

Crop Yield Prediction

Arshad Hussain, Sravya Jupalli , Vani Karanam

¹Associate Professor, Department of CSE, Bhoj Reddy Engineering College for Women, India

^{2,3}B.Tech Student, Department of CSE, Bhoj Reddy Engineering College for Women, India

Abstract:

Agriculture growth mainly depends on diverse soil parameters, namely Nitrogen, Phosphorus, Potassium, Crop rotation, Soil moisture, pH, surface temperature and weather aspects like temperature, rainfall, etc. Technology will prove to be beneficial to agriculture which will increase crop productivity resulting in better yields to the farmer. The proposed project provides a solution for Smart Agriculture by monitoring the agricultural field which can assist the farmers in increasing productivity to a great extent. This work presents a system, in a form of a website, which uses Machine Learning techniques in order to predict the most profitable crop in the current weather and soil conditions. This system can also help in predicting the yield of the crop using weather parameter, soil parameter and historic crop yield. Thus, the project develops a system by integrating data from various sources, data analytics, prediction analysis which can improve crop yield productivity and increase the profit margins of farmer helping them over a longer run.

Introduction

The project mainly focuses on the basic types of crops and their nature of yield in various types of seasons. This project is included with a huge amount of data sets wherein almost all types of crops and their wide range of behavior and yield in various types of seasons is provided. These large amount of data sets helps in predicting the right crop with high amount of productivity. Through this, a person who

does farming as their major occupation will get to know which type of plant serves them with best results.

Usually a farmer does not know the exact reason for failure of their crops. Despite of not having much awareness on the type of crop in type of season etc., farmers plant and harvest wrong type of crop in the wrong time or wrong season. This will have very much effect on his crop yield and in turn makes his well-being difficult. This project helps the farmers to predict which crop to be ploughed at the right situation and in the required area to get results at high stakes.

Problem Definition

Agriculture is a business with risk which depends on climate, geography and economic factors. The main intent and objective of the project is to help the farmers choose the right crop in the right time to increase their earnings and returns on their farms. This project estimates which crop is to be harvested in the right time and in the right season. Based on the requirements of the users (farmer), prediction is done and the necessary information is provided to him. Through this, they get benefitted with high returns for the amount of crop they planted and grown it with at most efforts.

Literature Survey

- Even now, Agriculture supports about 58% of total population, which is 75% at the time of independence i.e., a drop of 17%.

- A good amount of people in villages are leaving the agriculture and adopting other professions due to poor yields and returns.
- In a recent study, about 76% of farmers want to give up farming as there is no market and amount of production.
- All the money returns of farmer's crops is been eaten up by the brokers who are working intermediate to farmers and common people.
- They are earning huge amount of returns from the crop of farmers.

DESIGN

In designing the software following principles are followed:

1. **Modularity and partitioning:** software is designed such that, each system should consists of hierarchy of modules and serve to partition into separate function.
2. **Coupling:** modules should have little dependence on other modules of a system.
3. **Cohesion:** modules should carry out in a single processing function.
4. **Shared use:** avoid duplication by allowing a single module be called by other that need the function it provide

DATA FLOW DIAGRAM:

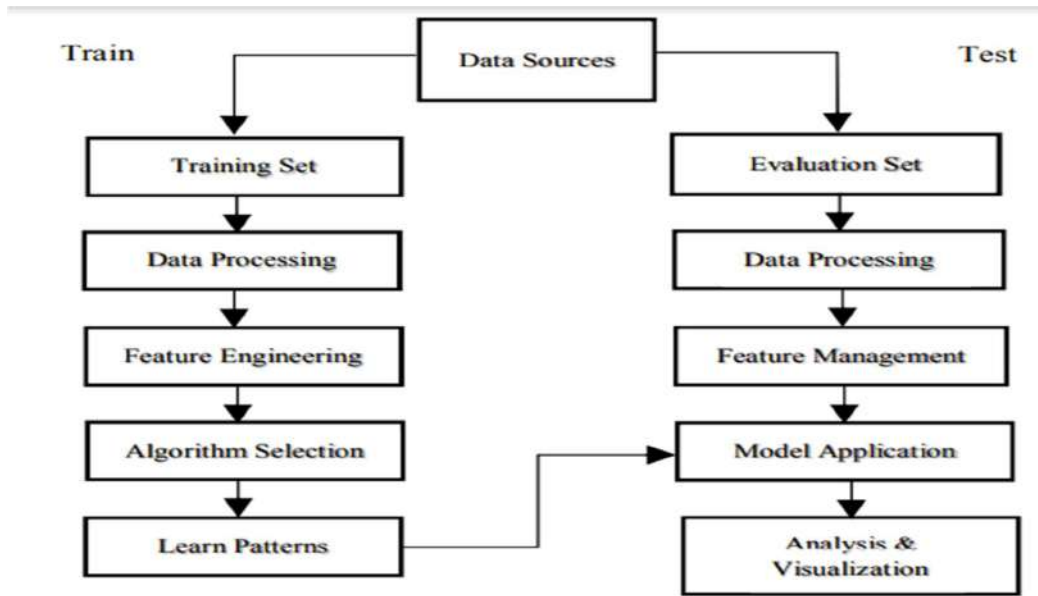
1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing

carried out on this data, and the output data is generated by this system.

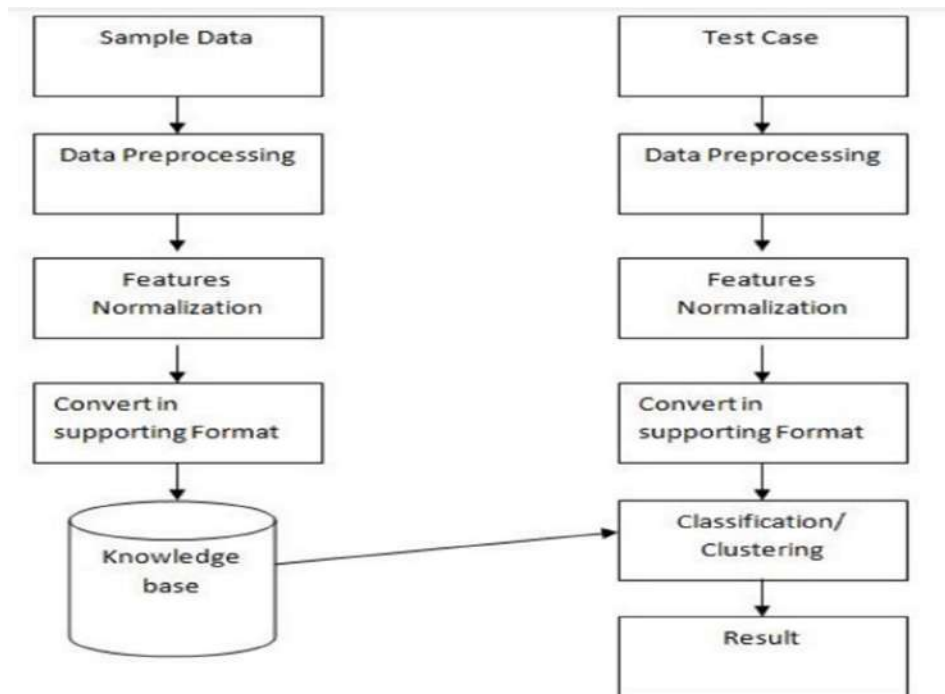
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional details

Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation.

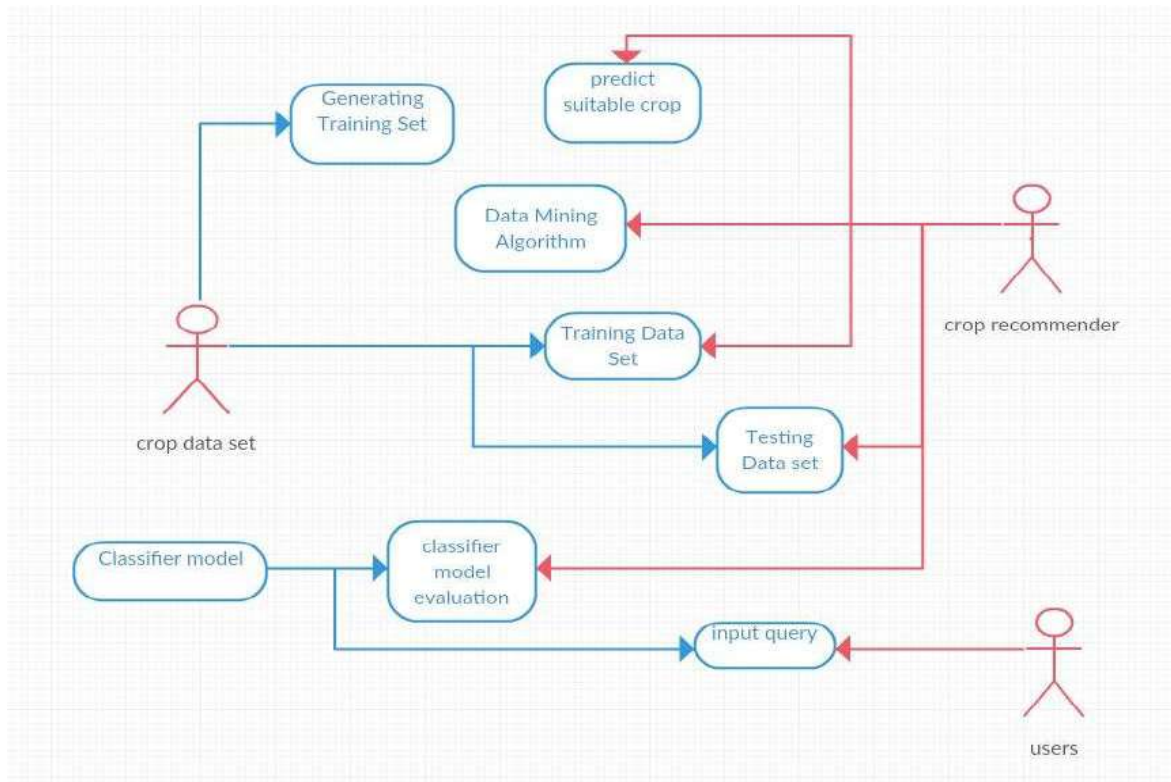
Data flow diagrams can be divided into logical and physical. The logical data flow diagram describes flow of data through a system to perform certain functionality of a business. The physical data flow diagram describes the implementation of the logical data flow.



System Architecture:



Technical Architecture:



Implementation:

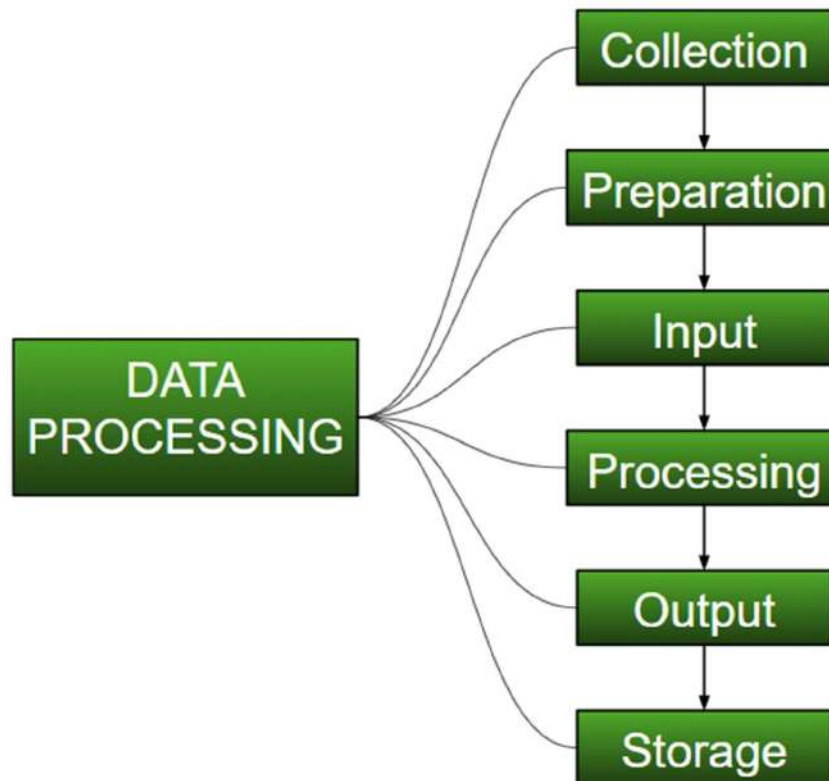
We proposed as an alternative to the user-based neighborhood approach. We first consider the dimensions of the input and output of the neural network. In order to maximize the amount of training data we can feed to the network, we consider a training example to be a user profile (i.e. a row from the user-item matrix R) with one rating withheld. The loss of the network on that training example must be computed with respect to the single withheld rating. The consequence of this is that each individual rating in the training set corresponds to a training example, rather than each user. As we are interested in what is essentially a regression, we choose to use root mean squared error (RMSE) with respect to known ratings as our loss function. Compared to the mean absolute error, root mean squared error more heavily penalizes predictions which are further off. We reason that this is good in

the context of recommender system because predicting a high rating for an item the user did not enjoy significantly impacts the quality of the recommendations. On the other hand, smaller errors in prediction likely result in recommendations that are still useful—perhaps the regression is not exactly correct, but at least the highest predicted rating are likely to be relevant to the user.

Data Processing is a task of converting data from a given form to a much more usable and desired form i.e. making it more meaningful and informative. Using Machine Learning algorithms, mathematical modeling and statistical knowledge, this entire process can be automated. The output of this complete process can be in any desired form like graphs, videos, charts, tables, images and many more, depending on the task we are performing and the requirements of the machine. This might seem to be simple but when it comes to really big

organizations like Twitter, Facebook, Administrative bodies like Paliament, UNESCO and

health sector organizations, this entire process needs to be performed in a very structured manner.



Collection:

The most crucial step when starting with ML is to have data of good quality and accuracy. Data can be collected from any authenticated source like data.gov.in, [Kaggle](https://www.kaggle.com/) or [UCI dataset repository](https://archive.ics.uci.edu/). For example, while preparing for a competitive exam, students study from the best study material that they can access so that they learn the best to obtain the best results. In the same way, high-quality and accurate data will make the learning process of the model easier and better and at the time of testing, the model would yield state of the art results. A huge amount of capital, time and resources are consumed in collecting data. Organizations or researchers have to decide what kind of data they

need to execute their tasks or research.

Example: Working on the Facial Expression Recognizer, needs a large number of images having a variety of human expressions. Good data ensures that the results of the model are valid and can be trusted upon.

Preparation:

The collected data can be in a raw form which can't be directly fed to the machine. So, this is a process of collecting datasets from different sources, analyzing these datasets and then constructing a new dataset for further processing and exploration. This preparation can be performed either manually or from the automatic approach. Data can also be prepared in numeric forms also which would fasten the model's learning.

Example: An image can be converted to a matrix of $N \times N$ dimensions, the value of each cell will indicate image pixel.

Input:

Now the prepared data can be in the form that may not be machine-readable, so to convert this data to readable form, some conversion algorithms are needed. For this task to be executed, high computation and accuracy is needed. Example: Data can be collected through the sources like MNIST Digit data(images), twitter comments, audio files, video clips.

Processing:

This is the stage where algorithms and ML techniques are required to perform the instructions provided over a large volume of data with accuracy and optimal computation.

Output:

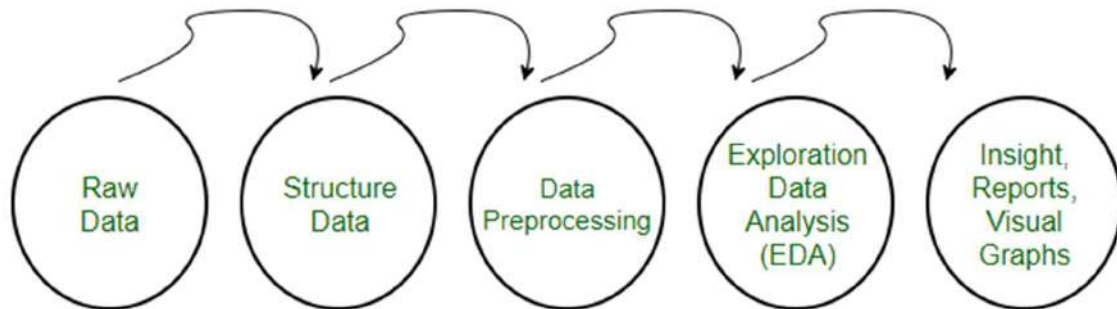
In this stage, results are procured by the machine in a meaningful manner which can be inferred easily by the user. Output can be in the form of reports, graphs, videos, etc

Storage:

This is the final step in which the obtained output and the data model data and all the useful information are saved for the future use.

Data Preprocessing for Machine learning in Python

- Pre-processing refers to the transformations applied to our data before feeding it to the algorithm.
- Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis.



Need of Data Preprocessing

- For achieving better results from the applied model in Machine Learning projects the format of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format, for example, Random Forest algorithm does not support null values, therefore to

execute random forest algorithm null values have to be managed from the original raw data set.

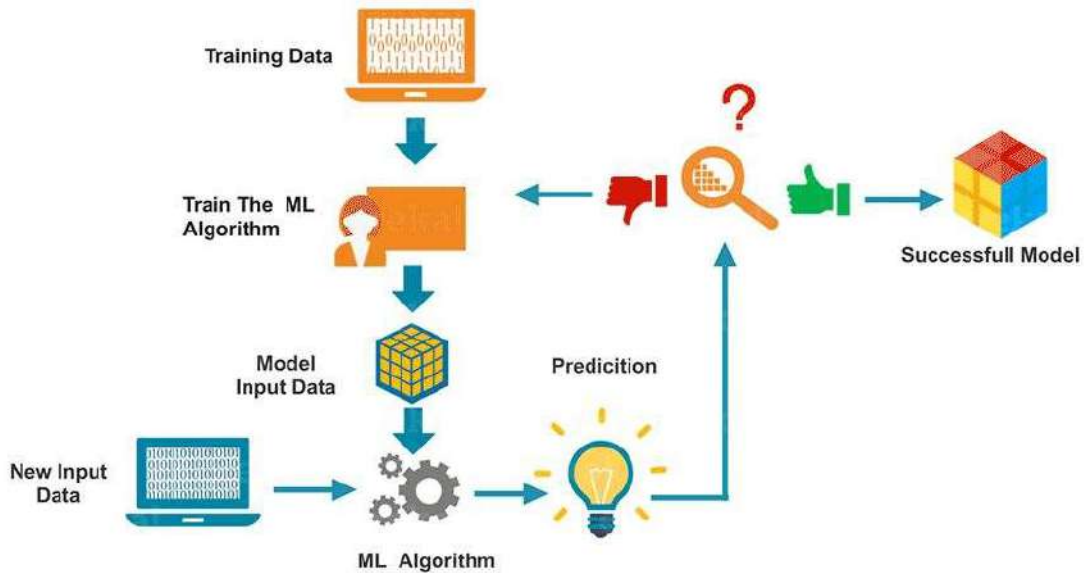
- Another aspect is that data set should be formatted in such a way that more than one Machine Learning and Deep Learning algorithms are executed in one data set, and best out of them is chosen.

Machine Learning algorithm is trained using a training data set to create a model. When new input

data is introduced to the ML algorithm, it makes a prediction on the basis of the model.

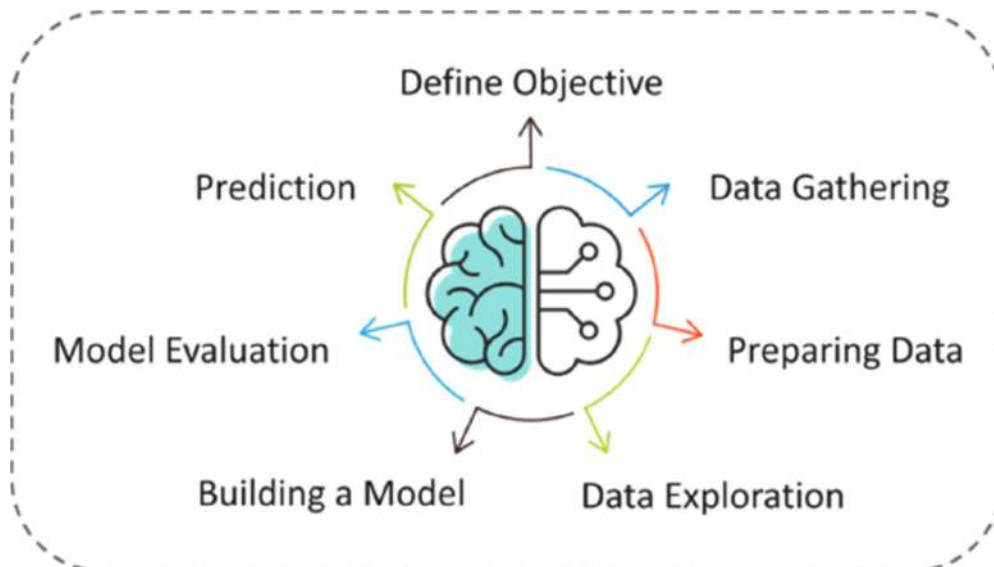
The prediction is evaluated for accuracy and if the accuracy is acceptable, the Machine Learning

algorithm is deployed. If the accuracy is not acceptable, the Machine Learning algorithm is trained again and again with an augmented training data set.



The Machine Learning process involves building a Predictive model that can be used to find a solution for a Problem Statement. To understand the Machine

Learning process let's assume that you have been given a problem that needs to be solved by using Machine Learning.



The below steps are followed in a Machine Learning process:

Step 1: Define the objective of the Problem Statement

At this step, we must understand what exactly needs to be predicted. In our case, the objective is to predict the possibility of rain by studying weather conditions. At this stage, it is also essential to take mental notes on what kind of data can be used to solve this problem or the type of approach you must follow to get to the solution.

Step 2: Data Gathering

At this stage, you must be asking questions such as,

- What kind of data is needed to solve this problem?
- Is the data available?
- How can I get the data?

Once you know the types of data that is required, you must understand how you can derive this data. Data collection can be done manually or by web scraping. However, if you're a beginner and you're just looking to learn Machine Learning you don't have to worry about getting the data. There are 1000s of data resources on the web, you can just download the data set and get going.

Coming back to the problem at hand, the data needed for weather forecasting includes measures such as humidity level, temperature, pressure, locality, whether or not you live in a hill station, etc. Such data must be collected and stored for analysis.

Step 3: Data Preparation

The data you collected is almost never in the right format. You will encounter a lot of inconsistencies in the data set such as missing values, redundant variables, duplicate values, etc. Removing such inconsistencies is very essential because they might lead to wrongful computations and predictions. Therefore, at this stage, you scan the data set for any inconsistencies and you fix them then and there.

Step 4: Exploratory Data Analysis

Grab your detective glasses because this stage is all about diving deep into data and finding all the hidden data mysteries. EDA or Exploratory Data

Analysis is the brainstorming stage of Machine Learning. Data Exploration involves understanding the patterns and trends in the data. At this stage, all the useful insights are drawn and correlations between the variables are understood.

For example, in the case of predicting rainfall, we know that there is a strong possibility of rain if the temperature has fallen low. Such correlations must be understood and mapped at this stage.

Step 5: Building a Machine Learning Model

All the insights and patterns derived during Data Exploration are used to build the Machine Learning Model. This stage always begins by splitting the data set into two parts, training data, and testing data. The training data will be used to build and analyze the model. The logic of the model is based on the Machine Learning Algorithm that is being implemented.

Choosing the right algorithm depends on the type of problem you're trying to solve, the data set and the level of complexity of the problem. In the upcoming sections, we will discuss the different types of problems that can be solved by using Machine Learning.

Step 6: Model Evaluation & Optimization

After building a model by using the training data set, it is finally time to put the model to a test. The testing data set is used to check the efficiency of the model and how accurately it can predict the outcome. Once the accuracy is calculated, any further improvements in the model can be implemented at this stage. Methods like parameter tuning and cross-validation can be used to improve the performance of the model.

Step 7: Predictions

Once the model is evaluated and improved, it is finally used to make predictions. The final output can be a Categorical variable (eg. True or False) or

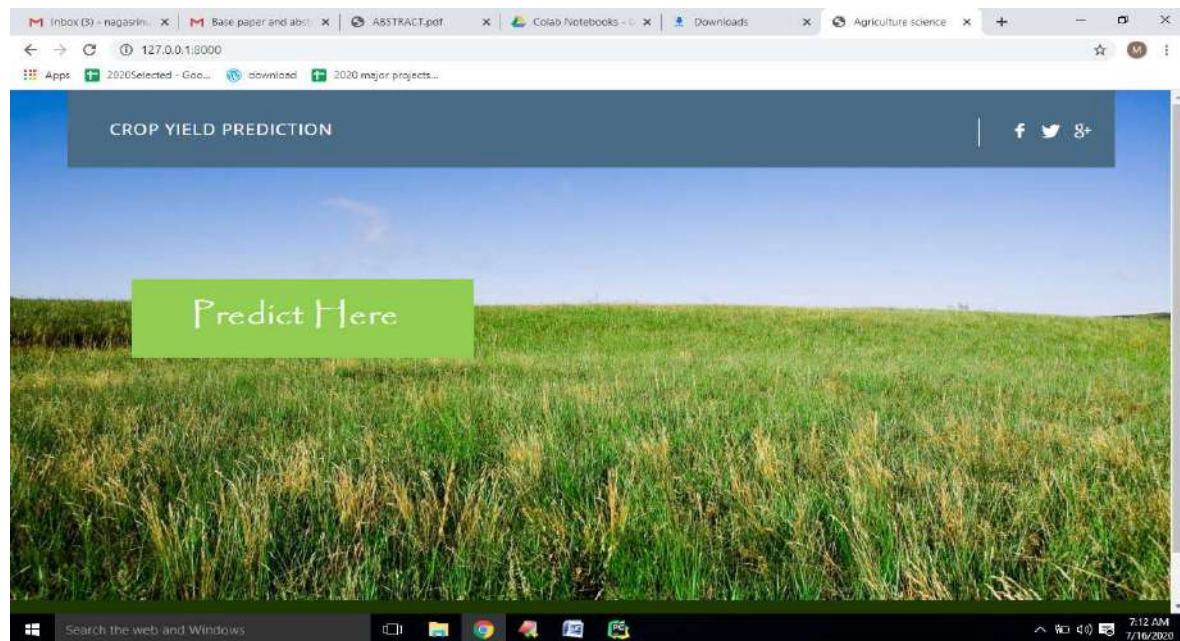
it can be a Continuous Quantity (eg. the predicted value of a stock).

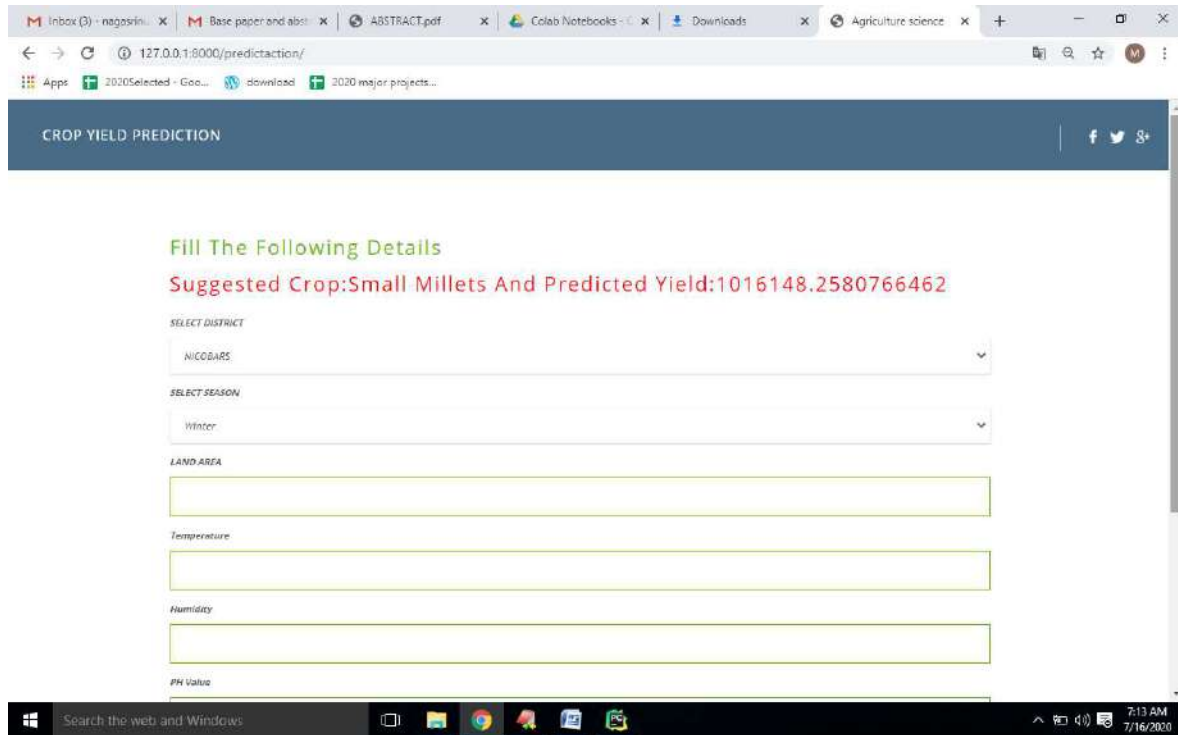
In our case, for predicting the occurrence of rainfall, the output will be a categorical variable.

RESULTS

Screen Shots:

Original Data Set





CROP YIELD PREDICTION

Fill The Following Details

Suggested Crop:Small Millets And Predicted Yield:1016148.2580766462

SELECT DISTRICT
NIOBARIS

SELECT SEASON
Winter

LAND AREA

Temperature

Humidity

PH Value

Test Cases

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Test objectives

Test cases

Tested	Test name	Inputs	Expected output	Actual Output	Result
1	load dataset	dataset	dataset loaded	successfully loaded	pass

Tested	Test name	Inputs	Expected output	Actual Output	Result
2	Splitting dataset into training and validation set	dataset	spitted to training and validation	successfully spitted to train data and validation data	pass

Tested	Test name	Inputs	Expected output	Actual Output	Result
--------	-----------	--------	-----------------	---------------	--------

4	Model creation	Keras module with lstm	model created	successfully model created	pass
---	----------------	------------------------	---------------	----------------------------	------

Tested	Test name	Inputs	Expected output	Actual Output	Result
5	training set evaluation	train data	training done	successfully trained	pass

Tested	Test name	Inputs	Expected output	Actual Output	Result
6	Validating test dataset	test data	test dataset validated	successfully validated	pass

Tested	Test name	Inputs	Expected output	Actual Output	Result
7	getting accuracy	test dataset	accuracy in percentage	successfully got accuracy	pass

Tested	Test name	Inputs	Expected output	Actual Output	Result
8	load test set	test data	detecting price	successfully detected	pass

Tested	Test name	Inputs	Expected output	Actual Output	Result
9	print result	test results	printing test results or save in csv file	Successfully saved	success

Conclusion:

With this model, we have created a rudimentary model that is able to forecast to a certain extent. Even though the model is not perfect, we have one that can approximate to the past data pretty well. But for new data, we would require more parameter tuning.

There exists great potential to improve sales forecasting accuracy in the Ecommerce domain. One good opportunity is to utilize the correlated and similar sales patterns available in a product portfolio. In this paper, we have introduced a novel demand forecasting framework based on LSTMs that exploits non-linear relationships that exist in the

E-commerce business. We have used the proposed approach to forecast the sales demand by training a global model across the items available in a product assortment hierarchy. Our developments also present several systematic grouping strategies to our base model, which are in particular useful in situations where product sales are sparse. Our methodology has been evaluated on a real-world E-commerce database from Walmart.com. To demonstrate the robustness of our framework, we have evaluated our methods on both category level and super-department level datasets. The results have shown that our methods have outperformed the state-of-the-art univariate forecasting techniques.

Furthermore, the results indicate that E-commerce product hierarchies contain various cross-product demand patterns and correlations are available, and approaches to exploit this information are necessary to improve the sales forecasting accuracy in this domain.

Feature Scope:

In the future, we are planning to explore the ability to incorporate multiple stores with a single LSTM to extract cross-series information to improve forecasting accuracy. We expect such features to improve time-series forecasting by comprehending the interdependencies between the stores such as competition, partnerships, market distribution etc. Moreover, it is interesting to investigate the importance of incorporating information that describes the future beyond the day being predicted. For instance, the customer buying behaviour for a particular day can significantly affect the fact whether the store is going to be closed in the following day. Yet, the time-series models may not be able to anticipate such relationships without explicitly providing information that represents the future even beyond the day that is being forecast. Therefore, we will be exploring such extensions with our technique in the future.

References

- Hyndman, R. et al., 2008. Forecasting with Exponential Smoothing: The State Space Approach, Springer Science & Business Media.
- Box, G.E.P. et al., 2015. Time Series Analysis: Forecasting and Control, John Wiley & Sons.
- Box, G.E.P. & Cox, D.R., 1964. An Analysis of Transformations. Journal of the Royal Statistical Society. Series B, Statistical methodology, 26(2), pp.211-252.
- Yeo, J. et al., 2016. Browsing2purchase: Online Customer Model for Sales Forecasting in an E-Commerce Site. 25th Int.Conf.Comp. on World Wide Web.
- Ramanathan, U., 2013. Supply chain collaboration for improved forecast accuracy of promotional sales. Int. Journal of Operations & Production Management.
- Kulkarni, G., Kannan, P.K. & Moe, W., 2012. Using online search data to forecast new product sales. Decision support systems, 52(3), pp.604-611.
- Zhao, K. & Wang, C., 2017. Sales Forecast in E-commerce using Convolutional Neural Network. arXiv [cs.LG].
- Seeger, M.W., Salinas, D. & Flunkert, V., 2016. Bayesian Intermittent Demand Forecasting for Large Inventories. In Proceedings of the 29th NIPS.
- Snyder, R., Ord, J.K. & Beaumont, A., 2012. Forecasting the intermittent demand for slow-moving inventories: A modelling approach. IJF, 28(2), pp.485-496.
- Zhang, G., Patuwo, B.E. & Hu, M.Y., 1998. Forecasting with artificial neural networks: The state of the art. International journal of forecasting, 14(1), pp.35-62.
- Yan, W., 2012. Toward automatic time-series forecasting using neural networks. IEEE transactions on neural networks and learning systems, 23(7), pp.1028-1039.
- Zimmermann, H.-G., Tietz, C. & Grothmann, R., 2012. Forecasting with RNNs: 12 Tricks. In Neural Networks. Lecture Notes in Computer Science. pp. 687-707.
- Trapero, J.R., Kourentzes, N. & Fildes, R., 2015. On the identification of sales forecasting models in the presence of promotions. The Journal of the ORS.
- Borovykh, A., Bohte, S. & Oosterlee, C.W., 2017. Conditional Time Series Forecasting with Convolutional Neural Networks arXiv [cs.AI].

15. Flunkert, V., Salinas, D. & Gasthaus, J., 2017. DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks. arXiv [cs.AI].
16. Wen, R. et al., 2017. A Multi-Horizon Quantile Recurrent Forecaster. arXiv [stat.ML].