Crop Yield prediction using Machine Learning Models

Nihar Ranjan Nath
E-mail- 19mtcs014.niharranjannath@giet.edu
M.Tech Scholar(GIET University)

MS Jemarani Jaypuria
E-mail – jemarani253@giet.edu
Asst. Professor(GIET University)

Abstract - India's dependence on wheat grows day by day with increasing population. To forecast regional and worldwide food security and commodities markets, accurate yield forecasting is required. Machine-learning algorithms can properly estimate wheat output for the country two months before the crop matures, according to a recent study published in Agricultural and Forest Meteorology.
When the predictive power of a conventional statistical technique is compared to the predictive power of machine-learning algorithms, the machine-learning algorithms consistently beat the old method. Scientists have produced reasonably accurate agricultural production predictions using climatic data such as temperature, weather, and satellite data, or both, in recent years. The researchers were able to estimate wheat production with around 75% accuracy two months before the conclusion of the growing season using climate and satellite data. We use machine learning to develop models that can forecast the likelihood of losses and investigate the most important variables.
In this project we would like to predict the winter weather yield in India by a given weather features and raw crop physiological features (such as NDVI, day in season).

Key Words: Crop Yield, Machine learning, Linear regression, SVM, DecisionTree, Scikit-Learn, Jupyter Notebook

1.INTRODUCTION

Accurately and timely estimating winter wheat yield in India is highly required as its considered as one of the three top grains. Researchers have been working on improving agricultural production prediction using a variety of techniques, including empirical statistical models and process-oriented crop growth models, over the past few years. These models have a limited spatial generalisation capacity, which makes them challenging to apply to wider regions.

Machine learning has proven to be effective in data mining and agricultural studies, such as crop classification and yield prediction. Crop yield prediction on a wider scale often necessitates a huge quantity of data and complicated data processing, implying expensive acquisition and processing expenses. Machine learning approaches can help yield prediction models improve dramatically. Many studies included factors that were dependent on the whole growing season, which meant that the ultimate yield couldn't be predicted until the harvesting day. Finding the optimal moment to record the features of crop growth can help to enhance yield performance.

Here we try to predict the winter weather yield by a set of data consisting of
=> Location and time(county name, state, latitude, and longitude), raw
=> Weather features (temperature, precipitation, wind speed, and pressure) => Raw crop physiological features such as NDVI, day in season, and yield.

Here we user different machine learning algorithms by scikit learn library includes linear regression, SVM, and Decision Tree to train the model for predicting yield.

2. Literature Survey

2.1 Shastry et al. (2017) fitted various regression models to forecast the crop yield in India by using data mining techniques. Maize, wheat and cotton crop yield are selected to study using time series data, soil and weather parameters. The regression techniques can be fitted well for yield forecasting for the crop yielddata.
The outcomes demonstrated that the proposed regression growth model is a suitable method for forecasting yield production of wheat, maize and cotton.

2.2 Panwar (2014), studied the forecasting of growth rates of wheat yield of Uttar Pradesh through nonlinear growth models.The yield data collected for the period of 1970 – 2010 of wheat crop in Uttar Pradesh.
In the studies of the various goodness of fit, results indicated that logistic model fitted well followed by Gompertz and Monomolecular growth model for forecasting of wheat production in UP.

3. REQUIREMENTS

3.1 Algorithms:

- ✓ Regression: simple, avoid overfitting.
- ✓ SVM (polynomial): catch interaction weather features
- ✓ DecisionTree: When have a lot of "leaves", in our case the yield values

3.2 Datasets:

- ✓ All of the datasets utilised in the study came from the Indian government's easily available databases.
- ✓ Only a small number of key parameters with the greatest influence on agricultural production were chosen for the current study from the large initial dataset.

3.3 Tools Needed

- **Scikit-Learn**
  It is a machine learning software that is open-source. Because it is applied for many purposes, it is a unified platform. Regression, clustering, classification, dimensionality reduction, and preprocessing are all aided by it. NumPy, Matplotlib, and SciPy are the three primary Python libraries that Scikit-Learn is built on top of. Additionally, it will assist you in both testing and training your models.

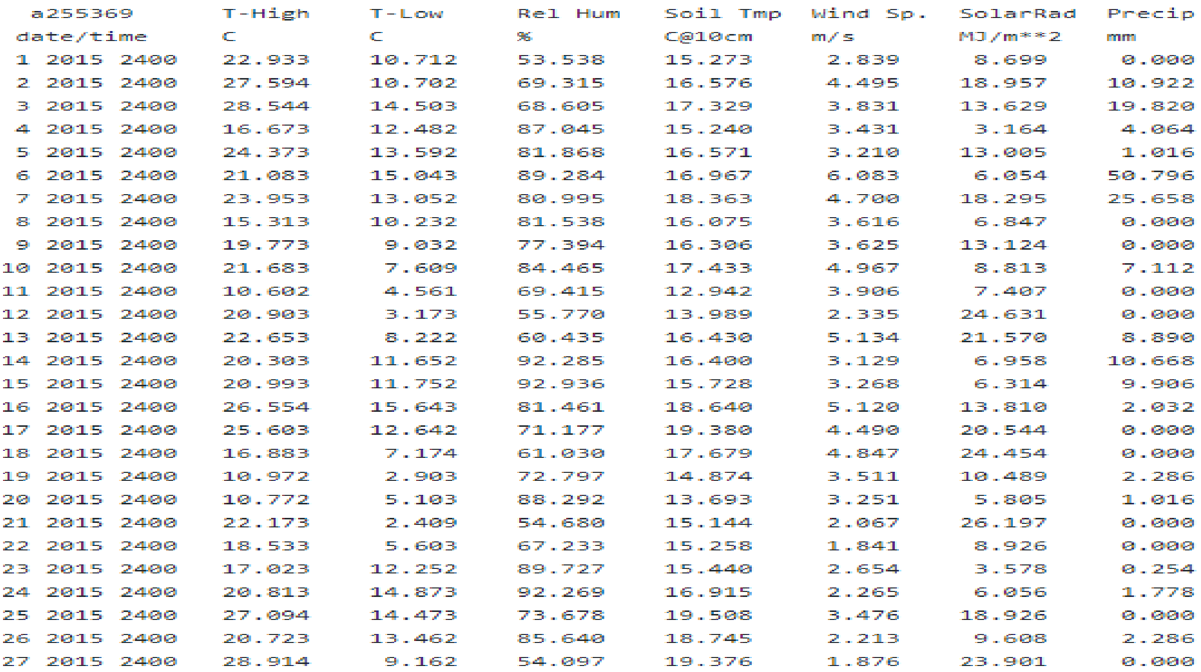
- **Jupyter Notebook**
  One of the most commonly used machine learning tools is Jupyter notebook. It's a platform that can process data quickly and efficiently. It also supports three languages: Julia, R, and Python.

4. IMPLEMENTATION

4.1 Dataset Used

- All of the information utilised in the study came from the Indian government's freely accessible records, and only a small number of key characteristics with the greatest influence on agricultural production were chosen for the study from the large initial dataset.

- Here is how the dataset looks like (given below)



```
  a255369      T-High     T-Low      Rel  Hum   Soil  Tmp  Wind  Sp.  SolarRad   Precip
 date/time     C          C          %          C@10cm     m/s        MJ/m**2    mm
 1 2015 2400   22.933     10.712     53.538     15.273      2.839      8.699      0.000
 2 2015 2400   27.594     10.702     69.315     16.576      4.495     18.957     10.922
 3 2015 2400   28.544     14.503     68.605     17.329      3.831     13.629     19.820
 4 2015 2400   16.673     12.482     87.045     15.240      3.431      3.164      4.064
 5 2015 2400   24.373     13.592     81.868     16.571      3.210     13.005      1.016
 6 2015 2400   21.083     15.043     89.284     16.967      6.083      6.054     50.796
 7 2015 2400   23.953     13.052     80.995     18.363      4.700     18.295     25.658
 8 2015 2400   15.313     10.232     81.538     16.075      3.616      6.847      0.000
 9 2015 2400   19.773      9.032     77.394     16.306      3.625     13.124      0.000
10 2015 2400   21.683      7.609     84.465     17.433      4.967      8.813      7.112
11 2015 2400   10.602      4.561     69.415     12.942      3.906      7.407      0.000
12 2015 2400   20.903      3.173     55.770     13.989      2.335     24.631      0.000
13 2015 2400   22.653      8.222     60.435     16.430      5.134     21.570      8.890
14 2015 2400   20.303     11.652     92.285     16.400      3.129      6.958     10.668
15 2015 2400   20.993     11.752     92.936     15.728      3.268      6.314      9.906
16 2015 2400   26.554     15.643     81.461     18.640      5.120     13.810      2.032
17 2015 2400   25.603     12.642     71.177     19.380      4.490     20.544      0.000
18 2015 2400   16.883      7.174     61.030     17.679      4.847     24.454      0.000
19 2015 2400   10.972      2.903     72.797     14.874      3.511     10.489      2.286
20 2015 2400   10.772      5.103     88.292     13.693      3.251      5.805      1.016
21 2015 2400   22.173      2.409     54.680     15.144      2.067     26.197      0.000
22 2015 2400   18.533      5.603     67.233     15.258      1.841      8.926      0.000
23 2015 2400   17.023     12.252     89.727     15.440      2.654      3.578      0.254
24 2015 2400   20.813     14.873     92.269     16.915      2.265      6.056      1.778
25 2015 2400   27.094     14.473     73.678     19.508      3.476     18.926      0.000
26 2015 2400   20.723     13.462     85.640     18.745      2.213      9.608      2.286
27 2015 2400   28.914      9.162     54.097     19.376      1.876     23.901      0.000
```

5. WORK FLOW OF THE PROJECT

5.1 Fit the Models

- ✓ Load Data
- ✓ Prepare Data
- ✓ Train algorithm
- ✓ Fit the model on training data set
- ✓ Save the model to disk

Once All of the models are saved

- ✓ Make Prediction on Test data using each of the 3 models
- ✓ Calculate Accuracy Score

After we get accuracy score in each of the models

- ✓ Compare Accuracy Scores
- ✓ Choose the best model
- ✓ Apply the best model on test data

5.2 Data Exploration And Munging

- ✓ A small number of places report measurements for less than 14 days out of the whole 365-day period each year.
- ✓ The data for these places was deleted.
- ✓ Each year, 5% of the sites were eliminated using this method, accounting for only 0.2 percent of the raw data.
- ✓ ✓ As given, the data set was already quite clean and comprised just a tiny number of missing or NULL/NaN values.
- ✓ Several unnecessary columns were dropped.
- ✓ Two types of weather features in the dataset are important

=> Temperature

=> Accumulated precipitation

Temperature is important as temperatureMax and temperatureMin, determines the daily accumulated heat (GDD = (temperatureMax + temperatureMin) /2 - TemperatureBase).

The accumulated precipitation is important because it determines if wheat can get enough water supply. DayInSeason determines will wheat can grow longer. The longer it grows, the more biomass (yield) it can accumulate.

The kept columns included

- ✓ precipAccumulation
- ✓ temperatureMin
- ✓ temperatureMax
- ✓ average temperature
- ✓ DayInSeason
- ✓ Soil Temp
- ✓ Wind Speed
- ✓ Solar Radiation

5.3 Using different models to fit training data

For calculation of accuracy score we have chosen three models

- ✓ Linear regression
- ✓ SVM
- ✓ DecisionTree

Linear Regression Model

```
#import the libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
dataset = pd.read_csv('D:\Datasets\wheat_data.csv')
dataset.shape # display no of rows and columns
dataset.head() # display first 5 rows

#Prepare Data

y=data.yield
# y is target column to predict
x=data.drop('yield',axis=1)
#take all rows data of all columns except yield column
#Split Data
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
#The line test_size=0.2 suggests that the test data should be 20% of the dataset and the rest should be train data
#Train algorithm

from sklearn.linear_model import LinearRegression
model = LogisticRegression()
model.fit(X_train, Y_train)
# save the model to disk
```

```
lr_model_file = 'lr_model.sav'
pickle.dump(model, open(lr_model_file, 'wb'))
```

5.4 SVM Model

```
#Apply Polynomial Kernel
 from sklearn.svm import SVC
 svclassifier = SVC(kernel='poly', degree=8)
 svclassifier.fit(X_train, y_train)
 svm_model_file = 'svm_model.sav'
 pickle.dump(model, open(svm_model_file, 'wb'))
```

5.5 Decision Tree Model

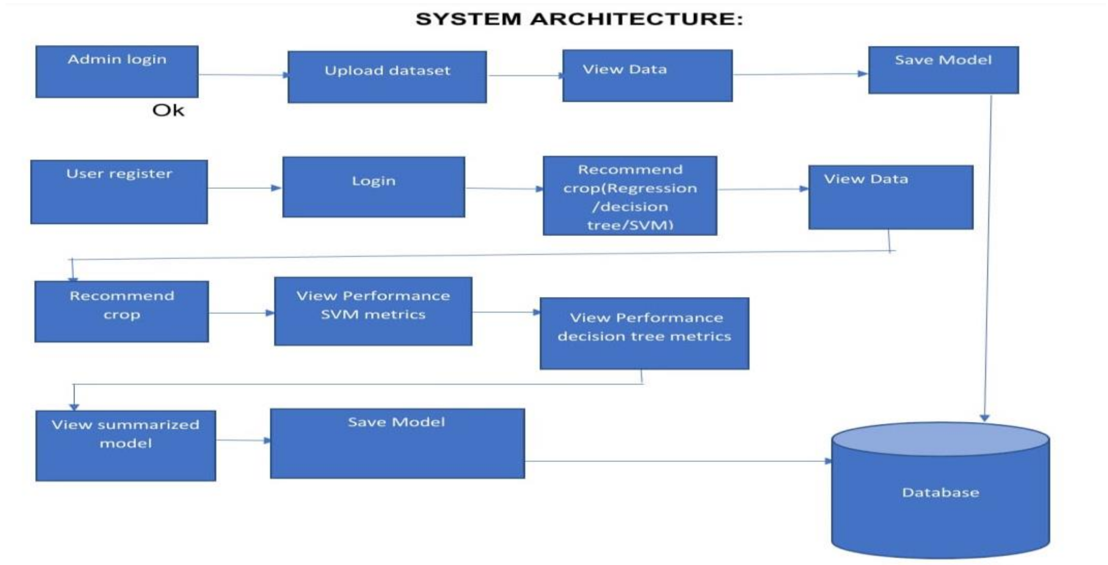
```
#Apply Decision Tree
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier()
classifier.fit(X_train, y_train)
dt_model_file= 'dt_model.sav'
pickle.dump(model, open(dt_model_file, 'wb'))
```

5.6 Prediction using the models and finding accuracy scores

```
# load the  linear regression model from disk and find score
loaded_lr_model = pickle.load(open(lr_model_file, 'rb'))
lr_result = loaded_lr_model.score(X_test, Y_test)
# load the  svm model from disk and find score
loaded_svm_model = pickle.load(open(svm_model_file, 'rb'))
lr_result = loaded_svm_model.score(X_test, Y_test)
# load the decision tree  model from disk and find score
loaded_dt_model = pickle.load(open(dt_model_file, 'rb'))
lr_result = loaded_dt_model.score(X_test, Y_test)
```

Out of the above three Models Linear Regression seem to give more accurate result.

6. System Design



7. CHALLENGES

Our main challenge was being unfamiliar with the subject matter. More domain knowledge would have helped in engineering more powerful features and provided better intuition in judging performance.

The compromise was then to use my best judgment for overcoming our limited domain knowledge as well as having to make assumption on the concrete business incentive.

It was also tricky to overcome overfitting completely. Getting more data would have been an obvious solution, but for brevity's sake I decided against this effort. Switching to a different algorithm might have helped, but (possibly) at the cost of performance. More careful feature engineering has the potential to offset this effect. Or using ensemble techniques.

8. SCOPE FOR FUTURE WORK

- ✓ While the performance of the model appears quite good, there is also some residual overfitting, even after careful tuning.
- ✓ In future iterations, these issues could be addressed by: getting more data, engineering additional and/or different features.
- ✓ We can use Raspberry pi to gather real-time data and use them for real-time prediction.
- ✓ We can use soil moisture sensor, pH sensor, Temperature and humidity sensor for collecting moisture info, pH of the soil, temperature and humidity.
- ✓ We can create an IoT Analytics channel, a pipeline, and a data store to store these sensor values in a mysql db/ rds in aws.
- ✓ The system can be extended to the mobile application to help the farmers to see the results real-time.

9. CONCLUSION

The study presented in this work introduce practical, cheap, and easy-to-use tool that is useful to increase the productivity of farmers , save environmental resources and pursue economic profits.

More features included in a model can improve the accuracy. However, the "right" feature included in the model can significantly improve predicting accuracy.

(For example, when I exclude latitude and longitude from the model, the accuracy score drops to 0.1. While when I include the latitude and longitude, the accuracy score includes to 0.23)

Using only time series features (like weather data) to train and predict the label (may not be the best way. Especially when the yield is strongly related to other factors like environment condition (weather, soil, and geolocation).

Adding geolocation related data to the dataset which representative geo-features. Such as soil features, what soil type for the given geolocations.

REFERENCES

1. Vaneesbeer Singh,Abid Sarwar, “Analysis of soil and prediction of crop yield (Rice) using Machine Learning approach” IJARCSE,vol. 5, Issue 8,2017.

2. Sabri Arik, Tingwen Huang, Weng Kin Lai, Qingshan Liu ,“Soil Property Prediction: An Extreme Learning Machine Approach” Springer, vol. 3, Issue 4,666-680,2015

3. Aditya Shastry, H.A Sanjayand E.Bhanushree,“Prediction of crop yield using Regression Technique”, International Journal of computing12 (2):96-102 2017,ISSN:1816-9503

4. E. Manjula , S. Djodiltachoumy,“A Model for Prediction of Crop Yield”, International Journal of Computational Intelligence and Informatics, Vol. 6: No. 4, March 2017

5. Mrs.K.R.Sri Preethaa, S.Nishanthini, D.SanthiyaK.Vani Shree ,“CropYield Prediction”,International Journal On Engineering Technology and Sciences – IJETS™ISSN(P): 2349-3968, ISSN (O):2349-3976 Volume III,Issue III, March- 2016

6.Xia Zhang, Yanli Sun, Kun Shang, Lifu Zhang, Senior Member, IEEE, and Shudong Wang “IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING” Pages 01,Fevruary-2016.

7. Wei Yao, OtmarLoffeld - Application and Evaluation of a Hierarchical Patch Clustering Method for Remote Sensing Images, VOL. 9, NO. 6, JUNE 2016 2279 – 2289.

8.Michael Johnson, William Hsieha, AlexJ. Cannonb, Andrew Davidsonc, FrédéricBédardd -Crop yield forecasting on the Canadian Prairies by remotely sensed vegetation indices and machine learning methods on Agricultural and Forest Meteorology 218–219 (2016).

9. Z. Xue, J. Li, L. Cheng, and P. Du, “Spectral–spatial classification of hyperspectral data via morphological component analysis-based image separation,” IEEE Transaction, vol. 53, no. 1, pp. 70–84, Jan. 2015.

10. J. L. Starck, M. Elad, and D. L. Donoho, “Image decomposition via the combination of sparse representations and a variational approach,” IEEE Trans. Image Process., vol. 14, no. 10, pp. 1570–1582, Oct. 2015.