CIS 890 Introduction to Food Informatics

Project Report

EXPLORING CROP YIELD PREDICTION USING MACHINE LEARNING TECHNIQUES

By

Revanth Babu Raavi

Table of Contents

1. Introduction

Predicting the yield of a crop has been significant these days especially for farmers and analysts. It even plays a major role in building financial economy of country. But the characteristics of climate and soil significantly influence agricultural outcomes, impacting yield production. However, conventional farming methods often rely on monoculture, excessive fertilizer use, and established practices. This project seeks to leverage Machine Learning models to forecast crop yield primarily focusing on wheat yield, providing farmers and agronomy analysts with valuable insights tailored to their needs.

1.1 Problem Statement:

This project tackles the main issue of crop yield unpredictability arising from fluctuating climatic and soil conditions. Farmers frequently encounter challenges in scheduling planting and harvesting activities due to a lack of clarity regarding the influence of climate and soil patterns on crop yields. This ambiguity often leads to less-than-ideal choices, resulting in insufficient production. The goal here is to reduce these uncertainties by offering a dependable predictive solution. This solution utilizes climate and soil data to forecast crop yields, empowering farmers with valuable insights to improve decision-making and refine their agricultural methodologies.

1.2 Objectives:

This project is built upon three fundamental goals aimed at improving decision-making in agriculture:

- **Data collection:** The initial step involves collecting data from past records. As the data has to be coming from specific domain people, unfortunately it didn't happened. To address this issue, the data was collected from kaggle. However, the newly acquired data lacked measuring units, prompting the need to make assumptions to compensate for this absence.
- **Machine Learning Techniques Implementation:** The second aim is to develop a machine learning model that can analyze this data effectively to achieve highly accurate crop yield predictions.
- **User Interface Development:** The ultimate aim is to deploy a user-friendly application that serves as a decision-making tool. Farmers and agronomy analysts can input climate and soil conditions and receive data-driven crop yield predictions. It enhances users with some insights for informed decisions on crop cultivation based on climatic and soil conditions.

1.3 Software Requirements:

Editor - Google Colab

IDE - VS Code / Pycharm

Framework - FLASK

Web server - XAMPP

GUI tool for managing and interacting MySQL databases - SQLYOG

2. Theoretical Background

2.1 Crop Agronomy and Nature Impact:

The prediction of wheat yield based on climate and soil data rests on the convergence of agricultural science, climatology. A variety of crops globally being served as food for a large population, is impacted by a complex blend of environmental, and management factors. Among these, soil element like planting location and climate element like rainfall affect various growth stages of crop, that includes sprouting, growth initiation, stem elongation, heading, blossoming, seed development, and ripening. The duration of these conditions impacts the yield during the evolving phases of crops.

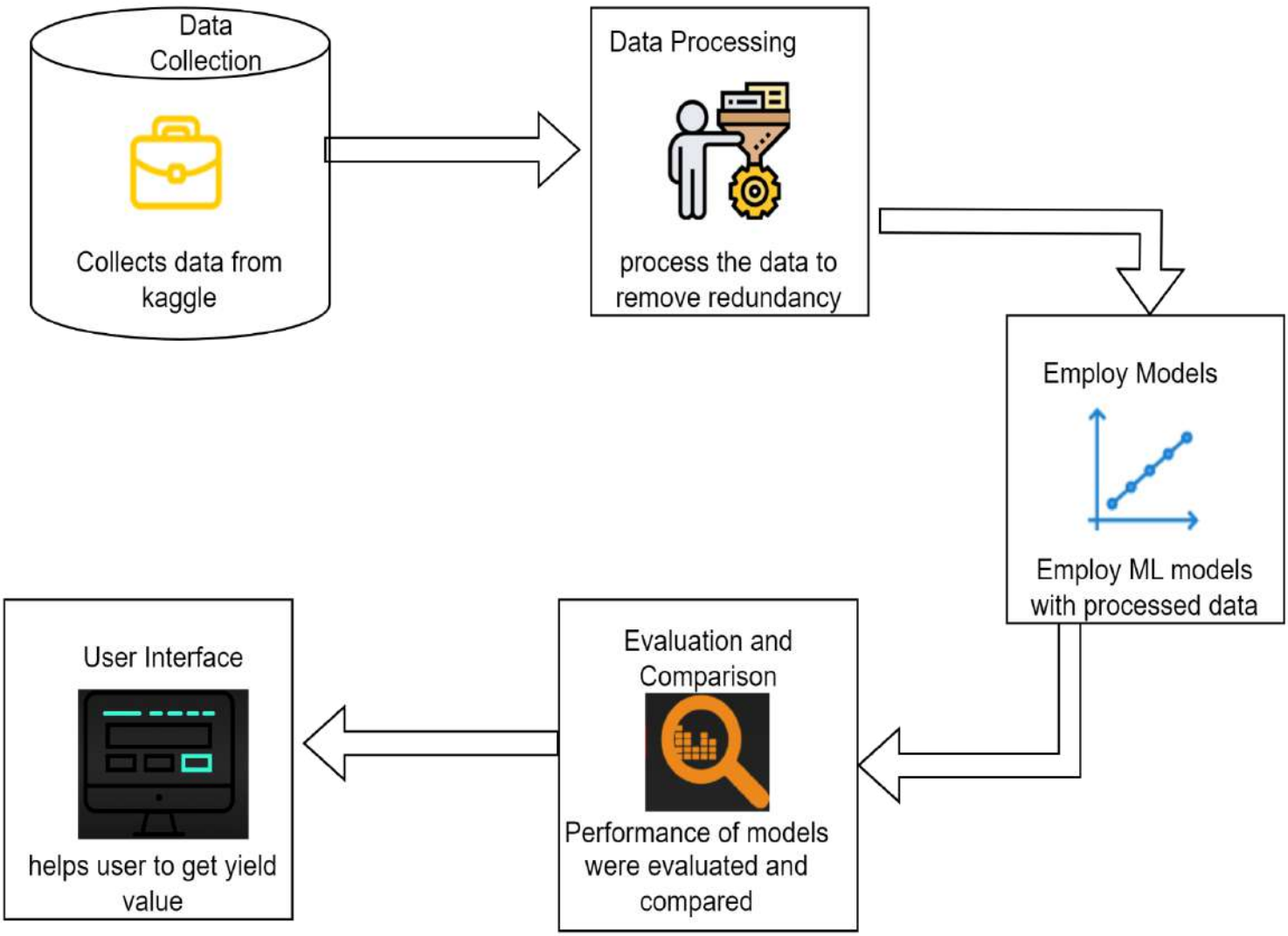
2.2 Machine Learning Techniques:

Machine learning models which are a subset of Artificial Intelligence leverage algorithms to learn from data patterns, enabling predictions without explicit programming. They improve by analyzing past data and making predictions on new, unseen information. They find applications in many areas and notably in agricultural yield prediction using historical climate and soil data.

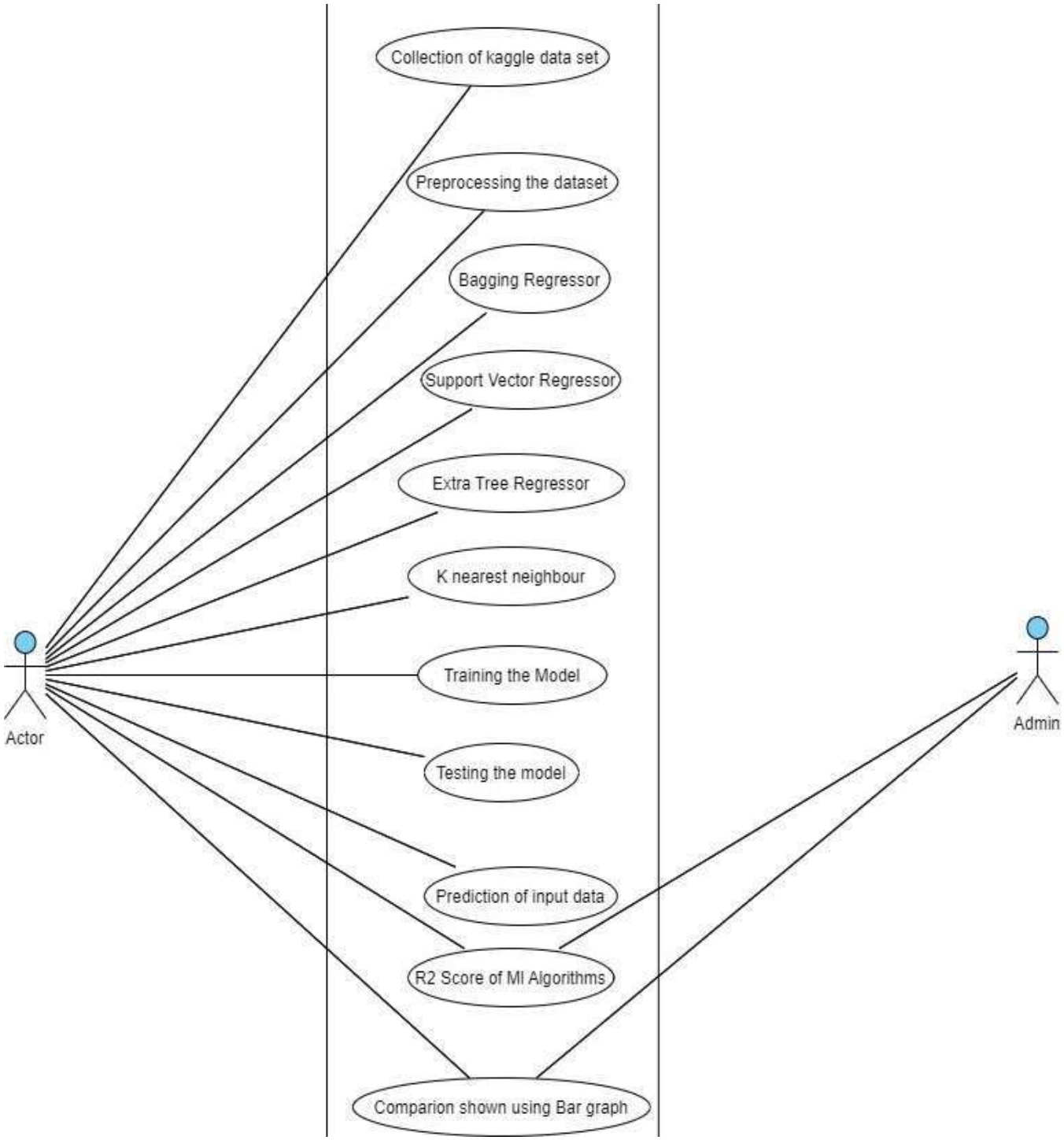
3. Project Workflow

Same like every project, as this also involves machine learning models, this project follows a sequence of steps that needs to be done which are explained down with architectural and use case diagram.

3.1 Architectural Diagram:



3.2 Use Case Diagram:



Actor is the one who had done all the process and developing models and admin is the public user who uses the interface to know the prediction the yield.

- **Data Collection:** This is the basic stage where data is collected from various sources and from my case it was collected from kaggle as there is no specific data from domain people.
- **Data preprocessing and cleaning:** Once after the data is collected, it needs to be processed and cleaned. This involves removing unnecessary data, handling missing data, converting strings to numerical values. This step is important because it helps to train the machine learning models accurately that significantly turns out to make better and accurate predictions.
- **Employ Models:** Here, machine learning models are applied to processed data. A wide variety regressor models like Bagging, Support Vector, K-Nearest Neignbor and Extra Tree Regressor are used in this project in order to compare and finds out the best one among all.
- **Model evaluation and Comparison:** This step is important for validation. Machine learning models have to be evaluated to determine the performance using metrics like r2_score and time. So that, the best model is preserved to predict the yield for tested dataset and could compare with the actual yield.
- **User Interface:** This is the last step and this includes creating a web application that is user-friendly that allows them to predict the yield of the crop. It enables the users to input all the data and inturn could get accurate predictions which inturn helps for farmers or analysts to make decisions based on models output.

4. Data collection

As i said, the data is collected from kaggle and there are no units and it needs to be assumed to remove the redundancies.

	State	Year	Season	Crop	Area	Productio	Rainfall
0	Andaman	2000	Kharif	Arecanut	1254	2000	2763.2
1	Andaman	2000	Kharif	OtherKhar	2	1	2763.2
2	Andaman	2000	Kharif	Rice	102	321	2763.2
3	Andaman	2000	WholeYea	Banana	176	641	2763.2
4	Andaman	2000	WholeYea	Cashewnu	720	165	2763.2
5	Andaman	2000	WholeYea	Coconut	18168	65100000	2763.2
6	Andaman	2000	WholeYea	Dryginger	36	100	2763.2
7	Andaman	2000	WholeYea	Sugarcane	1	2	2763.2

5. Data Processing

The data is processed by encoding the null values, converting strings to numerical values and splitting the data into 80% training and 20% testing and for training and testing there are two variables X and Y in each of them, the X is the feature that we use to predict the Y target, and the same for the testing also.

	State	Year	Season	Crop	Area	Production	Rainfall
0	0	0	2000	1	0	1254.0	2000.0	2763.2
1	1	0	2000	1	50	2.0	1.0	2763.2
2	2	0	2000	1	63	102.0	321.0	2763.2
3	3	0	2000	4	3	176.0	641.0	2763.2
4	4	0	2000	4	13	720.0	165.0	2763.2
...	...	...	...	...	...	...	...	...
74970	236473	11	2014	4	47	26.0	172.0	1287.4
74971	236474	11	2014	4	59	5346.0	4630.0	1287.4
74972	236475	11	2014	4	63	9919.0	16499.0	1287.4
74973	236476	11	2014	4	72	3.0	3.0	1287.4
74974	236477	11	2014	4	78	35417.0	49689.0	1287.4

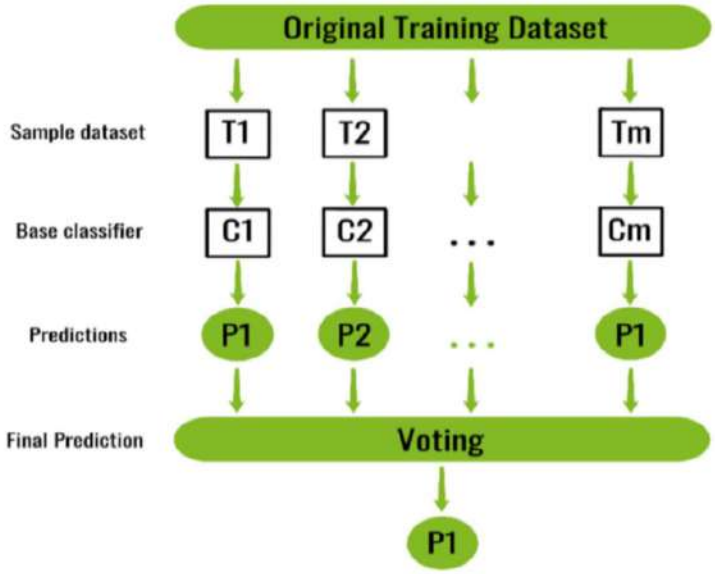
```
[ ] from sklearn.model_selection import train_test_split
    #from sklearn.cross_validation import train_test_split

    x_train, x_test, y_train, y_test = train_test_split(x.values, y.values, test_size=0.20, random_state=0)
```

6. Employ Models

In this project, regression models are chosen because predicting the yield of a crop involves calculating continuous numerical data as output based on input data. So, these models helps to find relationship between dependent variables(X) and independent variables(Y).

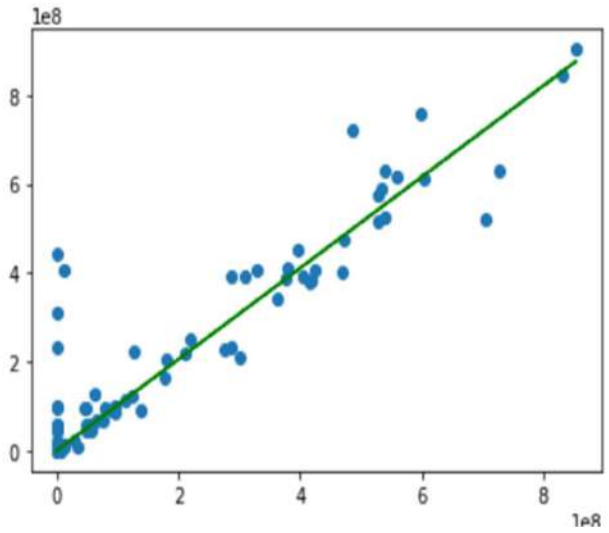
6.1 Bagging Regressor:

Bagging Regressor works by training multiple regression models on bootstrapped subsets of the data, combining their predictions through averaging to reduce variance and overfitting, resulting in a more stable and generalized model.



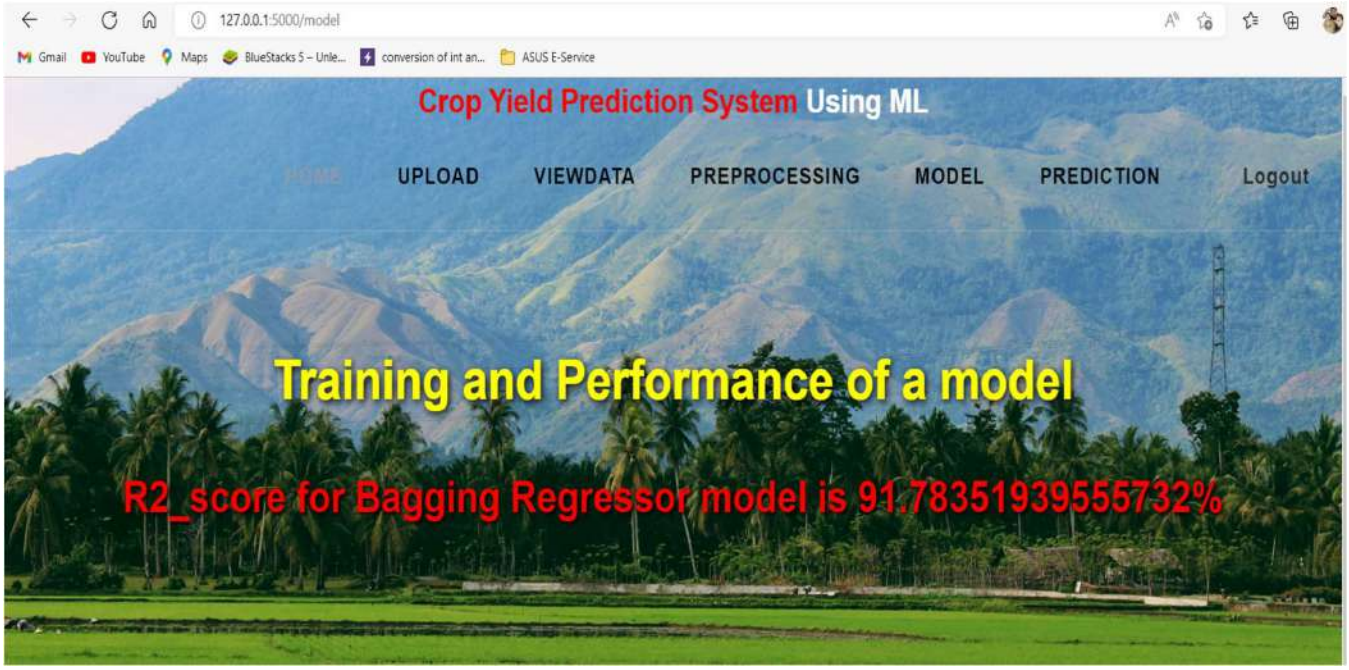
When i applied slope intercept formula for bagging regressor, the r2_ score graph came like below. It clearly shows that the distance between actual points and the regression line is very short and the model can predict near to the actual value related to crop yield.

```
slope, intercept, r_value, p_value, std_err = stats.linregress(y_test,y_pred)
def linefitline(b):
    return intercept + slope * b
line1 = linefitline(y_test)

#plot line
plt.scatter(y_test,y_pred)
plt.plot(y_test,line1, c = 'g')
plt.show()
```



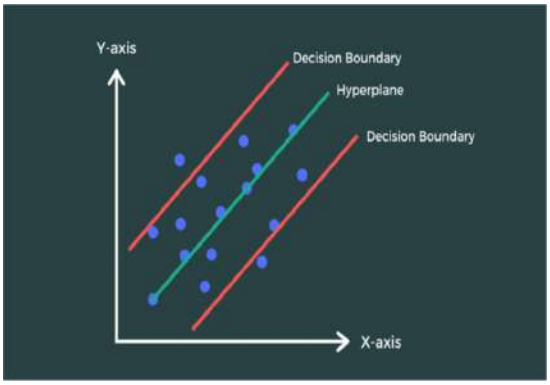
R2_score graph for bagging regressor model

After training the model, the R2_score for bagging regressor is around 91.7 %

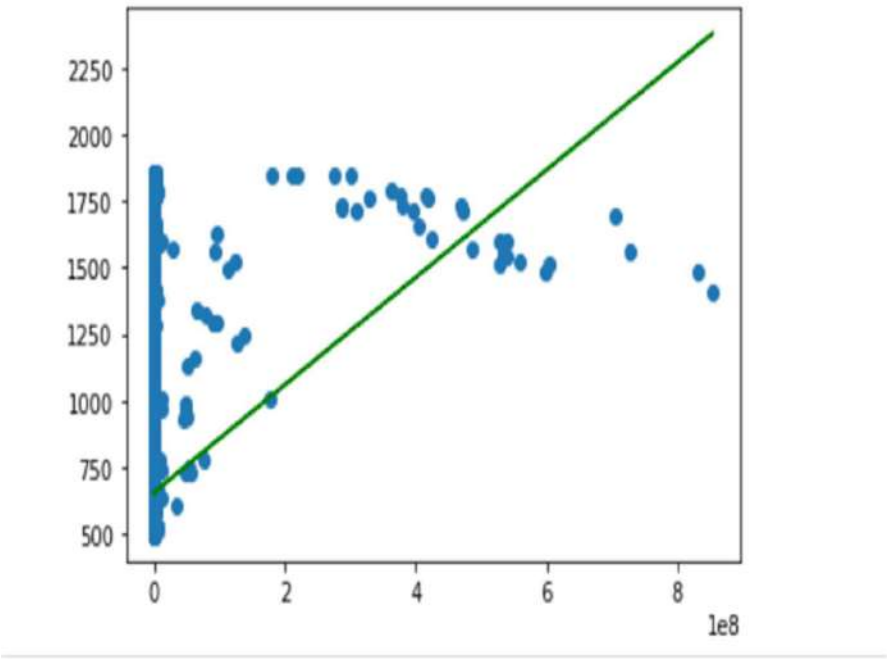


6.2 Support Vector Regressor:

- Support Vector Regression (SVR) works by finding a hyperplane that best fits the data within a specified margin of tolerance (epsilon), aiming to minimize errors while maximizing the margin. Using kernel functions, SVR handles nonlinear relationships, focusing on support vectors within an epsilon-insensitive tube to build a robust regression model that generalizes well to new data.
- SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector.

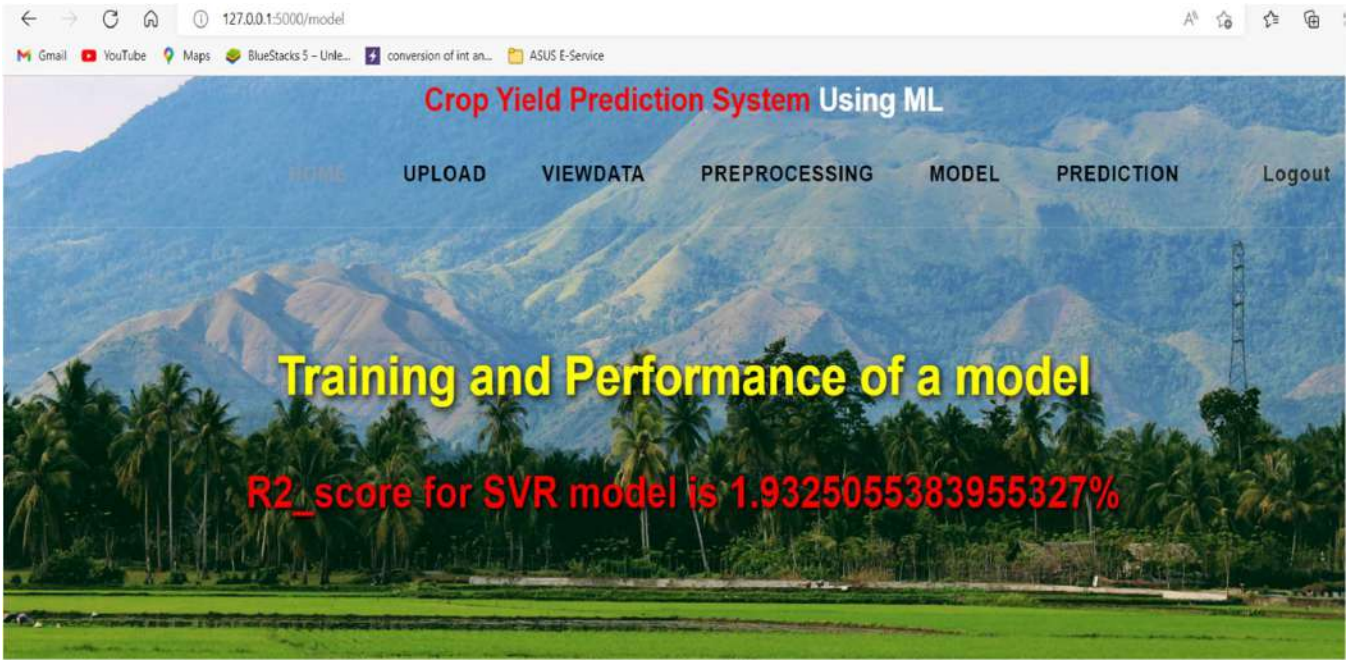


Similarly, for SVR the graph is represented as follows. It consists of very high variance. Because of high variance, SVR Model cannot give better predictions of crop yield and it takes more time for training.



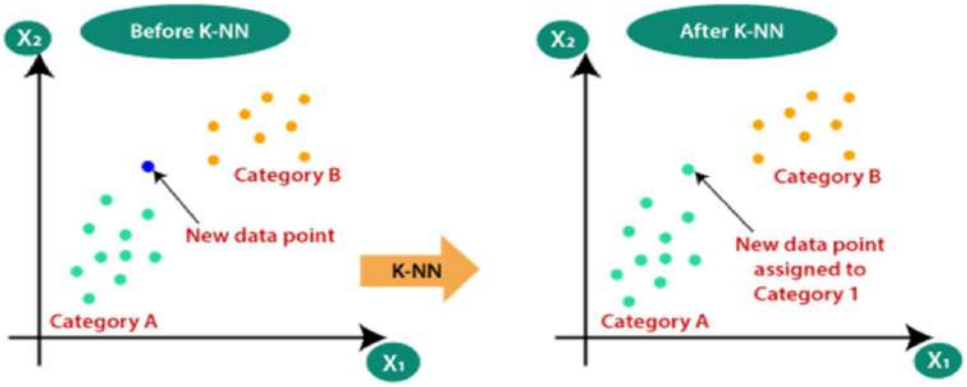
R2_score graph for SVR model

Similarly for SVR, the r2_score is 1.93%

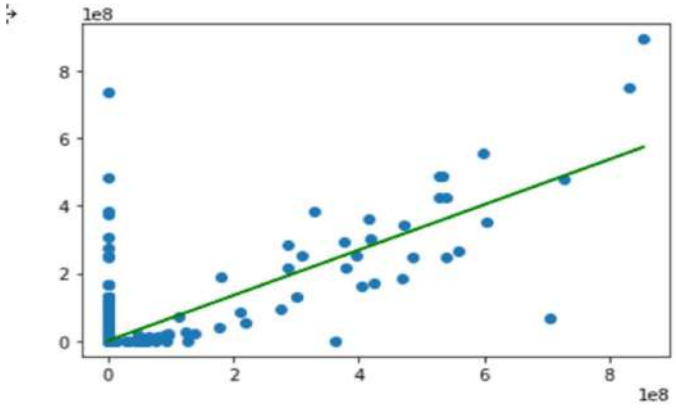


6.3 K-Nearest Neighbor Regressor:

- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity.



For, K-NN the graph is below and it shows that the distance between the regression line and the actual points is not that much short. So, the model able to predict but not near to the actual value related to crop yield.

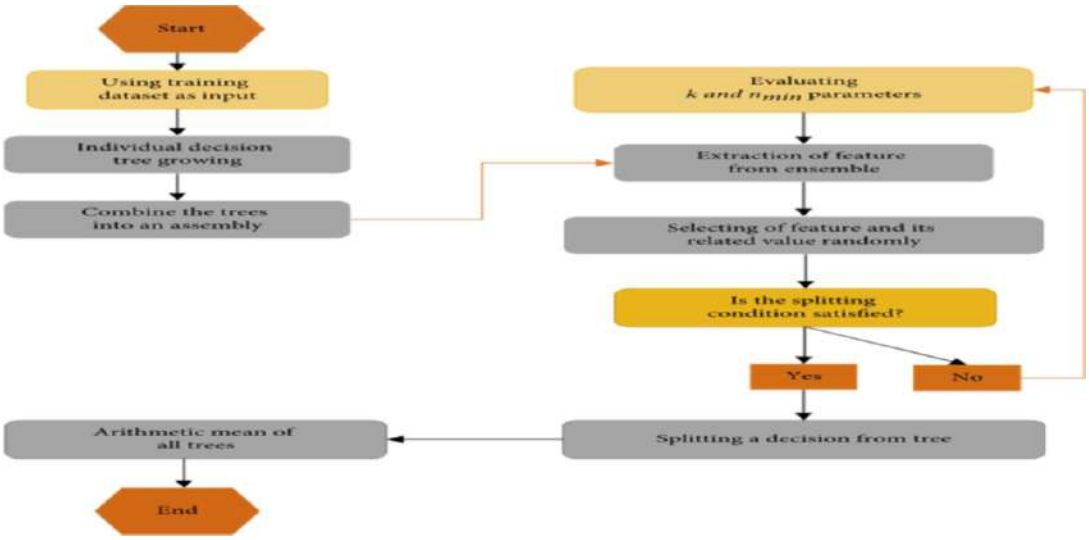


R2_score graph for K-NN model

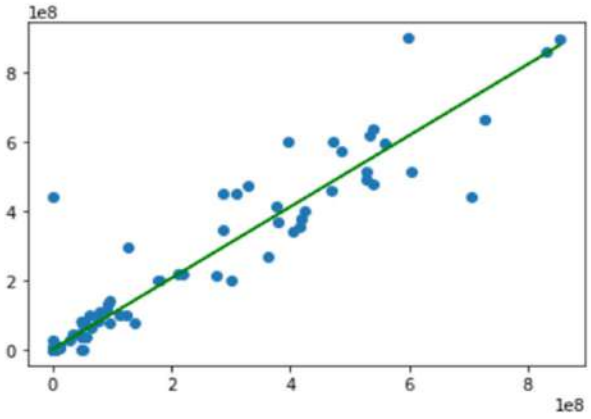
For K-NN model, the r2_score is around 60%.

6.4 Extra Tree Regressor:

The Extra Trees Regressor constructs an ensemble of decision trees with increased randomness by selecting random features and thresholds at each node split. By aggregating predictions from these diverse trees, it aims to reduce overfitting while maintaining predictive accuracy, forming a robust regression model suitable for handling complex datasets.



For extra-tree regressor, it contains low variance and the distance between the regression line and the actual points are shorter than all other models. So, the model can predict almost accurate values related to crop yield.



R2_score graph for Extra-tree regressor model

Similarly, for extra-tree regressor the r2_score is around 94%.

7. Model Evaluation and Comparison

7.1 Evaluation metrics:

R2_score: Have taken the below formulas as reference for predicting r2_score and time

Take y_pred values.

```
y_mean = np.mean(y_actual)

ss_tot = np.sum((y_actual - y_mean) ** 2)

ss_res = np.sum((y_actual - y_pred) ** 2)

r_squared = 1 - (ss_res / ss_tot)
```

where,

y_pred are predicted values

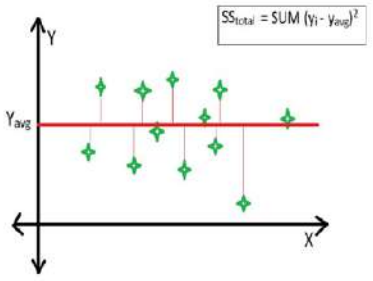


Figure 3.5: Formula of SS_{tot}

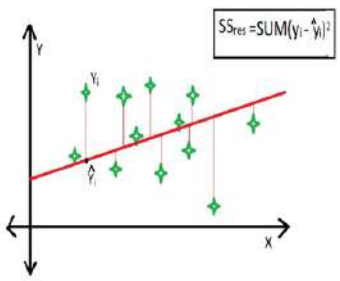


Figure 3.6: Formula of SS_{res}

y_actual are y_test values

y_mean is mean of y_test

SSres is the sum of the squares of the residual mistakes.

SStot is the entire sum of the errors.

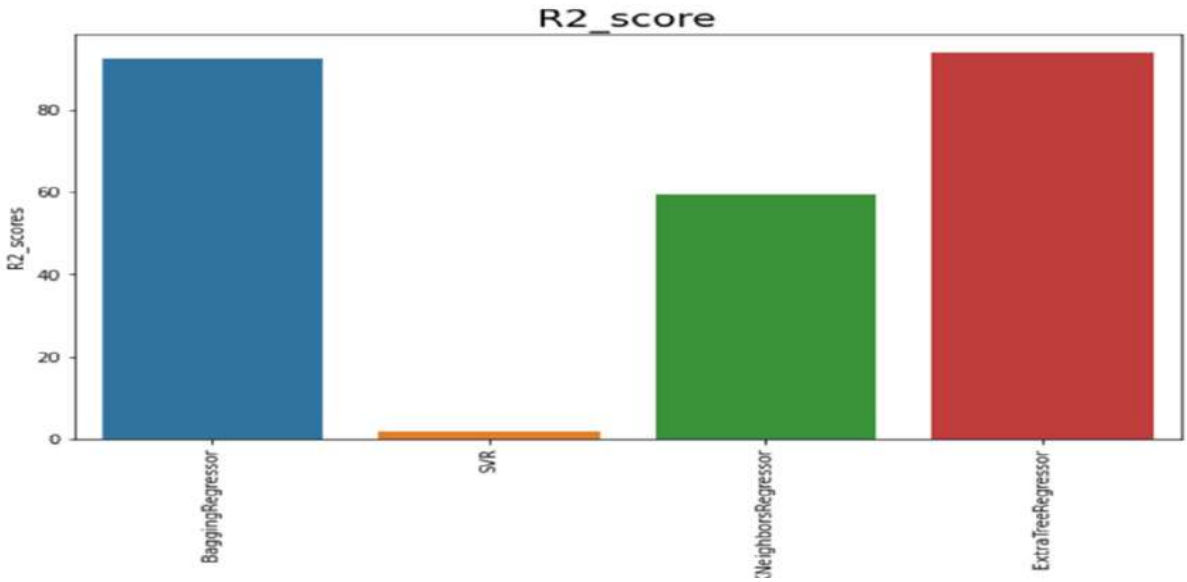
Time:

Calculated the time taken by each model using below formula.

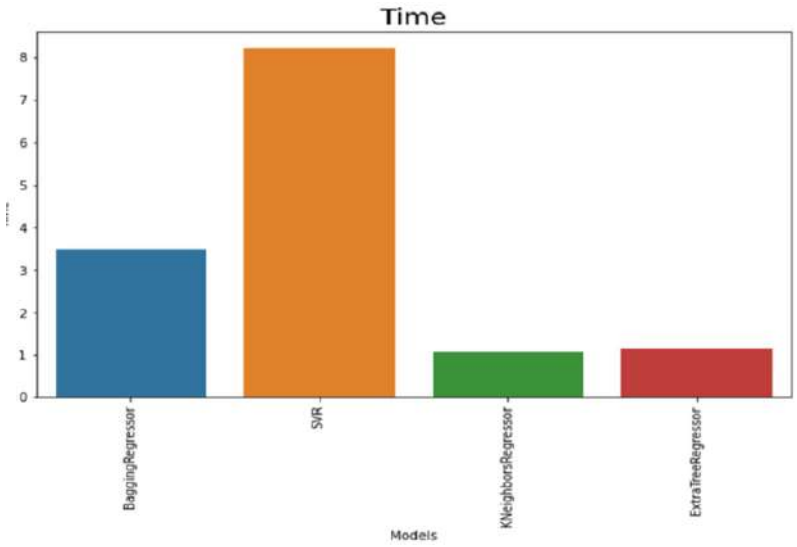
```
bg = BaggingRegressor()


# Code to Measure time taken by program to execute.
import time

# store starting time
begin1 = time.time()
bg.fit(x_train,y_train)
time.sleep(1)
# store end time
end1 = time.time()

# total time taken
print(f"Total runtime of the program is {end1 - begin1}")


Total runtime of the program is 3.578505754470825
```

7.2 Comparison:

From the below graph, it is observed that extra tree regressor is performing best when compared to remaining models. All though the r2_score fluctuates for bagging and extra-tree regressor sometimes, which is common for machine learning models sometimes, the average value showed better accuracy for extra-tree regressor.



From the below graph, it represents time taken by each model is 3.59 sec for the BR Model, 8.13 sec for the SVR Model, 1.07 for the KNN Model and 1.13 for the ETR Model.



8. User Interface

After the models are evaluated and compared, the best model is preserved to predict the yield. I developed a user friendly front-end web application using Flask framework and HTML. The users can input the data and can retrieve the prediction yield value. I gave the whole process like uploading data, processing it, comparing models and testing yield value to access in the

front-end part also so that user can have a clear understanding of the structure. The code has to be saved in project directory to have easy access in front end part.

I used Visual Studio Code for scripting in python. I also used XAMPP and SQL Yog server which are responsible for starting and connecting the databases. For running web application to run, one needs to Import the code into Visual Studio Code and run the program. Before running it, make sure all the libraries are installed for smooth running of application. The libraries are

Pip install flask

Pip install pandas

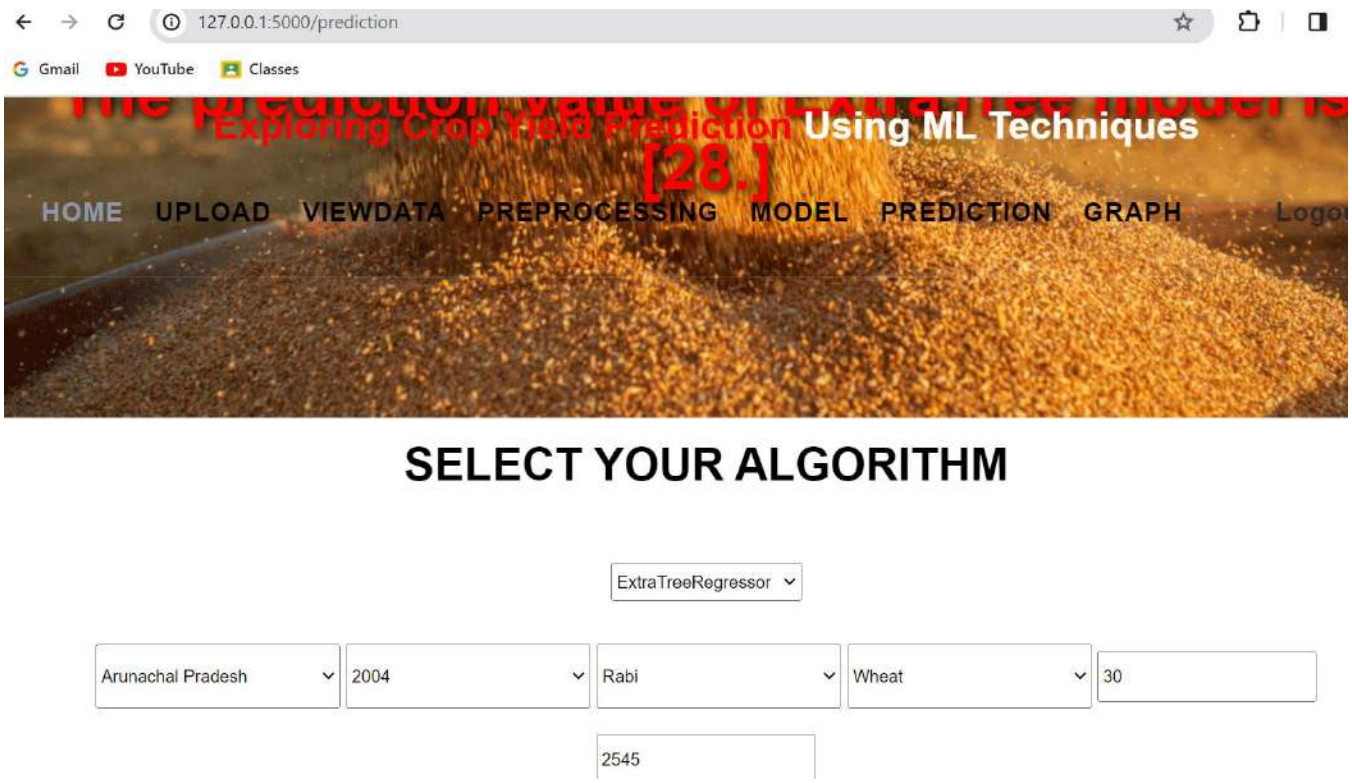
Pip install mysql-connector

Pip install scikit-learn

Pip install matplotlib

To launch it, use **python [application_name].py** command and execute it. It opens with the home page where one needs to register and sign in and then give the input variables to get the yield value.

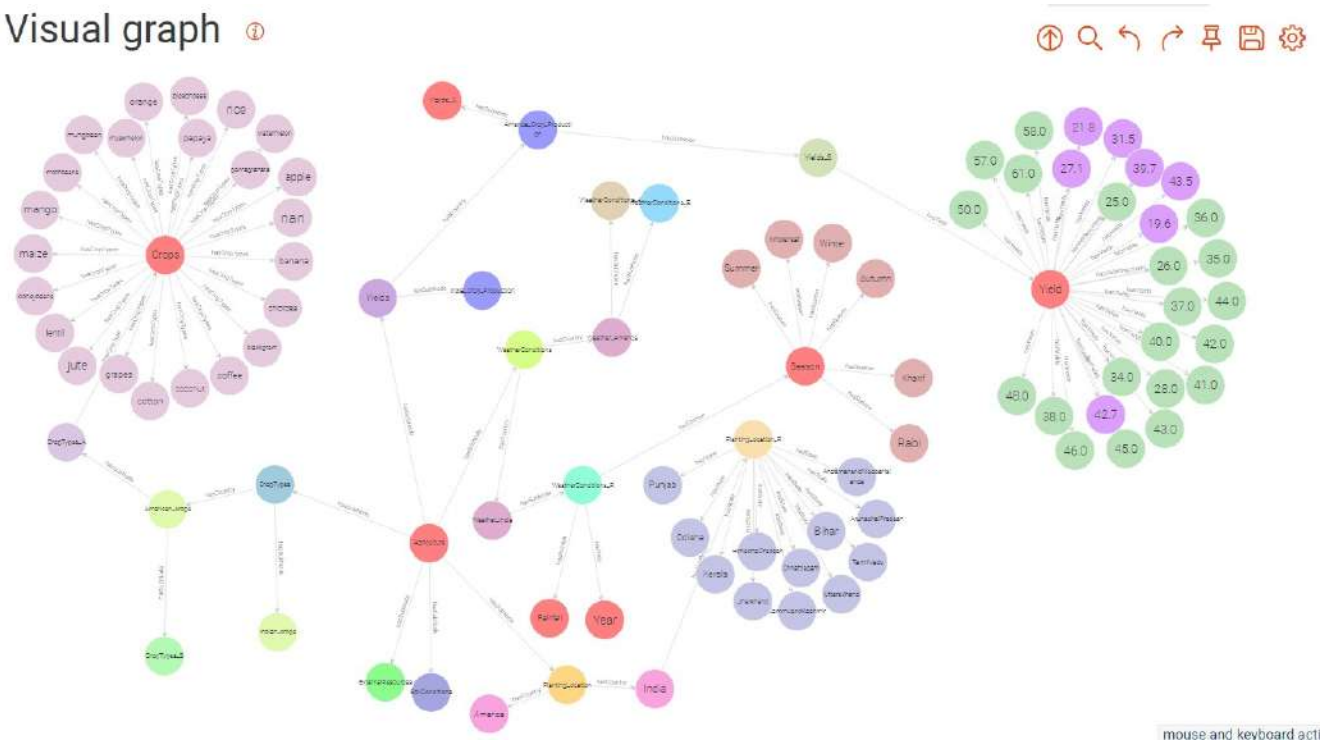
Then upon completing each step it redirects to prediction page where it predicts the yield of a crop. Below is the prediction for a random data from dataset. The predicted value is 28, which is near to actual value.



With this, one can select and check the yield of crops and gets an idea of how this model is performing.

9. Graph DB Instance

Alongside this machine learning prediction, i have also implemented the knowledge graphs in Graph DataBase by taking my data as well as my classmates data. Executed a python code to convert csv files into RDF data, imported the RDF file in Graph DB and checked how the graph is taking the relations. Below figures shows the relationship between data.



10. Conclusion:

- This model specializes in the estimation and determining of crop yield with assist of ML techniques. Though some fluctuations occur, after taking average value the final Accuracy is 93.54% for the ETR Model, 91.7% for the BR Model, 59.5% for the KNN Model and 1.93% for the SVR Model. Time taken to train is 3.5 sec for the BR Model, 8.13 sec for the SVR Model, 1.07 for the KNN Model and 1.13 for the ETR Model.
- Extra Tree Regressor model achieves the highest R2 score among all other ML models and gives an accurate prediction of crop yield within less time. This system is scalable as it can be used to test on different crops. This application is developed to seek out the information about crop yield which able to be deployed to make green and beneficial harvesting.
- Implemented Graph Database to observe the relations between data.

11. Future work

In future, I will try to add more features to dataset and check how Machine Learning models are reacting based on accuracy for predicting the yield. I will also apply knowledge graphs and test how well regression models are performing.

12. References

https://www.kaggle.com/datasets/chinmaynagesh/crop-yield-per-state-and-rainfall-data-of-india?resource=download

https://r.search.yahoo.com/_ylt=Awrx_qzM_ndl7cMHny.7HAx.;_ylu=Y29sbwNzZzMEcG9zAzEEdnRpZAMEc2VjA3Ny/RV=2/RE=1702391628/RO=10/RU=https%3a%2f%2fsourceforge.net%2fprojects%2fxampp%2ffiles%2fXAMPP%2520Windows%2f8.2.4%2fxampp-windows-x64-8.2.4-0-VS16-installer.exe%2fdownload/RK=2/RS=g6xxMMC.yCQDHJJPijyvzrAwKjk- LINK TO

https://s3.amazonaws.com/SQLyog_Community/SQLyog+13.2.0/SQLyog-13.2.0-0.x64Community.exe