# 6-Week Summer Internship In Data Science

## Project On
## Prediction of Agriculture Crop Production in India

**Name :** Jay Jigarkumar Soni
**Degree:** Bachelor of Computer Engineering
**College:** **A.D Patel Institute Of Technology Anand(Gujarat) India**

# Table of Content

# 1. Introduction

## 1.1 Problem Statement

The problem statement of this project is to predict agriculture crop production in India by leveraging historical data. The objective is to develop accurate and reliable models that can forecast crop yields, allowing farmers and policymakers to make informed decisions, optimize resource allocation, and address the challenges faced in agriculture production. By tackling this problem, the project aims to contribute to food security, economic stability, and sustainable agricultural practices in India.

## 1.2 Introduction

This project aims to utilize historical data on agriculture crop production in India to develop predictive models that can forecast future crop yields. By analyzing and understanding the patterns and trends present in the data, we can gain valuable insights into the factors that impact crop production in the country. With India's vast population and heavy dependence on agriculture, accurately predicting crop production can have significant implications for food security, economic planning, and sustainable agricultural practices.

The availability of comprehensive data on crop cultivation, production, quantity, variety, season, cost, and recommended zones provides a rich source of information for this project. By merging and consolidating these diverse datasets, we can create a unified and coherent dataset that captures the relevant variables and their interdependencies. This consolidated dataset will serve as the foundation for building robust prediction models that can accurately forecast crop production for future ye

## 1.3 Aim and Objectives

### Aim:

To predict agriculture crop production in India based on historical data from 2001 to 2014.

### Objective:

- To develop robust and reliable models that can accurately predict crop yields based on historical data.
- To identify key factors that significantly affect crop yields, such as farming costs, production costs, crop costs, and recommended locations.
- To provide farmers with valuable insights and forecasts, enabling them to make informed decisions on crop selection, input allocation and optimal cropping strategies.
- To provide forecasts and analyzes to policymakers for support targeted agricultural policies, allocation strategies, and risk management strategies.

# 2. Design Methodology

## 2.1 Process and Diagram

1. **Data Collection:** Obtain historical data on agriculture crop production, cultivation costs, production costs, yield, and recommended zones from reliable sources such as data.gov.in. Ensure data quality and completeness.

2. **Data Preprocessing:** Clean the collected data by handling missing values, removing duplicates, and addressing any inconsistencies or errors. Perform necessary data transformations, such as standardizing units or converting data types.

3. **Exploratory Data Analysis (EDA):** Conduct EDA to gain insights into the dataset. Analyze the distribution of variables, identify patterns, correlations, and outliers. Visualize the data using plots, charts, and graphs to understand the characteristics and relationships of the data.

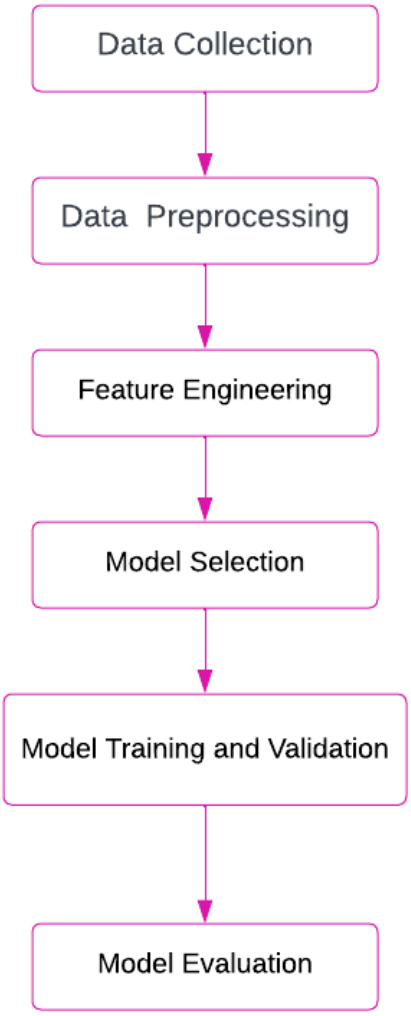
4. **Feature Engineering:** Extract and create meaningful features from the available data to enhance the predictive power of the models. This may involve feature scaling, one-hot encoding of categorical variables, creating new variables based on domain knowledge, or aggregating data at different levels.

5. **Model Selection:** Select appropriate machine learning models for predicting crop production. Consider models such as linear regression, decision trees, random forests, gradient boosting, or neural networks. Evaluate different models based on their performance metrics, interpretability, and computational efficiency.

6. **Model Training and Validation:** Split the dataset into training and validation sets. Train the selected models using the training data. Validate the models using the validation data to assess their performance and ensure they generalize well to unseen data.

7. **Model Evaluation:** Evaluate the trained models using appropriate evaluation metrics such as mean squared error, root mean squared error, or R-squared. Compare the performance of different models to select the best-performing one.

**Diagram :**


Data Collection
Data Preprocessing
Feature Engineering
Model Selection
Model Training and Validation
Model Evaluation

# 3. **Implementation**:

In [124]:
```
import pandas as pd
import numpy as np
```

In [125]:
```
df1 = pd.read_csv('datafile1.csv')
```

In [126]:
```
df1.head(7)
```

Out[126]:

| | | | | | Cost of Production (`/Quintal) C2 | Yield (Quintal/ Hectare) |
|---|---|---|---|---|---|---|
| 0 | ARHAR | Uttar Pradesh | 9794.05 | 23076.74 | 1941.55 | 9.83 |
| 1 | ARHAR | Karnataka | 10593.15 | 16528.68 | 2172.46 | 7.47 |
| 2 | ARHAR | Gujarat | 13468.82 | 19551.90 | 1898.30 | 9.59 |
| 3 | ARHAR | Andhra Pradesh | 17051.66 | 24171.65 | 3670.54 | 6.42 |
| 4 | ARHAR | Maharashtra | 17130.55 | 25270.26 | 2775.80 | 8.72 |
| 5 | COTTON | Maharashtra | 23711.44 | 33116.82 | 2539.47 | 12.69 |
| 6 | COTTON | Punjab | 29047.10 | 50828.83 | 2003.76 | 24.39 |

In [127]:
```
df1.shape
```

Out[127]: (49, 6)

In [128]:
```
df1['Crop'] = df1['Crop'].str.title()
print(df1['Crop'].head(5))
```

```
    Arhar
    Arhar
    Arhar
    Arhar
    Arhar
Name: Crop, dtype: object
```

In [129]:
```
df1.head(5)
```

Out[129]:

| | | | | | Cost of Production (`/Quintal) C2 | Yield (Quintal/ Hectare) |
|---|---|---|---|---|---|---|
| 0 | Arhar | Uttar Pradesh | 9794.05 | 23076.74 | 1941.55 | 9.83 |
| 1 | Arhar | Karnataka | 10593.15 | 16528.68 | 2172.46 | 7.47 |
| 2 | Arhar | Gujarat | 13468.82 | 19551.90 | 1898.30 | 9.59 |
| 3 | Arhar | Andhra Pradesh | 17051.66 | 24171.65 | 3670.54 | 6.42 |
| 4 | Arhar | Maharashtra | 17130.55 | 25270.26 | 2775.80 | 8.72 |

In [130]:
```
df1['Cost of Cultivation ']=df1['Cost of Cultivation (`/Hectare) C2']+df1['Cost of Cultivation (`/Hectare) A
```

In [131]:
```
f1.drop(['Cost of Cultivation (`/Hectare) A2+FL','Cost of Cultivation (`/Hectare) C2'],axis=1,inplace=True)
```

In [132]:
```
df1.rename(columns={'Cost of Production (`/Quintal) C2': 'Cost of Production', 'Yield (Quintal/ Hectare)' :
```

In [133]:
```
df1.head()
```

Out[133]:

| | Crop | State | Cost of Production | Yield (Quintal/ Hectare) | Cost of Cultivation |
|---|---|---|---|---|---|
| 0 | Arhar | Uttar Pradesh | 1941.55 | 9.83 | 32870.79 |
| 1 | Arhar | Karnataka | 2172.46 | 7.47 | 27121.83 |
| 2 | Arhar | Gujarat | 1898.30 | 9.59 | 33020.72 |
| 3 | Arhar | Andhra Pradesh | 3670.54 | 6.42 | 41223.31 |
| 4 | Arhar | Maharashtra | 2775.80 | 8.72 | 42400.81 |

In [134]:
```
df2 = pd.read_csv('datafile2.csv')
```

In [135]:
```
df2.head(5)
```

Out[135]:

| | Crop | Production 2006-07 | Production 2007-08 | Production 2008-09 | Production 2009-10 | Production 2010-11 | Area 2006-07 | Area 2007-08 | Area 2008-09 | Area 2009-10 | Area 2010-11 | Yield 2006-07 | Yield 2007-08 | Yield 2008-09 | Y 2( |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Total Foodgrains | 158.8 | 168.6 | 171.3 | 159.4 | 178.9 | 128.5 | 128.8 | 127.6 | 126.0 | 131.7 | 123.6 | 130.9 | 134.3 | 12 |
| 1 | Rice | 200.8 | 207.9 | 213.3 | 191.6 | 206.4 | 168.5 | 168.9 | 175.1 | 161.2 | 164.8 | 119.2 | 123.1 | 121.8 | 1' |
| 2 | Wheat | 131.6 | 136.4 | 140.1 | 140.3 | 150.8 | 115.0 | 115.2 | 114.0 | 116.9 | 119.5 | 114.4 | 118.4 | 122.8 | 12 |
| 3 | Jowar | 124.3 | 137.8 | 126.0 | 116.5 | 121.8 | 120.7 | 110.6 | 107.3 | 111.0 | 105.2 | 103.0 | 124.6 | 117.4 | 1( |
| 4 | Bajra | 136.4 | 161.5 | 143.9 | 105.4 | 167.9 | 94.5 | 95.1 | 87.0 | 88.5 | 95.6 | 144.3 | 169.7 | 165.4 | 1' |

In [136]:
```
df2.shape
```

Out[136]: (55, 16)

In [137]:
```
df2['Total_Production'] = df2['Production 2006-07'] + df2['Production 2007-08'] + df2['Production 2008-09']
```

In [138]:
```
df2['Total_Production'].head(5)
```

Out[138]:
```
0     837.0
1    1020.0
2     699.2
3     626.4
4     715.1
Name: Total_Production, dtype: float64
```

In [139]:
```
df2['Total_Area'] = df2['Area 2006-07'] + df2['Area 2007-08'] + df2['Area 2008-09'] + df2['Area 2009-10'] +
```

In [140]:
```
df2['Total_Area'].head(5)
```

Out[140]:
```
0    642.6
1    838.5
2    580.6
3    554.8
4    460.7
Name: Total_Area, dtype: float64
```

In [141]:
```
df2['Total_Yield'] = df2['Yield 2006-07'] + df2['Yield 2007-08'] + df2['Yield 2008-09'] + df2['Yield 2009-1
```

In [142]:
```
df2['Total_Yield'].head(5)
```

Out[142]:
```
0
1    651.2
2    608.2
3    601.9
4    565.8
     774.2
Name: Total_Yield, dtype: float64
```

In [143]:
```
df21=df2[['Crop','Total_Production','Total_Area','Total_Yield']]
df21.head(5)
```

Out[143]:

| | Crop | Total_Production | Total_Area | Total_Yield |
|---|---|---|---|---|
| 0 | Total Foodgrains | 837.0 | 642.6 | 651.2 |
| 1 | Rice | 1020.0 | 838.5 | 608.2 |
| 2 | Wheat | 699.2 | 580.6 | 601.9 |
| 3 | Jowar | 626.4 | 554.8 | 565.8 |
| 4 | Bajra | 715.1 | 460.7 | 774.2 |

In [144]:
```
df3 = pd.read_csv('datafile3.csv')
```

In [145]:
```
df3.drop(['Unnamed: 4'],axis=1,inplace=True)
```

In [146]:
```
df3.sample(5)
```

Out[146]:

| | Crop | Variety | Season/ duration in days | Recommended Zone |
|---|---|---|---|---|
| 73 | Mesta | SHRESTHA (JRM-5) | - | Andhra Pradesh, Orissa, Assam, Maharashtra, Bi... |
| 64 | Napier Bajra Hybrid | Phule Jaywant (RBN-13) | NaN | Madhya Pradesh, Maharashtra, Gujarat, Southern... |
| 39 | Linseed | JLS-67 (Shival) | 114 | Bundelkhand part of Uttar Pradesh, Madhya Prad... |
| 61 | Sugarcane | Karan 6 (Co 0239) | NaN | Punjab, Haryana, Rajasthan, Uttarakhand, Centr... |
| 41 | Linseed | JLS-73 (SLS-73) | NaN | Madhya Pradesh, Rajasthan and Bundelkhand of U... |

In [147]:
```
df3.shape
```

Out[147]:
```
(78, 4)
```

In [148]:
```
df4 = pd.read_csv('datafile4.csv')
```

In [149]:
```
df4.head(5)
```

Out[149]:

| | Crop | 2004-05 | 2005-06 | 2006-07 | 2007-08 | 2008-09 | 2009-10 | 2010-11 | 2011-12 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | Rice | 100.0 | 101.0 | 99.0 | 105.0 | 112.0 | 121.0 | 117.0 | 110.0 |
| 1 | Wheat | 100.0 | 101.0 | 112.0 | 115.0 | 117.0 | 127.0 | 120.0 | 108.0 |
| 2 | Coarse Cereals | 100.0 | 107.0 | 110.0 | 115.0 | 113.0 | 123.0 | 122.0 | 136.0 |
| 3 | Pulses | 100.0 | 108.0 | 134.0 | 124.0 | 124.0 | 146.0 | 137.0 | 129.0 |
| 4 | Vegetables | 100.0 | 109.0 | 103.0 | 118.0 | 113.0 | 124.0 | 128.0 | 115.0 |

In [150]:
```
df4.shape
```

Out[150]:
```
(13, 9)
```

In [151]:
```
df5 = pd.read_csv('produce.csv')
```

[152]:
```
df5.head(5)
```

Out[152]:

| | Particulars | Frequency | Unit | 3-1993 | 3-1994 | 3-1995 | 3-1996 | 3-1997 | 3-1998 | 3-1999 | ... | 3-2005 | 3-2006 | 3-2007 | 3-2008 | 3-200 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Agricultural Production Foodgrains | Annual, Ending mar Of Each Year | mn Ton mn | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 198.36282 | 208.6016 | 217.28212 | 230.77504 | 234.46617 |
| 1 | Agricultural Production Foodgrains Kharif | Annual, Ending mar Of Each Year | Ton mn | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 103.30942 | 109.8734 | 110.57622 | 120.95724 | 118.13857 |
| 2 | Agricultural Production Foodgrains Rabi | Annual, Ending mar Of Each Year | Ton mn | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 95.05340 | 98.7282 | 106.70590 | 109.81780 | 116.32760 |
| 3 | Agricultural Production Foodgrains Rice | Annual, Ending mar Of Each Year | Ton mn | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 83.13170 | 91.7934 | 93.35530 | 96.69290 | 99.18250 |
| 4 | Agricultural Production Foodgrains Rice Kharif | Annual, Ending mar Of Each Year | Ton | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 72.23000 | 78.2719 | 80.17080 | 82.65940 | 84.90820 |

5 rows × 25 columns

In [153]:
```
df5.shape
```

Out[153]: (429, 25)

In [154]:
```
merged_df = pd.merge(df1, df21, on='Crop')
```

In [174]:
```
merged_df.head()
```

Out[174]:

| | Crop | State | Yield (Quintal/ Hectare) | Total_Production | Total_Area | Total_Yield | Variety | Season/ duration in days | Recommended Zone | cost |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Groundnut | Karnataka | 4.71 | 826.8 | 650.1 | 631.1 | Girnar - 3 (PBS 12160) | 108 | West Bengal, Orissa and Manipur under Kharif r... | 34445.31 |
| 1 | Groundnut | Karnataka | 4.71 | 826.8 | 650.1 | 631.1 | Kadiri Harithandhra (K 1319) | 122 | Karnataka and Maharashtra under timely sown ir... | 34445.31 |
| 2 | Groundnut | Karnataka | 4.71 | 826.8 | 650.1 | 631.1 | GPBD 5 | 105-110 | Jharkhand and Manipur in Kharif Season. | 34445.31 |
| 3 | Groundnut | Andhra Pradesh | 11.97 | 826.8 | 650.1 | 631.1 | Girnar - 3 (PBS 12160) | 108 | West Bengal, Orissa and Manipur under Kharif r... | 54218.53 |
| 4 | Groundnut | Andhra Pradesh | 11.97 | 826.8 | 650.1 | 631.1 | Kadiri Harithandhra (K 1319) | 122 | Karnataka and Maharashtra under timely sown ir... | 54218.53 |

In [156]:
```
merged_df = pd.merge(merged_df, df3, on='Crop')
```

In [157]:
```
merged_df.head()
```

Out[157]:

| | Crop | State | Cost of Production | Yield (Quintal/ Hectare) | Cost of Cultivation | Total_Production | Total_Area | Total_Yield | Variety | Season/ duration in days | Recomme |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Groundnut | Karnataka | 3484.01 | 4.71 | 30961.30 | 826.8 | 650.1 | 631.1 | Girnar - 3 (PBS 12160) | 108 | West Be Oriss Manipur Kha |
| 1 | Groundnut | Karnataka | 3484.01 | 4.71 | 30961.30 | 826.8 | 650.1 | 631.1 | Kadiri Harithandhra (K 1319) | 122 | Karnatak Mahara under sov |
| 2 | Groundnut | Karnataka | 3484.01 | 4.71 | 30961.30 | 826.8 | 650.1 | 631.1 | GPBD 5 | 105-110 | Jharkhan Mani Kharif Se |
| 3 | Groundnut | Andhra Pradesh | 2554.91 | 11.97 | 51663.62 | 826.8 | 650.1 | 631.1 | Girnar - 3 (PBS 12160) | 108 | West Be Oriss Manipur Kha |
| 4 | Groundnut | Andhra Pradesh | 2554.91 | 11.97 | 51663.62 | 826.8 | 650.1 | 631.1 | Kadiri Harithandhra (K 1319) | 122 | Karnatak Mahara under sov |

In [158]:
```
#merged_df = pd.merge(merged_df, df5, left_on='Crop', right_on='Particulars')
```

In [159]:
```
merged_df['cost']=merged_df['Cost of Cultivation ']+merged_df['Cost of Production']
```

In [161]:
```
merged_df.drop(['Cost of Cultivation ','Cost of Production'],axis=1, inplace=True)
```

In [162]:
```
merged_df.head()
```

Out[162]:

| | Crop | State | Yield (Quintal/ Hectare) | Total_Production | Total_Area | Total_Yield | Variety | Season/ duration in days | Recommended Zone | cost |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Groundnut | Karnataka | 4.71 | 826.8 | 650.1 | 631.1 | Girnar - 3 (PBS 12160) | 108 | West Bengal, Orissa and Manipur under Kharif r... | 34445.31 |
| 1 | Groundnut | Karnataka | 4.71 | 826.8 | 650.1 | 631.1 | Kadiri Harithandhra (K 1319) | 122 | Karnataka and Maharashtra under timely sown ir... | 34445.31 |
| 2 | Groundnut | Karnataka | 4.71 | 826.8 | 650.1 | 631.1 | GPBD 5 | 105-110 | Jharkhand and Manipur in Kharif Season. | 34445.31 |
| 3 | Groundnut | Andhra Pradesh | 11.97 | 826.8 | 650.1 | 631.1 | Girnar - 3 (PBS 12160) | 108 | West Bengal, Orissa and Manipur under Kharif r... | 54218.53 |
| 4 | Groundnut | Andhra Pradesh | 11.97 | 826.8 | 650.1 | 631.1 | Kadiri Harithandhra (K 1319) | 122 | Karnataka and Maharashtra under timely sown ir... | 54218.53 |

In [166]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

In [167]:
```
features = ['Total_Area', 'Total_Yield', 'cost']
target = 'Total_Production'
```

In [170]:
```
model = LinearRegression()
model.fit(X_train, y_train)
```

Out[170]:
```
LinearRegression()
```

In [168]:

```
X = merged_df[features]
y = merged_df[target]
```

In [169]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [171]:
```
y_pred = model.predict(X_test)
```

In [172]:
```
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
```

In [173]:
```
print("Mean Squared Error:", mse)
print("R-squared:", r2)
```

```
Mean Squared Error: 7.575233571087326
R-squared: 0.9998599436818815
```

## 4. Lesson Learned

Throughout the project, valuable lessons I learned:

1. **Data Preprocessing is Essential:** Cleaning and preprocessing the data is a crucial step that significantly impacts the quality and reliability of the analysis. Devoting time to thoroughly clean the data and handle missing values ensures accurate predictions.

2. **Feature Engineering:** Transforming the raw data into meaningful features allows the models to capture the essential patterns and relationships in the data.

3. **Exploratory Data Analysis (EDA):** Gain insights into the dataset. Analyze the distribution of variables, identify patterns, correlations, and outliers.

4. **Model Evaluation and Selection**: The models were evaluated based on their performance metrics, interpretability, and computational efficiency.