🌾 Exploratory Analysis of Rainfall Data in India for Agriculture

Project Description

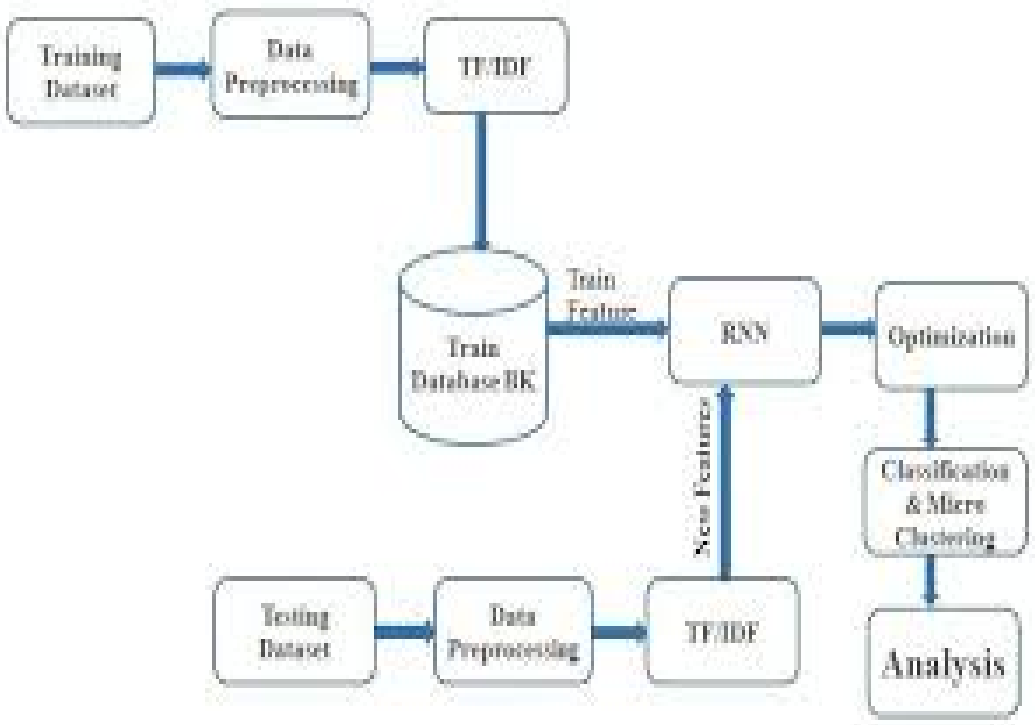
Rainfall is one of the most critical factors influencing agricultural productivity in India. Since a large portion of Indian agriculture depends on monsoon rainfall, understanding rainfall patterns is essential for crop planning, irrigation management, and food security.

Rainfall varies significantly across different states and across years. By analyzing historical rainfall data, meaningful insights can be extracted regarding seasonal trends, regional distribution, and long-term variability.

This project focuses on performing Exploratory Data Analysis (EDA) on Indian rainfall data to identify patterns, variations, and trends that can support agricultural planning and decision-making.

Machine Learning techniques may further be used to analyze and model rainfall behavior.

Technical Architecture



The project consists of:

- **Dataset Collection from Kaggle**
  The rainfall dataset was collected from Kaggle, which provides publicly available datasets for data analysis and research purposes.

  Dataset link:https://www.kaggle.com/datasets/rajanand/rainfall-in-india

- **Data Loading and Inspection**
  The dataset is loaded into the environment using Python libraries such as Pandas. Initial inspection is performed using functions like `head()`, `info()`, and `describe()` to understand structure, data types, and summary statistics.

- **Exploratory Data Analysis (EDA)**
  EDA is conducted to identify rainfall trends across different states and years. Patterns, variations, and extreme values are analyzed to understand climatic behavior.

- **Data Preprocessing**
  Missing values are handled, column names are standardized, and relevant features are selected to prepare the dataset for further analysis and modeling.

- **Visualization of Rainfall Patterns**
  Graphs and charts such as line plots and bar charts are used to visualize rainfall variations across states and years. These visualizations help identify highest and lowest rainfall regions.

- **Optional Machine Learning Modeling**
  Predictive models can be applied to estimate rainfall based on historical data, supporting agricultural forecasting.

- **Flask-Based Web Visualization (If Integrated)**
  A Flask web application is used to create an interactive interface where users can input rainfall-related parameters and view prediction results.

Pre-Requisites

Software Required:

- Anaconda Navigator

- Jupyter Notebook / VS Code

Python Packages:

Install using Anaconda Prompt:

```
# libraries required
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn import preprocessing
from sklearn import model_selection
from sklearn import metrics
from sklearn import linear_model
from sklearn import ensemble
from sklearn import tree
from sklearn import svm
import xgboost
```

Project Objectives

By completing this project:

- Understand rainfall trends across Indian states
- Perform univariate, bivariate, and multivariate analysis
- Identify states with highest and lowest rainfall
- Observe year-to-year rainfall variations
- Derive agricultural insights from rainfall data
- Develop a structured EDA workflow

Project Flow

1. Dataset collection
2. Loading and understanding data
3. Performing exploratory data analysis
4. Identifying rainfall trends
5. Drawing conclusions for agriculture

Milestone 1: Data Collection

Rainfall dataset is collected from Kaggle.

Dataset contains:

- State/Union Territory
- Rainfall data from 2004–05 to 2021–22
- Year-wise rainfall measurements in millimeters

Dataset is downloaded using:

```
import kagglehub

path = kagglehub.dataset_download("kushajmallick/rainfall-india")
print("Path:", path)
```

Milestone 2: Loading and Understanding the Dataset

```
import pandas as pd

df = pd.read_csv("rainfall.csv")
df.head()
```

In [4]:
```
import os

os.listdir(path)
```

Out[4]:
```
['rain-india-updated.csv']
```

In [6]:
```
import pandas as pd

df = pd.read_csv(os.path.join(path, "rain-india-updated.csv"))
df.head()
```

Out[6]:

	State/Union Territory	2004-05	2005-06	2006-07	2007-08	2008-09	2009-10	2010-11	2011-12	2012-13	2013-14	2014-15	2015-16	2016-17	2017-18	2018-19	2019-20	2020-21	2021-22
0	Andhra Pradesh	873.8	1221.6	1159.6	1099.4	1107.5	790.5	1712.5	861.8	1318.3	1120.4	875.1	1011.0	909.0	892.7	663.8	899.2	738.2	1148.9
1	Arunachal Pradesh	2545.8	2335.5	2259.4	3020.6	2244.5	1750.0	2855.6	2193.6	3440.5	2042.9	2403.1	2767.5	2706.8	2745.5	2032.5	2433.3	1943.7	2083.8
2	Assam	2994.2	2468.2	1898.7	2752.7	2339.6	2068.2	2711.4	1743.5	2609.4	1816.5	2206.2	2471.0	2266.7	2711.6	1807.1	2084.7	1651.1	2083.8
3	Bihar	1147.6	907.8	1052.8	1600.1	1197.6	889.5	629.2	1097.3	1032.4	1069.8	1061.0	872.7	1158.0	1112.0	860.6	1194.7	1272.5	1512.7
4	Chhattisgarh	1144.7	1287.3	1317.8	1281.2	1108.6	956.7	1306.6	1302.6	1377.4	1420.0	1278.6	1117.7	1315.9	1124.5	1211.9	1420.3	1234.3	1309.6

The dataset contains rainfall information for states such as:

- Andhra Pradesh
- Tamil Nadu
- Kerala
- Goa
- Haryana
- Delhi
- Rajasthan
- Punjab
- Maharashtra
- Karnataka

Each column represents rainfall data for a specific year.

Milestone 3: Exploratory Data Analysis (EDA)

EDA is performed to understand rainfall behavior.

1️⃣ Descriptive Analysis

```
df.info()
df.describe()
```

```
]: data.describe() # gives the descriptive statistics of the data
```

]:

	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustSpeed	WindSpeed9am	WindSpeed3pm	Humidity9am	Humidity3p
count	143975.000000	144199.000000	142199.000000	82670.000000	75625.000000	135197.000000	143693.000000	142398.000000	142806.000000	140953.0000
mean	12.194034	23.221348	2.360918	5.468232	7.611178	40.035230	14.043426	18.662657	68.880831	51.5391
std	6.398495	7.119049	8.478060	4.193704	3.785483	13.607062	8.915375	8.809800	19.029164	20.7959
min	-8.500000	-4.800000	0.000000	0.000000	0.000000	6.000000	0.000000	0.000000	0.000000	0.0000
25%	7.600000	17.900000	0.000000	2.600000	4.800000	31.000000	7.000000	13.000000	57.000000	37.0000
50%	12.000000	22.600000	0.000000	4.800000	8.400000	39.000000	13.000000	19.000000	70.000000	52.0000
75%	16.900000	28.200000	0.800000	7.400000	10.600000	48.000000	19.000000	24.000000	83.000000	66.0000
max	33.900000	48.100000	371.000000	145.000000	14.500000	135.000000	130.000000	87.000000	100.000000	100.0000

Provides:

- Data types
- Missing values
- Mean rainfall
- Minimum and maximum rainfall
- Standard deviation

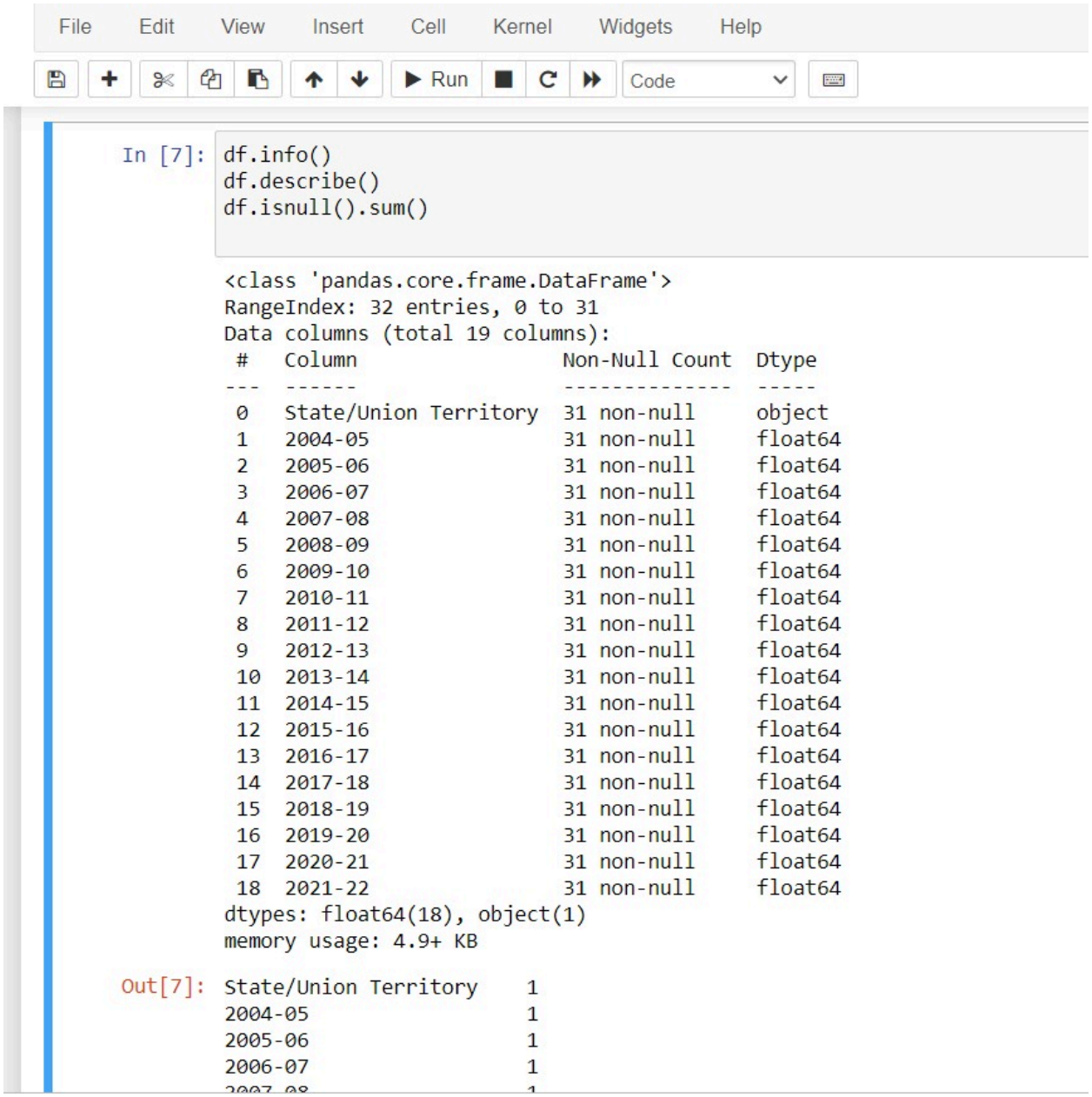
This helps in understanding overall rainfall distribution.

2️⃣ Checking Missing Values

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

Missing values are handled using:

```
df.fillna(df.mean(), inplace=True)
```



---

3 Univariate Analysis

Univariate analysis studies single features.

Example: Rainfall distribution for a particular state.

```
import matplotlib.pyplot as plt
```

```
ap_data = df[df['State/Union Territory'] == 'Andhra Pradesh']
ap_data.iloc[:, 1:].T.plot()
plt.title("Rainfall Trend - Andhra Pradesh")
plt.show()
```

In [10]:
```
df.columns = df.columns.str.strip()

# Convert wide format to long format
df_long = df.melt(id_vars='State/Union Territory',
                  var_name='Year',
                  value_name='Rainfall')

df_long.head()
```

Out[10]:

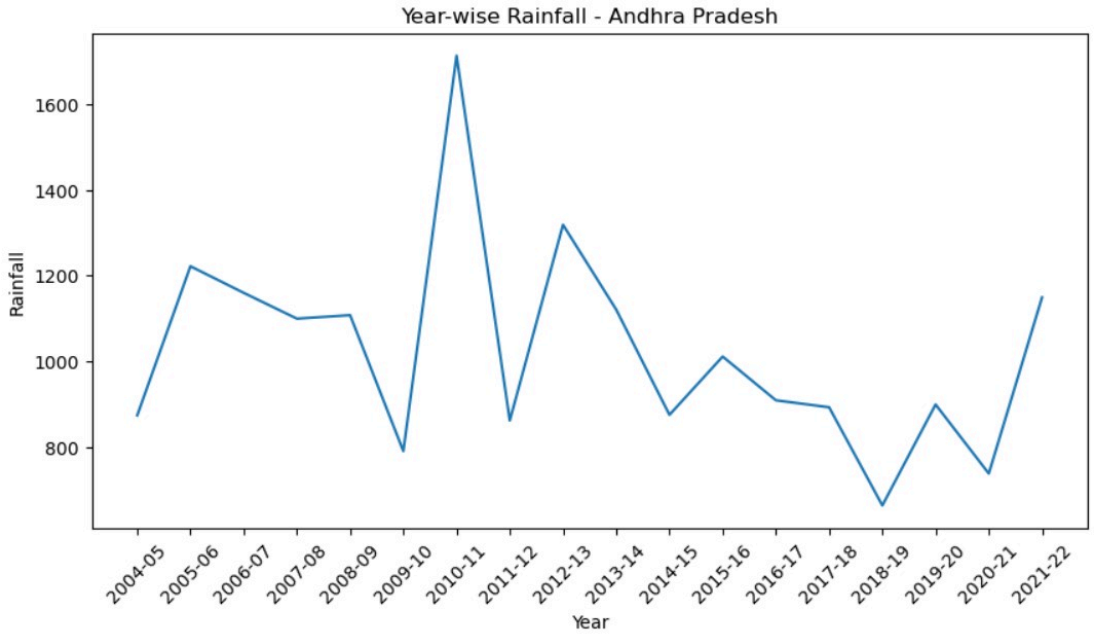
	State/Union Territory	Year	Rainfall
0	Andhra Pradesh	2004-05	873.8
1	Arunachal Pradesh	2004-05	2545.8
2	Assam	2004-05	2994.2
3	Bihar	2004-05	1147.6
4	Chhattisgarh	2004-05	1144.7

In [11]:
```
state_data = df_long[df_long['State/Union Territory'] == 'Andhra Pradesh']

import matplotlib.pyplot as plt

plt.figure(figsize=(10,5))
plt.plot(state_data['Year'], state_data['Rainfall'])
plt.xticks(rotation=45)
plt.title("Year-wise Rainfall - Andhra Pradesh")
plt.xlabel("Year")
plt.ylabel("Rainfall")
plt.show()
```

```
plt.ylabel("Rainfall")
plt.show()
```

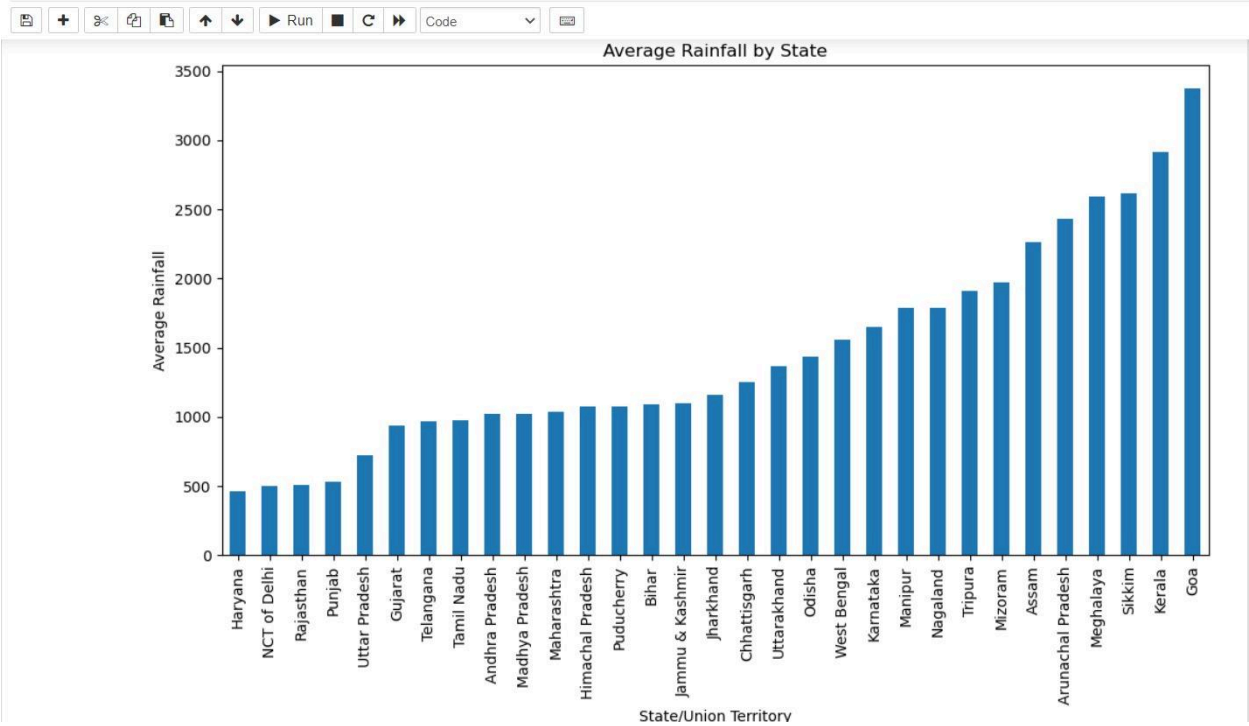


Observation:

- Rainfall fluctuates year to year.
- Lower rainfall observed between certain years.
- Higher rainfall observed in recent years.
- Trend variation impacts crop planning.

4️⃣ State-Wise Comparison

```
df.set_index('State/Union
Territory').mean(axis=1).sort_values().plot(kind='bar')
plt.title("Average Rainfall by State")
plt.show()
```



Observation:

- Goa shows highest rainfall (~3300 mm).
- Northern states show comparatively lower rainfall.
- Southern and coastal states show higher rainfall levels.

5️⃣ Correlation Analysis (Multivariate)

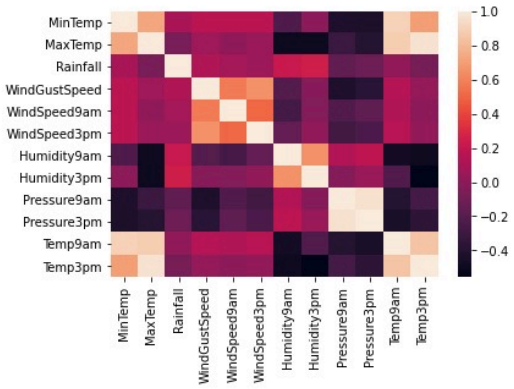
```
import seaborn as sns

plt.figure(figsize=(10,8))
sns.heatmap(df.corr())
plt.title("Correlation Between Years")
plt.show()
```

```
In [12]: cor = data.corr()

In [13]: sns.heatmap(data=cor,xticklabels=cor.columns.values,yticklabels=cor.columns.values)

Out[13]: <AxesSubplot:>
```



This shows relationship between rainfall across different years.

High correlation indicates similar rainfall trends between years.

Agricultural Insights

From the analysis:

- Coastal states receive higher rainfall, suitable for paddy cultivation.
- Northern states with lower rainfall require irrigation support.
- Year-to-year rainfall fluctuation affects crop yield stability.
- Understanding rainfall variability helps in crop selection and planning.

Optional: Model Building

Rainfall prediction model can be built using Random Forest Regressor.

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

```
from sklearn.ensemble import RandomForestRegressor

X = df.drop("2021-22", axis=1)
y = df["2021-22"]

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

model = RandomForestRegressor()
model.fit(X_train, y_train)
```

Model can be saved using:

```
import pickle

with open("rainfall_model.pkl", "wb") as f:
    pickle.dump(model, f)
```

Web Application Integration

Project Structure:

```
Rainfall_Project/
│
├── app.py
├── rainfall_model.pkl
├── templates/
│   ├── home.html
│   ├── predict.html
│   └── result.html
```

Application allows user to:

- Enter state and rainfall details

- Submit input
- View predicted rainfall

Run using:

```
python app.py
```

Final Outcome

This project successfully performs exploratory analysis of rainfall data across Indian states.

The analysis provides:

- Regional rainfall comparison
- Year-wise rainfall trends
- Agricultural insights
- Data-driven understanding of monsoon variability

This study supports better agricultural planning, irrigation management, and climate awareness.

THANK YOU