Agricultural analysis and prediction

Project Report

SUPERVISOR: Mahesh Kumar Bhandari

SUBMITTED BY

Khushi Garg 20001570019

Garvit 20001570019

Purbak Sengupta 20001570039

Nishant Giri 20001570035

2023

Department of Computer Science

ACHARYA NARENDRA DEV COLLEGE

ACKNOWLEDGEMENT

It is a great pleasure for us to express respect and deep sense of gratitude to our supervisor Mr. Mahesh Kumar Bhandari, Associate Professor, Department of Computer Science, Acharya Narendra Dev College, as a convener of this course for his wisdom, vision, expertise, guidance, enthusiastic involvement and persistent encouragement during the planning and development of this work.

We also gratefully acknowledge his painstaking efforts in thoroughly going through and improving the manuscripts without which this work could not have been completed. We are also obliged to our parents for their moral support, love, encouragement, and blessings to complete this task.

Finally, we are indebted and grateful to the Almighty for helping us in this endeavour.

Khushi Garg

Garvit

Purbak Sengupta

Nishant Giri

ACHARYA NARENDRA DEV COLLEGE

(UNIVERSITY OF DELHI)

CERTIFICATE

This is to certify that the work contained in this report entitled **"Agricultural Analysis and Prediction"** *being submitted by* **Khushi Garg (20001570052), Garvit (20001570019),Purbak Sengupta (20001570039) and Nishant Giri (20001570035)** *carried out in the Department of Computer Science, Acharya Narendra Dev College, University of Delhi, is a bona fide work of my supervision.*

Supervisor

Mahesh Kumar Bhandari

Table of Contents

Topic **Page No**

Chapter 1

PROBLEM STATEMENT

This project will analyze the agriculture data and find optimal parameters to maximize the crop production using data mining techniques like DBSCAN and Decision Tree Regression, K-Means. The dataset consists of features like year, District, crop, season, area, production (in tons), nitrogen(kg/Ha), phosphorus (Kg/Ha), Potassium (Kg/Ha) etc. The major goal of the proposed system is understanding data mining techniques and applying it to the dataset.

This project uses several data mining techniques to extract information from agriculture data and to give suggestions regarding crops and make future predictions so that agriculture

can be carried out in a planned manner. The objectives of the project include1. Finding trends in crops in terms of production, area, etc. over the years and studying

the reasons behind the changing trends.

2. Finding how different factors that affect production are related to each other.

3. Prediction of suicide rate of the farmers.

4. Study of crops that do not follow the general trends and show an abnormal trend such

as reduction in production.

5. Finding similar crops and similar states based on various factors.

6. Predictions of crops that might be rarely produced and the main crops that might be preferred by the farmers.

Chapter 2
DATA MINING TECHNIQUES

2.1 DM Techniques

2.1.1 Clustering

Clustering refers to the process of grouping a series of different data points based on their characteristics. By doing so, data miners can seamlessly divide the data into subsets, allowing for more informed decisions in terms of broad demographics (such as consumers or users) and their respective behaviours.

Methods for **Data Clustering**

Partitioning method: This involves dividing a data set into a group of specific clusters for evaluation based on the criteria of each individual cluster. In this method, data points belong to just one group or cluster.

Hierarchical method: With the hierarchical method, data points are a single cluster, which are grouped based on similarities. These newly created clusters can then be analysed separately from each other.

Density-based method: A machine learning method where data points plotted together are further analysed, but data points by themselves are labelled “noise” and discarded.

Grid-based method: This involves dividing data into cells on a grid, which then can be clustered by individual cells rather than by the entire database. As a result, grid-based clustering has a fast-processing time.

Model-based method: In this method, models are created for each data cluster to locate the best data to fit that model.

2.1.2 Linear Regression

Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and

independent variables they are considering, and the number of independent variables getting used.

2.1.3 Decision Tree

Decision Tree is one of the most used, practical approaches for supervised learning. It can be used to solve both Regression and Classification tasks with the latter being put more into practical application. It is a tree-structured classifier with three types of nodes. The Root Node is the initial node which represents the entire sample and may get split further into further nodes.

The Interior Nodes represent the features of a data set and the branches represent the decision rules. Finally, the Leaf Nodes represent the outcome. This algorithm is very useful for solving decision-related problems.

Decision tree regression observes features of an object and trains a model in the structure of a tree to predict data in the future to produce meaningful continuous output. Continuous output means that the output/result is not discrete, i.e., it is not represented just by a discrete, known set of numbers or values.

Methods :-

K-nearest neighbours (KNN): This is an algorithm that tries to identify an unknown object by comparing it to others. For instance, grocery chains might use the K-nearest neighbors algorithm to decide whether to include a sushi or hot meals station in their new store layout based on consumer habits in the local marketplace.

Naive Bayes: Based on the Bayes Theorem of Probability, this algorithm uses historical data to predict whether similar events will occur based on a different set of data.

2.2 Data Mining Technique for This Project

2.2.1 K-Means Clustering

K-Means Clustering is an Unsupervised Learning algorithm, It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabelled dataset on its own without the need for any training.

It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters. The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

The k-means clustering algorithm mainly performs two tasks:

1) Determines the best value for K center points or centroids by an iterative process.

2) Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

Hence each cluster has datapoints with some commonalities, and it is away from other clusters.

2.2.2 Tree Based Methods

Decision Trees are tree-like structures formed from simple decision rules by analysing the data. Decision trees are supervised learning algorithms. These can be used on both classification as well as regression tasks and hence the name CART (Classification And Regression Trees). Unlike other supervised algorithms such as Logistic or Linear Regression, the decision trees are non-parametric, so training/fitting steps do not involve 'learning' the parameters. Instead, some simple decision-based rules are learnt and arranged in a hierarchical fashion to help in classification/regression tasks. Decision trees have an advantage over other algorithms that they can be visually quite intuitive to understand on what factors the target variable is being decided - which features contribute to what extent in this decision process.

Our approach involves the use of a classification decision tree, in which the decision rules to test are located in the nodes of the tree. Based on the result of the decision node, a particular branch at that node is chosen which leads to some other node other tree with some other decision to make. This process is continued in a recursive fashion until the leaves of the tree are reached at last which contains a class label. This class label is attributed to the data item.

2.2.3 Linear Regression

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

Linear regression can be further divided into two types of the algorithm:

Simple Linear Regression:

If a single independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Simple Linear Regression.

Multiple Linear regression:

If more than one independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Multiple Linear Regression.

Chapter 3

DATASET DISCRIPTION

3.1 DATASET

	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production
0	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Arecanut	1254.0	2000.0
1	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Other Kharif pulses	2.0	1.0
2	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Rice	102.0	321.0
3	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Banana	176.0	641.0
4	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Cashewnut	720.0	165.0

We have used agriculture data from:

1. Crop Production Statistics

2. Data gov crops related data

The data bases we have used are:

1) Crop Production Statistics : Contains crop production info from 2000 to 2014.

```
State_Name,District_Name,Crop_Year,Season,Crop,Area,Production
Andaman and Nicobar Islands,NICOBARS,2000,Kharif     ,Arecanut,1254.00,2000.00
Andaman and Nicobar Islands,NICOBARS,2000,Kharif     ,Other Kharif pulses,2.00,1.00
Andaman and Nicobar Islands,NICOBARS,2000,Kharif     ,Rice,102.00,321.00
Andaman and Nicobar Islands,NICOBARS,2000,Whole Year ,Banana,176.00,641.00
Andaman and Nicobar Islands,NICOBARS,2000,Whole Year ,Cashewnut,720.00,165.00
Andaman and Nicobar Islands,NICOBARS,2000,Whole Year ,Coconut ,18168.00,65100000.00
Andaman and Nicobar Islands,NICOBARS,2000,Whole Year ,Dry ginger,36.00,100.00
```

2) Crop Prices : Year wise price change of crops till 2013.

	Commodities	2002	2004	2005	2006	2007	2008	2009	2010	2011	2012
0	Paddy (Common)	550.0	560.0	570.0	580.0	645.0	850.0	950.0	1000.0	1080	1250.0
1	Paddy (Grade 'A')	580.0	590.0	600.0	610.0	675.0	880.0	980.0	1030.0	1110	1280.0
2	Wheat	630.0	640.0	650.0	750.0	1000.0	1080.0	1100.0	1120.0	1285	1350.0
3	Jowar (Hybrid)	490.0	515.0	525.0	540.0	600.0	840.0	840.0	880.0	980	1500.0
4	Jowar (Maldandi)	0.0	0.0	0.0	555.0	620.0	860.0	860.0	900.0	1000	1520.0

3) Crop Cultivation Area : Area of land a crop is produced on year by year for major crops from 2000 to 2009.

	Year	Rice	Jowar	Bajra	Maize	Ragi	Small millets	Wheat	Barley	Gram	...	Rubber	Banana	Sugarcane	Tobacco	Potato	Black pepper	Dry chillies	Ginger	Coconut
0	2000	44712	9856	9829	6611	1759	1424	25731	778	5185	...	400	459	4316	262	1222	214	836	86	1824
1	2001	44904	9795	9529	6582	1647	1310	26345	660	6416	...	401	489	4412	348	1208	219	880	91	1932
2	2002	41176	9300	7740	6635	1415	1201	25196	702	5906	...	408	460	4520	327	1345	224	827	88	1922
3	2003	42593	9331	10612	7343	1666	1191	26595	657	7048	...	428	391	3938	370	1289	233	774	85	1934
4	2004	41907	9092	8233	7430	1553	1101	26383	616	6715	...	440	404	3662	366	1318	228	738	95	1935

4) Crop Cultivation Cost : State wise cost of cultivation of crops per hectare and per quintal.

	Crop	State/ UT Name	Cost of Cultivation (`/Hectare) - A2+FL - 2008-09	Cost of Cultivation (`/Hectare) - A2+FL - 2009-10	Cost of Cultivation (`/Hectare) - A2+FL - 2010-11	Cost of Cultivation (`/Hectare) - A2+FL - 2011-12	Cost of Cultivation (`/Hectare) - A2+FL - 2012-13	Cost of Cultivation (`/Hectare) - A2+FL - 2013-14	Cost of Cultivation(`/Hectare) - C2 - 2008-09	Cost of Cultivation(`/Hectare) - C2 - 2009-10	Cost of Cultivation(`/Hectare) - C2 - 2010-11	Cul
0	Paddy	Andhra Pradesh	29664.84	35104.80	35090.78	37946.69	42669.85	46781.05	46450.20	54202.54	51505.34	
1	Paddy	Odisha	17478.05	19175.75	21894.98	27243.15	31723.25	35569.85	25909.05	28143.88	30318.40	
2	Paddy	Punjab	25154.75	29031.73	30793.25	31248.72	37103.96	39686.81	45291.24	50650.21	51279.34	
3	Paddy	Uttar Pradesh	17022.00	21336.61	21281.30	28147.45	29436.98	30982.85	28144.50	32327.78	32299.35	
4	Paddy	West Bengal	24731.06	28101.85	32872.72	37959.78	42770.22	45783.94	33046.12	38111.55	43019.85	

5) Mean Temperatures : Data of mean temperature from 2000-2012 for whole year and over interval of two months. This is used to determine effect of temperature on various crops.

6) Rainfall Statistics : State wise rainfall statics from year 2000-2015 annually and monthly in millimeter per square meter(area).

7) Suicide Statistics , Exports, Crops Growth Rate, etc.

3.2 DATASET VISUALIZATION

Data Visualization is an important part of Data Mining. It helps us get visual insights about the dataset, such as some striking features or patterns in the dataset, which can help us choose the appropriate machine learning algorithms to apply.

3.2.1 Crop Price Visualization

We first plotted a image from two dimensional numpy array of crop prices of various crops in different year in Rs/quintal.

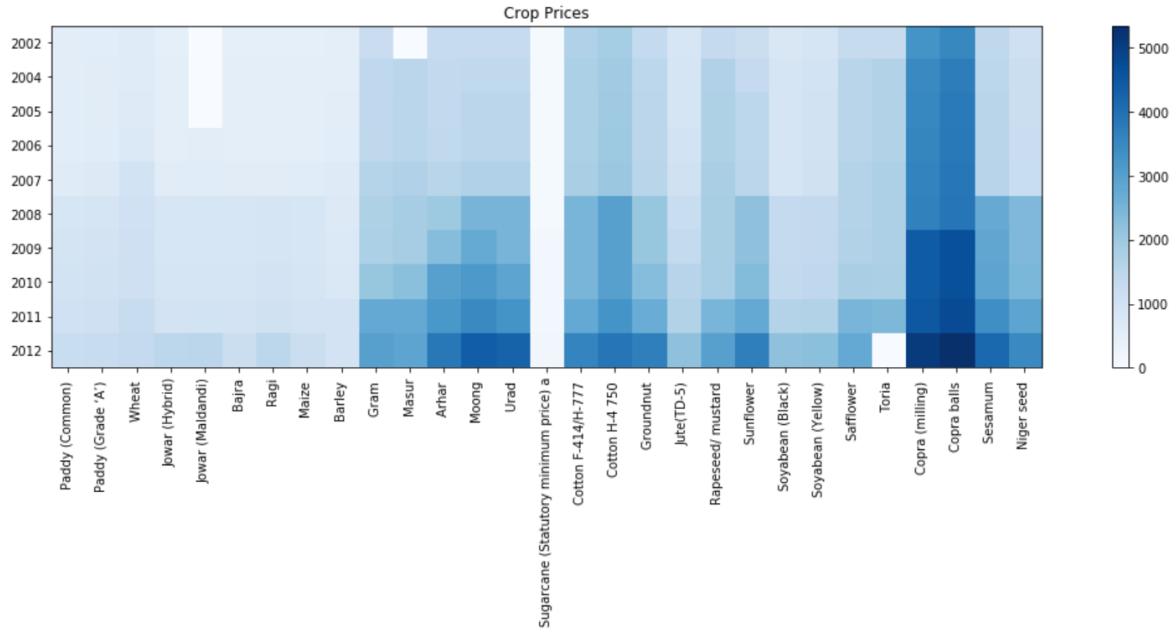

Figure 5: Crop price of various crops in different year in Rs/quintal

3.2.2 Visualization of cultivation area

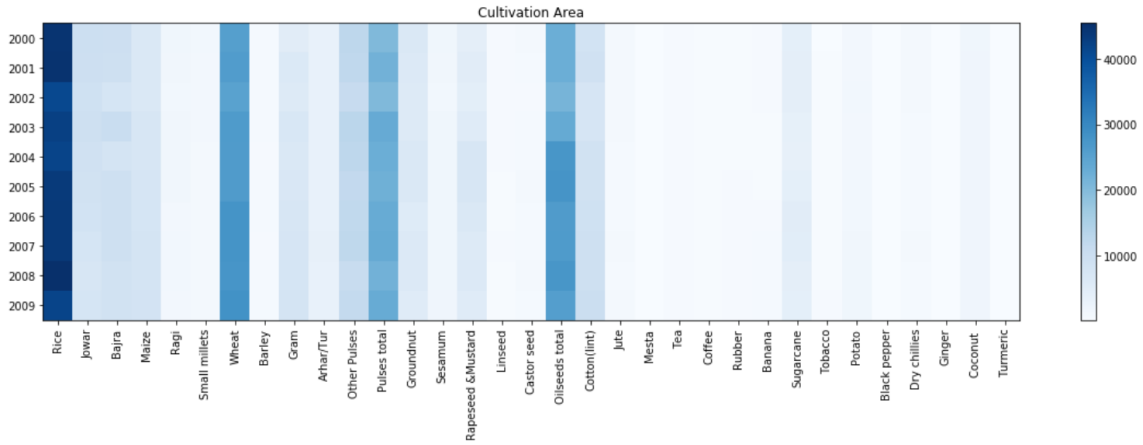

Figure 6: Cultivation area of various crops in hectares

3.2.3 Cultivation Cost of Major Crops

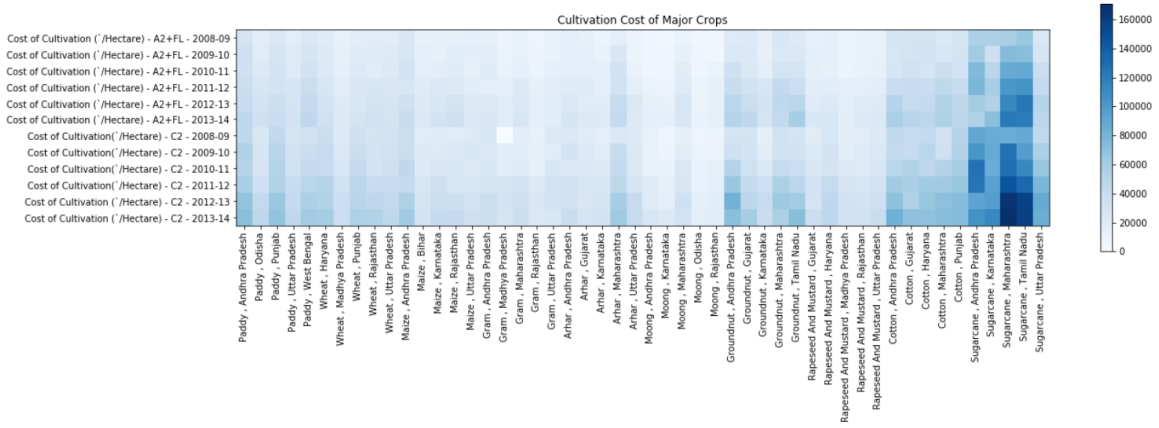

Figure 7: Cultivation cost by quintal of major crops in respective states

3.2.4 Temperature Variations

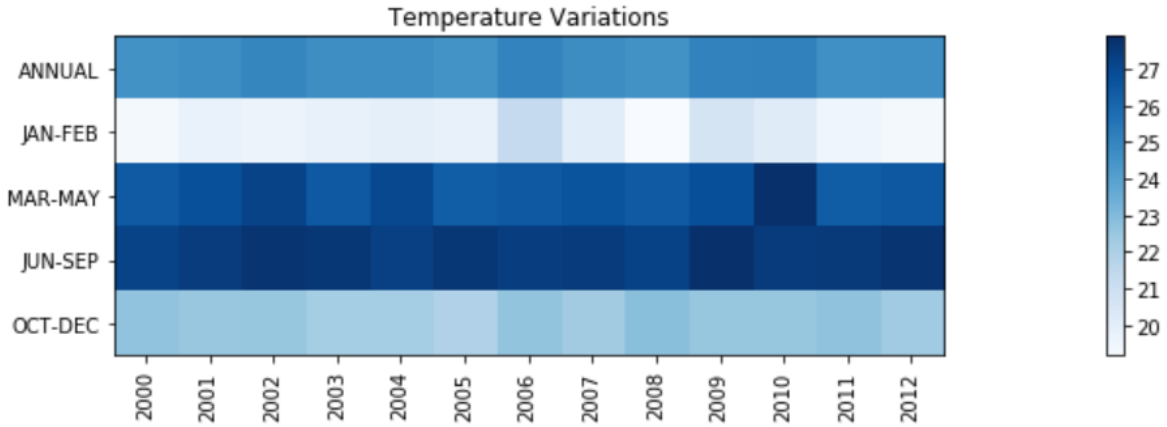

Figure 9: Temperature Variations of various year in Centigrade

3.2.5 Farmers Suicides

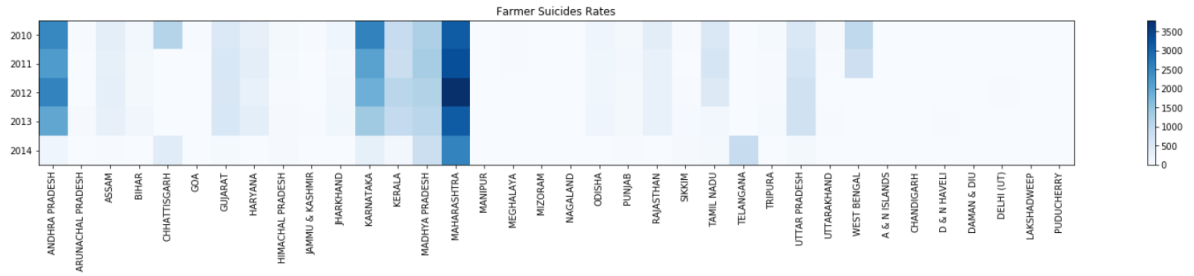

Figure 10: Farmers Suicides Rates state and year wise

3.2.6 Annual Rainfall

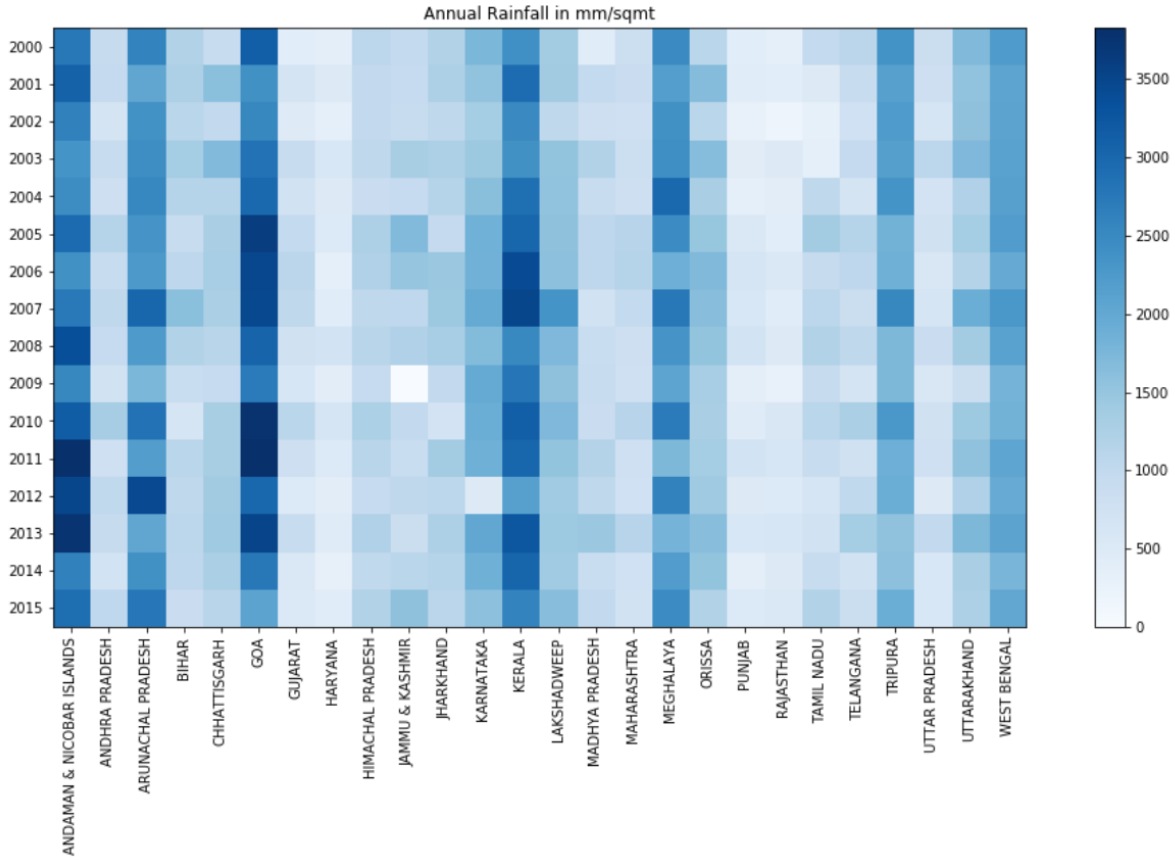

Figure 11: Annual Rainfall Stats

Chapter 4

DATA PREPROCESSING

During this stage of the Data Mining task, we clean the data, so as to remove any outliers or noise that may be present in the dataset. If any data is missing, then either that data needs to be filled (manually or automatically) or that tuple needs to be ignored completely. Fortunately, we checked for missing values in the dataset and found that none of the values were missing.

4.1 Data Cleaning :

The data needed to be cleaned in the beginning. The challenges faced while cleaning the data are :

1. The databases obtained composed of data of different years, which were not same across databases.

2. The names of some crops were not present in all the databases.

3. The database also contained a lot of missing data.

4. The data was of varying formats.

5. The naming conventions of crops and states were not the same across databases.

6. The units of measurements were different in different databases.

The databases were modified to contain data in a proper format and the missing values were replaced by the mean values of various years. Then the data was ready for any further processing.

4.2 Data Encoding :

Categorical data is data which has some categories such as, in our dataset; there are three categorical variables Season, Crop and District. Since the machine learning model completely works on mathematics and numbers, but if our dataset would have a categorical variable, then it may create trouble while building the model. While encoding if we encode these values, we will have many unique patterns which may be difficult in proceeding further. Hence, we

found out unique types of values in each categorical variable first and then encoded each of the variables using Label Encoder.

The following methodologies are typically used for text feature extraction:

1. **Tokenizing**: Splitting text in the form of tokens and assigning integer or floating-point value to each token.
2. **Counting**: counting the frequency or the occurrence of token in the document.
3. **Normalizing**: Penalizing the tokens or reducing the impact of those tokens that occur in most of the documents.

The below diagram sample of data after using Label Encoding :

```
[ ]  from sklearn.preprocessing import LabelEncoder
     lb_make = LabelEncoder()
     for col in X.columns[0:4]:
          X[col] = lb_make.fit_transform(X[col])
     X.head()
```

	Crop_Year	Season	Crop	District	pH	Avail-P	Exch-K	N	Rainfall	Area
0	0	0	1	0	6.19	7.13	41	8.89	928.5	21400
1	0	0	2	0	8.40	10.34	102	3.24	928.5	1400
2	0	0	10	0	7.10	8.46	46	5.54	928.5	1000
3	0	0	14	0	8.30	2.31	35	1.79	928.5	7300
4	0	0	17	0	6.40	6.08	76	22.26	928.5	3700

Figure 13: sample data after encoding

Then, the next step involves pre-processing the cleaned data, so that the data in the natural language form can be converted into machine readable form, on which the machine learning algorithms can be applied to get the result.

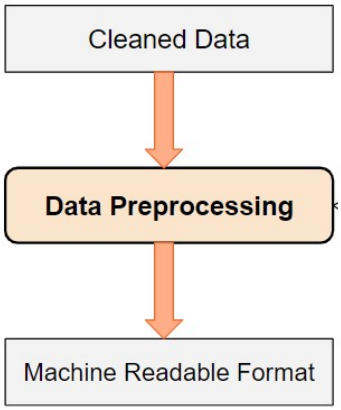

Figure 14: Data Pre-processing Flowchart

4.3 Splitting the Dataset into the Training set and Test set:

In machine learning data preprocessing, we divide our dataset into a training set and test set. This is one of the crucial steps of data pre-processing as by doing this, we can enhance the performance of our machine learning model. Suppose if we have given training to our machine learning model by a dataset and we test it by a completely different dataset. Then, it will create difficulties for our model to understand the correlations between the models.

4.4 Feature Selection

After pre-processing the acquired data, the next step is to identify the best features. The identified best features should be able to give high efficiency. Most of the features we have in our dataset are not numerical. So, the data needs to be encoded initially. For encoding the data, the author has used binary encoding for each feature. After encoding, correlation between the various features should be tested. The author has validated the correlations between the features using the Pearson correlation test. All the correlated features can be identified as the best features.

4.5 Encoding Methodology

The pre-processed data is still in natural language form, which is not understandable by the machine. So, it needs to be broken down into tokens or words, and these tokens need to be encoded in the form of integer or floating-point numbers, to be used by machine learning algorithms for prediction. This complete process of converting the pre-processed text in natural language form into encoded form is called vectorization or feature extraction.

Two popular techniques used for vectorization are Count Vectorizer and TF-IDF Vectorizer.

4.6 Encoding Methodology

The data from different tables were merged so that it can be analyzed. The tables were also unstacked when required, for proper understanding.

Chapter 5

BUILDING MODELS

5.1 Model using K-Means Algorithm

Clustering analysis is broadly used in many applications such as market research, pattern recognition, data analysis, and image processing.Clustering can also help marketers discover distinct groups in their customer base. And they can characterize their customer groups based on the purchasing patterns.Clustering also helps in identification of areas of similar land use in an earth observation database. It also helps in the identification of groups of houses in a city according to house type, value, and geographic location.Clustering also helps in classifying documents on the web for information discovery.Clustering is also used in outlier detection applications such as detection of credit card fraud.As a data mining function, cluster analysis serves as a tool to gain insight into the distribution of data to observe characteristics of each cluster.

Finding the clusters of states with year wise using all features

```python
# print(merged_all.columns)
from sklearn.cluster import AgglomerativeClustering
```

Python

```python
for_cluster = merged_all[['STATES','YEAR', 'Area', 'Production', 'Annual_Rain','Avg_Temperature', 'Price', 'Cost_per_Hectare', 'Cost_pe
for_cluster = for_cluster.groupby(['STATES','YEAR']).agg('mean').reset_index()
for_cluster.head()
X = TSNE(n_components=2).fit_transform(for_cluster.iloc[:,2:])
# k=4
# kmeans = km(n_clusters=k, random_state=0).fit(X)
# y = kmeans.labels_
clustering = AgglomerativeClustering(n_clusters=5).fit(X)
y=clustering.labels_
list(set(y))
N= len(list(set(y)))
cmap = mplot.cm.jet
colors = cmap(np.linspace(0, 1, N))
fig = mplot.figure(figsize=(10,10))
for i in list(set(y)):
    mplot.scatter(X[y==i,0],X[y==i,1],c=colors[i],label=str('Class '+str(i)))
mplot.legend()
mplot.title('Clustering of states with year using all features')
mplot.show()
```

Python

Figure 33: Clusters

5.2 Linear Regression

Linear regression is a statistical modeling technique used to examine the relationship between a dependent variable and one or more independent variables. It is commonly used in data mining and analysis to understand the nature of the relationship between variables and to make predictions based on this relationship.In the case of analyzing agricultural land, linear regression can be used to predict the yield of a particular crop based on various factors such as soil type, climate, fertilizer usage, etc. The model works by estimating the coefficients of the independent variables in a linear equation, which can then be used to predict the value of the dependent variable.

One of the advantages of using linear regression is its simplicity and interpretability. It is easy to understand the relationship between the variables and how they contribute to the outcome. Additionally, it is easy to visualize the results of the model by plotting the data and the line of best fit.

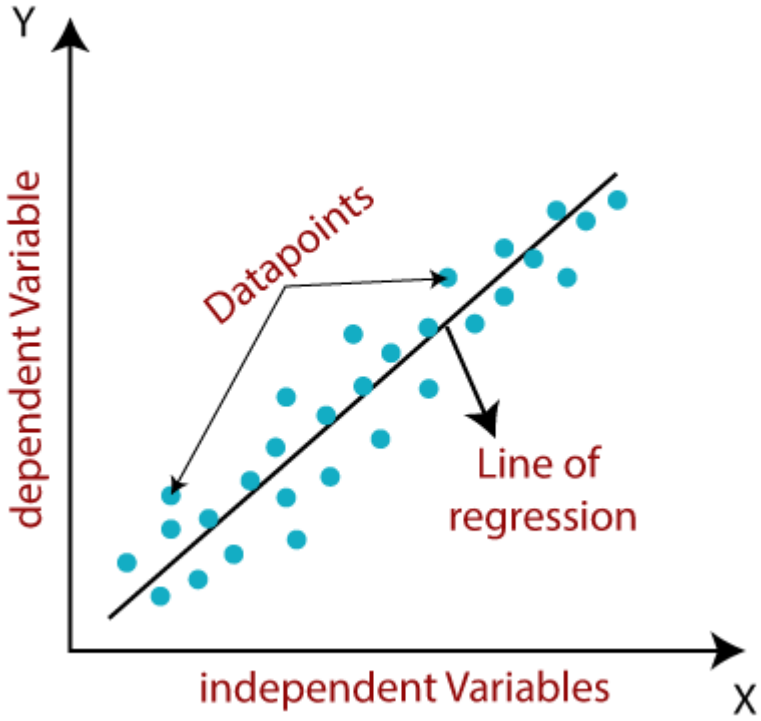

Figure 20: Linear Regression

```
#### Using linear regression to find the model for suicides
```

```
model = LR()
X = data_set.drop(columns = ['Suicides','YEAR'],axis=1)
y = data_set['Suicides']
tou = 1
Xx = X.values[y>tou]
Yy = y.values[y>tou]
v1 = Xx.mean(axis = 0)
v2 = Xx.std(axis = 0)
Xx = Xx - v1
Xx = Xx / v2
model.fit(Xx,Yy)
yd = model.predict(Xx)
model.score(Xx,Yy)
```

```
0.6207399311406134
```

```
mplot.plot(yd,Yy,'.')
mplot.title('Actual vs Predicted suicide rates')
mplot.show()
```

```
<Figure size 432x288 with 1 Axes>
```

Figure 36: Linear Regression

5.3 Decision Tree Regression :

Decision Tree is one of the most used, practical approaches for supervised learning. It can be used to solve both Regression and Classification tasks with the latter being put more into practical application. It is a tree-structured classifier with three types of nodes. The Root Node is the initial node which represents the entire sample and may get split further into further nodes. The Interior Nodes represent the features of a data set and the branches represent the decision rules. Finally, the Leaf Nodes represent the outcome. This algorithm is very useful for solving decision-related problems.

Decision tree regression observes features of an object and trains a model in the structure of a tree to predict data in the future to produce meaningful continuous output. Continuous output means that the output/result is not discrete, i.e., it is not represented just by a discrete, known set of numbers or values.

In scikit-learn python library, sklearn.tree. DecisionTreeRegressor module is used for carrying out Decision Tree regression. We will use our training dataset to fit the model.

Fig shows the sample code for the training model using Decision Tree regressor :

```
from sklearn.tree import DecisionTreeRegressor
dt_obj =DecisionTreeRegressor(random_state=1)
dt_obj.fit(X_train,y_train)
print('Train Score DT:',dt_obj.score(X_train,y_train))
print('Test Score DT:',dt_obj.score(X_test,y_test))
```

```
Train Score DT: 1.0
Test Score DT: 0.8162150580308982
```

Figure 27: Implementation of decision tree regression

```
data_set = data.fillna(0).groupby(['Crop','YEAR']).agg('mean').reset_index()
dicts = {}
for crop in data_set['Crop'].unique():

    X = data_set[data_set['Crop'] == crop]['YEAR'].values.reshape(-1,1)
    Y = data_set[data_set['Crop'] == crop]['Production'].values
    if(X.shape[0] > 5):
        linmodel = LR()
        linmodel.fit(X,Y)
        w = linmodel.coef_[0]
        dicts[crop] = [w]
#         print(w)
```

Finding the crops that have reduction in production over the years

- This will signify what crops are going be be rarely produced in future

```
slope_data = pd.DataFrame(dicts).T.reset_index().rename(columns = {0:'W_slope'})
sns.catplot(x='index', y='W_slope', data = slope_data,  height=10, aspect=12/8.27);
plt.ylim(-5000,5000)
plt.xticks(rotation='vertical')
plt.plot([0,slope_data.shape[0]],[-1000,-1000])
plt.title('Bar for crops that has more reduction in production over the years')
# plt.savefig('Figures/Crops_reduction')
plt.show()
```

```
<Figure size 1044.74x720 with 1 Axes>
```

Chapter 6

MODEL EVALUATION AND RESULTS

6.1 Performance Measures

The problem involves highly unbalanced dataset. So accuracy is not a well suited performance measure. With only 10% of the training data belonging to the positive class (hate tags), it is trivial to achieve 90% accuracy by a naive model which simply labels every input as clean. Indeed, we verified this by fitting a very simple naive bayes model which achieved a validation accuracy of whopping 97% which looks good on paper only as long as one doesn't analyse the model predictions manually or by some other performance measures. As quite expected, the recall was merely 65% which means the model is simply unable to recognise 35% of the hate comments. The precision scores were still poorer just 35% indicating even among the comments predicted as hate tags, 65% are misclassified. Such a model is not of any use which highlights the essence of a good evaluation metric. Precision-Recall or F1 score seem like the next obvious choice however they have their own share of limitations including selection of threshold value and relative importance to be given to precision vs recall. Hence, we finally settled on the ROC curve and AUC score which give a very accurate picture of the performance of a discriminative model.

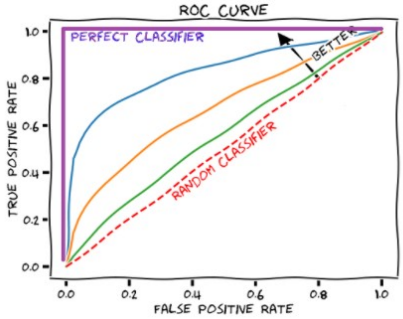

Figure 30: ROC

Accuracy of Decision Tree Classifier :-

```
from sklearn.tree import DecisionTreeClassifier

clf = DecisionTreeClassifier(random_state=42).fit(X_train, y_train)
clf.score(X_test,y_test)
```

```
0.9872727272727273
```

Accuracy of KNN :-

KNN Classifier for Crop prediction.

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier()
knn.fit(X_train_scaled, y_train)
knn.score(X_test_scaled, y_test)
```

```
0.9781818181818182
```

Confusion Matrix of KNN :-

Confusion Matrix

```
from sklearn.metrics import confusion_matrix
mat=confusion_matrix(y_test,knn.predict(X_test_scaled))
df_cm = pd.DataFrame(mat, list(targets.values()), list(targets.values()))
sns.set(font_scale=1.0) # for label size
plt.figure(figsize = (12,8))
sns.heatmap(df_cm, annot=True, annot_kws={"size": 12},cmap="terrain")
```

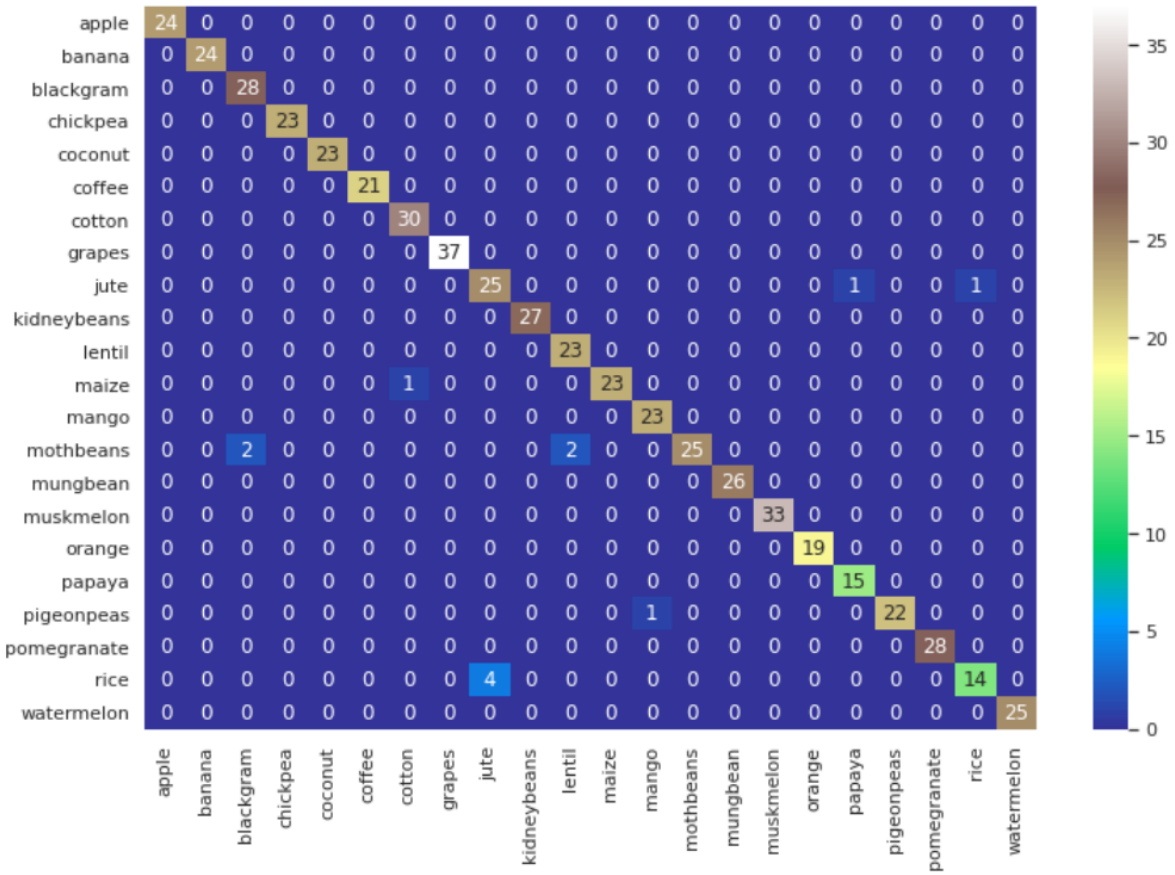

Figure :- Decision Matrix

6.1.1 Receiver Operating Characteristic

A Receiver Operating Characteristic is a curve which plots True Positive Rate vs True Negative Rate. These two factors are an indicator of the model performance.

1. **True Positive Rate (TPR)**:$TPR = \frac{TP}{TP+FN}$
2. **False Positive Rate (FPR)** :$FPR = \frac{FP}{FP+TN}$

The aim of any discriminative model is to maximize the TPR while at the same time keeping the FPR as low as possible. The adjoining figure shows a perfect classifier and a random classifier.

6.1.2 AUC

AUC denotes the complete Area Under the ROC curve for the given domain. AUC values can range from 0 to 1. The idea of AUC stems from the observation that a better model will have more area under the ROC curve with a perfect model having AUC=1 and a model which always predicts incorrectly having AUC score=0. In this sense AUC can be understood as the average of performance measures of the classifier across all thresholds. Another interesting interpretation is that it expresses the probability that the model gives a higher positive score to a hate comment in our case as compared to a clean comment for any threshold value chosen.

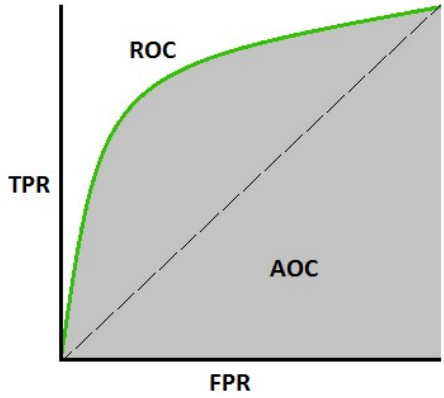

Figure 31: AOC ROC Curve

Advantages

1.Scale Invariant: AUC is independent of the scale and number of the predictions. It is based on fractions which always lie between 0 and 1. So it is a versatile model which can be used to compare the performance of several models alike.

2.Threshold Invariant: Unlike F-measure, it doesn't depend on the threshold set for classifying an example as positive.

Despite these, AUC does have some shortcomings which can be alleviated by combining it with some other metrics. We however limit ourselves to AUC for the purpose of this project.

6.2 Comparison Table

Model	Mean AOC ROC Score
K-Means	0.97
Linear Regression	0.79
Decision Tree	0.98

Chapter 7

INFERENCES AND CONCLUSION

We compared the performance of the model based on the mean AOC ROC scores. We observed that the classical models like decision tree and Linear Regression failed to achieve high AUC ROC scores. We also found out that the classifier chain method performed slightly better than binary relevance in this task. The tree based ensembling models performed significantly better than the classical models. We can further test the performance of state-of-the-art models like Transformers on this task. We can also experiment with more sophisticated models like GRUs. We can ensemble the results obtained from the various models in a majority vote fashion.