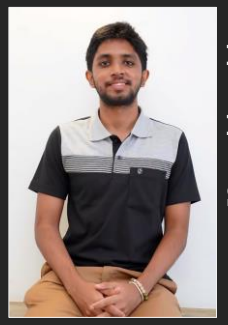
Agriprenuers

2021-090

Team

Mr.Dhammika De Silva

Ms.Janani Thamalaseelan



IT18110562

R.S.W.M.R.L Rangalla

SE

IT17023610

A.G.J.L.P

Rajapakse

SE

IT18111170

Silva S.H.I

SE

IT18027334

P.D.M Thilekarathna

CSN

Figure1: https://www.pewresearch.org/social-trends/2020/09/24/economic-fallout-from-covid-19-continues-to-hit-lower-income-families-the-hardest/

Background

- High impact on families with middle to low income.
- Extra source income is always welcomed.
- Entrepreneurship is a high-risk domain.

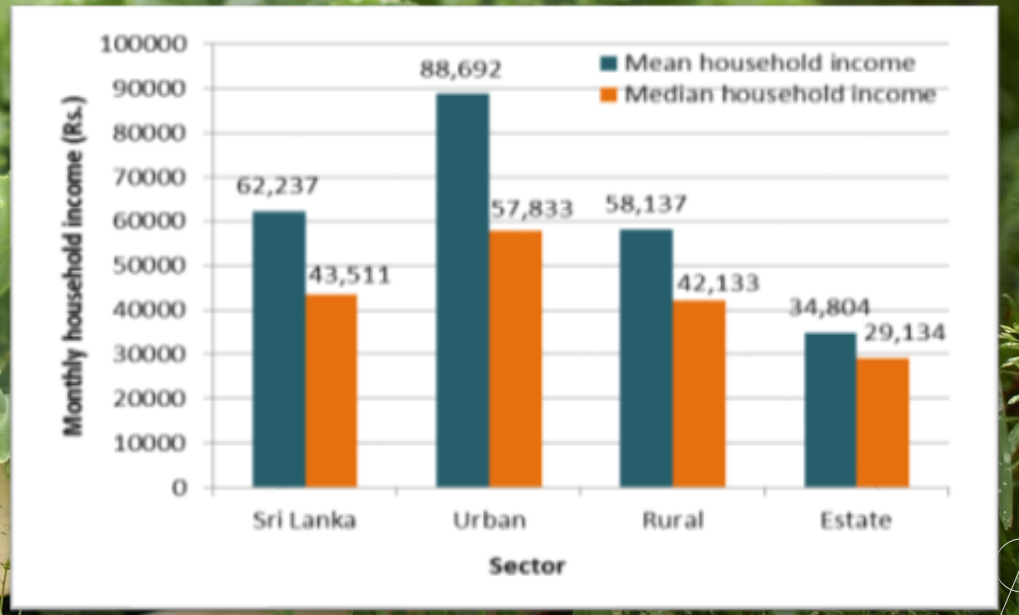


Figure2: Department of Census and Statistics Srilanka, 2016

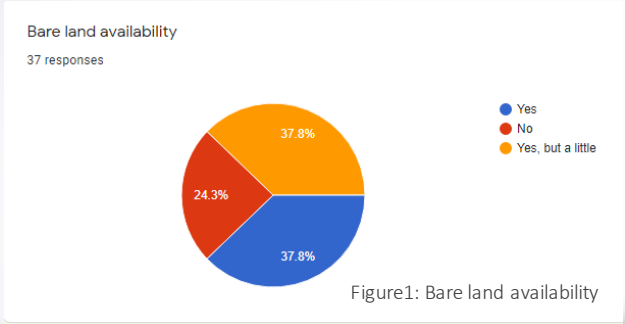


Figure1: Bare land availability

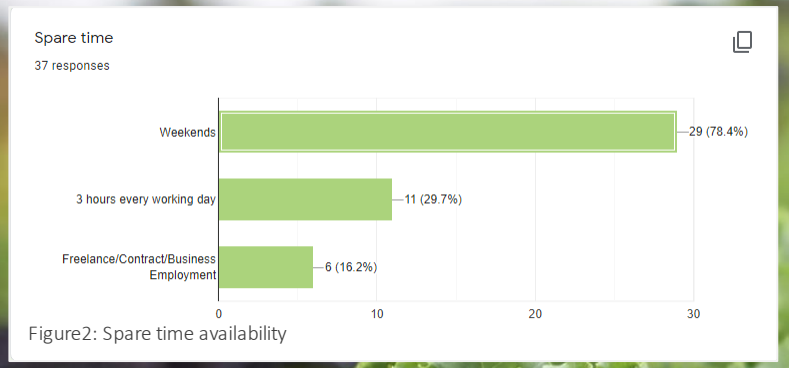


Figure2: Spare time availability

- High risk in Entrepreneurship domain.
- Less motivational strategies to promote entrepreneurship.
- No proper tool to promote entrepreneurship.
- Unavailability of stable values of a success rate figure to motivate a person to be involved in a business.
- Not utilizing bare land in many households, due to lack of time.

Research Problem

- Under Utilizing the scope of the Agriculture Domain in Urban areas
- Less motivation and promotion for entrepreneurship.

Figure3: Monthly expenditure

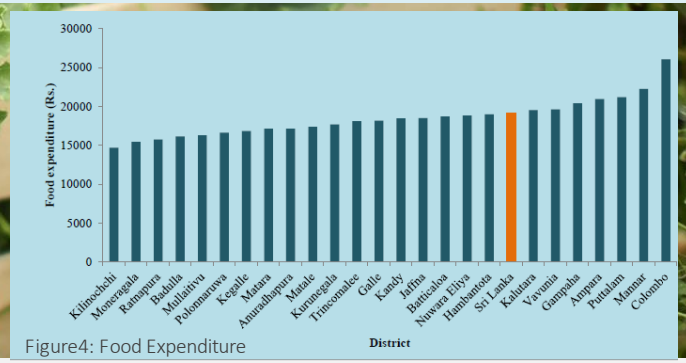


Figure4: Food Expenditure

Objectives

Main

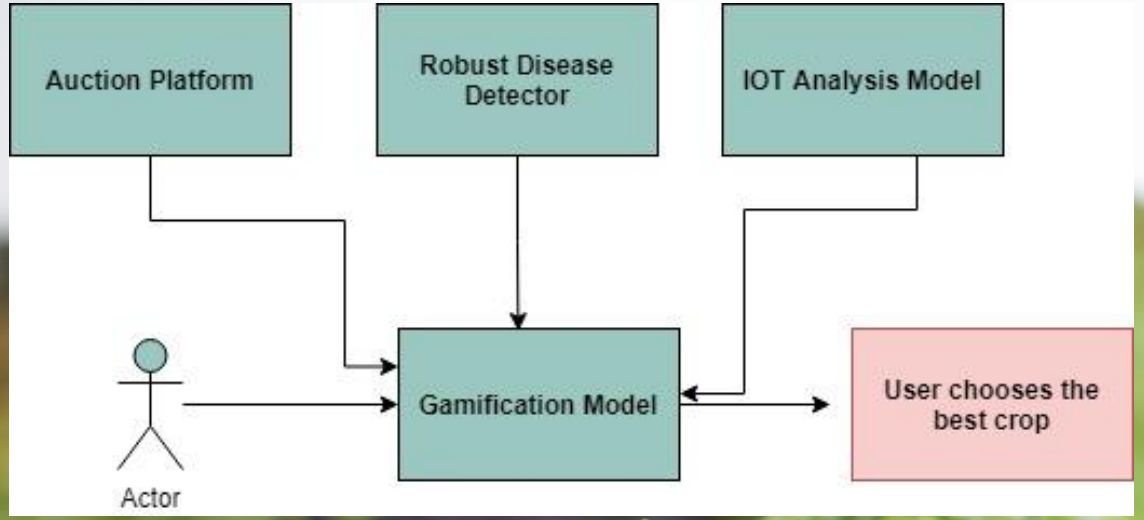
- Build a motivational platform to generate entrepreneurs.

Sub-objectives

- Gamification model.
- Realtime data analysis IoT model.
- Robust Image processing disease detection model and recommendation prediction.
- Auction Platform, Market Rate prediction model with time.

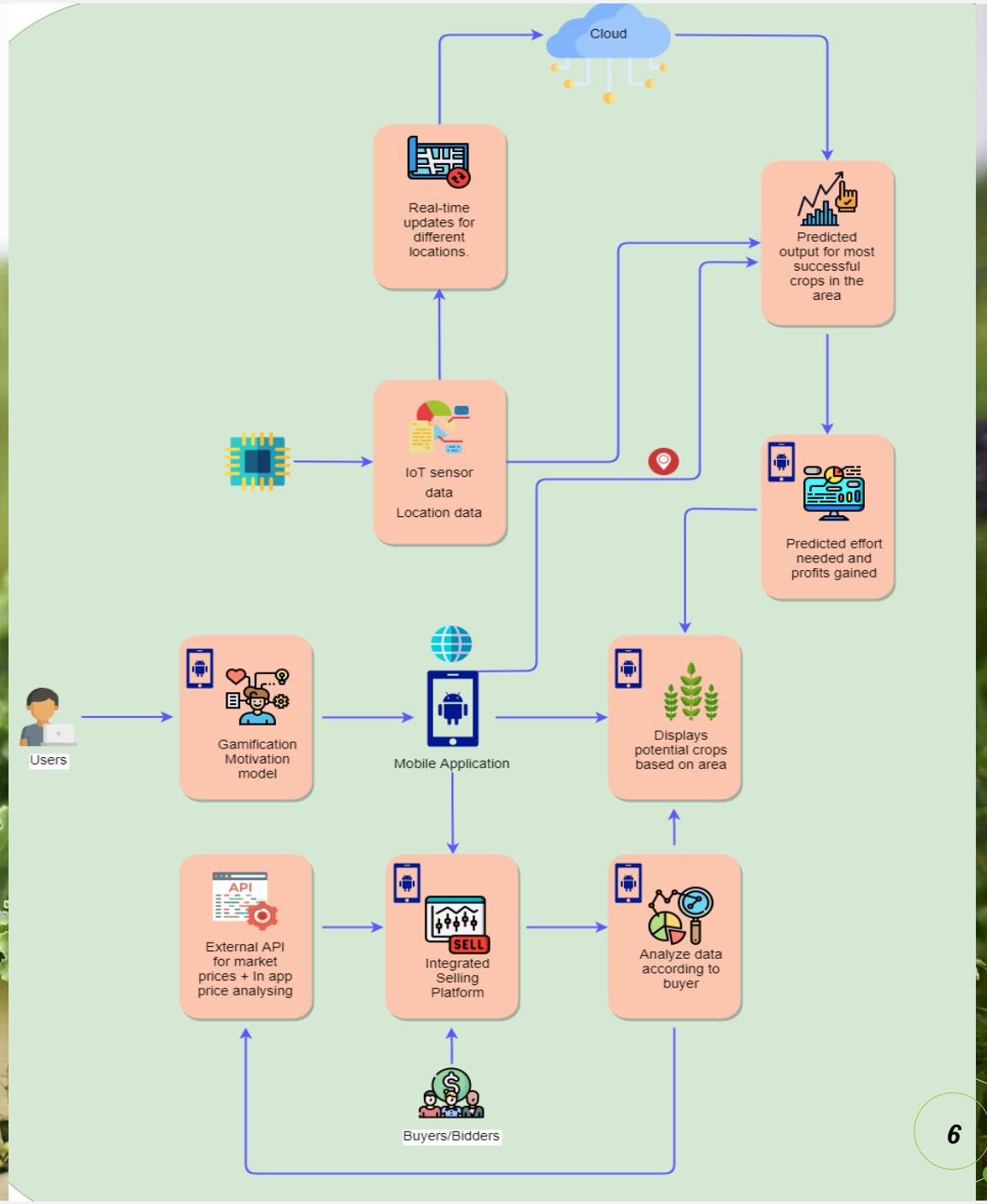
Other Objectives

- Social acceptance.
- Regional leaderboard implementation.
- Reward Scheme.
- Auction Platform.
- Deployment.



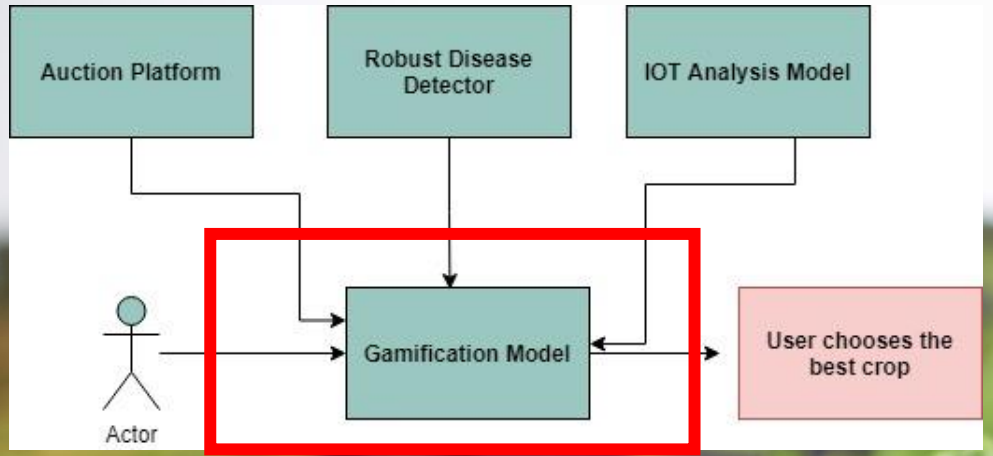
System Overview Diagram

- 2 Prediction Models with the help of Machine Learning.
- 1 IoT Analysis Model
- Gamification Model



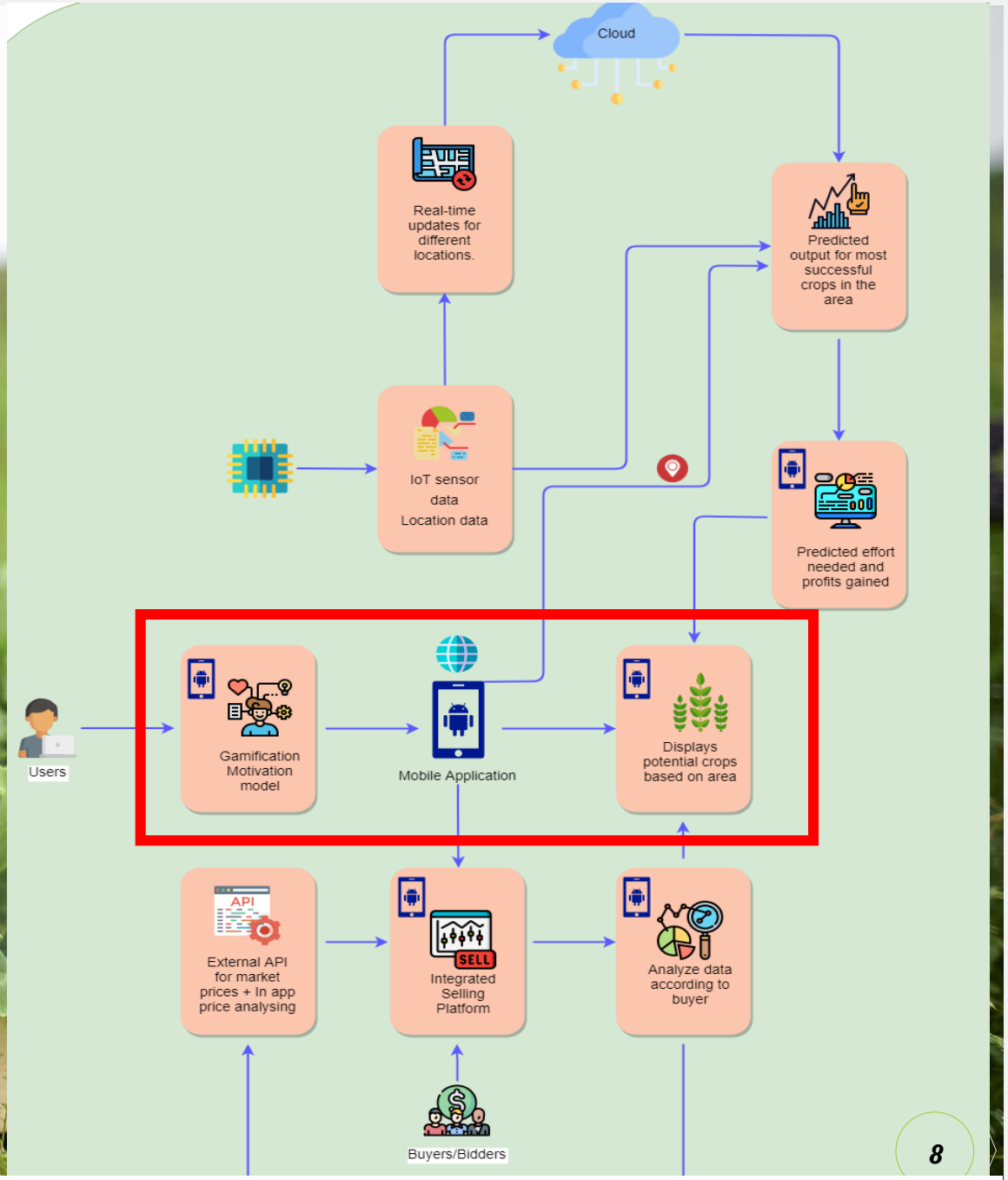
IT18111170 | Silva S.H.I

Specialization: Software Engineering



Gamification Motivation Model

- Leaderboard & Reward Schema
- Motivational Psychological model
- Gamification model



Introduction

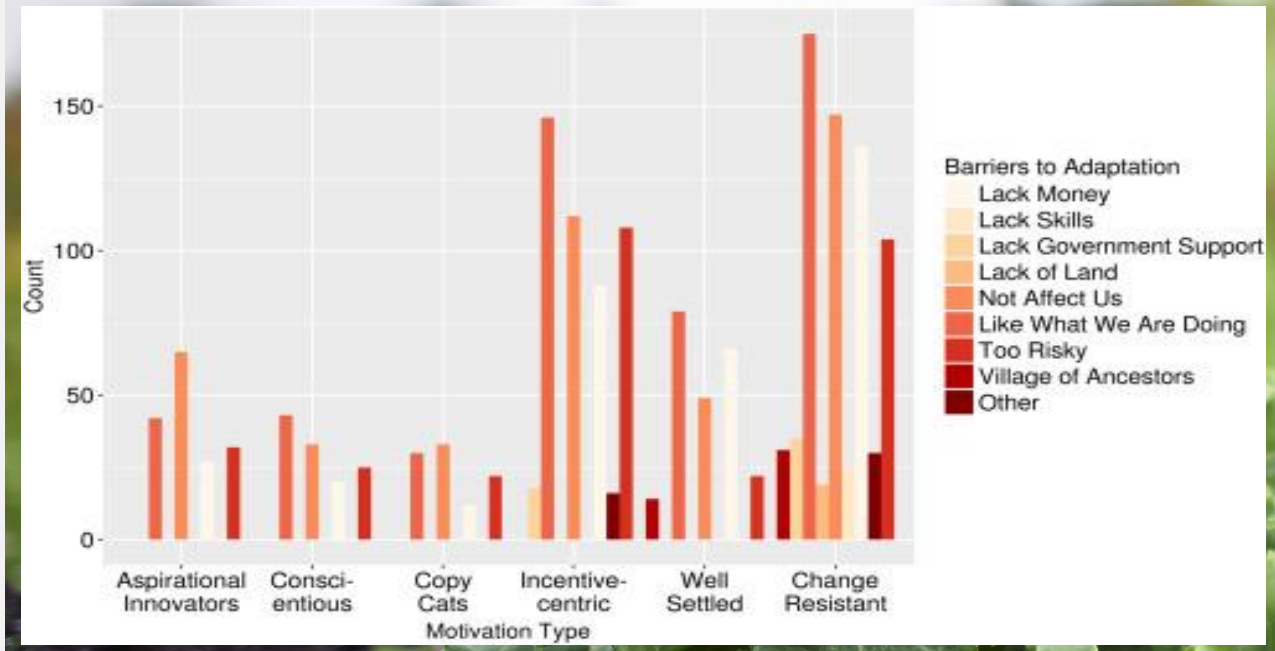


Figure: Motivation type against Barriers to adaption

Background and Research gap

- People are lazy to motivate and become entrepreneurs in agriculture even they are eligible
- Some motivational model in agriculture can be found but Mobile App using with psychology aspect are very less.
- Implement Mobile Application with considering the psychology motivational aspects including Gamification reward model.

Main Objective

- Motivate users to become self-made entrepreneurs and sustainable in economy.
- Basically creating the Mobile Application

Sub Objective

- Create Leaderboard which shows ranks of users based on money earns
- Create reward model proving discounts on taxes using points for user to addict and motivate.
- Gamification model that present the earned prices for crops

Research Problem

- There is lack of motivation on Agriculture for individuals to become entrepreneurs. Even if they are eligible

Research Methodology

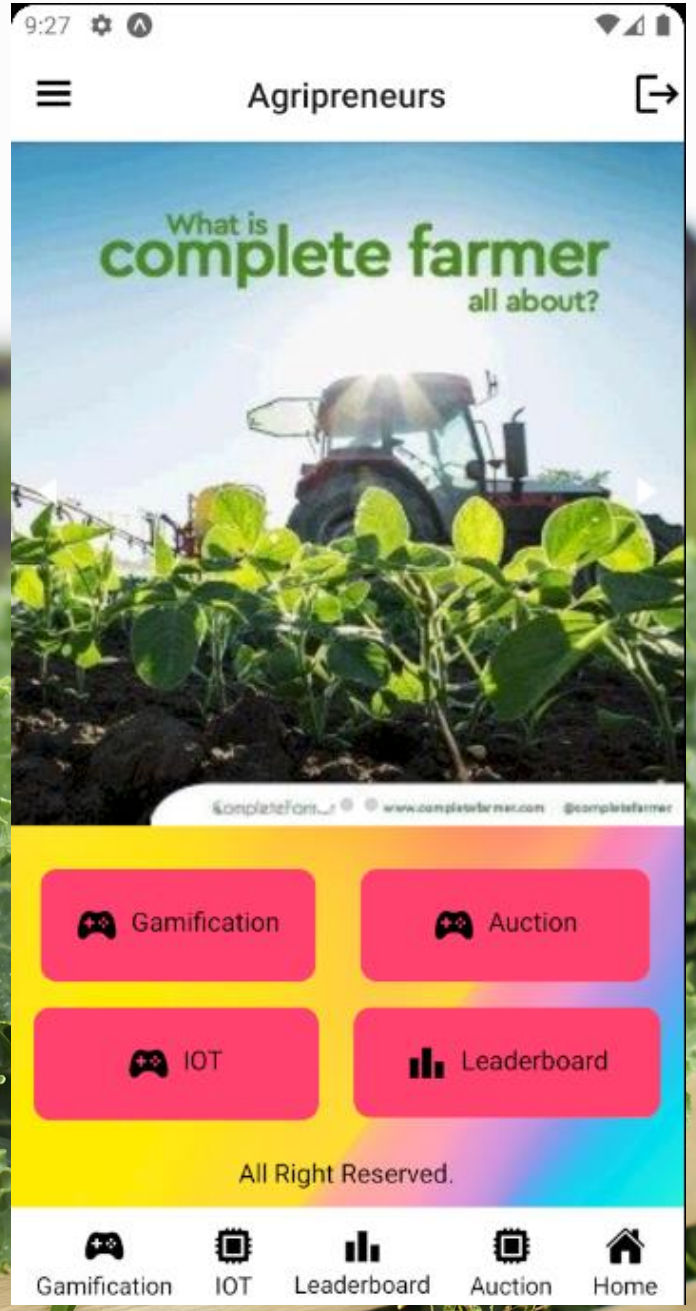
Technologies, Tools

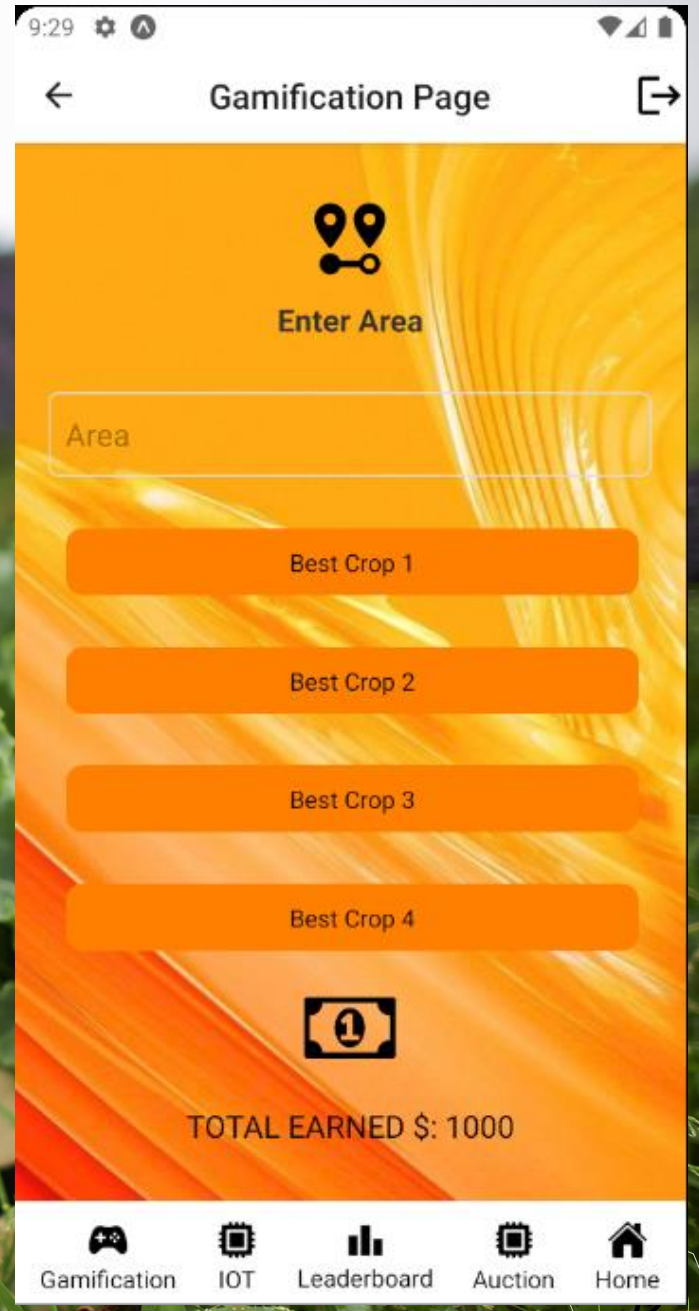
- Mobile App which has React native frontend with Python Flask framework for Backend
- MongoDB-Database
- Tools- VS code, Emulators, Expo, Postman

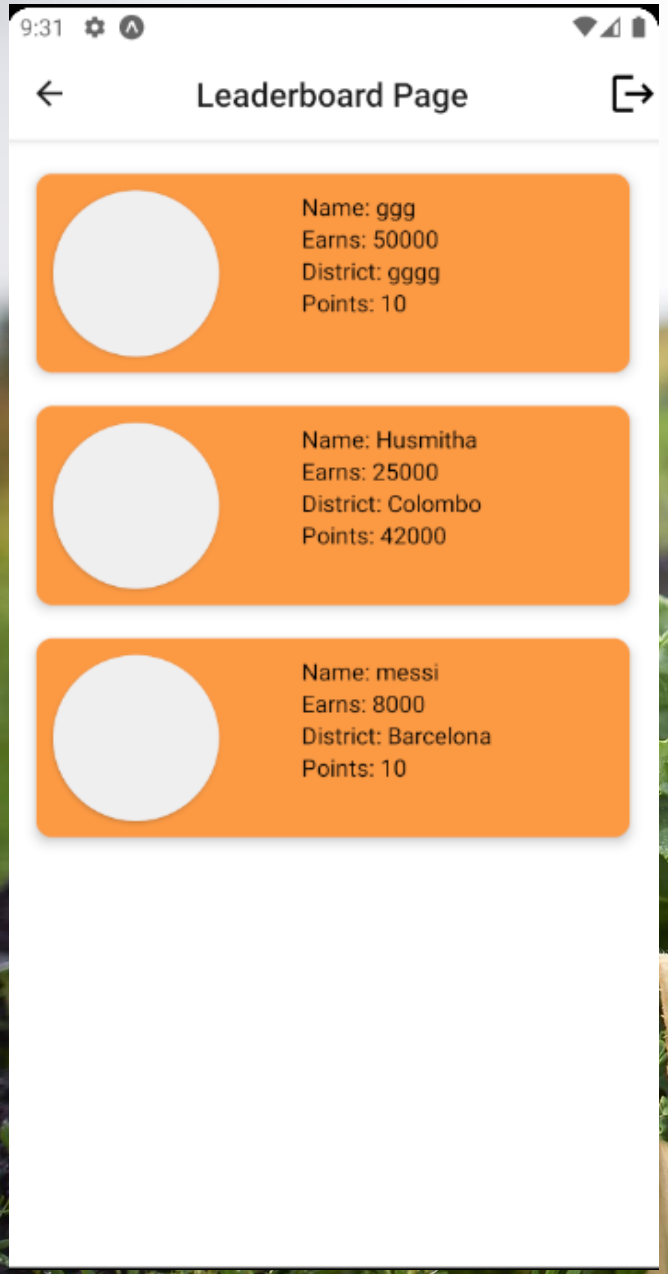
Methodology

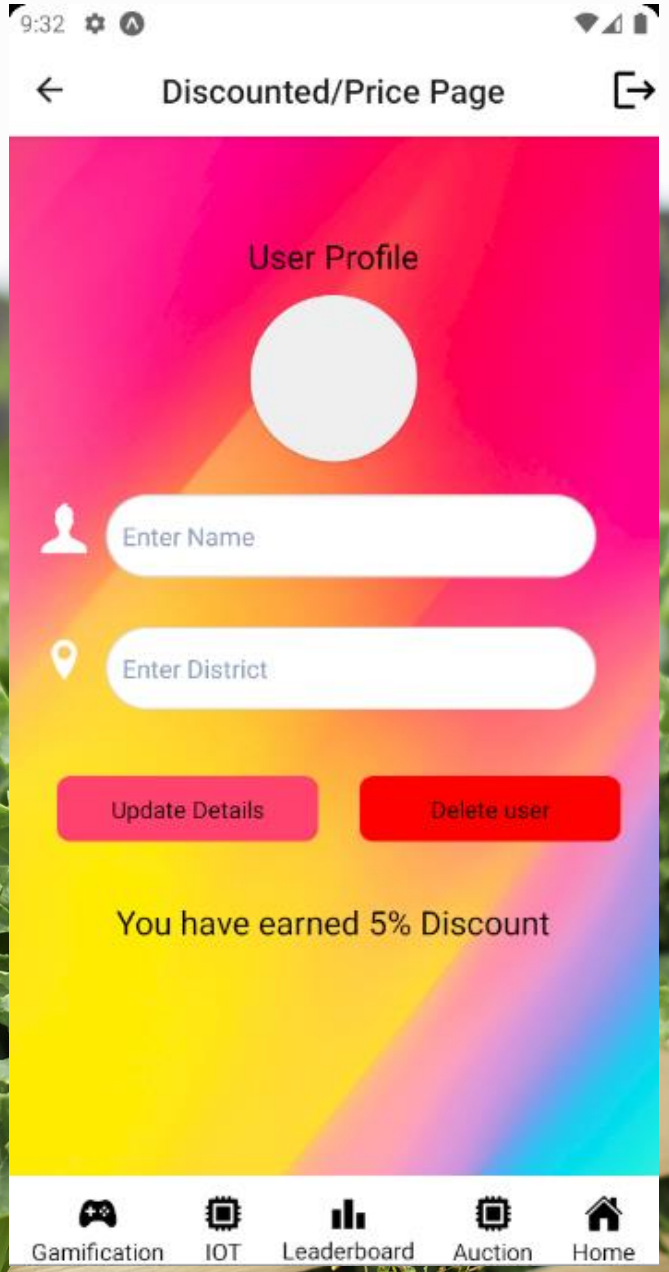
- Do a research, Watched videos and read motivational, psychological google articles for build UI
- Read articles of how colors effect humans and UX approaches
- According to sprint basic BE FE component implemented concurrently.
- Integrate FE BE using Axios, cors.
- 90% build basic UI and backend user, leaderboard CRUD operations


9:48
Heart of Agriculture
New update available, downloading...

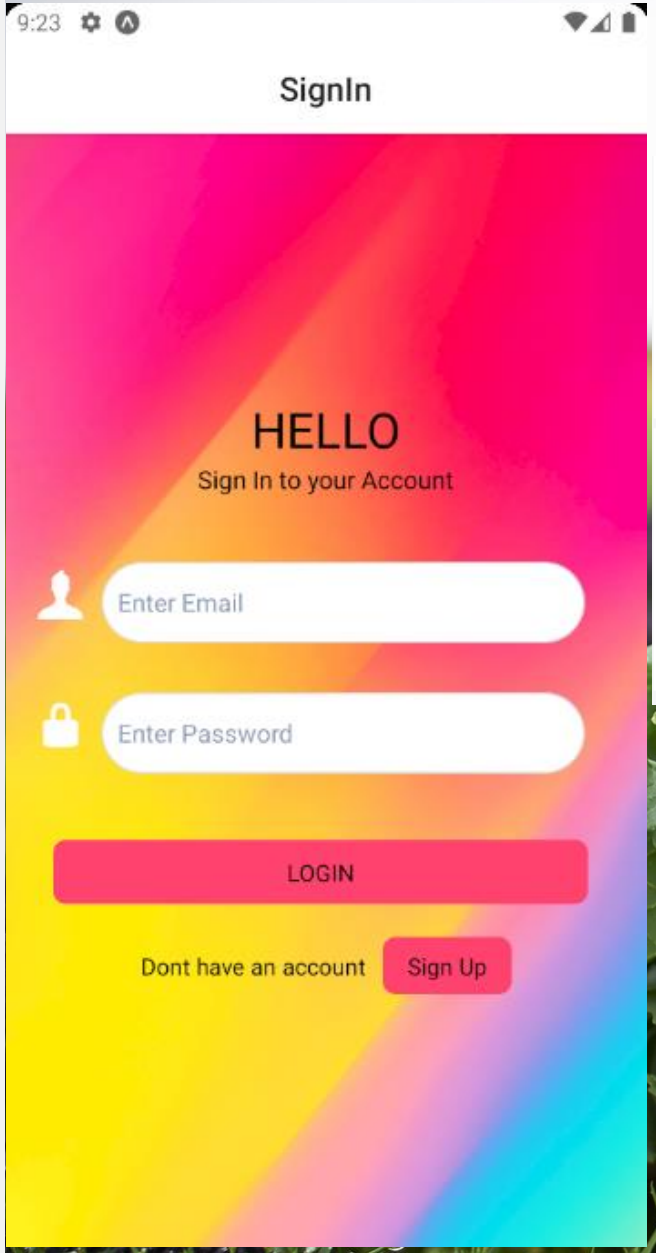

9:27
Agripreneurs
What is complete farmer all about?
Gamification
Auction
IOT
Leaderboard
All Right Reserved.
Gamification
IOT
Leaderboard
Auction
Home


9:29
Gamification Page
Enter Area
Area
Best Crop 1
Best Crop 2
Best Crop 3
Best Crop 4
TOTAL EARNED $: 1000
Gamification
IOT
Leaderboard
Auction
Home


9:31
Leaderboard Page
Name: ggg
Earns: 50000
District: gggg
Points: 10
Name: Husmitha
Earns: 25000
District: Colombo
Points: 42000
Name: messi
Earns: 8000
District: Barcelona
Points: 10

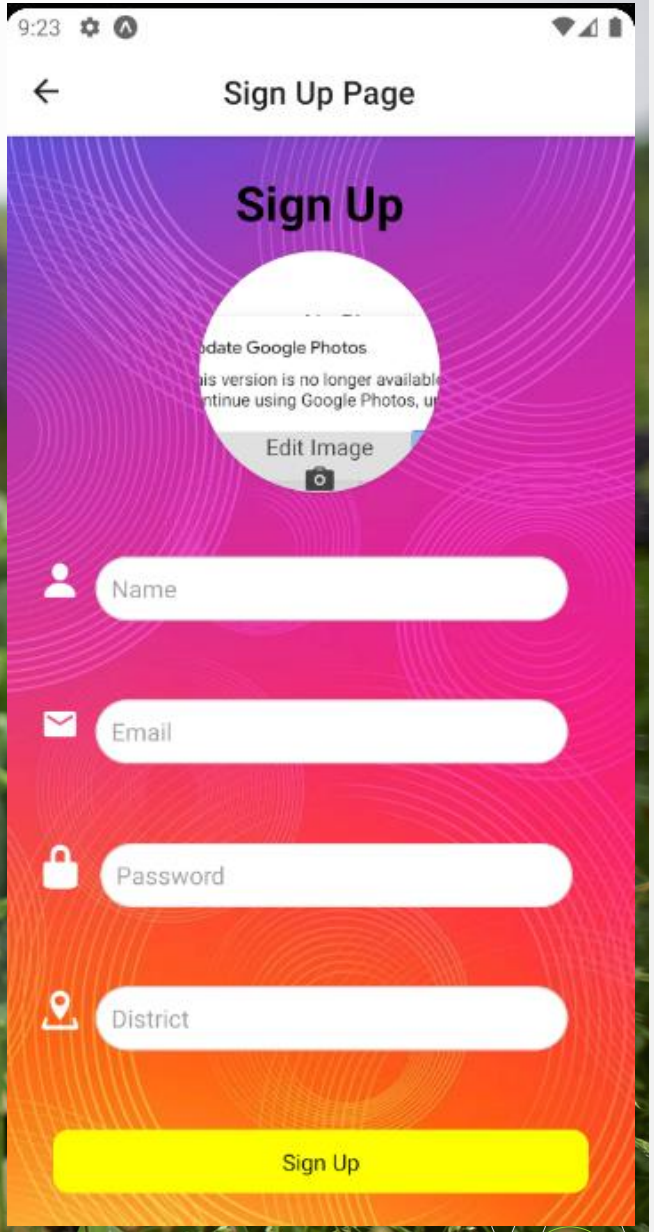

9:32
Discounted/Price Page
User Profile
Enter Name
Enter District
Update Details
Delete user
You have earned 5% Discount
Gamification
IOT
Leaderboard
Auction
Home


9:36
About Page
About Us
Agripreneurs solution Application created by the SLIIT final year Software Engineering undergraduates for the Research. Appreciated very much everyone who help in every way to make success the Application.
Vision: Empower the entrepreneurs who used the Application. guide them to become success entrepreneurs in their way. Mission: Hope to build the suceess most famous Agriculture Application in Sri Lanka
125, New Kandy road, Malabe. 0775555555 All Right Reserved.




```
_id: ObjectId("6166a9717000430e716d7528")
name: "user1"
email: "user1@user1.com"
password: "sha256$5214gSuQ0oJa5FPj$dc2ced1e1427a64ae70defe6ccad2490f3c246f20d09ba..."
district: "col"
prof_img_name: "file_data_user_0_host.exp.exponent_cache_ExperienceData_UNVERIFIED-192..."
```

```
_id: ObjectId("61670152f151ca257c77ef62")
name: "userofficial"
email: "officialu@user.com"
password: "sha256$WGZoUIJC91x3yjGz$d9ad0e060c45d1f0fcb1dea7544f101e0706cfc675eae8..."
district: "colombo"
prof_img_name: "file_data_user_0_host.exp.exponent_cache_ExperienceData_UNVERIFIED-192..."
```

- UI for the Mobile Application
- MongoDB-Database for CRUD operations for User
- Hope to add user guides, app tour

2021-090-frontend > screen > SignIn.js > ...

```
104
105             <View style={styles.sectionStyle}>
106               <View style={styles.textinputicon}>
107                 <Entypo name="lock" size={30} color="white" />
108                 <TextInput
109                   style={styles.input1}
110                   onChangeText={(val) => handlePasswordChange(val)}
111                   placeholder="Enter Password" //12345
112                   placeholderTextColor="#8b9cb5"
113                   // ref={passwordInputRef}
114                   // onSubmitEditing={Keyboard.dismiss}
115                   blurOnSubmit={false}
116                   secureTextEntry={true}
117                 />
118               </View>
119               {data.isValidPassword ? null : (
120                 <Text style={styles.errorMsg}>
121                   Password must be minimum 4 characters long.
122                 </Text>
123               )}
124             </View>
125             <View style={styles.buttons}>
126               <TouchableOpacity
127                 style={styles.buttonSub}
128                 activeOpacity={0.5}
129                 onPress={  () =>
130                   axios.post("http://192.168.8.126:5000/login", form)
131                     .then( async function (response) {
132                       // const stngobj = JSON.stringify()
133                       console.log(response.data);
134                       // alert("Successfully Logged in");
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE powershell

```
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Research project\Agripreneur_FE>
```

React Native Front end SignIn page

Agri_Backend > Agripreneur_App > Routes > PricePage.py > ...

```python
from extensions import mongo, db
from Main import app
from flask import Response, request, send_from_directory
# import pymongo
import json
from bson.objectid import ObjectId
from Agripreneur_App.Auth.Token import check_for_token
from bson.json_util import dumps
import os

# from json import stringify
@app.route("/getimage/<filename>", methods=["GET"])
# @check_for_token
def get_image(filename):
        print(request.headers)
        root_dir = os.path.dirname(os.getcwd())
        print(root_dir)
        return send_from_directory(os.path.join(root_dir, "Agri_Backend"), filename, as_attachment=True)


@app.route("/discount/<id>", methods=["GET"])
@check_for_token
def calculate_discount(id):
        # should have payload wi9th the user detaisl and it is send to FE

    user_earns = db.CropData.find_one({"_id": ObjectId(id)})
#       print(user_earns)

    user_points = user_earns['points']
#       print(user_points)

    if user_points == 1:
            discount = 1
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
C:\BE\Agri_Backend>.\venv\Scripts\activate

(venv) C:\BE\Agri_Backend>
```

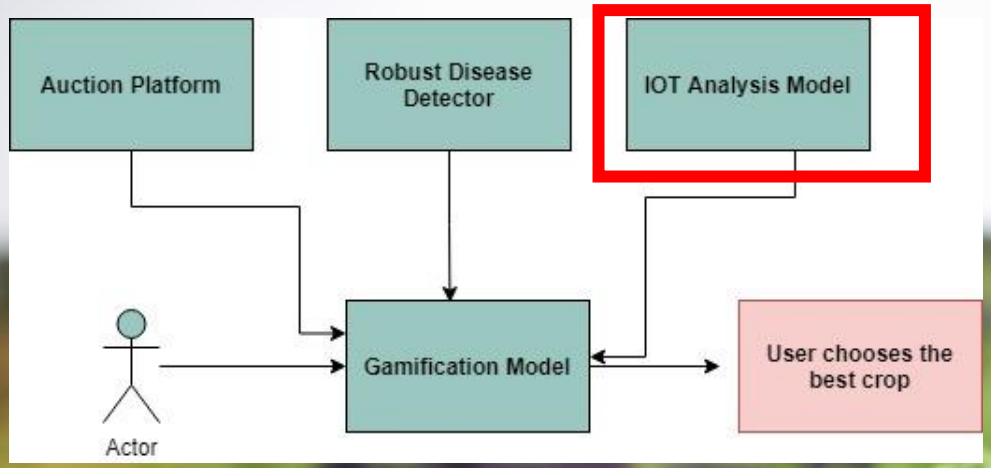
powershell
cmd

Flask Back end Discount Page

IT18111170 | Silva S.H.I | 2021-090

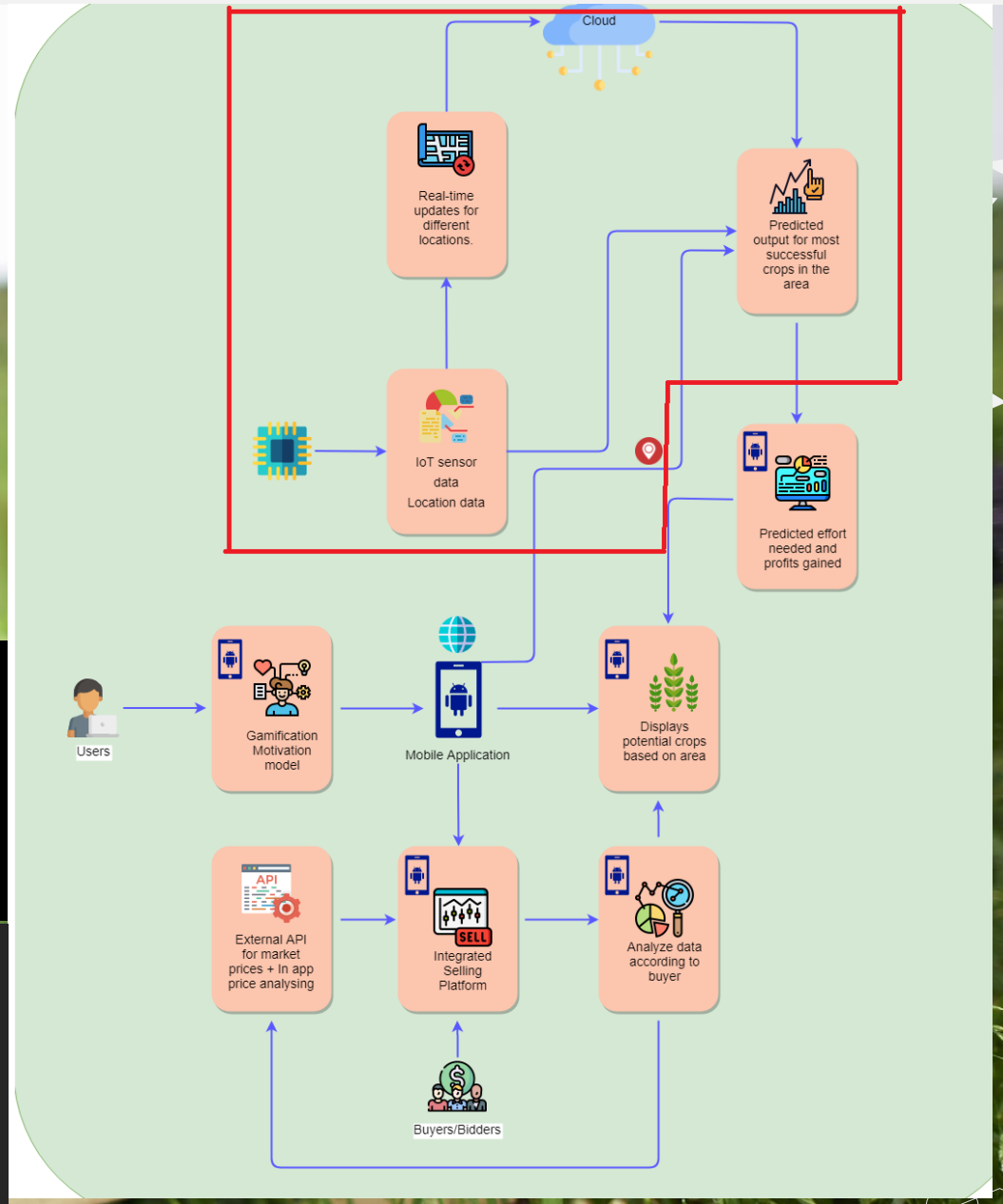
IT18027334 | Thilekarathna P.D.M

Specialization: Computer Systems & Network Engineering

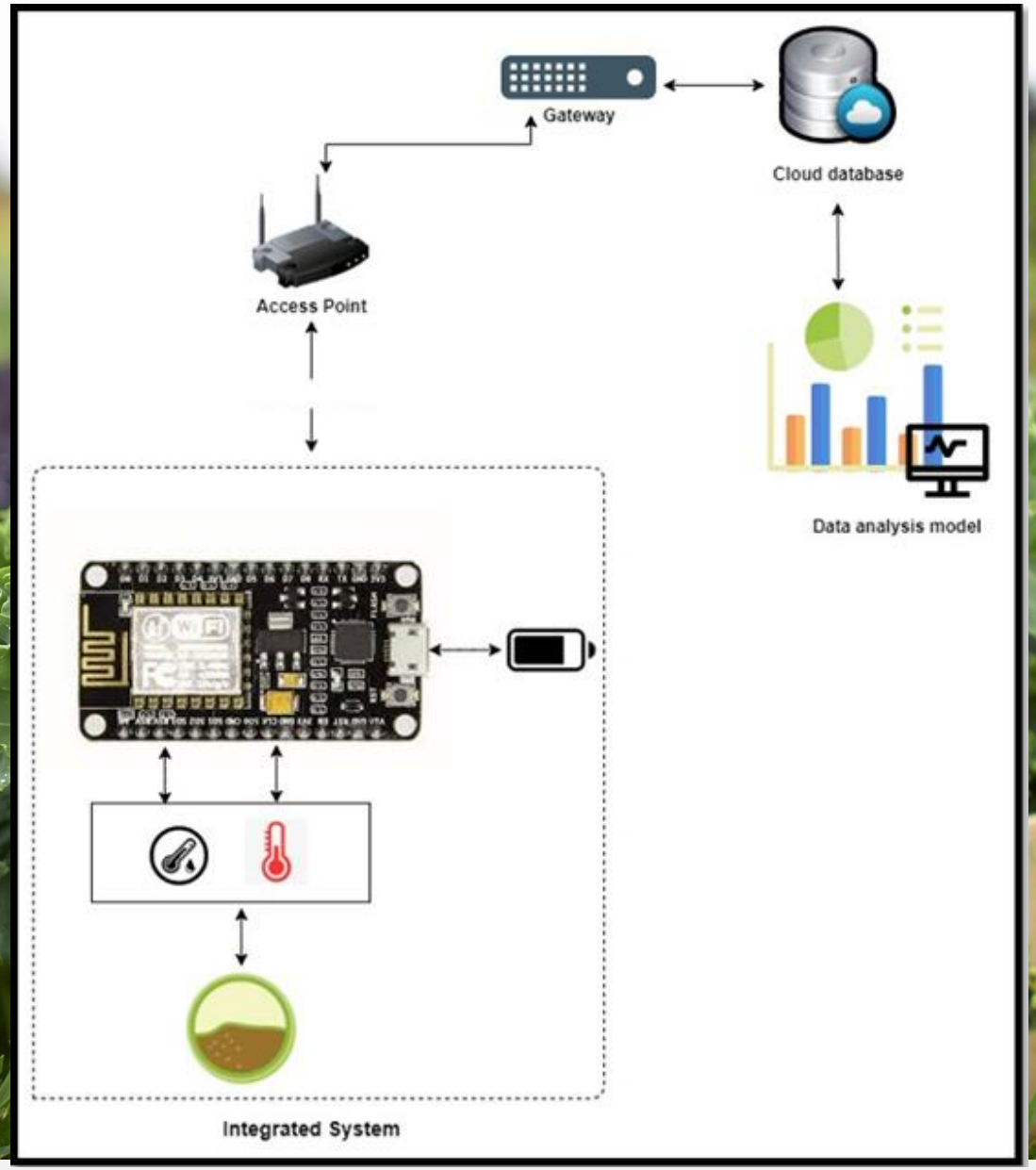


IoT based data analysis model

- ✓ Check the soil condition and Environment
- ✓ Data analysis model (ML model) of the suitable crops to be grown
- ✓ Real time data analysis



High Level System Architectural Diagram



RESEARCH GAP

Already existing systems are utilizing IoT methodologies;

- ✓ New type of devices
- ✓ Enables all smart farms to be connected

BUT LACK OF,

IoT methodology for motivate non- farmers.....

	Mobius using &Cube	Smart Agriculture app	Green Farm- DM	FieldCheck App	Agripreneurs
Using IoT sensor data to best crop prediction	NO	NO	NO	NO	YES
Real time update	YES	YES	YES	YES	YES
Motivational model	NO	NO	NO	NO	YES

RESEARCH QUESTION

- How does IoT effect to Motivate People to growing crops?
- How to motivate non-farmers using IoT methodologies?
- How to find most accurate ML model?
- Does it mobile device?

MAIN OBJECTIVE

•Building a IoT based data analysis model to predict the most suitable crop in the specific bare land

SUB OBJECTIVE

- Design IoT based sensor integrated hardware device to identify suitable areas of the land.
- Data analysis model to predict the suitable Crops.
- Real time update.

RESERCH METHODOLOGY

Technologies, Techniques

- NodeMCU development board as the Microcontroller (ESP8266 Microcontroller)
- Programming languages- Arduino & Python
- Tools- Arduino IDE / Jupyter notebook
- Database – Firebase
- Algorithms - Logistic Regression / Decision Tree / Random Forest / SVM / XGBoost / Naïve Bayes

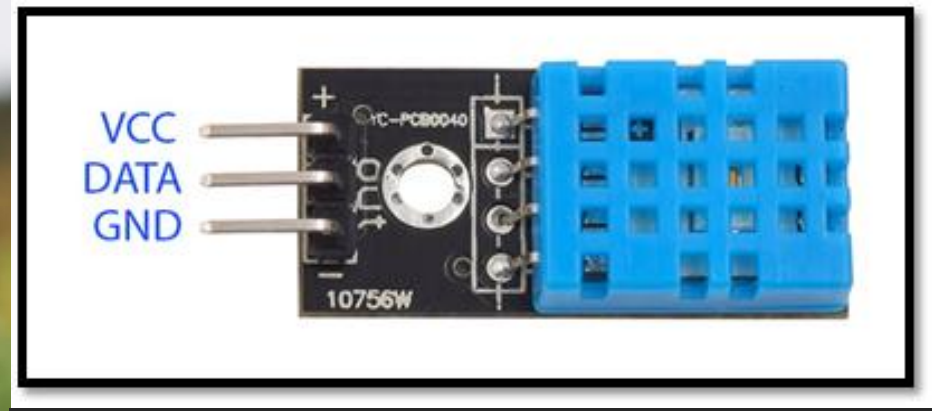
System, Software Requirements

- Network gateway need to be act as an intermediate.
- Humidity and temperature sensors (DHT11).
- Moisture detection sensor (YL-69)
- Mobile Application.
- HTTP Protocol

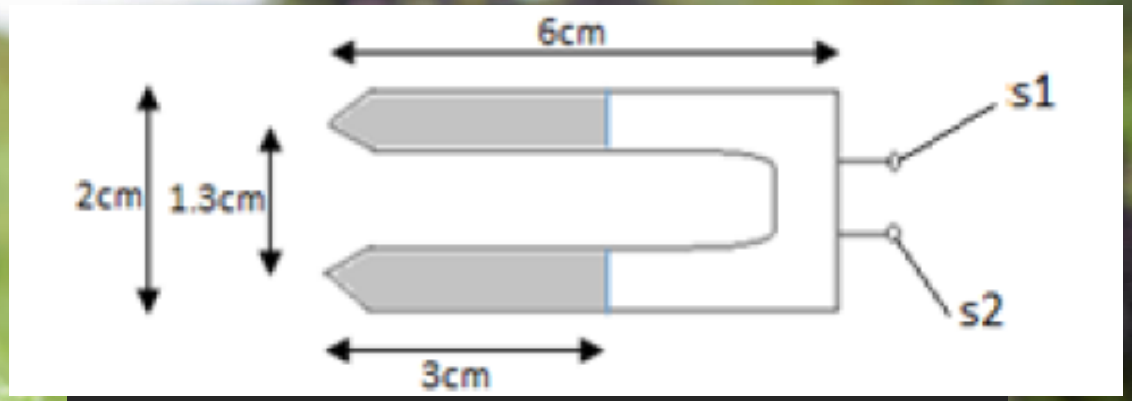
Data Sources

- Open data portal Sri Lanka
- NAICC- Govikam Magazines
- Department of Agriculture Srilanka Website
- Kaggle / YouTube tutorials / Stackoverflow

IoT Devices

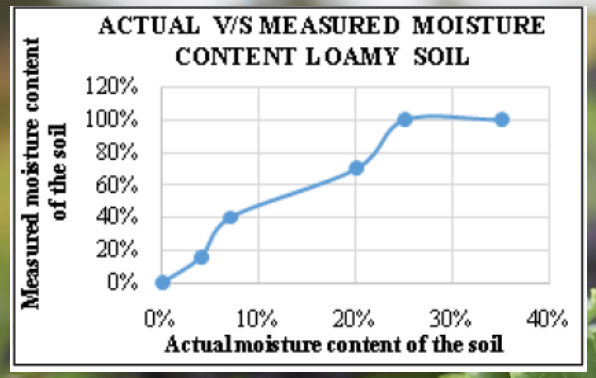


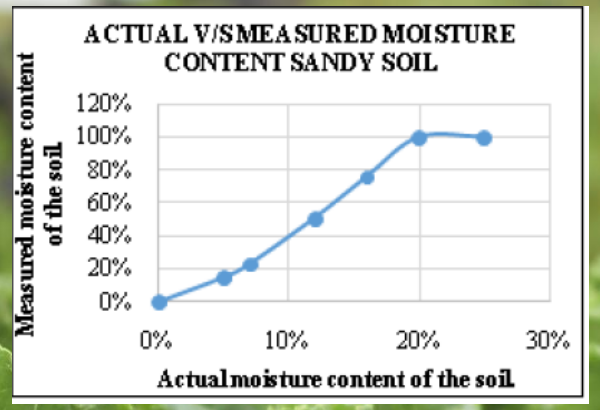
DHT11 Humidity/Temperature sensor

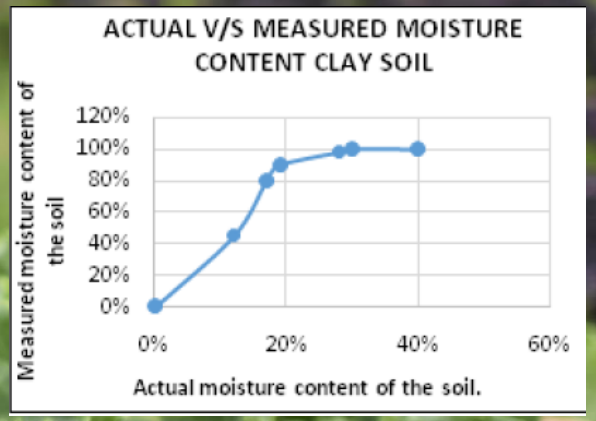


YL-69 Moisture sensor

Variations of different type of soil







Actual moisture content in soil	*Measured moisture content in soil*
0%	0%
4%	16%
7%	40%
20%	70%
25%	100%
35%	100%
40%	100%

Actual moisture content in soil	*Measured moisture content in soil*
0%	0%
5%	15%
7%	23%
12%	51%
16%	76%
20%	100%
25%	100%

Actual moisture content in soil	*Measured moisture content in soil*
0%	0%
12%	45%
13%	80%
19%	90%
28%	98%
30%	100%
40%	100%

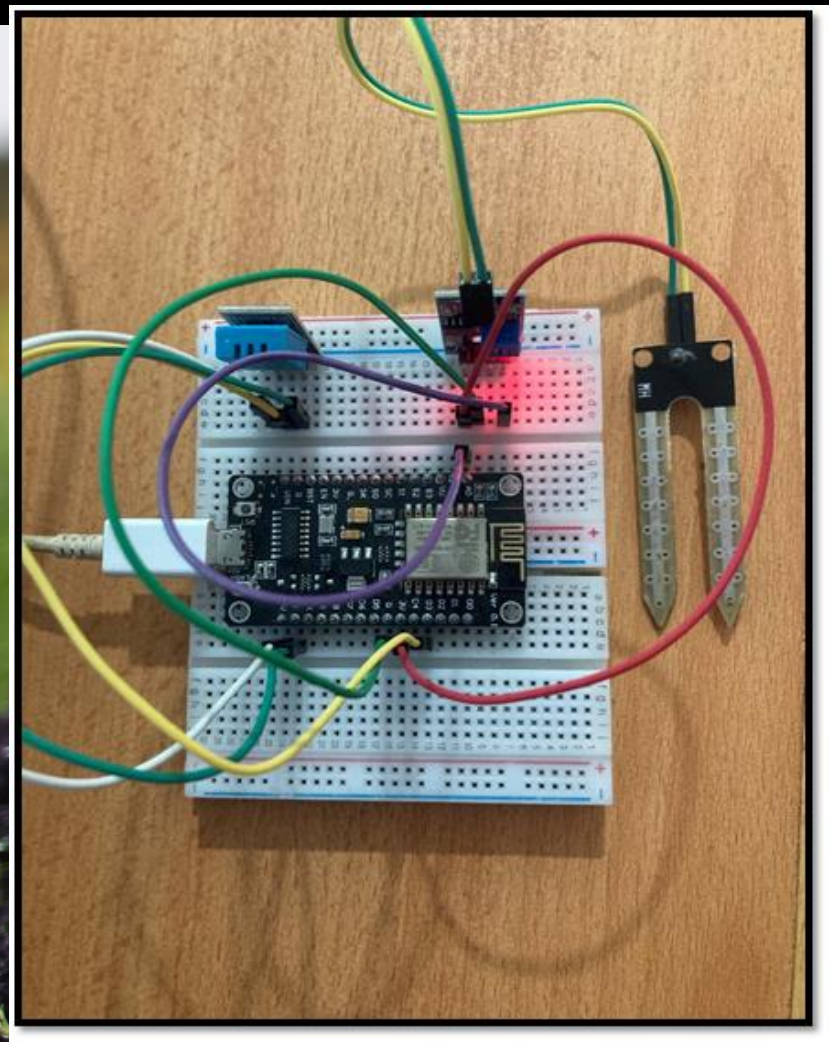
7% - 12%

Actual

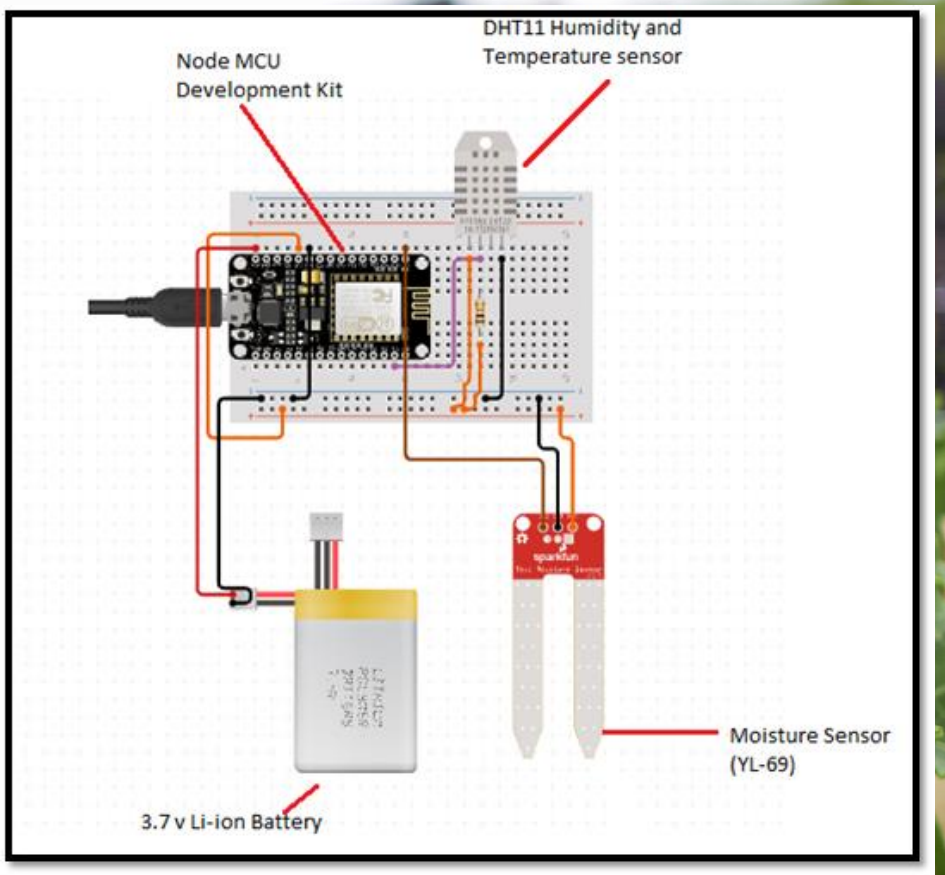
23% - 51%

Measured

Evidences for the completion



Hardware Device



Circuit diagram of IoT model

Evidences for the completion

```
In [27]:
from sklearn.linear_model import LogisticRegression

LogReg = LogisticRegression(random_state=2)

LogReg.fit(Xtrain,Ytrain)

predicted_values = LogReg.predict(Xtest)

x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('Logistic Regression')
print("Logistic Regression's Accuracy is: ", x)

print(classification_report(Ytest,predicted_values))
```

```
Logistic Regression's Accuracy is:  0.744776119402985
              precision    recall  f1-score   support

Bitter Gourd       0.52      0.44      0.48       101
     Brinjal       0.86      0.90      0.88       184
    Capsicum       0.92      0.91      0.91       166
      Chilli       0.60      0.78      0.68       178
      Tomato       0.00      0.00      0.00        41
```

Accuracy & cross validation score – Decision Tree

```
In [2]:
from __future__ import print_function
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import classification_report
from sklearn import metrics
from sklearn import tree
import warnings
warnings.filterwarnings('ignore')
```

```
In [3]:
PATH = 'F:\My Courses\DatasetFinal.csv'
df = pd.read_csv(PATH)
```

```
In [3]:
df.head()
```

Out[3]:

	Moisture	Humidity	Temperature	Crop
0	28	55	21	Chilli
1	28	55	22	Chilli
2	28	55	23	Chilli
3	28	55	24	Chilli
4	28	55	25	Chilli

Retrieving sample data set

Evidences for the completion

```
In [24]: from sklearn.naive_bayes import GaussianNB

         NaiveBayes = GaussianNB()

         NaiveBayes.fit(Xtrain,Ytrain)

         predicted_values = NaiveBayes.predict(Xtest)
         x = metrics.accuracy_score(Ytest, predicted_values)
         acc.append(x)
         model.append('Naive Bayes')
         print("Naive Bayes's Accuracy is: ", x)

         print(classification_report(Ytest,predicted_values))
```

```
Naive Bayes's Accuracy is:  0.6785714285714286
              precision    recall  f1-score   support

Bitter Gourd       0.50      0.71      0.59         7
     Brinjal       1.00      0.75      0.86         8
    Capsicum       0.94      0.85      0.89        20
      Chilli       0.53      0.64      0.58        14
      Tomato       0.20      0.14      0.17         7

    accuracy                           0.68        56
   macro avg       0.63      0.62      0.62        56
weighted avg       0.70      0.68      0.68        56
```

```
In [25]: # Cross validation score (NaiveBayes)
         score = cross_val_score(NaiveBayes,features,target,cv=5)
         score

Out[25]: array([0.60714286, 0.69642857, 0.89090909, 0.76363636, 0.61818182])
```

Previous Accuracy

```
In [21]: 1  from sklearn.naive_bayes import GaussianNB
         2
         3  NaiveBayes = GaussianNB()
         4
         5  NaiveBayes.fit(Xtrain,Ytrain)
         6
         7  predicted_values = NaiveBayes.predict(Xtest)
         8  x = metrics.accuracy_score(Ytest, predicted_values)
         9  acc.append(x)
         10 model.append('Naive Bayes')
         11 print("Naive Bayes's Accuracy is: ", x)
         12
         13 print(classification_report(Ytest,predicted_values))
```

```
Naive Bayes's Accuracy is:  0.7597014925373134
              precision    recall  f1-score   support

Bitter Gourd       0.64      0.61      0.63       101
     Brinjal       0.94      0.90      0.92       184
    Capsicum       0.94      0.86      0.90       166
      Chilli       0.62      0.65      0.63       178
```

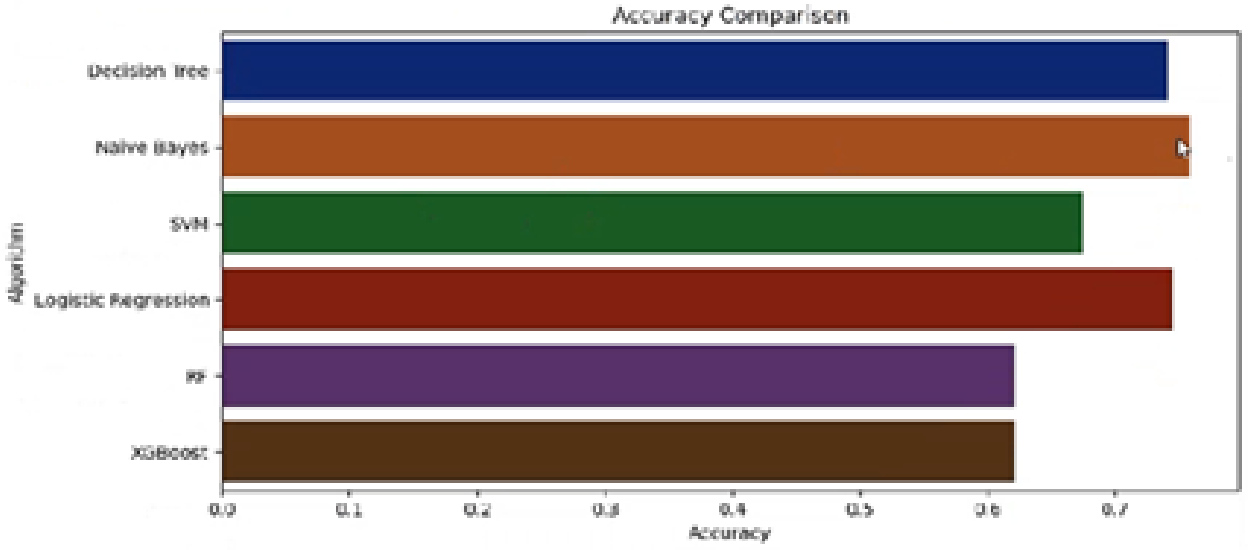
Present Accuracy

Accuracy & cross validation score- Naïve Bayes

Evidences for the completion

```
In [35]:
plt.figure(figsize=[10,5],dpi = 100)
plt.title('Accuracy Comparison')
plt.xlabel('Accuracy')
plt.ylabel('Algorithm')
sns.barplot(x = acc,y = model,palette='dark')
```

```
Out[35]: <AxesSubplot:title={'center':'Accuracy Comparison'}, xlabel='Accuracy', ylabel='Algorithm'>
```



Accuracy Comparison

```
In [36]:
data = np.array([[58,75,29.5]])
prediction = NaiveBayes.predict(data)
print(prediction)
```

```
['Chilli']
```

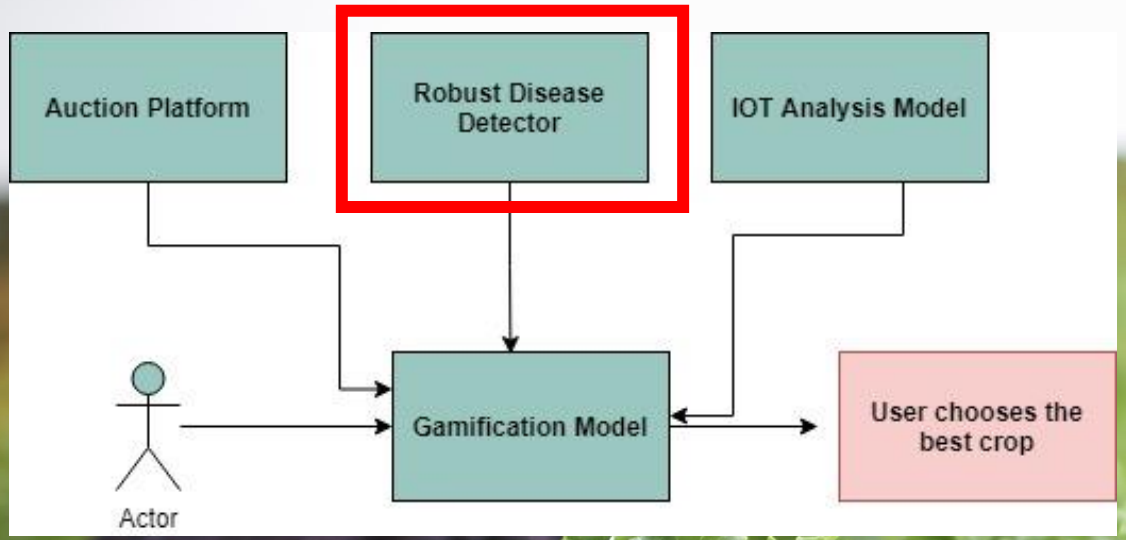
```
In [38]:
data = np.array([[30,31.40,25]])
prediction = RF.predict(data)
print(prediction)
```

```
['Brinjal']
```

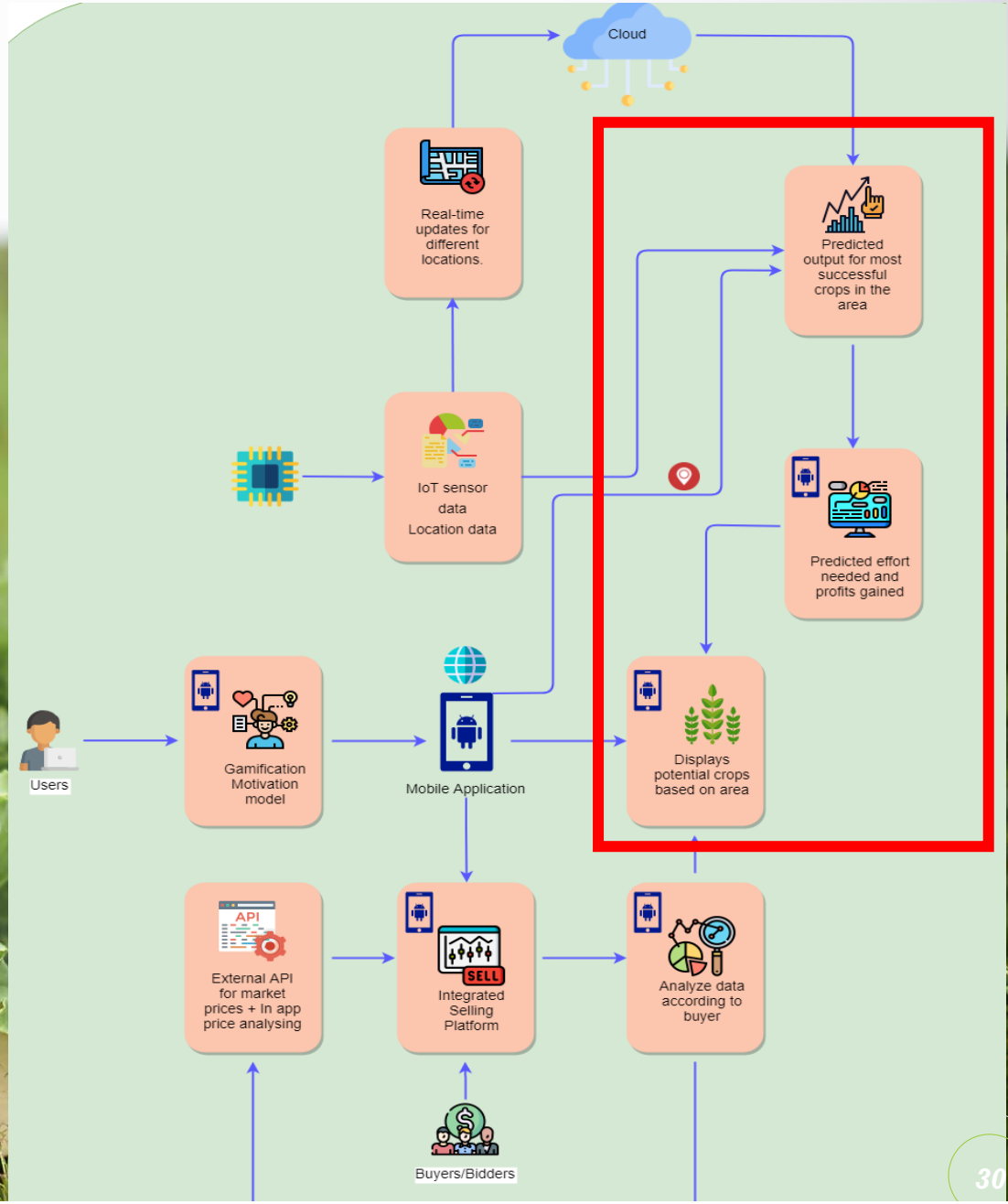
Predict more accurate result

IT18110562 | R.S.W.M.R.L. Rangalla

Specializing : Software Engineering



Robust Image processing model & Prediction model



Background

- Prediction machine learning mainly involves in thee supervised machine learning domain
- Detect abnormal crop groups due to diseases.
- Machine learning prediction algorithms have been evolving to give more predictive and accurate results in the form of fertilizers and other recommendations for users.

Research Question

- To build a robust plant disease detection model.
- Predicting fertilizer recommendations for users.

Research Gap

- **Robust image processing model** to detect disease over a wide range of crops and plants
- **Prediction model to recommend users with appropriate recommendations for each plant disease.**

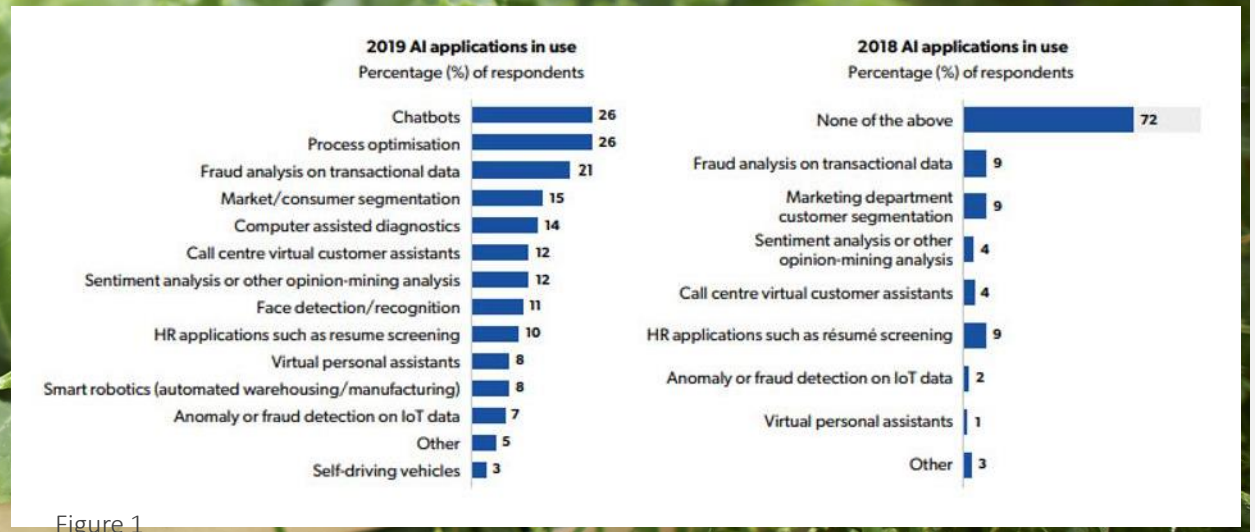


Figure 1

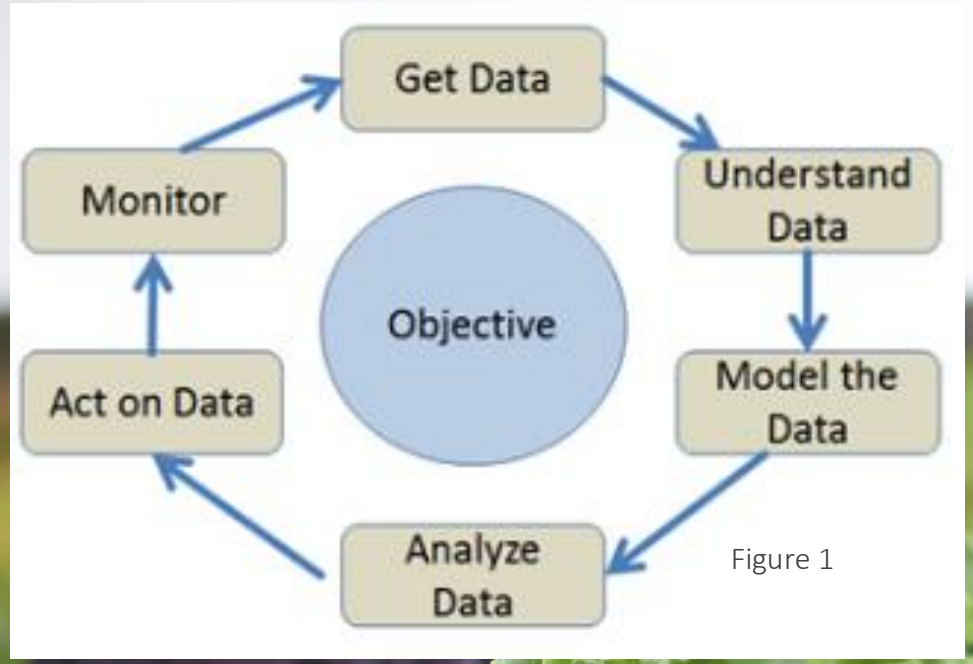


Figure 1

Objectives

- Motivate users.
- Robust Image processing model.
- Success

Sub objectives

- Discovering the most appropriate algorithm for disease detection.
- Robust image processing model.
- Detect abnormal plants by analysis the uploaded picture.
- Provide appropriate solutions/fertilizers.
- Obtaining a reliable dataset.

Methodology

- Built a data preprocessing model for feature extraction.
- Done by feature extraction methods which included Hu moments, Haralick and color histogram.
- To robustly utilize the model, only 2 main training labels are being introduced which are diseased plants and healthy plants respectively for only Apple Leaves
- Colors of the leaves are being analyzed to train the image processing model.

Click to add text

- The data-set is being divided into a ratio of 0.8 to 0.2 for validation and training respectively.
- 10 Fold Cross validation is done to obtain an accurate model, since a limited data set is available.
- Following this more classes were introduced.
- Accuracy of the model was 70 % on 38 classes

Algorithms

- All popular model training algorithms were compared with each other to other.
- Random Forest Tree had the highest accuracy in detecting a diseased plant of 98%.

Tools and Technologies

- Python 3 using Jupyter Notebook
- Tensorflow
- OpenCV
- Sklearn
- H5py to store data.
- Joblib to increase efficency.
- Azure Cloud
- React Native Expo

Click to add text

Algorithms

- Support Vector Machine
- Random Forest
- Naïve Bayes
- Decision Trees
- Logistic Regression
- Linear Discriminant

Evidence for Completion

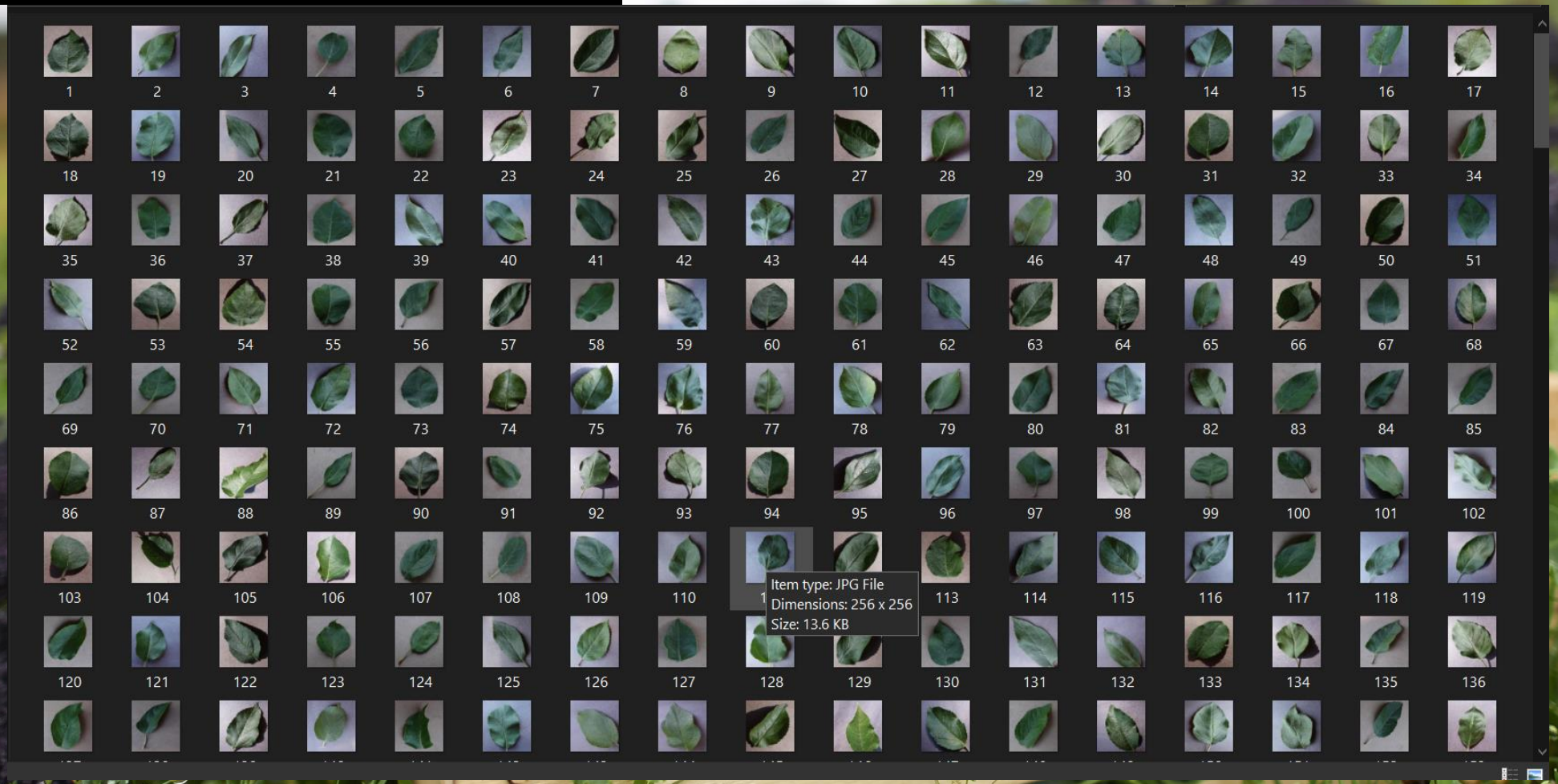


Figure B: Healthy Leaves for apple

Evidence for Completion

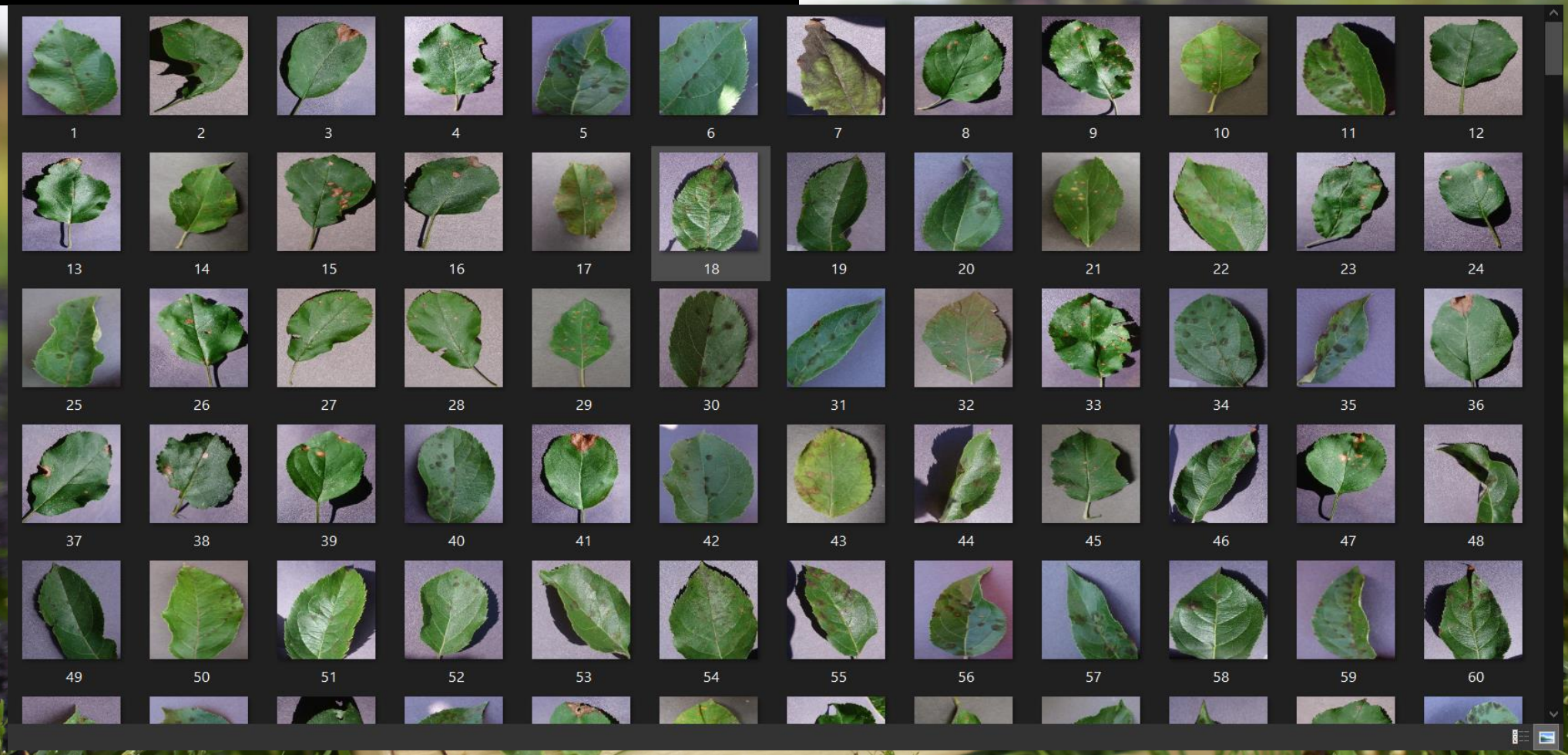


Figure B: Diseased Leaves for apples

Evidence for Completion

```python
#feature descriptor1: Hu moments (yellowish/fire)
def fd_hu_moments(image):
    image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    feature = cv2.HuMoments(cv2.moments(image)).flatten()
    return feature
```

```python
#feature descriptor2: Haralick texture(Quantify image according to the texture)
def fd_haralick(image):
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    haralick = mahotas.features.haralick(gray).mean(axis = 0)
    return haralick
```

```python
#featur descriptor3: Color Histogram
def fd_histogram(image, mask = None):
    image = cv2.cvtColor(image,cv2.COLOR_BGR2HSV)
    hist = cv2.calcHist([image],[0,1,2],None,[bins,bins,bins],[0,256,0,256,0,256])
    cv2.normalize(hist,hist)
    return hist.flatten()
```

Figure C: Defining the feature extraction algorithms

Evidence for Completion

jupyter classification_model_final Last Checkpoint: 6 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Run Code

In [10]:

```
# for loop to loop over the traininf data in the subfolder

for training_name in train_labels:

    #joining training data path and each species training folder
    dir = os.path.join(train_path, training_name)

    #getting the current training label
    current_label = training_name

    #looping over the images in each subfolder
    for x in range(1,images_per_class+1):

        #joining strings to get the image file name
        file = dir + "/" +str(x) + ".jpg"

        image =  cv2.imread(file)
        image = cv2.resize(image, fixed_size)

        #Running the function bit by bit

        RGB_BGR = rgb_bgr(image)
        BGR_HSV = bgr_hsv(RGB_BGR)
        IMG_SEG = img_segmentation(RGB_BGR, BGR_HSV)

        #Calling global feature descriptors

        fv_hu_moments = fd_hu_moments(IMG_SEG)
        fv_haralick = fd_haralick(IMG_SEG)
        fv_histogram = fd_histogram(IMG_SEG)

        #Concatenate
```

Figure D: Applying the feature extraction functions to the dataset, Looping over all the images on .jpg format

jupyter classification_model_final Last Checkpoint: 6 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Connected | Not Trusted | Python 3

Run Code

```
print("[Status] preprocessed folder:{}".format(current_label))

print("[Status] completed global feature extraction")
```

```
[Status] preprocessed folder:diseased
[Status] completed global feature extraction
[Status] preprocessed folder:diseased
[Status] completed global feature extraction
[Status] preprocessed folder:diseased
[Status] completed global feature extraction
[Status] preprocessed folder:diseased
[Status] completed global feature extraction
[Status] preprocessed folder:diseased
[Status] completed global feature extraction
[Status] preprocessed folder:diseased
[Status] completed global feature extraction
[Status] preprocessed folder:diseased
[Status] completed global feature extraction
[Status] preprocessed folder:diseased
[Status] completed global feature extraction
[Status] preprocessed folder:diseased
[Status] completed global feature extraction
[Status] preprocessed folder:diseased
[Status] completed global feature extraction
[Status] preprocessed folder:diseased
[Status] completed global feature extraction
[Status] preprocessed folder:diseased
[Status] completed global feature extraction
```

In [11]:

```
print("[Status] feature vector size{}",format(np.array(labels).shape))
```

```
[Status] feature vector size{} (1600,)
```

Figure E : Shows the result upon successful execution

Evidence for Completion

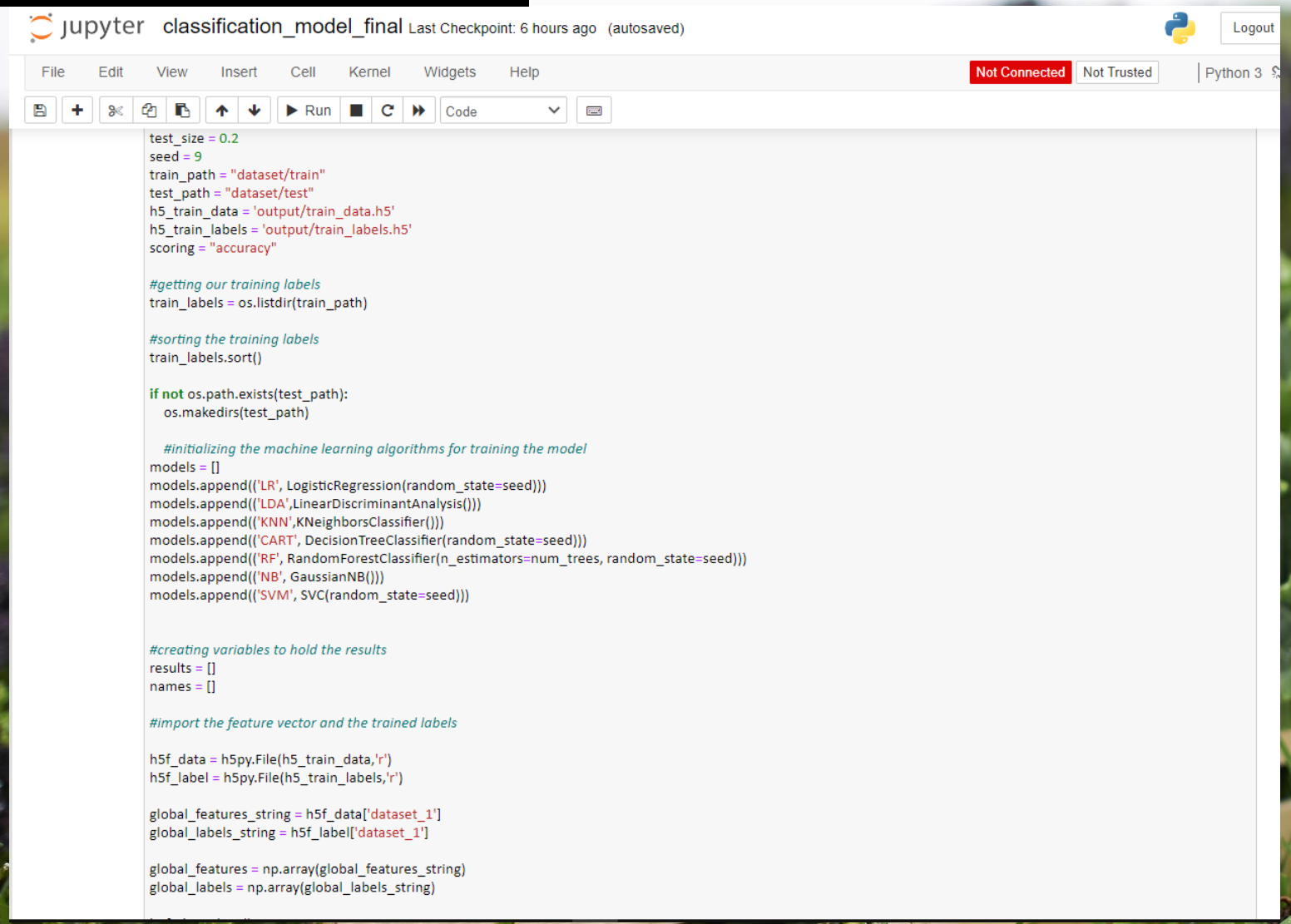


Figure F: Training the model

Evidence for Completion

jupyter classification_model_final Last Checkpoint: 6 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Connected | Not Trusted | Python 3

Run | Code

In [33]:
```
#10-fold cross validation to get the accuracy of each of the algorithms being used for training the model

for name, model in models:
    kfold = KFold(n_splits=10, random_state=seed,shuffle = True)
    cv_results = cross_val_score(model, trainDataGlobal, trainLabelsGlobal, cv=kfold, scoring=scoring)
    results.append(cv_results)
    names.append(name)
    msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
    print(msg)

# boxplot algorithm comparison
fig = pyplot.figure()
fig.suptitle('Machine Learning algorithm comparison')
ax = fig.add_subplot(111)
pyplot.boxplot(results)
ax.set_xticklabels(names)
pyplot.show()
```

```
LR: 0.916406 (0.021833)
LDA: 0.907813 (0.019702)
KNN: 0.920312 (0.012500)
CART: 0.906250 (0.027951)
RF: 0.957031 (0.013189)
NB: 0.857031 (0.010511)
SVM: 0.916406 (0.020086)
```

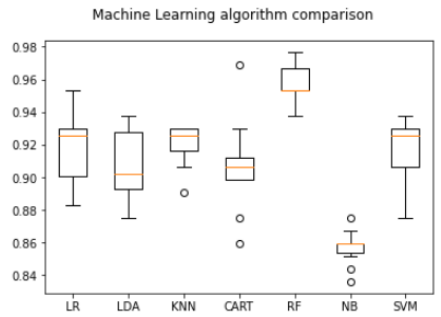


Figure G: Output of the model

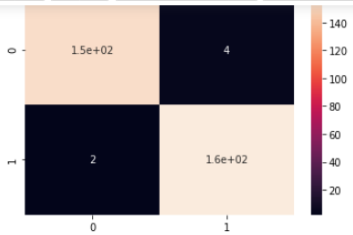
jupyter classification_model_final Last Checkpoint: 7 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Connected | Not Trusted | Python 3

Run | Code



In [47]:
```
print(classification_report(testLabelsGlobal, y_predict))
```

```
              precision    recall  f1-score   support

           0       0.99      0.97      0.98       158
           1       0.98      0.99      0.98       162

    accuracy                           0.98       320
   macro avg       0.98      0.98      0.98       320
weighted avg       0.98      0.98      0.98       320
```

In [49]:
```
from sklearn.metrics import accuracy_score
accuracy_score(testLabelsGlobal, y_predict)
```

Out[49]: 0.98125

In [ ]:

Figure H: Random Forest Tree gives 98% accuracy

Evidence for Completion

Python

```
{'Apple___Apple_scab': 0, 'Apple___Black_rot': 1, 'Apple___Cedar_apple_rust': 2, 'Apple___healthy': 3, 'Blueberry___healthy': 4,
'Cherry_(including_sour)___Powdery_mildew': 5, 'Cherry_(including_sour)___healthy': 6, 'Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot': 7,
'Corn_(maize)___Common_rust_': 8, 'Corn_(maize)___Northern_Leaf_Blight': 9, 'Corn_(maize)___healthy': 10, 'Grape___Black_rot': 11,
'Grape___Esca_(Black_Measles)': 12, 'Grape___Leaf_blight_(Isariopsis_Leaf_Spot)': 13, 'Grape___healthy': 14, 'Orange___Haunglongbing_(Citrus_greening)': 15,
'Peach___Bacterial_spot': 16, 'Peach___healthy': 17, 'Pepper,_bell___Bacterial_spot': 18, 'Pepper,_bell___healthy': 19, 'Potato___Early_blight': 20,
'Potato___Late_blight': 21, 'Potato___healthy': 22, 'Raspberry___healthy': 23, 'Soybean___healthy': 24, 'Squash___Powdery_mildew': 25,
'Strawberry___Leaf_scorch': 26, 'Strawberry___healthy': 27, 'Tomato___Bacterial_spot': 28, 'Tomato___Early_blight': 29, 'Tomato___Late_blight': 30,
'Tomato___Leaf_Mold': 31, 'Tomato___Septoria_leaf_spot': 32, 'Tomato___Spider_mites Two-spotted_spider_mite': 33, 'Tomato___Target_Spot': 34,
'Tomato___Tomato_Yellow_Leaf_Curl_Virus': 35, 'Tomato___Tomato_mosaic_virus': 36, 'Tomato___healthy': 37}
```

```
li = list(class_dict.keys())
print(li)
```

Python

```
['Apple___Apple_scab', 'Apple___Black_rot', 'Apple___Cedar_apple_rust', 'Apple___healthy', 'Blueberry___healthy', 'Cherry_(including_sour)___Powdery_mildew',
'Cherry_(including_sour)___healthy', 'Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot', 'Corn_(maize)___Common_rust_',
'Corn_(maize)___Northern_Leaf_Blight', 'Corn_(maize)___healthy', 'Grape___Black_rot', 'Grape___Esca_(Black_Measles)',
'Grape___Leaf_blight_(Isariopsis_Leaf_Spot)', 'Grape___healthy', 'Orange___Haunglongbing_(Citrus_greening)', 'Peach___Bacterial_spot', 'Peach___healthy',
'Pepper,_bell___Bacterial_spot', 'Pepper,_bell___healthy', 'Potato___Early_blight', 'Potato___Late_blight', 'Potato___healthy', 'Raspberry___healthy',
'Soybean___healthy', 'Squash___Powdery_mildew', 'Strawberry___Leaf_scorch', 'Strawberry___healthy', 'Tomato___Bacterial_spot', 'Tomato___Early_blight',
'Tomato___Late_blight', 'Tomato___Leaf_Mold', 'Tomato___Septoria_leaf_spot', 'Tomato___Spider_mites Two-spotted_spider_mite', 'Tomato___Target_Spot',
'Tomato___Tomato_Yellow_Leaf_Curl_Virus', 'Tomato___Tomato_mosaic_virus', 'Tomato___healthy']
```

Figure H: Addition of all 38 classes

Evidence for Completion

```
Epoch 1/10
150/150 [==============================] - 417s 3s/step - loss: 7.0892 - accuracy: 0.0561 - val_loss: 0.0822 - val_accuracy: 0.9717
Epoch 2/10
150/150 [==============================] - 345s 2s/step - loss: 5.8732 - accuracy: 0.0612 - val_loss: 0.0945 - val_accuracy: 0.9685
Epoch 3/10
150/150 [==============================] - 331s 2s/step - loss: 5.3996 - accuracy: 0.0636 - val_loss: 0.1110 - val_accuracy: 0.9645
Epoch 4/10
150/150 [==============================] - 321s 2s/step - loss: 5.1039 - accuracy: 0.0656 - val_loss: 0.1275 - val_accuracy: 0.9609
Epoch 5/10
150/150 [==============================] - 308s 2s/step - loss: 4.8865 - accuracy: 0.0745 - val_loss: 0.1389 - val_accuracy: 0.9602
Epoch 6/10
150/150 [==============================] - 298s 2s/step - loss: 4.7639 - accuracy: 0.0699 - val_loss: 0.1483 - val_accuracy: 0.9602
Epoch 7/10
150/150 [==============================] - 300s 2s/step - loss: 4.6131 - accuracy: 0.0775 - val_loss: 0.1589 - val_accuracy: 0.9598
Epoch 8/10
150/150 [==============================] - 289s 2s/step - loss: 4.6213 - accuracy: 0.0657 - val_loss: 0.1751 - val_accuracy: 0.9572
Epoch 9/10
150/150 [==============================] - 292s 2s/step - loss: 4.5059 - accuracy: 0.0702 - val_loss: 0.1877 - val_accuracy: 0.9547
Epoch 10/10
150/150 [==============================] - 290s 2s/step - loss: 4.4780 - accuracy: 0.0708 - val_loss: 0.1941 - val_accuracy: 0.9550
```

Figure I: Training model for 150 Epochs to increase accuracy

```
import os
import pickle
import predict
import numpy as np

Apple_diseases=['Apple___Apple_scab','Apple___Black_rot','Apple___Cedar_apple_rust','Apple___healthy']
Badam_diseases=['bad','good']
Blueberry_diseases=['Blueberry___healthy']
Cherry_diseases=['Cherry_(including_sour)___healthy','Cherry_(including_sour)___Powdery_mildew']
Corn_diseases=['Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot','Corn_(maize)___Common_rust_','Corn_(maize)___healthy','Corn_(maize)___Nort
Grape_diseases=['Grape___Black_rot','Grape___Esca_(Black_Measles)','Grape___healthy','Grape___Leaf_blight_(Isariopsis_Leaf_Spot)']
Orange_diseases=['Orange___Haunglongbing_(Citrus_greening)']
Paddy_diseases=['Bacterial leaf blight','Brown spot','Leaf smut']
Peach_diseases=['Peach___Bacterial_spot','Peach___healthy']
Pepper_diseases=['Pepper,_bell___Bacterial_spot','Pepper,_bell___healthy']
Potato_diseases=['Potato___Early_blight','Potato___healthy','Potato___Late_blight']
Raspberry_diseases=['Raspberry___healthy']
Rose_diseases=['anthracnose','mildew','rust','spot']
Soyabean_diseases=['Soybean___healthy']
Squash_diseases=['Squash___Powdery_mildew']
Strawberry_diseases=['Strawberry___healthy','Strawberry___Leaf_scorch']
Sunflower_diseases=['blight','mildew','rust']
Tomato_diseases=['Tomato___Bacterial_spot','Tomato___Early_blight','Tomato___healthy','Tomato___Late_blight','Tomato___Leaf_Mold','Tomato___Sep

diseases = [Apple_diseases,Badam_diseases,Blueberry_diseases,Cherry_diseases,Corn_diseases,Grape_diseases,Orange_diseases,Paddy_diseases,Peach_
Species = ["Apple","Badam","Blueberry","Cherry","Corn","Grape","Orange","Paddy","Peach","Pepper","Potato","Raspberry","Rose","Soyabean","Squash
```

Figure J: All training classes and their subclasses

Evidence for Completion

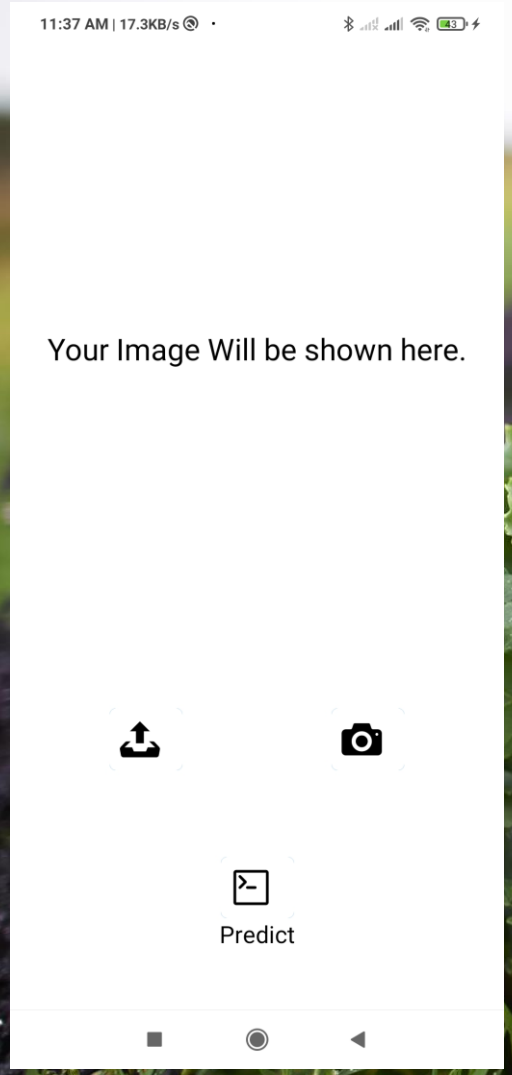
```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 54, 54, 96)        34944
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 27, 27, 96)        0
_________________________________________________________________
batch_normalization_1 (Batch (None, 27, 27, 96)        384
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 17, 17, 256)       2973952
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 8, 8, 256)         0
_________________________________________________________________
batch_normalization_2 (Batch (None, 8, 8, 256)         1024
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 6, 6, 384)         885120
_________________________________________________________________
batch_normalization_3 (Batch (None, 6, 6, 384)         1536
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 4, 4, 384)         1327488
_________________________________________________________________
batch_normalization_4 (Batch (None, 4, 4, 384)         1536
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 2, 2, 256)         884992
show more (open the raw output data in a text editor) ...
=================================================================
Total params: 28,117,790
Trainable params: 4,137,038
Non-trainable params: 23,980,752
_________________________________________________________________
```

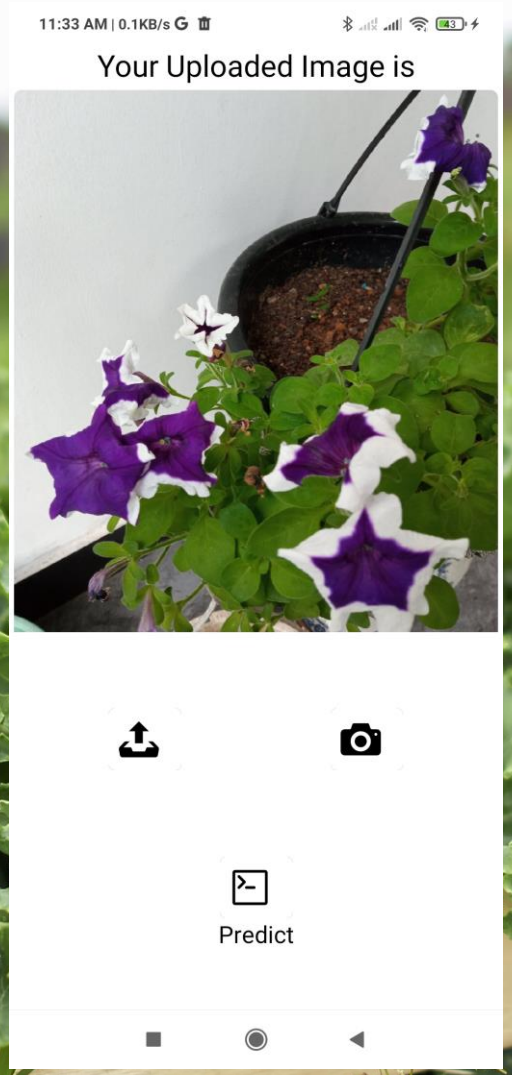
Figure K: Using Keras 2d layers and batch normalization for the main validation model

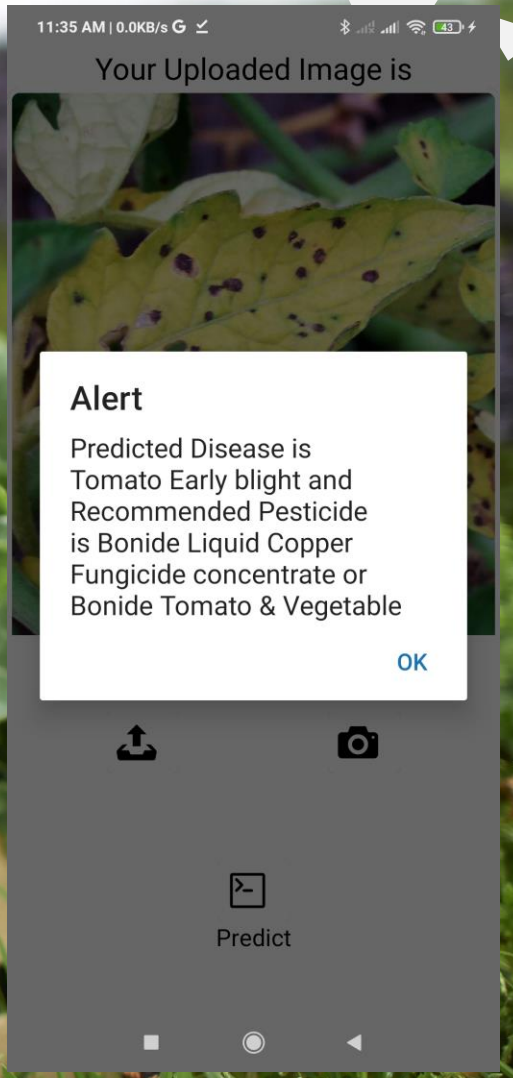
Evidence for Completion

```
label,pesticide,classes
0,Tebuconazole,Apple Apple scab
1,Myclobutanil,Apple Black rot
2,Myclobutanil (Immunox),Apple Cedar apple rust
3,,Apple healthy
4,,Blueberry healthy
5,Merivon fluxapyroxad+pyraclostrobin,Cherry (including sour) Powdery mildew
6,,Cherry (including sour) healthy
7,Patch Pro.□,Corn (maize) Cercospora leaf spot Gray leaf spot
8,Common Fungicides,Corn (maize) Common rust
9,"
Shamrock Al-sach With Logo Label Mark M45 Mancozeb ",Corn (maize) Northern Leaf Blight
10,,Corn (maize) healthy
11,"Mancozeb, and Ziram",Grape Black rot
12,"Mancozeb, and Ziram",Grape Esca (Black Measles)
13,"Bordeaux mixture (1.0%),Mancozeb (0.2%), Topsin-M (0.1%), Ziram (0.35%) or
Captan (0.2%)",Grape Leaf blight (Isariopsis Leaf Spot)
14,,Grape healthy
15,"No cure, Redo Process",Orange Haunglongbing (Citrus greening)
16,"□Copper, oxytetracycline",Peach Bacterial spot
17,,Peach healthy
18,Copper and Mancozeb sprays,Pepper   bell Bacterial spot
19,,Pepper   bell healthy
20,Avoid Overhead Irrigation and allow sufficient aeration,Potato Early blight
21,prophylactic spray of□mancozeb,Potato Late blight
22,,Potato healthy
23,,Raspberry healthy
24,,Soybean healthy
25,Apply sulfur or copper-based fungicides,Squash Powdery mildew
26,Copper-Count-N at 1.5 to 2 quart/100 gal water. Group M1 fungicide,Strawberry Leaf scorch
27,,Strawberry healthy
28,Bonide Liquid Copper Fungicide concentrate or Bonide Tomato & Vegetable,Tomato Bacterial spot
29,Bonide Liquid Copper Fungicide concentrate or Bonide Tomato & Vegetable,Tomato Early blight
30,Bonide Liquid Copper Fungicide concentrate or Bonide Tomato & Vegetable,Tomato Late blight
31,Bonide Liquid Copper Fungicide concentrate or Bonide Tomato & Vegetable,Tomato Leaf Mold
32,Bonide Liquid Copper Fungicide concentrate or Bonide Tomato & Vegetable,Tomato Septoria leaf spot
33,bifenthrin and permethrin,Tomato Spider mites Two-spotted spider mite
34,bifenthrin and permethrin,Tomato Target Spot
35,Imidacloprid,Tomato Tomato Yellow Leaf Curl Virus
36,"Viral Infection, Remove all infected Plants",Tomato Tomato mosaic virus
37,,Tomato healthy
```

Results and Interfaces

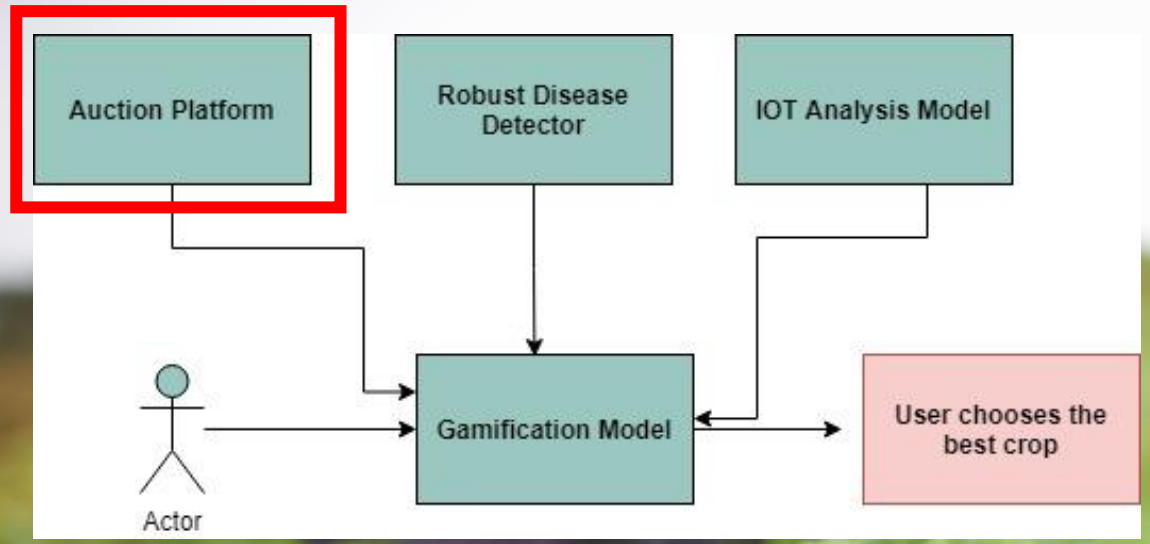






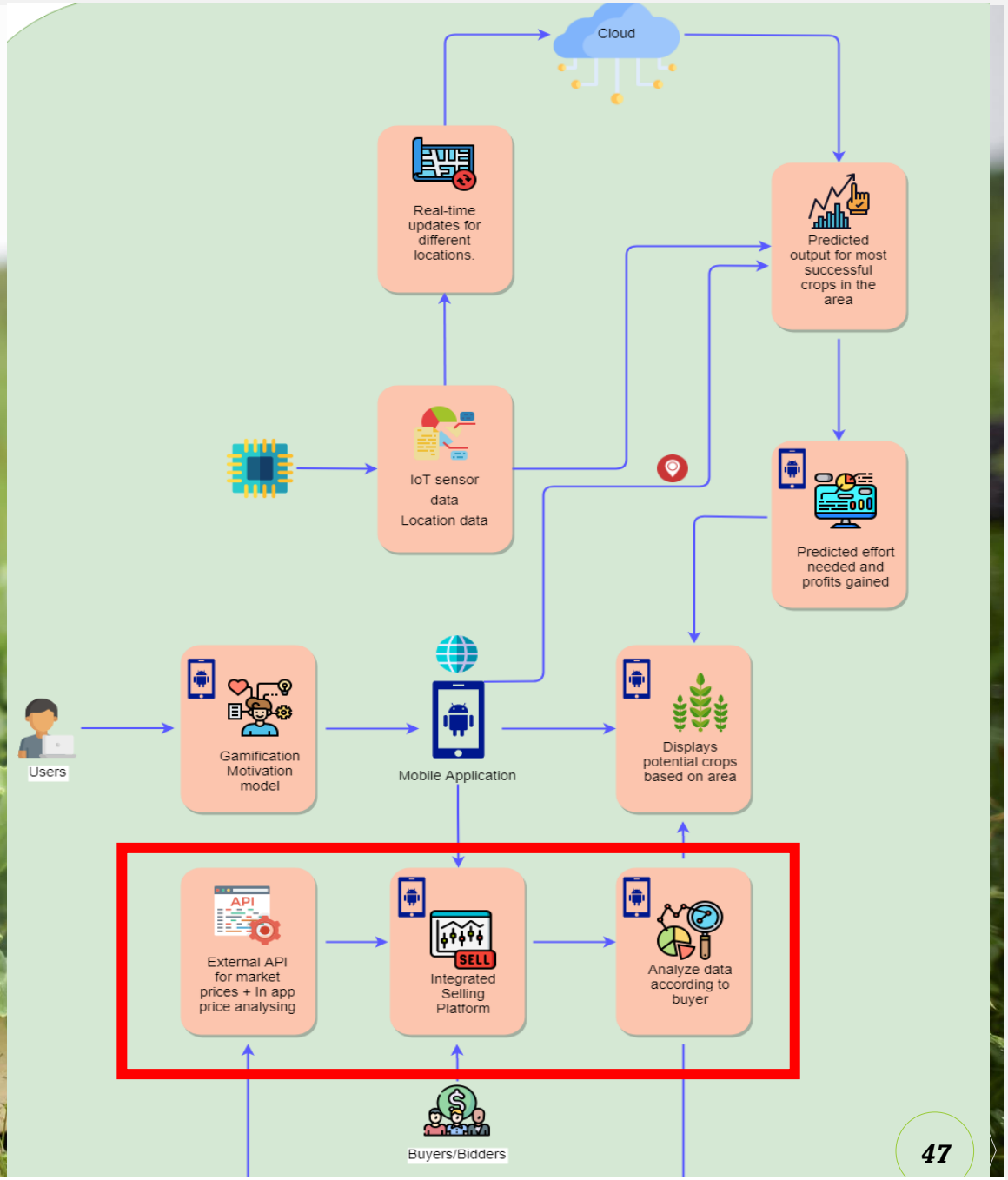
IT17023610 | A.G.J.L.P Rajapakse

Specialization: Software Engineering



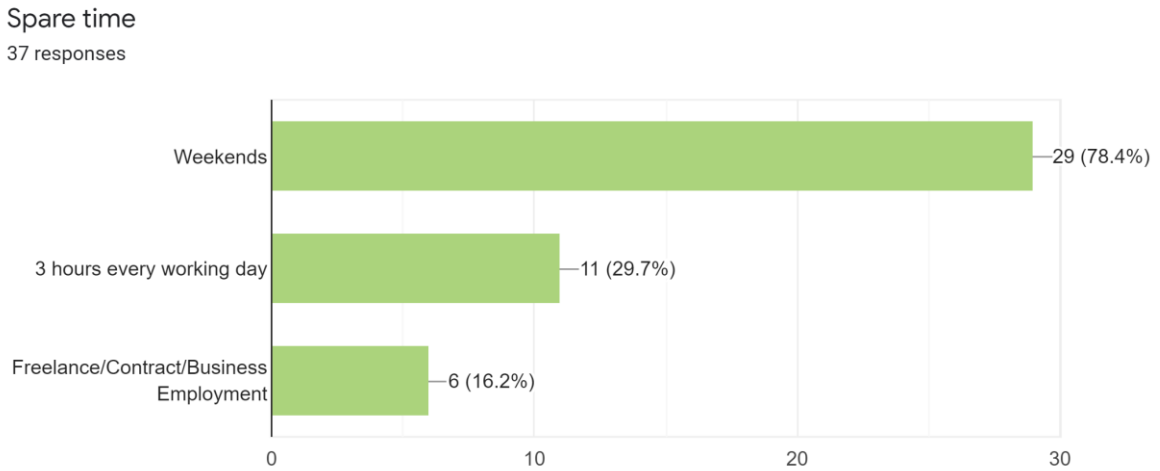
Integrated Selling Platform

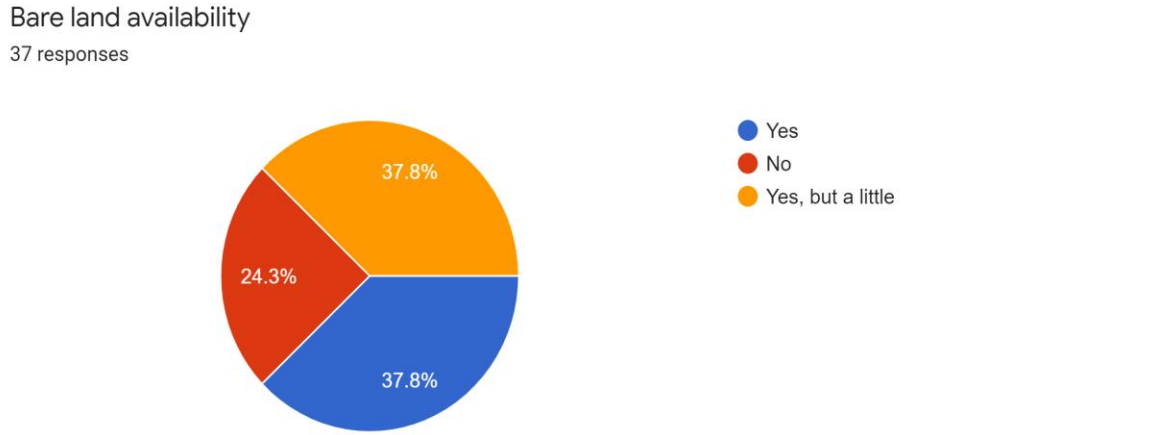
- Integrated Selling Platform
- Predicting trending crops based on analyzed data



Research Gap

- Creation of an integrated bidding platform.
- Creation of a data analysis model that uses machine learning in order to predict future trending crops.



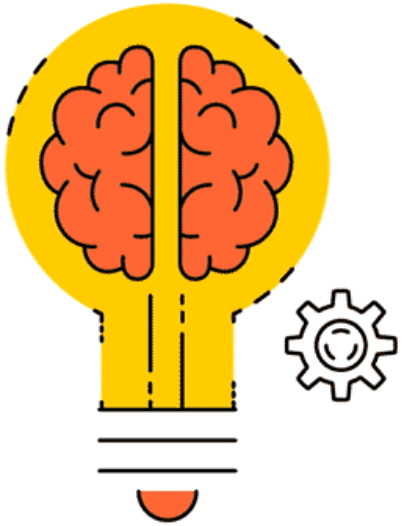


Research Question

- Is it possible to create a data analysis model that will help predict future trending crops in a given period of time?

Objectives

- Create an interactive bidding platform
- Using Machine Learning algorithms to predict future trending crops.



Sub objectives

- Designing and developing interactive interfaces for bidding platform.
- Training machine learning model using a clean dataset.
- Test the prediction accuracy of machine learning model.
- Create external API to retreive prices.

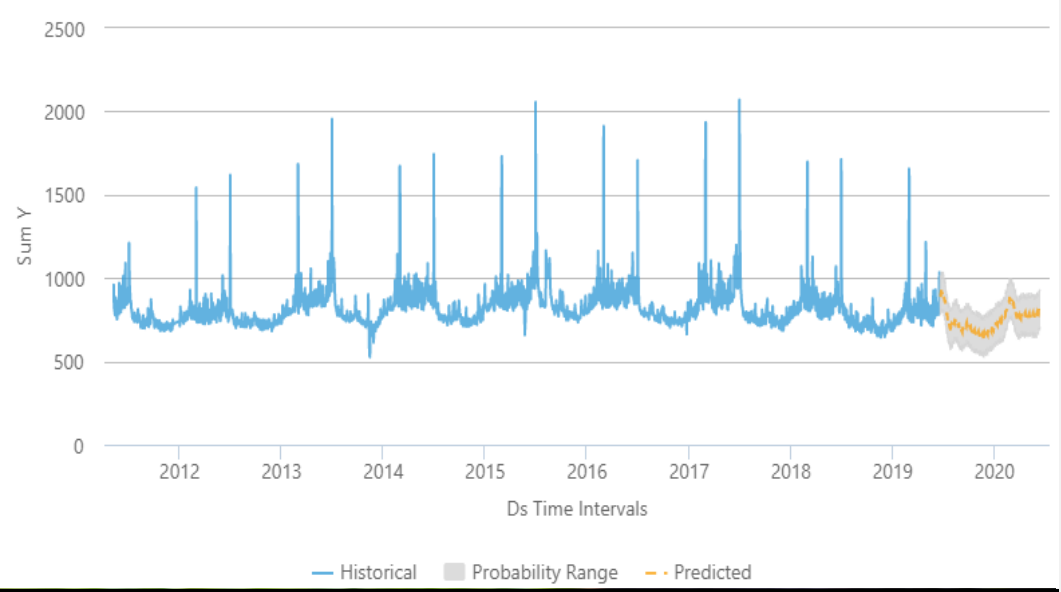
Tools and Technologies

Technologies

- React Native – Frontend
- Python Django Framework – Backend
- SQLite– Database

Tools

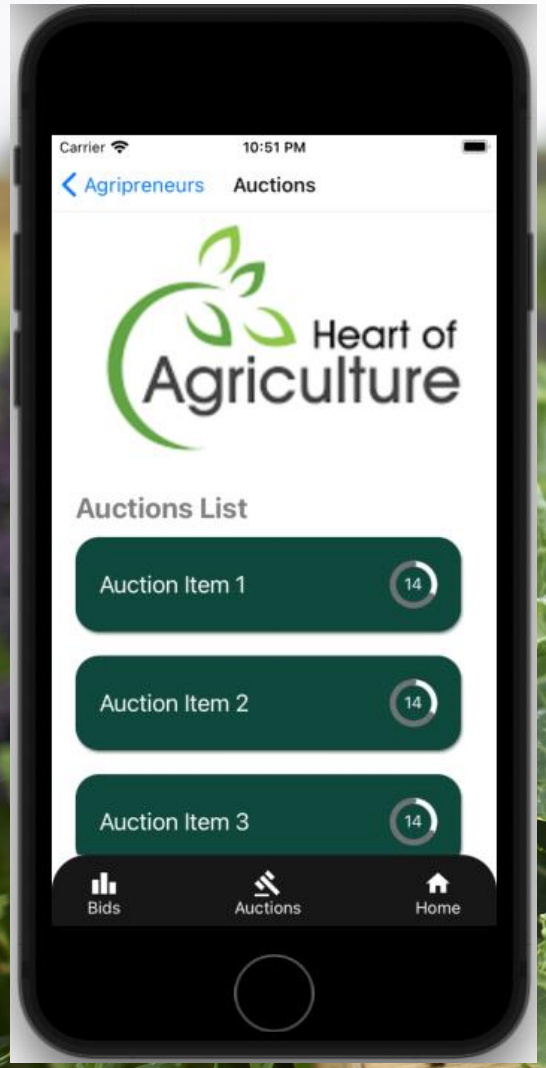
- Visual Studio Code

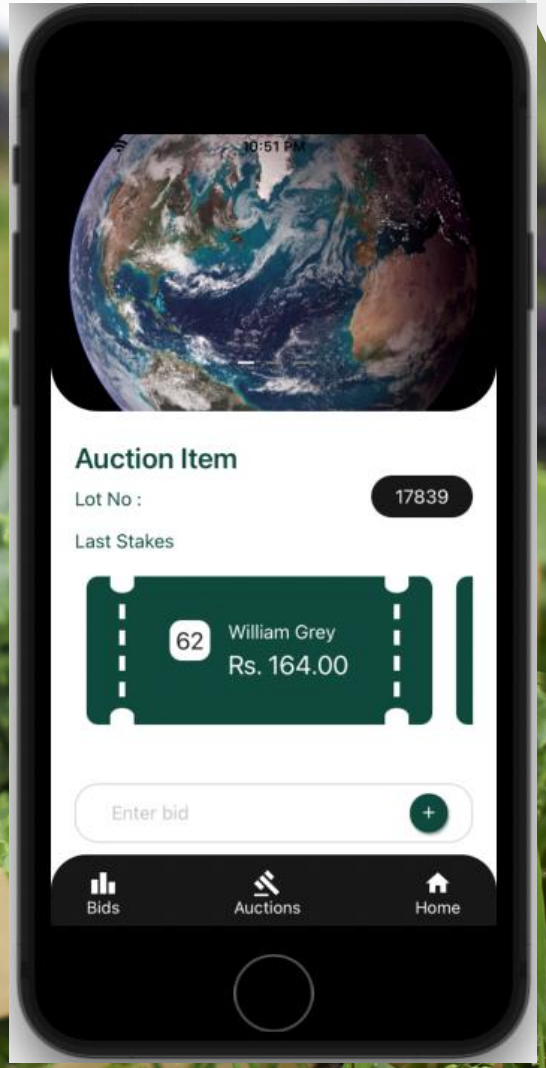


Algorithms

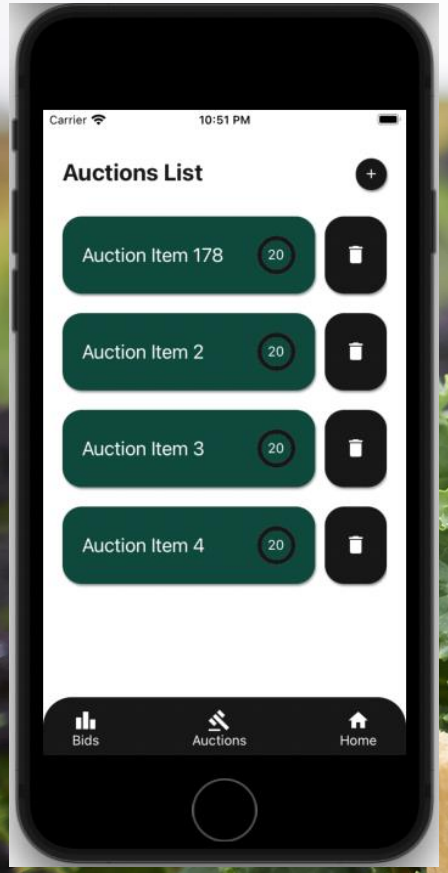
- Prophet algorithm (Time Series Model)

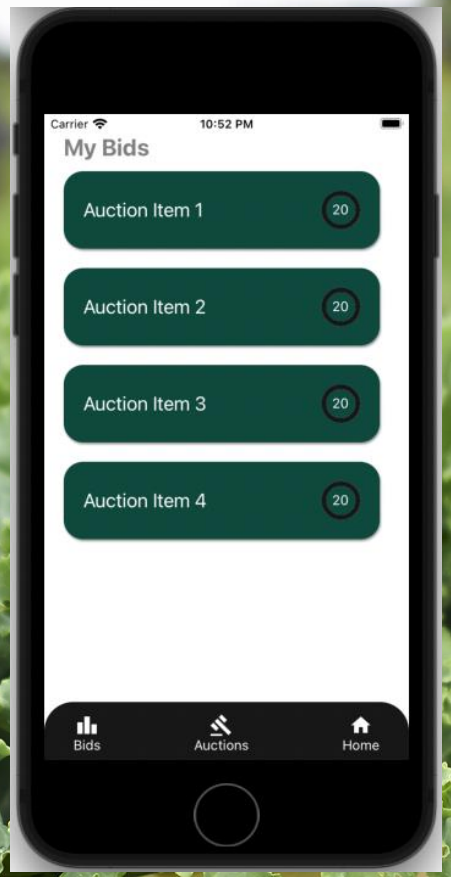
Evidence for Completion





Evidence for Completion

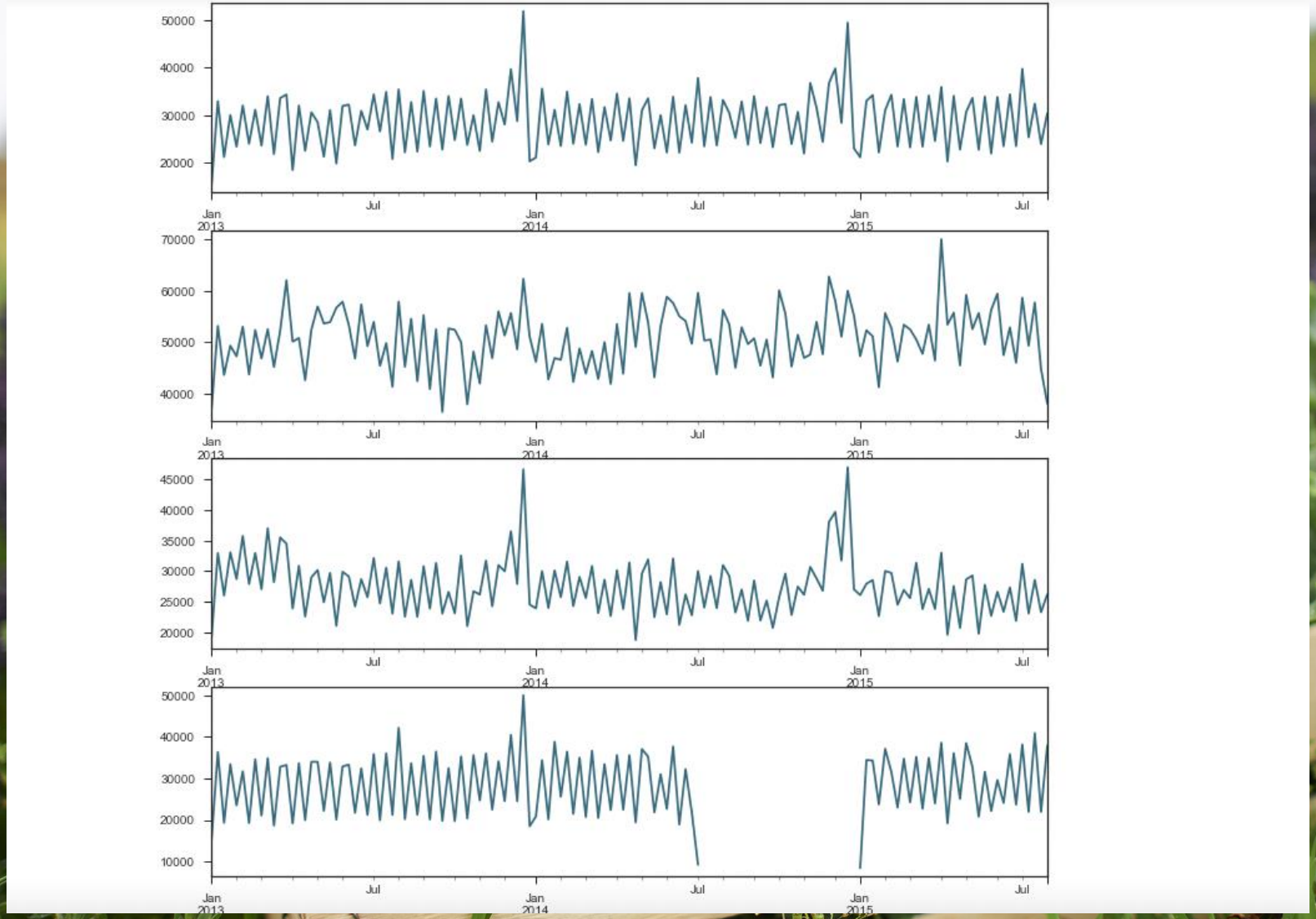




Evidence for Completion

	Store	DayOfWeek	Sales	Customers	Open	Promo	StateHoliday	SchoolHoliday
Date								
2015-07-31	1	5	5263	555	1	1	0	1
2015-07-31	2	5	6064	625	1	1	0	1
2015-07-31	3	5	8314	821	1	1	0	1
2015-07-31	4	5	13995	1498	1	1	0	1
2015-07-31	5	5	4822	559	1	1	0	1

Evidence for Completion



Requirements

System Requirements

- Laptop (Programming device)
- Internet
- GPS
- Smart Android Phone

Personal Requirements

- React Native, Python
- Clean dataset to train Machine Learning model
- Train Machine Learning algorithm correctly

Summary

- Combination of agriculture , entrepreneurship and technology
- Motivational Gamification Model to make users always involved.
- Image processing model to find diseased crops and predict appropriate fertilizer solutions.
- Machine learning model to predict best crops according to sales in the auction functionality.
- IoT sensors to get data real time according to the location/town.

References

- H. S. Rohitha Rosairo, David J. Potts, A study on entrepreneurial attitudes of upcountry vegetable farmers in Sri Lanka, Vol. 6 Issue: 1, Journal of Agribusiness in Developing and Emerging Economies, 2016.
- S. Mellon-Bedia,d, *, K. Descheemaekerb , B. Hundie-Kotua , S. Frimpongc , J.C.J. Groot ,Motivational factors influencing farming practices in northern Ghana , 2020
- Arun Kumar, Naveen Kumar and Vishal Vats , "EFFICIENT CROP YIELD PREDICTION USING MACHINE LEARNING ALGORITHMS", International Research Journal of Engineering and Technology (IRJET) Volume: 05 Issue: 06 | June-2018 e-ISSN: 2395-0056 |p-ISSN: 2395-0072, 2018.
- Leisa J. Armstrong and Sreedhar A. Nallan, "Agricultural Decision Support framework for visualization and prediction of western Australian crop production", - IEEE 978-9-3805-4421-2/16 – 201
- C. S. Herath, "The impact of motivation on farmers decision making on technology adoption with reference to Sri Lanka and the Czech Republic," *Knowl. Manag. Innov. A Bus. Compet. Edge Perspect. - Proc. 15th Int. Bus. Inf. Manag. Assoc. Conf. IBIMA 2010*, vol. 2, no. November 2010, pp. 790–801, 2010.
- N. Dobryagina, "Agricultural Entrepreneurship Motivation Policies: European Union Experience and Decision Theory Application," *Int. J. Rural Manag.*, vol. 15, no. 1, pp. 97–115, 2019, doi: 10.1177/0973005219834739.
- H. R. Rosairo, D. J. J. J. o. A. i. D. Potts, and E. Economies, "A study on entrepreneurial attitudes of upcountry vegetable farmers in Sri Lanka," 2016
- J. Kim and J.-W. Lee, "OpenIoT: An open service framework for the Internet of Things," in *2014 IEEE world forum on internet of things (WF-IoT)*, 2014, pp. 89-93: Ieee.
- C. Akshay *et al.*, "Wireless sensing and control for precision Greenhouse management," in *2012 Sixth International Conference on Sensing Technology (ICST)*, 2012, pp. 52-56: IEEE.


Thank
You
2021-090