📝 **Brief Report – Crop Recommendation Using IBM Watsonx.ai (Custom Dataset)**

---

**1. Introduction**

This project involved developing a Crop Recommendation System using IBM Watsonx.ai's AutoAI, applied to a dataset containing real-world agricultural parameters like season, state, area, pesticide usage, and annual rainfall. Instead of basic soil data, we used a richer dataset to make the predictions more region- and season-specific.

---

**2. Objective**

To build a machine learning model that:

Predicts the most suitable crop to grow based on seasonal and regional agricultural features.

Uses AutoAI to automatically generate pipelines.

Deploys the best pipeline as an online API for live testing.

---

**3. Dataset Description**

The dataset used had the following columns:

Column Description

State Indian state/region (e.g., Telangana, Bihar)

Season Season of cultivation (e.g., Kharif, Rabi)

Area Total cultivation area in hectares

Pesticides Quantity of pesticide used (in kg or liters)

Rainfall Annual rainfall in millimeters

Crop Crop grown (target/output label)

This data represents real agricultural inputs used to guide crop selection for a specific state and season.

---

**4. Tools and Platform Used**

IBM Cloud

Watsonx.ai Studio (AutoAI)

Custom Dataset (CSV format)

AutoAI Runtime for training, evaluation, and deployment

**5. Implementation Process**

Step 1: IBM Cloud Setup

Logged in to cloud.ibm.com using our IBM Academic account.

Accessed the Watsonx.ai Studio service via the dashboard.

Step 2: New Project Creation

Created a new project titled Smart Crop Selector.

Linked it with the Watsonx.ai Runtime (Free plan).

**Step 3: AutoAI Experiment Setup**

Launched a new AutoAI experiment.

Uploaded our custom dataset.

Selected “Crop” as the prediction target.

Chose No for time-series analysis.

Let AutoAI run and generate multiple pipelines.

Step 4: Evaluation of Pipelines

Viewed the pipeline leaderboard.

Pipeline 3 (example) was the best performer based on accuracy and confidence.

Verified metrics like training accuracy, precision, and model type used.

Step 5: Saving and Promoting Model

Saved the top pipeline as a Model Asset.

Created a Deployment Space named CropDeploymentSpace.

Promoted the model to the deployment space.

Step 6: Model Deployment and Testing

Created an Online Deployment of the model.

Opened the Test tab to input new values like:

State = Karnataka, Season = Kharif, Area = 220.5, Pesticides = 18.5, Rainfall = 910 mm

The model predicted the crop as Maize with over 95% confidence.

---

**6. What We Actually Did**

Chose a more detailed, domain-relevant dataset than the one suggested in the sample.

Understood how different features like season, region, and pesticide use affect crop selection.

Handled categorical columns like "State" and "Season" during AutoAI pre-processing.

Used AutoAI's pipeline automation to avoid manual ML model selection.

Deployed the model as an online API endpoint for real-time usage.

Successfully tested it with unseen inputs for accurate predictions.

---

**7. Observations & Learnings**

Categorical variable handling (like State, Season) was automatically managed by AutoAI.

Even without manual coding, we got highly accurate predictions.

Cloud deployment allowed testing on different hypothetical inputs instantly.

This approach saves time and helps decision-makers in agriculture choose suitable crops by inputting real data.

---

**8. Conclusion**

This project demonstrates how a smart, no-code AI system like Watsonx.ai can help revolutionize agriculture through predictive modeling. The use of regional and seasonal data makes the solution practical for real-world crop planning. We gained hands-on experience in building, deploying, and testing AI models in a cloud environment.

---

