It's what helps pay employees, cover operating expenses, buy more inventory, market new products and attract more investors. Sales forecasting is a crucial part of the financial planning of a business. It's acts as self-assessment tool that uses past and current sales statistics to intelligently predict future performance. As your business grows, sales forecasts continue to be an important measurement of your company's health. When attracting new investors, sales forecasts. The overall effect of accurate sales forecasting is business that runs more efficiently, saving money on excess inventory, increasing profit and serving its customers better

2 Motivation

Pharmaceutical companies have to spend millions of dollars for manufacturing, transportation and sales. These three processes in the supply chain model are dependent on geographical locations and drug usage of a geography.

Pharmaceutical companies yearn for a better supply chain model always. The companies have a lot of data regarding sales of drugs. This continuously growing dataset can be mined to arrive at conclusions which might end up saving capital.

A Practical example would be as follows. A company can decide to move or create their manufacturing site nearby a growing sales region to cut down transportation. While planning for procurement, production and logistics capacity taking in to consideration the resources one commit to sales and business-development activities and the length of time a sales process can take. No one wants to lose an order because they were unable to deliver on time. Sales forecasts bring a level of clarity to future delivery volumes and help you plan your own procurement and production schedule and avoid supply-chain shortages.

So, we planned to determine the most probable drug sales for a particular region.

3 Data Collection

3.1 Data source

We were provided data pertaining to pharmaceutical claims by Dr. Chad Hokama from Audax Health Solutions. We are currently working on a sample database having approximate size of 1 GB containing around 6 million records.

3.2 Pre-processing of Data

The data we received was in JSON format which was broken. The files contained incomplete JSON arrays. Since we used Hadoop Streaming and wrote our MapReduce code in Python which reads the data from standard input, we had to convert the broken JSON array to single lines using the Unix sed command. This ensured that problem is more parallelized since the data could be read by multiple mappers simultaneously since the data was now in multiple lines compared to a single line as was the case before pre-processing.

3.3 Sanitizing data in JSON format

We sanitized the data by taking care of improper or empty zip codes which could not be mapped to the User Interface.

”prescSpecialty”: ”10201”, ”threeDigitSubsZip”: ”852”, ”untsDispensedQuantity”: ”60.000”, ”generic”: ”Y”, ”gender”: ”F”, ”ndc”: ”0378-3125-01”, ”dispenseQuarter”: ”2011Q3”, ”ndc11Digit”: ”00378312501”, ”birthYear”: ”1951”, ”threeDigitPhmZip”: ”123”, ”daysSupplyCount”: ”30”, ”newRefillCount”: ”2”

3.4 Data Processing using MapReduce

We needed to get accumulated drug count grouped by where and when the drug was sold. For this, we decided to employ MapReduce as this problem fell into the category of problems which are embarrassingly parallel. MapReduce was an apt solution because it can handle such problems with ease and also scale to thousands machines without any change in the code.

As we know MapReduce has two basic steps, Map and Reduce and both these steps help process the data efficiently in a distributed framework. The Map phase generally involves the reading of the data from the distributed file system. This data is then converted into key–value pairs and passed on to the Reducer. The keys then, before being passed on to the reducer are shuffled and all the values belonging to a particular key are grouped together and sent to one particular reducer (in case of multiple reducers). That is, if we have more than one reducer, each reducer receives a specific key and all its values. So therefore, the key selection during the Map phase dictates how the entire application will perform.

- *Map:* Our Map phase consisted of taking the time period and the location together as a composite key since those were the fields on which we wanted to group and count.
- *Reduce:* The Reduce phase took all the counts for the above mentioned composite key and aggregated them.
- *Combiner:* We also employed a combiner, which though did not help in our single–machine cluster but would surely have a positive impact when the same process is carried out in the in a large cluster. What the combiner does is it aggregates the data in the Mapper level for each key and then sends it to the reducer. This way the Reducer has to do significantly less work while aggregating the data.

4 Algorithms used for drug count prediction

We calculate the predicted drug count for each cluster instead of a state because of geographic closeness of sales units.

- *Add 1 smoothing* to take into account the zero turnover sales outlets.
- *Weighted K–mediods* to find sales clusters.
- *Future prediction* using step difference.
- *Visualization of clusters* using convex hulls.
- *Population based counting* Effect of population on year to year change in drug count.

4.1 Add 1 smoothing to the zero weights in the cluster

Over the year range, We assign the weight (drug count) to be 1 for sales outlets in the zip Since weighted kmediods is used, the sales outlet which didnt had zero turnover will only have geographical effect on the cluster

4.2 Weighted K mediods

We used k mediods instead of k means. Instead of having average of all points in the cluster as our new center. We wanted the center of the cluster to be a point in the cluster itself. This is because we need a sales outlet to be center of a cluster so that there is no bias.

We use weighted average to recompute the new center of the cluster. This is because the clusters center should be based on the number of drugs sold in a sales unit too.

```
Distance_threshold = (sum of america latitude and longitude spread) /
    number of clusters
Take k points randomly or using gonzalez as centers
Until there distance between all k clusters is equal to
    Distance_threshold
     1: Take a new point x,y with weight w
     2: Find the center among k centers which has shortest distance from
         x,y
     #To Compute new center
     3: Find the weighted average of all points in the cluster
     4: New Center  = The point with shortest distance from the average
         point
```

4.3 Future prediction

We formed our basis for future prediction based on yaxis difference (drug count) for each step. Also we provide more weightage to the recent yaxis step difference.
Predicted value function ypred()

$$ypred(n) = \frac{\sum_{1}^{n-1} (y[i] - y[i-1]) * i}{(n-1) * n * 0.5} \tag{1}$$

The above predicted value can have a range of $\pm\epsilon$ around y[n]
Approximation range

$$\epsilon = \sum_{1}^{n-1} \frac{y[i] - ypred(i)}{i} \tag{2}$$

The below code succintly expresses the future count calculation for each cluster.

```
# FUNC: Predicts the value of the point specified by 'index'
def predict(ys, index, data):
     y_predicted = 0
     for i in range(index, 1, -1):
         y_predicted += data[str(i - 1)] * (i - 1)
     return y_predicted / sum(range(1, index))

#FUNC: Predicts the values for all the clusters.
def predict_all(drugs_per_clus, index):
     epsilons = {}

     clusters = drugs_per_clus.keys()
     clusters.sort()
     for i in range(0, len(clusters)):
         cluster = clusters[i]
         ys = drugs_per_clus[cluster].keys()
         ys.sort()
         data = drugs_per_clus[cluster]
         prediction = predict(ys, index, data)
         drugs_per_clus[cluster][str(index)] = prediction

         epsilon = 0
         for j in range(3, index):
             epsilon += data[str(j + 1)] - predict(ys, j + 1, data)
         epsilons[i + 1] = int(epsilon)

     return drugs_per_clus, epsilons
```

4.4 Visualization of clusters

We created convex hulls for Cluster visualization using Quick Hull algorithm.So that it is easy to know the geographical spread of a cluster. Convex hull provides an approximation to the cluster region. The classification process is achieved by evaluating the variation of the data density of that region.

Quick Hull Algorithm:
The below algorithm runs faster hence it is best to use that in User Interface based applications.

1. Find the points with minimum and maximum x coordinates, those are bound to be part of the convex

2. Use the line formed by the two points to divide the set in two subsets of points, which will be processed recursively.

3. Determine the point, on one side of the line, with the maximum distance from the line. The two points found before along with this one form a triangle.

4. The points lying inside of that triangle cannot be part of the convex hull and can therefore be ignored in the next steps

5. Repeat the previous two steps on the two lines formed by the triangle

6. Keep on doing so on until no more points are left, the recursion has come to an end and the points selected constitute the convex hull

4.5 Population based counting

From year to year the population also increases hence it would be wrong to derive drug count as a measure for sales. Hence, deciding the drug sales count based on population was thought of. We got data only for 2011. Hence the graphs following show only drug count.

5 Related Work

We came across a paper based on prediction of sales data in SAP ERP system using clustering algorithms written by Department of Computer Science & Systems Engineering, Andhra University,Vishakapatnam, India.

Sales and distribution function of an enterprise handles many important business processes like preparing annual monthly marketing plans, developing pricing strategies, handling service contracts, manage logistics for material dispatches. Pre-sales information is used to plan and evaluate marketing & sales strategies and as a basis for establishing long term relationship with customers.

They sourced information relating to customer demands for different steel products measured in terms of both Value and Volume. Then they applied clustering techniques to divide the sales data of enterprise into clusters that are meaningful, useful or both.

Cluster analysis helped them in correctly assessing the demand and the net sales realization for the product to be developed. It enabled them to identify lost order opportunities and improve sales volumes for new products. It is often observed that for certain products niche market exists and sales value will also be high. However by making use of the available sales data they were able to establish relations between the pre-sale activities like tenders, quotations, enquiries etc with the sale order and post sale activities like pricing, destinations, invoicing, refunds, returns etc. This helped analyzing monthly planned vs. actual sales and also an ability to analyze product wise, customer wise sales over a year and further the customer base. The sales data analysis can also be used to achieve higher stock to sales ratio by exploring the possibility of customer wise product wise pricing.

The success criterion for any data mining application is accounted in terms of how accurate and acceptable is the implemented solution. This aspect is measured in terms of learned descriptions like new rules framed or inferences drawn which can be utilized by a human user. In this context above analytical results produced by clustering techniques proved to quite reliable

6 Division of work

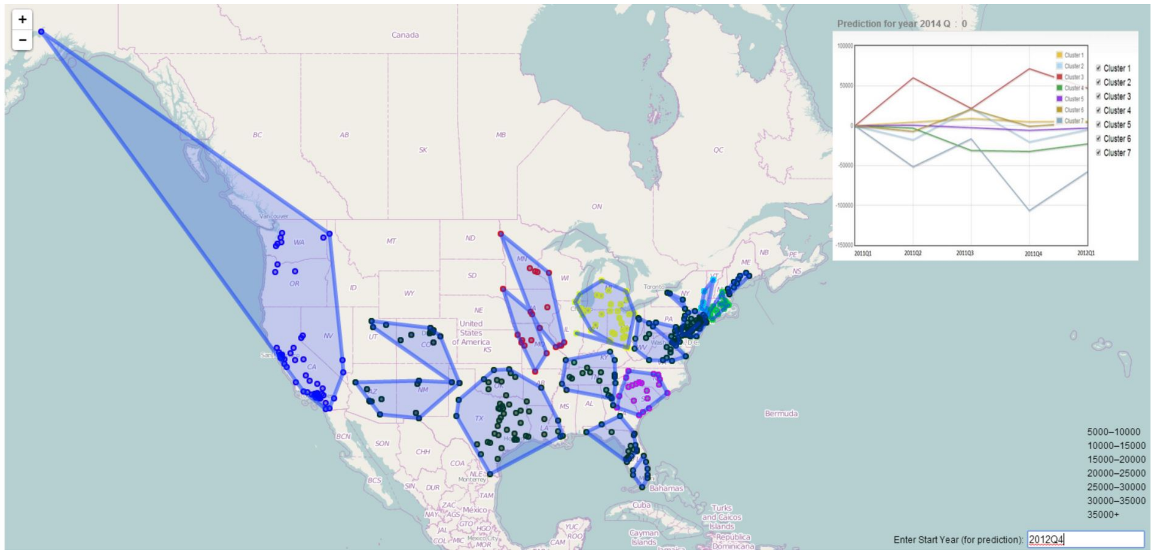
Table 1:

Work	Done By
Add 1 smoothing	Aravind, Arun
Weighted K–mediods	Aravind, Arun
Future prediction using step difference	Aravind, Prashant
Sanitize the JSON input and map reduce datasets	Aravind, Arun, Prashant
Comparison study - Weighted K–mediods, K–means, DBSCAN	Aravind, Arun, Prashant
Comparison study - Step difference, MATLAB interp1 (), Excel extrapolate	Prashant
Visualization of clusters using convex hull	Arun
Projecting data to Front end using php	Aravind

7 Code and screenshot of the tool in action

Code: https://github.com/ArunAllamsetty/drug_usage_prediction

Figures 1,2 and 3 shows the map and the clusters and graph of the predicted value



Prediction for year 2014 Q : 0

Enter Start Year (for prediction): 2012Q4

Figure 1: Tool in action

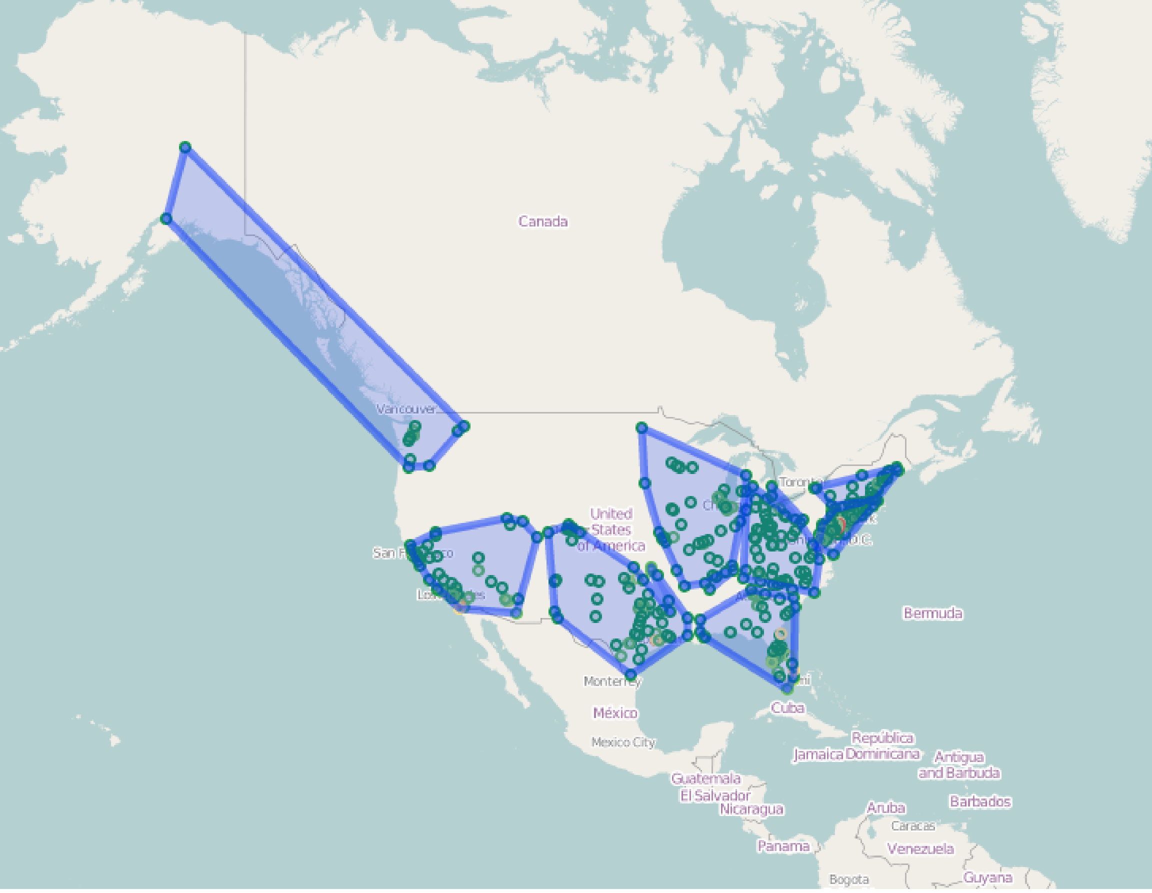


Figure 2: The data when the number of clusters were 7.

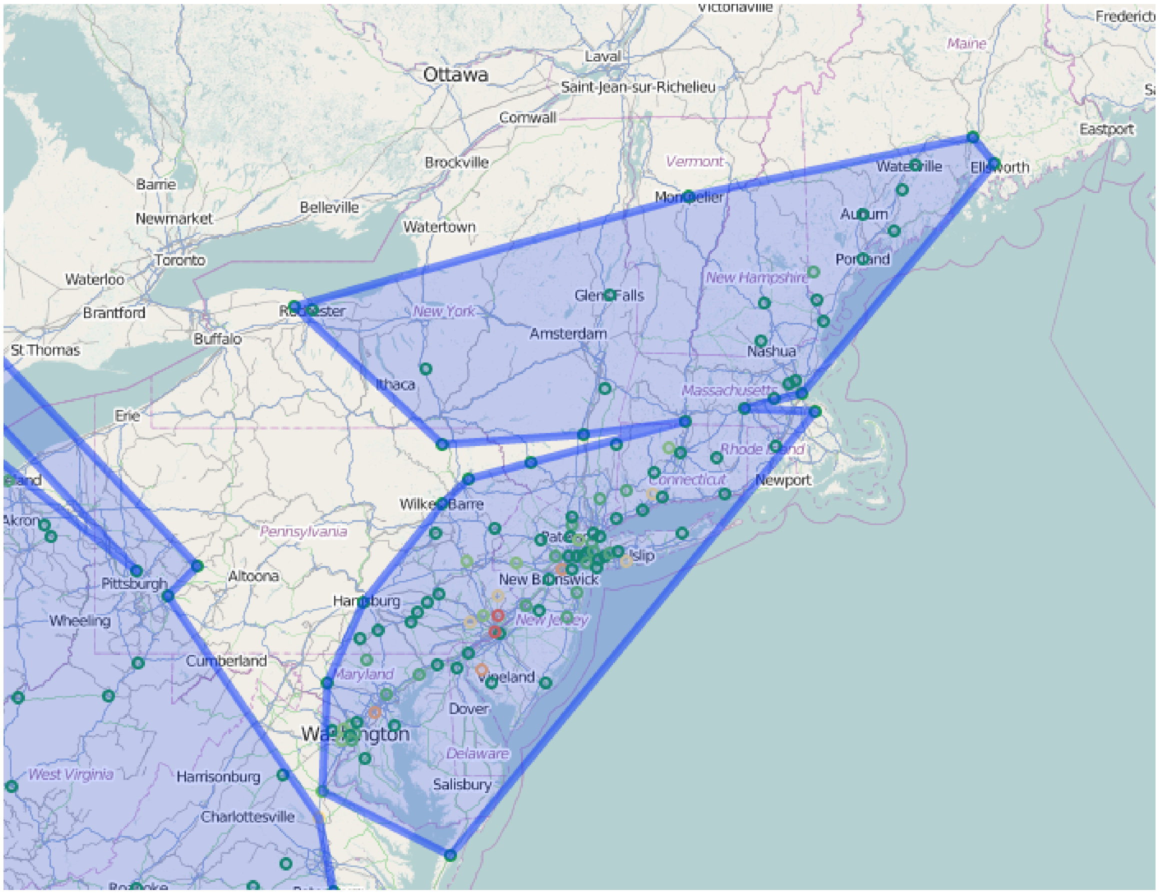


Figure 3: Zoomed into a cluster

8 Comparison Study

8.1 Our proposed method - Step difference based prediction

Figure 4 shows Step difference based prediction
Figure 5 shows predicted value for one of the clusters
Value of epsilson for cluster 1 calculated ±4069

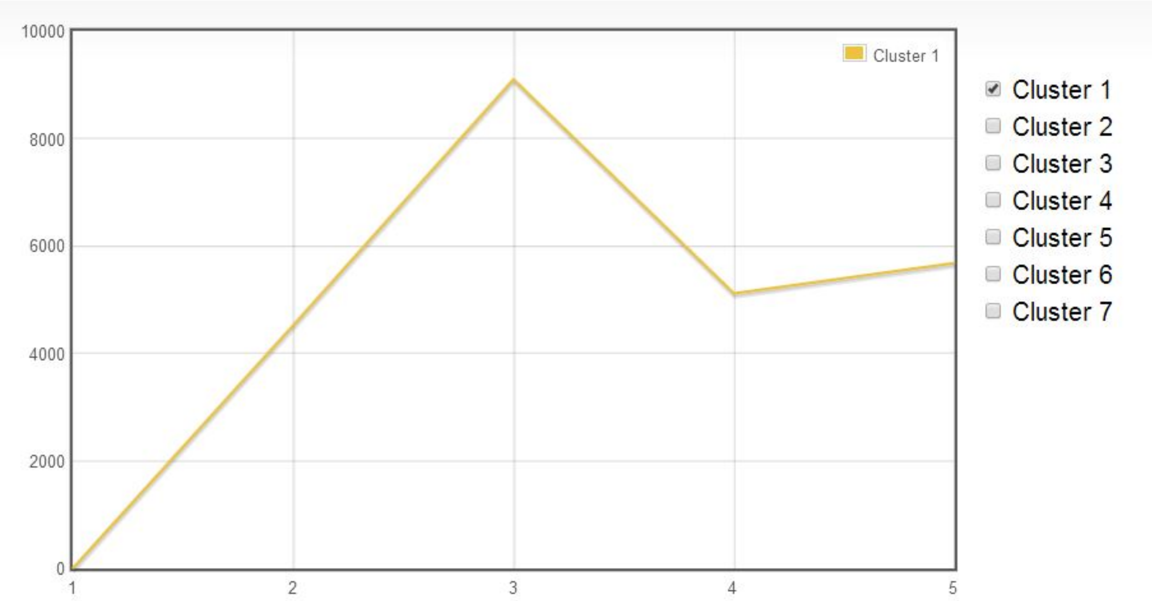
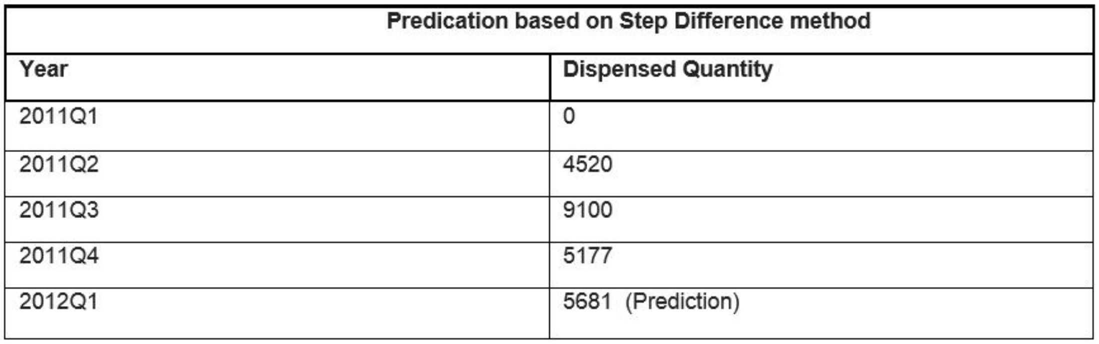


Figure 4: Step difference extrapolation



Predication based on Step Difference method	
Year	Dispensed Quantity
2011Q1	0
2011Q2	4520
2011Q3	9100
2011Q4	5177
2012Q1	5681 (Prediction)

Figure 5: Cluster comparison: Step Difference

8.2 MATLAB interp1

Figure 6 shows the Prediction using MATLAB .
Figure 7 shows predicted value for one of the clusters
Matlab uses Regression based method to form the linear fit.

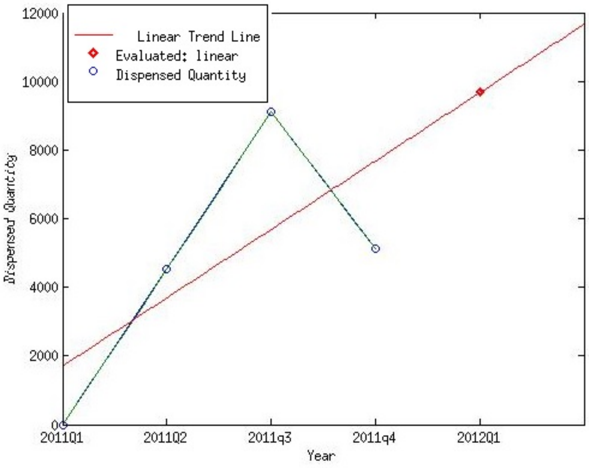
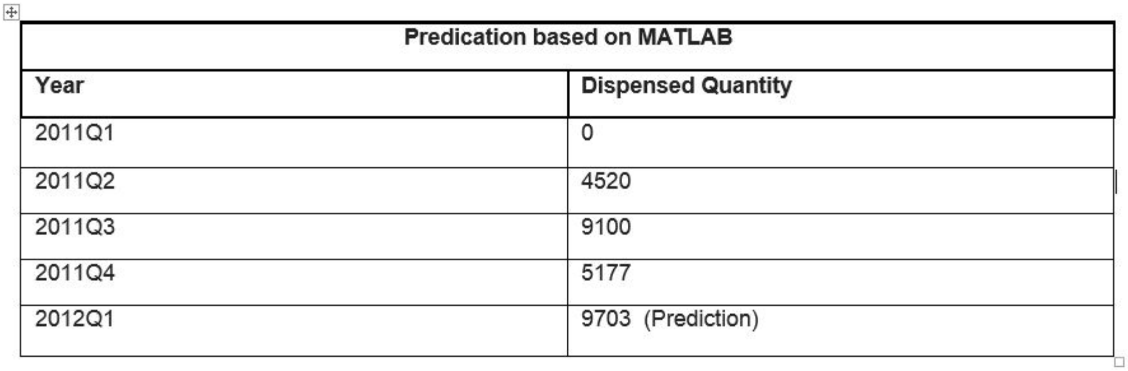


Figure 6: TrendLine: MATLAB



Predication based on MATLAB	
Year	**Dispensed Quantity**
2011Q1	0
2011Q2	4520
2011Q3	9100
2011Q4	5177
2012Q1	9703 (Prediction)

Figure 7: Cluster comparison: Matlab

8.3 Prediction using Excel

Figure 8 shows the Prediction using Microsoft Excel.
Figure 9 shows predicted value for one of the clusters

$$ypred(a + bx) = \frac{\sum(x - mean(x))(y - mean(y))}{\sum(x - mean(x))^2} \tag{3}$$

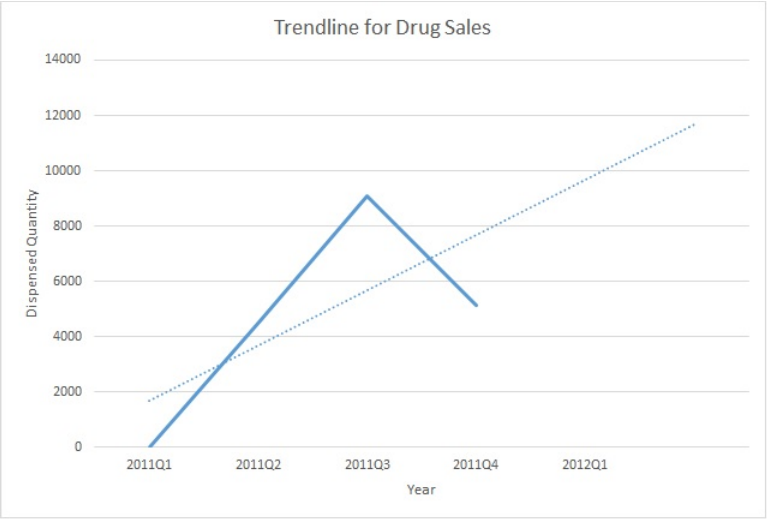
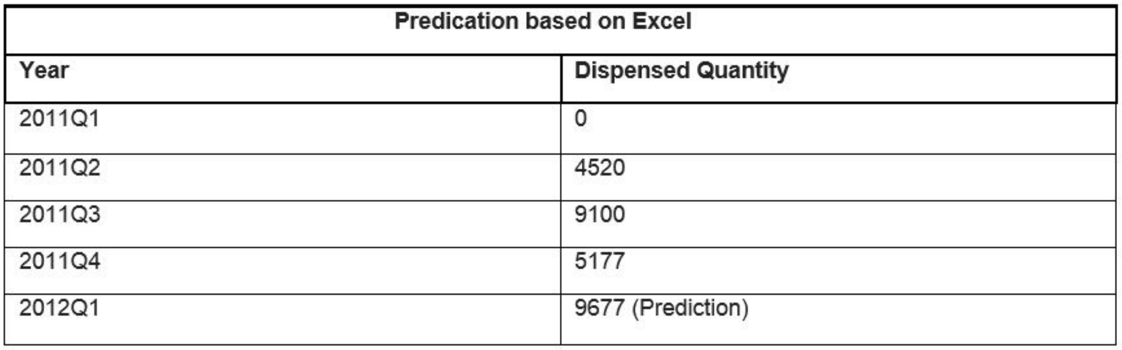


Figure 8: TrendLine: Excel



Predication based on Excel	
Year	Dispensed Quantity
2011Q1	0
2011Q2	4520
2011Q3	9100
2011Q4	5177
2012Q1	9677 (Prediction)

Figure 9: Cluster comparison: Excel

9 Comparison Study - Drug count Weighted K–mediods Clustering

9.1 Comparison with k-means

Our method takes care of the weights whereas k–means doesn't.

9.2 Comparison with DBscan

Although DBScan proves well while dealing with noise, we don't consider any point in here as noise. Outliers are considered as separate clusters in themselves.

10 Future Work

- We can get specific supply chain model of a specific drug based on identifiers called NDC.
- Before Cluster formation, we could omit outliers to be separate clusters. This will provide more sanitized data for prediction.
- For clustering and extrapolation we can take values from different methods and take an average to get perfect values.

11 References

KMeans : https://mahout.apache.org/users/clustering/k-means-clustering.html
NDC database : http://www.fda.gov/drugs/informationondrugs/ucm142438.htm
Population : http://blog.splitwise.com/2013/09/18/the-2010-us-census-population-by-zip-code-totally-free/
Basic idea of links: http://nlp.stanford.edu/IR-book/completelink.html
R-manual : http://stat.ethz.ch/R-manual/R-patched/library/stats/html/loess.html
Leaflet: http://leafletjs.com/index.html
Clustering methods: http://www.vldb.org/conf/1994/P144.PDF
Clustering methods: http://www.cs.uiuc.edu/homes/hanj/pdf/gkdbk01.pdf Quick hull: http://en.wikipedia.org/w

12 Conclusion

By analysing the distribution pattern, we would able to suggest the pharmaceutical company to incorporate a better supply chain model.