Chapter 1

Introduction

1.1 Introduction

Farming is one of India's most common occupations. For a large portion of India's population, agriculture is their primary source of income. The need for producing has increased dramatically over time. Loss may be reduced when it comes to crop productivity. Annually, India loses between 16 and 20 percent of its overall production. Crop yield forecast is critical in this circumstance. It supports farmers by assisting them in determining the best crop for a given set of circumstances. The goal of agricultural planning is to increase crop yields while exploiting a limited amount of land resources.

In our research, which we found in the previous research papers is that everyone uses climatic factors like rainfall, sunlight and agricultural factors like soil type, nutrients possessed by the soil (Nitrogen, Potassium, etc.) but the problem is we need to gather the data and then a third party does this prediction and then it is explained to the farmer and this takes a lot of effort for the farmer and he doesn't understand the science behind these factors. To make it simple and which can be directly used by the farmer this paper uses simple factors like which state and district is the farmer from, which crop and in what season (as in Kharif, Rabi, etc.).

In India, there are more than a hundred crops planted around the whole country. These crops are categorized for better understanding and visualization. The data for this research has been acquired from the Indian Government Repository [1]. The data consists of attributes - State, District, Crop, Season, Year, Area and Production with around 2.5 Lakh observations. The fig. 1. depicts the states and territories of India which visualize that which category of crops are famous in which season. We used advanced regression techniques - Lasso, ENet and Kernel Ridge and further we used stacking of these models to minimize the error And to obtain better prediction.

1.2 Project Objective:

The whole project is implemented and carried out in Python and we used Visual Studio for integration of Python and Web interface to build a web application. The project is documented as follows - Literature review which discusses the research papers based on our work and background theoretical analysis which has been studied in the section, The proposed methodology - which explains the detailing of our proposed and it's working with the data, elucidates the workflow and research analysis of our project, the implementation snippets of python and web application codes (HTML and CSS) Result analysis which consists of tabular values and detailing of results, Conclusion and Future work along with references.

Chapter 2

Literature Survey

Rashid, M., Bari, B. S., Yusup, Y., Kamaruddin, M. A., & Khan, N. (2021) made a Comprehensive Review of Crop Yield Prediction Using Machine Learning Approaches with Special Emphasis on Palm Oil Yield Prediction using Reinforced random forest model that the current status of palm oil yield around the world is presented [1]. Malik, P., Sengupta, S., & Jadon, J. S. (2021) made a Comparative Analysis of Soil Properties to Predict Fertility and Crop Yield using Machine Learning Algorithms using K nearest neighbor algorithm, Naive Bayes algorithm and Decision Trees that where Crop yield prediction has been performed on self-obtained dataset and Yield of three crops - tomato, potato and chilli in alluvial soil, red soil and black soil was predicted[2]. Guna Sekhar Sajja; Subhesh Saurabh Jha (2021) made An Investigation on Crop Yield Prediction Using Machine Learning using K Nearest Neighbour MAE - Mean Absolute Error RMSE - Root Mean Squared Error algorithms which Describes a machine learning-based system for agricultural yield prediction[3]. Saeed Khaki, Lizhi Wang and Sotirios V. Archontoulis(2020) developed a A CNN-RNN Framework for Crop Yield Prediction which combines CNNs, fully connected layers, and RNN and predicted the yields of corn and soybean for three years[4]. Miss.Snehal S. Dahikar , Dr.Sandeep V.Rode(2014) developed an Agricultural Crop Yield Prediction Using Artificial Neural Network Approach and concluded that ANN is beneficial tool for crop prediction[5]. Thomas van Klompenburga, Ayalew Kassahuna, Cagatay Catalb (2020) performed A systematic literature review on crop yield prediction system using machine learning techniques[6]. Elavarasan, D., & Durai Raj Vincent, P. M. (2021) proposed a Fuzzy deep learning-based crop yield prediction model that Investigates the crop yield prediction of paddy crop for the Vellore district located in the southern part of India[7]. Potnuru Sai Nishant, Pinapa Sai Venkat, Bollu Lakshmi Avinash, B.Jabber (2020) proposed a Crop Prediction using Machine Learning based on advanced regression algorithms such as Lasso, ridge, enet regression which predicts the yield of the crop[8]. Petkar,O. (2016, July), the same authors who applied SVM and neural networks for rice crop yield vaticination, proposed a new decision system that's an interface to give input and admit affair. (9).A. Chakrabarty etal. (2018, December) delved crop vaticination in Bangladesh, where they primarily cultivate three types of rice jute, wheat, and potato. Their study employed a deep neural network to dissect data that included 46 parameters. Among them were soil composition, toxin type, soil type and structure, soil thickness, response, and texture(10). Jintrawet,A. etal. (2008, May) used the SVR model for crops similar as rice to prognosticate yield, with the model divided into three way prognosticating soil nitrogen weight, rice stem weight, and rice grain weight. Along with those three way, their factors included solar radiation, temperature, and rush (11). Miniappan,N. etal. (2014, August) used an artificial neural network to model amulti-layer perceptron model with 20 retired layers for wheat yield vaticination, taking into account factors similar as sun, rain, frost, and temperature (12). Manjula, A etal. developed a crop selection and yield vaticination model that took into account colorful indicators similar as foliage, temperature, and regularized difference foliage as factors. For a better understanding, they distinguished between factors, agronomic factors, and other disturbances caused by the vaticination (13). Mariappan,A.K., and associates delved rice crop statistics in Tamil Nadu, India. They took into climate account rudiments similar as soil, temperature, sun, downfall, toxin, paddy, and nonentity type, as well as pollution and season (14).Verma,A. etal. (2015, December) used crop vaticination ways similar as Naive Bayes and the K-NN algorithm on soil datasets containing nutrients similar as zinc, bobby, manganese, pH, iron, sulfur, phosphorus, potassium, nitrogen, and organic carbon (15). Kalbande,D.R. etal. (2018) prognosticated sludge yield using support vector retrogression, multi polynomial retrogression, and arbitrary timberretrogression and estimated the models using criteria similar as MAE, RMSE, and R- forecourt values (16). Rahman,R.M., etal. (2015, June) primarily used clustering ways to prognosticate crop yield. The paperdescribed the analysis of major crops in Bangladesh and classified the variables as environmental and biotic. For bracket, direct retrogression, ANN, and the KNN approach were used (17). Hegde,M. etal. (2015, June) used multiple direct retrogression and neuro- fuzzy systems to prognosticate crop yield using biomass, soil water, radiation, and downfall as input parameters for the exploration, with wheat as the primary crop (18). Sujatha,R., and Isakki,P. (2016) used bracket ways similar as ANN, j48, Nave Bayes, Random Forest, and Support VectorMachines.In addition, they've included climatic and soil parameters as features in their modeling (19). Ramalatha,M. etal.(2018, October) combined Kmeans clustering and bracket grounded on a modified K-NN approach. The information was gathered in Tamil Nadu, India, where the most abundant crops were rice, sludge, ragi, sugarcane, and tapioca (20). Singh,C.D., etal. (2014, January) developed a crop-advice operation that works in a many Madhya Pradesh sections. The stonerwould input pall cover, downfall, temperature, and preliminarily recorded yield, and the system would anticipate the yield, marker the crop, and admit the results grounded on the detector values established (21).

Chapter 3

Software Requirements Specification

3.1 Software Requirements:

Operating System: Windows 11

Technologies Used: Machine Learning

Platform: Visual Studio

3.2 Hard Requirements:

CPU: Processors above Intel Corei5 10^{th} GEN

RAM: 16GB

3.3 Required Library:

- **Pandas:** Pandas is essentially used for data analysis. Pandas enables importing data from numerous file formats such as comma-separated-values, JSON, SQL, Microsoft Excel. Pandas enables numerous data manipulation operations such as merging, reshaping, selecting, as well as data cleaning, and data wrangling features.

- **Numpy:** NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices.

- **Sklearn:** Scikit-Learn is a free machine learning library for Python. It supports both supervised and unsupervised machine learning, providing diverse algorithms for classification, regression, clustering, and dimensionality reduction. The library is built using many libraries you may already be familiar with, such as NumPy and SciPy. It also plays well with other libraries, such as Pandas and Seaborn.

Chapter 4

System Analysis

4.1 Requirement Analysis

For the development of an efficient crop yield prediction model, there are several factors that have to be known and come into help to build a model from the raw data by analyzing the data and training the model accordingly.

4.2 Problem Statement:

The major problem for farmers that they explicitly need to test their soil and other parameters for testing and there is a need for third party who need to test and understand what the best crop for that particular instance of time could be. We are living in an advent of technology where everyone has a smartphone from all classes of people. So, we can create an application where the farmer himself or herself can get know what would be the best crop to cultivate on the basis of simple parameters which they give as input and those parameters have been discussed in the project documentation as well.

4.3 Functional Requirements

- Gathering the comments on the posts.
- Identifying and comparing various machine learning techniques.
- This model should provide accurate results for the dataset.

Chapter 5

Regression Techniques

5.1 Lasso Regression

In statistics and machine learning, lasso (least absolute shrinkage and selection operator; also Lasso or LASSO) is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the statistical model it produces. It was originally introduced in geophysics literature in 1986, and later independently rediscovered and popularized in 1996 by Robert Tibshirani, who coined the term and provided further insights into the observed performance.

Lasso was originally formulated for least squares models and this simple case reveals a substantial amount about the behavior of the estimator, including its relationship to ridge regression and best subset selection and the connections between lasso coefficient estimates and so-called soft thresholding. It also reveals that (like standard linear regression) the coefficient estimates do not need to be unique if covariates are collinear.

Though originally defined for least squares, lasso regularization is easily extended to a wide variety of statistical models including generalized linear models, generalized estimating equations, proportional hazards models, and M-estimators, in a straightforward fashion. Lasso's ability to perform subset selection relies on the form of the constraint and has a variety of interpretations including in terms of geometry, Bayesian statistics, and convex analysis. The LASSO is closely related to basis pursuit denoising.

- It also works as feature selection because lasso sets coefficients to zero.
- LOSS + α * $||X||$
- The only difference in the formula is there is no square for X

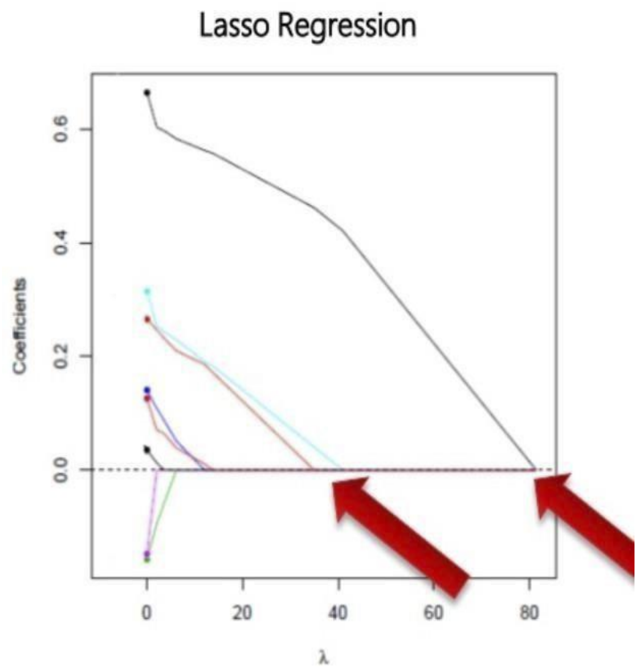


Figure No 5.1: Lasso Regression

5.2 Ridge Regression

Ridge regression belongs a class of regression tools that use L2 regularization. The other type of regularization, L1 regularization, limits the size of the coefficients by adding an L1 penalty equal to the absolute value of the magnitude of coefficients. This sometimes results in the elimination of some coefficients altogether, which can yield sparse models. L2 regularization adds an L2 penalty, which equals the square of the magnitude of coefficients. All coefficients are shrunk by the same factor (so none are eliminated). Unlike L1 regularization, L2 will not result in sparse models.

A tuning parameter (β) controls the strength of the penalty term. When $\beta = 0$, ridge regression equals least squares regression. If $\beta = \infty$, all coefficients are shrunk to zero. The ideal penalty is therefore somewhere in between 0 and ∞. Ridge regression uses a type of shrinkage estimator called a ridge estimator. Shrinkage estimators theoretically produce new estimators that are shrunk closer to the “true” population parameters. The ridge estimator is especially good at improving the least-squares estimate when multicollinearity is present.

Ridge regression is a way to create a parsimonious model when the number of predictor variables in a set exceeds the number of observations, or when a data set has multicollinearity (correlations between predictor variables).

- Used when there is multi-collinearity among the variables.
- LOSS + $\beta * ||X||^2$

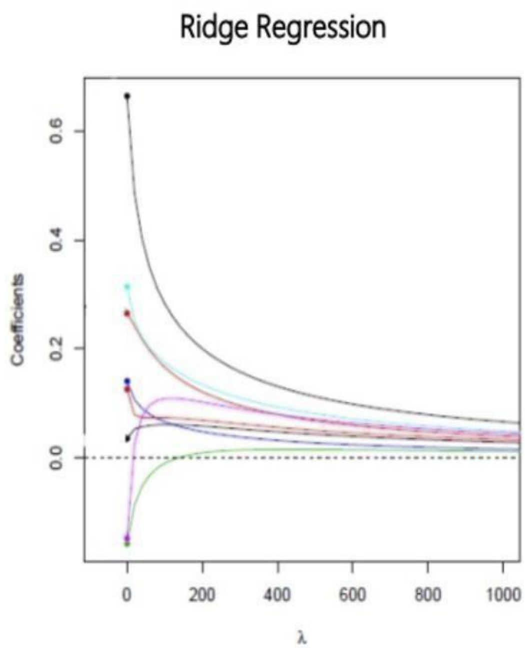


Figure No 5.2: Ridge Regression

5.3 Elastic-Net Regression

A third commonly used model of regression is the Elastic Net which incorporates penalties from both L1 and L2 regularization. In addition to setting and choosing a lambda value elastic net also allows us to tune the alpha parameter where X = 0 corresponds to ridge and X = 1 to lasso. Simply put, if you plug in 0 for alpha, the penalty function reduces to the L1 (ridge) term and if we set alpha to 1 we get the L2 (lasso) term. Therefore, we can choose an alpha value between 0 and 1 to optimize the elastic net. Effectively this will shrink some coefficients and set some to 0 for sparse selection.

- It also works as feature selection because lasso sets coefficients to zero.
- LOSS + $\beta * ||X||$
- The only difference in the formula is there is no square for W.

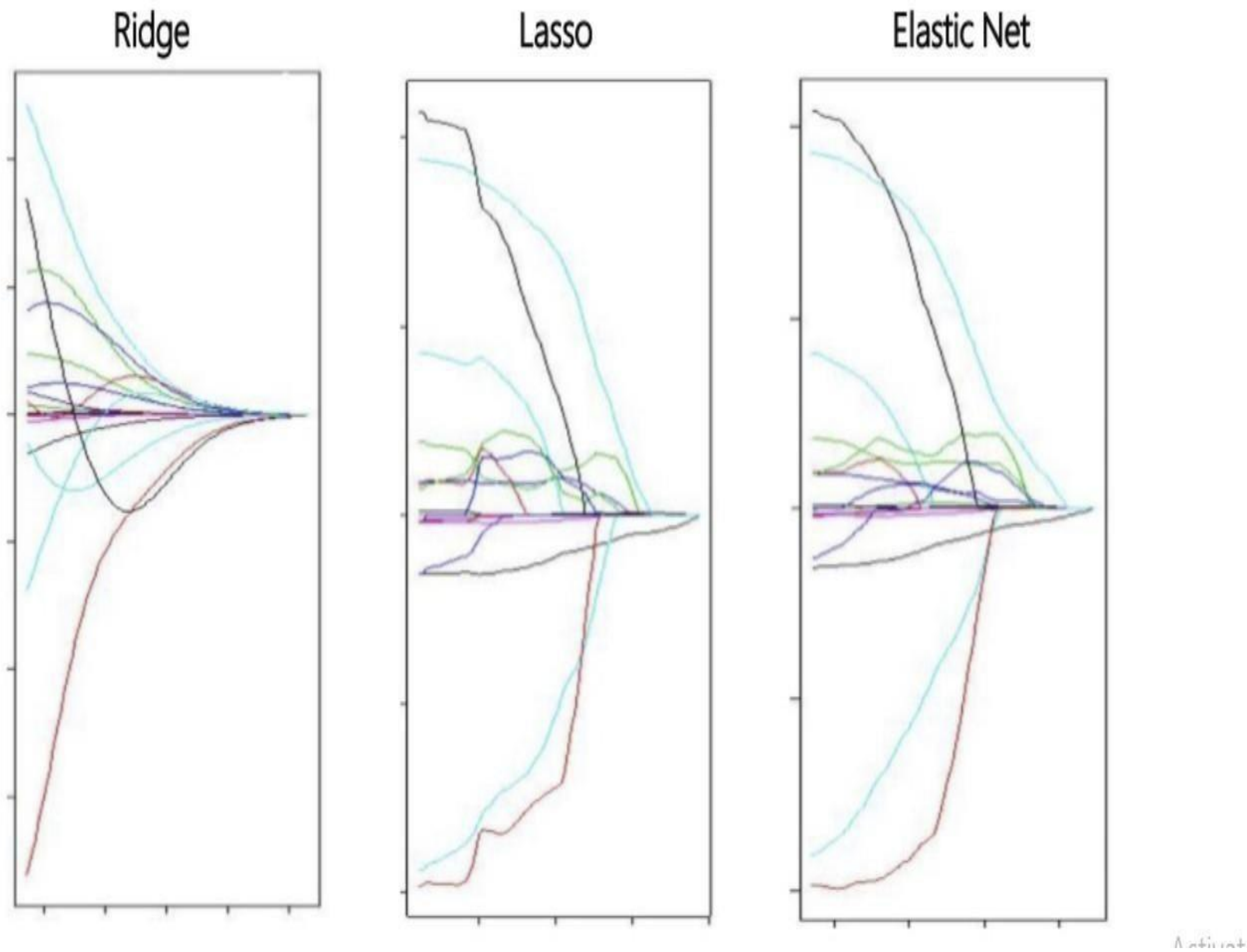


Figure No 5.3: Working of Three Regularization Techniques

Chapter 6

System Design

6.1 Architecture Diagram:

An architecture diagram is a graphical representation of a set of concepts, that are part of an architecture, including their principles, elements and components.

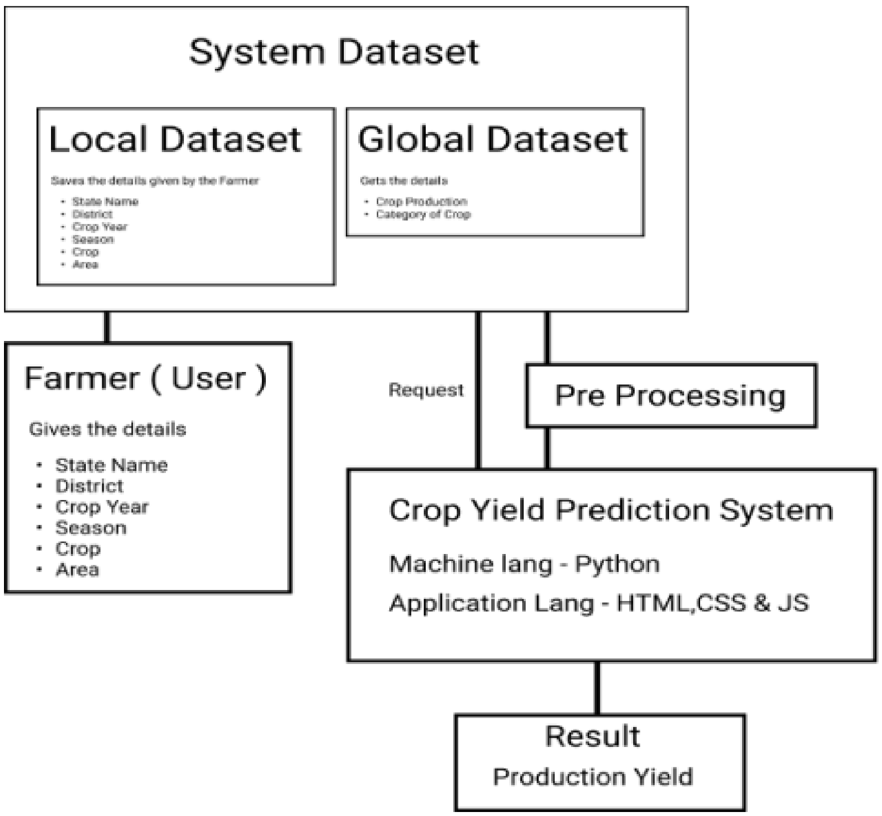


Figure No 6.1: Architecture Diagram

6.2 Use case Diagram:

Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally.

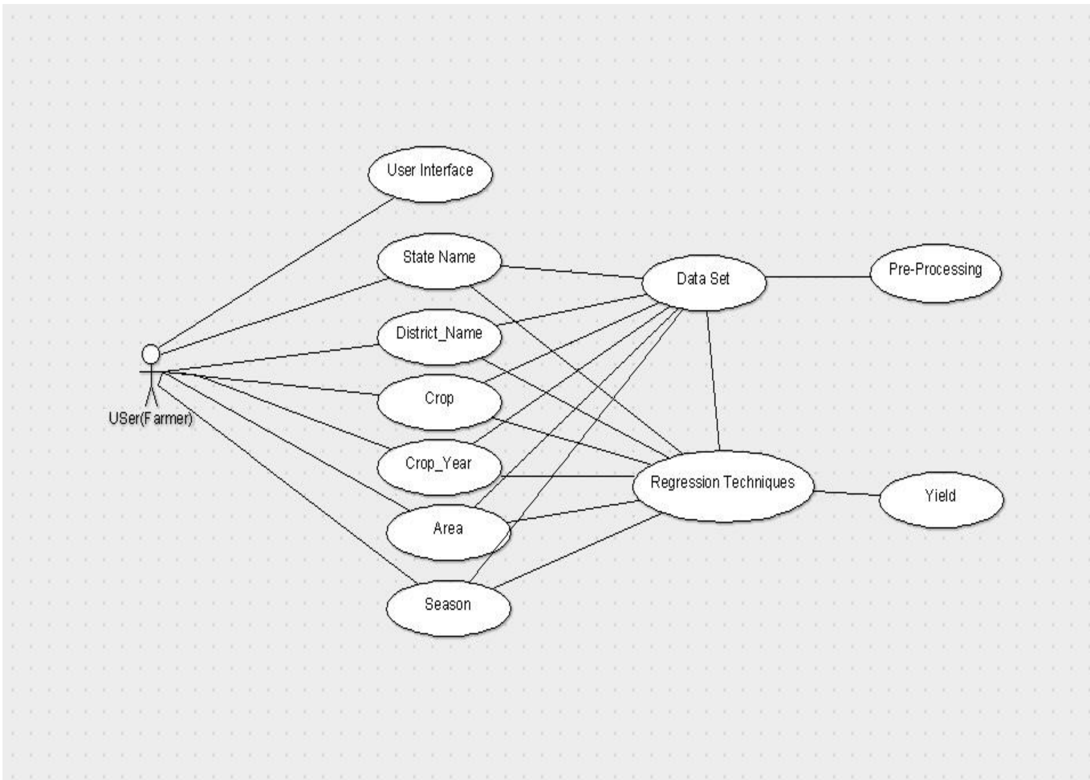


Figure No 6.2: Use case Diagram

6.3 State Diagram:

A state diagram is a type of diagram used in computer science and related fields to describe the behavior of systems. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics.

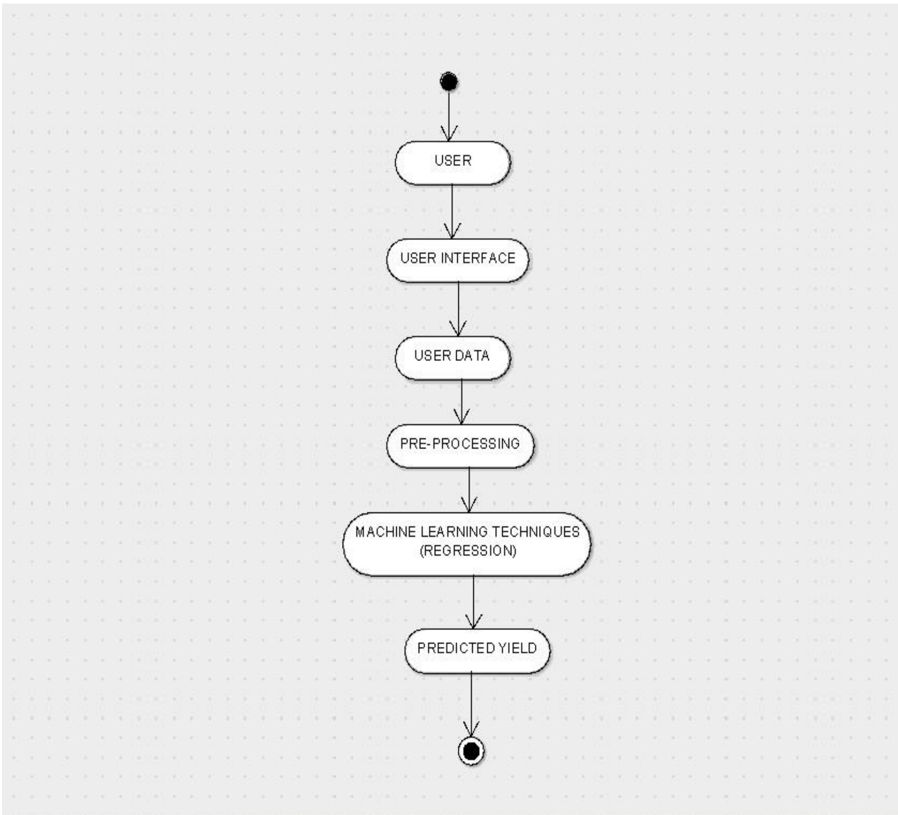


Figure No 6.3: State Diagram

Chapter 7

Project Planning

7.1 Methodology

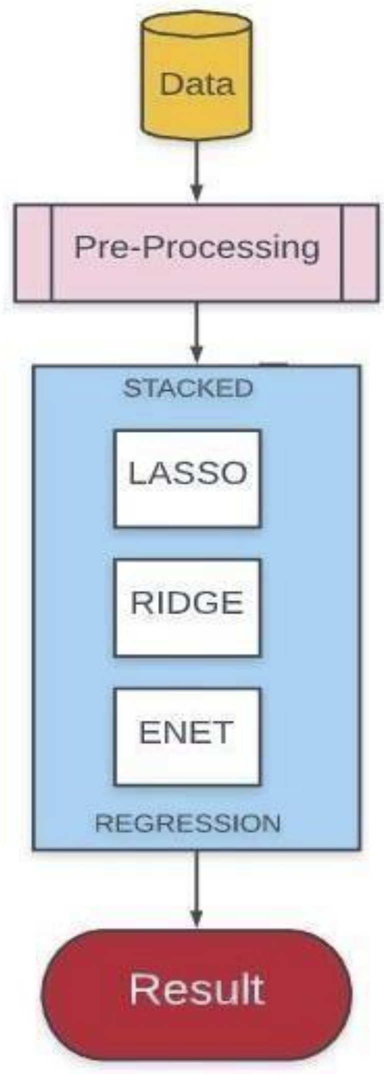


Figure No 7.1: Process chart of the research project

7.2 Pre-Processing

For the given data set, there are quite a few 'NA' values which are filtered in python. Furthermore, as the data set consists of numeric data, we used robust scaling, which is quite similar to normalization, but it instead uses the interquartile range whereas normalization is something which normalization shrinks the data in terms of 0 to 1.

7.3 Stacked Regression

This is a kind of ensembling but a little of enhancement of averaging. In this, we add a meta model and use the out of fold predictions of the other models used to train the main meta model.

Step-1: The total training set is again divided into two different sets. (train and holdout)

Step-2: Train the selected base models with first part (train).

Step-3: Test them with the second part. (holdout)

Step-4: Now, the predictions obtained from test part are inputs to the train higher level learner called meta-model.

Iteratively, the first three steps are completed. For example, if we take a 5-fold stacking, we divide the training data into 5 folds first. We will then do 5 iterations. We train each base model on 4 folds in each iteration and predict the remaining fold (holdout fold). So, after 5 iterations, we'll be confident that all the data will be used to get out - of-fold predictions that we'll use as a new feature in Step 4 to train our meta-model. We average the predictions of all base models on the test data for the predictive portion and used them as meta-features on which the meta-model is finally predicted. Here, our meta model is Lasso Regressor and that is the reason for being placed at the top. The stacked regression working can be understood from the figure 5.

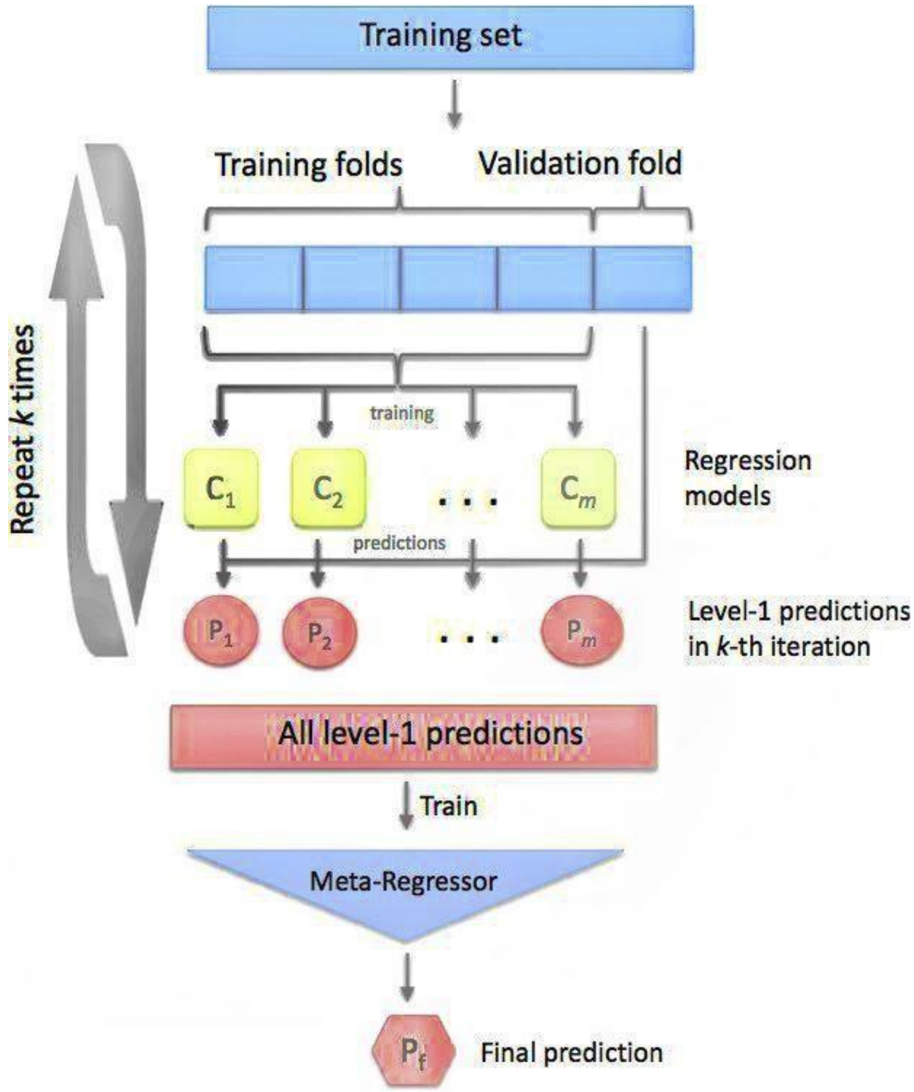
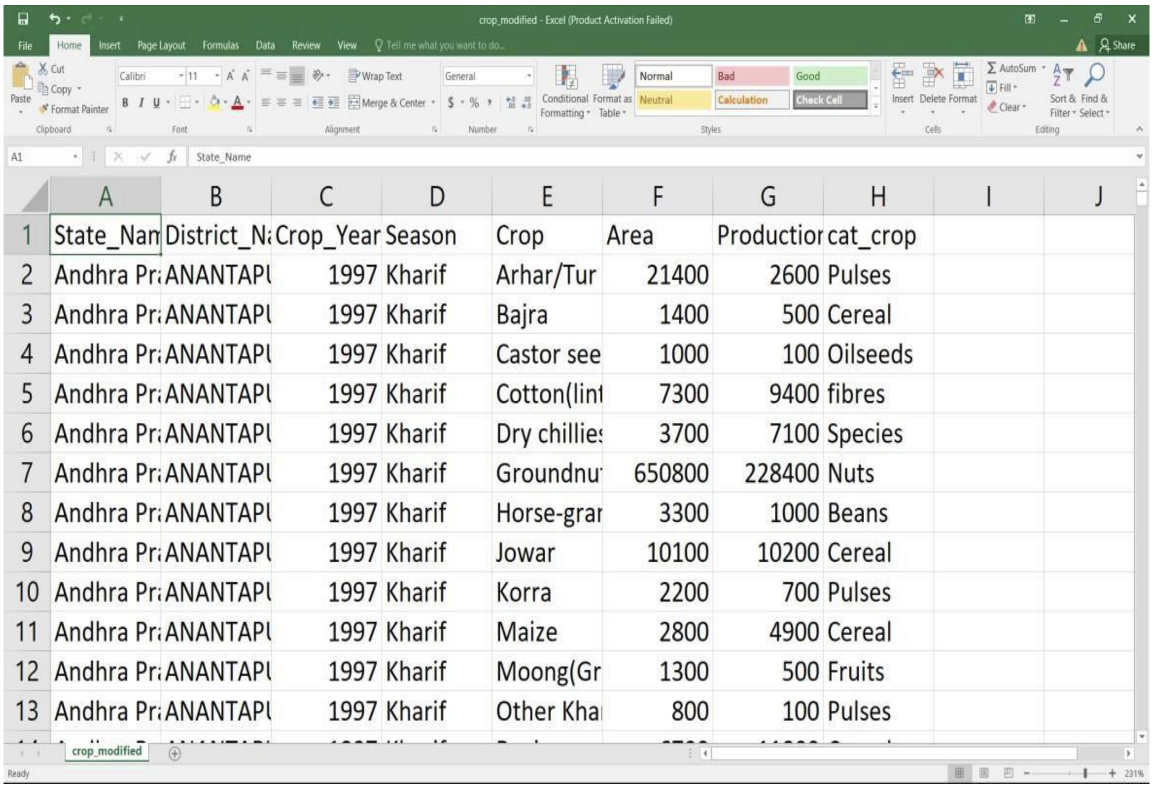


Figure No 7.2: Stacked Regression

7.4 Dataset

This dataset provides a huge amount of information on crop production in India ranging from several years. Based on the Information the ultimate goal would be to predict crop production using powerful machine learning techniques. Factors such as state and district is the farmer from, crop year, production, area which crop and in what season, with 24000 observations.



	A	B	C	D	E	F	G	H
1	State_Nam	District_N	Crop_Year	Season	Crop	Area	Productior	cat_crop
2	Andhra Pr	ANANTAPU	1997	Kharif	Arhar/Tur	21400	2600	Pulses
3	Andhra Pr	ANANTAPU	1997	Kharif	Bajra	1400	500	Cereal
4	Andhra Pr	ANANTAPU	1997	Kharif	Castor see	1000	100	Oilseeds
5	Andhra Pr	ANANTAPU	1997	Kharif	Cotton(lint	7300	9400	fibres
6	Andhra Pr	ANANTAPU	1997	Kharif	Dry chillies	3700	7100	Species
7	Andhra Pr	ANANTAPU	1997	Kharif	Groundnu	650800	228400	Nuts
8	Andhra Pr	ANANTAPU	1997	Kharif	Horse-grar	3300	1000	Beans
9	Andhra Pr	ANANTAPU	1997	Kharif	Jowar	10100	10200	Cereal
10	Andhra Pr	ANANTAPU	1997	Kharif	Korra	2200	700	Pulses
11	Andhra Pr	ANANTAPU	1997	Kharif	Maize	2800	4900	Cereal
12	Andhra Pr	ANANTAPU	1997	Kharif	Moong(Gr	1300	500	Fruits
13	Andhra Pr	ANANTAPU	1997	Kharif	Other Kha	800	100	Pulses

Figure No 7.3: Dataset

Chapter 8

Implementation

8.1 HTML code:

```
<html>
<head>
    <script>
    </script>
<title>Crop Yield Prediction System</title>
<meta name="viewport" content="width=device-width, initial-scale=1" charset="UTF-8">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<!-- Custom Theme files -->
<link href= "style.css" rel="stylesheet" type="text/css" media="all" />
<!-- //Custom Theme files -->
<!-- web font -->
<link
href="https://fonts.googleapis.com/css2?family=Source+Sans+Pro:ital,wght@0,200;0,300;0,
400;0,600;0,700;0,900;1,200;1,300;1,400;1,600;1,700;1,900&display=swap"
rel="stylesheet">
<!--<link href="//fonts.googleapis.com/css?family=Roboto:300,300i,400,400i,700,700i"
rel="stylesheet">
 //web font -->
</head>
<body>
    <!-- main -->
    <div class="main-w3layouts wrapper">
        <h1>CROP YIELD PREDICTION BASED ON INDIAN AGRICULTURE</h1>
        <div class="main-agileinfo">
            <div class="agileits-top">
                <form action="." method="post" autocomplete="off" name="one">
```

```
<select class="text" style="background-color: rgb(0, 0, 0,0.20);" type="text" name="state_name" placeholder="State Name" required="" id="state_name">
    <option disabled selected value> State Name </option>
    <option >Andhra Pradesh</option>
    <option >Arunachal Pradesh</option>
    <option >Assam</option>
    <option >Bihar</option>
    <option >Chhattisgarh</option>
    <option >Goa</option>
    <option >Gujarat</option>
    <option >Haryana</option>
    <option >Himachal Pradesh</option>
    <option >Jharkhand</option>
    <option >Karnataka</option>
    <option >Kerala</option>
    <option >Madhya Pradesh</option>
    <option >Maharashtra</option>
    <option >Manipur</option>
    <option >Meghalaya</option>
    <option >Mizoram</option>
    <option >Nagaland</option>
    <option >Odisha</option>
    <option >Punjab</option>
    <option >Rajasthan</option>
    <option >Sikkim</option>
    <option >Tamil Nadu</option>
    <option >Telangana</option>
    <option >Tripura</option>
    <option >Uttarakhand</option>
```

```
        <option >Uttar Pradesh</option>
        <option>West Bengal</option>


      </select>
      <!--<input class="text" type="text" name="state_name" placeholder="State
Name" required="" >-->
      <input class="text" type="text" name="district_name" placeholder="District
Name" required="">
      <input class="text" type="text" name="crop" placeholder="Crop" required="">
      <input class="text" type="text" name="area" placeholder="Area" required="">
      <input class="text" type="number" min="1990" max="2050" step="1"
name="year" placeholder="Year" required="">
      <!--<input class="text" type="text" name="season" placeholder="Season"
required="">-->
      <select class="text" type="text" name="season" placeholder="season"
required="" id="season">
        <option disabled selected value> season </option>
        <option >Kharif</option>
        <option >Whole Year</option>
        <option >Autumn</option>
        <option >Rabi</option>
        <option >Summer</option>
        <option >Winter</option>
      </select>
      <div class="wthree-text">
        <label class="anim">
        </label>
        <div class="clear"> </div>
      </div>
      <input type="submit" value="Predict">
    </form>
```

```
                <p>Production would be <a href="#">KGs</a></p>
                <p>Yield would be <a href="#">KGs/Sq.ft</a></p>
            </div>
        </div>
    </div>
    <!-- //main -->
</body>
</html>
```

8.2 CSS Code

```
html, body, div, span, applet, object, iframe, h1, h2, h3, h4, h5, h6, p, blockquote, pre, a,
abbr, acronym, address, big, cite, code, del, dfn, em, img, ins, kbd, q, s, samp, small, strike,
strong, sub, sup, tt, var, b, u, i, dl, dt, dd, ol, nav ul, nav li, fieldset, form, label, legend, table,
caption, tbody, tfoot, thead, tr, th, td, article, aside, canvas, details, embed, figure, figcaption,
footer, header, hgroup, menu, nav, output, ruby, section, summary, time, mark, audio, video {
  margin: 0;
  padding: 0;
  border: 0;
  font-size: 100%;
  font:inherit ;
  vertical-align: baseline;
}
article, aside, details, figcaption, figure, footer, header, hgroup, menu, nav, section {
  display: block;
}
ol, ul {
  list-style: none;
  margin: 0px;
  padding: 0px;
}
blockquote, q {
```

```
  quotes: none;
}
blockquote:before, blockquote:after, q:before, q:after {
  content: '';
  content: none;
}
table {
  border-collapse: collapse;
  border-spacing: 0;
}
/*-- start editing from here --*/
a {
  text-decoration: none;
}
.txt-rt {
  text-align: right;
}
/* text align right */
.txt-lt {
  text-align: left;
}
/* text align left */
.txt-center {
  text-align: center;
}
/* text align center */
.float-rt {
  float: right;
}
```

```
/* float right */

.float-lt {

  float: left;

}

/* float left */

.clear {

  clear: both;

}

body {
  background-image: url("crop.jpg");
  background-size: 1920px 1080px;
  background-position: center;
  background: opacity 20px;

}

h1 {
  font-size: 3em;
  text-align: center;
  color: rgb(31, 31, 31);
  font-weight: 100;
  text-transform: capitalize;
  letter-spacing: 4px;
  font-family: 'Roboto', sans-serif;
}

/*-- main --*/
.main-w3layouts {
  padding: 3em 0 1em;
}

.main-agileinfo {
  width: 35%;
  margin: 3em auto;
  background: rgba(0, 0, 0, 0.50);
  background-size: cover;
}

.agileits-top {
  padding: 3em;
}
```

```
input[type="text"], input[type="text"], input[type="text"] {
  font-size: 0.9em;
  color: rgb(0, 0, 0);
  font-weight: 100;
  width: 94.5%;
  display: block;
  border: none;
  padding: 0.8em;
  border: solid 1px rgba(0, 0, 0, 0.37);
  -webkit-transition: all 0.3s cubic-bezier(0.64, 0.09, 0.08, 1);
  transition: all 0.3s cubic-bezier(0.64, 0.09, 0.08, 1);
  background: -webkit-linear-gradient(top, rgba(0, 0, 0, 0) 96%, #fff 4%);
  background: linear-gradient(to bottom, rgba(0, 0, 0, 0) 96%, #fff 4%);
  background-position: -800px 0;
  background-size: 100%;
  background-repeat: no-repeat;
  color: rgb(0, 0, 0);
  font-family: 'Roboto', sans-serif;
}
select[type="text"], input[type="text"], input[type="text"] {
  font-size: 0.9em;
  color: rgb(246, 246, 246);
  font-weight: 100;
  width: 94.5%;
  display: block;
  border: none;
  padding: 0.8em;
  border: solid 1px rgba(0, 0, 0, 0.37);
  -webkit-transition: all 0.3s cubic-bezier(0.64, 0.09, 0.08, 1);
  transition: all 0.3s cubic-bezier(0.64, 0.09, 0.08, 1);
  background: -webkit-linear-gradient(top, rgba(255, 255, 255, 0) 96%, #fff 4%);
  background: linear-gradient(to bottom, rgba(255, 255, 255, 0) 96%, #fff 4%);
  background-position: -800px 0;
  background-size: 100%;
  background-repeat: no-repeat;
  color: rgb(255, 255, 255);
  font-family: 'Roboto', sans-serif;
}
select.text, select.text.w3lpass {
  margin: 2em 0;
}
input.text, input.text.w3lpass {
  margin: 2em 0;
}
```

```
.text:focus, .text:valid {
  box-shadow: none;
  outline: none;
  background-position: 0 0;
}

.text:focus::-webkit-input-placeholder, .text:valid::-webkit-input-placeholder {
  color: rgba(0, 0, 0, 0.7);
  font-size: .9em;
  -webkit-transform: translateY(-30px);
  -moz-transform: translateY(-30px);
  -o-transform: translateY(-30px);
  -ms-transform: translateY(-30px);
  transform: translateY(-30px);
  visibility: visible !important;
}

::-webkit-input-placeholder {
  color: #fff;
  font-weight: 100;
}

:-moz-placeholder {
  /* Firefox 18- */
  color: #fff;
}

::-moz-placeholder {
  /* Firefox 19+ */
  color: #fff;
}

:-ms-input-placeholder {
  color: #fff;
}
:-ms-select-placeholder {
  color: #fff;
}

input[type="submit"] {
  font-size: .9em;
  color: #fff;
  background: #79d1f1;
  outline: none;
  border: 1px solid #76b852;
```

```
  cursor: pointer;
  padding: 0.9em;
  -webkit-appearance: none;
  width: 100%;
  margin: 2em 0;
  letter-spacing: 4px;
}

input[type="submit"]:hover {
  -webkit-transition: .5s all;
  -moz-transition: .5s all;
  -o-transition: .5s all;
  -ms-transition: .5s all;
  transition: .5s all;
  background: #8DC26F;
}

.agileits-top p {
  font-size: 1em;
  color: rgb(0, 0, 0);
  text-align: center;
  letter-spacing: 1px;
  font-weight: 300;
}

.agileits-top p a {
  color: rgb(0, 0, 0);
  -webkit-transition: .5s all;
  -moz-transition: .5s all;
  transition: .5s all;
  font-weight: 400;
}

.agileits-top p a:hover {
  color: #76b852;
}

/*-- //main --*/
/*-- checkbox --*/
.wthree-text label {
  font-size: 0.9em;
  color: #fff;
  font-weight: 200;
  cursor: pointer;
  position: relative;
```

```
}

input.checkbox {
  background: #8DC26F;
  cursor: pointer;
  width: 1.2em;
  height: 1.2em;
}

input.checkbox:before {
  content: "";
  position: absolute;
  width: 1.2em;
  height: 1.2em;
  background: inherit;
  cursor: pointer;
}

input.checkbox:after {
  content: "";
  position: absolute;
  top: 0px;
  left: 0;
  z-index: 1;
  width: 1.2em;
  height: 1.2em;
  border: 1px solid #fff;
  -webkit-transition: .4s ease-in-out;
  -moz-transition: .4s ease-in-out;
  -o-transition: .4s ease-in-out;
  transition: .4s ease-in-out;
}

input.checkbox:checked:after {
  -webkit-transform: rotate(-45deg);
  -moz-transform: rotate(-45deg);
  -o-transform: rotate(-45deg);
  -ms-transform: rotate(-45deg);
  transform: rotate(-45deg);
  height: .5rem;
  border-color: #fff;
  border-top-color: transparent;
  border-right-color: transparent;
}
```

```
.anim input.checkbox:checked:after {
  -webkit-transform: rotate(-45deg);
  -moz-transform: rotate(-45deg);
  -o-transform: rotate(-45deg);
  -ms-transform: rotate(-45deg);
  transform: rotate(-45deg);
  height: .5rem;
  border-color: transparent;
  border-right-color: transparent;
  animation: .4s rippling .4s ease;
  animation-fill-mode: forwards;
}

@keyframes rippling {
  50% {
    border-left-color: #fff;
  }

  100% {
    border-bottom-color: #fff;
    border-left-color: #fff;
  }
}

/*-- //checkbox --*/
/*-- copyright --*/
.colorlibcopy-agile {
  margin: 2em 0 1em;
  text-align: center;
}

.colorlibcopy-agile p {
  font-size: .9em;
  color: #fff;
  line-height: 1.8em;
  letter-spacing: 1px;
  font-weight: 100;
}

.colorlibcopy-agile p a {
  color: #fff;
  transition: 0.5s all;
  -webkit-transition: 0.5s all;
  -moz-transition: 0.5s all;
  -o-transition: 0.5s all;
```

```
  -ms-transition: 0.5s all;
}

.colorlibcopy-agile p a:hover {
  color: #000;
}

/*-- //copyright --*/
.wrapper {
  position: relative;
  overflow: hidden;
}

.colorlib-bubbles {
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  z-index: -1;
}


@-webkit-keyframes square {
  0% {
    -webkit-transform: translateY(0);
    -moz-transform: translateY(0);
    -o-transform: translateY(0);
    -ms-transform: translateY(0);
    transform: translateY(0);
  }

  100% {
    -webkit-transform: translateY(-700px) rotate(600deg);
    -moz-transform: translateY(-700px) rotate(600deg);
    -o-transform: translateY(-700px) rotate(600deg);
    -ms-transform: translateY(-700px) rotate(600deg);
    transform: translateY(-700px) rotate(600deg);
  }
}

@keyframes square {
  0% {
    -webkit-transform: translateY(0);
    -moz-transform: translateY(0);
```

```
    -o-transform: translateY(0);
    -ms-transform: translateY(0);
    transform: translateY(0);
  }

  100% {
    -webkit-transform: translateY(-700px) rotate(600deg);
    -moz-transform: translateY(-700px) rotate(600deg);
    -o-transform: translateY(-700px) rotate(600deg);
    -ms-transform: translateY(-700px) rotate(600deg);
    transform: translateY(-700px) rotate(600deg);
  }
}

/*-- responsive-design --*/
@media(max-width:1440px) {
  input[type="text"], input[type="email"], input[type="password"] {
    width: 94%;
  }
}

@media(max-width:1366px) {
  h1 {
    font-size: 2.6em;
  }

  .agileits-top {
    padding: 2.5em;
  }

  .main-agileinfo {
    margin: 2em auto;
  }

  .main-agileinfo {
    width: 36%;
  }
}

@media(max-width:1280px) {
  .main-agileinfo {
    width: 40%;
  }
}
```

```
@media(max-width:1080px) {
  .main-agileinfo {
    width: 46%;
  }
}

@media(max-width:1024px) {
  .main-agileinfo {
    width: 49%;
  }
}

@media(max-width:991px) {
  h1 {
    font-size: 2.4em;
  }

  .main-w3layouts {
    padding: 2em 0 1em;
  }
}

@media(max-width:900px) {
  .main-agileinfo {
    width: 58%;
  }

  input[type="text"], input[type="email"], input[type="password"] {
    width: 93%;
  }
}

@media(max-width:800px) {
  h1 {
    font-size: 2.2em;
  }
}

@media(max-width:736px) {
  .main-agileinfo {
    width: 62%;
  }
}

@media(max-width:667px) {
```

```
  .main-agileinfo {
    width: 67%;
  }
}

@media(max-width:600px) {
  .agileits-top {
    padding: 2.2em;
  }

  input.email, input.text.w3lpass {
    margin: 1.5em 0;
  }
  select.email, select.text.w3lpass {
    margin: 1.5em 0;
  }
  input[type="submit"] {
    margin: 2em 0;
  }
}
```

8.3 Python Code

```
from flask import Flask, render_template, request
from sklearn.model_selection import KFold

app = Flask(__name__)
@app.route('/')
def index():
    return render_template("crop.html")
@app.route('/', methods=['post'])
def getvalue():
    try:

        state = request.form['state_name']
        district = request.form['district_name']
        district = district.upper()
        crop = request.form['crop']
        season = request.form['season']
        area = request.form['area']
        area_float = float(area)
        year = request.form['year']
        year_int = int(year)
        import pandas as pd
        import numpy as np
```

```
from sklearn.kernel_ridge import KernelRidge
from sklearn.linear_model import ElasticNet, Lasso, BayesianRidge, LassoLarsIC
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import RobustScaler
from sklearn.base import BaseEstimator, TransformerMixin, RegressorMixin, clone
from sklearn.ensemble import RandomForestRegressor
import os
# os.chdir(r"C:\Users\HP\OneDrive\Desktop\crop project")
crop_data = pd.read_csv("crop_modified.csv")
crop_data = crop_data.dropna()
crop_data['State_Name'] = crop_data['State_Name'].str.rstrip()
crop_data['Season'] = crop_data['Season'].str.rstrip()
a = crop_data[crop_data['State_Name'] == state]
b = a[a['District_Name'] == district]
c = b[b['Season'] == season]
f = c[c['Crop'] == crop]['Crop_Year']
x = c[c['Crop'] == crop]['Area']
y = c[c['Crop'] == crop]['Production']
from pandas import DataFrame
variables = {'Crop_Year': f, 'Area': x, 'Production': y}
final = DataFrame(variables, columns=['Crop_Year', 'Area', 'Production'])
X = final[['Crop_Year', 'Area']]
Y = final['Production']

class StackingAveragedModels(BaseEstimator, RegressorMixin, TransformerMixin):
    def __init__(self, base_models, meta_model, n_folds=5):
        self.base_models = base_models
        self.meta_model = meta_model
        self.n_folds = n_folds

    # We again fit the data on clones of the original models
    def fit(self, X, y):
        self.base_models_ = [list() for x in self.base_models]
        self.meta_model_ = clone(self.meta_model)
        kfold = KFold(n_splits=self.n_folds, shuffle=True, random_state=156)

        # Train cloned base models then create out-of-fold predictions
        # that are needed to train the cloned meta-model
        out_of_fold_predictions = np.zeros((X.shape[0], len(self.base_models)))
        for i, model in enumerate(self.base_models):
            for train_index, holdout_index in kfold.split(X, y):
                print(X.columns)
                instance = clone(model)
```

```
                self.base_models_[i].append(instance)
                instance.fit(X[train_index], y[train_index])
                y_pred = instance.predict(X[holdout_index])
                out_of_fold_predictions[holdout_index, i] = y_pred

        # Now train the cloned  meta-model using the out-of-fold predictions as new feature
        self.meta_model_.fit(out_of_fold_predictions, y)
        return self

    # Do the predictions of all base models on the test data and use the averaged predictions

    # meta-features for the final prediction which is done by the meta-model
    def predict(self, X):
        meta_features = np.column_stack([
            np.column_stack([model.predict(X) for model in base_models]).mean(axis=1)
            for base_models in self.base_models_])
        return self.meta_model_.predict(meta_features)

class StackedAveragingModels(BaseEstimator, RegressorMixin, TransformerMixin):
    def __init__(self, models):
        self.models = models

    # we define clones of the original models to fit the data in
    def fit(self, X, y):
        self.models_ = [clone(x) for x in self.models]

        # Train cloned base models
        for model in self.models_:
            model.fit(X, y)

        return self

    # Now we do the predictions for cloned models and average them
    def predict(self, X):
        predictions = np.column_stack([
            model.predict(X) for model in self.models_
        ])
        return np.mean(predictions, axis=1)

lasso = make_pipeline(RobustScaler(), Lasso(alpha=0.0005, random_state=1))
ENet = make_pipeline(RobustScaler(), ElasticNet(alpha=0.0005, l1_ratio=.9,
    random_state=3))
KRR = KernelRidge(alpha=0.6, kernel='polynomial', degree=2, coef0=2.5)
# from mlxtend.regressor import StackingRegressor
```

```
    # stack = StackingRegressor(regressors=[ENet, KRR],  meta_regressor=lasso)
    # model1=stack.fit(X,Y)
    # prod2 = model.predict([[year_int, area_float]])
    averaged_models = StackedAveragingModels(models=(KRR, lasso))

    # import pickle
    # pickle.dump(averaged_models,open('model.pkl','wb'))
    # model = pickle.load(open('model.pkl','rb'))
    model = averaged_models.fit(X, Y)
    prod2 = model.predict([[year_int, area_float]])
    prod2 = abs(prod2)
    print("Prediction is: ", prod2)
    yld = prod2 / area_float
    return render_template("crop.html", pr=prod2, yl=yld)
except Exception as e:
    return render_template("crop.html",err=e)



if __name__ == "__main__":

    app.run()
```

Chapter 9

Result Analysis

9.1 Comparing Regressions Techniques

The performance metric used in this project is Root mean square error. When the models applied individually, for ENet it was around 4%, Lasso had an error about 2%, Kernel Ridge was about 1% and finally after stacking it was less than 1%. The user or the farmer can enter the following details over the web application to get the prediction.

	Lasso	**ENet**	**KRR**	**Lasso-ENet-KRR**
1	0.45567507	0.45575077	0.37857143	0.41712325
2	0.2651388	0.26489328	0.13	0.1975694
3	0.87944585	0.87944408	0.81585616	0.847651
4	2.42507588	2.42506197	2.36972973	2.39740281

Table 9.1: Yield values for different Algorithms used

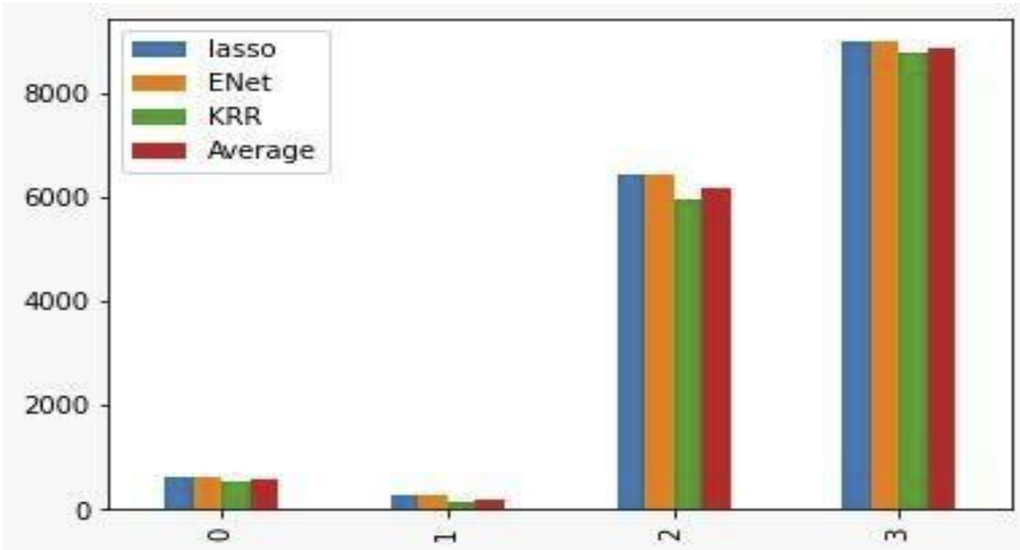


Figure No 9.1: Plotted graph for the above table

9.2 Outputs

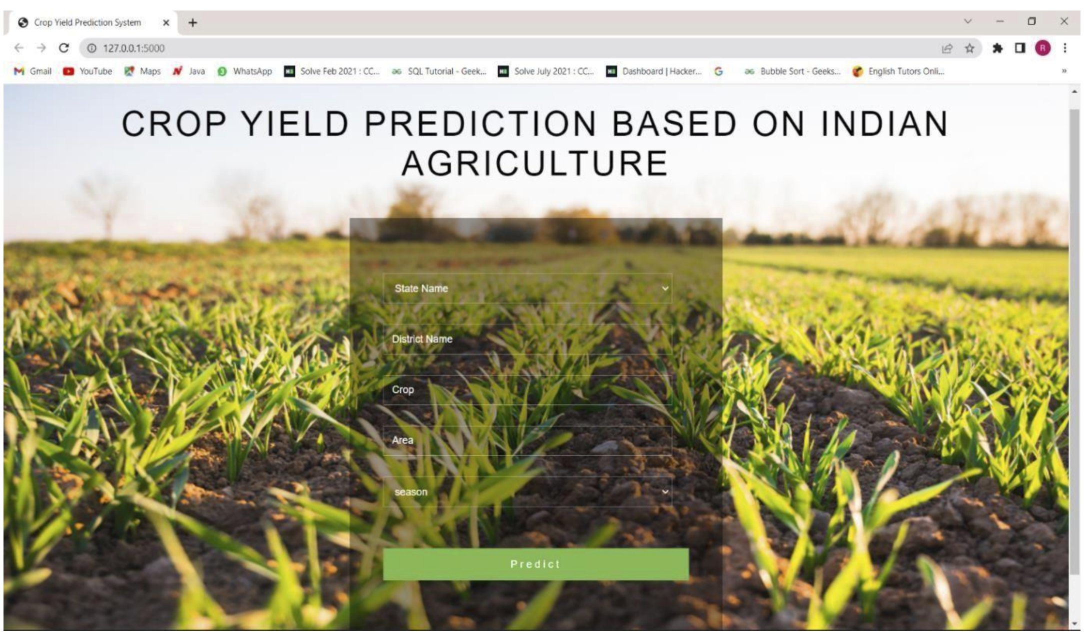


Figure No 9.2: User Interface

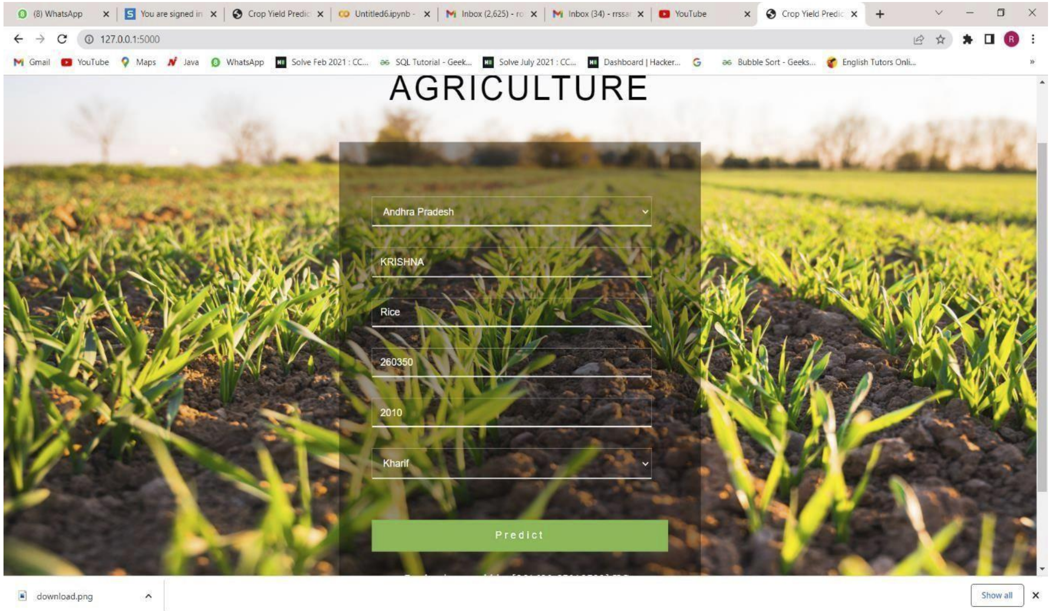


Figure No 9.3: User Input Data

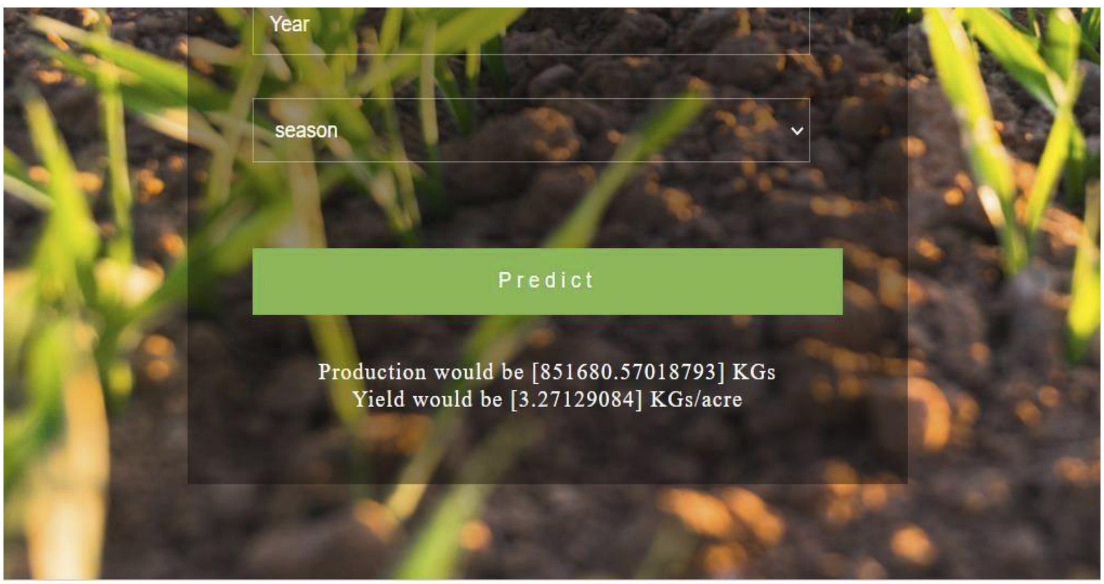


Figure No 9.4: Predicted Result

Chapter 10

Conclusion and Future Scope

When we apply stacked regression, the result has been so improvised than when those models were applied individually. The performance metric used in this project is Root mean square error. When the models applied individually, for ENet it was around 4%, Lasso had an error about 2%, Kernel Ridge was about 1% and finally after stacking it was less than 1%. The user or the farmer can enter the following details over the web application to get the prediction as depicted in the above output. The output which has been shown in figure is currently a web application. In this project we have used the method of stacked regression. Results were calculated when the models were applied individually as well as stacked regression. The results were found to be better than when the models were used separately. The output which has been shown in figure is currently a web application, but our future work would be building an application where the farmers can use it as app and converting the whole system in their regional language.

References

[1] “data.gov.in.” [Online]. Available: https://data.gov.in/

[2] Ananthara, M. G., Arunkumar, T., & Hemavathy, R. (2013, February). CRY—an improved crop yield prediction model using bee hive clustering approach for agricultural data sets. In 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering (pp. 473-478). IEEE.

[3] Awan, A. M., & Sap, M. N. M. (2006, April). An intelligent system based on kernel methods for crop yield prediction. In Pacific-Asia Conference on Knowledge Discovery and Data Mining (pp. 841-846). Springer, Berlin, Heidelberg.

[4] Bang, S., Bishnoi, R., Chauhan, A. S., Dixit, A. K., & Chawla, I. (2019, August). Fuzzy Logic based Crop Yield Prediction using Temperature and Rainfall parameters predicted through ARMA, SARIMA, and ARMAX models. In 2019 Twelfth International Conference on Contemporary Computing (IC3) (pp. 1-6). IEEE.

[5] Bhosale, S. V., Thombare, R. A., Dhemey, P. G., & Chaudhari, A. N. (2018, August). Crop Yield Prediction Using Data Analytics and Hybrid Approach. In 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA) (pp. 1-5). IEEE.

[6] Gandge, Y. (2017, December). A study on various data mining techniques for crop yield prediction. In 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT) (pp. 420-423). IEEE.

[7] Gandhi, N., Petkar, O., & Armstrong, L. J. (2016, July). Rice crop yield prediction using artificial neural networks. In 2016 IEEE Technological Innovations in ICT for Agriculture and Rural Development (TIAR) (pp. 105-110). IEEE.

[8] Gandhi, N., Armstrong, L. J., Petkar, O., & Tripathy, A. K. (2016, July). Rice crop yield prediction in India using support vector machines. In 2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE) (pp. 1-5). IEEE.

[9] Gandhi, N., Armstrong, L. J., & Petkar, O. (2016, July). Proposed decision support system (DSS) for Indian rice crop yield prediction. In 2016 IEEE Technological

Innovations in ICT for Agriculture and Rural Development (TIAR) (pp. 13-18). IEEE.

[10] Islam, T., Chisty, T. A., & Chakrabarty, A. (2018, December). A Deep Neural Network Approach for Crop Selection and Yield Prediction in Bangladesh. In 2018 IEEE Region 10 Humanitarian Technology Conference (R10-HTC) (pp. 1-6). IEEE

[11] Jaikla, R., Auephanwiriyakul, S., & Jintrawet, A. (2008, May). Rice yield prediction using a support vector regression method. In 2008 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology(Vol. 1, pp. 29-32). IEEE

[12] Kadir, M. K. A., Ayob, M. Z., & Miniappan, N. (2014, August). Wheat yield prediction: Artificialneural network based approach. In 2014 4th International Conference on Engineering Technology and Technopreneuship (ICE2T) (pp. 161-165). IEEE.

[13] Manjula, A., & Narsimha, G. (2015, January). XCYPF: A flexible and extensible framework for agricultural Crop Yield Prediction. In 2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO) (pp. 1-5). IEEE

[14] Mariappan, A. K., & Das, J. A. B. (2017, April). A paradigm for rice yield prediction in Tamilnadu. In 2017 IEEE Technological Innovationsin ICT for Agriculture and Rural Development (TIAR) (pp. 18-21) IEEE.

[15] Paul, M., Vishwakarma, S. K., & Verma, A. (2015, December). Analysis of soil behaviour and prediction of crop yield using data mining approach. In 2015 International Conference on Computational Intelligence and Communication Networks (CICN) (pp. 766-771). IEEE.

[16] Shah, A., Dubey, A., Hemnani, V., Gala, D., & Kalbande, D. R. (2018). Smart Farming System: Crop Yield Prediction Using Regression Techniques. In Proceedings of International Conference on Wireless Communication (pp. 49-56). Springer, Singapore.

[17] Ahamed, A. M. S., Mahmood, N. T., Hossain, N., Kabir, M. T., Das, K., Rahman, F., & Rahman, R. M. (2015, June). Applying data mining techniques to predict annual yield of major crops and recommend planting different crops in different districts in Bangladesh. In 2015 IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD) (pp. 1-6). IEEE

[18] Shastry, A., Sanjay, H. A., & Hegde, M. (2015, June). A parameter based ANFIS model for crop yield prediction. In 2015 IEEE International Advance Computing Conference (IACC) (pp. 253- 257). IEEE

[19] Sujatha, R., & Isakki, P. (2016, January). A study on crop yield forecasting using classification techniques. In 2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16) (pp. 1-4). IEEE.

[20] Suresh, A., Kumar, P. G., & Ramalatha, M. (2018, October). Prediction of major crop yields of Tamilnadu using K-means and Modified KNN. In 2018 3rd International Conference on Communication and Electronics Syst ems (ICCES) (pp. 88-93). IEEE

[21] Veenadhari, S., Misra, B., & Singh, C. D. (2014, January). Machine learning approach for forecasting crop yield based on climatic parameters. In 2014 International Conference on Computer Communication and Informatics (pp. 1-5). IEEE.