PROJECTREPORT

ESTIMATION OF CROP YIELD USING DATA ANALYTICS

Submitted by

PNT2022TMID13417

Vaishnavi S	-	951919CS109
PriyaDharshini G	-	951919CS073
SriDevi K	-	951919CS099
SriChandraLeha M	-	951919CS098

TABLEOFCONTENTS

CHAPTER1
INTRODUCTION

1.1 PROJECTOVERVIEW

Yield Prediction is an important agriculture problem. Every farmer is intrested in knowing how much tield he is about to expect. In the past, yield prediction was performed by consideering farmer's previous experience on a particula crop.Volume of data is enmormous in Indian Agriculture. IBM Cognoss Business Intelligence is a web based integrated business intelligence suite by IBM. It provides toolset for reporting, analytics, score carding, and monitoring of event and metrics. The software consist of several components designed to meet the different information requires in a company. IBM Cognos has several components which are used tohelp business users get fast answers to business related queries.

1.2 PURPOSE

It is been observed that farmers are facing the problem at the time of the yield of the crop because of the rapid changes in the weather where it effect the yield of the crop. Decrease the quality of the crop and which in turn provide less income to the farmers. This project works on achieving the more quality of the crop that will help the farmers to gain more money. In this project we have collected the datasets of all the factors that are depends of the crops of several years. Using this data the prediction is obtained to show that the harvest of the crop that is growth in that region.

CHAPTER2
LITERATURESURVEY

2.1 EXISTING PROBLEM

Other than blogging websites which provide information about the agriculture and agricultural accessories, there is no particular website for predicting the yield of the crop depending on the history in that specific geographical region.

2.2 REFERENCES

M. A. Jayaram and Netra marad, "Fuzzy Interference System for crop prediction", Journal of Intelligent Systems, 2012, 21 pp.363-372.

Prediction of crop yield is significant in order to accurately meet market requirements and proper administrationof agricultural activities directed towards enhancement in yield.

Several parameters such as weather, pests, biophysical and morphological features merit their consideration while determining the yield. However, these parameters are uncertain in their nature, thus making the dtermined amount of yield to be approximate. A huge database (around 1000 of records) of physio morphological features such as days of 50 percent flowering,dead heart percentage, plant height etc were consider for the development of model.

The results have clearly shown that the panicle length contributes fourth yield as the lone parameter reflected by very low RMS value.

P. Vindhya "Agricultural analysis for Next Generation High tech Farming in Data mining", Anna University, Trichy, Tamilnadu, India, 5 May 2015.

Recent developments in Information Technology for agriculture field have become an interesting research area to predict the crop yield.

In today's world, the amount of information stored has been enormously increasing day by day which is generally in the unstructured form and cannot be used for any processing to extract useful information using mining technique.

This paper present the brief analysis of data mining methods and agriculture techniques, farm types, soil types, Prediction using multiple linear regression (MLR) technique for the selected region.

It concentrate organic, inorganic and real estate datasets from which the prediction in agriculture will be acheived.

Veenadhari et al,(2014)

It described the purpose of data mining methods in the area of agriculture. A few of the data mining methods, such as the k-means,ID3 algorithms, the k nearest neighbor, support vector machines, artificial neural networks were presented.

Grajales et al ,(2015)

It have proposed a web application that utilizes open dataset like historical production, land cover, local climate conditions and integrates them to provide easy access to the farmers. The proposed architecture mainly focuses on open source tools for the development of the application. The user can select location from map for which the details are available at one click.

Study proposed to less complicated, easily accessable methods to determine and qualified the yield gaps between various agricultural fields. First method works closely with the constructive maps representating the average crop yield, it can be used directly to access specified crop yield influenzing factors for further studies whereas the second method use the remote sensing technology to retrieve the data for providing the useful information regarding the crop yield prediction and estimation.

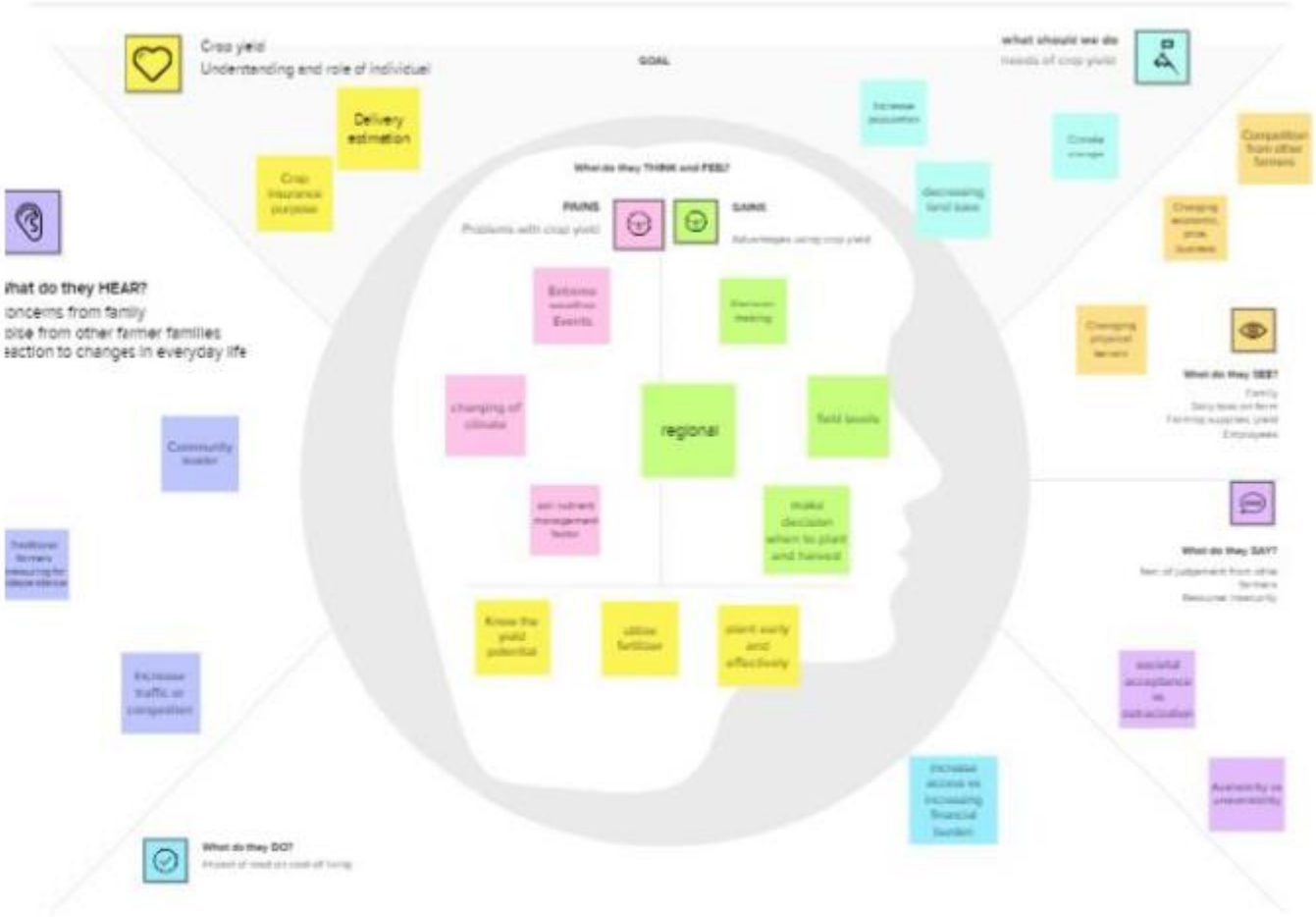
2.3 PROBLEMSTATEMENTDEFINITION

It is been observed that farmers are facing the problem at the time of the yield of the crop because of the rapid changes in the weather where it effect the yield of the crop. Decrease the quality of the crop and which in turn provide less income to the farmers. This project works on achieving the more quality of the crop that will help the farmers to gain more money. In this project we have collected the datasets of all the factors that are depends of the crops of several years. Using this data the prediction is obtained to show that the harvest of the crop that is growth in that region.

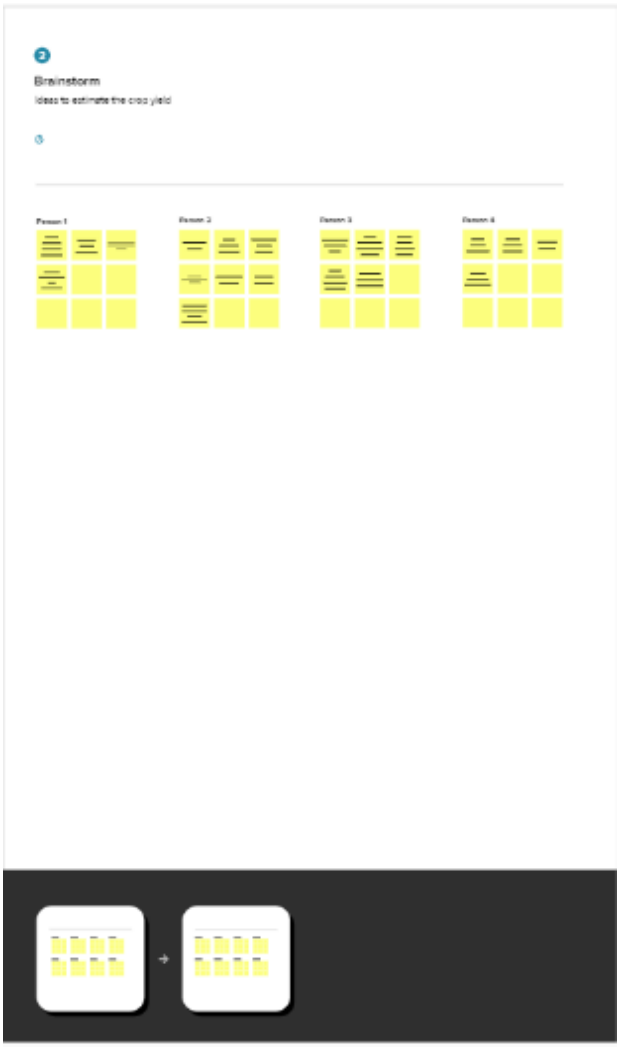
CHAPTER3

IDEATIONANDPROPOSEDSOLUTION

3.1 EMPATHYMAPCANVAS



3.2 IDEATION&BRAINSTORMING


2
Brainstorm
Ideas to estimate the crop yield


Template
Brainstorm
& idea prioritization
Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.
1
Define your problem statement
Estimation of crop yield using data analytics

3.3 PROPOSEDSOLUTION

S.No	Parameter	Description
1.	Problem Statement(Problem to be solved)	India is one of the top countries for agricultural output, making crop production one of the most significant sources of revenue in the country. Inputs like seed, water, pesticides, and fertilisers may be used precisely and at the proper moment for the crop to maximise production, quality, and yields due to digital farming. To choose the crops that will be grown in a field , the majority of farmers follow conventional agricultural practises. Farmers may make better Decisions for healthy crop production based on statistics.
2	Idea/Solution description	Crop production in India is one of the most important sources of income and India is one of the top countries to produce crops. As per this project we will be analyzing some important visualization, creating a dashboard and by going through these we will get most of the insights of Crop Production in India

3	Novelty/Uniqueness	Agriculture is important for human survival because it serves the basic need. Due to variations in climatic conditions, there exist bottlenecks for increasing the crop production in India. It has become challenging task to achieve desired targets in Agri based crop yield. To choose the crops that will be grown in a field , the majority of farmers follow conventional or traditional agricultural practises. Farmers may make better decisions for healthy crop production based on statistics. Agricultural statistics are useful for planning, monitoring and evaluation purposes. Therefore, we use IBM CognosBItoolinorderto provide a useful insights fromthedataregardingtheagricultureofIndiaandperformanalyticsandprovide

		Necessary statistics in order to increase the crop production.
4	Social Impact/Customer Satisfaction	Crop yield prediction is one of the important factors in agriculture practices. Farmers need information regarding crop yield before sowing seeds in their fields to achieve enhanced crop yield. The use of technology in agriculture has increased in recent year and data analytics is one such trend. By performing analytics in given data and providing useful insights such as average crop production season wise will help farmers to identify the season with high and least crop production with help of insight, and we can also get to know the area that's been used yearly for crop production, by producing such insights it will create a good impact in efficiency of Crop production in agriculture.
5	Business Model(Revenue Model)	Supply chain operation between farmers and Entrepreneurs. Helps the companies in project scheduling. Farmers can achieve enhanced crop yield by predicting the yield before sowing the seeds. farmers can over come the challenging tasks involved in crop production. The estimation of Production of crop help the companies in planning supply chain decision

6	Scalability of the Solution	In terms of scalability of the project, we can increase the crop yield production by performing analytics and inter preting useful insights from given data. Insights such as estimating the season wise average crop production, estimating yearly area used in crop production, by providing such insights this can help farmers taking a better decision I'm choosing suitable crops according to season and we can get to know the state in India with least crop production and can focus on those states to increase their crop production. Therefore, this solution can significantly increase the scalability of the crop production in India

3.4 PROBLEMSOLUTIONFIT

system design thought as the application of theory of the system for the development of the project.system design defines the architecture, data flow, use case,class, sequence, activity diagram of theproject development.

A)IBM cognosAnalytics:

Itisasetofbusinessintelligence toolavailableoncloudoron-premise.

The primary focus is in the area of Descriptive Analytics, to help users see the information in your datathrough dashboard, professional reportingandselfservice data exploration. In this work,we use the IBM cognos data analytics for analytics for analysing the crop yeild data.Important features:

1) Getconnected.
2) Prepareyourdata.
3) BuiltVisualization.
4) Identifypatterns.
5) Generate personalize report.
6) Gaininsight.
7) Stay connected.

CHAPTER4

REQUIREMENTANALYSIS

4.1 FUNCTIONALREQUIREMENTS

FR.NO	FunctionalRequirement(Epic)	SubRequirement(Story/Sub-Task)
FR1	UserRegistration	Registration throughForm RegistrationThrough Gmail RegistrationthroughLinkedIN
FR2	UserConfirmation	ConfirmationviaEmailC onfirmationviaOTP
FR3	LogintoDashboard	Visualizationofcropgrowthrate
FR4	InteractiveDashboard	Changethefieldsofvisualizations accordingtouserneeds

4.2 NONFUNCTIONALREQUIREMENTS

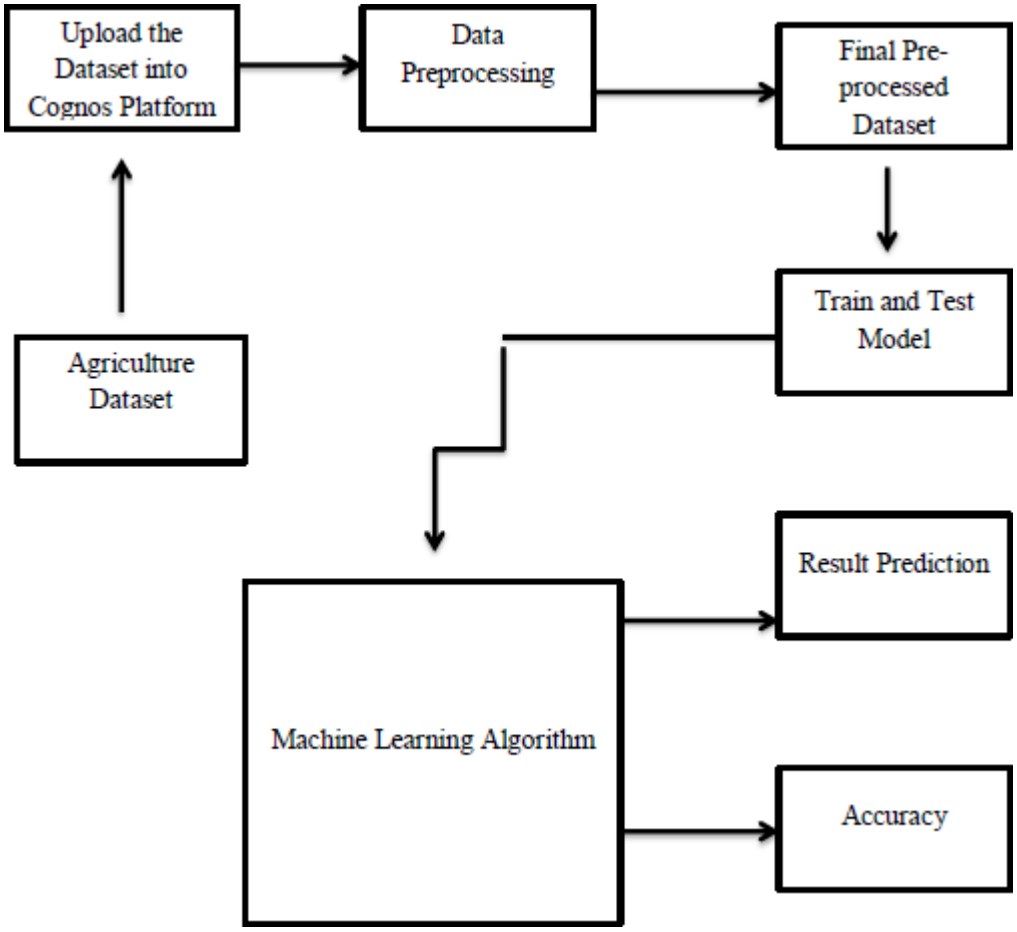
FR.NO	NonFunctionalRequirement	Description
NFR-1	Usability	Easytoaccessandusedashboareffe ctively
NFR-2	Security	Use login credentials are maintained in asecuredmannerandrestrictedto unauthorizedaccess
NFR-3	Reliability	Dataset used are collected fromtrustworthysitesanditisuptodate.
NFR-4	Performance	Highperformance
NFR-5	Availability	Activelyavailabletoallsources
NFR-6	Scalability	ItisscalablesinceithasinteractiveDas hboard

CHAPTER5

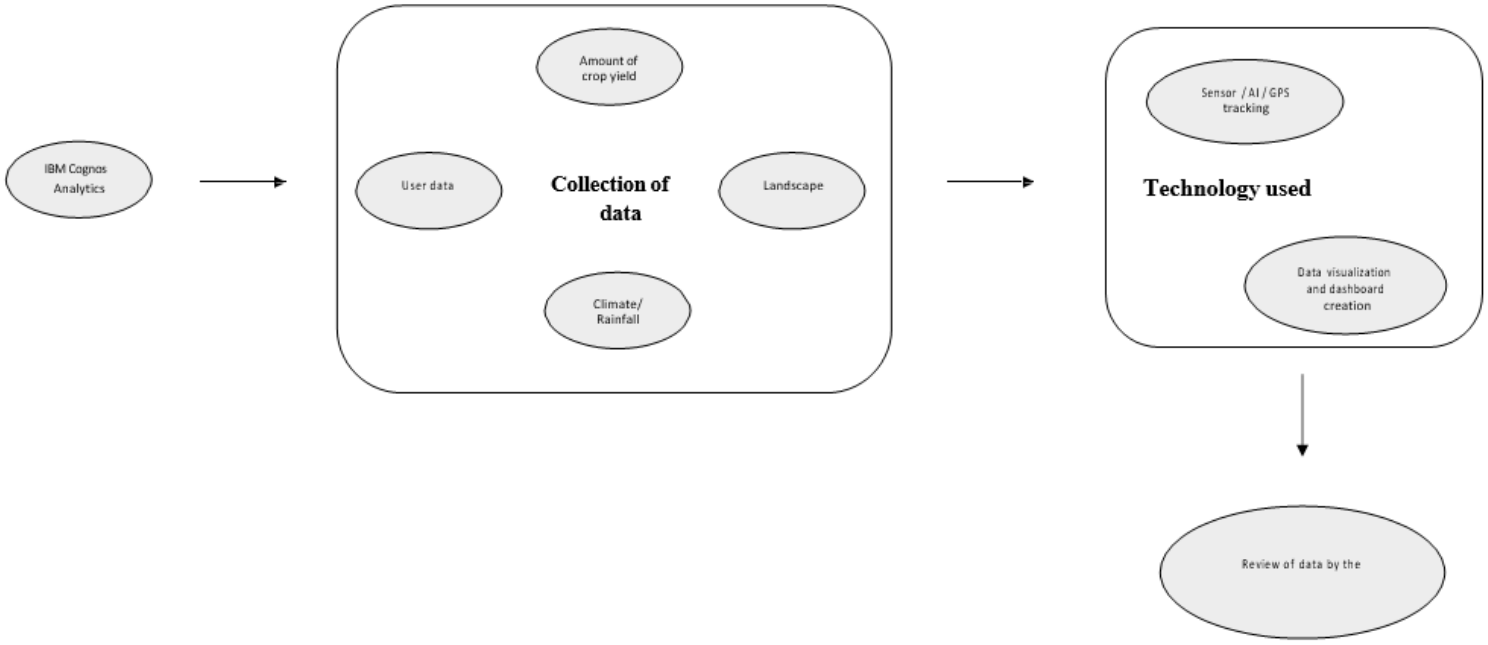
PROJECTDESIGN

5.1 DATAFLOWDIAGRAM

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows with in a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, andwhere data isstored.


Upload the Dataset into Cognos Platform
Data Preprocessing
Final Pre-processed Dataset
Train and Test Model
Agriculture Dataset
Machine Learning Algorithm
Result Prediction
Accuracy

5.2 SOLUTION&TECHNICALARCHITECTURE



S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Predict climate resilient	Absorb climatic changes and the factors affecting or contributing to the crop yield.	AI, IoT and blockchain
3.	Pesticide management	Management and usage of proper pesticides that contribute to the higher production of crops	IoT and conventional pesticides
4.	Farm management	Absorbing and implementing the decisions involved in organizing and operating a farm for maximum production and profit	Farm automation
5.	Database	A database is a collection of inter-related information or data stored electronically in a computer system	MySQL, PostgreSQL, Big Query
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	Data API	Data APIs within the IBM Environmental Intelligence Suite tap into the breadth and depth of climate, environmental and weather data to provide current and forecasted conditions, seasonal and sub-seasonal forecasts.	IBM Weather API, etc.
9.	Power API	It allows external applications to connect and interact with Power data, which is solar and meteorological data from satellite observations.	NASA APIs
10.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :l	Local, Cloud Foundry, Kubenetes, etc.

5.3 USERSTORIES

UserType	FunctionalRequirement (Epic)	User StoryNumber	User Story /Task	Acceptance criteria	Priority	Release
Customer(Mobileuser)	Registration	USN-1	As a user, Ican registerfor theapplicationby enteringmy email,password,andconfirmingmy password.	I can accessmyaccount /dashboard	High	Sprint-1
		USN-2	As a user, Iwill receiveconfirmationemail once Ihaveregisteredforthe application	I can receiveconfirmationemail & clickconfirm	High	Sprint-2
		USN-3	As a user, Ican registerfor theapplicationthroughgmailor facebook	I canregister &access thedashboardwithFacebook Login	Medium	Sprint-2
	Login	USN-4	Asauser,I canloginto	Icanlogin tothe	High	Sprint-1

			theapplicat ionby enteringem ail& password	application		
	Dashboard	USN-5	Go todashboar dand referthe contentabo utour project	I can readinstruct ionsalsoand thehome pageisuser-friendly.	Low	Sprint-1
	Upload Image	USN-6	As a user, Ican able toinput theimages ofdigitaldo cumentsto the application	As a user, Ican able toinput theimages ofdigitaldo cumentsto the application	High	Sprint-3
	Predict	USN-7	As a user Ican able toget therecognis eddigit asoutputfro mthe imagesof digitaldocu mentsorima ges	I can accessthere cognizeddi gits fromdigital documento rimages	High	Sprint-3

		USN-8	As a user, Iwilltrain andtestthe	I can able totrainand testthe	Medium	Sprint-4

			inputtogett hemaximu maccuracy of output.	application until it getsmaxim umaccurac yof theresult.		
Customer(Webuser)	Login	USN-9	As a user, Ican use theapplicat ionby enteringmy email, password.	I can accessmyac count	Medium	Sprint-4
Customer Care Executive	Dashboard	USN-10	upload theimage	Recognizea ndgetthe output	High	Sprint-1
Administrator	Security	USN-11	updatedthe features	checkingthe security	Medium	Sprint-1

CHAPTER6

PROJECTPLANNINGANDSCHEDULING

6.1 SPRINTPLANNINGANDESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Working with the data set	USN-1	Understanding the data set .	10	Medium	Vaishnavi, Sridevi
Sprint-1	Working with the data set	USN-2	Loading the data set.	10	High	Sridevi, Srichandraleha
Sprint-2	Prepare the data	USN-3	Convert the data's into required format	10	Medium	Priyadharshini, Sridevi
Sprint-2	Data exploration	USN-4	Explore the data's which is uploaded in the IBM cognos	10	Medium	Priya dharshini, Sri Chandra leha
Sprint-3	Data visualization	USN-5	Creating the data visualization chart	10	High	Priyadharsini, Vaishnavi
Sprint-3	Dashboard	USN-6	Creating a dashboard	10	High	Vaishnavi, Srichandra leha
Sprint-4	Report	USN-7	Creating the report	10	High	Vaishnavi,
Sprint-4	Export	USN-8	Export the report to the Github	20	High	Priyadharshini

6.2 SPRINTDELIVERYSCHEDULE

S.NO	MILESTONES	ACTIVITIES	STARTDATE	COMPLETEDATE
1	SolutionRequirements	CreatingtheIBMCognosforcreating dashboard and datavisualizationcharts.	22-Aug-2022	24-Aug-2022
2	ProjectObjectives	Preparetheprojectobjectives	22-Aug-2022	24-Aug-2022
3	Project Flow	Preparetheprojectflow	22-Aug-2022	24-Aug-2022
4	IBMCloudAccount	CreatingIBMcloudaccount	22-Aug-2022	24-Aug-2022
5	IBMCognosAnalytics	CreatingIBMcognosaccount	22-Aug-2022	24-Aug-2022
6	WorkingWiththeDataset	UnderstandingTheDatasetLoadingTheDataset	24-oct-2022	19-nov-2022

7	Datavisualizationcharts	SeasonsWithAverageProductions WithYearsUsageofAreaAndProduction Top10StateswithMostAreaStateWithCropProduction States With the CropProductionAlongwithSeason	24-oct-2022	19-nov-2022
8	Creating TheDashboard	CreatingTheDashboard	24-oct-2022	19-nov-2022
9	ExportTheAnalytics	ExportTheAnalytics	24-oct-2022	19-nov-2022
10	IdeationPhase	Literature Survey On TheSelectedProject &InformationGathering PrepareEmpathyMapIdeation	22-Aug-2022	27-Aug-2022

11	ProjectDesignPhase-I	Proposed SolutionProblem Solution FitSolutionArchite cture	22-Aug-2022	17-sep-2022
12	ProjectDesignPhase-II	CustomerJourneyFu nctionalRequiremen t	22-sep-2022	01-oct-2022

		Data Flow DiagramsTechnolog yArchitecture		
13	Project PlanningPhase	PrepareMilestone&Activi tyList SprintDeliveryPlan	17-oct-2022	22-oct-2022
14	ProjectDevelopm entPhase	Project Development - DeliveryofSprint-1 Project Development - DeliveryofSprint-2 Project Development - DeliveryofSprint-3 Project Development - DeliveryofSprint-4	24-oct-2022	19-nov-2022

CHAPTER7

CODING & SOLUTIONING

```
from sklearn.model_selection import train_test_split
```

```
X = crop_data.drop('Production',axis=1)
X.head()
```

```
y = crop_data['Production']
y.head()
```

```
0   2000.00
1      1.00
2    321.00
3    641.00
4    165.00
Name: Production, dtype: float64
```

```
X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.33, random_state=42)
```

```
X_test.shape
```

```
(70378, 735)
```

```
y_test.shape
```

```
from sklearn.linear_model import LinearRegression
```

```
crop_model = LinearRegression()
```

```
crop_model.fit(X_train,y_train)
```

```
LinearRegression()
```

Prediction

```
crop_predictions = crop_model.predict(X_test)
crop_predictions
```

```
array([  591668.81926186, -1224399.18087533,  -528237.48815455, ...,
         -79121.86366799,  -152631.26284787,   162578.32881211])
```

```
crop_model.coef_
```

```
crop_model.intercept_
```

```
-33686294.63228672
```

```
predicted_crop_val = pd.DataFrame({'Actual':y_test,'Predicted':crop_predictions})
predicted_crop_val
```

	Actual	Predicted
191452	480.00	591668.82
172763	290.00	-1224399.18
81954	26248.00	-528237.49
79750	6.00	-1579489.85
193433	20.00	-1355248.90
...	...	...
216969	5.00	-836851.13
1823	2500.00	9175562.30
121894	65.00	-79121.86
200451	2.00	-152631.26
107319	160.00	162578.33

70378 rows × 2 columns

```
from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_score
```

```
df['Production'].mean()
```

```
648868.9434560126
```

```
crop_predictions.mean()
```

```
567688.9612009758
```

```
mean_absolute_error(y_test,crop_predictions)
```

```
2199779.5972323604
```

```
mean_squared_error(y_test,crop_predictions)
```

```
301754861682964.25
```

```
np.sqrt(mean_squared_error(y_test,crop_predictions))
```

```
17371092.702618457
```

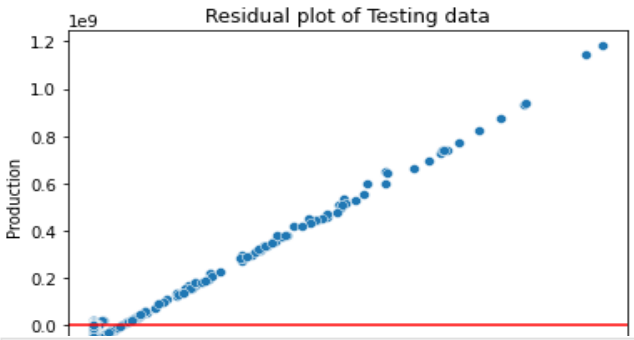
```
def mape(actual, pred):
    actual, pred = np.array(actual), np.array(pred)
    return np.mean(np.abs((actual - pred) / actual)) * 100

mape(y_test,crop_predictions)
```

```
test_residuals = y_test - crop_predictions
```

```
test_residuals
```

```
191452    -591188.82
172763    1224689.18
81954      554485.49
79750     1579495.85
193433    1355268.90
              ...
216969     836856.13
1823     -9173062.30
121894     79186.86
200451     152633.26
107319    -162418.33
Name: Production, Length: 70378, dtype: float64
```

```
sns.scatterplot(x=y_test,y=test_residuals)
plt.axhline(y=0,color='red')
plt.title('Residual plot of Testing data');
```




```
r = r2_score(y_test,crop_predictions)
print("R2score when we predict using Linear Regression is ",r)
```

```
R2score when we predict using Linear Regression is  0.18493399566622137
```

Results of training data

```
train_set_predictions = crop_model.predict(X_train)
train_set_predictions
```

```
array([-3259622.32998938,  -530500.35001556, -1254656.07860044, ...,
        587913.93084943,   474760.02963475,  1698161.198421  ])
```

```
train_set_predictions.mean()
```

```
632100.1644935672
```

```
sns.scatterplot(x=y_train,y=y_train-train_set_predictions)
plt.axhline(y=0,color='red')
plt.title('Residual plot of Training data');
```

```
r = r2_score(y_test,crop_predictions)
print("R2score when we predict using Linear Regression is ",r)
```

```
R2score when we predict using Linear Regression is  0.18493399566622137
```

Results of training data

```
train_set_predictions = crop_model.predict(X_train)
train_set_predictions
```

```
array([-3259622.32998938,  -530500.35001556, -1254656.07860044, ...,
         587913.93084943,   474760.02963475,  1698161.198421  ])
```

```
train_set_predictions.mean()
```

```
632100.1644935672
```

```
sns.scatterplot(x=y_train,y=y_train-train_set_predictions)
plt.axhline(y=0,color='red')
plt.title('Residual plot of Training data');
```

```
X2 = crop_data2_ac.drop('Production',axis=1)
X2.head()
```

	Crop_Year	Area	State_Name_Andaman and Nicobar Islands	State_Name_Andhra Pradesh	State_Name_Arunachal Pradesh
0	2000.00	1254.00	1	0	0
1	2000.00	2.00	1	0	0
2	2000.00	102.00	1	0	0
3	2000.00	176.00	1	0	0
4	2000.00	720.00	1	0	0

5 rows × 144 columns

```
y2 = crop_data2_ac['Production']
y2.head()
```

```
0    2000.00
1       1.00
2     321.00
3     641.00
4     165.00
Name: Production, dtype: float64
```

```
from sklearn.model_selection import train_test_split
```

```
def mape(actual, pred):
    actual, pred = np.array(actual), np.array(pred)
    return np.mean(np.abs((actual - pred) / actual)) * 100

mape(y_train,train_set_predictions)
```

```
7092939.504862983
```

```
Crop_data2 = df.drop(['District_Name'],axis=1)
```

```
Crop_data2.head()
```

	State_Name	Crop_Year	Season	Crop	Area	Production
0	Andaman and Nicobar Islands	2000.00	Kharif	Arecanut	1254.00	2000.00
1	Andaman and Nicobar Islands	2000.00	Kharif	Other Kharif pulses	2.00	1.00
2	Andaman and Nicobar Islands	2000.00	Kharif	Rice	102.00	321.00
3	Andaman and Nicobar Islands	2000.00	Whole Year	Banana	176.00	641.00
4	Andaman and Nicobar Islands	2000.00	Whole Year	Cashewnut	720.00	165.00

```
crop_data2_ac = pd.get_dummies(data=Crop_data2)
crop_data2_ac.head()
```

```
X2_train, X2_test, y2_train, y2_test = train_test_split( X2, y2, test_size=0.33, random_state=42)
```

```
from sklearn.linear_model import LinearRegression
crop_model2 = LinearRegression()
crop_model2.fit(X2_train,y2_train)
```

```
LinearRegression()
```

```
#Prediction
crop2_predictions = crop_model2.predict(X2_test)
crop2_predictions
```

```
array([ -27567.15607489, -242975.82827755,  193874.62415348, ...,
       -224034.83310432, -455078.28891501,  120365.51545634])
```

```
crop_model2.coef_
```

```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
df['Production'].mean()
```

```
648868.9434560126
```

```
crop2_predictions.mean()
```

```
556512.1489000395
```

```
mean_absolute_error(y_test,crop2_predictions)
```

```
2077972.4729596092
```

```
mean_squared_error(y_test,crop2_predictions)
```

```
306965695608223.44
```

```
np.sqrt(mean_squared_error(y_test,crop2_predictions))
```

```
17520436.51306164
```

```
r2_score(y_test,crop2_predictions)
```

```
0.1708590821320356
```

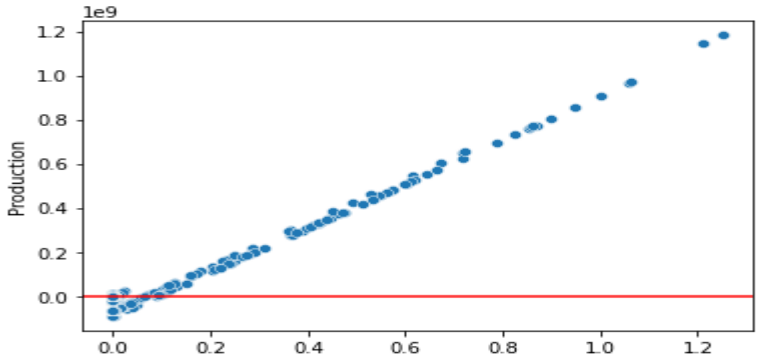
```
test2_residuals = y2_test - crop2_predictions
```

```
test2_residuals
```

```
test2_residuals
```

```
191452     28047.16
172763    243265.83
81954    -167626.62
79750    1687019.88
193433    574032.99
             ...
216969    659202.56
1823     -190376.83
121894    224099.83
200451    455080.29
107319   -120205.52
Name: Production, Length: 70378, dtype: float64
```

```
sns.scatterplot(x=y2_test,y=test2_residuals)
plt.axhline(y=0,color='red')
```



CHAPTER8
TESTING

8.1 TESTCASES

Test caseID	Feature Type	Component	TestScenario	ExpectedResult	ActualResult	Status
HP_TC_001	UI	HomePage	Verify UIelementsinthe HomePage	TheHomepagemustbedisplayedproperly	Workingasexpected	PASS
HP_TC_002	UI	HomePage	CheckiftheUIelementsaredisplayedproperly indifferentscreensizes	TheHomepagemust be displayedproperlyinallsizes	The UI is notdisplayedproperly inscreen size2560x1801and768x630	FAIL
HP_TC_003	Functional	HomePage	Checkifusercan uploadtheirfile	Theinputimageshould beuploaded to theapplicationsuccessfully	Workingasexpected	PASS
HP_TC_004	Functional	HomePage	Check if usercannotuploadunsupportedfiles	The applicationshould not allowuser to select anonimagefile	Userisabletouploadanyfile	FAIL
HP_TC_005	Functional	HomePage	Check if thepageredirectsto the resultpageonce theinputisgiven	The page shouldredirect to theresultspage	Workingasexpected	PASS

BE_TC_001	Functional	Backend	Checkifallthero utes areworkingpr operly	All the routesshouldpr operlywork	Workingasex pected	PASS
M_TC_001	Functional	Model	Check if themodel canhandlevari ousimage sizes	Themodelshouldre scale theimageand predict theresults	Workingasex pected	PASS
M_TC_002	Functional	Model	Check if themodelpr edictsthedi git	Themodelshouldpr edict thenumber	Workingasex pected	PASS
M_TC_003	Functional	Model	Check if themodel canhandleco mplexinputim age	Themodelshouldpr edict thenumber in thecompleximage	Themodelfailst o identify thedigit since themodel is notbuilt to handlesuchda ta	FAIL
RP_TC_001	UI	ResultPage	Verify Ulelementsi nthe ResultPage	The Result pagemustbedispl ayedproperly	Workingasex pected	PASS
RP_TC_002	UI	ResultPage	Check if theinputimag eisdisplayed properly	The input imageshould bedisplayedproperl y	Thesizeofthein put imageexceeds thedisplaycon tainer	FAIL
RP_TC_003	UI	ResultPage	Checkiftheres ult isdisplayedpr operly	Theresultshouldbe displayedproperl y	Workingasex pected	PASS
RP_TC_004	UI	ResultPage	Check if theotherpre dictionsaredi splayedprop erly	The otherpredictionssh ouldbe displayedproperl y	Workingasex pected	PASS

8.2 USERACCEPTANCETESTING

8.2.1 DEFECTANALYSIS

Resolution	Severity1	Severity2	Severity3	Severity4	Total
ByDesign	1	0	1	0	2
Duplicate	0	0	0	0	0
External	0	0	2	0	2
Fixed	4	1	0	1	6
NotReproduced	0	0	0	1	1
Skipped	0	0	0	1	1
Won'tFix	1	0	1	0	2
Total	6	1	4	3	14

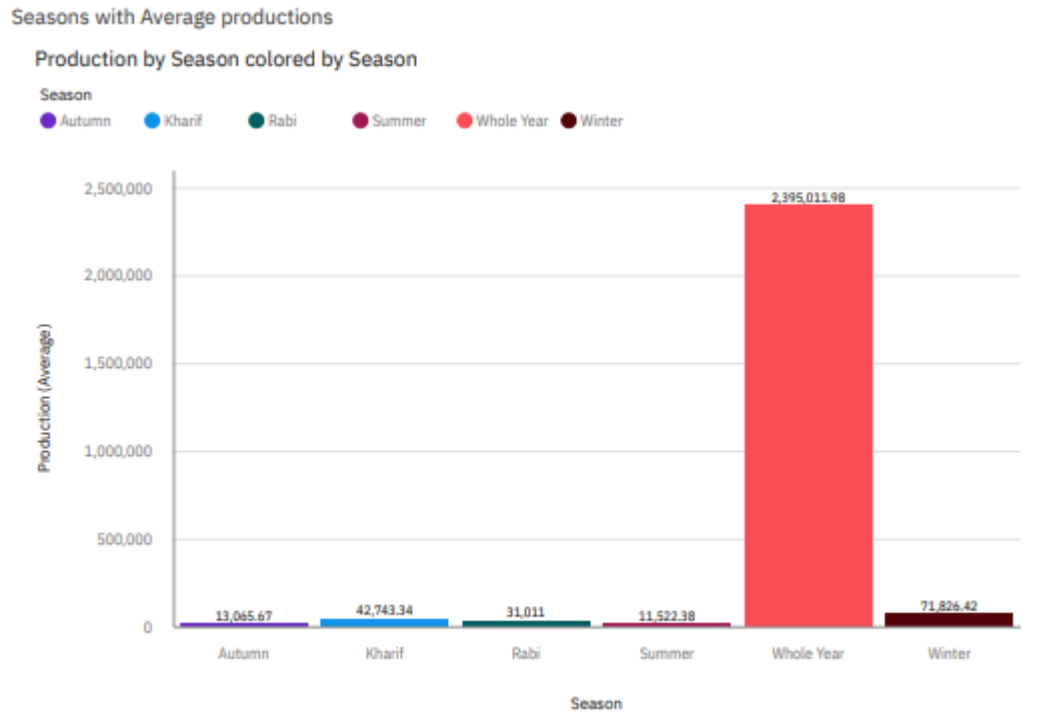
8.2.2 TESTCASEANALYSIS

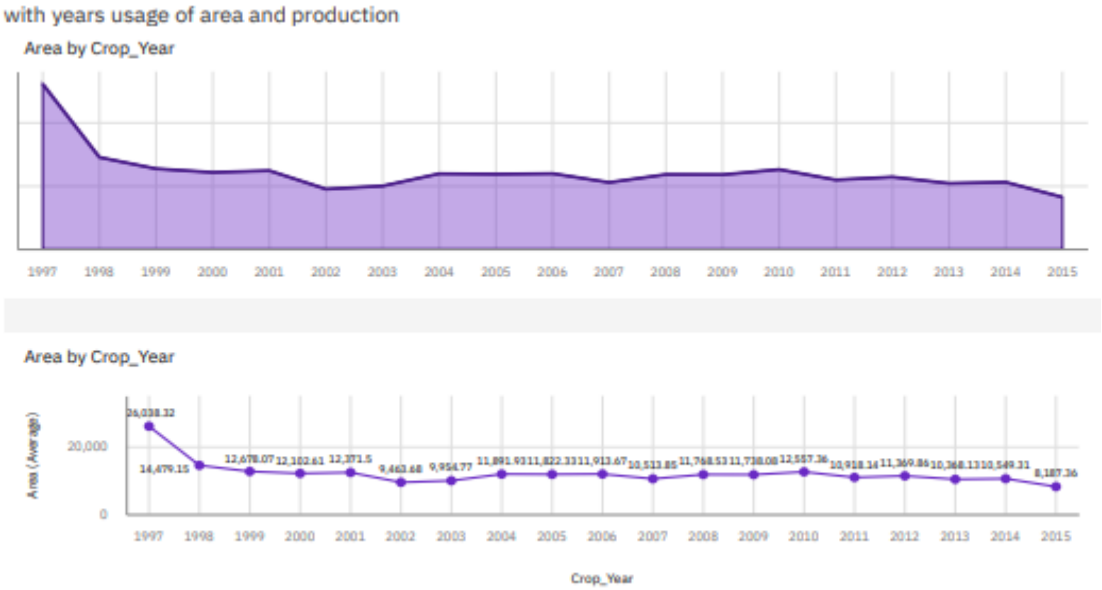
Section	Total Cases	NotTested	Fail	Pass
ClientApplication	10	0	3	7
Security	2	0	1	1
Performance	3	0	1	2
ExceptionReporting	2	0	0	2

CHAPTER9

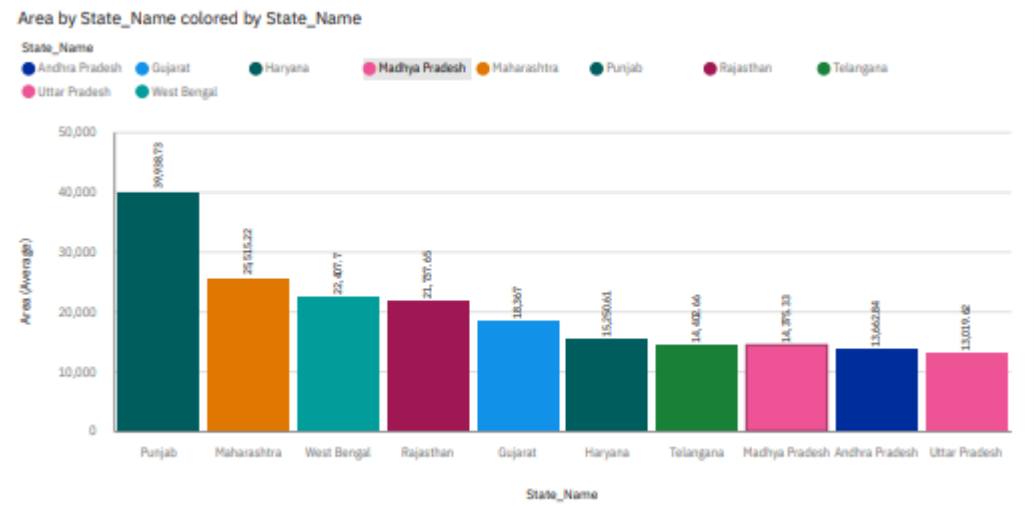
RESULTS

9.1 Dashboard:

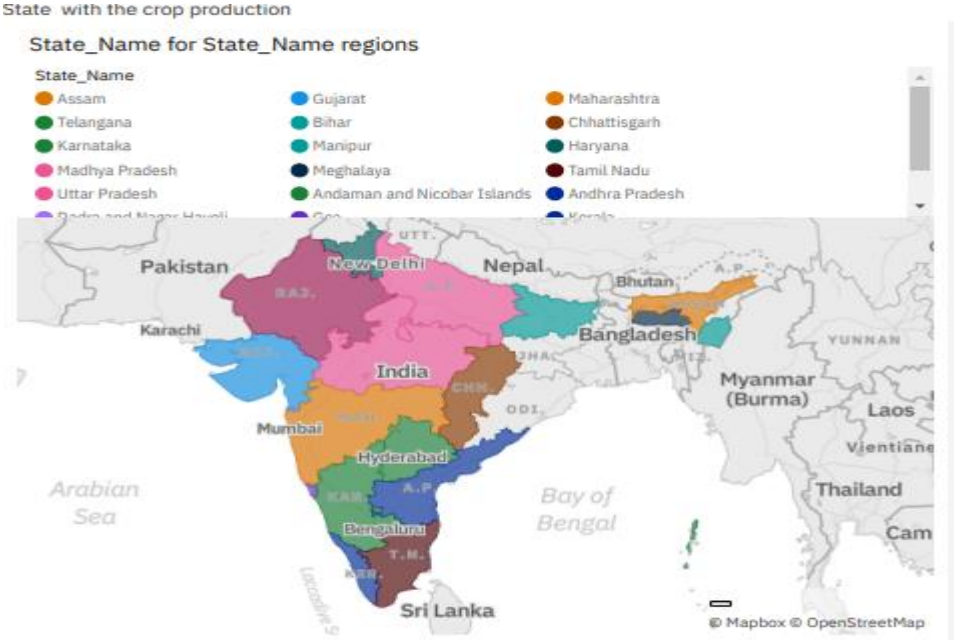




Top 10 states with Area



State with the crop production



States with the crop production along with season

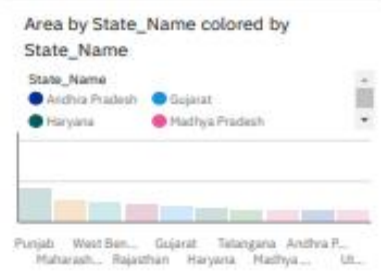
State_Name and Crop

Crop	State_Name
Banana	Andaman and Nicobar Isla...
	Andhra Pradesh
	Assam
	Bihar
	Chhattisgarh
	Dadra and Nagar Haveli
	Goa

Season and Crop

Crop	Season
Banana	Whole Year

crop production



State_Name and Crop

Crop	State_Name
Apple	Tamil Nadu
Arcanut (Processed)	Karnataka
	Andaman and Nico...

Season and Crop

Crop	Season
Arhar/Tur	Autumn
	Kharif
	Rabi

CHAPTER 10

EXPERIMENTAL RESULTS

Experimental Results

For every system, its efficiency and accuracy is important. Similarly in our sys- tem accuracy is the key feature to judge the correctness of the model. In our model, we considered Mysore region with average minimum and maximum temperature, average rainfall, and average minimum and maximum pressure datasets. We considered data points of all the above parameters for 1997 to 2014. In this 1997 to 2010 data points considered as training sets and 2011-2014 data points as testing sets. With these by using ”Multiple Linear Regression algorithm”we have evaluated the accuracy for Rice, Ragi and Sugarcane crops.

The accuracy for seasonal crops (Rice and Ragi) using our model we observed asfollows:

– Ragi - 93.39%

– Rice - 91.55%

Similarly when we applied our model on yearly crop-Sugarcane and observed accuracy of 72.17%. We observed little less accuracy for yearly crop since we had less data points available for this crop. The results of every predicted crop graph is included in the appendix.

CHAPTER11

CONCLUSION

As a result of penetration of technology into agriculture field, there is a marginal improvement into the productivity.
The innovations have led to new concepts like digital agriculture, smart farming, precision agriculture etc.
In the literature, it has been observed that analysis has been done on agriculture productivity, hidden patterns discovery using data set elated to seasons and crop yeilds data.
We have noticed and made analysis about different crops cultivated, area and production in different states and districts using IBM Cognos. Some of them are

1. Season with average production
2. Production by crop year
3. Production by district
4. Production by Area

CHAPTER12

FUTURESCOPE

The developed model is has data points from 1997 to 2014 of Mysore region. It is giving accuracy around 92% for seasonal and 72% for yearly crops. In future, this model can be implemented throughout the India by adding the data points for all the region. According to our analysis model will give more accuracy as the data points increases, so to get better accuracy model data points can be increased. Our system can be integrated with messaging module so that registered farmers can get the notification of the prediction directly to their registered mobile numbers.

APPENDIX

SOURCE CODE:

Import and Loading the dataset:

```
import numpy as np
np.seterr(divide='ignore', invalid='ignore')
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

```
df = pd.read_csv('crop_production.csv')
```

Data Exploration:

```
df.info()
```

```
RangeIndex: 51953 entries, 0 to 51952
Data columns (total 7 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   State_Name      51953 non-null  object 
 1   District_Name   51953 non-null  object 
 2   Crop_Year       51953 non-null  int64  
 3   Season          51953 non-null  object 
 4   Crop            51953 non-null  object 
 5   Area            51953 non-null  float64
 6   Production      51700 non-null  float64
dtypes: float64(2), int64(1), object(4)
memory usage: 2.8+ MB
```

```
df.describe()
```

	Crop_Year	Area	Production
count	51953.000000	51953.000000	5.170000e+04
mean	2005.937771	7338.703445	3.982276e+05
std	5.085025	27965.401646	1.209577e+07
min	1997.000000	0.200000	0.000000e+00
25%	2002.000000	74.000000	8.300000e+01
50%	2006.000000	426.000000	6.070000e+02
75%	2010.000000	2500.000000	5.269000e+03
max	2014.000000	877029.000000	7.801620e+08

```
len(df['State_Name'].unique())
```

```
7
```

```
len(df['Crop'].unique())
```

```
80
```

```
df.head()
```

	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

```
df.isnull().sum()
```

```
State_Name         0
District_Name      0
Crop_Year          0
Season             0
Crop               0
Area               0
Production       253
dtype: int64
```

```
df.dropna(inplace=True)
```

```
df = df[df['Production'] !=0]
```

Null Value Removed

```
df.info()
```

```
Int64Index: 51590 entries, 0 to 51952
Data columns (total 7 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   State_Name      51590 non-null  object 
 1   District_Name   51590 non-null  object 
 2   Crop_Year       51590 non-null  int64  
 3   Season          51590 non-null  object 
 4   Crop            51590 non-null  object 
 5   Area            51590 non-null  float64
 6   Production      51590 non-null  float64
dtypes: float64(2), int64(1), object(4)
memory usage: 3.1+ MB
```

```
df['Crop_Year'].unique()
```

```
array([2000, 2001, 2002, 2003, 2004, 2005, 2006, 2010, 1997, 1998, 1999,
       2007, 2008, 2009, 2011, 2012, 2013, 2014])
```

The Data has been collected from 1997-2015

```
df['State_Name'].unique()
```

```
array(['Andaman and Nicobar Islands', 'Andhra Pradesh',
       'Arunachal Pradesh', 'Assam', 'Bihar', 'Chandigarh',
       'Chhattisgarh'], dtype=object)
```

```
df['District_Name'].nunique()
```

```
116
```

```
pd.set_option('display.float_format', lambda x: '%.2f' % x)
df['Production'].sort_values(ascending = False)
```

```
2543     780162000.00
2432     729965000.00
2488     720895000.00
2378     719961050.00
9829     718991000.00
             ...
74               0.50
35               0.30
44               0.11
39899            0.10
48               0.10
Name: Production, Length: 51590, dtype: float64
```

```
df[df['Area'] == 82704.00]
```

State_Name	District_Name	Crop_Year	Season	Crop	Area	Production

```
no_of_diff_crops = df['Crop'].nunique()
types_of_crops = df['Crop'].unique()

print('There are {} different types of crops'.format(no_of_diff_crops))
print('-----------------------------------')
print('They different types of crops are :-',types_of_crops)
```

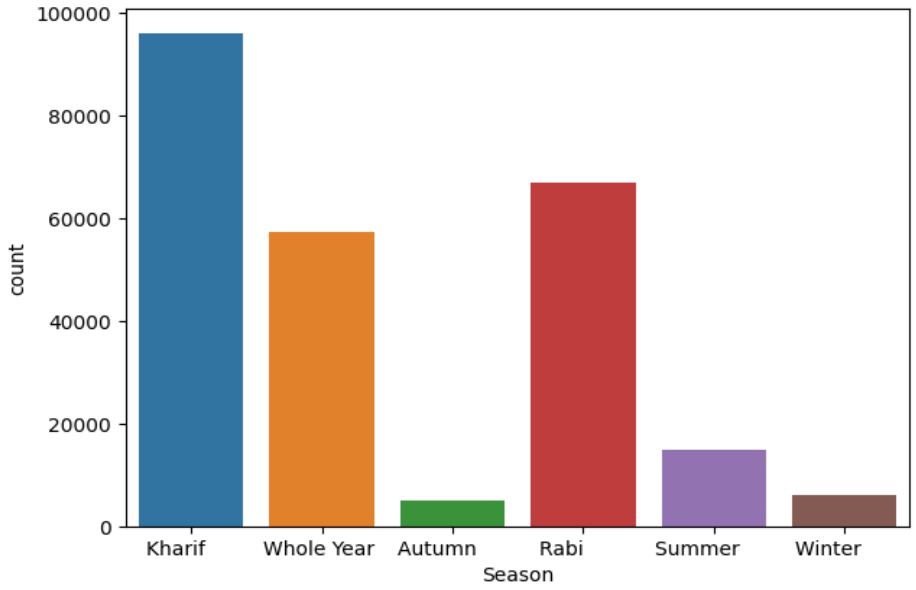
```
There are 80 different types of crops
-----------------------------------
They different types of crops are :- ['Arecanut' 'Other Kharif pulses' 'Rice' 'Banana' 'Cashewnut' 'Coconut '
 'Dry ginger' 'Sugarcane' 'Sweet potato' 'Tapioca' 'Black pepper'
 'Dry chillies' 'other oilseeds' 'Turmeric' 'Maize' 'Moong(Green Gram)'
 'Urad' 'Arhar/Tur' 'Groundnut' 'Sunflower' 'Bajra' 'Castor seed'
 'Cotton(lint)' 'Horse-gram' 'Jowar' 'Korra' 'Ragi' 'Tobacco' 'Gram'
 'Wheat' 'Masoor' 'Sesamum' 'Linseed' 'Safflower' 'Onion'
 'other misc. pulses' 'Samai' 'Small millets' 'Coriander' 'Potato'
 'Other  Rabi pulses' 'Soyabean' 'Beans & Mutter(Vegetable)' 'Bhindi'
 'Brinjal' 'Citrus Fruit' 'Cucumber' 'Grapes' 'Mango' 'Orange'
 'other fibres' 'Other Fresh Fruits' 'Other Vegetables' 'Papaya'
 'Pome Fruit' 'Tomato' 'Rapeseed &Mustard' 'Mesta' 'Cowpea(Lobia)' 'Lemon'
 'Pome Granet' 'Sapota' 'Cabbage' 'Peas  (vegetable)' 'Niger seed'
 'Bottle Gourd' 'Sannhamp' 'Varagu' 'Garlic' 'Ginger' 'Oilseeds total'
 'Pulses total' 'Jute' 'Peas & beans (Pulses)' 'Blackgram' 'Paddy'
 'Pineapple' 'Barley' 'Khesari' 'Guar seed']
```

```
df['Season'].value_counts()
```

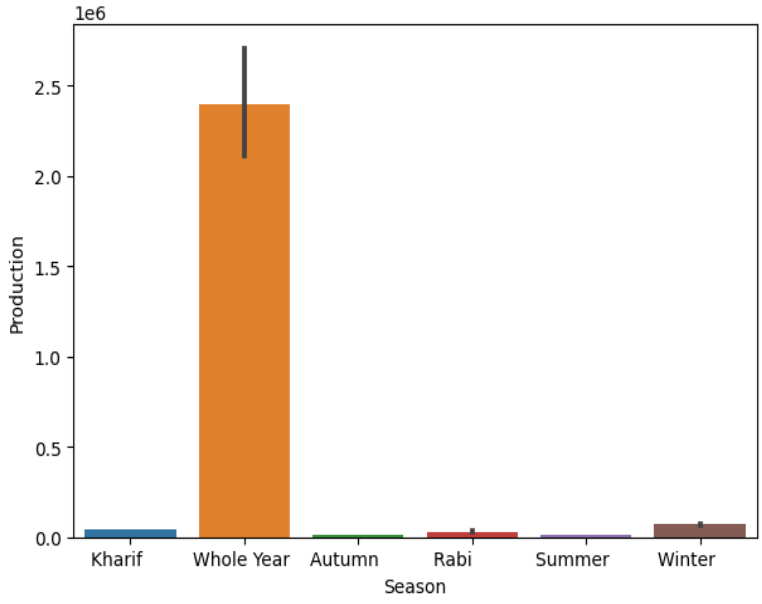
```
Kharif         17837
Rabi           16267
Whole Year     13185
Summer          2108
Autumn          1428
Winter          1128
Name: Season, dtype: int64
```

Data Visualization:

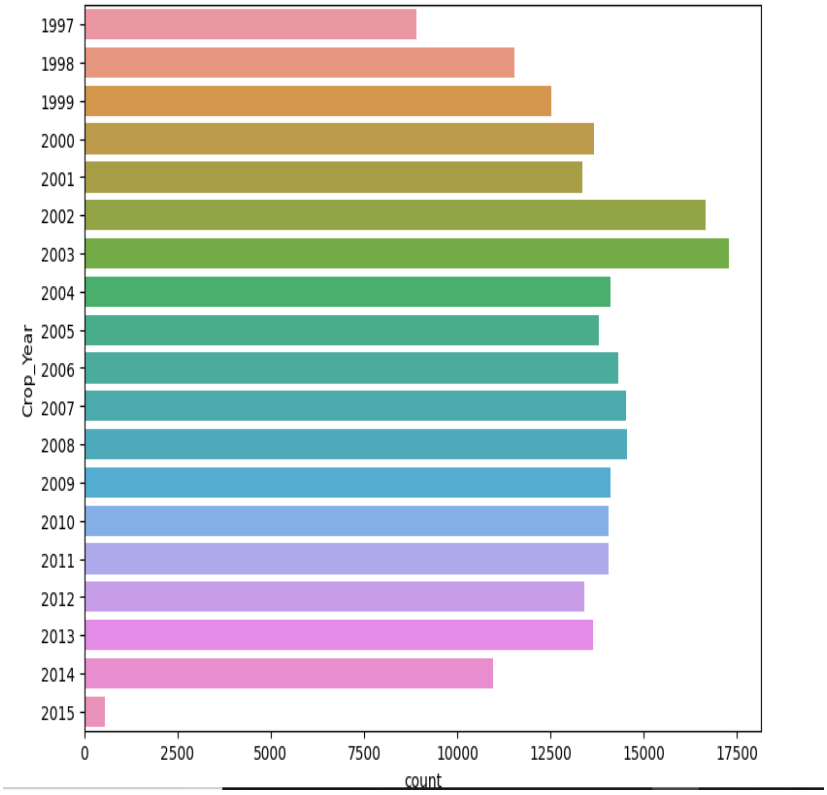
```
plt.figure(figsize=(7,5),dpi=100)
sns.countplot(data=df,x='Season');
```




```
plt.figure(figsize=(7,5),dpi=100)
sns.barplot(data=df,x='Season',y='Production');
```



```
plt.figure(figsize=(9,7),dpi=100)
sns.countplot(data=df,y='Crop_Year');
```



Most of the crop production came from Tamil Nadu let's analyze TamilNadu data

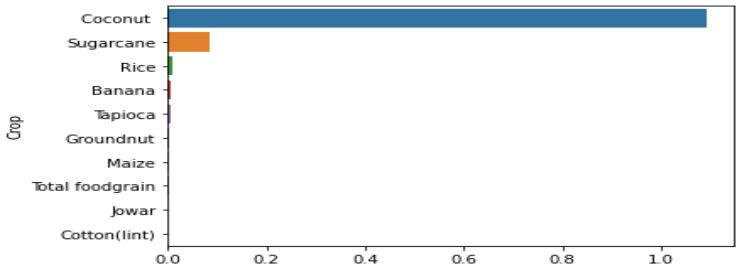
```
TamilNadu_data = df[df['State_Name'] == 'Tamil Nadu']
TamilNadu_data.head()
```

	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production
177668	Tamil Nadu	ARIYALUR	2008	Kharif	Rice	24574.0	NaN
177669	Tamil Nadu	ARIYALUR	2008	Whole Year	Arhar/Tur	209.0	NaN
177670	Tamil Nadu	ARIYALUR	2008	Whole Year	Bajra	565.0	NaN
177671	Tamil Nadu	ARIYALUR	2008	Whole Year	Banana	190.0	NaN
177672	Tamil Nadu	ARIYALUR	2008	Whole Year	Cashewnut	31113.0	NaN

```
top_prod_TN = TamilNadu_data.groupby('Crop').sum()["Production"].reset_index().sort_values(by='Production',ascending=False).nlargest(n=10,columns='Pro
top_prod_TN
```

	Crop	Production
21	Coconut	1.093774e+10
73	Sugarcane	8.474968e+08
66	Rice	1.001227e+08
5	Banana	5.871609e+07
76	Tapioca	5.564865e+07
31	Groundnut	1.893340e+07
40	Maize	1.120166e+07
79	Total foodgrain	9.121209e+06
35	Jowar	4.905140e+06
23	Cotton(lint)	4.277078e+06

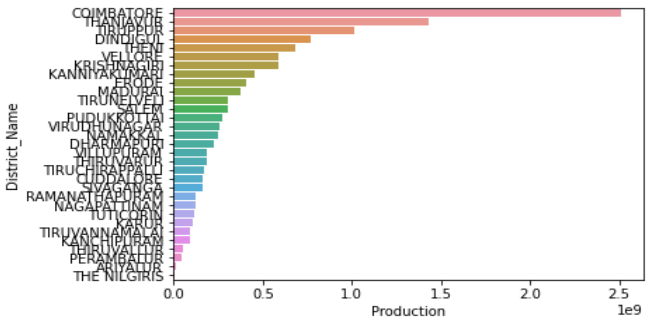
```
sns.barplot(data=top_prod_TN,y='Crop',x='Production')
```




```
TN_District = TamilNadu_data.groupby('District_Name').sum()['Production'].reset_index().sort_values(by='Production',ascending=False)
TN_District
```

	District_Name	Production
1	COIMBATORE	2.511855e+09
18	THANJAVUR	1.428293e+09
25	TIRUPPUR	1.013374e+09
4	DINDIGUL	7.673745e+08
20	THENI	6.808706e+08
28	VELLORE	5.908857e+08
9	KRISHNAGIRI	5.895962e+08
7	KANNIYAKUMARI	4.574093e+08
5	ERODE	4.078174e+08
10	MADURAI	3.736749e+08
24	TIRUNELVELI	3.076589e+08
16	SALEM	3.027947e+08
14	PUDUKKOTTAI	2.765577e+08
30	VIRUDHUNAGAR	2.591435e+08
12	NAMAKKAL	2.455126e+08
3	DHARMAPURI	2.231966e+08
29	VILLUPURAM	1.897088e+08
22	THIRUVARUR	1.896585e+08
23	TIRUCHIRAPPALLI	1.711727e+08

```
sns.barplot(data=TN_District,y='District_Name',x='Production')
```

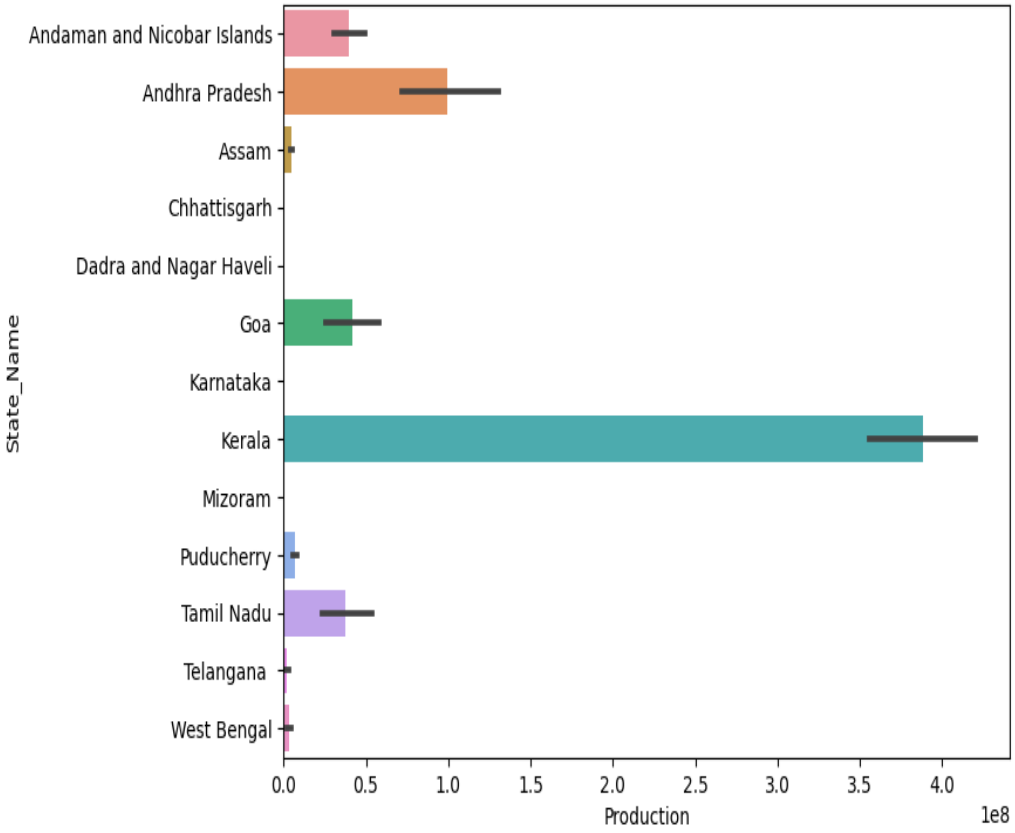


Now let's analyze the data across all the states in India

```
df.groupby('Season').sum()['Production'].nlargest()
```

```
Season
Whole Year     1.344248e+11
Kharif         4.029970e+09
Rabi           2.051688e+09
Winter         4.345498e+08
Summer         1.706579e+08
Name: Production, dtype: float64
```

```
plt.figure(figsize=(8,6),dpi=100)
sns.barplot(data=coconut_crop,x='Production',y='State_Name');
```

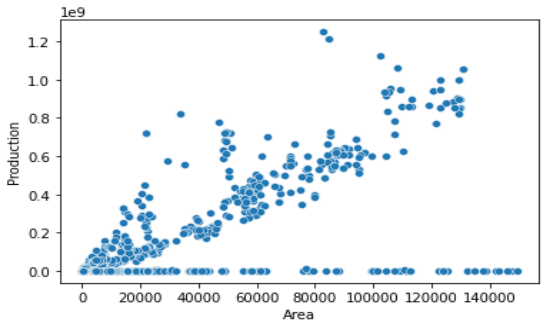


```
coconut_crop.groupby('Season').sum()['Production'].nlargest()
```

```
Season
Whole Year     1.299815e+11
Kharif         1.265920e+05
Name: Production, dtype: float64
```

```
coconut_season = coconut_crop.groupby('Season').sum()['Production'].reset_index()
px.bar(data_frame=coconut_season,x='Season',y='Production')
```

```
sns.scatterplot(data=coconut_crop,x='Area',y='Production');
```



Creating model:

```
from sklearn.model_selection import train_test_split
```

```
X = crop_data.drop('Production',axis=1)
X.head()
```

```
y = crop_data['Production']
y.head()
```

```
0   2000.00
1      1.00
2    321.00
3    641.00
4    165.00
Name: Production, dtype: float64
```

```
X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.33, random_state=42)
```

```
X_test.shape
```

```
(70378, 735)
```

```
y_test.shape
```

```
from sklearn.linear_model import LinearRegression
```

```
crop_model = LinearRegression()
```

```
crop_model.fit(X_train,y_train)
```

```
LinearRegression()
```

Prediction

```
crop_predictions = crop_model.predict(X_test)
crop_predictions
```

```
array([  591668.81926186, -1224399.18087533,  -528237.48815455, ...,
         -79121.86366799,  -152631.26284787,   162578.32881211])
```

```
crop_model.coef_
```

```
crop_model.intercept_
```

```
-33686294.63228672
```

```
predicted_crop_val = pd.DataFrame({'Actual':y_test,'Predicted':crop_predictions})
predicted_crop_val
```

	Actual	Predicted
191452	480.00	591668.82
172763	290.00	-1224399.18
81954	26248.00	-528237.49
79750	6.00	-1579489.85
193433	20.00	-1355248.90
...	...	...
216969	5.00	-836851.13
1823	2500.00	9175562.30
121894	65.00	-79121.86
200451	2.00	-152631.26
107319	160.00	162578.33

70378 rows × 2 columns

```
from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_score
```

```
df['Production'].mean()
```

```
648868.9434560126
```

```
crop_predictions.mean()
```

```
567688.9612009758
```

```
mean_absolute_error(y_test,crop_predictions)
```

```
2199779.5972323604
```

```
mean_squared_error(y_test,crop_predictions)
```

```
301754861682964.25
```

```
np.sqrt(mean_squared_error(y_test,crop_predictions))
```

```
17371092.702618457
```

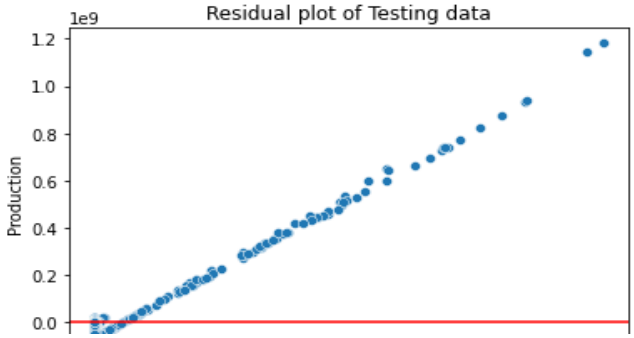
```
def mape(actual, pred):
    actual, pred = np.array(actual), np.array(pred)
    return np.mean(np.abs((actual - pred) / actual)) * 100

mape(y_test,crop_predictions)
```

```
test_residuals = y_test - crop_predictions
```

```
test_residuals
```

```
191452    -591188.82
172763    1224689.18
81954      554485.49
79750     1579495.85
193433    1355268.90
              ...
216969     836856.13
1823     -9173062.30
121894     79186.86
200451     152633.26
107319    -162418.33
Name: Production, Length: 70378, dtype: float64
```

```
sns.scatterplot(x=y_test,y=test_residuals)
plt.axhline(y=0,color='red')
plt.title('Residual plot of Testing data');
```



```
r = r2_score(y_test,crop_predictions)
print("R2score when we predict using Linear Regression is ",r)
```

```
R2score when we predict using Linear Regression is  0.18493399566622137
```

Results of training data

```
train_set_predictions = crop_model.predict(X_train)
train_set_predictions
```

```
array([-3259622.32998938,  -530500.35001556, -1254656.07860044, ...,
         587913.93084943,   474760.02963475,  1698161.198421  ])
```

```
train_set_predictions.mean()
```

```
632100.1644935672
```

```
sns.scatterplot(x=y_train,y=y_train-train_set_predictions)
plt.axhline(y=0,color='red')
plt.title('Residual plot of Training data');
```

```
X2 = crop_data2_ac.drop('Production',axis=1)
X2.head()
```

	Crop_Year	Area	State_Name_Andaman and Nicobar Islands	State_Name_Andhra Pradesh	State_Name_Arunachal Pradesh
0	2000.00	1254.00	1	0	0
1	2000.00	2.00	1	0	0
2	2000.00	102.00	1	0	0
3	2000.00	176.00	1	0	0
4	2000.00	720.00	1	0	0

5 rows × 144 columns

```
y2 = crop_data2_ac['Production']
y2.head()
```

```
0    2000.00
1       1.00
2     321.00
3     641.00
4     165.00
Name: Production, dtype: float64
```

```
from sklearn.model_selection import train_test_split
```

```
def mape(actual, pred):
    actual, pred = np.array(actual), np.array(pred)
    return np.mean(np.abs((actual - pred) / actual)) * 100

mape(y_train,train_set_predictions)
```

```
7092939.504862983
```

```
Crop_data2 = df.drop(['District_Name'],axis=1)
```

```
Crop_data2.head()
```

	State_Name	Crop_Year	Season	Crop	Area	Production
0	Andaman and Nicobar Islands	2000.00	Kharif	Arecanut	1254.00	2000.00
1	Andaman and Nicobar Islands	2000.00	Kharif	Other Kharif pulses	2.00	1.00
2	Andaman and Nicobar Islands	2000.00	Kharif	Rice	102.00	321.00
3	Andaman and Nicobar Islands	2000.00	Whole Year	Banana	176.00	641.00
4	Andaman and Nicobar Islands	2000.00	Whole Year	Cashewnut	720.00	165.00

```
crop_data2_ac = pd.get_dummies(data=Crop_data2)
crop_data2_ac.head()
```

```
X2_train, X2_test, y2_train, y2_test = train_test_split( X2, y2, test_size=0.33, random_state=42)
```

```
from sklearn.linear_model import LinearRegression
crop_model2 = LinearRegression()
crop_model2.fit(X2_train,y2_train)
```

```
LinearRegression()
```

```
#Prediction
crop2_predictions = crop_model2.predict(X2_test)
crop2_predictions
```

```
array([ -27567.15607489, -242975.82827755,  193874.62415348, ...,
       -224034.83310432, -455078.28891501,  120365.51545634])
```

```
crop_model2.coef_
```

```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
df['Production'].mean()
```

```
648868.9434560126
```

```
crop2_predictions.mean()
```

```
556512.1489000395
```

```
mean_absolute_error(y_test,crop2_predictions)
```

```
2077972.4729596092
```

```
mean_squared_error(y_test,crop2_predictions)
```

```
306965695608223.44
```

```
np.sqrt(mean_squared_error(y_test,crop2_predictions))
```

```
17520436.51306164
```

```
r2_score(y_test,crop2_predictions)
```

```
0.1708590821320356
```

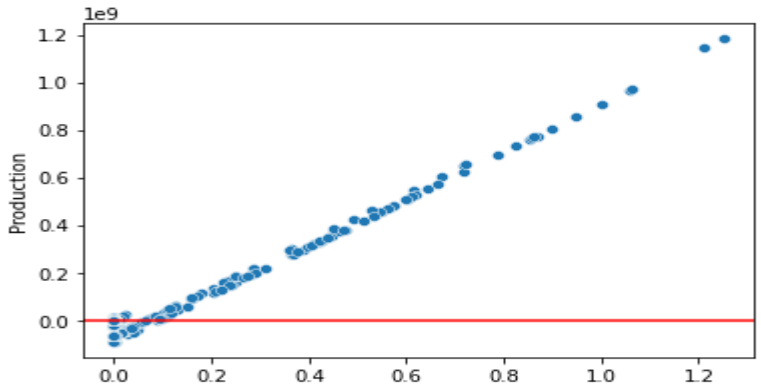
```
test2_residuals = y2_test - crop2_predictions
```

```
test2_residuals
```

```
test2_residuals
```

```
191452     28047.16
172763    243265.83
81954    -167626.62
79750    1687019.88
193433    574032.99
             ...
216969    659202.56
1823     -190376.83
121894    224099.83
200451    455080.29
107319   -120205.52
Name: Production, Length: 70378, dtype: float64
```

```
sns.scatterplot(x=y2_test,y=test2_residuals)
plt.axhline(y=0,color='red')
```



GITHUB

https://github.com/IBM-EPBL/IBM-Project-21237-1659775670

PROJECT DEMO

https://drive.google.com/drive/folders/1x41VxiGW21N-t-Qx8a6GtoxIJVDIv_pD?usp=share_link