

# Design and Analysis of an Effective Channel Distribution Approach for Agricultural Commodities using MongoDB

Arunkumar Thangavelu<sup>1</sup> and N. Manoharan<sup>2</sup>

<sup>1</sup>School of Computer Science and Engineering, VIT University, Vellore – 632014, Tamil Nadu, India;  
profarunkumart@gmail.com

<sup>2</sup>Computer Science, P. A. College, Manomaniam Sundaranar University, Tirunelveli – 627012, Tamil Nadu,  
India;manoonam22@yahoo.co.in

## Abstract

**Objectives** The farmers are getting far less cost for their farming products since they didn't know where to offer for the best cost. This issue could be settled by this paper. **Methods/Statistical Analysis:** We worked with Historical Data of various agricultural commodities production, demands and pricing at different locations using MongoDB, a NoSQL database tool. A Module where the Buyers of Agricultural Commodities would enter their need and the costs they offer, for commodities. Demand qualities and Current Price datasets are created. The calculations for each of the Farmer Module and Buyer Module were composed and executed to satisfy the point of guaranteeing that the farmers can offer his wares at the ideal cost. **Findings:** Making a Module, where the Buyers of Agricultural Commodities would enter their need and the costs they will offer, store these points of interest and in light of these subtle elements, Demand qualities, and Current Price datasets are created. Farmer modules were created with the available agricultural commodities and their cost, a location of availability. The calculations for each of the Farmer Module and Buyer Module were composed and executed effectively. The calculations were tried for different situations and the normal results were figured it out. Examine the information and the farmer's need (i.e) estimation called as horticulture yield investigation, and in this way discover the different valuable examples and making sense of the best methodologies at the agriculturists to cost and offer their yields in various locales at various times of the year. To satisfy the point of guaranteeing that the rancher can offer his wares at the most ideal cost. **Applications/Improvements:** This application could be moved to the cloud with MongoDB server and the farmers and purchasers will be given a separate login id. Total system may be available in web.

**Keywords:** Agriculture Business, Agricultural Commodities, Big Data, NoSQL, MongoDB

## 1. Introduction

India even after all the Industrialization and urbanization, its principle occupation still stays to be "Farming". The part offers work to about 60% of the nation furthermore is the real giver towards its GDP<sup>1</sup> (Gross Domestic Product)

Almost the whole rustic group is subject to this occupation and a large portion of their different occupations that are rehearsed have direct conditions on "Farming". On the off chance that we can calibrate the Agriculture segment and its general practices it will have direct positive

\*Author for correspondence

ramifications on the country populace mostly their monetary improvement.

Farming remains the real wellspring of pay to a lion's share of the general population of this nation notwithstanding the forceful industrialization, urbanization, IT unrest, and so forth. It gives the business to almost 70% of the general population, and dominant parts of the general population included have a place with the provincial group. Consequently, with regards to country India's advancement, with no uncertainty "Horticulture" is the central point and any kind of change in the region of agribusiness would positively affect the provincial improvement and the cultivating group.

The horticultural items wares delivered by the agriculturists are estimated varying at various costs because of the interest of the item in that specific area. Additionally, the costs of these items change at various times of the year in various areas in the nation all subject to the interest and how this interest is being met by their typical suppliers. At last, the items that are created by the farmers are typically brought for far not exactly the value that is eventually paid by the end customer. Be that as it may, the rancher is not really mindful of these components and he keeps on getting paid the lesser value independent of the interest for his produce and because of this element their financial development is influenced and is low contrasted with different divisions in the nation. This is the immediate effect of the broken Supply-Distribution chain in the farming framework. The cultivating group<sup>2</sup> does not grow monetarily contrasted with individuals flourishing in different divisions of the group; for the most part, contrasted with individuals in the urban side this has made a huge financial hole between the provincial group and the urban group. This financial crevice has further repercussions. The general population in the urban area makes a great deal of interest for items that are devoured by both the segment of the general population. This interest builds the cost of these items which the urban segment of individuals pays without quite a bit of an issue, yet the provincial segment of the general population confronts a predicament because of this ascent in the item cost.

In the meantime, the farmers don't get an ascent in the costs of the items which they create until quite a while later. Because of this, there is a substantial financial crevice made between the urban and country area of the general public and it is because of this reality the rustic segment of the general public contributes less towards the expense accumulations.

The issue could be comprehended if the agriculturists could offer their item to the area where they could get a higher cost for their yield furthermore, that specific area must be close to the rancher; i.e. it must be achievable. This would tackle the farmers' issue as well as the customers, since over a time frame the costs would drop as the interest would be met. In this way, fulfilling both the maker and the customer needs. Upgrading the cultivating group and through its advancement connecting the financial separation between the Urban and Rural India has been the sole inspiration driving, taking up this issue and in this manner attempting to locate a viable supply/dissemination approach that would guarantee the farmer settles on his choice of offering his products in an informed way.

## 1.1 Objectives

The farmers are getting far less cost for their farming products since they didn't know where to offer for best cost. This issue could be settled by this technique.

**To simplify, the aim of this project can be explained as follows**

- To analyze the current demand trend sets of all the commodities.
- To analyze the historical data of the products and their selling prices<sup>3</sup> over different periods of time.
- To create a module that accepts inputs from Buyers about their location and needs, such as the quantity of the commodities needed and storing their details.
- To create an algorithm that is based on these inputs alters the Demand trend sets of the respective commodities.
- To create a module that accepts inputs from farmers about their needs, such as their Required Price for the commodities and the price that is being currently offered at their current location.
- To create an algorithm that is based on these inputs and the earlier obtained input from the Farmers needs and Demand Trend Sets, choose the best possible location to sell the commodities and other alternatives as well. With this we hope to achieve a right and fair Distribution/Supply chain where the farmers could get a higher price for their yield and also the demands of the consumer are met thereby creating a win-win situation.

## 1.2 Challenges

Like each task there is a sure difficulty that must be overcome to satisfy the targets. The real test has been the gathering of the datasets with all the important data present in them. For the reasons for examining the data sets that are required must have the accompanying data:

- The estimating of the different farming items.
- The estimating of these different farming items at various areas.
- The rural wares evaluating at various times of the year.
- The estimating of these farming items at different locales covering the whole nation.
- The request pattern sets of these different rural wares.

Another test is to include the Location points of interest, for example, Latitude and Longitude data, in their individual areas in the previously mentioned data sets. This angle is exceptionally essential since to play out the Geo-Spatial figuring, which is the center of the calculation to be executed, the area data<sup>4</sup> is a need.

## 2. Algorithm and System Design

The framework design that delineates the general working of the whole framework is appeared in Figure 1. The changed areas of the framework<sup>5</sup> required with various calculations are likewise portrayed plainly in the framework engineering these calculations, in particular Farmer Algorithm and Buyer Algorithm, are talked about in subtle element in later segments.

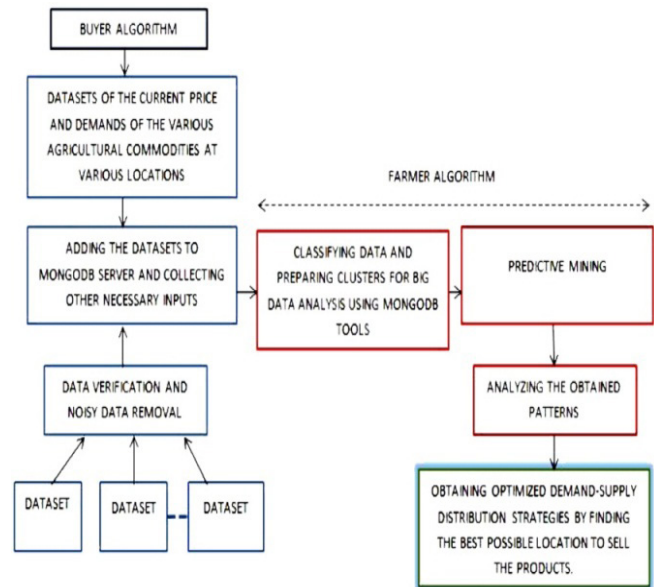


Figure 1. The main system architecture.

### 2.1 Methodology Used

The working of the above delineated design is depicted underneath. The datasets that are required for the investigation, i.e. the datasets containing past costs of Agriculture wares costs (Past and Present) and the Demand datasets, are gathered. At that point the datasets gathered in the Step 1 are checked in subtle element and afterward just the confirmed datasets are chosen with the end goal of preparing and investigations. After the clamor information (or) undesirable information<sup>6</sup> are expelled from the datasets for effective handling, these datasets are added to the MongoDB database in a particular gathering. Utilizing the “Purchaser Algorithm” we acquire information about the present costs and the interest pattern sets of the different items. The information acquired in Step 4 is added to the same database in MongoDB in partitioned accumulations.

Every one of the accumulations and different contributions from the client are subject to the “Farmer Algorithm”. The information is ordered into bunches. Every bunch speaks to a specific arrangement of related information. The groups of information are subject to prescient operation<sup>7</sup> where utilizing the information and certain arrangement of conditions we analyze the information. In view of the investigation<sup>8</sup>, we anticipate the most ideal area for the agriculturist to offer his items, likewise conceivable choices. Steps 7, 8 and 9 all things considered structure the “Farmer Algorithm”. So the outcome to the client.

The two calculations said above, i.e. the “Farmer Algorithm” and the “Buyer Algorithm” frames the center of the task. Despite the fact that they are separate they interrelated. The “farmer Algorithm” is reliant on the result of the “Buyer Algorithm, as the Buyer Algorithm’s yield is put away in the MongoDB database and the Farmer Algorithm utilizes the same database for the investigation and produces the yield. The “Agriculturist Algorithm” and the “Buyer Algorithm” are clarified in point of interest in the coming areas.

## 2.2 Buyer Algorithm

1. Accept the following inputs from the user:

- The commodity which the Buyer wishes to acquire.
- The required quantity of commodities the user wants..
- The price the Buyer is willing to offer for the commodities.
- The current location (or) market center of the Buyer.

2. Check if the Market Centre entered is present in the database. If yes, proceed to Step 4, else, request the user to check the spelling or enter another nearby Market Centre.

3. Retrieve the current demand value of the selected commodity from the MongoDB<sup>9</sup> database.

4. Alter the demand value of that particular commodity based on the quantity required by the Buyer. It is done as follows

If (quantity  $\leq 300$ ):  $d=0.05$

If (quantity  $\leq 500$ ) & (quantity  $> 300$ ):  $d=0.25$

If (quantity  $\leq 5000$ ) & (quantity  $< 500$ ):  $d=0.50$

If (quantity  $> 10000$ ) & (quantity  $> 5000$ ):  $d=0.75$

If (quantity  $> 10000$ ) & (quantity  $\leq 20000$ ):  $d=1.0$

If (quantity  $> 20000$ ) & (quantity  $< 30000$ ):  $d=1.25$

If (quantity  $> 30000$ ):  $d=1.50$

5. Add the value of ‘d’ obtained in Step 4 to the demand value of the respective commodity.

6. Check whether there are any other entries of the commodity in that market centre, in the Current Price Database. If yes proceed to Step 7, else add the entered details in the Current Price Database and go to Step 8.

7. Check if the offered price is greater than what is being currently offered at the market Centre for that particular commodity. If yes, replace the user entered price has the new offering price at that location in the Current Price Database.

8. Add the details of the User to the Buyer collection for reference purposes.

9. Intimate the user that the details have been added.

## 2.3 Farmer Algorithm

1. Accept the following inputs from the user:

- The commodity which the farmer wishes to sell.
- The required price the farmer wants for the commodities.
- The price the Market Centre people are offering for the commodities.
- The current location (or) Market Centre where the farmer wishes to sell his products.

2. The other input to the algorithm is a historical dataset of various agricultural commodities at various locations.

3. Check if the Market Centre entered is in the database. If yes, proceed to Step 4, else, request the user to check the Spelling or enter another nearby Market Centre.

4. Retrieve the current demand value of the selected commodity from the MongoDB database.

5. Calculate the Demand factor, say ‘D’, based on the demand value obtained in Step 4. It is calculated as follows:

- If demand value is equal to 0, then:  $D = 1$
- If (demand value  $> 0$ ) & (demand value  $< 3$ ), then:  $D = 1$
- If demand value  $\geq 3$ , then:  $D = 2$
- If (demand value  $< 0$ ) & (demand value  $> -3$ ), then:  $D = -1$
- If demand value  $\leq -3$ , then:  $D = -2$

6. From the historical data, find “Historical Data Factor”, ‘H’. It is calculated as follows:

- i. Find all the values of that particular commodity throughout the historical dataset. Let’s say the Total number of entries found = N.
- ii. If the value of those entries found in the database has a selling price of more than the price the The farmer wants, then  $T = T + 1$ .
- iii. If the value of those entries found in the database has a selling price lesser than the price the The farmer wants, then  $T = T - 1$ .
- iv. In the end calculate the Historical Data Factor,  $H = T/N$ .

7. Add the Demand Factor ‘D’ and the Historical Data Factor ‘H’ to give the search factor ‘S’ =  $D + H$ . The

value of  $S$  ranges from: -3 to +3. With two thirds weightage coming from Demand data and one third coming from historical data.

8. Based on the value of search factor ' $S$ ', the search radius is determined. It is done as follows.

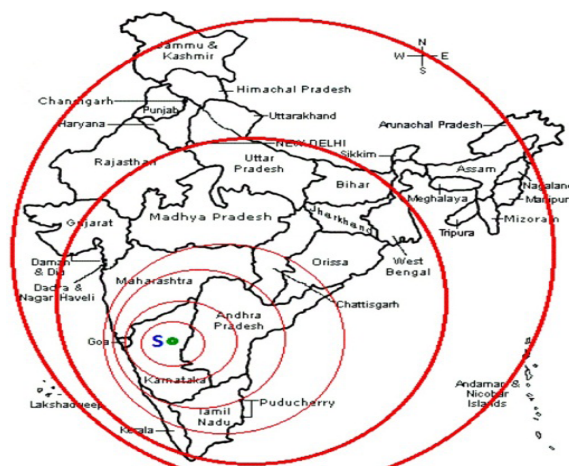
- If  $S \geq 2$ : Search in 2000 Km radius<sup>10</sup>
- If  $(S \geq 1) \& (S < 2)$ : Search in 1200 Km radius
- If  $(S \geq 0) \& (S < 1)$ : Search in 700 Km radius
- If  $(S \geq -1) \& (S < 0)$ : Search in 400 Km radius
- If  $(S \geq -2) \& (S < -1)$ : Search in 250 Km radius
- If  $S < -2$ : Search in 150 Km radius. By using radius we can able to find the locations by the Figure 2.

9. Find all the Market Centers within that radius that are offering a better price for the commodity than the required price mentioned by the farmer.

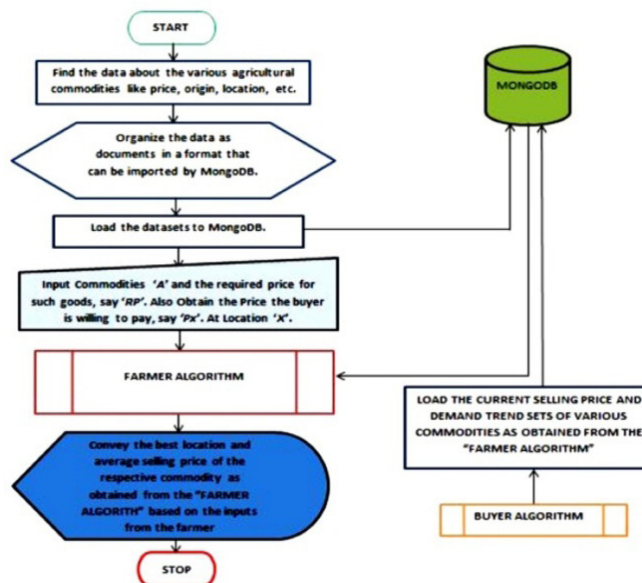
10. Display the Market Centre that offers the maximum price as the best location and other Market centers as alternatives

11. If there is no Market Centers are found offering then inform the user to sell the commodities at the current market center itself.

The Flowchart of Entire System is explained in Figure 3.



**Figure 2.** The search radius for the best place to sell the commodities from the location ' $S$ ' varying with respect to various Search Factor obtained from the Farmer Algorithm.



**Figure 3.** Flow Chart depicting the entire system.

## 3. Result and Analysis of Entire System

### 3.1 Testing the Farmer Module

#### CASE 1:

Location: Adoni Commodity: Cashew nuts.

Required Price: 1000 Offered Price: 900

As shown in the Figure 4

Analysis:

The demand factor for "Cashewnuts" = 1

Historical data factor =  $16/16 = 1$

The factor sum = 2. As per the algorithm, the radius of search should be 2000 km radius.

**Figure 4.** Farmer Module Case 1 query.

#### Result

All the locations mentioned above are within a 2000 km radius of the location "Adoni" and the prices are way above the farmer's required price. Irrespective of the transporta-

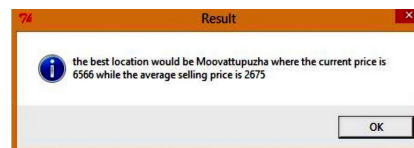
tion costs, he would definitely fetch a high value for his commodity.

By Farmers module (case1) result gives us the best location for sale Figure 5 and the alternative locations to sale is Figure 6.



**Figure 5.** Farmer Module Case 1 result for best location to sell.

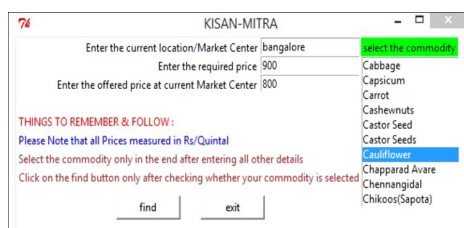
The search factor sum = 1.368. As per the algorithm, the radius of search should be 1200KM radius. The Figure 7 gives us the commodity cauliflower's best location and the average selling price. All the locations mentioned above are within a 1200 km radius of the location "Adoni" and the prices are way above the farmer's required price. Irrespective of the transportation costs, he would definitely fetch a high value for his commodity. The Figure 8 gives us the cauliflower alternative location for sale.



**Figure 7.** Farmer Module Case 2 result for best location to sell.



(a)



(b)

**Figure 6.** (a) Farmer Module Case 1 result for alternate locations to sell (b) Farmer Module Case 1 query.

#### CASE 2:

Location: Bangalore Commodity: Cauliflower.  
Record Price: 900 Offered Price: 800

The Figure 6.1 gives us the selection of commodity cauliflower.

#### Analysis

The Demand factor for cauliflower is = 1

Historical data factor =  $7/13 = 0.368$



**Figure 8.** Farmer Module Case 1 result for alternate locations to sell.

### 3.2 Testing the Buyer Module

The Buyer Module implements the Buyer algorithm. And we shall test the algorithm for various situations it could encounter.

#### CASE 1:

To add details about a new commodity if at that market center it doesn't exist. Let's say the market center is "Gooti". Let us check the market center at Gooti and the commodities available there, in MongoDB database currently Figure 9. Now when we enter details to the farmer Module about the need of a commodity currently unavailable at "Gooti", say rice Figure 10. Upon submission of the details, the below depicted dialog box pops up on the screen, confirming the addition of details to the database Figure 11. Now let us check whether the details have been actually added or not in the MongoDB database, by performing the query in MongoDB again as we had done

earlier. Figure 12 shows clearly that the entered details have been added to the MongoDB database successfully.

```

MongoDB shell version: 2.6.5
connecting to: test
> use project1
switched to db project1
> db.curpr.find(<<"Market Center": "Gooti">>).pretty()
2015-05-18T05:22:26.074+0530 TypeError: Property 'find' of object
is not a function
> db.curpr.find(<<"Market Center": "Gooti">>).pretty()
{
  "_id": ObjectId("554be49630254fc4f8b550bb"),
  "Market Center": "Gooti",
  "Commodity": "Paddy(Dhan)",
  "State": "Andhra Pradesh",
  "Loc": {
    "lat": 77.634,
    "lon": 15.12
  },
  "Unit of Arrivals": "Tonnes",
  "Variety": "Other",
  "Minimum Price": 1100,
  "Maximum Price": 1200,
  "Modal Price": 4312.5,
  "Unit of Price": "Rs/Quintal"
},
{
  "_id": ObjectId("554be49630254fc4f8b550bb"),
  "Market Center": "Gooti",
  "Commodity": "Groundnut",
  "State": "Andhra Pradesh",
  "Loc": {
    "lat": 77.634,
    "lon": 15.12
  },
  "Unit of Arrivals": "Tonnes",
  "Variety": "Local",
  "Minimum Price": 3400,
  "Maximum Price": 3500,
  "Modal Price": 1437.5,
  "Unit of Price": "Rs/Quintal"
}

```

**Figure 9.** Current Price database before the Buyer module execution.

**KISAN-MITRA**

Enter Your current location/Market Center: gooti

Enter the quantity of commodity you require: 450

Enter the price you are willing to offer: 3000

**THINGS TO REMEMBER & FOLLOW :**  
 Please Note that all Prices measured in Rs/Quintal  
 Select the commodity only in the end after entering all other details  
 Click on the find button only after checking whether your commodity is selected

done exit

Ready to Help You!!!

**Figure 10.** Buyer Module Query.

**Details Added!!!**

Your needs and entered details have been added successfully to the database

OK

**Figure 11.** Confirmation from Buyer Module that details added.

## CASE 2:

To replace the old buying price from the database of a commodity at a particular location, if the new buyer offers a higher price at the same market center and for the same commodity. Continuing with the previous case's example; now let us see what hap-

pens if we offer a higher price for the commodity "Rice" than what was previously offered at the location "Gooti"

Previously Offered: 3000

Newly Offered: 5000 shown in Figure 12.

```

15.12
{
  "Unit of Arrivals": "Tonnes",
  "Variety": "Local",
  "Minimum Price": 3400,
  "Maximum Price": 3500,
  "Modal Price": 1437.5,
  "Unit of Price": "Rs/Quintal"
},
{
  "_id": ObjectId("554be49630254fc4f8b550bb"),
  "Market Center": "Gooti",
  "Commodity": "Paddy(Dhan)",
  "State": "Andhra Pradesh",
  "Loc": {
    "lat": 77.634,
    "lon": 15.12
  },
  "Unit of Arrivals": "Tonnes",
  "Variety": "Other",
  "Minimum Price": 1100,
  "Maximum Price": 1200,
  "Modal Price": 4312.5,
  "Unit of Price": "Rs/Quintal"
},
{
  "_id": ObjectId("554be49630254fc4f8b550bb"),
  "Market Center": "Gooti",
  "Commodity": "Groundnut",
  "State": "Andhra Pradesh",
  "Loc": {
    "lat": 77.634,
    "lon": 15.12
  },
  "Unit of Arrivals": "Tonnes",
  "Variety": "Local",
  "Minimum Price": 3400,
  "Maximum Price": 3500,
  "Modal Price": 1437.5,
  "Unit of Price": "Rs/Quintal"
},
{
  "_id": ObjectId("554e9e8223f7ae15fc1856fd"),
  "Market Center": "Gooti",
  "Loc": {
    "lat": 77.634,
    "lon": 15.12
  },
  "Unit of Price": "Rs/Quintal",
  "Commodity": "Rice",
  "State": "Andhra Pradesh",
  "Modal Price": 3000
}

```

(a)

**KISAN-MITRA**

Enter Your current location/Market Center: gooti

Enter the quantity of commodity you require: 750

Enter the price you are willing to offer: 5000

**THINGS TO REMEMBER & FOLLOW :**  
 Please Note that all Prices measured in Rs/Quintal  
 Select the commodity only in the end after entering all other details  
 Click on the find button only after checking whether your commodity is selected

done exit

Ready to Help You!!!

(b)

**Figure 12.** (a) Current Price database after the Buyer Module execution (b) Buyer Module Query case 2.

Now let us check whether the new higher price has been successfully replaced the older and lower price in the MongoDB database, by performing the query in MongoDB again as we had done earlier. Figure 13, shows clearly that the entered new higher price has replaced the older lower successfully. The test is successful.

**CASE 3:**

To test whether the entered details are being saved in a separate database irrespective of the price offered is lowered/higher. The data entered was being saved in a collection named "Buyer". Let us see whether if that has been happening by querying in MongoDB. Figure 14, clearly shows that the details of entered, have been saved in the collection "Buyer" successfully. Hence the test is successful.

```

Select mongo.exe
> use Buyer
> find()
{
  "_id" : ObjectId("554e49638254fc4f8b558bc"),
  "Market Center" : "Gooti",
  "Commodity" : "Groundnut",
  "State" : "Andhra Pradesh",
  "Loc" : {
    "lat" : 15.12,
    "lon" : 77.634
  },
  "Unit of Price" : "Rs/Quintal",
  "Unit of Arrivals" : "Tonnes",
  "Variety" : "Local",
  "Minimum Price" : 3400,
  "Maximum Price" : 3500,
  "Modal Price" : 1432.5,
  "Unit of Price" : "Rs/Quintal"
},
{
  "_id" : ObjectId("554e49e8223f7ae15fc1856fd"),
  "Market Center" : "Gooti",
  "Loc" : {
    "lat" : 15.12,
    "lon" : 77.634
  },
  "Unit of Price" : "Rs/Quintal",
  "Commodity" : "Rice",
  "State" : "Andhra Pradesh",
  "Modal Price" : 3000
},
{
  "_id" : ObjectId("554e49a2823f7ae042c43c18d"),
  "Market Center" : "Gooti",
  "Commodity" : "Rice",
  "Unit of Price" : "Rs/Quintal",
  "Loc" : {
    "lat" : 15.12,
    "lon" : 77.634
  },
  "State" : "Andhra Pradesh",
  "Modal Price" : 4000
},
{
  "_id" : ObjectId("554e49a2a23f7ae1a5c071fd9"),
  "State" : "Andhra Pradesh",
  "Modal Price" : 5000,
  "Commodity" : "Rice",
  "Market Center" : "Gooti",
  "Loc" : {
    "lat" : 15.12,
    "lon" : 77.634
  },
  "Unit of Price" : "Rs/Quintal"
}

```

**Figure 13.** Current Price database after the Buyer Module execution Case 2.

```

Select mongo.exe
> use Buyer
> find()
{
  "_id" : ObjectId("554e49bba23f7ae1e80b798"),
  "Requirement" : 2000,
  "Market Center" : "Adoni",
  "Commodity" : "Rice",
  "Loc" : {
    "lat" : 15.63,
    "lon" : 77.28
  },
  "Modal Price" : 4000
},
{
  "_id" : ObjectId("554e49e8223f7ae15fc1856fd"),
  "Market Center" : "Gooti",
  "Loc" : {
    "lat" : 15.12,
    "lon" : 77.634
  },
  "Commodity" : "Rice",
  "Modal Price" : 3000,
  "Requirement" : 450
},
{
  "_id" : ObjectId("554e49a2823f7ae042c43c18d"),
  "Market Center" : "Gooti",
  "Loc" : {
    "lat" : 15.12,
    "lon" : 77.634
  },
  "Commodity" : "Rice",
  "Modal Price" : 4000
},
{
  "_id" : ObjectId("554e49a2a23f7ae1a5c071fd9"),
  "Modal Price" : 5000,
  "Requirement" : 750,
  "Commodity" : "Rice",
  "Market Center" : "Gooti",
  "Loc" : {
    "lat" : 15.12,
    "lon" : 77.634
  }
}

```

**Figure 14.** The Buyer Database where buyer details are stored.

**CASE 4**

To test whether the Demand values stored in the Demand collection in MongoDB, are changed depending on the quantity of commodities needed by the Buyer. For the case study, let us take "Black Gram Dal" as the commodity. Let us check the demand value of the commodity by querying in MongoDB Figure 15. Now, after the following query is performed using the farmer Module, let us check the demand value of the commodity by querying in MongoDB in again. As per the farmer algorithm, for the quantity entered by the Buyer in the module Figure 16 there should be an increase of "0.5". And as shown in Figure 17 the increase has taken place. The test is successful.

```

Select mongo.exe
> use Demand
> find()
{
  "_id" : ObjectId("554497ee9fc89bceba64405"),
  "Commodity" : "Ber (Zizyphus)",
  "Demand" : 0.074
},
{
  "_id" : ObjectId("554497ee9fc89bceba64406"),
  "Commodity" : "Betal Leaves",
  "Demand" : 2
},
{
  "_id" : ObjectId("554497ee9fc89bceba64407"),
  "Commodity" : "Bhindi(Ladies Finger)",
  "Demand" : 2.4
},
{
  "_id" : ObjectId("554497ee9fc89bceba64408"),
  "Commodity" : "Bitter gourd",
  "Demand" : 1.29
},
{
  "_id" : ObjectId("554497ee9fc89bceba64409"),
  "Commodity" : "Black Grams (Urd Beans)",
  "Demand" : -3.2
}
type "it" for more
>

```

**Figure 15.** Demand of Commodity before execution.

**KISAN-MITRA**

Enter Your current location/Market Center: salem

Enter the quantity of commodity you require: 1000

Enter the price you are willing to offer: 7000

**THINGS TO REMEMBER & FOLLOW:**

Please Note that all Prices measured in Rs/Quintal

Select the commodity only in the end after entering all other details

Click on the find button only after checking whether your commodity is selected

done exit

Ready to Help You!!!

**select the commodity**

- Betal Leaves
- Bhindi(Ladies Finger)
- Big Gram
- Bitter gourd
- Black Grams (Urd Beans)**
- Black pepper
- Bottle gourd
- Brinjal
- Cabbage
- Capsicum

**Figure 16.** Query in farmer Module Case 4.

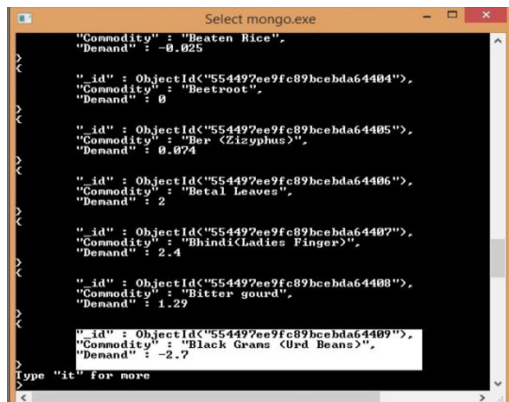


Figure 17. Demand of Commodity before execution.

## 4. Conclusion and Future Scope

The Modules for the Farmer and Buyer sides were made. The calculations for each of the Farmer Module and Buyer Module were outlined and executed effectively. The calculations were tried for different situations and the normal results were figured it out. Generally, the point and goals set toward the start of the undertaking have been acknowledged and the modules give the agriculturist the most ideal approach to offer his products. As a future improvement, the application could be moved to the cloud with any of the clustering algorithms<sup>11</sup>. The MongoDB server introduced in the cloud and the farmers and purchasers will be given separate records and login ID's on a devoted site and the site would be associated with the MongoDB in the cloud. By this procedure, the Middlemen element would be wiped out and agriculturists can offer their products straight forwardly to the dealer and get most extreme advantages, by getting to the proposed site. As the current information digging systems can't work for MapReduce programming in a disseminated environment for this, equal strategy for k-Anonymity<sup>12</sup> can be utilized. Agricultural Yield Analysis<sup>13</sup> gives a thought for a specific element offers great results and fills a decent need, then there can be various augmentations to the said substance with the advancement of time. In this world, there are not very many things that are superior to what serves the welfare of the farmers, all things considered; the extent of change of this task can be unquestionably upgraded advance more.

## 5. Tools Used

### 5.1 MongoDB

A MangoDB is one of many cross-platform document-oriented databases. It is categorized as a NoSQL database, MongoDB<sup>8</sup> discards the traditional table-based relational database structure in favour of JSON-like documents with dynamic schemas, which is referred to as BSON, making the integration of data in certain types of applications easier and faster. It is released under a combination of the GNU Affero General Public License and the Apache License, MongoDB is a free and open-source software. Due to the features and strengths of MongoDB has been adopted as backend software by a number of major organizations including Craigslist, eBay, Foursquare, Source Forge, Viacom, and the New York Times among others MongoDB is the most popular NoSQL database system.

Some of the key features of MongoDB are High Performance, High Availability, Automatic Scaling, Automatic Sharding, Geo-Spatial Calculations, etc. Let's focus more on the Geo-Spatial Features since it is a major part of the implementation. As mentioned earlier, one of the important features of MongoDB that has been utilized in this project is the ability to perform geo-spatial calculations. There are many forms of geo-spatial calculations possible in MongoDB, but the one that is used in this project is to find the locations that fall within a particular radius of another specific point. This is how it works:

- First of all the locations should have their location data included in every collection in a particular field. i.e. their latitude and longitude information.
- This particular field in which the location data is stored has to be indexed. Indexing means specifying what kind of a model, in this case it would be the GEO2D, upon which the locations would be mapped on a model like the one that resembles our Earth.
- Then after, when a query is received MongoDB would perform the calculations based on the model indexed and the query parameters. MongoDB uses the WGS84 geodetic datum model to perform the geo-spatial calculations for real world model involving latitudes and longitudes.

## 5.2 Python

Python<sup>14</sup> is a widely used general-purpose, high-level programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C++ or Java. The language provides constructs intended to enable clear programs on both a small and large scale. Python supports multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles. It features a dynamic type system and automatic memory management and has a large and comprehensive standard library. Python is used for interacting with MongoDB and thereby accessing and manipulating the data stored in MongoDB. We build both the algorithms mentioned above using Python. Python 3.3.2 is used in this project also the GUI for the Modules is built using Python, mainly its Tkinter widget.

## 6. References

1. A portal of the Government of India. Available from: [www.data.gov.in](http://www.data.gov.in). Date accessed: 31/03/2015.
2. The Department of Agricultural Research and Education (DARE) an entity of Government India.
3. A government of India portal for past prices datasets. Available from: [www.agmarket.nic.in](http://www.agmarket.nic.in). Date accessed: 25/08/2016
4. Mani G, Bari N, Liao D, Berkovich S. Organization of knowledge extraction from big data systems. 2014.
5. Bakshi K. Considerations for Big Data: Architecture and Approach Cisco Systems Inc, USA. 2013. p. 1-9.
6. Qiu XGJQ . Supporting queries and analyses of large-scale social media data with customizable and scalable indexing techniques over NoSQL Databases USA. 2014. p. 1-4.
7. Zheng Z, Zhu J, Michael M, Lyu L. Service-generated Big Data and Big Data-as-a- Service: An Overview Hong Kong, Hong Kong, China. 2013. p. 1-8.
8. Chana SA. A survey of clustering techniques for big data analysis. 2016; 9(3):1-12.
9. The Mongodb portal for all Mongodb related documentation. Available from: [www.mongodb.org](http://www.mongodb.org). Date accessed 16/08/2016.
10. For finding the relevant Map locations of market centers. [maps.google.com](http://maps.google.com). Date accessed 16/08/2016.
11. Sajana T, Rani CMS, Narayana KV. A survey on clustering techniques for big data Mining. Indian Journal of Science and Technology. Jan 2016; 9(3):1-12.
12. Radhika D, Kumari DA. A framework for exploring algorithms for big data mining. Indian Journal of Science and Technology. May 2016; 9(17):17.
13. Nagini S, Rajinikanth TV, Kiranmayee BV. Agriculture yield analysis using som classifier algorithm along with enhanced preprocessing technique. Indian Journal of Science and Technology. Jul 2016; 9(27):1-6.
14. The python language official documentation and libraries. Available from: [www.python.org](http://www.python.org) Date accessed: 16/08/2016.