AN INTELLIGENT SYSTEM FOR FORECASTING FARMER'S REQUIREMENTS IN MAHAWELI PROJECT

Project Id – 20_21-J11

Thenuwara V.S. (IT16073760)

Wijetunge W. P. T. T. (IT17167024)

De Silva D.K.G. (IT17172806)

Kavindi H.G.A. (IT17178150)

Final (Draft) Report

B.Sc. Special Honors Degree in Information Technology

Department of Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

May 2021

AN INTELLIGENT SYSTEM FOR FORECASTING FARMER'S REQUIREMENTS IN MAHAWELI PROJECT

Project Id – 20_21-J11

Thenuwara V.S. (IT16073760)

Wijetunge W. P. T. T. (IT17167024)

De Silva D.K.G. (IT17172806)

Kavindi H.G.A. (IT17178150)

Final (Draft) Report

(The dissertation was submitted in partial fulfilment of the requirements for the B.Sc. Special Honors degree in Information Technology)

B.Sc. Special Honors Degree in Information Technology

Department of Software Engineering

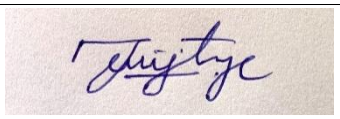
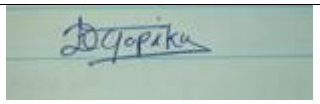
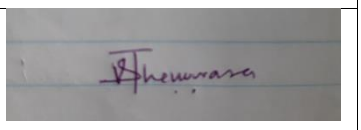
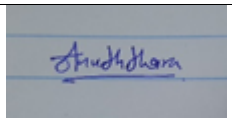
Sri Lanka Institute of Information Technology

Sri Lanka

May 2021

DECLARATION

We declare that this is our own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. Also we hereby grant to Sri Lanka Institute of Information Technology the non-exclusive right to reproduce and distribute my dissertation in whole or part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as article or books).

Name	Student ID	Signature
Wijetunge W. P. T. T.	IT17167024	
De Silva D.K.G.	IT17172806	
Thenuwara V.S	IT16073760	
Kavindi H.G.A	IT17178150	

The above candidate is carrying out research for the undergraduate Dissertation under my supervision.

…… ……..

Prof. Pradeep Abeygunawardhana Date

(Supervisor)

ABSTRACT

Mahaweli development project is the largest irrigation development program in the agricultural sector of Sri Lanka. However, the lack of performance of the project has increasingly come under criticism for the past decade [1]. Hence the solution is introduced for the farmers and buyers as 'Mahaweli Market' where the application is designed to analyze and predict the harvesting, marketplace, yield quality, and disorders and enhanced with the trading platform for sales. This article explains predicting future harvesting, future trends in the marketplace of cultivated crops, buying and selling platform, and crop healthiness predicting modules integrated with an android mobile app. Understanding the reasons for the inclusion of the above-mentioned modules is crucial, Specifically, the harvest prediction is done using a linear regression model and predicted seasonally. Secondly market analysis and prediction are done for the initial stage gather model data and create linear regression model. Thirdly an online trade platform Mahaweli market to support buyers and sellers is developed as another module developed using Collaborative Filtering, K-Nearest Neighbors (K-NN). Finally, the component that wraps out, checking healthiness of a specific crop. Healthiness of crops are predicted for the first phase of the implementation; the crop identification part has been done using image classification by Keras Sequential model using python.

Key words: intelligent system, checking healthiness, agriculture

ACKNOWLEDGMENT

This research would not be possible without the assistance and dedication of numerous individuals and in the process of making an idea to a successful working product, we had to go through a difficult time period. Hardworking of the group members and the great support and the encouragement which we had laid the red carpet for this research.

First, we want to appreciate to our group members Vishal Thenuwara, Thamal Wijetunge, Anuththara kavindi and Gopika De Silva for their greatest participation in our research. This research has become a reality with the dedication and hard working of the group members.

As well, we express my greatest appreciation to Prof. Pradeep Abeygunawardhana for his invaluable support for this research. Our research "An Intelligent System for Forecasting Farmer's Requirements in Mahaweli Project" has become a reality with the guidance and the immense supervision within few months with the bless of our supervisor.

Finally, we dedicate my gratitude to our parents who were always beside and helping me in every second in the way of achieving this success. And all others who were with us in every leap and bounds in the process of successions this research. As well, we are thankful for everyone who gave their priceless support and guidance to make this idea a success.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Background literature

During the ancient kingdoms Sri Lanka was leading agrarian societies in the world. Sri Lanka has a great standing history for the agriculture and became paddy cultivation destination at that time. In that time agricultural purposes were narrowed and mainly forcing for the domestic consumptions. Sri Lanka faced significant changes during the colony era in every industry. Therefore agriculture industry also changed.it changed to the trade and international market over consumption basis agriculture. Considering agriculture there are mainly 4 categories we can categorized, Planting, Fishing, livestock and forestry and further we can divided into 16 subcategories for accomplishing better understand. In this research we considered only plantation agriculture.

As mention before agriculture is the one of most important economical industry of the country. In Mahaweli area consider as a dry zone of the country and mainly cultivated rice. Approximately 95 percent of domestic needs cultivated from here. Considering Mahaweli area it devoted form 'Mahaweli Ganga'. In 1977 government decided to further develop the project in order to make country in self-sufficient in rice. They originally decided to cultivate 450,000 acres from 900,000 acres. But due to the irrigation issues Mahaweli project estimate to cultivate about 500,000 acres in dry zone of the land and more than 140,000 families were settled there. In addition 71,000 acres developed in Mahaweli H existing cultivated 80,000 acres land.

The elaboration of the technology for this industry is lack comparing to other industries. Therefore the one of main economical important industry may perform under the expected level. If the technology involved more productivity will be increased. As a final year undergraduate students we decided to introduce new system for Mahaweli project under four aspects. Harvest, Plant healthiness, Market analysis and Trade platform will be developed as our main four components in the system. We hope using our system farmers can cultivate and get an expected income.

There were many researches have done to gather requirements for design the solutions for the forecasting farmer's requirements based on the product details and user experience. Most of them are done by Indians for their agricultural industry. Therefore have found several research papers and blogs related to forecasting farmer's requirements while comparing with this proposed system.

[1] This research paper(2020) were explained about the new way to improve the Indian agriculture with considering of several sectors such as, crop health monitoring, Soilsens, Weather forecasting and supply chain efficiency. According to the Dilksha Manaware, by using AI farmers would be able to clearly understand market demand and also customer's willing according to the seasons. In fact farmers can improve their bottom line by reducing the suffered in managing the logistics and assembly of a middleman as I proposed in my system as well.

[2] This research paper were explained about how the promoting the transformation and upgrading of traditional agriculture and promoting the development of intelligent agriculture. And also this research paper explains according to the current major development needs and in modern world should explore promoting agricultural artificial intelligence fusion.

[3] With the use of machine learning techniques with proper optimizations and fine-tuned selection of the classifying algorithm, a system can be built that considers the data set comprising soil conditions, weather conditions of the past, builds a statistical model through learning and thus seeks to provide accurate and precise decision help with respect to the crops that can be grown profitably during the upcoming period of time.

[4] The purpose of this report is to shed light on the information and knowledge needs in low-income smallholder farms and agricultural micro-enterprises in Bangladesh, India, Sri Lanka and Thailand. The micro-enterprises in this study only include traders, collectors and small retailers that sell agricultural produce. In particular this report also explores the use of Information and Communication Technologies (ICTs) and especially mobile phones amongst these micro-enterprises.

[5] This paper will try to stock market prediction which is the act of trying to predict the future value of a company stock or other financial instrument traded on a financial exchange. The successful stock prediction will maximize investor's gains future price. This paper showed machine learning model to predict stock market price. The proposed algorithm integrates Particle swarm optimization (PSO) and least square support vector machine (LS-SVM). The PSO algorithm is engaged to optimize LS-SVM to predict the stock prices daily. This PSO model is based on the study of stocks historical data and technical measure. PSO algorithm selects best free parameters combination for LS-SVM to avoid over-fitting and local minima problems and improve accuracy of a prediction. The proposed model was applied, and estimate using thirteen benchmark financials datasets and compared with artificial neural network with Levenberg-Marquardt (LM) algorithm. The gained results showed that the proposed model has better prediction accuracy and the potential of PSO algorithm in optimizing LS-SVM.

In order to become more productive in agricultural resources some farmers in worldwide have applied Agricultural information Technology (AIT) [6]. And as sub system of AIT, using technology for agricultural management has a huge impact on managing crop harvesting and in the production side also. It used to analyze features that capture from agriculture data also.

Researches have used to find out numerous ways to capture a difference of a specific crop. As an example, few people have developed a computer system to recognize and classify freeze-dried carrot and microwave vacuumed dry products [7]. In order to implement this system, they have used image processing and artificial neural networks.

1.2. Research Gap

Research Gap for Harvest Prediction

Current trends in machine learning apply to the most advanced software solutions in every nook and corner of the world. But there is a significant drawback with the application of machine learning algorithms in agriculture-based software solutions. Out of the few such intelligent systems, we came across one linear regression model application, one model that doesn't specifically identify forecasting for different seasons. In addition, water use efficiency has been adjusted for the total biomass and grain yield of each important crop. In addition, biomass production and water consumption data were collected and redesigned to assess the potential for generating a comprehensive regression equation related to production. Under all seasonal conditions, water use and total production use is a linear regression model. Hence, we proposed more accurate methodology which is not currently used in any other web or mobile solutions. We propose to introduce multiple models for each season and android application manages the relevant model and run accordingly according to user input. According to this method 2*n number of models are necessary for accurate prediction. (n – Number of crops)

Research Gap for Healthiness of crops

In predicting healthiness or quality of a specific crop after harvesting, there isn't any researches done related to this area even in the world, according to my knowledge. In order to get the idea of what quality their crops are in; farmers need to capture an image and it will predict that crop's healthiness.

But in earlier, people have done researches on capturing a fruit and check whether it is dried or not [3]. This will predict the status of the fruit other than healthiness or the quality. In my proposed component's solution, it will **directly be addressed into the healthiness** of that specific product and will predict answer.

As another example, some researches have done a research on image analyzing using neural network to find out the variety of potatoes [4]. The aim is to classify and keep different varieties of potatoes in heap. And this also, will not aim on the side of predicting the healthiness or the quality of specific crop.

So, currently, there is **no direct platform to predict healthiness** of a specific crop. As the gap defines likewise, I hope this research component will help to predict the healthiness or quality and achieve farmers' requirements in categorizing their crops healthiness wise.

Research Gap for Trade Market Analysis

Currently in Sri Lanka the main way of doing trading is farmers have to go the market physically and sell their products to wholesale buyers or retail buyers.

The existing applications:

Colombo Manning Market has an online trading platform for buyers and sellers. The features of this system are users can purchase well after creating an account, sell products and request products.

Govipala is a mobile application which allows users to only can selling and buying products with user details, product details and contact details.

In above both systems simply do the buy and send products according to the users' choice. When the user comes to their websites the user has to register his/herself and after that user can request for the products, sell products and purchase products with various categorizations of the products.

Features	Existing Platforms for Online Trading				
	OpenFood Network (US)	Govipola (SL)	Colombo Manning Market (SL)	AgriBuzz (India)	Mahaweli Market
Buyer And Seller Recommendation System	✗	✗	✗	✗	✓
Search nearby buyers based on location based data	✗	✗	✗	✗	✓
Showing the product quality with the product	✓	✗	✗	✗	✓
No middlemen	✓	✓	It depends	✓	✓
Mobile Responsiveness	✓	✓	✓	✓	✓
Native language and English language	✓	✓	✓	✓	✓

Figure 1 - Comparison between other applications

As the research gap between the existing ones and my proposed system, I have designed this system to farmers can put their products on the page and wholesale, retailer buyers or customers can purchase them and also users will able to see the recommendations about products based on reviews and ratings. After completing a transaction user can give their idea of this as a comment and give the rating to the product.

The most important feature is in our proposed system is the system will able to show farmers who are retailers' buyer, wholesale buyers and who are nearby buyers. Our proposed system will automatically match buyers' and seller's requirement and make them appear to each other. And, user will be able to search by various parameters about product and search items very easily.

Research Gap for Market Analysis

Currently in Sri Lanka and foreign countries has marketing analysis tool for analyzing the stock market. There are some apps to analyze the stock market price for certain companies stocks are available. MarketWatch, Stock Market Tracker, Stock Exchange are some of example for the market analyze apps. There all are real time apps, shows financial news, market data and investing analysis in major stocks. Some of major stocks; Apple Inc, Bank of America Corporation, Tesla, Amazon, Netflix, American Express Company ect. Through these apps we can get to know the latest update about the stock prices, get real time stock alerts and breaking news to track markets.

Considering the agriculture perspective there are several apps to download and get information about the modern agricultural techniques. 'Krushi Advisor' is a one of popular agricultural app in Sri Lanka. This will provide agriculturally advices to food crop cultivation in Sri Lanka. Mainly this app covers all the information related to the agriculture for an example water management, weed management, fertilizer recommendation, and disease management, suitable locations for particular crop, seed and planting material requirements.

'Crop Farmers App' is another app for summarizing details in crops, fruits and vegetables. This app helps to farmer to summarized the details about the the growing of the crop and fruits and also describes the diseases and pests attacks with the details of symptoms, causes and how they attack and spread to those crops. This app basically acts as a guide to cultivate the crops and fruits. The app shows good practices and techniques farmers can adopt to improve the harvest.

Comparison those facts we can identify the marketing analysis app for the agriculture are rare. There were apps to analyze market and also there were apps to help to agricultural purposes, but the combination of these two are rare. The purposed Marketing analyze tool for the agriculture acts like bridge in between marketing and agriculture to predict the next season trend crops. Weather condition, cultural events, analyzing past marketing data give a result to the farmers. This covers machine learning area and using Time Series algorithms (TS) and Linear Regression algorithms

preceding the results. To make the results more accurate this will convert to the real time app. Because of that adding new data can update the TS module and preceding the more accurate results.

Most of researches about market analysis conduct by using Support Vector Machine (SVM) and Practical Swarm Optimization (PSO). Because of the dynamic behavior of the stock market. Therefore this research can be used as the attempt to proof stock market analysis can be done by Linear Regression algorithm.

2. RESEARCH PROBLEM

Each and every agricultural person lives in Sri Lanka need to use the newest technologies in agricultural side. Nowadays, we cannot see farmers in Sri Lanka using technologies in order to cultivate their crops, increase their harvest, and get the real value for the dedication towards agriculture. Regarding harvest, farmers don't use any harvest predictions in order to enhance the capability to increase their harvest in all seasons. And in healthiness side of the crops, they need to categorize according to various quality levels, because that's how they can divide them into different pricing categories. But nowadays in Sri Lanka, there isn't any king of proper quality checking method for farmers. Instead they just look at harvested crops at eye level and predict a near quality for them. When buying and selling, there aren't any kind of platform which uses new technologies to farmers to find the buyers in order to sell their crops. And also, they don't do any market analysis on order to get the real idea about the current market. Because of these problems farmers face huge disadvantages.

The purpose of the research is to analyze and predict harvest requirements, healthiness of a crop and the marketplace of the specific crop for Mahaweli project farmers in Sri Lanka. And to develop a trade platform for buyers and sellers. Most of the agricultural activities in Sri Lanka are related to the Mahaweli Development Project. Sri Lanka has a long history of irrigated agriculture and now 31.8% of the population is engaged in agriculture. Agriculture and allied sectors such as afforestation and fisheries accounted for 18% of GDP in 2014 and about 26.4% of labor or employment [4]. Hence significant output of agriculture is dependent on Mahaweli development project. Most of the research done for the development of the Mahaweli project and mainly satisfy farmer's needs. But it's clear that farmers' satisfaction is significantly low according to the conclusions of research [3].

3. RESEARCH OBJECTIVES

3.1. Main Objective

The main objective of this research is to develop the Sri Lanka's agriculture with collaborating Mahaweli farmers. And also to provide a healthiness predictions and provide correct predictions to the farmers to manage their works.

3.2. Specific Objectives

- To provide reliable predictions based on the harvest requirements for next year based on past result set.
- To measure the healthiness of some selected crops.
- To provide reliable predictions of available marketplace for given crop.
- To develop a trade platform for crop buying and selling.

4. METHODOLOGY

4.1 Methodology for harvesting

A) Data gathering

Data gathering mainly based totally from direct contact of the Mahaweli Development Authority. The provided official documentation contained all the data that the authority doncucted surveys on and also which are necessary for their own statistical analysis. This report contains information on the progress of socio-economic and environmental activities with the Sri Lankan Mahaweli Authority annually.

Table 3.5: Irrigation Water Issues, Water Duty and Water Productivity in Yala season

	Unit	Yala										
		2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018
System B												
Water Issues	MCM	398.00	193.80	369.10	399.40	323.40	375.50	249.10	385.20	376.90	195.64	308.90
Water Duty	M	2.30	1.68	1.89	2.04	2.17	2.02	2.67	2.05	1.94	1.88	1.56
Water Productivity	Kg/M^{-3}	0.21	0.27	0.26	0.29	0.22	0.21	0.26	0.26	0.26	0.27	0.31
System C												
Water Issues	MCM	433.60	177.10	426.00	513.60	358.60	415.30	328.90	456.30	461.10	217.45	383.44
Water Duty	M	2.02	1.85	1.94	2.31	1.97	1.81	1.44	1.99	2.02	1.73	1.74
Water Productivity	Kg/M^{-3}	0.26	0.28	0.27	0.25	0.23	0.23	0.29	0.28	0.27	0.29	0.31
System G												
Water Issues	MCM	113.80	82.40	122.60	120.90	106.80	137.70	49.10	139.40	100.40	35.71	103.32
Water Duty	M	2.21	2.91	2.33	2.36	1.99	1.96	1.41	1.99	1.88	1.02	1.90
Water Productivity	Kg/M^{-3}	0.23	0.17	0.23	0.22	0.23	0.13	0.33	0.22	0.19	0.50	0.31
System H												
Water Issues	MCM	385.50	227.10	333.40	408.80	302.60	362.70	161.70	265.50	226.50	192.34	281.89
Water Duty	M	1.23	1.72	1.04	1.28	5.53	5.32	0.94	1.01	1.56	1.29	1.42
Water Productivity	Kg/M^{-3}	0.45	0.31	0.52	0.45	0.03	0.17	0.57	0.58	0.36	0.47	0.44
System HU												
Water Issues	MCM								114.70	67.10	287.61	325.55
Water Duty	M								1.30	1.14	4.91	3.75
Water Productivity	Kg/M^{-3}								0.66	0.76	0.18	0.17
System Uw												
Water Issues	MCM	458.90	397.70	495.70	506.30	278.80	420.70	389.30	402.80	537.90	266.90	424.79
Water Duty	M	2.51	2.41	2.51	2.52	3.56	3.77	1.66	1.43	2.48	1.69	2.15
Water Productivity	Kg/M^{-3}	0.33	0.34	0.36	0.33	0.21	0.17	0.58	0.43	0.32	0.50	0.41

Source: Seasonal Summary Report for the Yala Season: Water Management Secretariat

Figure 2 - Irrigation water issues in Yala season

Irrigation water is released to Yala and Maha seasons in the areas of the Mahaweli development authority of Sri Lanka. The capacity of irrigation reservoirs used for agricultural activities as on 31.12.2018 is given in the tables above.

Table 7.4: Paddy and Rice Availability in Mahaweli Areas

Component	2010	2011	2012	2013	2014	2015	2016	2017	2018
Paddy Production Maha (Mt)	507,635	329,808	515,246	471,613	512,522	545,578	541,414	290,816	469,176
Paddy Production Yala (Mt)	421,574	459,423	312,312	378,200	241,280	460,116	343,454	217,011	436,875
Total Paddy Production (Mt)	929,209	789,231	827,558	849,813	753,802	1,005,694	884,868	507,827	906,051
Wastage = 6% (2012 & 2013)	46,460.45	39,462	49,653	50,989	45,228	60,342	53,092	30,470	54,363
Total Extent of Paddy (Ha) Yala+ Maha	172,492	175,113	158,766	161,230	130,626	180,870	158,628	98,931	165,420
Total Seed Paddy Requirement (Mt) (102.35/1000*Total Extent of Paddy)	17,655	17,923	16,250	16,502	13,370	18,512	16,236	10,126	16,931
Availability of Paddy for human Consumption (Mt)	865,094	731,847	761,655	782,322	695,204	926,840	815,540	467,233	834,758
Rice availability for human Consumption (Milling ratio =0.68) (Mt)	588,264	497,656	517,925	531,979	472,739	630,251	554,567	317,718	567,635
Mahaweli Population	864,657	878,152	884,942	905,228	1,001,096	1,007,904	1,128,404	1,140,887	1,116,646
Per Capita Rice Availability in Mahaweli Areas (Kg/Year)	**680**	**567**	**585**	**588**	**472**	**625**	**491**	**278**	**508**

Source: Crop Forecast, Socio Economics & Planning Centre, Department of Agriculture

Figure 3 - Paddy and rise availability in Mahaweli Areas.

According to the table 7.4 above data related to annual and seasonal paddy production can directly obtained. This value first divided by the area under cultivation and obtain its harvest per unit hectare. Then it is used as a data for modelling

B) Modelling

Representation is a linear equation that combines the input value (x) and the predicted output (y) of these input values. Therefore, both the input value (x) and the output value are numbers. The linear equation assigns a scale factor to each input value or column, called the large-scale beta factor (B). An additional coefficient is added, which gives the line more degrees of freedom (for example, moving a two-dimensional shape more and more downwards), and is often referred to as the absorption coefficient or tilt coefficient.

$y = B0 + B1 * x1$

Ridge regression is a sample tuning method used to analyze any data affected by multi-colinearities. This method regulates L2. When there is a problem with multi-colinearities, at least the squares are neutral and the variables are large, resulting in the predicted values being further away than the actual values.

$$\hat{\beta}^{ridge} = \underset{\beta \in \mathbb{R}}{argmin} \|y - XB\|_2^2 + \lambda \|B\|_2^2$$

Figure 4 – Ridge regression coefficient

Based on the above models linear regression model is identified as the model with least error. Hence proceeding with linear regression model for two seasons 'Yala' and 'Maha' and three crops 'Paddy' , 'Potato' and 'onion' six different models trained and converted to tensor flow lite models in a way that it can applied in android applications.

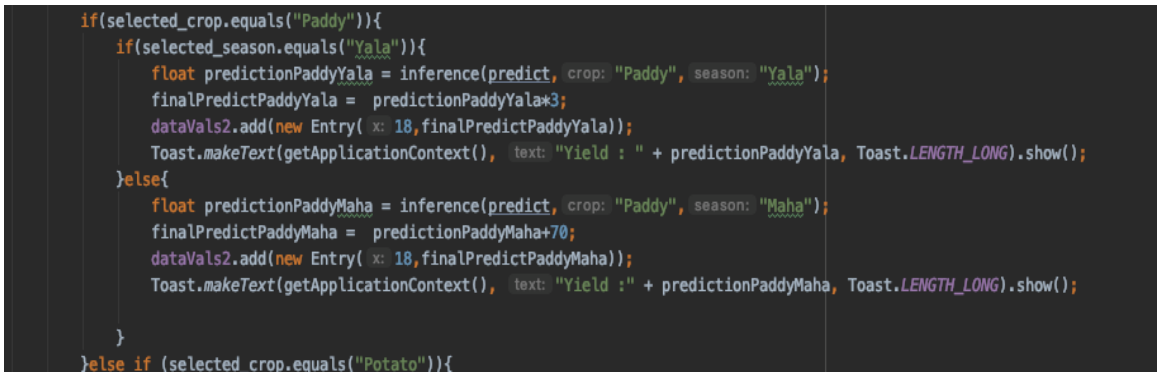
C) Model Integration with Android App

All the linear regression models relevant to crops separately trained using TensorFlow. But TensorFlow models can't be used in android sdk environment. Hence the models are further converted to TensorFlow lite models and added to assets folder in Android project as .tflite.

Figure 5 - Tensorflow lite models in sdk

D) Manipulating set of models in java backend android environment.

Models are varying according to the crop and season that user input. Simultaneously relevant model executes and on plot command and plot the forecast result along with the historical records retrieved from firebase real time database.



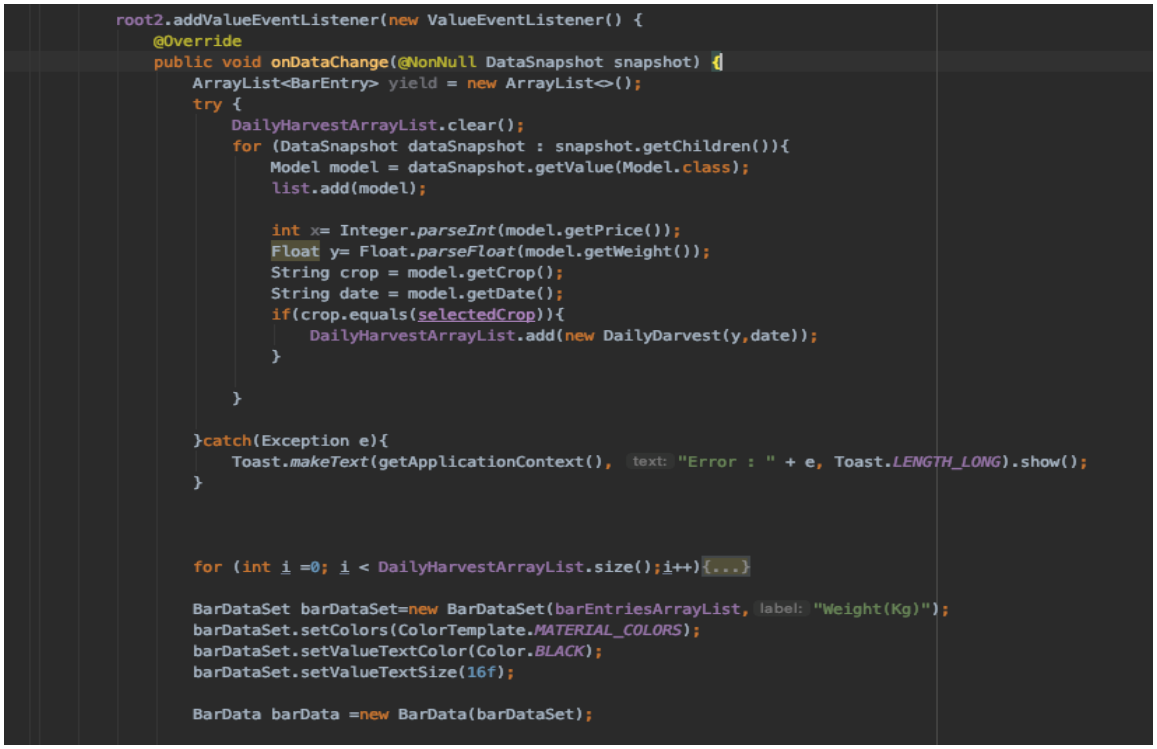
```
if(selected_crop.equals("Paddy")){
    if(selected_season.equals("Yala")){
        float predictionPaddyYala = inference(predict, crop: "Paddy", season: "Yala");
        finalPredictPaddyYala =  predictionPaddyYala*3;
        dataVals2.add(new Entry( x: 18,finalPredictPaddyYala));
        Toast.makeText(getApplicationContext(),  text: "Yield : " + predictionPaddyYala, Toast.LENGTH_LONG).show();
    }else{
        float predictionPaddyMaha = inference(predict, crop: "Paddy", season: "Maha");
        finalPredictPaddyMaha =  predictionPaddyMaha+70;
        dataVals2.add(new Entry( x: 18,finalPredictPaddyMaha));
        Toast.makeText(getApplicationContext(),  text: "Yield :" + predictionPaddyMaha, Toast.LENGTH_LONG).show();

    }
}else if (selected_crop.equals("Potato")){
```

Figure 6 – Handling models in android sdk

This code repeatedly used for all the crops that included in the list. Hence if the number of crops taken as ‘n’ related number of models would be 2*n.

E) Real time selling data plot.

Real time data update and plotting scenario implemented using firebase real-time database. It keeps track of data changes that occur in real time db. Whenever changes occur it retrieve new set of data and plot the graph.



```
root2.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {
        ArrayList<BarEntry> yield = new ArrayList<>();
        try {
            DailyHarvestArrayList.clear();
            for (DataSnapshot dataSnapshot : snapshot.getChildren()){
                Model model = dataSnapshot.getValue(Model.class);
                list.add(model);

                int x= Integer.parseInt(model.getPrice());
                Float y= Float.parseFloat(model.getWeight());
                String crop = model.getCrop();
                String date = model.getDate();
                if(crop.equals(selectedCrop)){
                    DailyHarvestArrayList.add(new DailyDarvest(y,date));
                }

            }

        }catch(Exception e){
            Toast.makeText(getApplicationContext(), text: "Error : " + e, Toast.LENGTH_LONG).show();
        }


        for (int i =0; i < DailyHarvestArrayList.size();i++){...}

        BarDataSet barDataSet=new BarDataSet(barEntriesArrayList, label: "Weight(Kg)");
        barDataSet.setColors(ColorTemplate.MATERIAL_COLORS);
        barDataSet.setValueTextColor(Color.BLACK);
        barDataSet.setValueTextSize(16f);

        BarData barData =new BarData(barDataSet);
```

Figure 7 - Plotting real time data.

F) Weather prediction

Weather prediction for next week is simply done using a request call from open weather API. Users need to provide the location and it converts to latitude and longitude.

```
requestQueue = Volley.newRequestQueue( context: this);
String URL = "https://api.openweathermap.org/data/2.5/onecall?lat="+lat+"&lon="+lng+"&exclude=current,minutely,hourly,alerts&appid=462f44510
```

Figure 8 – API call to open weather API

G) User experience and UI

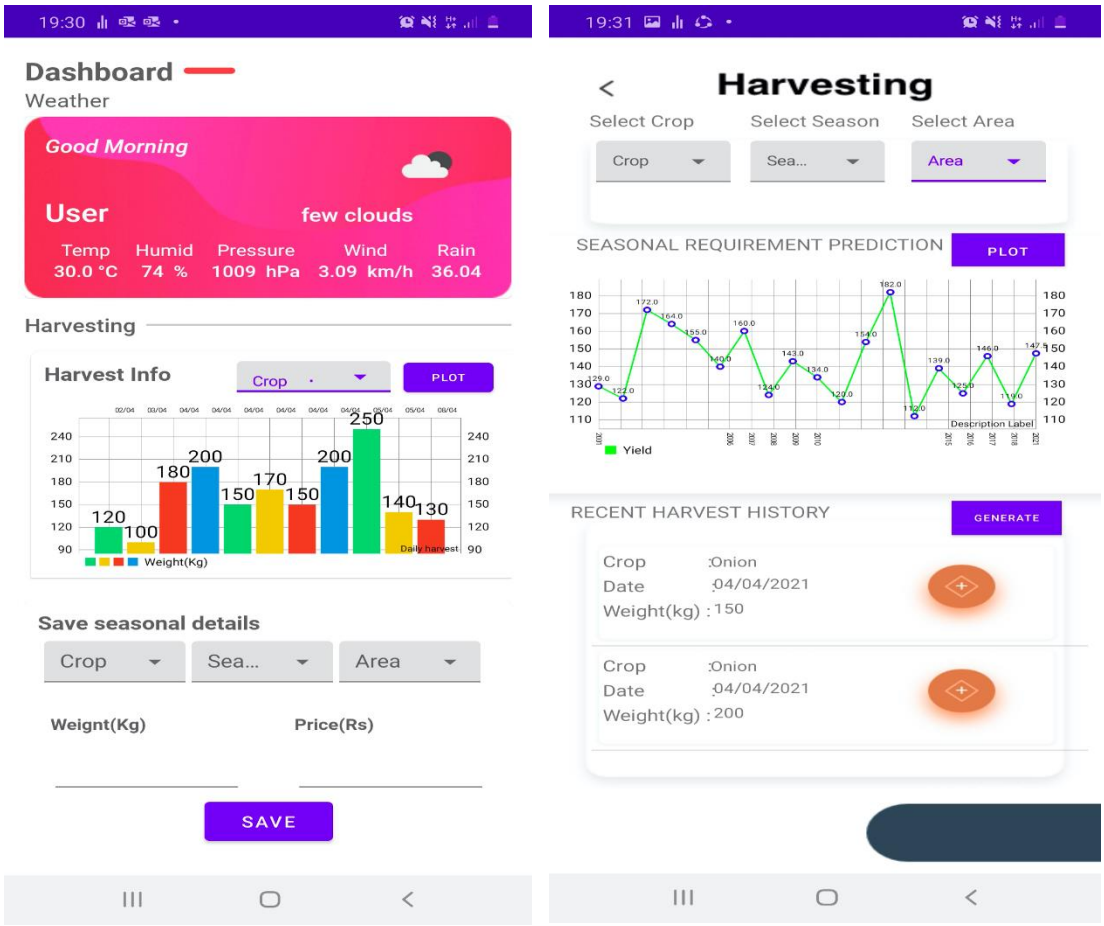


Figure 9 - Weather Prediction interface

4.2 Methodology for addressing healthiness of the crops

First of all, farmer or seller needs to login to the system as a seller. Then from the recycler view of the home page, user needs to select healthiness in order to redirect into the healthiness component. Then by clicking check health button user can move onto the camera component which will predict healthiness levels for potatoes and onions. User only needs to hold the potato or onion to the camera, and the level of healthiness will be displayed at the bottom of the interface. User also can click on 'HEALTHINESS' button as there will be an option on a popup to save the specific healthiness crop with the name of the crop. Later user can refer the savings by going through the view health button, and it will be displayed as a list view with date and time as well. The home page which will be the page to choose healthiness component and the interface that divides two sections is attached below as figures.

a) Android implementation

We developed our system using android technology in order to support our users like farmers, sellers and buyers in user friendliness and easiness to use. In thinking of the user friendliness, we were able to implement the camera component to deliver the result of healthiness level just by holding the phone camera into the desired crop. In order to create the camera component, we implemented the 'OnImageAvailableListner' into the 'CameraActivity'. The camera services are defined as a user-feature as well in the Manifest file. And also in order to access the camera using the mobile device, the permission of the user is required.

In order to achieve the requirement of camera access, we used the user-permission of the camera access in the Manifest file as well as using android.camera.

```
<!-- Camera Features -->
<uses-feature android:name="android.hardware.camera" />
<uses-feature android:name="android.hardware.camera.autofocus" />

<uses-permission android:name="android.permission.CAMERA" />
```

Figure 10 - Camera features in Manifest

After getting the permissions, it is all set to launch the camera. But before launching the camera, the camera fragment is need to be set according to a specific frame size. In order to do that, first camera Id is take from a method that includes the camera services. And a new instance is created from the camera connection fragment. Then using onPreviewSizeChosen method, the preview height and the preview height is need to be set. And using getFragmentManager core function the fragment transaction is started.

```
protected void setFragment() {
  String cameraId = chooseCamera();
  Fragment fragment;
    CameraConnectionFragment camera2Fragment =
        CameraConnectionFragment.newInstance(
            new CameraConnectionFragment.ConnectionCallback() {
              @Override
              public void onPreviewSizeChosen(final Size size, final int rotation) {
                previewHeight = size.getHeight();
                previewWidth = size.getWidth();
                CameraActivity.this.onPreviewSizeChosen(size, rotation);
              }
            },
            imageListener: this,
            getLayoutId(),
            getDesiredPreviewFrameSize());
    camera2Fragment.setCamera(cameraId);
    fragment = camera2Fragment;
    getFragmentManager().beginTransaction().replace(R.id.container, fragment).commit();
}
```

Figure 11 - Set fragment

After all permissions have granted through the application, a preview of the camera can be started. The ‘CameraConnectionFragment’ activity is intended after permission granting. Mainly there are three functions regarding the state of the camera device.

Those are ‘onOpened’, ‘onDisconnected’ and ‘onError’. ‘OnOpened’ is the method that calls the camera API to launch a camera preview to capture the image. It created a new camera capture session for the camera preview. And the preview can be displayed to the user using ‘previewRequestBuilder’. Then from the camera Id, generated from ‘onOpened’ method, open camera method is fired and by creating e new camera manager object that taken from the camera service the camera is launched.

In the camera connection fragment activity, auto focusing of the camera also is handled. As I mentioned above, there is a function called createCameraPreviewSession that creates a new camera capture session for the camera preview.

```
// Auto focus
previewRequestBuilder.set(
    CaptureRequest.CONTROL_AF_MODE,
    CaptureRequest.CONTROL_AF_MODE_CONTINUOUS_PICTURE);
// Flash to be enable when necessary
previewRequestBuilder.set(
    CaptureRequest.CONTROL_AE_MODE, CaptureRequest.CONTROL_AE_MODE_ON_AUTO_FLASH);
```

Figure 12 - Set autofocus

Using the preview request builder service, I defined both autofocus mode and auto focus with continuous picture. And using above function, flashing also set when necessary. Because of these two features, farmers and sellers are able to capture the image of the crop very easily without any trouble on focusing or in low lite.

The deep learning model which needs to predict the level of healthiness is positioned inside the assets folder in the project folder structure. Then there is a classifier class

To access that model with the labels. To access both model and the labels text file, two methods are implemented.

```
@Override
protected String getModelPath()
{
    return "health_unquant.tflite";
}

@Override
protected String getLabelPath()
{
    return "health_labels.txt";
}
```

Figure 13 - Get model and labels

The result which are generated from the model are shown in the bottom sheet which is positioned in the camera interface itself. Recognition value text view is used to set the value into that bottom sheet.

```
protected void showResultsInBottomSheet(List<Recognition> results) {
  if (results != null && results.size() >= 2) {
    Recognition recognition = results.get(0);
    if (recognition != null) {
      if (recognition.getTitle() != null) recognitionTextView.setText(recognition.getTitle());
      if (recognition.getConfidence() != null)
        recognitionValueTextView.setText(
                String.format("%.2f", (100 * recognition.getConfidence())) + "%");
    }
    recognition1TextView.setText("Made by Thamal Wijetunge.");
  }
}
```

Figure 14 - Set result values

b) Deep Learning model implementation

According to the literature survey I have done during the proposal period and further, I have found out that this research component is a huge task which can be done by all four members as a single project. Because of the wide range of the component, I have narrowed out the task by moving to image classification from image processing methods. In order to complete the component using image classification itself, there was a mandatory need of huge dataset which suits to my component as well.

i. Dataset preparation

My research component is to develop a platform to address a level of healthiness for some specific crop in the Mahaweli project. So, I needed to find out some most cultivating crops by Mahaweli farmers. According to the Mahaweli authorized documentation we all chose potato, onion and paddy as our research data. Because of that, I used potato and onion as data to my research component as well. In gathering the dataset, I was not forced to use the data from internet as those image datasets were not that much accurate with the type of crops that cultivated by Mahaweli farmers. In Sri Lanka there is a whole new level of crop cultivation other than foreign countries. Because of that, I didn't use any image data from the internet of any other media source. And, there wasn't any image dataset of the crops they cultivate maintained by the Mahaweli authority as well. Because of this situation, I had to seek out local potatoes and onion for my research component. Any other local farmers' vegetables are much similar to the Mahaweli cultivated vegetables. It was a huge task to find potatoes and onions with different healthiness levels in grocery shops. Anyhow, I bought many potatoes and onions with four different level of healthiness from each of them. Then I captured photos of those potatoes and onions and created more than 3000 images for the training dataset by four levels of healthiness from each one and gathered enough images for the test and valid datasets as well.

ii. Training the model

For python coding and training the deep learning model, I used jupyter notebook associated with anaconda navigator. First I created the variable that assigns train, valid and test paths. So, it will point to the location disk where are different datasets were positioned.

In order to train the deep learning model, first some packages were needed to be imported. Those are numpy, Tensorflow, Keras, Activation, Dense, Flatten, Batch Normalization, Conv2D, MaxPool2D, Adam optimizer, image categorical_crossentropy, ImageDataGenerator, Sequential, confusion_matrix, itertools, matplotlib.pyplot etc.

```
In [1]: import numpy as np
        import tensorflow as tf
        from tensorflow import keras
        from tensorflow.keras.layers import Activation, Dense, Flatten, BatchNormalization, Conv2D, MaxPool2D
        from tensorflow.keras.optimizers import Adam
        from tensorflow.keras.metrics import categorical_crossentropy
        from tensorflow.keras.preprocessing import image
        from tensorflow.keras.preprocessing.image import ImageDataGenerator
        from tensorflow.keras.models import Sequential
        from sklearn.metrics import confusion_matrix
        import itertools
        import os
        import shutil
        import random
        import glob
        import matplotlib.pyplot as plt
        import warnings
        from PIL import Image
```

Figure 15 - Python imports

In implementing the deep learning model, I used the Keras Sequential Model. I needed to put the data in a format that model expects when train the model. When using Keras sequential model the model receives the data whenever the fit function is called. Because of that, the image dataset were put into a format of a Keras generator. First, the train, valid and test batches were created and set them equal to flow from directory followed by image data generator which returns a directory iterator. Basically it creates batches of data from the directories where the datasets aside.

And the batches of data are able to pass to the sequential model using the fit function. To the image data generator, I specified the pre-processing function and set that equal to 'tf.keras.applications.mobilenet.preprocess_input'. This is a function that applies some type of pre-processing on the images before they get passed to the network that uses. By this, I process the images in the dataset in the same format as which images that get passed to the vgg16 model over process.

In the flow from directory function the directory is equals to the path that defines before where dataset locates in the disk. The target size is equal to 100x100. This is the height and width that I wanted the potatoes and onions to be resized. Then the classes were specified which are just the classes for the potential labels of my dataset and the batch size is set to ten.

The exact same thing is done towards the validation set and the test set. The only difference is, in test batch, shuffle parameter is equaled to false. That is because, whenever the test batch is used later for inference to get the model to predict on images of potatoes and onions and after training the validation has been completed, the prediction results is needed to be shown in a confusion matrix. In order to do that, it need to be able to access the unshuffled labels for the test set.

```
In [37]: train_path = 'E:/potato-vs-onion/train'
         valid_path = 'E:/potato-vs-onion/valid'
         test_path = 'E:/potato-vs-onion/test'

In [38]: train_batches = ImageDataGenerator(preprocessing_function=tf.keras.applications.mobilenet.preprocess_input) \
                         .flow_from_directory(directory=train_path, target_size=(100,100), classes=['Onion - 25% Health', 'Onion - 50% Hea
         valid_batches = ImageDataGenerator(preprocessing_function=tf.keras.applications.mobilenet.preprocess_input) \
                         .flow_from_directory(directory=valid_path, target_size=(100,100), classes=['Onion - 25% Health', 'Onion - 50% Hea
         test_batches = ImageDataGenerator(preprocessing_function=tf.keras.applications.mobilenet.preprocess_input) \
                         .flow_from_directory(directory=test_path, target_size=(100,100), classes=['Onion - 25% Health', 'Onion - 50% Heal
```

Figure 16 - Data preprocessing

Then the function plotImages is defined that use to plot the images from the train batches that obtained above. This function is directly taken from tensor flow's website. Next the defined function is used to plot the images from the batches defined earlier and print the corresponding labels for those images.

After plotting the images, there can be seen, what s batch of training data looks like. In there we can see that the color data has been a little bit distorted and that is due to the pre-processing function that called earlier to preprocess the batch images. It skewed the RGB data in some way.

Below the plotted images, the corresponding for the data can be seen. Those are one hot encoded vectors that represent specific healthiness level of potato or onion. Each place that '1' occurs represent each level of healthiness of a crop (the class for the healthiness level).

```
In [40]: def plotImages(images_arr):
             fig, axes = plt.subplots(1, 10, figsize=(20,20))
             axes = axes.flatten()
             for img, ax in zip( images_arr, axes):
                 ax.imshow(img)
                 ax.axis('off')
             plt.tight_layout()
             plt.show()
```

```
In [41]: plotImages(imgs)
         print(labels)
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```

```
[[0. 0. 0. 0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 0. 0. 1. 0.]
 [0. 0. 0. 1. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 1. 0.]
 [0. 0. 0. 1. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 1.]
 [0. 0. 0. 0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 1. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 1. 0. 0.]]
```

Figure 17 - Plot images

To create the convolutional neural network, Keras sequential model is taken as the reference. The sequential model can include number of layers as developer wishes. In my model creation, I included six layers into the sequential model.

The first layer to the model that I passed is a Conv2D layer. This is the standard convolutional layer that accepts image data and to this layer I arbitrarily set the filter value equal to 32 with a kernel size of 3x3. Even though the choice of 32 for the filter value is bit arbitrary, the kernel size of 3x3 is a very common choice for image data. And this Cnv2D layer is followed by the 'relu' activation function and the padding is specified as same. It means that the images will have zero padding to outside that the dimensionality of the images aren't reduced after the convolution operations. And the last parameter for the first layer is the input shape of the data. This Conv2D is actually the first hidden layer as the input layer is made up of the input data itself. And shape of the input data should be mentioned there. As we set the target size parameter in data pre-processing to 100x100, we use that here as well as 100x100x3. The last 3 is regarding the color channels since the images are in RGB format. Then the first convolutional layer is followed by a max pooling layer where the pool size is set to 2 by 2 and strides by 2. This is to cut the image dimensions in half.

After the max pooling layer, another convolutional layer is added is almost same as first convolutional layer excepting the input shape parameter as it is only specified in the firsts hidden layer. And filters is set to 64 instead of 32. Again 64 is an arbitrary choice, but the general rule of increasing functions as layers go until later layer of the network is common practice. Then the second convolutional layer is followed by another max pooling layer identical to the first one.

Then I flatten all of the passing data through layers into a one dimensional tensor before passing it to the dense output layer which has nodes corresponding to the classes that declared about according to the healthiness levels of the potatoes and onions. And the output layer is followed by the softmax activation function which gives probabilities for each corresponding output from the model. After that we can visualize the summary of the created model.

```
In [42]: model = Sequential([
             Conv2D(filters=32, kernel_size=(3, 3), activation='relu', padding = 'same', input_shape=(100,100,3)),
             MaxPool2D(pool_size=(2, 2), strides=2),
             Conv2D(filters=64, kernel_size=(3, 3), activation='relu', padding = 'same'),
             MaxPool2D(pool_size=(2, 2), strides=2),
             Flatten(),
             Dense(units = 8, activation='softmax'),
         ])
```

Figure 18 - Sequential model

```
In [43]: model.summary()
```

```
Model: "sequential_6"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_12 (Conv2D)           (None, 100, 100, 32)      896
_________________________________________________________________
max_pooling2d_12 (MaxPooling (None, 50, 50, 32)        0
_________________________________________________________________
conv2d_13 (Conv2D)           (None, 50, 50, 64)        18496
_________________________________________________________________
max_pooling2d_13 (MaxPooling (None, 25, 25, 64)        0
_________________________________________________________________
flatten_6 (Flatten)          (None, 40000)             0
_________________________________________________________________
dense_6 (Dense)              (None, 8)                 320008
=================================================================
Total params: 339,400
Trainable params: 339,400
Non-trainable params: 0
_________________________________________________________________
```

Figure 19 - Model summary

The model is built at to this point. The model can be prepared for training by calling 'model.compile'. The optimizer is set as equal to the 'Adam' optimizer with a learning rate of 0.0001. To measure loss I used categorical_crossentropy and the accuracy as the metrics to be able to judge the model performance. Categorical_crossentropy is used because I can't use binary_crossentropy as there are more than two outputs. Using categorical_crossentropy and using softmax activation function for the output layer are done because those are the common approaches when there are more than two classes according to the research I have done.

After compiling the model, the model is trained using 'model.fit' function. To the fit function the first specified parameter is the training data which is stored in the train batches. Then the validation data is specified, which is stored in valid batches. I didn't use validation split, because I already created a validation set separately before fitting the model. Then epochs is equaled to 10 as I needed to run it 10 times. And the verbose is equaled to 2, because most verbose output can be seen during training.

The reason why I didn't specified 'y' is, when data is stored as generator as I done in early stages, the generator itself contains the corresponding labels. After finishing the result, by tenth epoch, the accuracy of the training set has reached 100% but the validation accuracy is calculated as around 98%. As a whole with the summary of the trained set, by the tenth epoch model has reached a good position to predict results.

```
In [44]: model.compile(optimizer=Adam(learning_rate=0.0001), loss='categorical_crossentro

In [45]: model.fit(x=train_batches, validation_data=valid_batches, epochs=10, verbose=2)
Epoch 1/10
160/160 - 17s - loss: 0.8460 - accuracy: 0.7362 - val_loss: 0.2473 - val_accura
cy: 0.9525
Epoch 2/10
160/160 - 16s - loss: 0.1173 - accuracy: 0.9837 - val_loss: 0.0679 - val_accura
cy: 0.9950
Epoch 3/10
160/160 - 16s - loss: 0.0364 - accuracy: 0.9994 - val_loss: 0.0323 - val_accura
cy: 1.0000
Epoch 4/10
160/160 - 16s - loss: 0.0148 - accuracy: 1.0000 - val_loss: 0.0153 - val_accura
cy: 0.9975
Epoch 5/10
160/160 - 16s - loss: 0.0082 - accuracy: 1.0000 - val_loss: 0.0095 - val_accura
cy: 1.0000
Epoch 6/10
160/160 - 16s - loss: 0.0045 - accuracy: 1.0000 - val_loss: 0.0077 - val_accura
cy: 1.0000
Epoch 7/10
160/160 - 17s - loss: 0.0032 - accuracy: 1.0000 - val_loss: 0.0055 - val_accura
cy: 1.0000
Epoch 8/10
160/160 - 17s - loss: 0.0021 - accuracy: 1.0000 - val_loss: 0.0060 - val_accura
cy: 0.9975
Epoch 9/10
160/160 - 17s - loss: 0.0017 - accuracy: 1.0000 - val_loss: 0.0055 - val_accura
cy: 0.9975
Epoch 10/10
160/160 - 18s - loss: 0.0013 - accuracy: 1.0000 - val_loss: 0.0045 - val_accura
cy: 0.9975

Out[45]: <tensorflow.python.keras.callbacks.History at 0x166cdb2a2e0>
```

Figure 20 - Training the model

After training the model, the thing I had to do is test the model using test data. The test results will be explained in the testing and results section and the implementation part of the testing will be covered here.

First thing I had done was, get a batch of test data from the test batches. And this also can be plotted using plotImages function that I mentioned above. And the section test_batches.classes is implemented to get a array that has all of the corresponding

labels for each image in the test set. Because we pass the shuffle parameter into false in the test set earlier, I was able have the one-one direct mapping from the unshuffled labels to the test dataset and if true was passed to shuffle parameter of test dataset every time a batch was generated, then I wouldn't be have the correct mapping between labels and samples. I cared about having the correct mapping because later after I get the predictions from the model, I had plotted the predictions into a confusion matrix and so I wanted the corresponding labels that belongs to the samples in the test set.

```
In [47]: test_batches.classes

Out[47]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1,
                1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2,
                2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3,
                3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4,
                4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
                5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
                6, 6, 6, 6, 6, 6, 6, 6, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7,
                7, 7, 7, 7, 7, 7])
```

Figure 21 - Classes of test batches

The next step is to go ahead and obtain the predictions by calling model.predict. To the parameter 'x' I had specified the test batches of all the test dataset. And the verbose is chosen to be 0 to get no output whenever the predictions are run. The predictions can be printed as the rounded predictions of the model. And again the predictions can be passed to the plot confusion matrix and view the test results in a more familiar way. These parts will be covered in the testing and results sections.

A tool named confusion matrix is used here to visualize the results. The confusion matrix is created using confusion_matrix function from scikit-learn. For the first parameter I have passed the true labels using test batches classes and for the predicted label parameter, I have passed the predictions from the model. And the argmax is used to pass in the index of where the most probable prediction was from the prediction list.

Next the plot confusion matrix function is implemented which is directly taken from scikit-learn's website. This allowed to plot the confusion matrix.

Then the class indices are declared as the 8 classes I have used for 4 level of healthiness used for each potato and onion. Using them the plot labels are defined for the confusion matrix.

```
In [50]: cm = confusion_matrix(y_true=test_batches.classes, y_pred=np.argmax(predictions, axis=-1))
```

```
In [51]: def plot_confusion_matrix(cm, classes,
                                  normalize=False,
                                  title='Crop Identification confusion matrix',
                                  cmap=plt.cm.Blues):
             plt.imshow(cm, interpolation='nearest', cmap=cmap)
             plt.title(title)
             plt.colorbar()
             tick_marks = np.arange(len(classes))
             plt.xticks(tick_marks, classes, rotation=45)
             plt.yticks(tick_marks, classes)

             if normalize:
                 cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
                 print("Mormalized confusion matrix")
             else:
                 print('Confusion matrix, without normalization')

             print(cm)

             thresh = cm.max() / 2.
             for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
                 plt.text(j, i, cm[i, j],
                          horizontalalignment="center",
                          color="white" if cm[i, j] > thresh else "black")

             plt.tight_layout()
             plt.ylabel('True label')
             plt.xlabel('Predicted label')
```

```
In [52]: test_batches.class_indices

Out[52]: {'Onion - 25% Health': 0,
          'Onion - 50% Health': 1,
          'Onion - 75% Health': 2,
          'Onion - 100% Health': 3,
          'Potato - 25% Health': 4,
          'Potato - 50% Health': 5,
          'Potato - 75% Health': 6,
          'Potato - 100% Health': 7}
```

Figure 22 - Plot confusion matrix define

4.3 Methodology for Market Analysis

a) Description of experimental area

The related data for the model creation gathered from "Mahaweli Authority of Sri Lanka" form 2013 to 2018. According to the data average temperature is 31°C and the average rainfall is 500mm. Evaluation pressure of the tested period of 800 hPa. Consider the soil texture Mahaweli area has mixed soil. All related irrigation, fertilization, pest and disease control were performed when it's necessary.

Considering the Stock market prices depend on the harvest that obtains in each season. To calculate the harvest using below equation.

$$\delta = 0.3964 + 3.631sin\left(\frac{360D}{365}\right) - 22.97cos\left(\frac{360D}{365}\right) + 0.03838sin\left(2\frac{360D}{365}\right) - 0.3885cos\left(2\frac{360D}{365}\right) + 0.07659sin\left(3\frac{360D}{365}\right) - 0.1587cos\left(3\frac{360D}{365}\right) - 0.01021cos\left(4\frac{360D}{365}\right)$$

Figure 23 – Harvest calculation equation

δ = sun declination
D = number of the days (1, 2, 3….. 365)

b) Linear Regression model

From 2013 to 2018 samples of crops stock market prices were recorded. Below table shows the particular data stored to the train the model.

	A	C	D	E	F	G	H	I	J	K	L	M	N	O
1		date	close	high	low	open	volume	adjClose	adjHigh	adjLow	adjOpen	adjVolume	divCash	splitFactor
14	12	2015-06-12 00:00:	127.17	128.33	127.11	128.185	36886246	117.1901309	118.259098	117.1348395	118.1254772	36886246	0	1
15	13	2015-06-15 00:00:	126.92	127.24	125.71	126.1	43988946	116.9597501	117.254638	115.8447068	116.2041009	43988946	0	1
16	14	2015-06-16 00:00:	127.6	127.85	126.37	127.03	31494131	117.586386	117.816767	116.4529122	117.0611177	31494131	0	1
17	15	2015-06-17 00:00:	127.3	127.88	126.74	127.72	32918071	117.309929	117.844413	116.7938759	117.6969688	32918071	0	1
18	16	2015-06-18 00:00:	127.88	128.31	127.22	127.23	35407220	117.8444126	118.240668	117.2362071	117.2454224	35407220	0	1
19	17	2015-06-19 00:00:	126.6	127.82	126.4	127.71	54716887	116.6648626	117.789121	116.4805579	117.6877536	54716887	0	1
20	18	2015-06-22 00:00:	127.61	128.06	127.08	127.49	34039345	117.5956012	118.010287	117.1071938	117.4850184	34039345	0	1
21	19	2015-06-23 00:00:	127.03	127.61	126.8792	127.48	30268863	117.0611177	117.595601	116.9221519	117.4758032	30268863	0	1
22	20	2015-06-24 00:00:	128.11	129.8	127.12	127.21	55280855	118.0563629	119.613737	117.1440548	117.2269919	55280855	0	1
23	21	2015-06-25 00:00:	127.5	129.2	127.5	128.86	31938100	117.4942337	119.060823	117.4942337	118.7475055	31938100	0	1
24	22	2015-06-26 00:00:	126.75	127.99	126.51	127.67	44066841	116.8030911	117.94578	116.5819255	117.6508927	44066841	0	1
25	23	2015-06-29 00:00:	124.53	126.47	124.48	125.46	49161427	114.7573092	116.545065	114.711233	115.6143259	49161427	0	1
26	24	2015-06-30 00:00:	125.425	126.12	124.86	125.57	44370682	115.5820726	116.222531	115.0614119	115.7156935	44370682	0	1
27	25	2015-07-01 00:00:	126.6	126.94	125.99	126.9	30238811	116.6648626	116.978181	116.1027333	116.9413196	30238811	0	1
28	26	2015-07-02 00:00:	126.44	126.69	125.77	126.43	27210952	116.5174189	116.7478	115.8999982	116.5082036	27210952	0	1
29	27	2015-07-06 00:00:	126	126.23	124.85	124.94	28060431	116.1119486	116.323899	115.0521967	115.1351338	28060431	0	1

Figure 24 – Table of model data

A numerical demonstrate is of regression when it interfaces the behavior of a variable in function of another. A regression demonstrate that contains more than one regressor variable is named different, and is broadly used to alter the organized information in a linear frame with the obscure coefficients that take after the regressor variables.

According to that the relation between above variable and the stock market prices calculation will be generating according to this equation.

$$\delta = \big((high - low) \div (close - open)\big) * volume\ \big(\ (ao - ac) + (ah - al)\big)av + splitFactor + \in$$

Figure 25 – Market price calculation equation

δ = Predicted price
high = high price consider the period range
low = low price consider the price range
close = close price
open = open price of the crop
ao = adjOpen
ac = adjClose
ah = adjHigh
al = adjLow
av = adjVolume
splitFactor = price of the split range of the crop
€ = random error term

In expansion, it is vital to choose an suitable set of regressors, based on a model that incorporates all variables, and not all regressors are essential. Hence, an examination ought to be consider to select an appropriate demonstrate that contains adequate regression factors for forecast, of low support cost and easy to utilize.

Many criteria can be used for perform this task. In here I used high, low price of the particular crop, close and open price of the crop, adjOpen, adjClose, adjHigh, adjLow, adjVolume to predict the market price of the specific crop.

To alter the models, the centrality level of the regressive factors was utilized at 5% and concomitant to the open value, which alludes to the most reduced level of centrality for acknowledgment of the model received.

In arrange to have a common thought of the precision of the forecasts of the received model, it is essential to decide the confidence intervals (CI), which gives the likely values for the normal reaction and too the prediction intervals (PI) that comes about within the likely values for an reply to an x not belonging to the analyzed information, but which have a place to the extent of variety examined

c) Mathematical model to predict stock market

Having a show that employments a single linear regression function, in differentiate to one created beneath the strategies of neural systems (with higher number of regressive capacities) is curiously and computationally lighter, in any case considers of numerical modeling connected to farming are still beginning. From the existing regression models, there are those classified as linear and non-linear models, the final being valuable for depicting the development of natural materials over time, as they utilize parameters of natural translation that encourage examination.

To help the Linear regression model to analyzed crop prices data follow below steps,

I. At first, the assessment of the finest subsets to compose the show was assessed, utilizing the Collaborator tool
II. At that point, through an examination of the ordinary probability of the buildups charts, (residues of x balanced values and residues x arrange of perception), two exceptions were found, that's , information that were outside normality which were causing misfortunes within the translation of comes about of the factual tests connected to the samples.
III. At last, a new modeling was performed, coming about within the condition displayed with the outcomes gotten from the expectation coefficients.

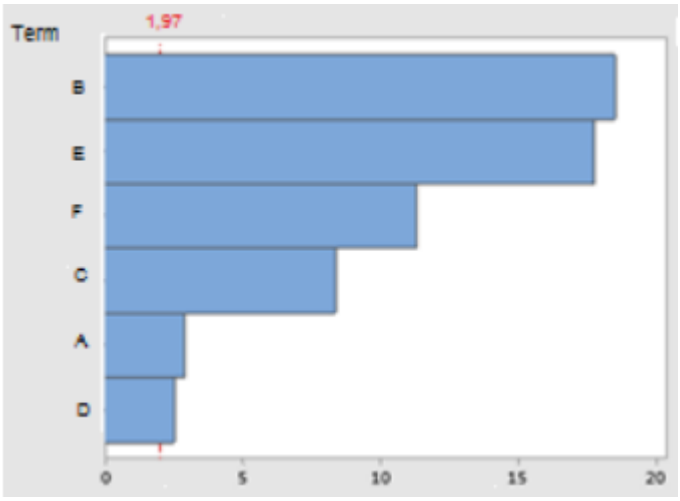


Figure 26 - Pareto chart of standardized effects

From the Pareto's chart of the impacts it was conceivable to compare the relative magnitude and statistical significance of the terms, in diminishing arrange of the absolute values, Above figure mention the standard effect of the model used. With the reference line highlighted in red showing the critical terms for the model (with a centrality level of 5%, i.e. α = 0.05), whose t esteem (tα/2) - utilized to decide the certainty and forecast interims - was 1.97.

Concurring to the comes about appeared chart, the factors chosen for the demonstrate are factually noteworthy, considering that they display values over the reference esteem, with the month (B) being the foremost conspicuous (18.5232), due to the variety the sun declination over the days of the year that influence the necessary photoperiod for crops in realizing photosynthesis; the emphasis month and photoperiod (E) was the moment most imperative calculate (17.8005); and the photoperiod raised to the moment control (F) was the third highlighted (11.3096).

Besides, it is vital that the demonstrate displayed S values break even with to 10.808, Mallows' Cp of 7.0 and open-values less than 5%, as the level of centrality utilized in this venture was α = 5% and with regard to Evans run the show (n/k be at slightest 10), we have that n=362 and k=6, whose relationship come about in 60.33, that's, the values found are palatable for the utilize of the relapse equation.

In order to confirm in case to demonstrate proposed in this consider meets the suspicions of the examination, the charts of ordinary probability of the residues, residues x order, histogram of residues and residues x adjustments were proved. It will mention below figure.

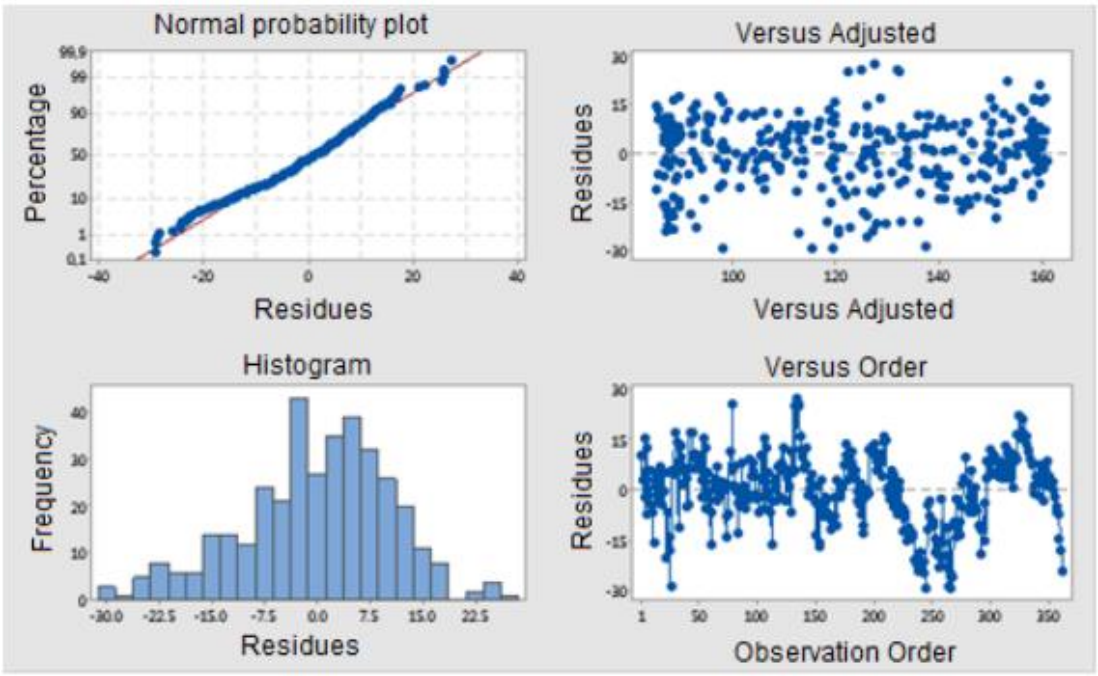


Figure 27 - Graphs of Gestation residue

The typical probability plot of the residues (upper left side of Figure 3.3.3) was utilized to confirm the suspicion of dissemination of these in arrange to take after roughly a straight line, hence confirming that there's no prove of variation from the norm, exceptions or unidentified variables.

because of the huge number of information focuses (n = 362), the buildups histogram (lower left side of Figure 3.3.3) proved to be successful, where each bar contained sufficient focuses to dependably outline the asymmetry.

As for the chart of residues x adjusted values (upper right side of Figure 3.3.3), it was discovered that the residues were haphazardly conveyed on both sides along line 0, with no identified designs within the focuses.

In any case, in a point by point examination, it was identified that 22 focuses had huge residues and were not perfectly balanced, which are stamped in red within the chart in Figure below, which conceivably are given due to the delay within the collection of bunches, without noteworthy misfortunes for the clarification of the demonstrate.

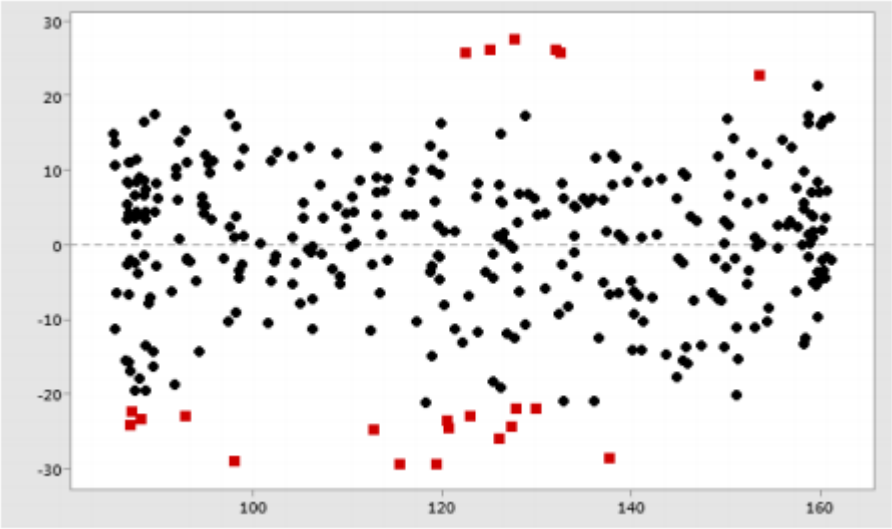


Figure 28 - Graph of Adjusted Values x Waste

At last, the chart of residues x order of collection (lower right side of Figure 3), shows that the residues have palatable freedom from each other, by showing themselves haphazardly around the central line.

With a importance level of 5%, i.e. α=0.05 and t value (tα/2) of 1.97, the certainty and forecast intervals were separately ± 4.88 and ± 21.30, and the comes about displayed within the application are related to the expectation.

In arrange to have a common thought of the precision of the forecasts of the received model, it is essential to decide the confidence intervals (CI), which gives the likely values for the normal reaction and too the prediction intervals (PI) that comes about within the likely values for an reply to an x not belonging to the analyzed information, but which have a place to the extent of variety examined

Besides, it is vital that the demonstrate displayed S values break even with to 10.808, Mallows' Cp of 7.0 and open-values less than 5%, as the level of centrality utilized in this venture was $\alpha = 5\%$ and with regard to Evans run the show (n/k be at slightest 10), we have that n=362 and k=6, whose relationship come about in 60.33, that's, the values found are palatable for the utilize of the relapse equation.

4.4 Methodology for Online Trade Platform

a. Recommending the best farmers based on the rates, nearby locations and gross net of the products.

Basically, Recommending systems are categorized into collaborative filtering, content based, utility based, demographic-based, knowledge-based, and hybrid-based. In these days the most using filtering approaches are content based and collaborative based filtering.

In the collaborative based filtering identifies the products user's ratings and from historical data which works by developing a database of the user's preferences for each items. Collaborative filtering are classified into item-based filtering and user-based filtering.

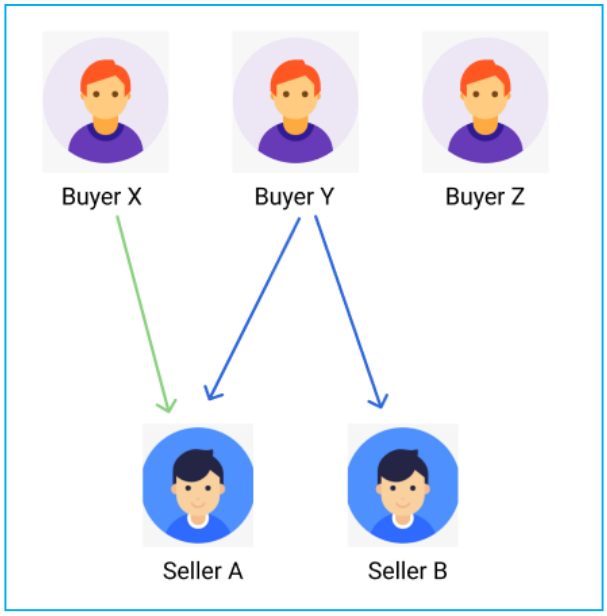


Figure 29 - Collaborative based filter

The sample scenario I have applied in this is that if we assume that buyer x buy only from seller A and Buyer B but both from seller A and B. So eventually system will recommend the seller B to Buyer X. Main attributes are the user specific buying history details of products and seller details.

Content based recommending system where features values must be manually entered. This can be manageable for small datasets, but when thousands of new products are being added daily, this task is impossible. Content-based filtering does not require other users' data, as the predicted recommendations are user-specific. These techniques measure up the system to holder many users. Content-based filtering is user-independent since this system only requires analyzing the items and user profile for recommendations.

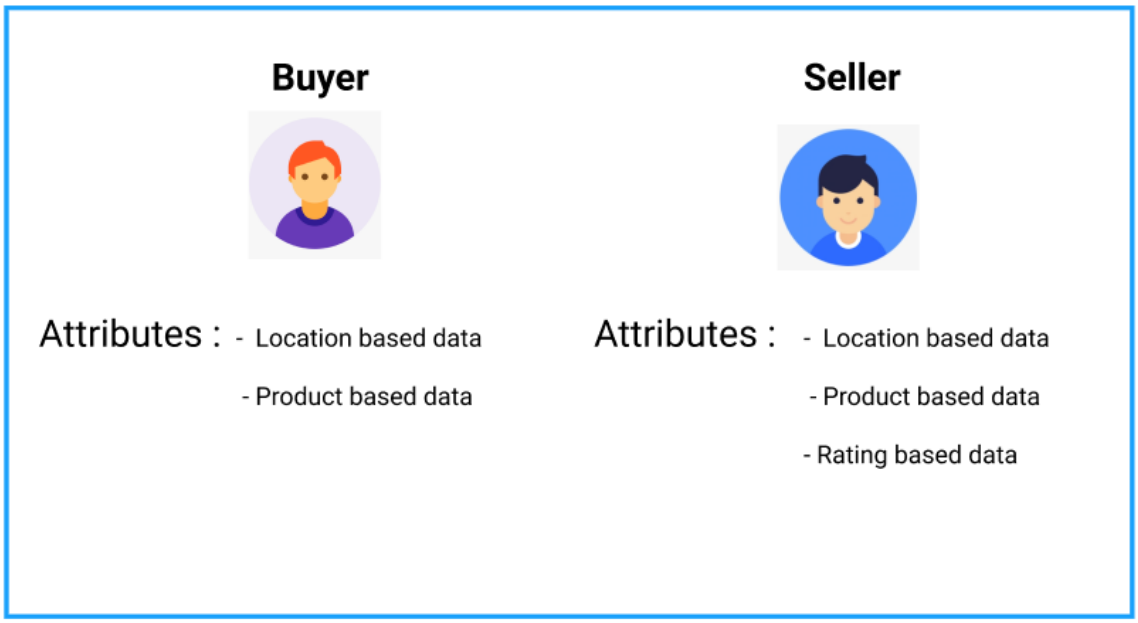
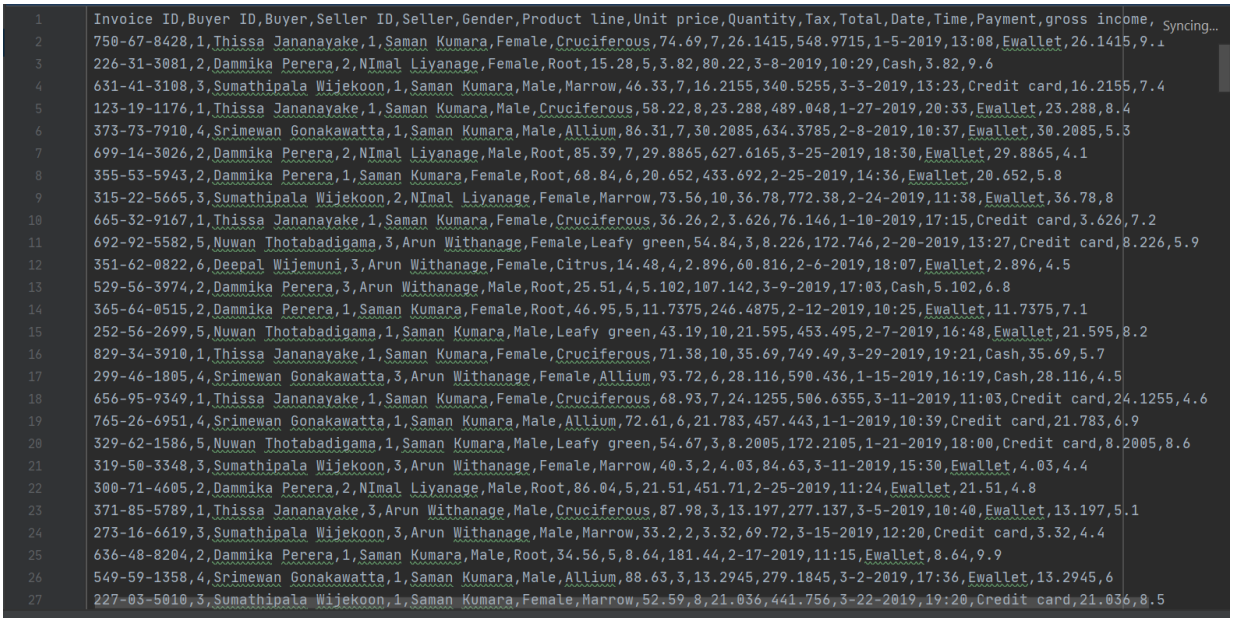


Figure 30 - Content based filter

In the figure 3.3.2 shown when I consider the content based recommendation algorithm I have considered the location based data and product based data as attributes from buyer aspect and location based data, product based data and rating based data from the farmer's aspect.

The main and important point of this research was to implement the recommendation system for buyers to select the best farmers in the Mahaweli. As a first task to implement this recommending system I have found the dataset from mahaweli authority and trained the dataset using machine learning algorithm. I have considered main 4 factors to implement this recommending system which are farmer's selling product, farmer's rate, location details and farmer's gross net.

As a first task I have created a dataset with all related data considering farmer's products, name, gross selling rates, ratings, invoice id, buyer, seller id, product line, unit price, quantity, tax, total, date, time, payment, gross income and also gender. This dataset consists data more than 1000 records.



```
Invoice ID,Buyer ID,Buyer,Seller ID,Seller,Gender,Product line,Unit price,Quantity,Tax,Total,Date,Time,Payment,gross income,
750-67-8428,1,Thissa Jananayake,1,Saman Kumara,Female,Cruciferous,74.69,7,26.1415,548.9715,1-5-2019,13:08,Ewallet,26.1415,9.
226-31-3081,2,Dammika Perera,2,NImal Liyanage,Female,Root,15.28,5,3.82,80.22,3-8-2019,10:29,Cash,3.82,9.6
631-41-3108,3,Sumathipala Wijekoon,1,Saman Kumara,Male,Marrow,46.33,7,16.2155,340.5255,3-3-2019,13:23,Credit card,16.2155,7.4
123-19-1176,1,Thissa Jananayake,1,Saman Kumara,Male,Cruciferous,58.22,8,23.288,489.048,1-27-2019,20:33,Ewallet,23.288,8.4
373-73-7910,4,Srimewan Gonakawatta,1,Saman Kumara,Male,Allium,86.31,7,30.2085,634.3785,2-8-2019,10:37,Ewallet,30.2085,5.3
699-14-3026,2,Dammika Perera,2,NImal Liyanage,Male,Root,85.39,7,29.8865,627.6165,3-25-2019,18:30,Ewallet,29.8865,4.1
355-53-5943,2,Dammika Perera,1,Saman Kumara,Female,Root,68.84,6,20.652,433.692,2-25-2019,14:36,Ewallet,20.652,5.8
315-22-5665,3,Sumathipala Wijekoon,2,NImal Liyanage,Female,Marrow,73.56,10,36.78,772.38,2-24-2019,11:38,Ewallet,36.78,8
665-32-9167,1,Thissa Jananayake,1,Saman Kumara,Female,Cruciferous,36.26,2,3.626,76.146,1-10-2019,17:15,Credit card,3.626,7.2
692-92-5582,5,Nuwan Thotabadigama,3,Arun Withanage,Female,Leafy green,54.84,3,8.226,172.746,2-20-2019,13:27,Credit card,8.226,5.9
351-62-0822,6,Deepal Wijemuni,3,Arun Withanage,Female,Citrus,14.48,4,2.896,60.816,2-6-2019,18:07,Ewallet,2.896,4.5
529-56-3974,2,Dammika Perera,3,Arun Withanage,Male,Root,25.51,4,5.102,107.142,3-9-2019,17:03,Cash,5.102,6.8
365-64-0515,2,Dammika Perera,1,Saman Kumara,Female,Root,46.95,5,11.7375,246.4875,2-12-2019,10:25,Ewallet,11.7375,7.1
252-56-2699,5,Nuwan Thotabadigama,1,Saman Kumara,Male,Leafy green,43.19,10,21.595,453.495,2-7-2019,16:48,Ewallet,21.595,8.2
829-34-3910,1,Thissa Jananayake,1,Saman Kumara,Female,Cruciferous,71.38,10,35.69,749.49,3-29-2019,19:21,Cash,35.69,5.7
299-46-1805,4,Srimewan Gonakawatta,3,Arun Withanage,Female,Allium,93.72,6,28.116,590.436,1-15-2019,16:19,Cash,28.116,4.5
656-95-9349,1,Thissa Jananayake,1,Saman Kumara,Female,Cruciferous,68.93,7,24.1255,506.6355,3-11-2019,11:03,Credit card,24.1255,4.6
765-26-6951,4,Srimewan Gonakawatta,1,Saman Kumara,Male,Allium,72.61,6,21.783,457.443,1-1-2019,10:39,Credit card,21.783,6.9
329-62-1586,5,Nuwan Thotabadigama,1,Saman Kumara,Male,Leafy green,54.67,3,8.2005,172.2105,1-21-2019,18:00,Credit card,8.2005,8.6
319-50-3348,3,Sumathipala Wijekoon,3,Arun Withanage,Female,Marrow,40.3,2,4.03,84.63,3-11-2019,15:30,Ewallet,4.03,4.4
300-71-4605,2,Dammika Perera,2,NImal Liyanage,Male,Root,86.04,5,21.51,451.71,2-25-2019,11:24,Ewallet,21.51,4.8
371-85-5789,1,Thissa Jananayake,3,Arun Withanage,Male,Cruciferous,87.98,3,13.197,277.137,3-5-2019,10:40,Ewallet,13.197,5.1
273-16-6619,3,Sumathipala Wijekoon,3,Arun Withanage,Male,Marrow,33.2,2,3.32,69.72,3-15-2019,12:20,Credit card,3.32,4.4
636-48-8204,2,Dammika Perera,1,Saman Kumara,Male,Root,34.56,5,8.64,181.44,2-17-2019,11:15,Ewallet,8.64,9.9
549-59-1358,4,Srimewan Gonakawatta,1,Saman Kumara,Male,Allium,88.63,3,13.2945,279.1845,3-2-2019,17:36,Ewallet,13.2945,6
227-03-5010,3,Sumathipala Wijekoon,1,Saman Kumara,Female,Marrow,52.59,8,21.036,441.756,3-22-2019,19:20,Credit card,21.036,8.5
```

Figure 31 – Trade platform dataset

Since I have more than 1000 records in my dataset, I have divided my dataset into 6 main categories;

- Allium.csv
- Citrus.csv
- Cruciferous.csv
- Leafy green.csv
- Marrow.csv
- Root.csv

```python
@app.route('/best_seller', methods=['POST'])
def best_seller():
    product = request.form['product']
    sale = Seller.Sale('Dataset/dataset.csv', product)
    # sale = Seller.Sale('Dataset/dataset.csv', 'Citrus')
    data = sale.analyze()
    return_json = []

    for p in data:
        print(p)
        return_json.append('{ "seller_name" : "' + str(p[1]) + '", "rating" :' + str(round(p[0], 2)) + '}')

    temp_start = '{ "data" :['
    temp_end = ']}'

    return_j = temp_start

    for x in return_json:
        if x == return_json[-1]:
            return_j += x
        else:
            return_j += x + ','
    return_j += temp_end

    return json.loads(return_j)
```

Figure 32 - Best seller API

b. Predict the best crops to harvest for next week or next month.

My other core objective is to predict the best crops to harvest for next month or year. This function will be more important in farmer's aspect because by using this function farmers can get idea of crops which are best to harvest in the future to get the better market sale.

For this I needed the sale prediction data set which is like historical data along with the farmer's details. Data from past years can help to predict the budget sales in future days.

So, for this implementation I have created the dataset called predictedOutput.csv which is consist of the data of the parameters I have considered to predict the best crop for harvest in the next month. The parameters are ds, trend, ,ds,trend,yhat_lower,yhat_upper,trend_lower,trend_upper,additive_terms,additive_terms_lower,additive_terms_upper,weekly,weekly_lower,weekly_upper,multiplicative_terms,multiplicative_terms_lower,multiplicative_terms_upper,yhat.

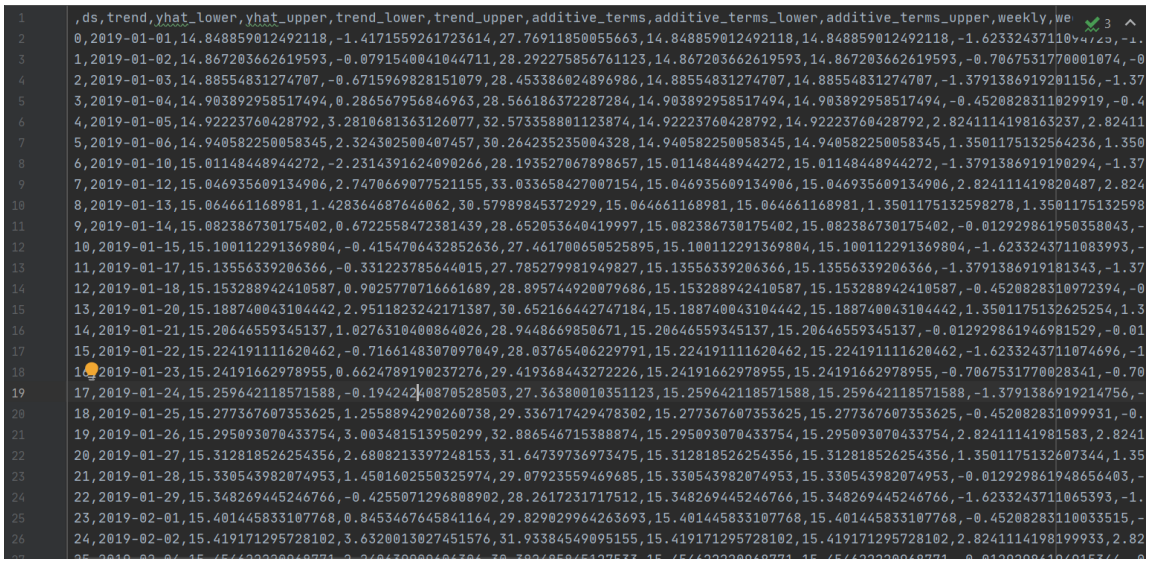


Figure 33 – Predicted dataset

I’m using the time series algorithms for prediction and the data is organized by relatively deterministic timestamps and compared to random sample data, contain the additional information we can extract. Time series can help with forecasting the finds in data analytics. At the beginning I was considered about the two time series algorithms which are Arima (Autoregressive Integrated Moving Average) and Prophet Algorithm.

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

```
@app.route('/predict_product', methods=['POST'])
def predict_product():
    days = request.form['days']
    days = int(days)
    predicted_v = []
    return_json = []
    dirs = os.listdir('FBP_Data')

    for d in dirs:
        fb = FBP.FB(d)
        if days == 7:
            predicted_v.append((fb.get_week(), d.replace('.csv', '')))
        else:
            predicted_v.append((fb.get_month(), d.replace('.csv', '')))
    print('Predicted Data---')
    for p in predicted_v:
        print(p)
    predicted_v.sort()
    predicted_v.reverse()
    print('Sorted Data---')
    for p in predicted_v:
        print(p)
        return_json.append('{ "product_name" : "' + str(p[1]) + '", "count" :' + str(round(p[0], 2)) + '}')

    temp_start = '{ "data" :['
    temp_end = ']}'
```

Figure 34 – Best crop API

c. Implementation of Android Application with UI/UX aspects.

Since this system will be mainly used by the farmers we had to consider the simplest UI/UX aspects and we have implemented the 3 languages which are 3 major languages in the Sri Lanka. We have implemented the android application with Sinhala, Tamil and English. Our system has 4 main component which are Harvest prediction, measure the Crop Healthiness, Market analysis and online trade platform. There will be 2 main users' category in our system; one category is Farmers or Sellers and other category is the Buyers. This is the user flow of our developed system; Online trade platform flow.

As a summary of my component which is online trade platform there is a user flow;

First user needs to select the one option from 3 languages and user has to log in to the system. Then user has to choose the option which is the user will be logged in as farmer or buyer. If the user selected the farmer option user can logged in as the farmer and select the trade platform and see the crop will be best to harvest in next month or next year according to the prediction which is given by the backend algorithm.

If the user selected the user type as buyer user logged in as buyer and he or she can select one of the crop categories shown in the page and when he or she clicks the find best seller buttons system shoes the farmers who has the highest ratings on that product.

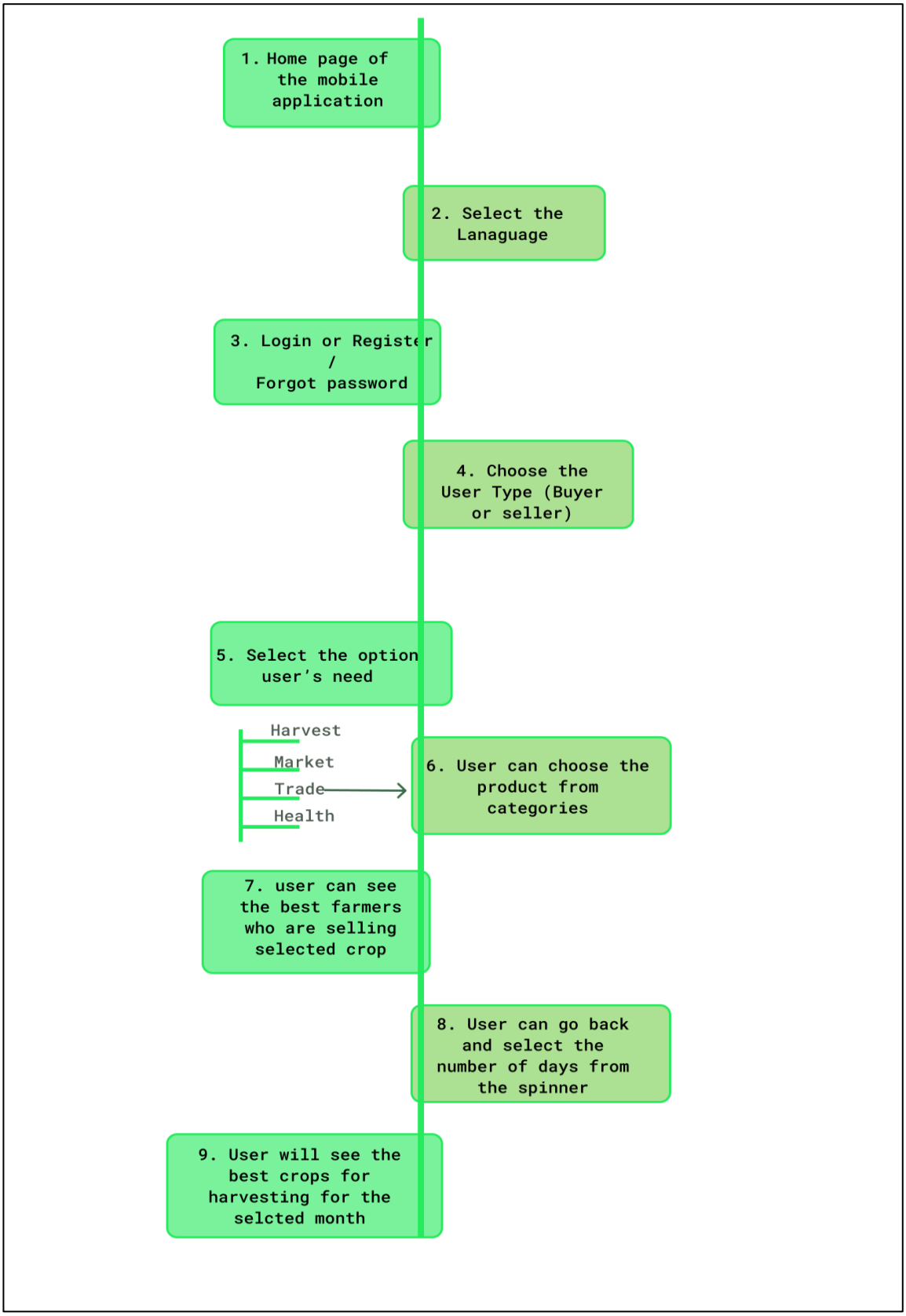


Figure 35 – Trade platform user flow

4.5 Commercialization

4.5.1. Commercializing for Harvest Prediction Module

'Mahaweli market' mobile app basically designed for Mahaweli development project. It's based on the requirements of buyers and sellers of harvest in different seasons. And some other requirements raised from the authorities related to Mahaweli development project. Hence the usage of application is directed towards a community directly dealing with the Mahaweli development project. But the whole solution has its own features to fulfill the basic requirements of sellers and buyers of crops around the country. Basically, models are trained based on 'Yala' and 'Maha' seasons. Hence the solution is applicable for Sri Lanka compared to currently available features. When considering the farmers and other wholesale business dealers we can firmly tell that this kind of software solution will end their plenty of headaches.

4.5.2. Commercializing of Online Trading Platform

Today commercialization of agriculture is an inevitable reality throughout the whole world. There are number of factors affecting the commercialization process in agriculture. This app gives a direct link of farmers to customers, such that a customer is able to talk directly to a farmer and have the bargaining power with the farmer. Customer needs and changes are then shared with farmers to ensure a better service of the product or service.

By using this proposed system farmers no need to go the physical meeting with buyers, it will reduce the transportation cost.In this system we have a recommendation engine for farmers to find a perfect buyer, this will cause to quick selling the products and save the cost for hiring middlemen.

Buyers can also search the nearby farmers who are selling products, so buyers can quickly buy the product freshly. Because of the quickness and freshly products buyers will satisfy with products and gain the trustworthy with farmers.This system will help to gain good communication between buyers and seller.Since this system will developing considering mobile responsiveness, it will much easier for the buyers to

make deals. This helps farmers to have more confidence in their sales and will make farming in Sri Lanka have more value than as it is at the moment where middle men kill the farming value chain.

4.5.3. Commercializing of addressing healthiness of crops

In this component, I was able to develop and implement a successful mobile application feature to measure a level of healthiness of specific crops just by holding the mobile camera towards the desired crop.

As I mentioned in the introduction, in agriculture sector measuring the correct level of healthiness is very much important as the prices for those crops is solely decided according to the healthiness of that crop. Currently the job is done by just measuring the healthiness of that crop using human eye level in most of the developing countries. So, this research component feature is a very useful feature that will help to entire agriculture sector, farmers and sellers to measure the healthiness of their crops and categorize them into different criterias that later when selling they can assign prices according to those healthiness criteria.

After successful development of this whole app, the Mahaweli farmers and sellers will use this application as an intelligent application towards their farming and marketing strategy. So, they can get use of this component's feature and improve their quality of selling by measuring correct healthiness of their crops to assign prices to them.

For a specific time period, we can let the Mahaweli farmers and sellers to use the intelligent application for their day to day basis and we can get a clear idea that whether the app is doing its job properly as a prior to commercial it to the outer farmers and sellers as well. So, if this application with especially the healthiness feature is successful among the Mahaweli farmers and sellers, we can step up to next step to commercialize to other Sri Lankan farmers and sellers as well.

After the deployment of the application towards the farmers and seller of the entire country, we can get a clear knowledge about how they use the application and how the application is needed to be improved. And later, we can improve the things that needs

to be improved and use this agriculture related application especially measuring the healthiness of the crops. Then we hopefully can successfully commercialize the intelligent application to the globe as well.

5. IMPLEMENTATION AND TESTING

5.1. Implementation

Implementation of this application was done through stories. Each story delivered a value instead of a single task. I have used the pystan==2.19.1.1, Prophet, Flask, Flask-Cors for the backend implementation and for the algorithm I have used the prophet algorithm. I have developed my backend implementation separately and it hosted in remote server.

```
if __name__ == '__main__':
    app.run(host="0.0.0.0", port=5000, debug=True)
```

Figure 36 – Server host

File structure wise my backed application has 3 main packages.

- Dataset
- FBP_Data
    - In here I have divided my dataset into categories;
        - Allium.csv
        - Citrus.csv
        - Cruciferous.csv
        - Leafy green.csv
        - Marrow.csv
        - Root.csv
- Predicted_Data

And in the root directory I have created the 4 main files;

- API.py
- DataAnalyze.py
- FBP.py
- Seller.py

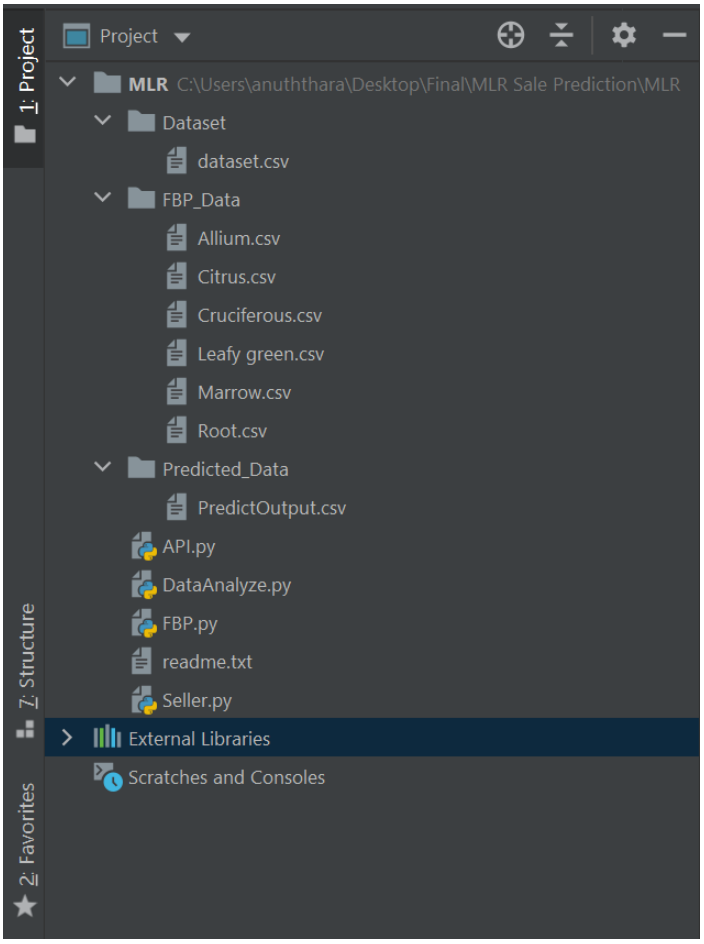


Figure 37 – Backend Application Directory Structure

Backend of the system:

Used technologies:

- Python 3.8
- Rest API
- Flask
- Prophet
- Java (In android)

There are set of dependencies which we have used in order to achieve specific features when using android. Some of the main dependencies are androidx, google with android material, google firebase, Tensorflow lite etc.

```
dependencies {

    implementation 'androidx.appcompat:appcompat:1.2.0'
    implementation 'com.google.android.material:material:1.2.1'
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
    implementation 'com.google.firebase:firebase-auth:20.0.1'
    implementation 'com.google.firebase:firebase-firestore:22.0.0'
    implementation 'com.google.firebase:firebase-storage:19.2.0'

    // Import the BoM for the Firebase platform
    implementation platform('com.google.firebase:firebase-bom:26.8.0')

    // Declare the dependency for the Realtime Database library
    implementation 'com.google.firebase:firebase-database'

    //------- Added by Thamal ----------
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    //------- Tflite
    implementation 'org.tensorflow:tensorflow-lite:0.0.0-nightly'
    implementation 'org.tensorflow:tensorflow-lite-gpu:0.0.0-nightly'
    implementation 'org.tensorflow:tensorflow-lite-support:0.0.0-nightly'

    implementation 'com.squareup.picasso:picasso:2.71828'
    implementation "androidx.cardview:cardview:1.0.0"
    implementation 'androidx.recyclerview:recyclerview:1.1.0'
    implementation 'androidx.legacy:legacy-support-v4:1.0.0'
    implementation 'com.android.volley:volley:1.1.1'
    implementation 'com.google.android.material:material:1.2.1'
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
    implementation 'de.hdodenhof:circleimageview:3.1.0'
    implementation 'org.jetbrains:annotations:15.0'
    implementation 'com.google.firebase:firebase-database:19.7.0'
    implementation 'org.tensorflow:tensorflow-lite-support:0.1.0-rc1'
    implementation 'org.tensorflow:tensorflow-lite-metadata:0.1.0-rc1'
```

Figure 38 - Gradle dependencies

5.2.Testing

Testing part of this project has done through the four modules separately.

5.2.1 Testing of Harvest Prediction Module

Unit testing

The unit testing is carried out by dividing the harvest prediction module and weather forecasting module into smaller parts known as units. Each and every individual unit should be tested under unit testing. When the system design documents are received, the work is divided into modules/ units, and actual coding is started. The system is first developed in small programs called units, which are integrated into the later phase. Each unit is designed and tested for its functionality. Unit testing mainly verifies if the modules/ units meet their specifications.

Test cases for the next season harvest prediction

Next season harvest is expected to have in a range of data plotted in past 18 years.

Test case ID	002
Description	Next season harvest requirement prediction.
Steps	Register/Log in to the application. Select seller. Select Harvest component in home page. Click on harvesting division in Harvest Dashboard. Select a Crop, Season, Area from drop down list. Click on Plot.
Input Data	Crop, Season, Area
Expected Output	Line chart plotting last 18 year's harvest data in relevant season with next coming season forecast.
Actual Output	Line chart plotting last 18 year's harvest data in relevant season with next coming season forecast.

Table 1 - Generating next season forecast test case

5.2.2. Testing of addressing healthiness of crops

In the research component of measuring healthiness levels, the main testing that needed to be done is the deep learning model testing. The deep learning model testing is first done from the python coding itself in the jupyter notebook. As I mentioned in the implementation methodology, first thing I had done was, get a batch of test data from the test batches.

By calling model.predict we can obtain the prediction of the trained deep learning model. To the x parameter I had specified the test batches of all the test dataset. And the verbose is chosen to be 0 to get no output whenever the predictions are run.

```
In [48]: predictions = model.predict(x=test_batches, verbose=0)

In [49]: np.round(predictions)

Out[49]: array([[1., 0., 0., ..., 0., 0., 0.],
                [1., 0., 0., ..., 0., 0., 0.],
                [1., 0., 0., ..., 0., 0., 0.],
                ...,
                [0., 0., 0., ..., 0., 0., 1.],
                [0., 0., 0., ..., 0., 0., 1.],
                [0., 0., 0., ..., 0., 0., 1.]], dtype=float32)
```

Figure 39 - Model prediction testing

The predictions can be printed as the rounded predictions of the model. The way that we can read the preview of the prediction is, first one of the arrays is a prediction for a single sample. If we look at the first sample of the test set as above figure displays, wherever there is a one with each prediction is the index for the output class that had the highest probability from the model. In this case, the 0^{th} index has the highest probability. So, we can get finalized that the label that the model predicted for this first sample was a zero, because there is a 1 in the 0^{th} index. And also the predictions can be passed to the plot confusion matrix and view the test results. This part will cover in the results section of the document.

5.2.3. Testing of trade platform

Since this application is a lot more focusing on real time data, most of the testing we conducted was manual testing. We used automated tests as well.

Our application uses testing libraries like, jest, mocha and chai.

1. Unit tests - Done using “jest” ,For assertions we have used “mocha” also.
2. Integration Tests - All the Rest Endpoints were tested using unit tests and for the assertions we have used “assert” and “chai”.
3. UI / End to end testing - These tests are conducted manually. We have done the user acceptance testing as well.

6. RESULTS AND DISCUSSION

6.1. Results and Research Findings

6.1.1. Results and Research findings for Harvest Analysis

The initial steps for the research were taken from visiting the Mahaweli authority of Sri Lanka. According to the requirements the authority provided us most important data and information necessary for the research via the official statistical book '2018 Socio-Economic Statistics' of Mahaweli Authority of Sri Lanka.

The results obtained from the report had to reorganize which were gathered referring important tables and which contained official data of harvest in 'Yala' and 'Maha' seasons in different crops. So, the average seasonal harvest is given in Mt/Ha and water supply in MCM.

Figure 40 - Model comparison graph

According to above graph most accurate model for long run is linear regression model which is represented in orange bars. Even though most of the occurrences of ridge

regression tally with linear regression it's easy to use linear regression due to lesser complexity.

6.1.2. Results and Research findings of Addressing Healthiness

Mainly results can be divided into two sections. Those are the results comes out from the measuring level of healthiness feature of the intelligent application, and the test results came out from the trained deep learning model. The third objective of this research component is to complete the application by creating deep learning model to predict healthiness using the camera component which was integrated into this feature component. As it is developed towards the farmers and sellers of Mahaweli market and according to the requirement of Mahaweli authority, the system can predict four levels of healthiness for potatoes and onions. Below 2 figures will show the results for onion and for potatoes.

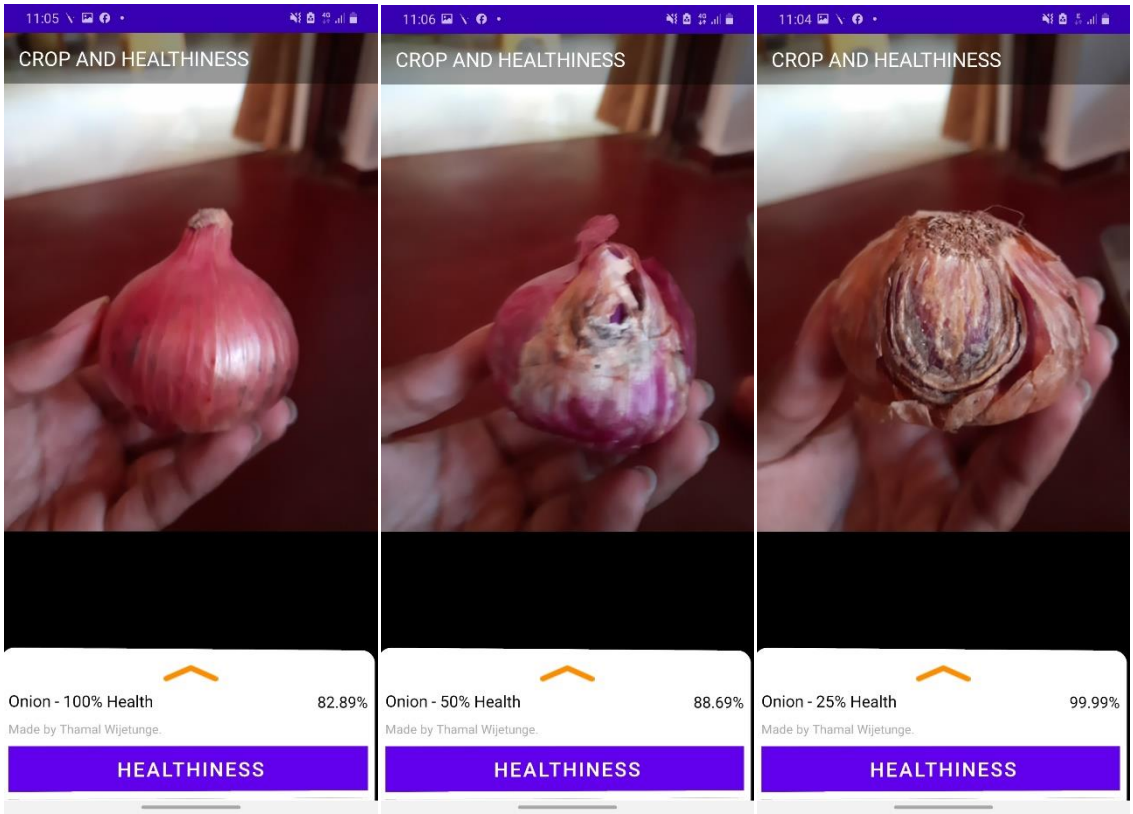


Figure 41 - Onion healthiness predict interface

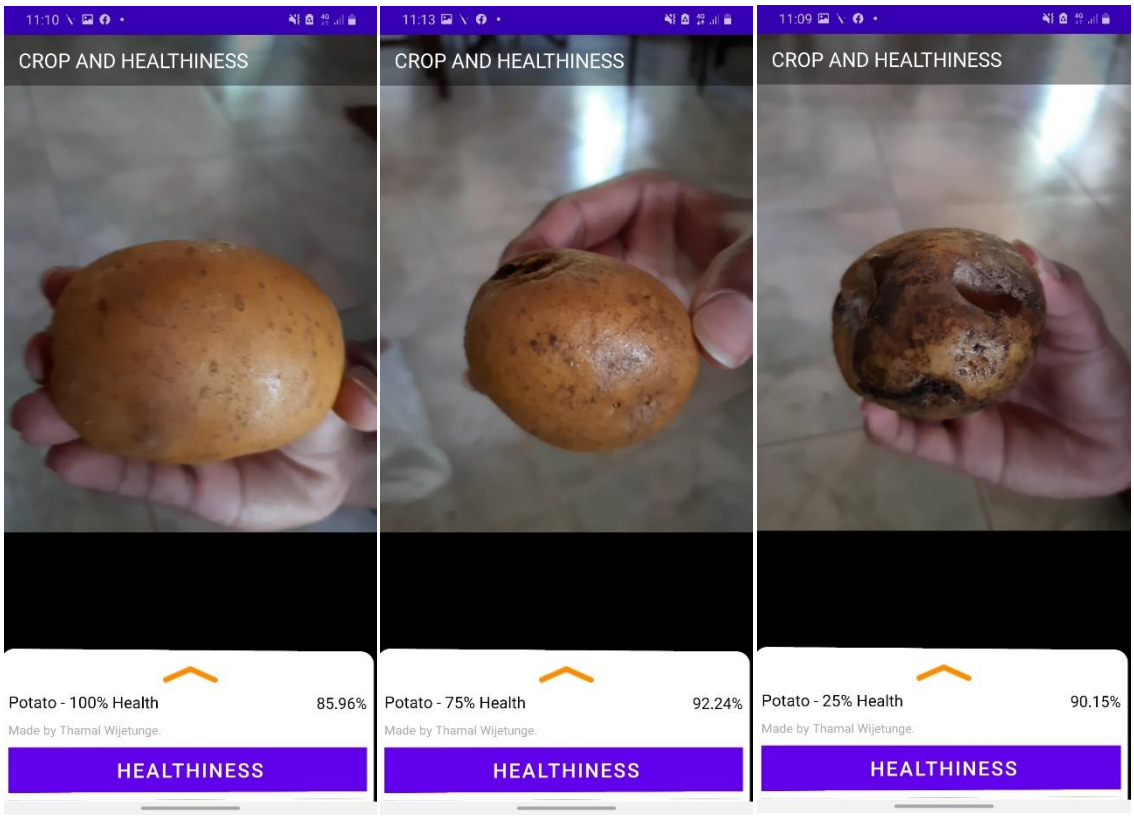


Figure 42 - Potato healthiness predict interface

According to above figures, the results are shown to the users in the bottom part of the interface. Users can see whether the crop is potato or onion and the level of healthiness for that crop out of four levels as 25%, 50%, 75%, 100%. The novelty feature of this component is that the user don't need to wait after clicking a button or do something to get the result as the healthiness. User can simply hold the camera above the desired crop and it will automatically predict the crop with the level of healthiness real-time. This resulting feature will help a lot for Mahaweli farmers and sellers by saving their time. Also the UX part that is user friendliness and easy to use is clearly depicted through this feature.

Also, in any case that farmer or seller need to verify the healthiness of that specific crop, he/she can simply click the 'HEALTHINESS' button which positioned under the predicting label.

When user clicks the ‘HEALTHINESS’ button, it pops up a simple modal that shows the information about what the crop is and the level of healthiness the crop has. Further, there are two buttons in the modal. By clicking the save button user can save that crop with the level of healthiness including the date and the time as well. For that, firebase database system is used as mentioned in the methodology. The pop up modal is shown like the below figure.

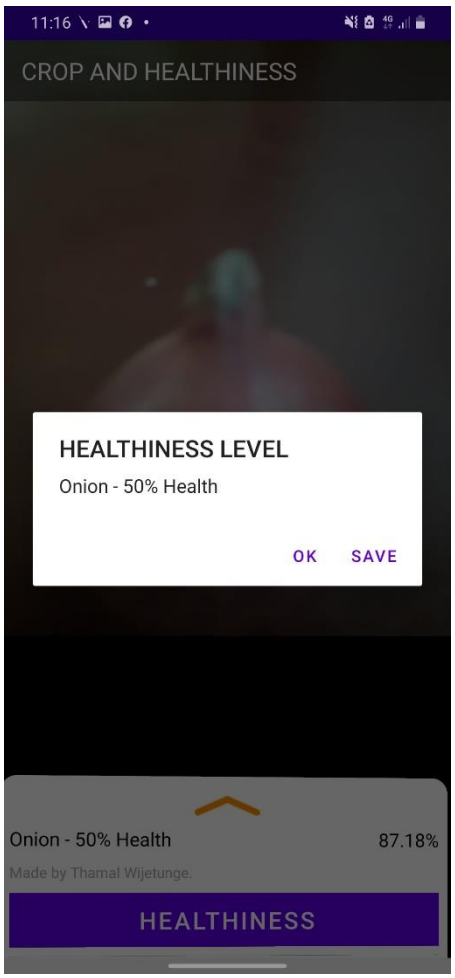


Figure 43 - Modal pop up interface

If user doesn’t need to save the crop with healthiness for future needs, he/she can just click on ‘OK’ button to close the modal.

The extra feature I have provided in this research component is to save the crop and the level of healthiness of that crop to database for future purposes. As I mentioned above, farmer or seller can save the information by clicking the button underneath the camera fragment and then clicking on the save button which is positioned on the pop up modal.

The farmer or seller also can view the saved data from a different interface. It can be navigated from the two button interface by clicking on the 'VIEW HEALTH' button. Click that button will redirects user to the view interface of the crop and healthiness information. The list of data are looped into a list view and it is shown as the figure below.

Figure 44 - List view interface

Not only the results that mentioned above, but there are test results that given from the deep learning model predictions as well.

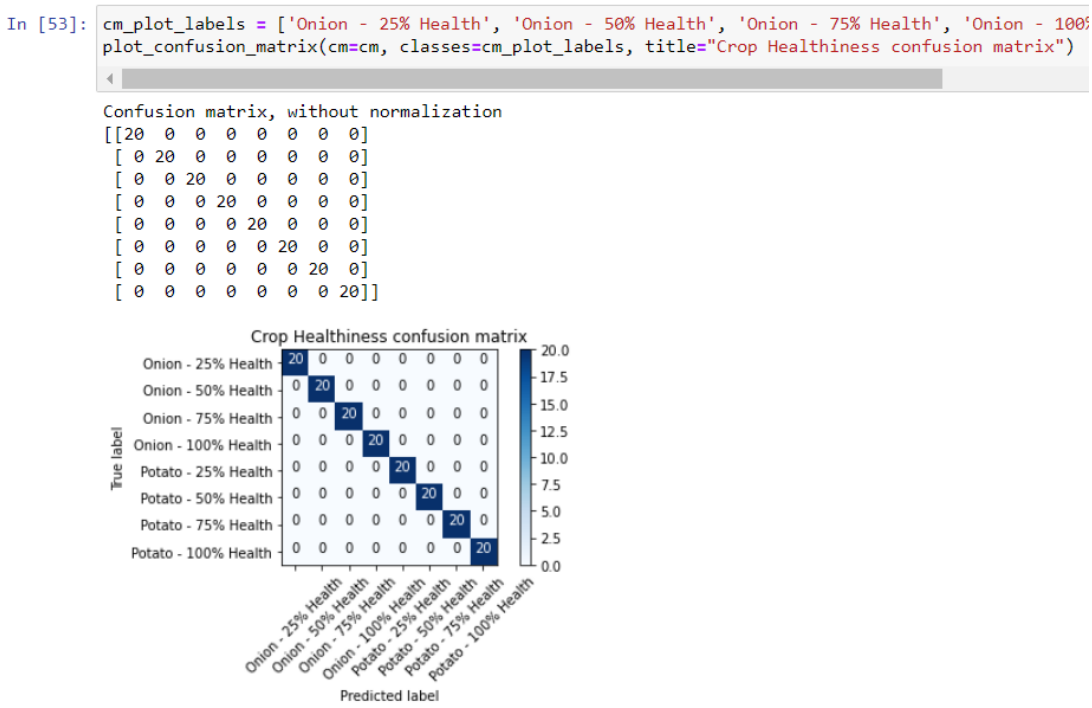


Figure 45 - Confusion matrix results

As the above figure, in order to plot confusion matrix the plot labels are defined in the correct order. Next 'plot_confusion_matrix' function is called with some parameters. First parameter is the confusion matrix itself. Then the labels for the confusion matrix is passed along side with a title for the entire matrix.

In the left side of the matrix, the true labels are positioned from top to bottom as the y axis. In the bottom the predicted labels are positioned from left to right. The diagonal which started from top left corner to bottom right corner indicates the correct probability that the model predicts. If any number is positioned outside the diagonal it means that the label is predicted incorrectly.

6.1.3. Results and Research findings of Marketing Analysis

The initial step for the research was taken by finding the data related to the stock market prices of certain crops. It was quite challenging due the present situation of the country. We could find the data from our main resource person “Mahaweli Authority of Sri Lanka”.[4]. The data found dived into two parts training and testing.

The entire research component was based on that data. Model generation, Price prediction all the output generation process conduct by using that data. Using 5years (2013 - 2018) crop price generating the model. The accuracy can be improved the model by using latest data and considering more years rather than 5years.

Linear Regression algorithm was the suitable algorithm when generating the numeric values, but considering the stock prices analyzing most of the researchers were used Support Vector Machine (SVM) and Practical Swarm Optimization (PSO). But in in this research I showed Linear Regression algorithm also can be used for the stock market prices predictions. The accuracy of the prediction approximately 88.9%.

Later if we can integrate this linear model with the SVM or PSO models to predict the results accuracy will be improved.

6.1.4. Results and Research findings for Online Trading Platform

According to the first phase of the implementation I have been able to achieve successful results.

In the component of the online trade platform I have achieved to complete the android application with the main function to gaining the farmer and buyer relationship.

My goal was here is to create a beneficial environment for farmers and buyers, allowing them to maximize on their profits, without compromising on quality using the best, efficient and easiest way using a machine learning models that is able to predict the changing patterns of farmers. Being able to notify customers whenever the

harvest of a particular crop reduces and being able to alert farmers of the current needs of customers in terms of variety and preferences.

He/she then can decide to order whatever they prefer, the app studies this specific users action and saves on a machine model which then comes us with a predictive analysis of what specific customers and genders like and at what period and under what conditions.

This gives more insights on the farmer's aspect, they are able to do farming geared towards users predilections based on rates. This gives a further analysis of what the farmer should essence on more in order to maximize the production value chain. Profits are maximized as customers cooperate directly with farmers and they can negotiate basing on users favorites. This helps farmers to have more self-confidence in their sales and will make farming in Sri Lanka have more value than as it is at the moment where middle men destroy the farming value chain.

6.2. Discussion

6.2.1 Discussion for Harvest Prediction Module

This requirement is fulfilled using linear regression model as the seasonal average yield is linearly dependent upon water supply from reservoirs in different areas. This approach was not straight forward, even though the plotted graph of yield against water supply had a linear distribution linear regression model Ridge regression model and gradient tree model also compared to obtain most suitable model with minimum standard error. One key factor related to modeling was when the yield is plotted against water supply it was clear that lower values and upper values of water supply had two separate linear relationships. With this it was quite clear that significant difference is due to seasonal change. When the dataset is separated as Yala and Maha two different models are introduced to each crop. Further three main crops namely Paddy, Potato and Onion named for the analysis hence six different models are trained to increase the accuracy of each forecast.

6.2.2 Discussion for addressing crop healthiness

After implementing all these mentioned methods, finally I was able to complete the research component successfully. The reason I used image classification instead of image processing is the wide range of this component. If using image processing, the single component can be done as a full research project according to the literature surveys I have done. Our sole intention at the beginning was to deliver a complete intelligent agricultural app for the Mahaweli farmers and sellers.

From this research component measuring the levels of healthiness for potatoes and onions Mahaweli farmers and seller now won't have to just measure their crops healthiness by eye level. They just need to log into the system, continue as a seller, redirect to the healthiness checking camera fragment and hold the camera to the

desired crop. This helps the farmers and sellers to categorize the crops according to level of healthiness that the application predicts.

We hope that the Mahaweli farmers and sellers get the most of this agricultural intelligent application, especially the healthiness measuring component as it clearly help them to maintain the pricings and selling in a correct manner. We can let Mahaweli farmers and sellers use it for a specific time period and get an idea about allowing the app to use by the farmers and sellers all around Sri Lanka as well

6.2.3 Discussion for Online Trading Platform

Most of the research papers I have reviewed were based on simple scenario such as only buying goods from farmers and selling goods for buyers. But the research finding in my component is I have implement the recommendation system for buyers and farmers according to their preferences based on multiple factors such as ratings, tracking locations and product types. And also other than the online trading platform our system has healthy prediction, harvest and weather prediction and also sales predictions. So that users can get maximum outcomes from one place.

Based on the research I have one through all over the internet I have found that there have not many systems for online trading for farmer and buyers. Currently in Sri Lanka there have only 2 main mobile application for such scenario which are called Govipala, Colombo manning market.

But in both applications I have found that there were no system for recommending farmers or products to the buyers and not having the language preferences also. But in our application we have integrated that our own recommendation engine which is recommend the best farmers based on the ratings and locations and also by using our system farmers can get high prediction about the next harvesting crop.

6.2.4. Discussion for Market Analysis

This developed app mainly focus on the Mahaweli farmer's to reduce their effort and wastage. Because cultivators put lot of effort to get a good products; if they unable to sell them it will be huge lost. Sri Lanka is a developing country. Therefore our mainly income was the agriculture. During past few years agriculture field heavily damaged due to COVID 19 pandemic. Therefore we believe our effort towards agriculture will be help to reduce the farmer's effort and improve the productivity.

Background literacy cover the topic of the component of the system. Linear Regression model chose due to its effectiveness to predict the numeric values and novelty of the predicted stock market prices. To understand the model little difficult because of that novelty.

We developed mobile application instead of web application due to user friendliness. Therefore more user base can created for us toward this application. Mobile app developed using Android studio and Machine Learning model was implemented by using Tensorflow lite. Integrate machine learning model with the front end and display the predicted value of the particular crop for particular season and the year was main objective of the component.

Hope to implement the model with SVM and PSO model for further to obtain more accurate results.

7. CONCLUSION

Mahaweli market is next generation mobile application to solve basic but important requirements of Mahaweli farmers in Sri Lanka. The proposed solution as separate harvest predicting module which consists of sub functionalities for climate prediction real time selling price demonstration component and harvest forecasting feel requirements of farmers. Technically the weather forecast is done for next coming seven days, and it is done using open weather API. Real time data collection is done using firebase real-time database. Harvest forecasting for next season is a critical requirement. Future of the application is more about finding out further environmental features to increase the accuracy of models. Currently the harvesting module specifically solves all the requirements of Mahaweli farmers is fulfilled and the objective of creating such module is successfully achieved with expectations of further development. Farmers and seller have measured the healthiness of their cultivation by just human eye level. But that method brings lots of problem towards selling their crops to a correct pricing. They cannot even categorize their cultivated crops according to different healthiness levels. That may become a huge disadvantage towards farmers and sellers when selling they're crops with low prices to customers. That is the problem and reason to develop a platform to address the healthiness levels for specific crops with a very easy manner. Further, this component of the application is developed to use with ease to the users. Farmers or sellers who uses this application simply need to hold their camera phone by their desired crop and the application will generate a prediction of healthiness with the name of the crop in real-time. In this system introduced new algorithm can be used for the stock prediction. The proposed solution provides predicts stock prices thought the use of Linear Regression algorithm. Most of numeric value's prediction done by through the Linear regression. With the prevailing challenges in the country like pandemics, economy, digitalization Mahaweli Market App suggests extraordinary solution to the government. It has become the most prominent platform to connect buyers, sellers and government officials together under one roof. All the most important features integrated to the application which were tabled on feasibly study discussions and meetings. Specifically in harvest prediction

analysis, it shows the relationship between weather and harvest is linear. Also, Market prices increases linearly. These conclusions help the solution to predict values with least error. Also considering the theory of matching applied in trade platform using K-Nearest Neighbors. The crops images are processed to identify the healthiness of the crop. Nature of such dataset reveals the use of Keras Sequential Model for predictions. Basically, the app provides different aspects of cultivation process ease with decision making. Hence larger portion of the future of cultivation in Sri Lanka will be handled by Mahaweli Market App.

8. REFERENCES

[1] Diksha Manaware. 2020. Artificial Intelligence: A New Way to Improve Indian Agriculture. Int.J.Curr.Microbiol.App.Sci. 9(03): 1095-1102.

[2] Baogui Zhoa 2020 J. Phys.: Conf. Ser. 1574 012139 – The Application of Artificial Intelligence in Agriculture.

[3] .Osman Hegazy, Omar S. Soilman, Mustafa Abdul Salam, 'A

[4] A Review of the Usage of Machine Learning in Real-time

Systems https://www.researchgate.net/publication/311546058_A_Review_of_the_Usage_of_Machine_Learning_in_Real-time_Systems

[5] Machine Learning Model for Stock Market prediction' Faculty of computers and informatics Cario University.

[6] Duan Yan-e, Design of Intelligent Agricultural Management Information System Based on IoT. Beijing University of Agriculture, Beijing, China, 2011.

[7] K. Koszela, M. Lukomski, W. Mueller, K, Gorna, O, Okon, P. Boniecki, M. Zaborowicz, D. Wojicieszak, Classification of Dried Vegetables using Computer image analysis and artificial neural networks. Poznan Univ. of Life Science, Poland.

[8] Arjuna Gunawardhana 'Agriculture sector Performance in Sri Lanka economy'

[9] Osman Hegazy, Omar S. Soilman, Mustafa Abdul Salam, 'A Machine Learning Model for Stock Market prediction' Faculty of computers and informatics Cario university

[10] W.G. Somarathne 'Sri Lanka agriculture for next decade' university of Peradeniya

[11] Menale Kassie, Bekele Shiferaw and Geoffrey Muricho, Agricultural Technology, Crop income, and Poverty Alleviation in Uganda. International Maize and Wheat Improvement Center, Nairobi, Kenya. Vol 39, 2011.

[12] Duan Yan-e, Design of Intelligent Agricultural Management Information System Based on IoT. Beijing University of Agriculture, Beijing, China, 2011.

[13] K. Koszela, M. Lukomski, W. Mueller, K, Gorna, O, Okon, P. Boniecki, M. Zaborowicz, D. Wojicieszak, Classification of Dried Vegetables using Computer image analysis and artificial neural networks. Poznan Univ. of Life Science, Poland.

[14] K. Przybyl, K. Gorna, D. Wojcieszak, The recognition of potato varieties using of neural image analysis method. Poznan Univ. of Life Science, Poland.

[15] Priyanka S. Mankoji, Uttam Patil, A Versatile Mobile Application for Quality Prediction of Wheat Grain using Machine Learning Algorithms. Belagavi Karnataka, India. Vol. 6, Issue 7, July 2007.

[16] Kavindra Paranage, The Mahaweli Development Project and the 'rendering technical' of agrarian development in Sri Lanka. School of Social Sciences, Monash University, Wellington Road, Clayton, VIC, 3168, Australia.

[17] T Gunda, J T Bazuin, J Nay, K L Yeung, Impact of seasonal forecast use on agricultural income in a system with varying crop costs and returns: an empirically-grounded simulation. Vanderbilt Institute, United States of America.

9. APPENDICES

Document Viewer

Turnitin Originality Report

Processed on: 16-May-2021 20:09 +0530
ID: 1586013687
Word Count: 12396
Submitted: 2

20_21-J11_Group report By IT17172806 De Silva D K G

Similarity Index
9%

Similarity by Source

Internet Sources:	7%
Publications:	4%
Student Papers:	6%

include quoted | include bibliography | exclude small matches | mode: quickview (classic) report | Change mode | print | download

1% match (student papers from 02-Dec-2017)
Submitted to Sri Lanka Institute of Information Technology on 2017-12-02

1% match (Internet from 08-Dec-2020)
https://www.mdpi.com/2076-3417/10/21/7748/html

1% match (student papers from 04-Jun-2020)
Submitted to Amity University on 2020-06-04

1% match (student papers from 04-Apr-2021)
Submitted to National School of Business Management NSBM, Sri Lanka on 2021-04-04

1% match (Internet from 12-Dec-2020)
https://medium.com/analytics-vidhya/time-series-forecasting-arima-vs-prophet-5015928e402a

1% match (Internet from 14-May-2019)
http://ijarcsms.com

1% match (Internet from 05-Jun-2017)
http://lirneasia.net

Figure 46 – Turnitin similarity report