FARMERS ASSISTANCE AND AGRICULTURAL PRODUCTS MARKET FORECASTING PORTAL

APROJECT REPORT

Submitted by

R ADITHYA (170501003)
V AKASH (170501008)
J S ANANDATIRTHA (170501011)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

SRI VENKATESWARA COLLEGE OF ENGINEERING
(An Autonomous Institution; Affiliated to Anna University, Chennai-600025)

ANNA UNIVERSITY :: CHENNAI 600 025

MAY 2021

SRI VENKATESWARA COLLEGE OF ENGINEERING
(An Autonomous Institution; Affiliated to Anna University, Chennai-600025)

ANNA UNIVERSITY :: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **“FARMERS ASSISTANCE AND AGRICULTURAL PRODUCTS MARKET FORECASTING PORTAL”** is the bonafide work of “**R ADITHYA (170501003), V AKASH (170501008) and J S ANANDATIRTHA (170501011)”** who carried out the project work under my supervision.

SIGNATURE

Dr. R. ANITHA
HEAD OF THE DEPARTMENT

COMPUTER SCIENCE & ENGG

SIGNATURE

Dr. P. JANARTHANAN
SUPERVISOR
ASSOCIATE PROFESSOR
COMPUTER SCIENCE & ENGG

Submitted for the project viva-voce examination held on ………………..

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

India is the world's largest producer of Pulses, Rice, wheat and Spice products, still, there is no interface to sell farmers products directly. The farmers' welfare depends on better market value to their products but they struggle to get it. Because the mediator collects agricultural products from farmers and makes a profit out of them and farmers are getting only 20% of consumer price. To overcome the above bottlenecks, in this project, we developed a portal to sell the products which significantly increase farmer’s income. The app consists of two sections (i) Farmers can directly market their products to the consumers by eliminating the mediators, (ii) The final phase is the Time Series Forecasting model to suggest the product price range using historical data. The FbProphet technique is used to forecast their commodity price. And this model is evaluated using Mean Squared Error (MSE), Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE) to prevent the loss. Subsequently, the farmer can trace the days’ worth, with the help of a visualization method that represents the entire data, including item, price of the day in graphical visualization. Thus our project mainly aims to provide the farmer's income to the maximum level and decreases the actual selling price due to the elimination of agents.

ACKNOWLEDGEMENT

We thank our Principal **Dr. S. Ganesh Vaidyanathan,** Sri Venkateswara College of Engineering for being the source of inspiration throughout our study in this college.

We express our sincere thanks to **Dr. R. Anitha,** Head of the Department, Computer Science and Engineering for her encouragement accorded to carry this project.

With profound respect, we express our deep sense of gratitude and sincere thanks to our guide, **Dr P Janarthanan, Associate Professor,** for his valuable guidance and suggestions throughout this project.

We are also thankful to our project coordinators **Dr. R. Jayabhaduri, Associate Professor, Dr. N. M. Balamurugan, Associate Professor and Ms.V.Rajalakshmi, Assistant Professor** for their continual support and assistance.

We thank our family and friends for their support and encouragement throughout the course of our graduate studies.

R ADITHYA
V AKASH
J S ANANDATIRTHA

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

MSE	Mean Squared Error
RMSE	Root Mean Squared Error
APMC	Agriculture and Marketing Committee
ARIMA	Auto-Regression Integrated Moving Average
GARCH	Generalized Autoregressive Conditional Heteroskedasticity
ACF	Autocorrection Function
LSTM	Long-short Term Memory
MAPE	Mean Absolute Percentage Error
SARIMA	Seasonal Autoregressive Integrated Moving Average
BP	Back Propagation
RBF	Radial Basis Function
QR-RBF	Quantile Regression- Radial Basis Function
ELM	Extreme Learning Machines
IDE	Integrated Development Environment
XML	Extensible Markup Language
HTML	Hypertext Markup Language
BaaS	Backend as a service
JSON	JavaScript Object Notation
BSD	Berkeley Software Distribution

CHAPTER 1

INTRODUCTION

Agricultural products account for a large proportion of the market. India is a global agricultural powerhouse. India's agriculture is composed of many crops, with the foremost food staples being rice and wheat. Indian farmers also grow pulses, potatoes, sugarcane, oilseeds, and such non-food items as cotton, tea, coffee, rubber, and jute. Agricultural markets in most states of India are established and regulated under State APMC (Agriculture and Marketing Committee) Acts. These led to the creation of regulated markets, also called mandis. Today, the country has 7,246 such mandis each of which caters to an average area of nearly 450 sq km. The mandis control almost the entire wholesale trade of agricultural goods. The Operation of these mandis has a huge bearing on the retail prices of fruits, vegetables and other agricultural produce. Introduced in the 1960s, APMC Acts prohibited farmers from dealing directly with retailers and required them to sell their products to licensed middlemen or market functionaries. In the absence of market information, farmers have not got remunerative prices and the middlemen have got the major share in profit. These differences in agricultural productivity are a function of local infrastructure, soil quality, micro-climates, local resources, farmer knowledge and innovations [7]. The biggest obstacle in increasing farmers' income in India is the profiteering middleman. Commission agents, traders, and wholesalers take a major chunk of profit from farmers. For example, in Punjab, there is no provision for the sale of crops other than paddy and wheat. As a result, farmers are forced to sell their products to middlemen at low prices. The agricultural product prices are determined by the supply and demand for a relevant year [9]. The Middlemen pay less for the crop value on the pretext of quality factors. Concerning the price hike, farmers say that the crisis is not helping them in any way because the middlemen make the most gain.

Suicide among farmers has been rising in India for the last 20 years. Nearly 3,00,000 farmers have ended their lives by ingesting pesticides or by hanging themselves due to the loss they get from crop price. The suicide rate among Indian farmers had increased by 47%, according to a 2011 census in a country where agriculture remains the largest employment sector. Taking a step towards betterment, farmers have to sell their products directly to customers or adopt the online portal for selling products rather than selling them to middlemen. The direct marketing and contract farming, which helps in eliminating intermediaries leads to better prices for consumers and return for farmers. Especially, providing more agricultural product predictions can help establish a well-planned management strategy in advance and contribute to the stability of supply and demand in the agricultural market [13]. Technology can help the farmer to sell the products directly to the customer and their income would be increased by eliminating the middlemen. In addition, this approach will increase the GDP of the nation.

1.1 TIME SERIES FORECASTING

Time series data is a collection of quantities that assembled over even intervals in time and ordered chronologically. The time interval at which the dataset is being collected is based on time series frequency. Time series forecasting of agricultural products plays a major role in the sustainability of agricultural production. Providing price forecasting information would help in decision making for managing agricultural supplies and helping to improve purchasing behaviors of consumers. In addition to the seasonality on the production side caused by year-round cycle climate change, the prices are affected by the users' preferences to products and suppliers' trading strategy and behavior. These human factors are not always seasonal but they could be represented as relations between events in past and the current status. Time Series Prediction has been used in many applications such as financial forecasting and agricultural price forecasting [10].

Time-series graph plots have observed values on the y-axis against an increment of time on the x-axis. These graphs visually represent the behavior and patterns in the data. More specifically, visualizing time series data provides a preliminary tool for detecting if data:

- Is mean-reverting or has explosive behavior?
- Has a time trend?
- Exhibits seasonality?
- Demonstrates structural breaks?

This, in turn, can help guide the testing, diagnostics, and estimation methods used during time series modeling and analysis. A useful abstraction for selecting forecasting methods is to break a time series down into systematic and unsystematic components.

- **Systematic**: Components of the time series that have consistency or recurrence and can be described and modeled.
- **Non-Systematic**: Components of the time series that cannot be directly modeled.

A given time series consists of three systematic components: level, trend, seasonality, and one non-systematic component called noise. These components are defined as follows:

- Level: The average value in the series.
- Trend: The increasing or decreasing value in the series.
- Seasonality: The repeating short-term cycle in the series.
- Noise: The random variation in the series.

For our case, we are keen on three factors, i.e., active cases, deaths and recovered cases. The granularity of these features can be adjusted tantamount to the frequency of observations and the number of records available in the sample space. For cross-referencing our findings we relied on comparing results with expected pandemic trends and made sure to provide consistent results.

To filter out the systematic and non – systematic components many algorithms can be adapted. Of those, additive and multiplicative models are the most primitive and basic ones.

1.1.1 The Additive Model

Synthetically it is a model of data in which the effects of the individual factors are differentiated and added to model the data. It can be represented by

$$Y\ (t) = \text{Level} + \text{Trend} + \text{Seasonality} + \text{Noise} \qquad (1.1)$$

In the additive model, the behavior is linear where changes over time are consistently made by the same amount, like a linear trend. In this situation, the linear seasonality has the same amplitude and frequency. The additive model is useful when the seasonal variation is relatively constant over time.

1.1.2 The Multiplicative Model

In this situation, trend and seasonal components are multiplied and then added to the error component. It is not linear, can be exponential or quadratic and represented by a curved line as below:

$$Y(t) = \text{Level} * \text{Trend} * \text{Seasonality} * \text{Noise} \tag{1.2}$$

Different from the additive model, the multiplicative model has an increasing or decreasing amplitude and/or frequency over time. The multiplicative model is useful when the seasonal variation increases over time.

Though these algorithms effectively decompose into multiple categories, they might not be as accurate and fail at modeling the true distributions because of the simplicity. Therefore, we move onto much more intricate and sophisticated algorithms which not only improve accuracy but also capture correlations efficiently.

CHAPTER 2

LITERATURE SURVEY

Assis and Remali compared the forecasting performances of different time series methods for forecasting cocoa bean prices. For types of univariate time series models were compared, namely ARIMA, Generalized Autoregressive Conditional Heteroskedasticity (GARCH) and the mixed ARIMA and GARCH models. Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Theil's inequality coefficient were used as the selection criteria to determine the best forecasting model. This study revealed that the time series data were influenced by a positive linear trend factor while a regression test result showed the non-existence of seasonal factors. Moreover, the Autocorrection Function (ACF) and the Augmented Dickey-Fuller tests have shown that the time series data was not stationary but became stationary after the first order of the differentiating process was carried out. Based on the results of the forecasting model, the mixed ARIMA/GARCH model outperformed the exponential smoothing, ARIMA and GARCH models [2].

Adanacioglu and Yercan analyzed the seasonal price variation of tomato crop and forecasted the monthly tomato prices in Turkey using Seasonal ARIMA (SARIMA). They removed the high seasonality of tomatoes using a seasonal index. SARIMA (1,0,0)(1,1,1) model was selected as the most suitable model to forecast of tomato prices and the tomato prices adjusted seasonally on October [1].

Helin Yin et-al proposed the STL-ATTLSTM model, which integrates the seasonal trend decomposition using Loess (STL) preprocessing method and attention mechanism based on long-short term memory (LSTM). They forecasts monthly vegetable prices using various types of information, such as vegetable prices, weather information of the main production areas, market trading volumes. This method decomposes time-series vegetable price data into trend, seasonality and remainder components. The proposed STL-ATTLSTM was applied to five crops, namely cabbage, radish, onion, hot pepper, and garlic, and its performance was compared to three benchmark models. The performance results show that the LSTM model combined with the STL method (STL-LSTM) achieved a 12% higher prediction accuracy than the attention LSTM model that did not use the STL method and solved the prediction lag arising from high seasonality. The attention LSTM model improved the prediction accuracy by approximately 4% to 5% compared to the LSTM model. The STL-ATTLSTM model achieved the best performance, with an average root mean square error (RMSE) of 380, and an average mean absolute percentage error (MAPE) of 7% [12].

Paredes-García et-al, proposed a methodology for price forecasting of fruits and vegetables using Queretaro State. The daily prices of several fruits and vegetables were extracted from the National System of Market Information. The prices were used to compute the weekly average price of each product and their span commercialization in Q4 and over the median of historical data. Moreover, product characterization was performed to propose a methodology for future price forecasting of multiple agricultural products within the same mathematical model and it resulted in the identification of 18 products that fit the Seasonal Auto-Regressive Integrated Moving Average (SARIMA) model. Finally, future price estimation and validation was performed to explain the product price fluctuations between weeks and it was found that the relative error for cost of products modeled was less than 10%, e.g. Hass avocado (7.01%) and Saladette tomato

(8.09%). The results suggest the feasibility for the implementation of systems to provide information for better decisions by Mexican farmers [8].

In this paper, the theory and construction methods of four models are presented for predicting the vegetable market price, which are BP neural network model, the neural network model based on genetic algorithm, RBF neural network model and an integrated prediction model based on the three models above. The four models are used to predict the Lentinus edodes price for Beijing Xinfadi wholesale market. A total of 84 records collected between 2003 and 2009 were fed into the four models for training and testing. In summary, the predicting ability of BP neural network model is the worst. The neural network model based on genetic algorithm was generally more accurate than RBF neural network model. The integrated prediction model has the best results [3].

Dongqing Zhang identified the Quantile Regression-radial basis function (QR-RBF) neural network model to forecast the soybean market. The model has two characteristics: (1) using quantile regression models to describe the distribution of the soybean price range; and (2) using RBF neural networks to approximate the nonlinear component of the soybean price. In order to optimize the QR-RBF neural network model parameters, a hybrid algorithm known as GDGA, based on a combination of the genetic algorithm (performing a global search) and a gradient descent method (performing a local search), is proposed. Data regarding the monthly domestic soybean price in China were analyzed and the results indicate that the proposed hybrid GDGA is effective. Furthermore, the results suggest that the influencing factors of soybean price vary at different price levels. Money supply and port distribution price of imported soybean were found to be important across a range of quantiles; output of domestic soybean and consumer confidence index were important only for low quantiles; and import volume of soybean and consumer price index were important only for high quantiles [4].

In view of the importance of seasonal forecasting of agricultural commodity price, particularly vegetable prices, and the limited research attention paid to it previously, this study proposes a novel hybrid method combining seasonal-trend decomposition procedures based on loess (STL) and extreme learning machines (ELMs) for short-, medium-, and long-term forecasting of seasonal vegetable prices. In the formulation of the proposed method (termed STL-ELM), the original vegetable price series are first decomposed into seasonal, trend, and remainder components. Then, the ELM is used to forecast the trend and remainder components independently, while the seasonal-naïve method is used to forecast seasonal components with a 12-month cycle. Finally, the prediction results of the three components are summed to produce an ensemble prediction of vegetable prices. In addition, an iterated strategy is used to implement multi-step-ahead forecasting. In terms of two accuracy measures and the Diebold-Mariano test, the experimental results show that the proposed method is the best-performing method relative to the competitors listed in this study, indicating that the proposed STL-ELM model is a promising method for vegetable price forecasting with high seasonality [5].

Yanqun et-al proposed a PSO-BP neural network for vegetable price prediction. In order to predict vegetable price accurately, 117 sets of green pepper and related factors price data from 2012 to 2015 in Dan Zhou city were selected as the sample data, of which 100 groups were training data and 17 groups were test data. Based on analyzing fluctuant features of vegetable price, with the global stochastic optimization idea to optimize initial weights and thresholds of back propagation (BP) neural network, the PSO-BP prediction model concerning vegetable retail price was set up by using the particle swarm optimization (PSO) algorithm. The experimental results indicated that compared with the traditional BP method, the PSO-BP method could overcome the over-fitting problem and the local minima problem, effectively reduced training error and increased the predicting precision [11].

Lu Ye et-al identified the Back Propagation (BP) neural network forecasting model and Autoregressive Integrated Moving Average (ARIMA) forecasting model of Hainan vegetables price are set up. Based on above two models, linear combination and nonlinear combination forecasting model of vegetables price are established by linear programming method and BP neural network method. The results indicate that nonlinear combination model is superior to linear combination model, and linear combination model is superior to any single model, reflecting that combination model can make individual models complement each other's advantages to improve prediction accuracy [6].

CHAPTER 3

PROPOSED SYSTEM

The scope of the project revolves around two phases: 1) A simple interface by which farmers can sell their goods directly to customers by using an APP, 2) we use time series models to forecast multiple features, which are elucidated in the section.

3.1 DESCRIPTION OF THE APP AND ITS FEATURES

We describe the algorithm, which explains how the system will work, i.e., the processing logic behind the application. The flowchart represents the pictorial representation of the processing logic of the data flow diagram of the Farmer's Market.

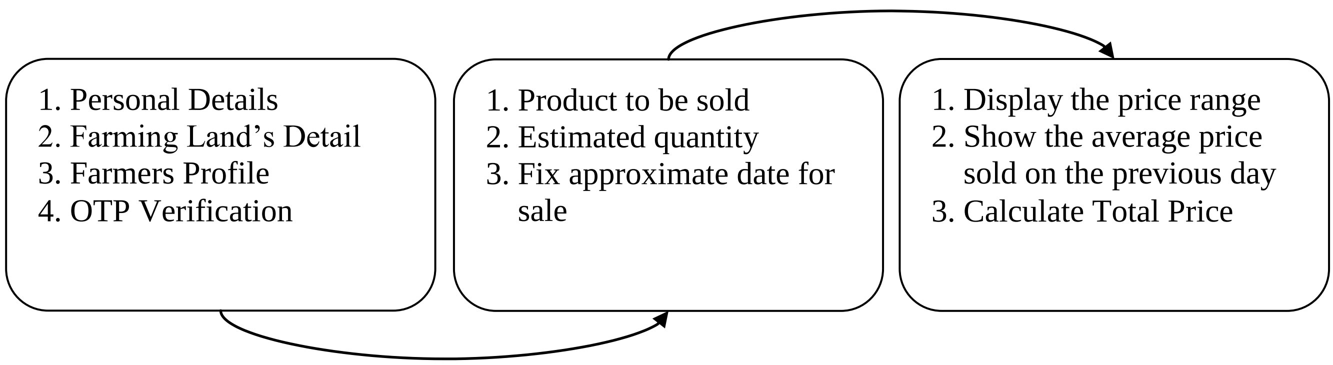

Figure 3.1 Agricultural Products Market Forecasting Portal System

Farmers who want to use the platform must register themselves by creating a user ID and password. In order to be a seller the farmer must login using user-name and password. For Stream less verification of the user-ID and password in correct format we have used StreamBuilder. Once availed with the username and a password, the user can perform different operations like updating the available items and checking how much is remaining to sale, and viewing the account information. After logging in, the user can navigate to an Upcoming market page. For example, HOSUR FARMER MARKET:1, the number 1 is a unique ID given to each market, which contains details of market-like location, date, and details of the products.

On clicking the Vendor page on the middle right of the screen, it takes to a new page where a farmer can add the products available, and the price of the product is predicted using machine learning- Time Series Forecasting. Time series forecasting is used to analyze the data to display the average price and the price range for the particular day. Then, it will take you to the next page, where the product's available details will be displayed. We have used user-ID Mapping to map farmer ID with details and his products to that particular farmer’s app screen. It will start once the user login and helps the user to get a personal interface of the app. Thus no 2 famers will have the same app screen because of their unique activity and User Id Mapping. The final page is the Market which shows the products available for sales. This Market Page is same for everyone in the particular area. The project’s main objective is to increase the farmer's income to the maximum level. The features in the app will significantly help farmers be self-sufficient sellers.

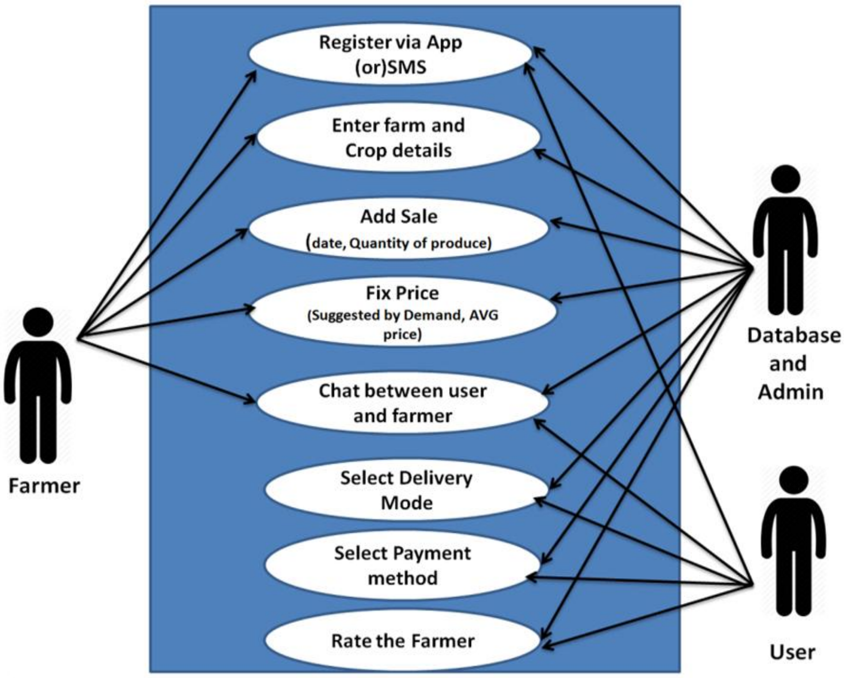

Figure 3.2 Use Case Diagram of the Portal

3.2 TIME SERIES FORECASTING

The application of machine learning (ML) techniques to time series forecasting is not straightforward. One of the main challenges is to use the ML model for actually predicting the future in what is commonly referred to as forecasting. This issue stems from the temporal structure of the data since, at variance with standard ML projects, it is not enough to apply a pre-trained model on new data points to get the forecasts but, as we will see in this post, additional steps are required. Very few examples of time series forecasting with ML are available online which are really end-to-end since they keep the focus on testing the model on available data and overlook the forecasting part.

3.2.1 Proposed Model for Forecasting

Our proposed work is to forecast the price of the vegetables. Vegetable price changes fast and unstable which makes great impact in our daily life. Time Series Forecasting can be used to develop an innovative model to predict the market price of respective commodity. Price predictions is highly useful in agriculture for forecasting the market price of respective commodities and also useful for farmers to plan their crop cultivation activates so that they could fetch more price in the market. Government can use the market forecast price for planning and implementation of agriculture development programs to stabilize the market price for the respective commodity. Forecasting technique can be used to develop an innovative model to predict the market price of the particular commodity. Approaches like ARIMA, SARIMA, and Prophet plays an important role in forecasting.

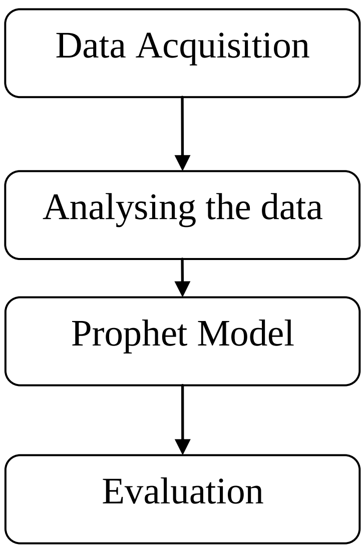

Figure 3.3 Time Series Forecasting Architecture

3.2.1.1 Data Acquisition

Forecasting requires the historical data to make better predictions. For this purpose, the vegetable price dataset is obtained from National Horticultural Research and Development Foundation website which consists of 122 rows and 10 columns. The columns represents the recent prices of vegetables such as Bitter Guard, Sweet Corn, Tindly, Tomato, Lady’s Finger, and Brinjal. It contains the daily prices from January 2021 to April 2021.

3.2.1.2 Analysing the data

Exploratory is exploited to analyse and investigate dataset for summarizing their main characteristics. It helps determine how best to manipulate data sources to get the answers you need, making you need, making it easier for data scientists to discover patterns, test hypothesis, or to verify assumptions

EDA is primarily used to observe, what the data can reveal beyond the formal modeling or hypothesis testing task and provides a better understanding of data set variables and the relationships between them. It can also help determine if the statistical techniques you are considering for data analysis are appropriate.

3.2.1.3 Prophet Model

The Prophet is a straightforward algorithm to develop the forecasting model for time-oriented data that offers sensible and more prominent forecasts. In 2017, this algorithm was developed by Facebook's Core Data Science Team. This algorithm can be implemented in both python and R. This model depends on the additive regression approach that identifies trends, season, holidays and then consolidates them together using the eqn (3.1)

$$\text{Forecasting} = g(t) + s(t) + h(t) + e(t) \tag{3.1}$$

CHAPTER 4

SYSTEM REQUIREMENTS

4.1 HARDWARE REQUIREMENTS

PROCESSOR	:	Intel(R) Core(TM) i5-8250U CPU @1.60GHz 1.80GHz
HARD DISK DRIVE	:	1 TB
RAM	:	4 GB
GPU	:	NVIDIA GeForce

4.2 SOFTWARE REQUIREMENTS

OPERATING SYSTEM	: Windows 10
PROGRAMMING LANGUAGE	: Python 10

4.3 DESCRIPTION OF THE TOOLS USED

This project is implemented using many packages and the application was developed using Android Studio. The following are the tools used:

- Python
- Android studio
- Flutter
- Xml
- Firebase
- Json

4.3.1 Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA. On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as: A flexible Gradle-based build system, A fast and feature-rich emulator, A unified environment where you can develop for all Android devices, Apply Changes to push code and resource changes to your running app without restarting your app, Code templates and GitHub integration to help you build common app features and import sample code, Extensive testing tools and frameworks, Lint tools to catch performance, usability, version compatibility, and other problems, C++ and NDK support.

4.3.2 Flutter

Flutter is Google’s new open-source technology for creating native Android and iOS apps with a single codebase. Unlike other popular solutions, Flutter is not a framework; it’s a complete SDK – software development kit – which already contains everything you will need to build cross-platform applications. This includes a rendering engine, ready-made widgets, testing and integration APIs, and command-line tools.

Dart is Flutter’s object-oriented language that uses Ahead-of-Time compilation techniques and compiles into native code without that additional bridge. This noticeably speeds up the app startup time.

Flutter is Google's SDK for crafting beautiful, fast user experiences for mobile, web, and desktop from a single codebase. Flutter works with existing code, is used by developers and organizations around the world, and is free and open source. We think Flutter will help you create beautiful, fast apps, with a productive, extensible and open development model.

4.3.3 XML

XML stands for an extensible markup language. XML is a simple, very flexible text format derived from SGML. A markup language is a set of codes, or tags that describes the text in a digital document. The most famous markup language is a hypertext markup language (HTML), which is used to format Web pages. XML, a more flexible cousin of HTML, makes it possible to conduct complex business over the Internet. XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere.

Whereas HTML tells a browser application how a document should look, XML describes what's in the document. In other words, XML is concerned with how information is organized, not how it is displayed. This is a powerful way to store data in a format that can be stored, searched, and shared. Most importantly, since the fundamental format of XML is standardized, if you share or transmit XML across systems or platforms, either locally or over the internet, the recipient can still parse the data due to the standardized XML syntax.

4.3.4 Firebase

Firebase is a Backend-as-a-Service (Baas). Google Firebase is a Google-backed application development software that enables developers to develop iOS, Android and Web apps. It provides developers with a variety of tools and a service to help them develop quality apps, grows their user base, and earns profit. It is built on Google's infrastructure. Firebase is categorized as a NoSQL database program, which stores data in JSON-like documents.

4.3.5 JSON

JSON stands for JavaScript Object Notation. It is an open standard file format and data interchange format that uses human-readable text to store and transmit data objects consisting of attribute-value pairs and arrays. JSON is a lightweight format for storing and transporting data. JSON is often used when data is sent from a server to a web page. The JSON format is used for serializing and transmitting structured data over a network connection. It is primarily used to transmit data between a server and web applications. Web services and APIs use JSON format to provide public data. It can be used with modern programming languages and very common data format with a diverse range of applications.

JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python and many others. These properties make JSON an ideal data-interchange language. These are universal data structures. Virtually all modern programming languages support them in one form or another. It makes sense that a data format that is interchangeable with programming languages also be based on these structures.

4.3.6 Pandas

The pandas is an open-source, BSD-licensed Python library providing high-performance, and easy-to-use data structures and data analysis tools for the python programming language. Python with pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Pandas is a Python package that provides fast, flexible, and expressive data structures designed to make working with structured (tabular, multidimensional, potentially heterogeneous) and time series data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis / manipulation tool available in any language. It is already well on its way toward this goal.

4.3.7 Prophet

Prophet is a Python micro framework for financial markets. Prophet strives to let the programmer focus on modeling financial strategies, portfolio management, and analysing backtests. It achieves this by having few functions to learn to hit the ground running, yet being flexible enough to accommodate sophistication. Prophet's data generators, order generators, and analysers are all executed sequentially which is conducive to allowing individuals to build off of others work.

Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. It works best with time series that have strong seasonal effects and several seasons of historical data. Prophet is robust to missing data and shifts in the trend, and typically handles outliers well.

CHAPTER 5

IMPLEMENTATION MODULES

Flutter is an open-source mobile SDK development that can be used to build native-looking Android and iOS applications from the same code base. The central idea behind Flutter is the use of widgets. It is by combining different widgets that developers can build the entire UI. Each of these widgets defines a structural element (like a button or menu), a stylistic element (a font or color scheme), a layout aspect (like padding), and many others. We used StreamBuilder to check whether the user name and password are in the correct formats. Like almost everything in Flutter, **StreamBuilder** is a Widget, and as you can assume - it rebuilds its UI according to the new values passed via Stream it listens to one of the exciting things about Flutter is that it blurs the line between Activities in contrast to Android JVM. Any time you would need to display something to the user, you are to provide a Widget. Those widgets can be nested into other widgets (just like views). The widgets care about visible elements only, which works nicely with architectures like MVI/MVP. If we use reactive streams and provide all UI updates, the widgets play well with them. The reactive streams are even easier to use in Flutter than in Java/Kotlin. StreamBuilder is a Widget that can convert a stream of user-defined objects to widgets. This takes two arguments:

- A stream
- A builder that can convert the elements of the Stream to widgets

Suppose you have a Stream that updates if there is any UI update (maybe from user interaction, or maybe resulted from network updates). If your "main" widget includes a StreamBuilder, which listens to the Stream, it can act as the element in charge of translating your states to views.

5.1 FORECASTING

5.1.1 DATASET DESCRIPTION

The **National Horticultural Research and Development Foundation** (NHRDF) is engaged in Research and developmental programmes. The NHRDF implements various research projects funded by the Ministry of Agriculture Government of India. Research projects are carried out onion, garlic and other vegetable crops during the Kharif, Late Kharif and Rabi seasons at Regional Research Stations.

NHRDF offers an excellent range of market data of various vegetables like Tomato, Beetroot, Tindly, Lady’s Finger and other vegetables. Also provides the data for each state and in particular each districts. This contain the data for the past 10 years.

Date	Tomato	Brinjal	Ladys Finger	Spinach	Carrot	Beetroot	Tindly	Bitter Guard	Sweet Corn
01-01-2021	15	18	15	10	16	35	26	10	11
02-01-2021	15	18	19	12	20	38	23	15	9
03-01-2021	15	25	16	10	13	27	26	17	9
04-01-2021	15	30	20	10	13	22	25	15	8
05-01-2021	10	26	20	11	20	22	26	12	9
06-01-2021	8	28	20	10	20	23	26	15	9
07-01-2021	8	24	20	10	16	30	30	12	10
08-01-2021	9	26	19	13	16	24	26	12	5
09-01-2021	18	22	20	13	35	22	25	10	15
10-01-2021	12	22	19	12	35	15	30	11	14
11-01-2021	15	22	17	12	35	13	36	12	14
12-01-2021	10	20	20	12	35	15	33	12	15
13-01-2021	14	20	18	11	35	17	34	10	10
14-01-2021	19	22	19	10	35	18	40	11	10
15-01-2021	20	18	19	12	35	10	43	10	9
16-01-2021	25	22	20	12	35	12	43	10	9
17-01-2021	26	20	20	11	35	10	45	15	9
18-01-2021	20	32	20	11	30	10	41	14	9
19-01-2021	15	38	17	10	25	15	40	15	9
20-01-2021	10	39	20	10	20	20	40	13	13
21-01-2021	5	15	21	11	20	30	43	15	15
22-01-2021	5	13	22	10	15	38	46	10	10

market_data

Figure 5.1 Vegetable Price Dataset

5.1.2 Prophet Model

A prophet is a procedure for forecasting time series data based on an additive model where non-linear trends fit yearly, weekly and daily seasonality. It works best with time series that have strong seasonal effects and several seasons of historical data. Prophet is robust to missing data and shifts in the trend and typically handles outliers well.

Prophet provides a) Saturation growth and b) piecewise linear model for g(t). Piecewise linear model is considered as the default approach for forecasting and can be measured using the eqn (3.2). This model is flexible to both linear and non-linear functions and robust to outliers. The prophet is persistent in handling missing data and can capture transitions in the trend. The accuracy of the forecasting model will be similar to that of professional forecasters.

$$g(t) = (K + a(t)^T \delta)t + (m + a(t)\gamma) \quad (5.1)$$

Where k indicates growth rate, denotes rate adjustments, and m implies offset parameter. Also, the seasonality model allows frequency modifications such as daily, weekly, yearly, quarterly and this often helps in business time series analysis. Prophet uses Fourier series to construct the model based on periodic as in eqn (3.3)

$$s(t) = \sum_{n=1}^{N} a_n \cos\left(\frac{2\pi nt}{P}\right) + b_n \sin\left(\frac{2\pi nt}{P}\right) \quad (5.2)$$

The prophet library is an open-source library designed for making forecasts for univariate time-series data. It is easy to use and designed to automatically find a good set of hyper parameters for the model to make skillful forecasts for data with trends and seasonal structure by default.

Algorithm: Prophet Forecast

Step 1: The Prophet algorithm mainly depends on two input parameters: (a) ds (Datestamp) column – ought to be desired format YYYY-MM-DD as demanded by pandas and (b) y column – incorporates only the numeric values taken into consideration for forecasting the future value**.**

Step 2: To extend the ds column with a specified number of date's makes_future_dataframe method should be applied.

Step 3: Specify the total number of days in Periods parameter

Step 4: Now, invoke the predict method that produces each row with the date and corresponding predicted values 'yhat'. Along with yhat, the objects contain the lower bound (yhat_lower) and upper bound (yhat_upper)

Step 5: Finally, performance_metrics is used to obtain various evaluation metrics for each horizon by comparing actual value and forecasted value (yhat)

CHAPTER 6

SNAPSHOTS OF MODULES

6.1 FARMERS PORTAL

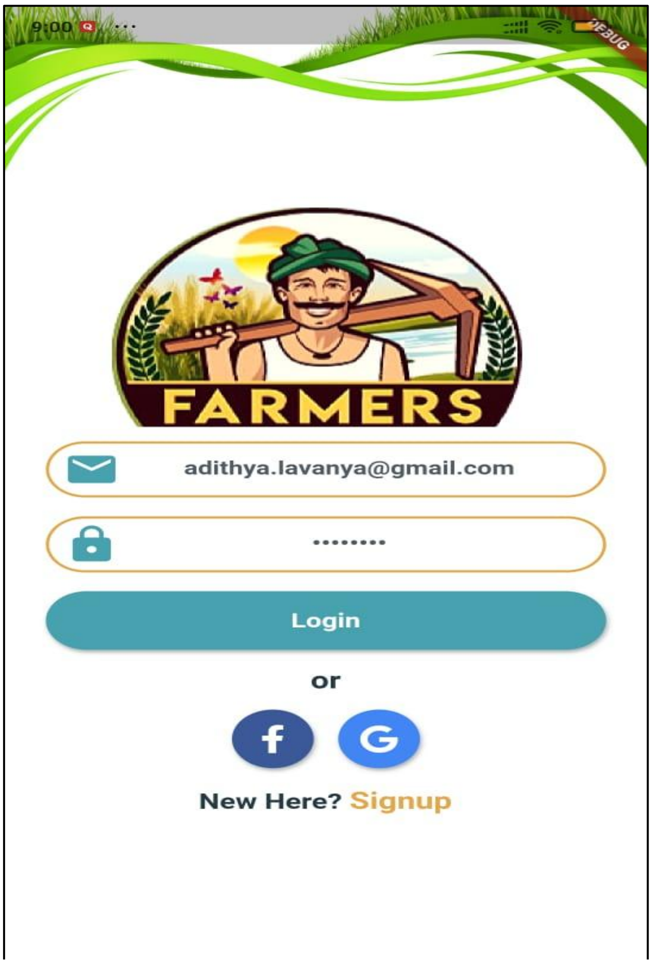

Figure 6.1 Registration and Login Page

Once availed with the username and a password for the app, the user can perform different operations like sales, viewing the account information, adding the products and their details. On the other hand, the user can sign in using their

Google or Facebook account. This project's main objective is to build an app that will help farmers from India to sell their products directly to the public.

Figure 6.2 Upcoming Markets

The main feature of our project is upcoming markets. After register, you have to log into the account using a given user name and password, and this will take you to the Upcoming Market page; active markets are shown, for example; HOSUR-FARMER MARKET: 1 ID 1 is a unique ID given to each farmer's Market that displays in the app. This feature can help both vendor and customer in getting knowledge about the markets. The ID comprises all the information of the market, such as name, location and date of the Market registered.

Figure 6.3 Adding Products from Vendor Side

When you click the vendor page, it will take you to a new page to add products. The section contains the product name, market ID unit price, and available units; you can add an image to the product and save the product. You can also make changes to the current product like price change and this will be reflected in the database. These details are stored in the database along with the User Id. This user is generated once the user registers for the app and travels with user all along whenever he uses this app. This user id also sent to database and is used when the farmer reviews his products. Only the products he uploaded will be shown and thus the upload screen will be unique to each farmer.

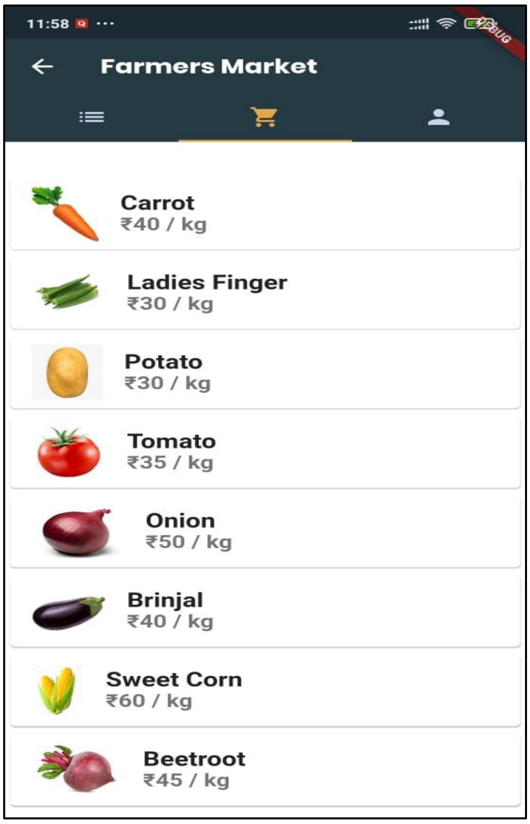

Figure 6.4 Displaying the Vegetable Prices

The price of the product can be suggested using Time series algorithm to help farmer. Price prediction is highly useful in agriculture for forecasting the market price for the respective commodities and also useful for farmers to plan their crop cultivation activities so that they could fetch more price in the market. The following image shows the added image in the ADD product by the farmer.

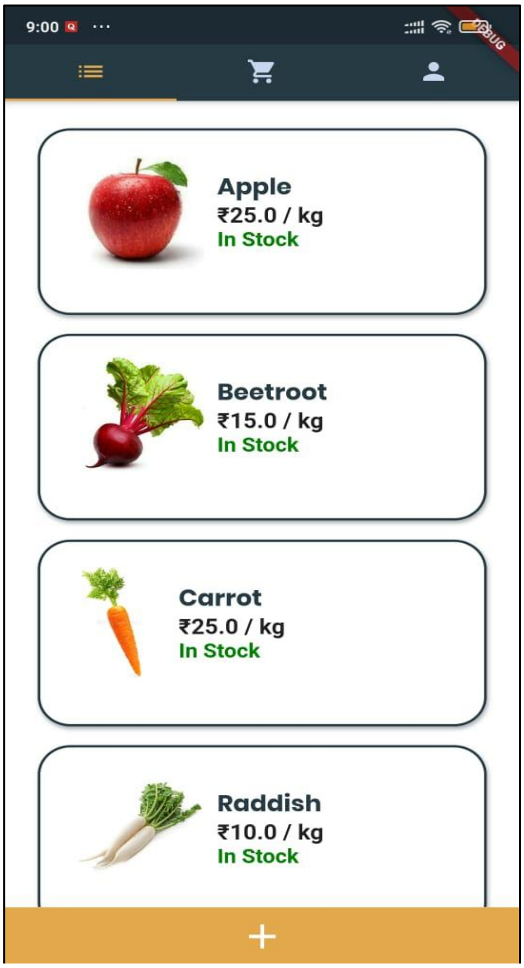

Figure 6.5 Displaying the Farmer Products

This page will be different for different farmer. Not all vegetables displayed will be same for the farmer and this page will vary depending the farmers upload. This is due to use of USER-ID Mapping which we have implemented.

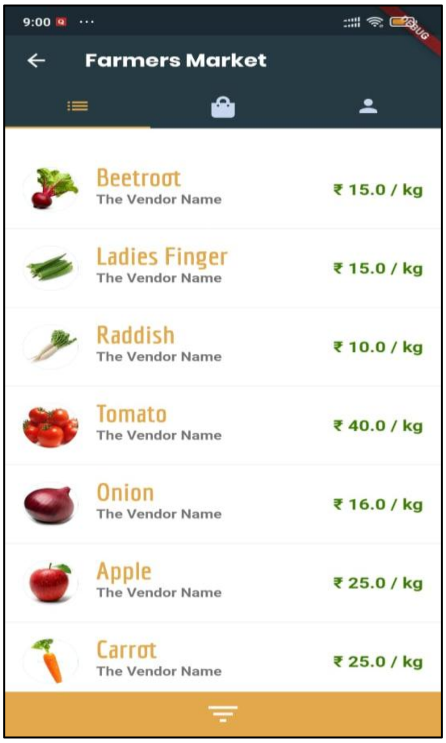

Figure 6.6 Market Page

The above images show the products available in the farmer market. This is common for every farmer. The farmer cannot make changes here; he does not have access to edit this page. He can only view the markets like a customer. The market page shows available products and the product's information. This market shown here will vary upon the ID. The market with ID: 2 will have completely different products shown. We have also used User id mapping here, but here this is Market Id mapping. Only the products uploaded in that market are shown. Each market will be unique.

Figure 6.7 Vendor Page

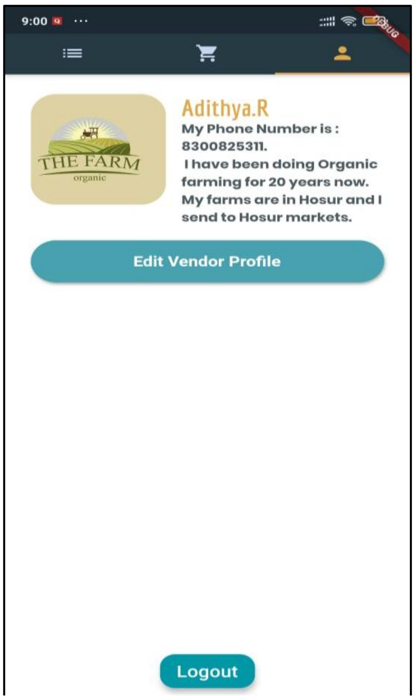

Figure 6.8 Vendor Details

This module describes the profile set of the farmers in the platform. Every customer should know about the farmer, i.e., phone number, name, location of farmers, and edit the profile by clicking the edit vendor page.

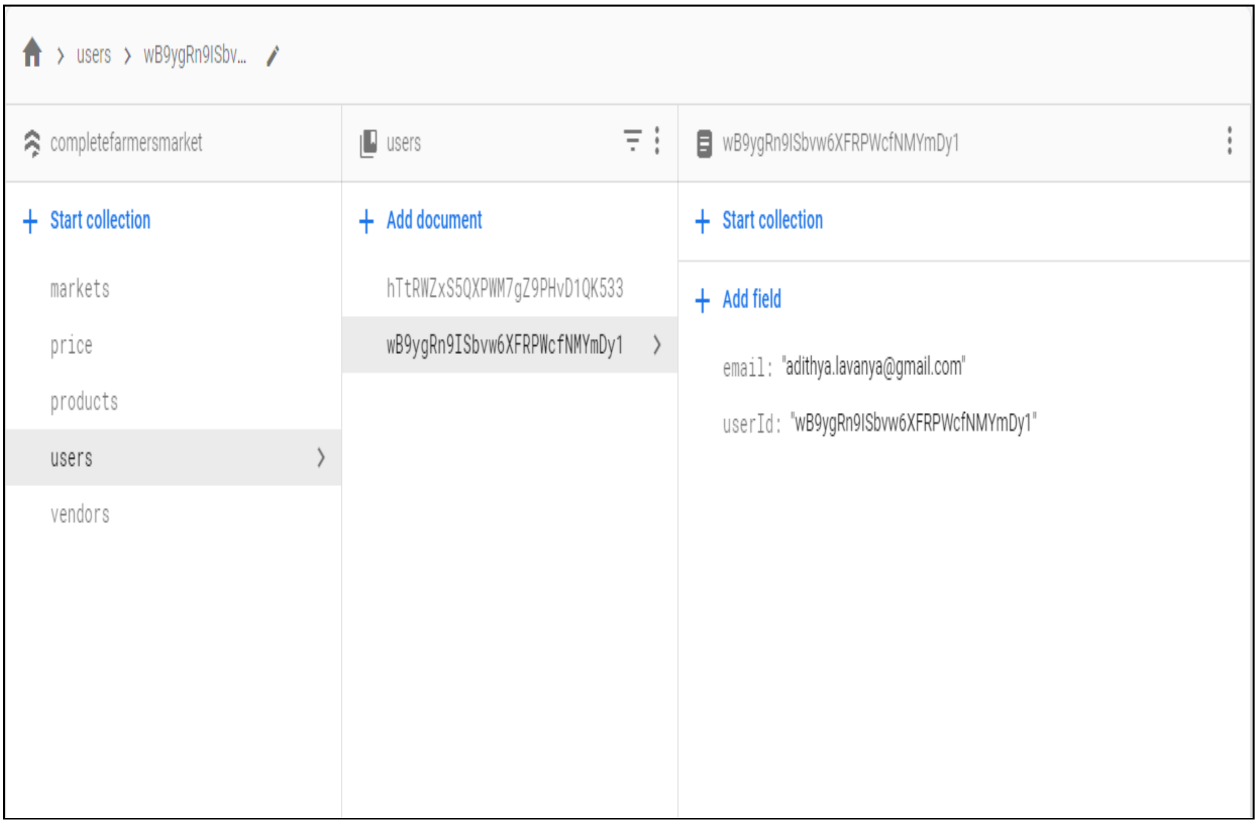

Figure 6.9 Databases of the Vendors

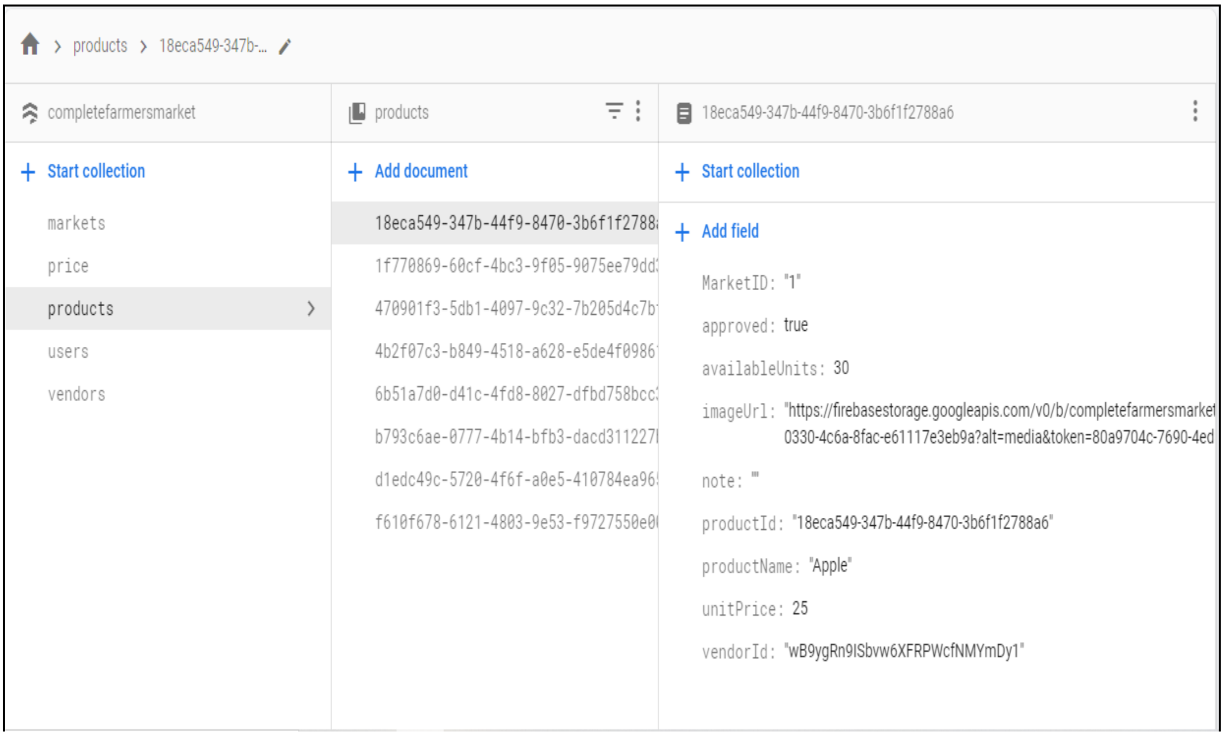

Figure 6.10 Databases of the Product Uploaded

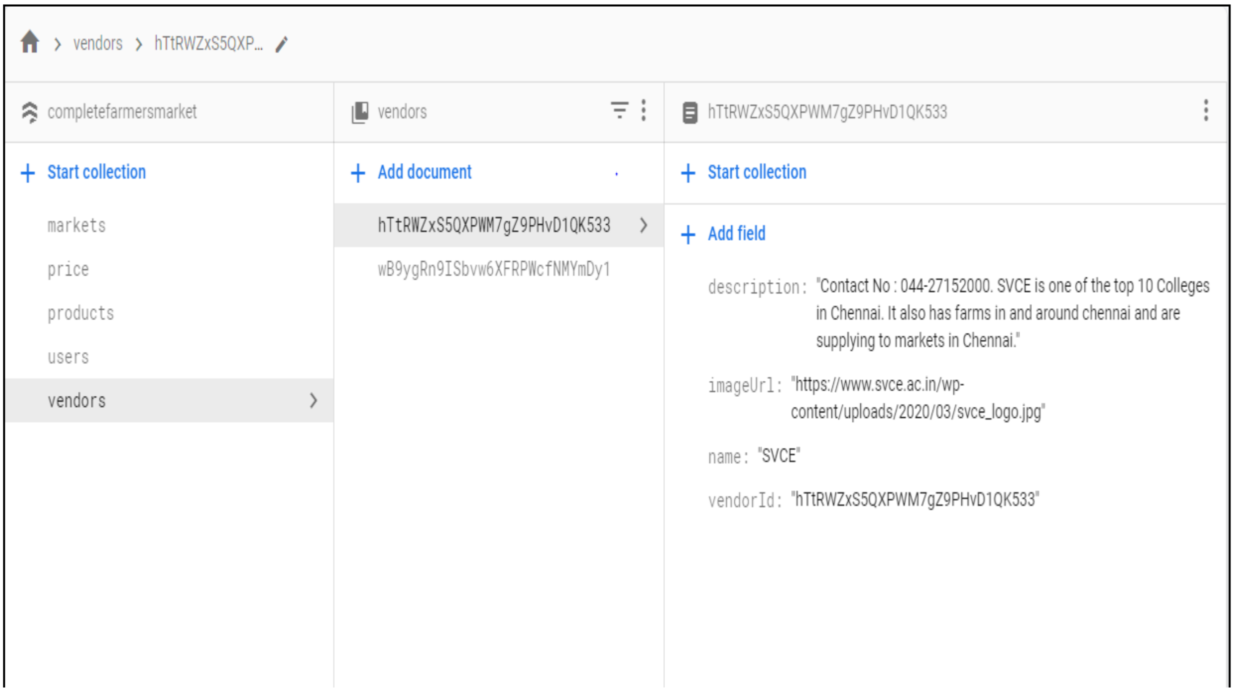

Figure 6.11 Databases of the Vendors

6.2 Time Series Forecasting

```
[2] import pandas as pd
    from fbprophet import Prophet
    import matplotlib.pyplot as plt
    from sklearn.metrics import mean_squared_error
    from sklearn.metrics import mean_absolute_error
```

```
[3] dataset = pd.read_csv('market_data.csv',parse_dates=['Date'])
    dataset.head(5)
```

	Date	Tomato	Brinjal	Ladys Finger	Spinach	Carrot	Beetroot	Tindly	Bitter Guard	Sweet Corn
0	2021-01-01	15	18	15	10	16	35	26	10	11
1	2021-02-01	15	18	19	12	20	38	23	15	9
2	2021-03-01	15	25	16	10	13	27	26	17	9
3	2021-04-01	15	30	20	10	13	22	25	15	8
4	2021-05-01	10	26	20	11	20	22	26	12	9

6.12 Prophet Forecasting Model

```
[ ] dataset.shape
```

```
(121, 10)
```

```
[ ] dataset.describe()
```

	Tomato	Brinjal	Ladys Finger	Spinach	Carrot	Beetroot	Tindly	Bitter Guard	Sweet Corn
count	121.000000	121.000000	121.000000	121.000000	121.000000	121.000000	121.000000	121.000000	121.000000
mean	13.280992	26.512397	22.115702	9.652893	22.611570	23.603306	34.752066	21.363636	9.950413
std	5.851244	11.401839	4.058304	1.458942	9.451077	9.138271	6.186384	7.422937	3.203777
min	5.000000	6.000000	13.000000	7.000000	10.000000	10.000000	20.000000	10.000000	5.000000
25%	9.000000	18.000000	20.000000	8.000000	16.000000	16.000000	30.000000	15.000000	8.000000
50%	12.000000	25.000000	23.000000	9.000000	20.000000	22.000000	36.000000	22.000000	10.000000
75%	15.000000	34.000000	25.000000	11.000000	28.000000	30.000000	40.000000	26.000000	11.000000
max	30.000000	60.000000	31.000000	13.000000	53.000000	45.000000	46.000000	40.000000	20.000000

6.13 Prophet Forecasting Model

In this project, we are concern about the Vegetable prices in India. So, we assign ds column with timestamp and y data frame with prices of Vegetables in India as required by the Prophet model. Figure 5.14 demonstrates the transformation of the dataset as requisite by the algorithm.

```
[ ] #creating the dataframe for the particular product for forecasting
    Tomato_price = pd.DataFrame()
    Tomato_price['ds'] = dataset['Date']
    Tomato_price['y'] = dataset['Tomato']
    Tomato_price.columns = ['ds','y']

[ ] Tomato_price.head(5)
```

	ds	y
0	2021-04-20	15
1	2021-04-21	15
2	2021-04-22	15
3	2021-04-23	20
4	2021-04-24	15

6.14 Dataframe for the particular Product for Forecasting

```
m = Prophet(interval_width=0.95) # interval width represents the confidence level
m.fit(Tomato_price) #fitting the dataframe to train
future = m.make_future_dataframe(periods=10)  #This creates the future dates based on periods
future.tail(10)
```

```
INFO:numexpr.utils:NumExpr defaulting to 2 threads.
INFO:fbprophet:Disabling yearly seasonality. Run prophet with yearly_seasonality=True to override this.
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
INFO:fbprophet:n_changepoints greater than number of observations. Using 8.
```

	ds
12	2021-05-01
13	2021-05-02
14	2021-05-03
15	2021-05-04
16	2021-05-05
17	2021-05-06
18	2021-05-07
19	2021-05-08
20	2021-05-09
21	2021-05-10

6.15 Extending the Datestamp column

Once the dataset is fit into the model, then the process is to predict the values. In order to, predict the new prices of the vegetables, we have to extend the datestamp column which is done with the help of makes_future_dataframe. This extends the ds column with the specified number of dates using the parameter period. Figure 5.15 explains the predicted values (yhat) along with upper bound (yhat_upper) and lower bound (yhat_lower).

```python
forecast = m.predict(future)   # Forecasting the values
forecast['yhat'] = forecast['yhat'].round()  # actual value of the crop
forecast['yhat_lower'] = forecast['yhat_lower'].round()  #lower range value of the crop
forecast['yhat_upper'] = forecast['yhat_upper'].round()  #Maximum price
forecast[['ds', 'yhat']].tail(10)
```

	ds	yhat
12	2021-05-01	15.0
13	2021-05-02	9.0
14	2021-05-03	9.0
15	2021-05-04	12.0
16	2021-05-05	12.0
17	2021-05-06	12.0
18	2021-05-07	14.0
19	2021-05-08	15.0
20	2021-05-09	9.0
21	2021-05-10	9.0

6.16 Prophet Forecasting Model

```
Tomato: 13.0
Brinjal: 26.0
Spinach: 9.0
Lady's Finger: 22.0
Beetroot: 23.0
Tindly: 34.0
Corn: 10.0
Bitter Guard: 23.0
```

6.17 Prophet Forecasting Model

CHAPTER 7

CONCLUSION AND FUTURE WORKS

7.1 CONCLUSION

Despite being the world’s largest producer and consumer of rice, vegetables, fruits, and many other agricultural produces, the life of Indian farmers have not improved much. Farmers tend to suffer much because of decreased income due to the role of Middle men in the block chain. There is wide difference between the price at which they buy produce from farmer and sell it in the market. In order to diminish the need of Middle men in the system, and make farmer a direct seller to/in the market, we have designed an App. This app has machine learning which can help farmers fix the price when uploading their produce. The app will also make the entire process of farmer selling his produce, much simpler, efficient and organised. We have made the app using dart language on flutter platform, thus making app to be run on both android and IOS. The backend of this project is Firebase. For Price forecasting algorithm we used Prophet Algorithm.

7.2 FUTURE WORK

For future work, we have decided to add new languages and delivery planning, and voice assistant for illiterate farmers. We are not including payment options because there are many complications involved and government procedures to be undertaken. In future, we have also planned to integrate payment part after completing the procedures. In addition, Chat bot and Farming guidelines will be added in the application part to improve the working of the App. Furthermore, trend analysis will be done to help the farmers in crops cultivation, or to decide the crop based on the demand.

REFERENCES

1. Adanacioglu, H., and Yercan, M. (2012) 'An Analysis of Tomato Prices at Wholesale Level in Turkey: An Application of SARIMA Model'. Custos e @Gronegócio 2012, Vol. 8, pp. 52–75.

2. Assis, K., Amran, A., and Remali, Y. (2010) 'Forecasting Cocoa Bean Prices using Univariate Time Series Models'. Res.World, Vol. 1, pp. 71-79.

3. Changshou, L., Qingfeng, W., Liying, Z., Junfeng, Z., and Sufen, S. (2010) 'Prediction of Vegetable Price Based on Neural Network and Genetic Algorithm'. 4th Conference on Computer and Computing Technologies in Agriculture (CCTA), Nanchang, China. pp. 672-681.

4. Dongqing, Z., Guangming, Z., Jing, L., Kaiping, M., & Huan, L. (2018) 'Prediction of Soybean Price in China using QR-RBF Neural Network Model'. Computers and Electronics in Agriculture, Vol.154, pp. 10-17.

5. Li, Y., Zhu, Z., Kong, D., Han, H., and Zhao, Y. (2019) 'EA-LSTM: Evolutionary Attention-Based LSTM for Time Series Prediction'. Knowl. Based Syst, Vol.181, pp. 123 – 127.

6. Lu, Y., Xiaoli, Q., Yuping, L., Yanqun, L., and Weihong L. (2016) 'Vegetables price forecasting in Hainan province based on linear and nonlinear combination model'. 13th International Conference on Service Systems and Service Management (ICSSSM), pp. 1-5.

7. Mahadevan, R. (2003) 'Productivity Growth in Indian Agriculture: The Role of Globalization and Economic Reform'. Asia-Pacific Development Journal, Vol. 10 No. 2, pp.57–72.

8. Paredes-García, Wilfrido, J. (2019) 'Price Forecasting and Span Commercialization Opportunities for Mexican Agricultural Products'. Agronomy, Vol. 9, pp. 1521 – 1530.

9. Rao, J.M. (1989) 'Agricultural Supply Response: A Survey'. Agric. Econ, Vol. 3, pp.1–22.

10. Xiong, T., Li, C., & Bao, Y. (2018) 'Seasonal Forecasting of Agricultural Commodity Price using a Hybrid STL and ELM Method: Evidence from the Vegetable Market in China'. Neurocomputing, Vol. 275, pp. 2831–2844.

11. Yin, L., Yuping, L., Weihong, L., Qidao, S., Yanqun, L., and Xiaoli, Q. (2015) 'Vegetable Price Prediction Based on PSO-BP Neural Network'. 8th International Conference on Intelligent Computation Technology and Automation (ICICTA), pp. 1093-1096.

12. Yin, H., Jin, D., Gu, Y.H., Park, C.J., Han, S.K., and Yoo, S.J. (2020) 'STL-ATTLSTM: Vegetable Price Forecasting Using STL and Attention Mechanism-Based LSTM'. Agriculture, Vol. 10, No.12, pp. 612 - 620.

13. Yoo, D. (2016) 'Developing Vegetable Price Forecasting Model With Climate Factors', Korean J. Agric. Econ, Vol. 57, pp.1-24.