**A PRELIMENERY REPORT ON**

**“****COMMODITY STOCK RECOMMENDATIONS AND PRICE PREDICTION USING PRESCRIPTIVE ANALYTICS TECHNIQUES”**

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE

IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE

OF

**BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)**

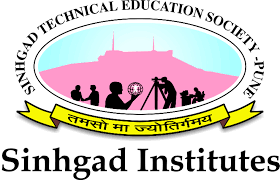
##### SUBMITTED BY

**STUDENT NAME Exam No: B210808**

**STUDENT NAME Exam No: B210808**

**STUDENT NAME Exam No: B210808**

**STUDENT NAME Exam No: B210808**



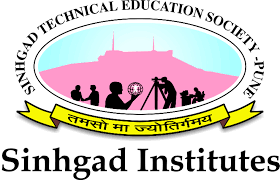
## DEPARTMENT OF COMPUTER ENGINEERING

**STES’S SMT. KASHIBAI NAVALE COLLEGE OF ENGINEERING**

**VADGAON BK, OFF SINHGAD ROAD, PUNE 411041**

**SAVITRIBAI PHULE PUNE UNIVERSITY**

**2020-21**



**CERTIFICATE**

This is to certify that the project report entitles

**“COMMODITY STOCK RECOMMENDATIONS AND PRICE PREDICTION USING PRESCRIPTIVE ANALYTICS TECHNIQUES”**

Submitted by

**STUDENT NAME Exam No: B210808**

**STUDENT NAME Exam No: B210808**

**STUDENT NAME Exam No: B210808**

**STUDENT NAME Exam No: B210808**

are bonafide students of this institute and the work has been carried out by them under the supervision of **Prof. A. B. C.**  and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of **Bachelor of Engineering** (Computer Engineering).

**(Prof. .......................)** (**Dr. P. N. Mahalle)**

Guide Head,

Department of Computer Engineering Department of Computer Engineering

**(Dr. A. V. Deshpande)**

Principal,

Smt. Kashibai Navale College of Engineering Pune – 41

Place: Pune

Date:

**ACKNOWLEDGEMENT**

It gives us great pleasure in presenting the preliminary project report on ‘COMMODITY STOCK RECOMMENDATIONS AND PRICE PREDICTION USING PRESCRIPTIVE ANALYTICS TECHNIQUES’.

We would like to take this opportunity to thank our project guide Prof. A. B. C. for giving us all the help and guidance we needed. We are really grateful to him/her for kind support.

In particular we indebted to Prof. A. B. C. who had a faith in this idea, believed in our ability, whispered the words of encouragement and made helpful suggestions from time to time. We would forever remain indebted to him/her.

Student Name

Student Name

Student Name

Student Name

(BE Computer Engineering)

**ABSTRACT**

Prescriptive Analytics is a method that helps businesses to make better decisions by analyzing raw data. Historic raw data can be analyzed using data mining algorithms to get knowledge from that data. Using this knowledge, we can make commodity stock recommendations and price predictions. But fluctuations in these predictions affect the global economic activity. They have a major impact on overall performance in the agricultural field. So, we have proposed an efficient method for commodity stock and price predictions to improve product sales. In order to forecast commodity stock and price, we have used different data mining algorithms like Decision Tree, K-Nearest Neighbor, XGBoost, Logistic Regression, Support Vector Machine, and Naive Bayes. Further, we have compared all these algorithms based on performance metrics such as accuracy, precision, recall, F1 score to get the best performing algorithm and provide predictions according to that algorithm. This provides more accurate predictions of commodity stock and price.

**TABLE OF CONTENTS**

LIST OF ABBREVATIONS viii

LIST OF FIGURES ix

LIST OF TABLES x

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CHAPTER** | | | | **TITLE** | | **PAGE NO** |
| **01** | | | | **Introduction** | | 1 |
|  | 1.1 | | | Overview | | 1 |
|  | 1.2 | | | Motivation | | 1 |
|  | 1.3 | | | Problem Definition and Objectives | | 2 |
|  | 1.4 | | | Project Scope & Limitations | | 2 |
|  | 1.5 | | | Methodologies of Problem solving | | 3 |
| **02** | | | | Literature Survey | | 4 |
| **03** | | | | Software Requirements Specification | | 7 |
|  | | 3.1 | | Assumptions and Dependencies | | 7 |
|  | | 3.2 | | Functional Requirements | | 7 |
|  | |  | | 3.2.1 | Price Prediction & Stock Recommendations | 7 |
|  | | 3.3 | | External Interface Requirements | | 8 |
|  | |  | | 3.3.1 | User Interfaces | 8 |
|  | |  | | 3.3.2 | Hardware Interfaces | 9 |
|  | |  | | 3.3.3 | Software Interfaces | 9 |
|  | | 3.4 | | Nonfunctional Requirements | | 10 |
|  | |  | | 3.4.1 | Performance Requirements | 10 |
|  | |  | | 3.4.2 | Safety Requirements | 10 |
|  | |  | | 3.4.3 | Security Requirements | 10 |
|  | |  | | 3.4.4 | Software Quality Attributes | 11 |
|  | | 3.5 | | System Requirements | | 11 |
|  | |  | | 3.5.1 | Database Requirements | 11 |
|  | |  | | 3.5.2 | Software Requirements (Platform Choice) | 11 |
|  | |  | | 3.5.3 | Hardware Requirements | 12 |
|  | | 3.6 | | Analysis Models: Waterfall Model | | 12 |
| **04** | | | | System Design | | 14 |
|  | | 4.1 | | System Architecture | | 14 |
|  | | 4.2 | | Mathematical Model | | 15 |
|  | | 4.3 | | Data Flow Diagrams | | 18 |
|  | | 4.4 | | Entity Relationship Diagrams | | 19 |
|  | | 4.5 | | UML Diagrams | | 20 |
| **05** | | | | Project Plan | | 24 |
|  | | | 5.1 | Project Estimate | | 24 |
|  | | |  | 5.1.1 | Reconciled Estimates | 24 |
|  | | |  | 5.1.2 | Project Resources | 24 |
|  | | | 5.2 | Risk Management | | 25 |
|  | | |  | 5.2.1 | Risk Identification | 25 |
|  | | |  | 5.2.2 | Risk Analysis | 25 |
|  | | |  | 5.2.3 | Overview of Risk Mitigation, Monitoring, Management | 26 |
|  | | | 5.3 | Project Schedule | | 28 |
|  | | |  | 5.3.1 | Project Task Set | 28 |
|  | | |  | 5.3.2 | Task Network | 29 |
|  | | |  | 5.3.3 | Timeline Chart | 30 |
|  | | | 5.4 | Team Organization | | 31 |
|  | | |  | 5.4.1 | Team structure | 31 |
|  | | |  | 5.4.2 | Management reporting and communication | 31 |
| **06** | | | | Project Implementation | | 32 |
|  | | | 6.1 | Overview of Project Modules | | 34 |
|  | | | 6.2 | Tools and Technologies Used | | 34 |
|  | | | 6.3 | Algorithm Details | | 34 |
|  | | |  | 6.3.1 | Decision Tree | 34 |
|  | | |  | 6.3.2 | K-Nearest Neighbor (KNN) | 36 |
|  | | |  | 6.3.3 | XGBoost | 36 |
|  | | |  | 6.3.4 | Logistic Regression | 36 |
|  | | |  | 6.3.5 | Support Vector Machine (SVM) | 37 |
|  | | |  | 6.3.6 | Naïve Bayes | 37 |
|  | | | 6.4 | Evaluation of Algorithms | | 38 |
| **07** | | | | Software Testing | | 40 |
|  | | | 7.1 | Type of Testing | | 40 |
|  | | | 7.2 | Test cases & Test Results | | 40 |
| **08** | | | | Results | | 42 |
|  | | | 8.1 | Outcomes | | 42 |
|  | | | 8.2 | Screen Shots | | 42 |
| **09** | | | | Conclusions | | 44 |
|  | | 9.1 | | Conclusions | | 44 |
|  | | 9.2 | | Future Work | | 44 |
|  | | 9.3 | | Applications | | 44 |
|  | | | | **Appendix A**: Problem statement feasibility assessment using, satisfiability analysis and NP Hard, NP-Complete or P type using modern algebra and relevant mathematical models.  **Appendix B**: Details of paper publication: name of the conference/journal, comments of reviewers, certificate, paper.  **Appendix C**: Plagiarism Report of project report. | | 45  48  49 |
|  | | | | **References**  Thomas Noltey, Hans Hanssony, Lucia Lo Belloz,”Communication Buses for Automotive Applications” In *Proceedings of the* 3rd *Information Survivability Workshop (ISW-2007)*, Boston, Massachusetts, USA, October 2007. IEEE Computer Society. | | 50 |

**LIST OF ABBREVATIONS**

|  |  |
| --- | --- |
| **Abbreviation** | **Illustration** |
| KNN | K-Nearest Neighbor |
| SVM | Support Vector Machine |
| XGBoost | Extreme Gradient Boosting |
| BI | Business Intelligence |
| TP | True Positive |
| TN | True Negative |
| FP | False Positive |
| FN | False Negative |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure** | **Illustration** | **Page No.** |
| 3.3.1 | User Interface | 9 |
| 3.6.1 | Waterfall Model | 13 |
| 4.1 | System Architecture | 14 |
| 4.2 | P, NP, NP-Hard, NP-Complete Classes | 17 |
| 4.3 | Data Flow Diagram | 18 |
| 4.4 | ER Diagram | 19 |
| 4.5.1 | Use Case Diagram | 20 |
| 4.5.2 | Sequence Diagram | 21 |
| 4.5.3 | Activity Diagram | 22 |
| 4.5.4 | Class Diagram | 23 |
| 5.3.2 | Task Network | 29 |
| 5.3.3 | Timeline Chart | 30 |
| 6.4.1 | Confusion Matrix | 38 |
| 8.2.1 | Evaluation of Algorithms for Soya Oil | 42 |
| 8.2.2 | Before selecting commodity price and performance metric | 43 |
| 8.2.3 | After selecting commodity price and performance metric | 43 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table** | **Illustration** | **Page No.** |
| 2.1 | Literature Survey | 5 |
| 5.2.1 | Risk Table | 25 |
| 5.2.3.1 | Risk Probability Definition | 26 |
| 5.2.3.2 | Risk Impact Definition | 26 |
| 5.2.3.3 | Risk Description for risk 1 | 27 |
| 5.2.3.4 | Risk Description for risk 2 | 27 |

1. **INTRODUCTION**
   1. **OVERVIEW**

The economical growth of farmers is deteriorating continuously. Because of this, there is an increase in disinterest in farm and farm investments. Current systems available for commodity price prediction do not give the expected result. Their accuracy is low as different algorithms may give fluctuating results. So, there is a need for a system that can provide the best outcomes for each commodity.

We have built a system that uses various prescriptive analytics algorithms like Decision Tree, K-Nearest Neighbor (KNN), XGBoost, Logistic Regression, Support Vector Machine (SVM), and Naive Bayes to predict commodity price and recommend the stock. This system takes daily market price and weather data as input and then compares all these algorithms based on performance metrics to get the best algorithm as output. More accurate predictions are made according to that best algorithm.

* 1. **MOTIVATION**

India is the land of agriculture and as per the 2010–11 census report, about 70% of its population depends on agriculture for their livelihood. Presently, because of the unaffordability of proper facilities, farmers sell their agricultural produce at a cheap price to intermediates in the village. Further, intermediates sell this produce to market trader in nearby city at higher prices. Due to a lack of market knowledge, farmers are unaware of actual prices of their produce that are prevalent in the market. They cannot make wise decisions such as what price will give them higher profit, where to sell and when to sell. Intermediates take advantage of farmer’s ignorance, helplessness and offer them a cheap price for their produce. Due to this, the economic growth of Indian farmers is deteriorating with growing disinterest in farm and farm investments. Hence, business intelligence (BI) will help farmers to make wise decisions and improve their financial condition.

Business Intelligence (BI) is nothing but a process that includes collecting agricultural data, preprocess it, extract useful insights from it. Prescriptive Analytics is a common function of BI technologies. Using prescriptive analytics techniques, farmers can take decisions such as which crop should be selected, whether to sell farm produce now or later, what will be the price of a particular crop in the future, how much stock should be produced. Prescriptive analytics help farmers by predicting the price of crops and recommending the stock of crops to be produced.

* 1. **PROBLEM DEFINITION AND OBJECTIVES**

Build a system that can predict commodity prices and stocks so that farmers can easily take appropriate decisions regarding their agricultural produce.

The main goal is to use prescriptive analytics techniques such as Decision Tree, K-Nearest Neighbor (KNN), XGBoost, Logistic Regression, Support Vector Machine (SVM), and Naive Bayes to predict commodity price and recommend the stock.

* 1. **PROJECT SCOPE AND LIMITATIONS**

There is a need for a system that can help farmers to take decisions such as which crop should be selected, whether to sell farm produce now or later, what will be the price of a particular crop in the future, how much stock should be produced. Using such systems, farmers can improve their economical growth. This system has great scope in the agricultural field and mostly for farmers.

The limitation to this system is we require daily price data for each commodity. A large volume of data requires a system with high configurations.

* 1. **METHODOLOGIES OF PROBLEM SOLVING**

The purpose of this system is to predict commodity prices and recommend the stock. For this we have used the following methodologies:

1. We conducted a literature survey and found that various algorithms were used for commodity price predictions.
2. We studied algorithms that can be used for commodity price predictions and stock recommendations.
3. We use them together and evaluate them using some performance metrics.
4. Based on these metrics, we got the best algorithm which we then used for price prediction and stock recommendations for selected commodity, city and year.
5. **LITERATURE SURVEY**

Commodity price prediction is very important for farmers. They should be aware of these predictions so that they can make wise decisions to improve their financial condition using Market Intelligence. The Auto Regressive Integrated Moving Average (ARIMA) forecasting technique and Recurrent Neural Network (RNN) are the deep learning technique which can be used for short term and long-term commodity price predictions respectively [1]. The daily market price data and weather information are required to make agricultural commodity price predictions. The agriculture-related data and weather information are time series in nature. Time series data is a sequence of well-defined data points measured at a certain interval of time. The deep learning techniques are suitable for prediction based on these time series data. These techniques discover patterns in time series data and extrapolate them into future values of time series. Various studies give applications of deep learning [2] and forecasting [3] of agricultural produce price.

In data mining, supervised and unsupervised learning techniques can be used to extract useful information from bulky data set in reasonable format [4]. Effective business decision-making can be done using appropriate sales prediction techniques, which are discussed here [5]. For business organizations, an intelligent sales prediction system is required to handle a large volume of data. Business decisions are based on the accuracy of algorithms used in intelligent sales predictions. Various data mining techniques are used to produce models that are comprehensive and reliable [6].

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No.** | **Title** | **Author** | **Year** | **Description** |
| 1 | Market Intelligence for Agricultural Commodities Using Forecasting and Deep Learning Techniques | S. Shrivastava, S. N. Pal, and R. Walia | 2019 | Use of Market Intelligence. For commodity price predictions, they used ARIMA and RNN techniques. |
| 2 | Business Analytics: The Science of Data-Driven Decision Making | Kumar U. D. | 2017 | Various applications of deep learning techniques were discussed. |
| 3 | Data Mining and Predictive Analytics | Larose D. T., Larose C. D. | 2017 | Various deep learning techniques were introduced for price prediction. |
| 4 | Review paper on clustering techniques | Mann A. K., and Kaur N. | 2013 | Supervised and Unsupervised learning techniques were used to extract useful information from bulky data set in reasonable format. |
| 5 | Sales Prediction Using Effective Mining Techniques | Shah N., Solanki M., Tambe A., and Dhangar D | 2015 | Discussed about effective business decision-making that can be done using appropriate sales prediction techniques |
| 6 | Intelligent Sales Prediction Using Machine Learning Techniques | S. Cheriyan, S. Ibrahim, S. Mohanan, S. Treesa | 2018 | Various data mining techniques like Generalized Linear Model (GLM), Decision Tree (DT) and Gradient Boost Tree (GBT) were used for price prediction. |

**Table 2.1 Literature Survey**

1. **SOFTWARE REQUIREMENT SPECIFICATION.**
   1. **ASSUMPTIONS AND DEPENDENCIES**

We developed this system in Python and therefore it requires Python3 to be installed on the user’s system. This applies to Windows, Mac OS, and Linux users. Sklearn is the Machine Learning library require for this project. From this library, various algorithms are used in this project for training, modeling, and predicting price. We have prepared a user interface using the Tkinter library. We require visual studio code to run this system.

* 1. **FUNCTIONAL REQUIREMENTS**
     1. **Price Prediction & Stock Recommendations**
        1. Description & Priority

This system uses various data mining algorithms such as Decision Tree, K-Nearest Neighbor (KNN), XGBoost, Logistic Regression, Support Vector Machine (SVM), and Naive Bayes for modeling. We evaluated these models based on some performance metrics such as accuracy, precision, recall, and F1 score. An algorithm with the highest accuracy is considered for further predictions. All these algorithms were implemented using Sklearn python library.

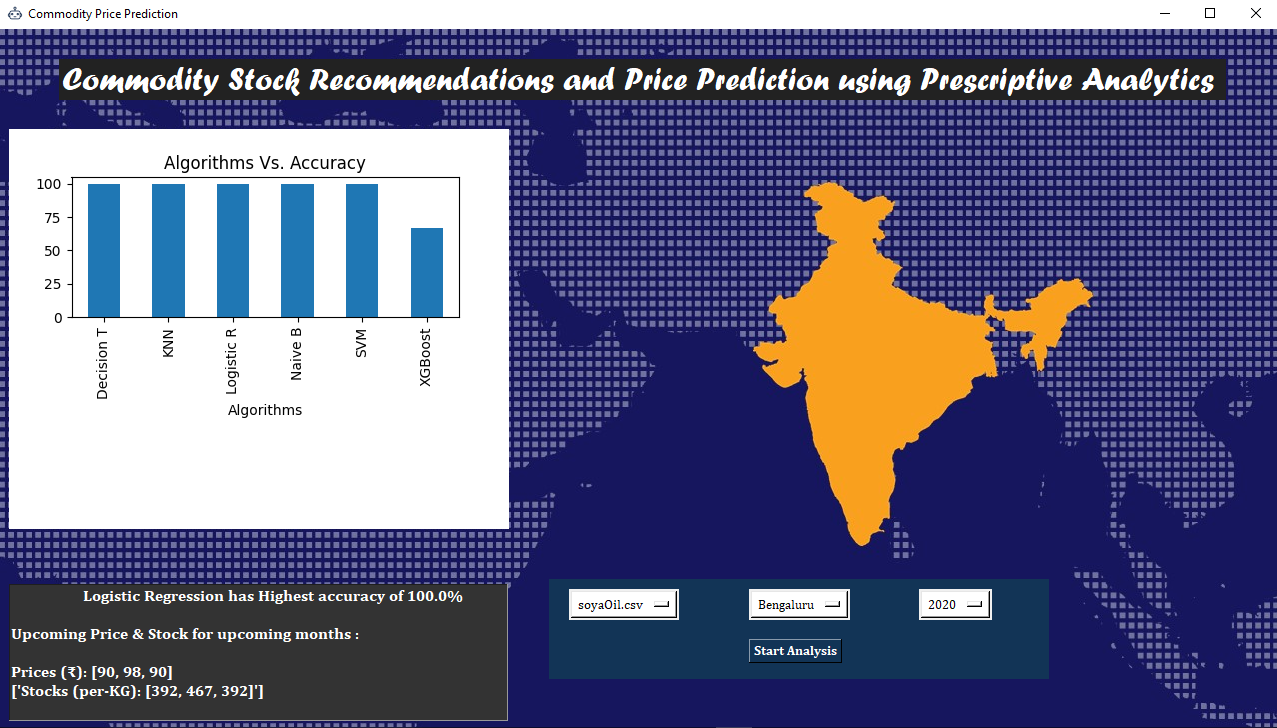
* + - 1. Stimulus/ Response Sequences

When the user runs the system, a user interface will be shown to user. Here, we have to select the commodity type and performance metrics using which we have to evaluate the algorithms. After that system will train the model using the csv file of selected commodity and evaluate using the selected performance metric. Graph of evaluation will be shown to the user. The algorithm having the result will be selected for price predictions. These predictions are displayed to the user.

* + - 1. Functional Requirements
* REQ-1: The user needs .csv files for each commodity.
  1. **EXTERNAL INTERFACE REQUIREMENTS**
     1. **User Interface**

The user interface for this system is as shown in Fig. 3.3.1. For the user interface, we have used the tkinter library of python. This library is fast and easy to use for creating user interface. User Interface is the efficient way using which users can communicate with this system, visualize the evaluation results and get commodity price prediction values from this system.

The user interface shown in Fig. 3.3.1 appears when we start the system. Here, we have to select the commodity and performance metrics using which we have to evaluate the algorithms. After that system will train the model using the csv file of selected commodity and evaluate using the selected performance metric. Graph of evaluation will be shown as in Fig. 3.3.1. The algorithm having the result will be selected for price predictions. These predictions are displayed to the user as shown in Fig. 3.3.1.



**Fig. 3.3.1 User Interface**

* + 1. **Hardware Interface**

The minimum hardware requirements for this project are 8 GB RAM or more, 500 GB HDD, intel processor core i3 or more.

* + 1. **Software Interface**

This project requires Python to be installed on the system, more specifically Python version 3 for its latest release. This system requires following python libraries:

* Sklearn
* Tkinter
* Seaborn
* Xgboost
* Numpy
* Pandas
* Matplotlib
* os
  1. **NON-FUNCTIONAL REQUIREMENTS**
     1. **Performance Requirements**

The hardware requirements of this system are a minimum of 8 GB RAM and an Intel processor i3 or more. The response time of this system depends on the size of the data set. The data set file must be in .csv format only. The larger the dataset, the larger is the time required to train the model. The performance of the system can be evaluated using performance metrics such as accuracy, precision, recall, and F1 score.

* + 1. **Safety Requirements**

There are no safety requirements needed for this system.

* + 1. **Security Requirements**

There are no security requirements for this system. Anyone can view the results of the system and take advantage form these price predictions and stock recommendations.

* + 1. **Software Quality Attributes**

This system is used to predict commodity prices and recommend the stock. Because of its well-designed and easy-to-use interface, it can be used by both experts and typical users. It is very easy to view price predictions and stock recommendations for any particular selected commodity. The corresponding graph and prices will be shown to the user using a user interface. Users can select performance metrics using which model evaluation will be performed. We use one best algorithm having the highest value for selected performance metrics among all the other algorithms. This gives more accurate predictions.

* 1. **SYSTEM REQUIREMENTS**
     1. **Database Requirements**

We are not storing data on any of the clouds. We require data sets for commodities which were locally stored on the device. There is no need for a database.

* + 1. **Software Requirements (Platform Choice)**

This project requires Python to be installed on the system, more specifically Python version 3 for its latest release. This system requires following python libraries:

* Sklearn
* Tkinter
* Seaborn
* Xgboost
* Numpy
* Pandas
* Matplotlib
* os

Also, we require visual studio code to run this project.

* + 1. **Hardware Requirements**

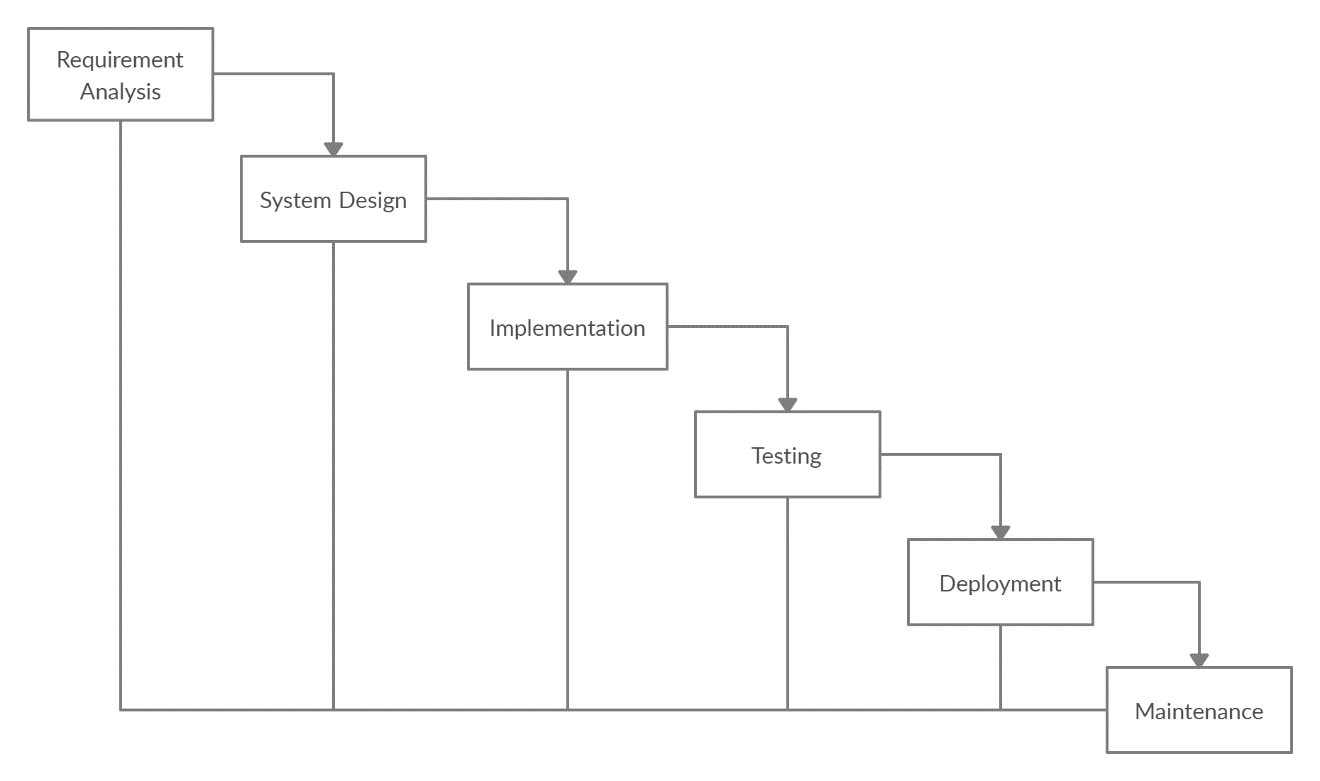
The minimum hardware requirements for this project are 8 GB RAM or more, 500 GB HDD or more, intel processor core i3 or more.

* 1. **ANALYSIS MODELS: WATERFALL MODEL**

We are using Waterfall model of the Software Development Life Cycle. We are using this model because of the following reasons:

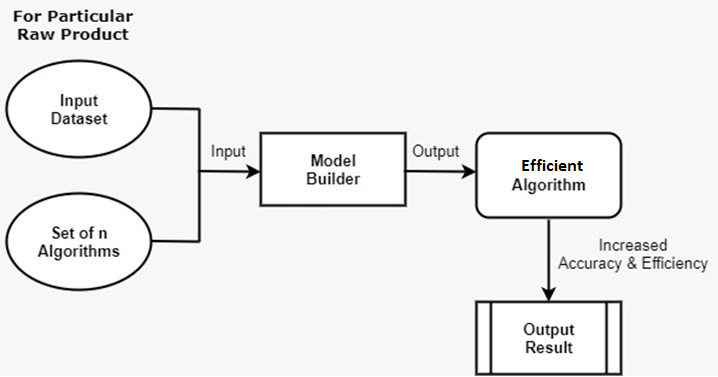
* Requirements are clear and fixed.
* Product definition is stable.
* Technology is understood and is not dynamic.
* There are no ambiguous requirements.

Our project’s requirements are fixed that we have to use Decision tree, K-Nearest Neighbor (KNN), XGBoost, Logistic Regression, Support Vector Machine (SVM), and Naive Bayes to train the model and predict the prices and recommend the stock. So, this is the most suitable SDLC model for our project.



**Fig. 3.6.1 Waterfall Model.**

1. **SYSTEM DESIGN**
   1. **SYSTEM ARCHITECURE**

****

**Fig. 4.1 System Architecture**

The above diagram shows the system architecture for our system. The inputs to this system are daily market price and a set of n algorithms. After model building, the efficient algorithm can be decided using performance metrics such as accuracy, precision, and recall. When we got an efficient algorithm, commodity price prediction and stock recommendations are done. All this is done for a single commodity. For another commodity, we have to repeat this process.

* 1. **MATHEMATICAL MODEL**

Let P1 = Given success cases viz., Predicted prices are correct, predicted stock values are correct, and evaluation gives correct efficient algorithm.

For a Problem P1 to be NP-Hard, Satisfiability problem (SAT) must be reducible to P1;

i.e. SAT ∝ P1;

Let for CNF-SAT,

CNF =

X1: True (i.e.1) if predicted prices are correct

X2: True (i.e.1) if predicted stock values are correct

X3: True (i.e.1) if evaluation gives correct efficient algorithm

Here, there are 8 possibilities for which CNF will be satisfied. They are:

|  |  |  |
| --- | --- | --- |
| **X1** | **X2** | **X3** |
| 0 | 0 | 0 |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |
| 1 | 1 | 1 |

This means, we have 8 possibilities for 3 variables. We can check whether CNF is true or not for these 8 possibilities.

i.e. For n variables, we have 2n possibilities.

This tells us that our problem P1 is taking exponential time. This is similar to satisfiability problem as it is also an exponential time taking algorithm. As satisfiability problem is reducible to our problem P1, problem P1 is NP-Hard.

We will devise non-deterministic algorithm for problem P1:

Algosat ()

{

for i= 1 to 8

{

Xi=Choice (true, false) -------- Non- deterministic statement

if (CNF == true)

Success () -------- Non- deterministic statement

else

Failure () -------- Non- deterministic statement

}

}

This algorithm will take O(1) time i.e. polynomial time if we know the correct values for Xi.

So, this is a non-deterministic algorithm that takes polynomial time O(1).

We have proved that P1 is NP- Hard. Also, we have written non-deterministic polynomial time taking algorithm for P1. So, our problem P1 is NP-Complete.

**NP**

Have non-deterministic algorithm

**NP – Hard**

SAT ∝ P1

**SAT**

**P**

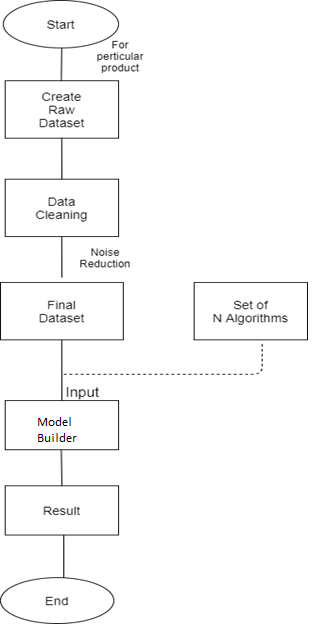
**NP Complete**

**Fig. 4.2 P, NP, NP-Hard, NP-Complete Classes**

Now, to prove that this non-deterministic algorithm will be solve in future, we have to prove that P = NP.

According to Cook’s theorem, satisfiability problem is in P if P = NP.

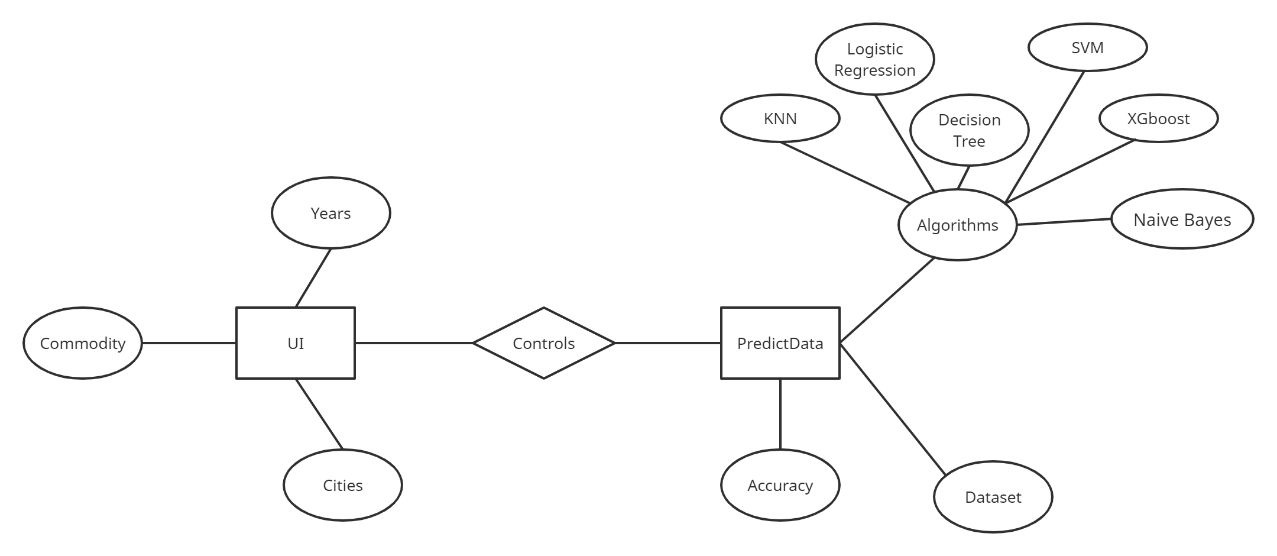
* 1. **DATA FLOW DIAGRAMS**

****

**Fig. 4.3 Data Flow Diagram**

The above figure 4.3 shows the data flow diagram of our system. It represents the main functions in the process of agricultural commodity price predictions and stock value recommendations.

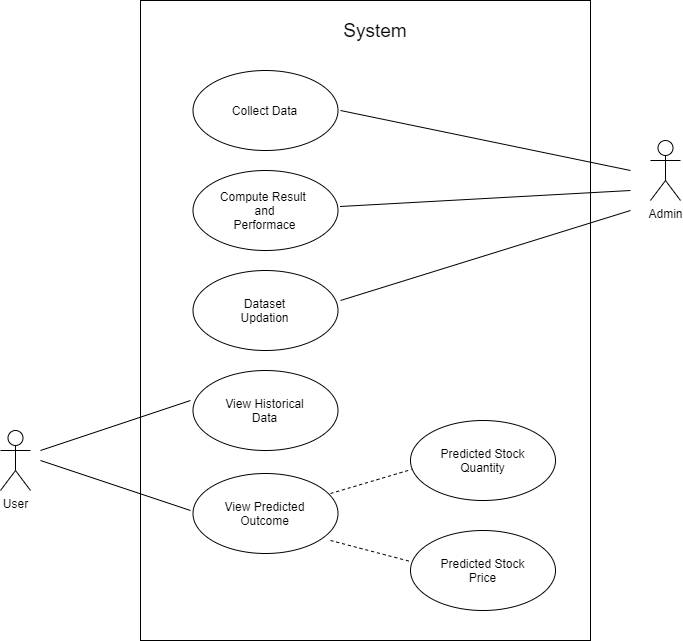
* 1. **ENTITY RELATIONSHIPS DIAGRAMS**



**Fig. 4.4 ER Diagram**

The above diagram Fig 4.4 shows the Entity-Relationship diagram of our system. Our system hastwo4 main entities: UI and PredictData. UI controls PredictData. UI is a class that is related to the user interface. PredictData class handles modeling using various algorithms.

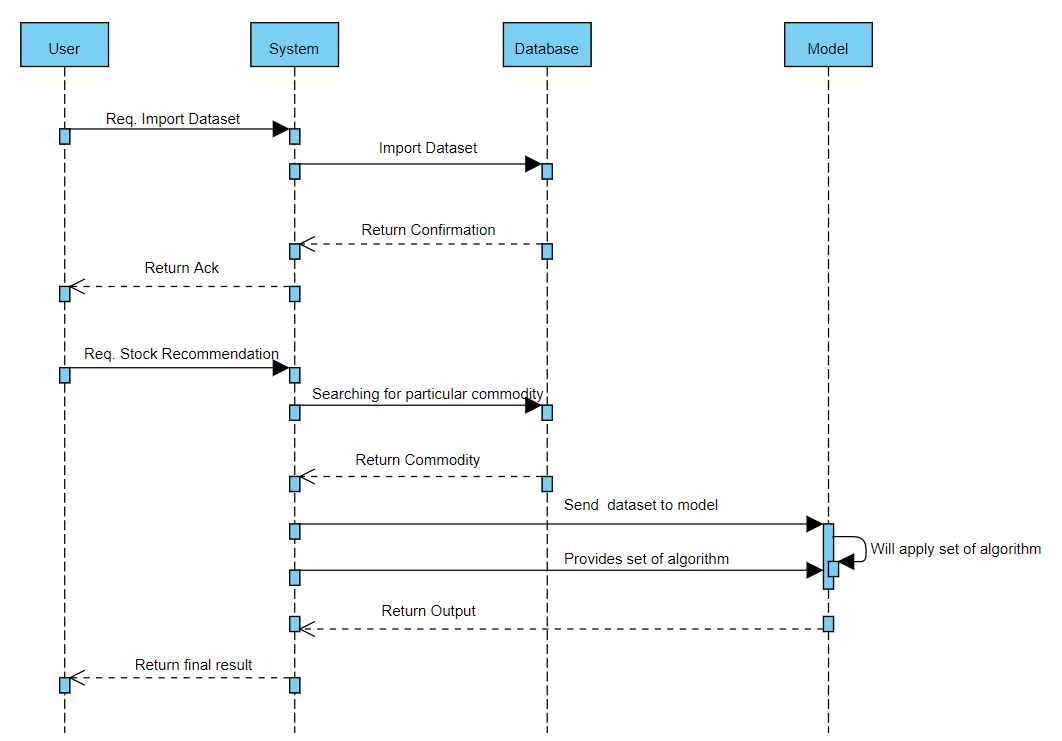
* 1. **UML DIAGRAMS**
     1. **Use case Diagram**

****

**Fig. 4.5.1 Use case Diagram**

The above use case diagram shows that our system has two key actors: User, and Admin. This system shows the functions to be performed by each actor. Admin has to perform data collection and model preparation. Users can view the results including predicted price and predicted stock quantity.

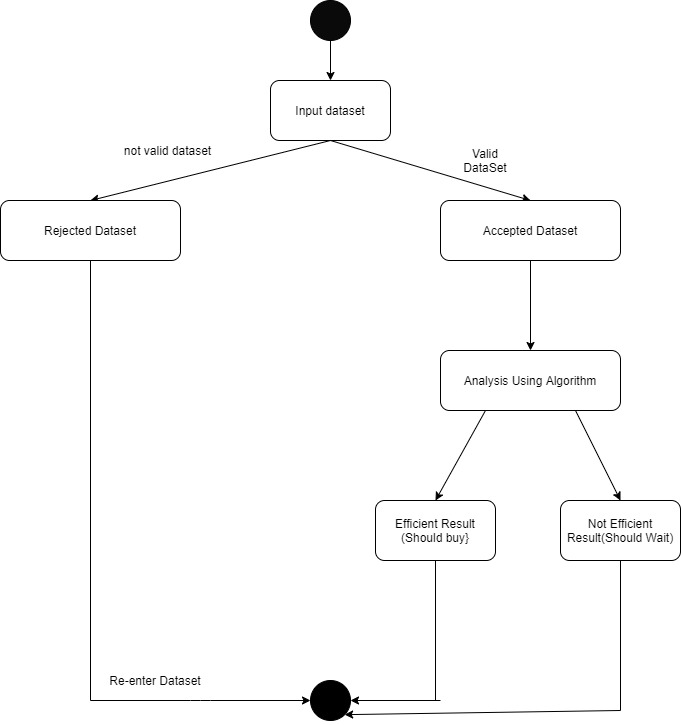
* + 1. **Sequence Diagram**

****

**Fig. 4.5.2 Sequence Diagram**

This is the sequence diagram for our system. It shows the steps in which users act and get the required results. The diagram shows that the user will create data set. Using this dataset, he can request price predictions.

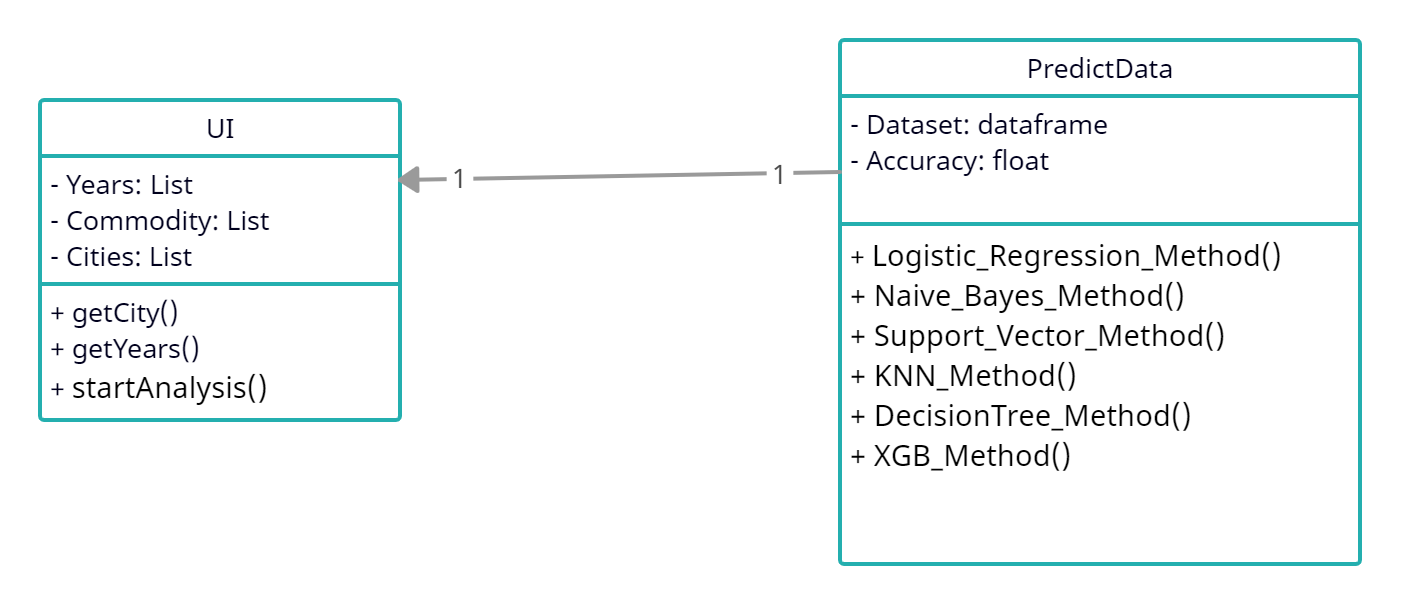
* + 1. **Activity Diagram**



**Fig. 4.5.3 Activity Diagram**

The above diagram shows the flow of different activities in our system. It shows that after uploading the dataset, it checks for a valid dataset. If the dataset is valid, then different algorithms are applied to it to get an efficient algorithm.

* + 1. **Class Diagram**



**Fig. 4.5.4 Class Diagram**

The above diagram shows the class diagram of our system. Our system has two main classes: UI and PredictData. UI handles user interface and PredictData handles actual predictions using various algorithms.

1. **PROJECT PLAN**
   1. **PROJECT ESTIMATE**
      1. **Reconciled Estimates**

Python is used to develop this system. So, it doesn’t require any type of money.

* + 1. **Project Resources**

We are four group members who worked on this project. We have used Python to implement this project. So, this system requires a laptop with Python 3 installed on it. After that, it requires visual studio code to run the python files. This system requires following python libraries:

* Sklearn
* Tkinter
* Seaborn
* Xgboost
* Numpy
* Pandas
* Matplotlib
* Os

The minimum hardware requirements for this project are 8 GB RAM or more, 500 GB HDD or more, intel processor core i3 or more.

The daily price dataset required for this system are obtained from the Open Government Data platform.

* 1. **RISK MANAGEMENT**
     1. **Risk Identification**

We asked some questionnaire for risk identification that revealed some risks. These are categorized as per their categories as shown in table 5.2.1. The list questionnaire is as follows:

1. Are requirements are fully understood by the software engineering team and its customers?
2. Do end-users have realistic expectations?
3. Are project requirements stable?
4. Is the number of people on the project team adequate to do the job?
5. Do all customer/user constituencies agree on the importance of the project and on the requirements for the system/product to be built?
   * 1. **Risk Analysis**

The risks for the Project can be analyzed within the constraints of time and

quality

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Risk Description** | **Probability** | **Impact** | | |
| **Schedule** | **Quality** | **Overall** |
| 1 | Response Time increases if data set is large | Low | Low | High | High |
| 2 | User is unsatisfied with predicted values | Low | Low | High | High |

**Table 5.2.1 Risk table**

* + 1. **Overview of Risk Mitigation, Monitoring, Management**

Following are the details for each risk.

|  |  |  |
| --- | --- | --- |
| **Probability** | **Value** | **Description** |
| High | Probability of occurrence is | > 75% |
| Medium | Probability of occurrence is | 26 – 75% |
| Low | Probability of occurrence is | < 25% |

**Table 5.2.3.1 Risk Probability Definition**

|  |  |  |
| --- | --- | --- |
| **Impact** | **Value** | **Description** |
| Very High | > 10% | Schedule impact or Unacceptable quality |
| High | 5 – 10% | Schedule impact or Some parts of the project have low quality |
| Medium | < 5% | Schedule impact or Barely noticeable degradation in quality Low Impact on schedule or Quality can be incorporated |

**Table 5.2.3.2 Risk Impact Definition**

|  |  |
| --- | --- |
| Risk ID | 1 |
| Risk Description | Response Time increases if data set is large |
| Category | Requirements |
| Source | Software requirement Specication document |
| Probability | Low |
| Impact | High |
| Response | Accept |
| Strategy | Dataset with limited observations should be used It should be in some thousands only. |
| Risk Status | Occurred |

**Table 5.2.3.3 Risk Description for risk 1**

|  |  |
| --- | --- |
| Risk ID | 2 |
| Risk Description | User is unsatisfied with predicted values |
| Category | Testing |
| Source | Software Testing document |
| Probability | Low |
| Impact | High |
| Response | Mitigate |
| Strategy | Apply changes to satisfy user requirements. |
| Risk Status | Identified |

**Table 5.2.3.4 Risk Description for risk 2**

* 1. **PROJECT SCHEDULE**
     1. **Project Task Set**

Major tasks in Project stages are:

* Literature Survey:

Various research papers and survey papers are studied to know about previous implementations in the field of Predictions. Identify their advantages and disadvantages.

* Planning:

After literature survey, we decided to apply Decision Tree, K-Nearest Neighbor (KNN), XGBoost, Logistic Regression, Support Vector Machine (SVM), and Naive Bayes algorithms for price prediction. We decided t apply these predictions in agricultural field.

* Requirement Analysis:

We gather some datasets of daily market price of agricultural commodities. We studied how to perform model evaluation.

* Design:

We decided the steps using which actual coding will be done. We decided to perform data preprocessing before model training. After that we decided the outline for user interface.

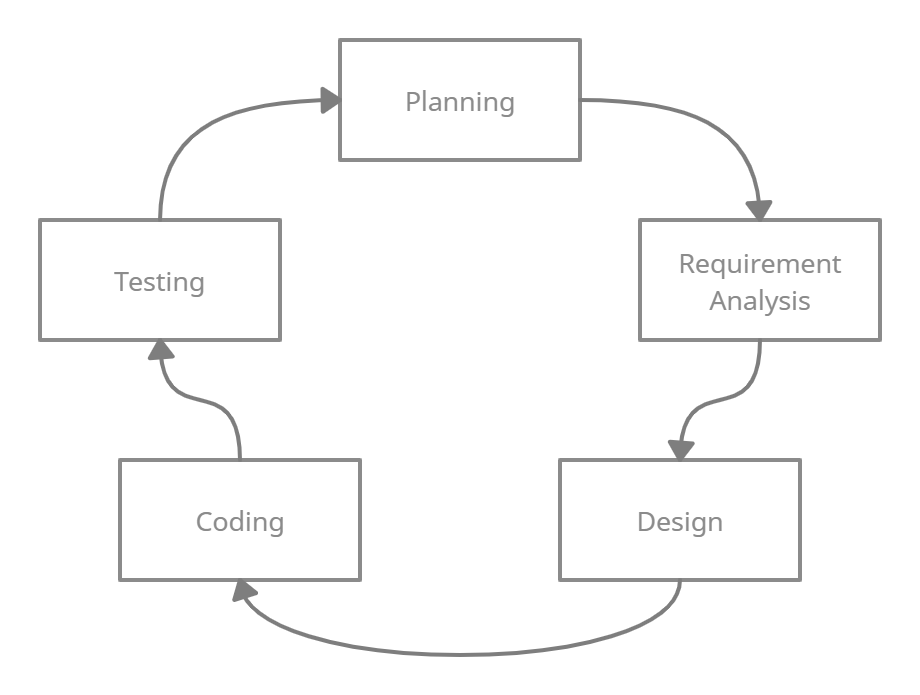
* Coding:

This phase contains the actual implementation of the system using Python.

* Testing:

We performed unit testing and integration testing. We created test cases to perform testing on the system.

* + 1. **Task Network**



**Fig. 5.3.2 Task Network**

* + 1. **Timeline Chart**
  1. **TEAM ORGANIZATION**
     1. **Team Structure**

The four team members are Vishal S. Wankhede, Harish D. Hatmode, Pragati S. Bankar, and Onkar G. Kulkarni. Onkar and Pragati conducted the research, Vishal and Harish created and analyzed the data. Different Algorithms were studied by all members. Pragati wrote the research paper that was approved by all the team members. Code implementation is performed by all the 4 team members.

* + 1. **Management reporting and communication**

All the four team members worked in balance to complete this project. We communicate with each other using different methods to discuss about the progress of the project. We attended Project reviews at every step which was conducted by our project guide.

1. **PROJECT IMPLEMENTATION**
   1. **OVERVIEW OF PROJECT MODULES**
      1. **Data Collection and Preparation**

The data about various agricultural commodities are available on the Open Government Data platform [7]. The weather information is provided by Agricultural Meteorology Division portal [8].

The inputs to this system are daily market price and weather datasets. The daily market price dataset with data from 2015 to 2020 is taken from Agmarknet [9]. It contains daily market price data of rice, wheat and soya oil. The dataset for Agricultural Commodities contains agricultural commodity name, state, district, market, date of arrival, minimum price of the day, maximum price of the day, average price of the day and, total quantity that arrived on the day, as attributes. The weather data has local time, cloudiness, precipitation, pressure, temperature, humidity, wind (speed, gusts, direction), sunrise/sunset, moonrise/moonset, and moon phase as attributes.

* + 1. **Data Preprocessing**

Before using a raw dataset, which is discussed in the previous subsection, data preprocessing techniques are applied to it. To get accurate and quality results, data cleaning is required to be performed on datasets. Hence, we applied data cleaning techniques to remove outliers and for filling missing values. Using a label encoder, we have converted categorical data into numerical data. We integrated the information of the market price dataset with temperature and rainfall attributes of weather data. Data transformation techniques were used to normalize the values of the dataset. After transformation, we get cleaned and useful data.

* + 1. **Data Repository**

After data preprocessing, the extracted, cleaned, and integrated data is loaded into the Data Repository. Data stored in a data model will be used for analytics and visualization of the system.

* + 1. **Prediction**

Prediction deals with future events. Data Mining can be used to solve classification and clustering problems. It uses statistics to solve these problems. In this paper, we discussed various Data Mining algorithms which can be applied to predict commodity price. These algorithms are Decision Tree, K-Nearest Neighbor (KNN), XGBoost, Logistic Regression, Support Vector Machine (SVM), and Naive Bayes.

This system takes agriculture’s daily price data along with weather data as input. The preprocessed dataset and all the data mining algorithms are provided to model builder to train the model. After training, the trained data helps to calculate the accuracy and other performance metrics of all the algorithms. Depending upon these performance metrics, the system compares all the algorithms to give efficient an algorithm as output. Using this efficient algorithm, the predictions about commodity prices are provided to the user.

This is done for price predictions of a single commodity product. For another commodity, the process of model building, training dataset, and giving efficient algorithm with the highest accuracy is repeated. Each commodity can have a different efficient algorithm. Based on that efficient algorithm, we have provided more accurate predictions. For example, soya oil is having decision tree as the best algorithm. Rice is having XGBoost as the best algorithm. Fig. 1 shows the system architecture for a particular raw product.

* 1. **TOOLS AND TECHNOLOGIES USED**

We are using python to implement this system. Python is a highly readable programming language. It supports many libraries using which we can build any type of project. We have used the Sklearn library to implement this system. All the modules of algorithms are available in the Sklearn library. XGBoost is implemented using the XGBoost library. Along with these libraries, we have used the following libraries:

* Tkinter
* Seaborn
* Numpy
* Pandas
* Matplotlib
* os

To run this system, we require visual studio code.

* 1. **ALGORITHM DETAILS**
     1. **Decision Tree**

Decision Tree is a supervised learning technique that can be used for solving classification problems. It is easy to understand as it shows a tree-like structure. In a decision tree, data is spilt based on certain decision kept at the root node which is decided by using the Attribute Selection Method (ASM). Using ASM, we can find the best attribute for the root node of the tree. In ASM, we have two methods - Information Gain and Gini Index.

* Information Gain:

Information gain (IG) calculation requires a value of entropy, which is a measure of impurity in the attribute.

*IG(S, A) = Entropy(S) - Entropy(Sv)*.

where,

S = Set of instances,

A = Attribute,

Sv = Subset of S with A = v

And entropy is given by,

*Entropy(S) = -P(y)P(y)- P(n)P(n)*.

where,

S = total number of samples,

P(y) = Probability of Yes,

P(n) = Probability of No.

* Gini Index:

Sklearn library uses Gini Index by default. It is given by,

*Gini Index = 1 -* .

* + 1. **K-Nearest Neighbor (KNN)**

KNN is a supervised learning technique that assumes the similarity between new data and available data. It put new data into the category that is most similar to it. It is used for both classification and regression.

* + 1. **XGBoost**

Extreme Gradient Boosting (XGBoost) is a decision-tree-based ensemble algorithm. It uses a gradient boosting framework. XGBoost is an optimized distributed algorithm that is highly efficient, flexible, and portable. It applies the principle of boosting using gradient descent architecture. XGBoost combines the estimates of weaker models and predicts the target variable. It has the ability of handling missing values and tree pruning. XGBoost is flexible because it has customized objective function and evaluation metrics. Also, it has a built-in cross-validation function.

* + 1. **Logistic Regression**

Logistic regression is a supervised learning method used for predicting the probability of a target variable. The output of logistic regression is binary or dichotomous. Logistic regression predicts the probability of dependent variable Y as the function of X. It is used to describe the data and explain the relationship between one dependent variable and one or more independent variables Dependent variables are binary having values 0(success/yes) and 1(failure/no) while independent variables are nominal, ordinal, interval or ratio-level. The formula for logistic regression is given by,

|  |
| --- |
| ln[] = |

Logistic model assumes that the natural log of the odds p/(1-p) is a linear function of the regressors. This model is less interpretable.

* + 1. **Support Vector Machine (SVM)**

Support Vector Machine (SVM) is a supervised learning algorithm that learn from the dataset and is used for classification. The goal of SVM is to find a decision boundary between two classes that are maximally far from any point in the training data. SVM assigns new data elements to one of the labeled categories.

For given set of training examples, SVM builds a model that predicts whether a new example falls into one class or the other. SVM is also used in predictive analysis.

* + 1. **Naive Bayes**

Naive Bayes classifier is a supervised learning algorithm based on Bayes’ theorem with strong independence assumptions between the features. Naive Bayes is a probabilistic classifier as it predicts based on probabilities of the object.

Bayes’ theorem calculates the probability of hypothesis with prior knowledge. It calculates conditional probability called a posterior or revised probability. According to Bayes’ rule, the probability of any two random variables A and B is given by,

|  |  |
| --- | --- |
|  | *P(A/B)* = . |

Where,

P(A/B) = Posterior probability,

P(A) = Prior probability,

P(B) = Marginal probability,

P(B/A) = Likelihood probability

Naive Bayes theorem predicts a class value for a given set of attributes. For each known class value,

* Naive Bayes calculates probabilities for each attribute, conditional on the class value.
* It uses product rule to obtain a joint conditional probability for the attributes.
* It uses Bayes rule to derive conditional probabilities for the class variable.

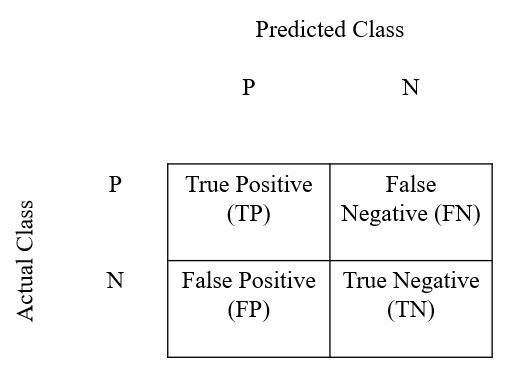
When all class values are calculated, output the class with the highest probability.

* 1. **EVALUATION OF ALGORITHMS**

All the algorithms discussed in the previous section are evaluated using metrics such as accuracy, precision, recall, and F1 score. These metrics are measured using a confusion matrix.

* + 1. **Confusion Matrix.**

Confusion matrix is a tool for analyzing how well our classifier can recognize tuples of different classes. Fig. 6.4.1 shows the confusion matrix.



**Fig. 6.4.1 Confusion Matrix**

* + 1. **Accuracy**

Accuracy of any classification algorithm is the percentage of test set tuples that are correctly classified by the model.

|  |
| --- |
| *Accuracy* = . |

TP = True Positive, TN = True Negative

FP = False Positive, FN = False Negative

* + 1. **Precision**

Precision means exactness of algorithm. It is the percentage of tuples that are correctly classified as positive are actual positive.

*Precision* = .

TP = True Positive, FP = False Positive

* + 1. **Recall**

Recall is the measure of completeness. It is the percentage of positive tuples which the classifier labeled as positive.

*Recall* = .

TP = True Positive, FN = False Negative

* + 1. **F1 Score**

F1 Score gives the balance between Precision and Recall. It is the harmonic mean of precision and recall.

*F1* = .

After calculating the values of all these metrics for each algorithm, we decide which algorithm is best for predictions based on any of the metrics selected by the user. Using this best algorithm, we predict the commodity prices more accurately.

1. **SOFTWARE TESTING**
   1. **TYPE OF TESTING**
      1. **Unit Testing**

We performed unit testing by testing each module individually. We first performed data preprocessing and tested that if data is completely preprocessed or not. After that we tested each algorithm separately and calculated values of accuracy, precision, recall, and F1 score.

* + 1. **Integration Testing**

We combinedly tested all the algorithms and evaluated them to check which one is more efficient. We tested this system by integrating it with a user interface created using the Tkinter Python library.

* 1. **TEST CASES & TEST RESULTS**
* **Test 1:** Software Crash Test

Test: Use every, option with random input in every possible combination.

Expected Result: All works properly.

Possible Failure: Software Crashes with some error.

**Result:** No Error Result: Crash Test Passed.

* **Test 2:** Input Test

Test: Give all possible commands to user inteface

Expected result: All commands should be performed properly.

Possible Failure: Server crash due to undefined requests.

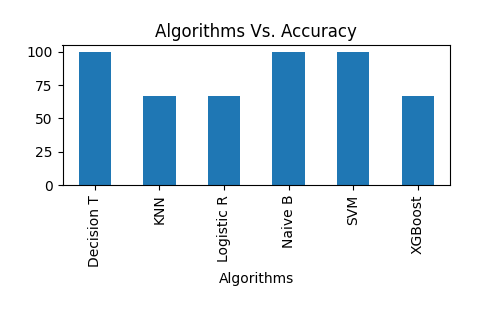
Actual Result: All commands working properly.

**Result:** Software working completely fine.

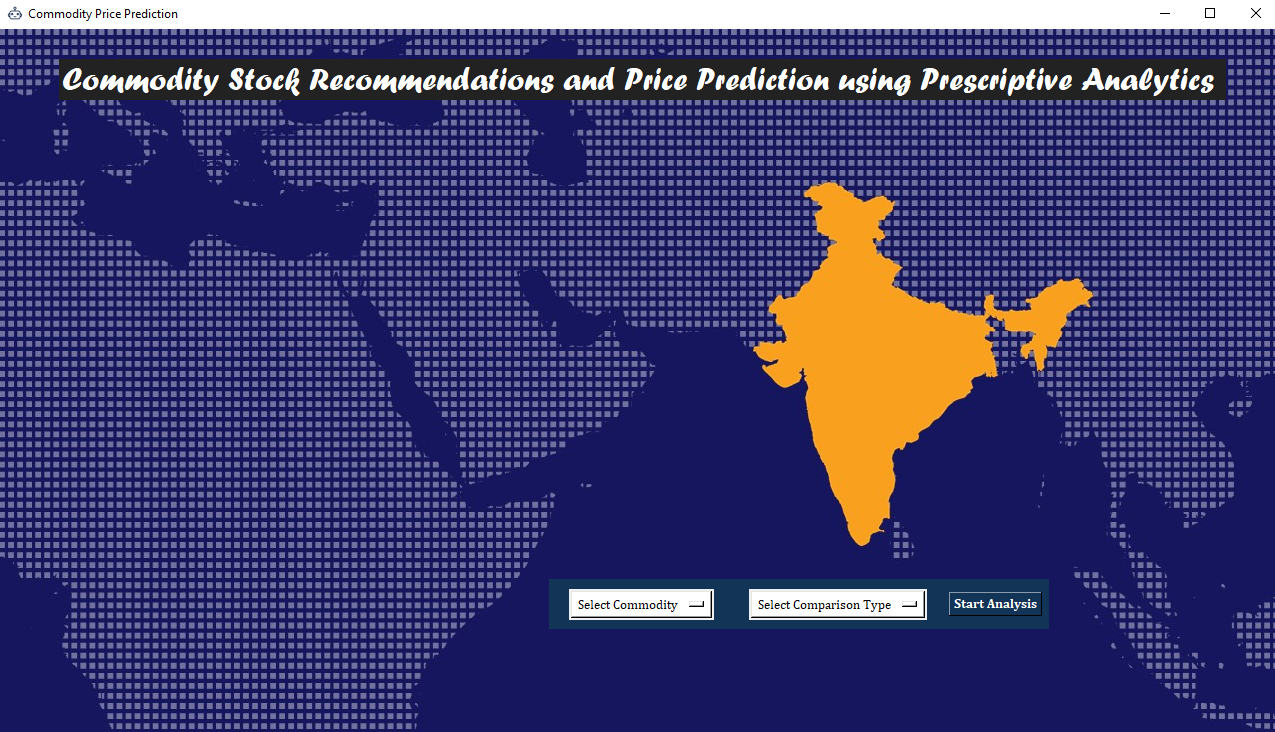
1. **RESULTS**
   1. **OUTCOMES**

Using this system, first, we get the algorithm that has the highest values for selected performance metrics. We plot the graph of evaluation to easily understand which algorithm works better for the selected commodity. See Fig. 8.2.1 Using this algorithm, the predicted prices and stock values are then displayed to the user. See Fig. 8.2.3

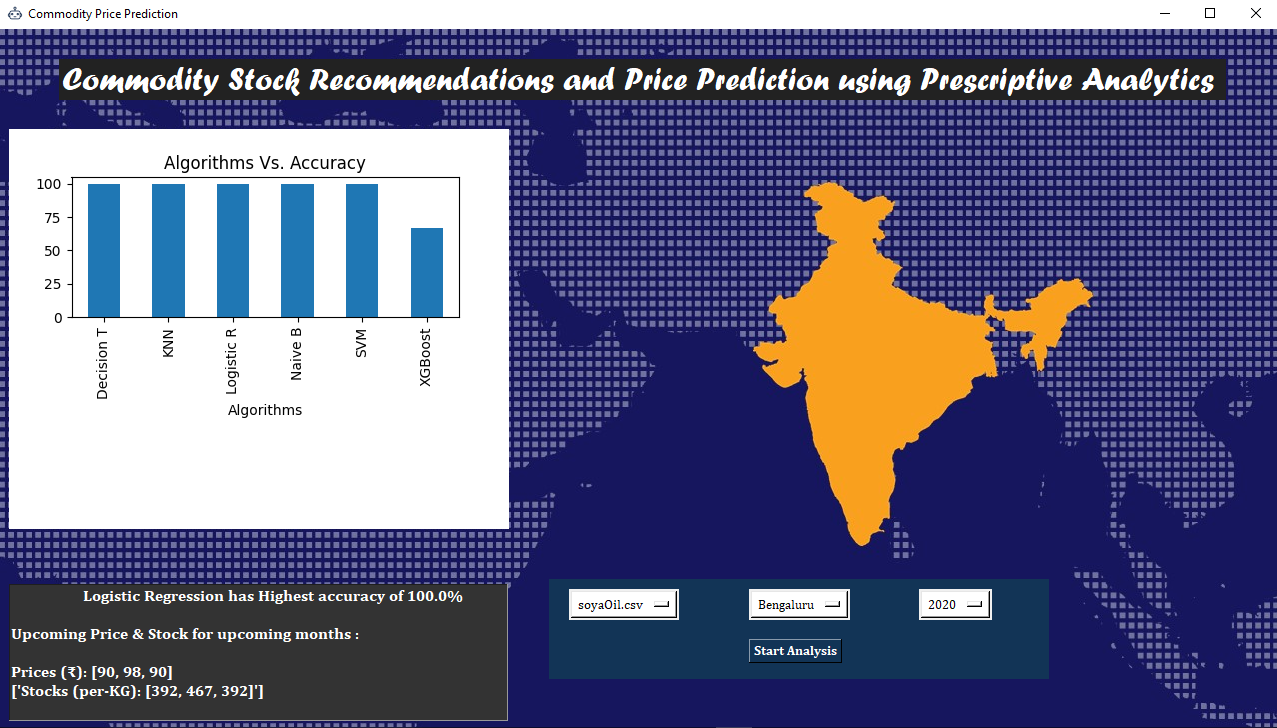
* 1. **SCREENSHOTS**



**Fig. 8.2.1 Evaluation of Algorithms for Soya Oil.**



**Fig. 8.2.2 Before selecting commodity price and performance metric**



**Fig. 8.2.2 After selecting commodity price and performance metric**

1. **CONCLUSIONS**
   1. **CONCLUSIONS**

The process of comparison between data mining algorithms based on metrics such as accuracy, precision, recall, and F1 score gives us an efficient algorithm for each commodity. This efficient algorithm may differ from commodity to commodity depending upon their measure of accuracy for that commodity. Due to this, we get more accurate and quality results. Accurate predictions of commodity prices and stock will help farmers in the decision-making process.

* 1. **FUTURE WORK**

In the future, we can implement an algorithm that predicts prices and stocks with more accuracy. We can implement crop selection based on future prices and the environment near the farm.

* 1. **APPLICATIONS**

This system has its applications in the agriculture field. Farmers can use this system on large scale to improve their financial growth.

**APPENDIX A**

**Problem statement feasibility assessment**

Let P1 = Given success cases viz., Predicted prices are correct, predicted stock values are correct, and evaluation gives correct efficient algorithm.

For a Problem P1 to be NP-Hard, Satisfiability problem (SAT) must be reducible to P1;

i.e. SAT ∝ P1;

Let for CNF-SAT,

CNF =

X1: True (i.e.1) if predicted prices are correct

X2: True (i.e.1) if predicted stock values are correct

X3: True (i.e.1) if evaluation gives correct efficient algorithm

Here, there are 8 possibilities for which CNF will be satisfied. They are:

|  |  |  |
| --- | --- | --- |
| **X1** | **X2** | **X3** |
| 0 | 0 | 0 |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |
| 1 | 1 | 1 |

This means, we have 8 possibilities for 3 variables. We can check whether CNF is true or not for these 8 possibilities.

i.e. For n variables, we have 2n possibilities.

This tells us that our problem P1 is taking exponential time. This is similar to satisfiability problem as it is also an exponential time taking algorithm. As satisfiability problem is reducible to our problem P1, problem P1 is NP-Hard.

We will devise non-deterministic algorithm for problem P1:

Algosat ()

{

for i= 1 to 8

{

Xi=Choice (true, false) -------- Non- deterministic statement

if (CNF == true)

Success () -------- Non- deterministic statement

else

Failure () -------- Non- deterministic statement

}

}

This algorithm will take O(1) time i.e. polynomial time if we know the correct values for Xi.

So, this is a non-deterministic algorithm that takes polynomial time O(1).

We have proved that P1 is NP- Hard. Also, we have written non-deterministic polynomial time taking algorithm for P1. So, our problem P1 is NP-Complete.

**NP**

Have non-deterministic algorithm

**NP – Hard**

SAT ∝ P1

**SAT**

**P**

**NP Complete**

Now, to prove that this non-deterministic algorithm will be solve in future, we have to prove that P = NP.

According to Cook’s theorem, satisfiability problem is in P if P = NP.

**APPENDIX B**

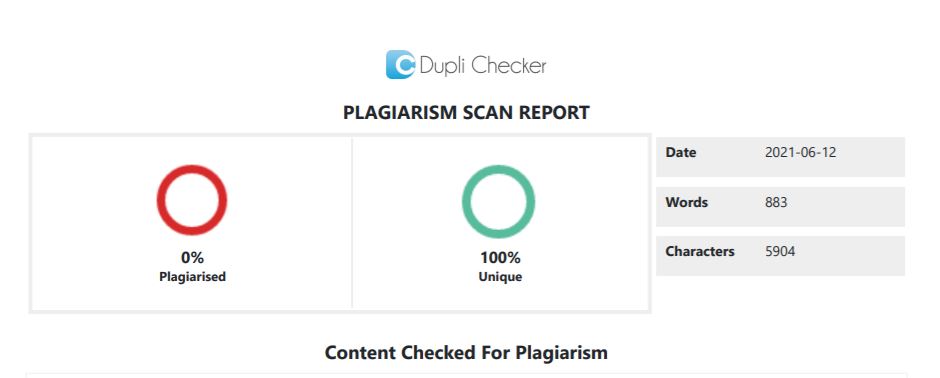
**Details of paper publication**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Name of Journal** | **ISBN/ISSN No** |
| 1 | Journals of Communications | ISSN: 1796-2021 |

1. **Comments of Reviewers:**
2. **Certificates:**
3. **Paper:**

**APPENDIX C**

**Plagiarism Report**



**REFERENCES**

1. S. Shrivastava, S. N. Pal, and R. Walia, “Market Intelligence for Agricultural Commodities Using Forecasting and Deep Learning Techniques,” Springer Nature Switzerland, 2019.
2. Kumar U. D., “Business Analytics: The Science of Data-Driven Decision Making,” Wiley, New York, 2018.
3. Larose D. T., Larose C. D., “Data Mining and Predictive Analytics,” Wiley, New York, 2017.
4. Mann A. K., and Kaur N., “Review paper on clustering techniques,” Global Journal of Computer Science and Technology, 2013.
5. Shah N., Solanki M., Tambe A., and Dhangar D, “Sales Prediction Using Effective Mining Techniques,” International Journal of Computer Science and Information Technologies, Vol. 6 (3), 2015.
6. S. Cheriyan, S. Ibrahim, S. Mohanan, S. Treesa, “Intelligent Sales Prediction Using Machine Learning Techniques,” IEEE, 2018.
7. Open Government Data (OGD), India. <https://data.gov.in/>
8. Agricultural Meteorology Division. <http://imdagrimet.gov.in/>
9. Agmarknet, Agriculture Marketing Portal, <https://agmarknet.gov.in/>