1. Crop: The name of the crop cultivated.
2. Crop\_Year: The year in which the crop grew.
3. Season: The specific cropping season (e.g., Kharif, Rabi, Whole Year).
4. State: The Indian state where the crop was cultivated.
5. Area: The total land area (in hectares) under cultivation for the specific crop.
6. Production: The quantity of crop production (in metric tons).
7. Annual\_Rainfall: The annual rainfall received in the crop-growing region (in mm).
8. Fertilizer: The total amount of fertilizer used for the crop (in kilograms).
9. Pesticide: The total amount of pesticide used for the crop (in kilograms).
10. Yield: The calculated crop yield (production per unit area).

OB\_DT=Observation\_date,

PMAIZMT= Price of Maize per Metric Tonn/USD(MedianValue)

Why can't we use Kmeans, with PCA or KNN type model?

Why Not K-Means (Clustering) for This Task?

* K-Means is for Unsupervised Learning, while your task (predicting corn price/yield) is a Supervised Learning problem (i.e., we have labeled data with known targets).
* K-Means groups data points into clusters but does not predict continuous values like price or yield.
* Even if we use PCA (Principal Component Analysis) to reduce dimensionality, it won’t help with forecasting because PCA doesn’t handle time dependencies.

Why Not KNN (K-Nearest Neighbors)?

* KNN is a lazy learning model, meaning it makes predictions based on the nearest neighbors.
* It works well for classification or simple regression but does not capture time-dependent patterns well.
* Time series data has trends and seasonality, which KNN does not consider, it treats all data points as independent.

What about supporting vector machines, linear programming, Markov decision process?

Support Vector Machines (SVM) – Not Ideal for Time Series

* SVM is mainly used for classification and regression tasks where the data points are independent.
* Time-series data has autocorrelation (past values affect future values), which SVM doesn’t handle well.
* It doesn’t naturally account for time-based dependencies, trends, or seasonality.
* Computationally expensive for large datasets.

When Can SVM Be Used?

* If we engineer features like lagged values or rolling averages, then SVM regression (SVR) might work, but it’s usually not as effective as specialized time-series models like ARIMA, Prophet, or XGBoost.

Linear Programming – Not for Forecasting

* Linear programming is used for optimization problems (e.g., minimizing cost, maximizing efficiency).
* It does not predict future values—instead, it finds the best decision given constraints.
* Example: You could use Linear Programming to optimize crop yield based on available resources, but not to forecast corn prices/yield.

When Is It Useful?

* If your goal is resource allocation or supply chain optimization, Linear Programming (e.g., using scipy.optimize) can be a great tool!

Markov Decision Process (MDP) – Only for Decision-Based Problems

* MDP is a reinforcement learning technique, useful for decision-making problems where an agent interacts with an environment over time.
* It models state, action, reward, and transition probabilities—but it doesn’t predict continuous values like price or yield.

When Is It Useful?

* If you wanted to build a policy-based system (e.g., a model that decides when to plant or harvest crops based on market prices), MDP would be helpful.