SMART INDIA HACKATHON

Problem Statement ID : SIH1647

Problem Statement Title : Development of AI-ML based models for predicting prices of agri-horticultural commodities such as pulses and vegetable (onion, potato, etc.)

Theme : Miscellaneous

PS Category: Software

Team ID: 4279

Team Name: CTRL K

Ctrl K

1

KRUSHI JYOTHISHI

SMART INDIA HACKATHON 2024

Optimizing Prediction of Horticultural crop prices with ML Models

Leveraging Machine Learning to Improve Price Prediction and Market Decision-Making.

PURPOSE:

To ensure stability in essential food commodity prices by accurately forecasting demand and adjusting buffer stock levels accordingly.

KEY CHALLENGE:

Inefficiencies in buffer stock management and market interventions due to limitations in current ARIMA model for price prediction.

Model	Strengths	Weaknesses	Suitable For
NHITS	Handles hierarchical time series well; good accuracy.	Computationally expensive; requires hyperparameter tuning.	Hierarchical time series forecasting (e.g., sales across regions/categories)
LSTM	Captures long-range dependencies; robust to vanishing gradients; handles non-linearity.	Computationally expensive; requires large datasets; difficult interpretation.	Time series with long-term dependencies (e.g., stock prices, weather)
GRU	Similar to LSTM but computationally less expensive.	May not always outperform LSTMs; interpretation can be challenging.	Time series with long-term dependencies; computationally constrained scenarios
N-BEATS	Interpretable; high accuracy; handles various time series types.	Computationally expensive for very long series; interpretation not fully straightforward.	High accuracy and some interpretability needed; various time series problems
SARIMA	Simple; well-established; accurate for stationary time series and to handle seasonality.	Assumes stationarity; requires pre-processing; limited for non-linear patterns; requires careful parameter selection.	Stationary time series; simple forecasting; limited data; interpretability key; Time series with clear seasonal patterns.

Why not ARIMA?

- ❖ Talks only about linear aspects of a feature
- ❖ Cant handle long-term Dependencies
- ❖ Sensitive to outliers and noise
- ❖ Inability to capture Seasonal Variations and External Factors

Why ML and Neural Networks?

- ★ Can take complex features into account as model can be custom made to read intricate trends
- ★ Can handle multiple long-term and short - term Dependencies
- ★ Insensitive to outliers and noise is handled well
- ★ Capture periodic variations and impact factors

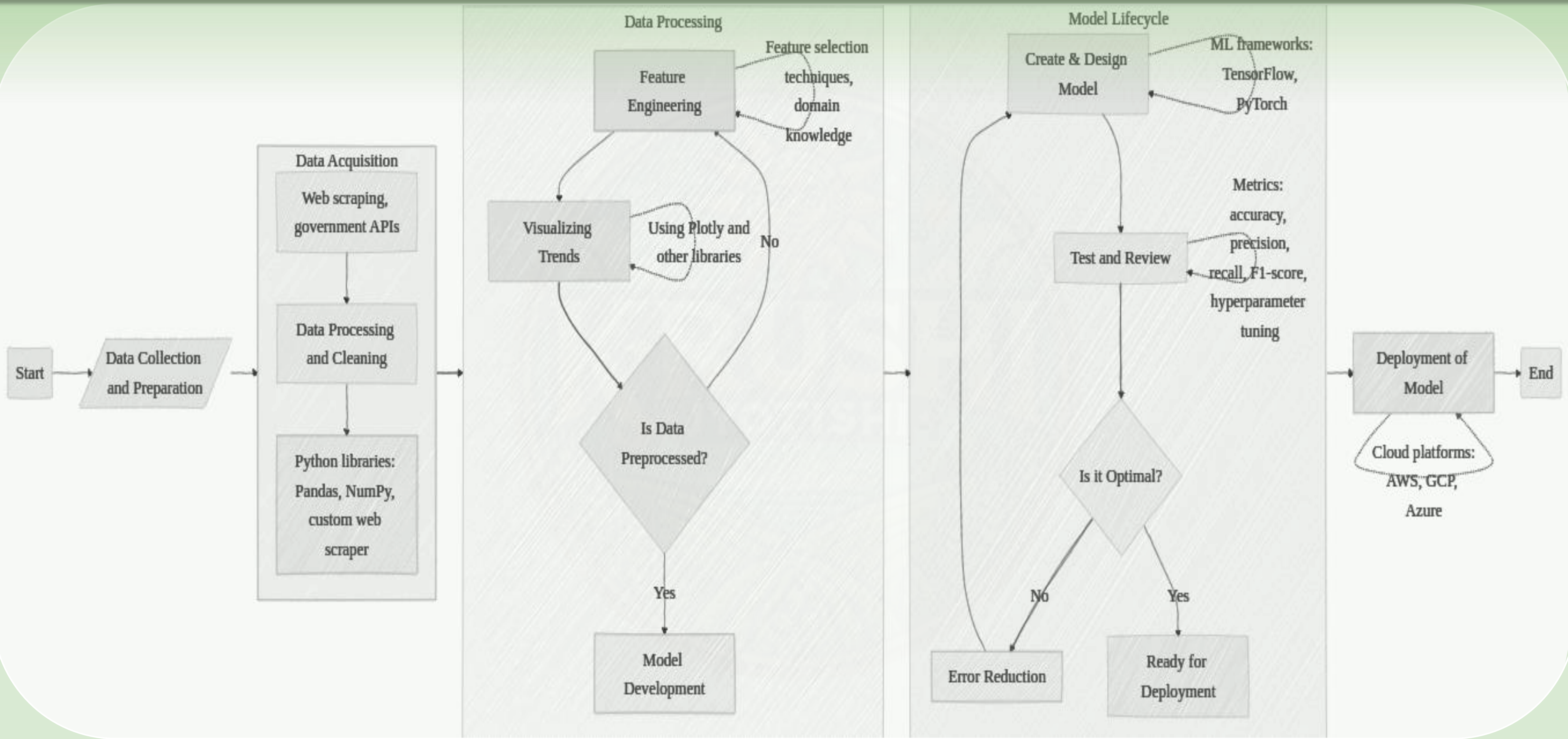
Direct Benefits of ML Model

- ★ **Increased Forecasting Accuracy**: Predictions for price fluctuations.
- ★ **Improved inference of price fluctuation :** The model can also give a good inference on direct and indirect effects of different parameters

Indirect Benefits of ML Model

- ➢ **Faster Market Interventions:** React to market changes and prevent crises.
- ➢ **Optimized Stock Levels:** Avoids shortages or excess of commodities

TECHNICAL APPROACH

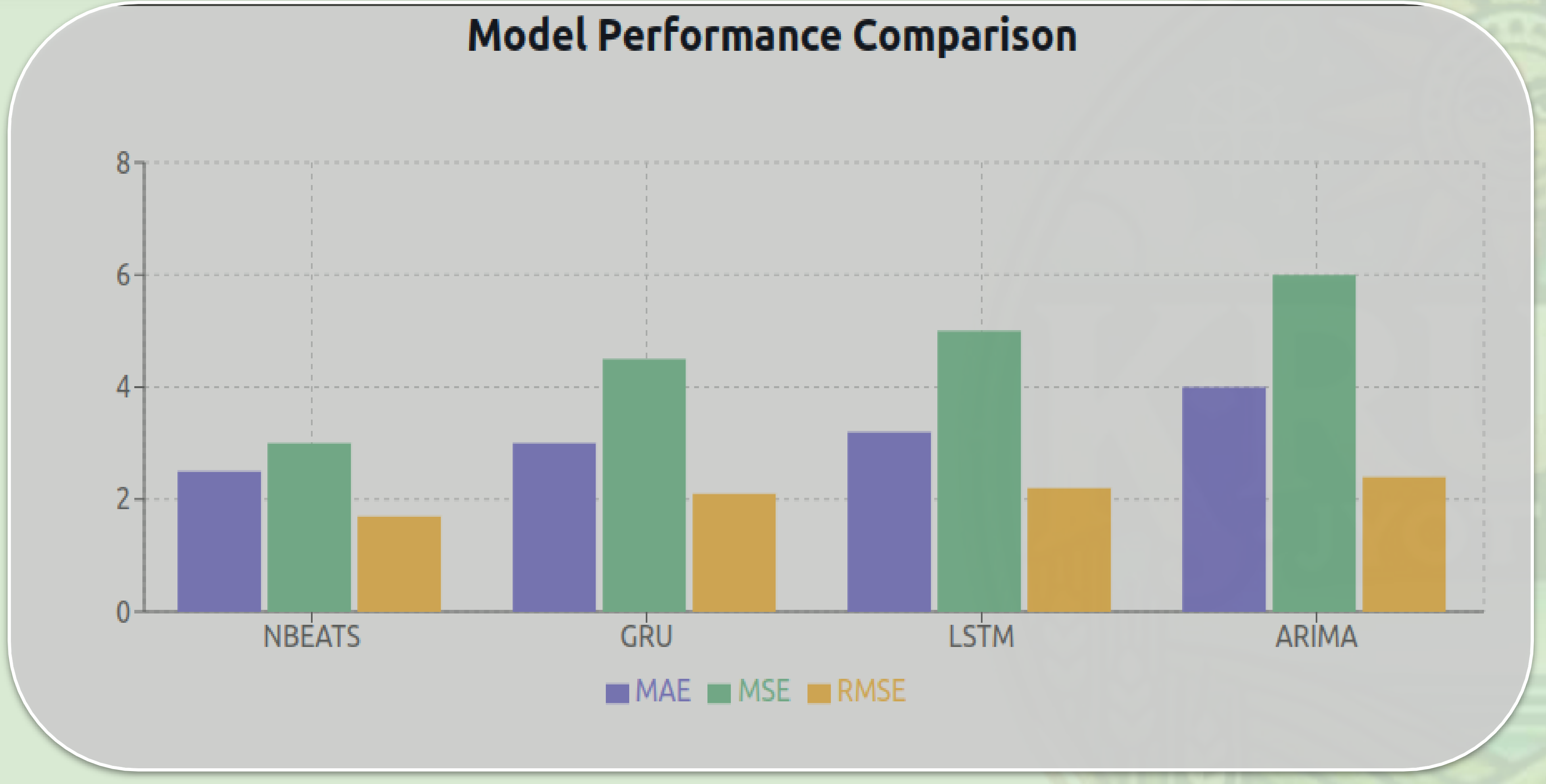


Ctrl K

3 FEASIBILITY & VIABILITY

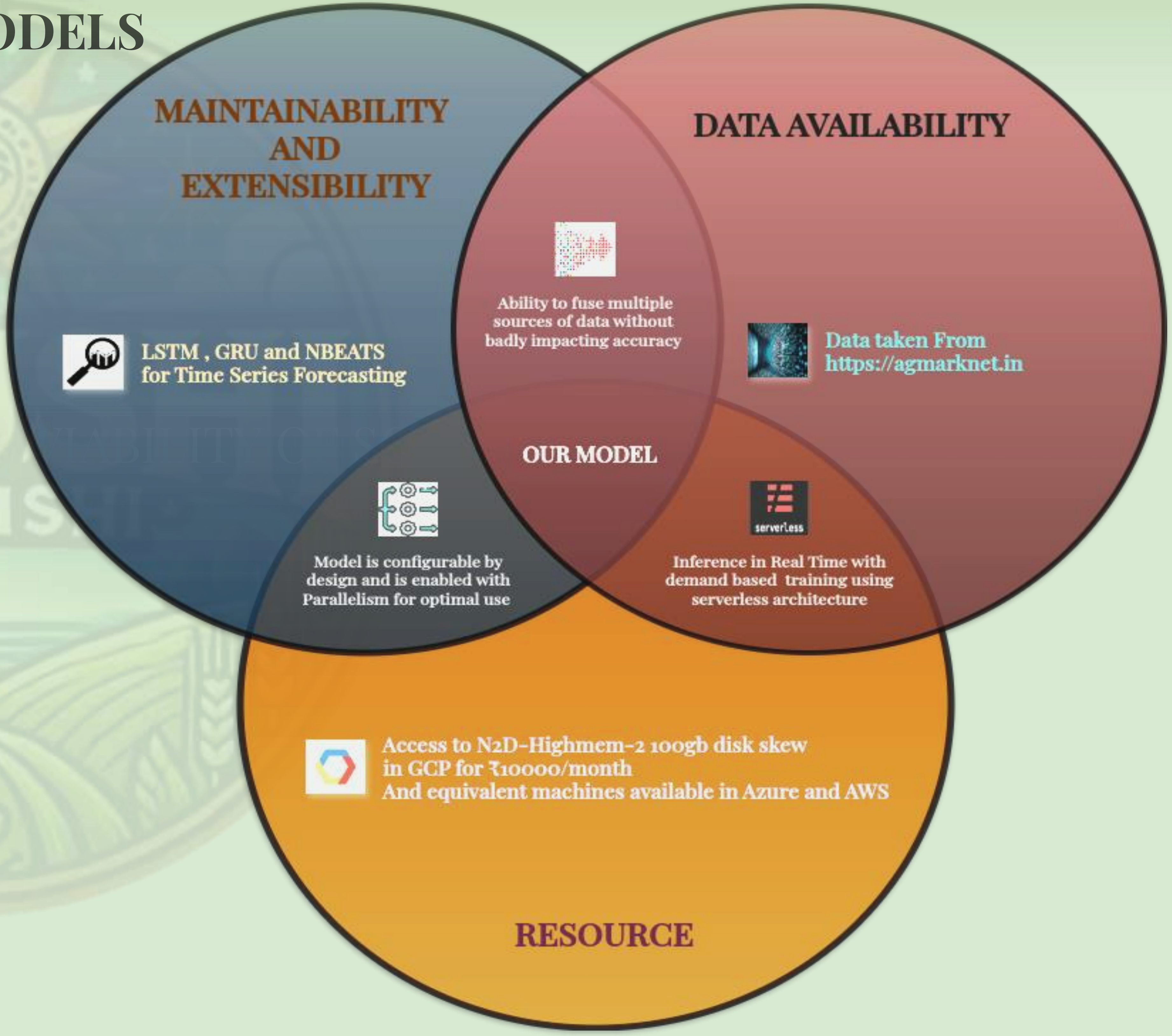
SMART INDIA HACKATHON 2024

GENERIC COMPARISON REPORT OF DIFFERENT MODELS



- ★ **Mean Absolute Error (MAE):** Measures the average absolute difference between predicted and actual values.
- ★ **Mean Squared Error (MSE):** Measures the average squared difference between predicted and actual values.
- ★ **Root Mean Squared Error (RMSE):** The square root of MSE, providing a more interpretable measure of error.

(Lower of these errors indicates better accuracy.)



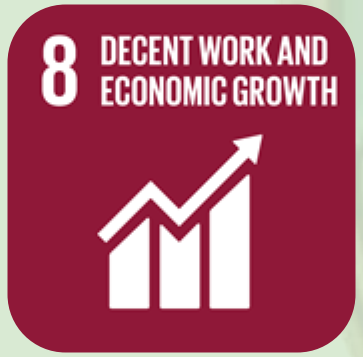
Ctrl K

4 IMPACT OF GOVERNMENT AGENCY

5 BENEFITS OF SOLUTION

SMART INDIA HACKATHON 2024

Joint Sustainable Outcomes

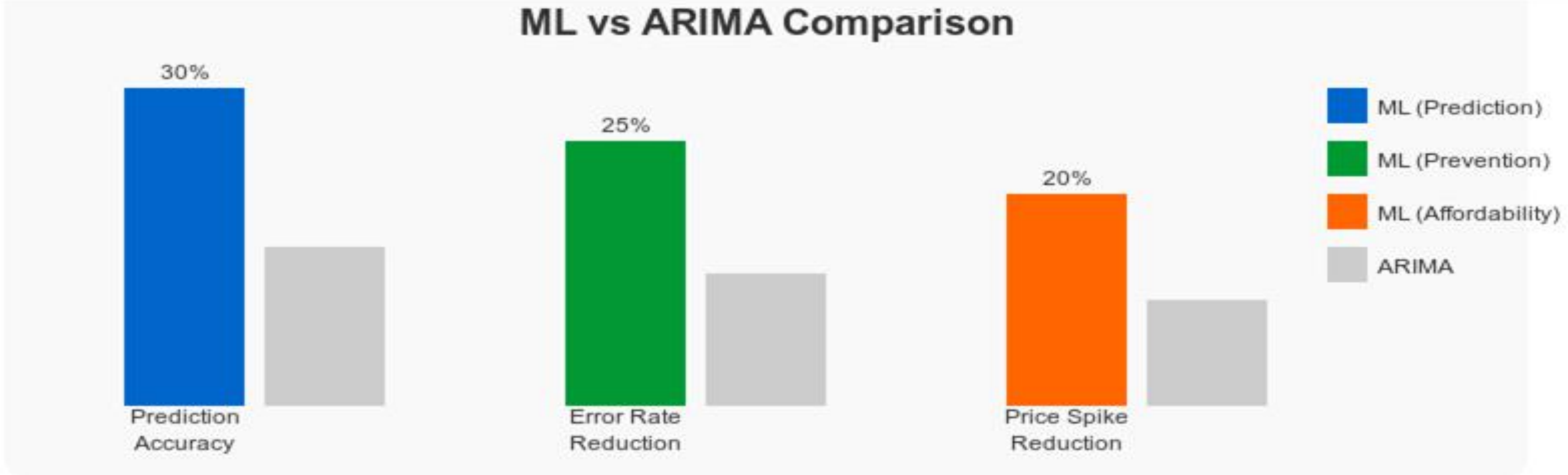


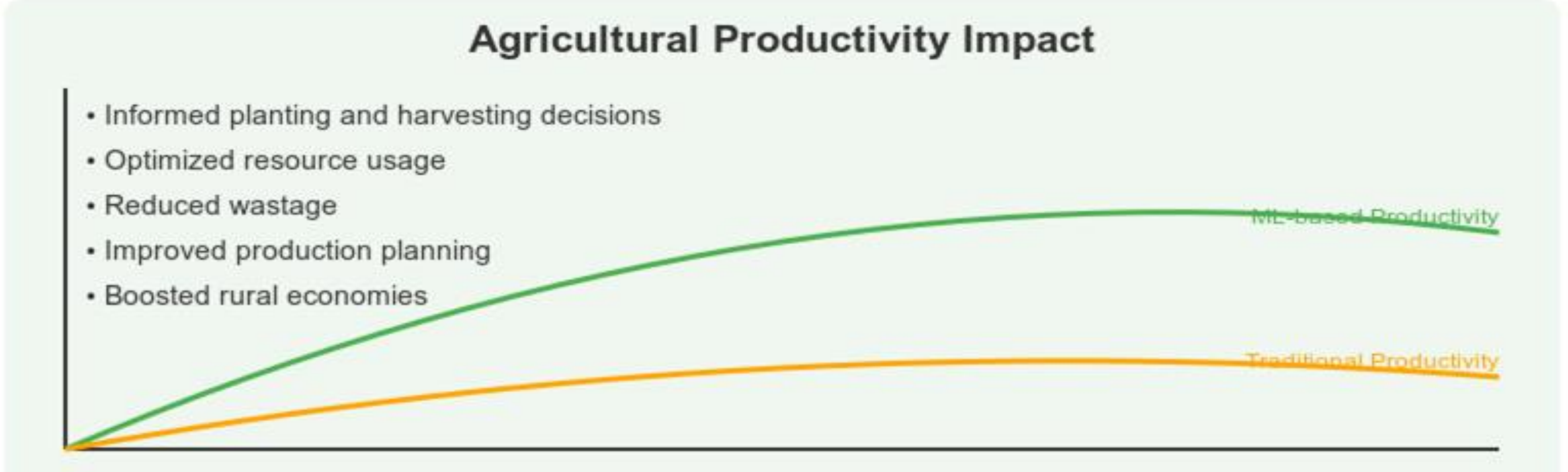
For the Government

★ Informed Policies
★ Informed decisions
★ Gain valuable insights
★ Address needs of farmers

For the Farmers

★ Risk Management
★ Income Stability
★ Predict potential challenges
★ Improving the yield





Ctrl K

6 K.J. V - 1 DESCRIPTION

SMART INDIA HACKATHON 2024

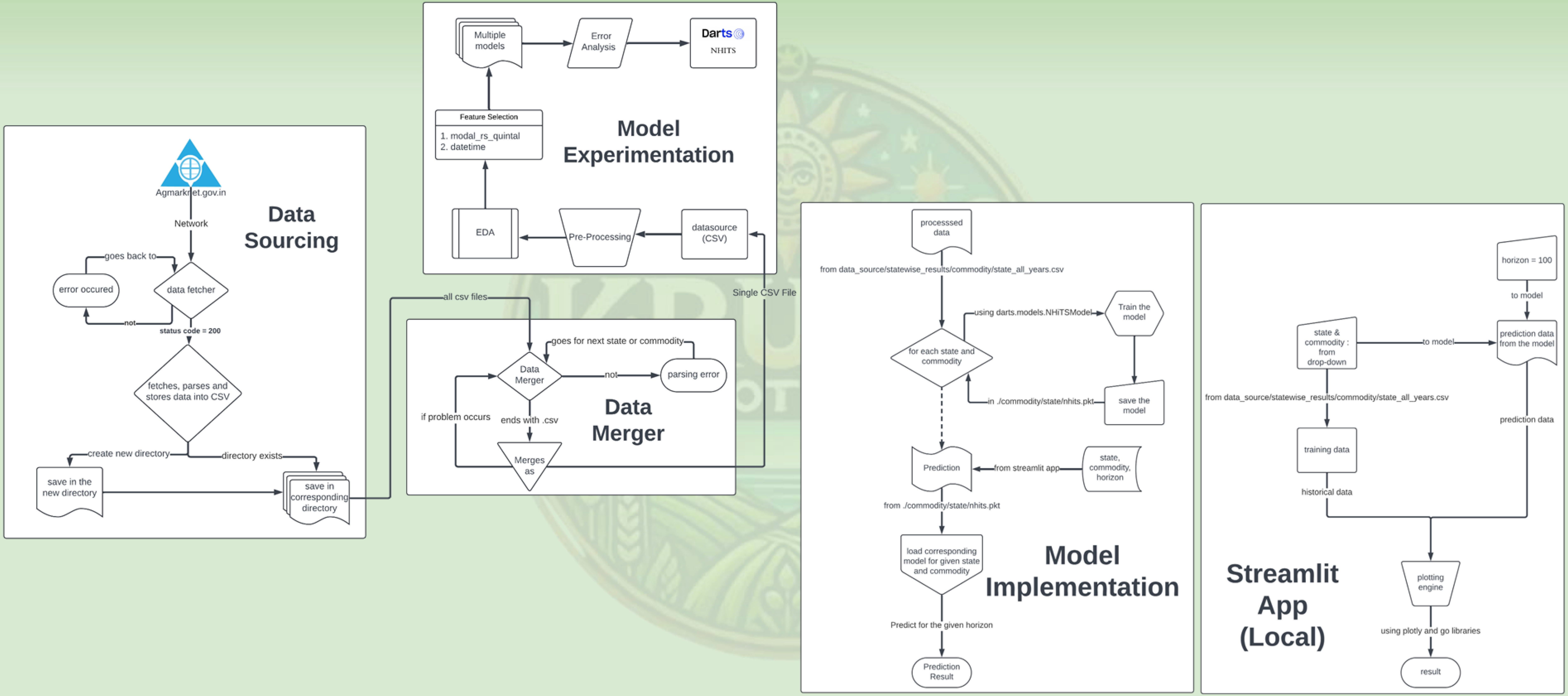
- ★ Generally time series data with absence of stationality are difficult to predict but Neural Hierarchical Interpolation for Time Series (NHITS), a better version of NBEATS (Neural Basis Expansion for Time Series), is 25% more accurate than Informer and runs 50 times faster by dividing data into pieces based on frequency.
- ★ It can predict the price of essential commodities with an accuracy of 99.4% , better than ARIMA which is around 75%.
- ★ It can use 5-year data of the state/commodity even if it is irregular to capture trends

	GRU Metrics	LSTM Metrics	N-BEATS Metrics
MSE	91397.416676	89559.558728	90802.782643
MAE	196.677272	203.061116	209.586583
RMSE	302.320057	299.265031	301.335001
F1	0.890323	0.896774	0.883871
R2	0.797508	0.801580	0.798826

Ctrl K

6.1 K.J. V - 1 PREVIEW

SMART INDIA HACKATHON 2024



Ctrl K

6.1 K.J. V - 1 DATA FETCHING & MERGING

SMART INDIA HACKATHON 2024

- ★ Data is fetched from agmarknet.gov.in (government website)
- ★ Data fetcher: webscraping engine using beautiful soup
- ★ If the network status code is 200, then it fetches (scraps) data from the website, parses it according to the specifications provided in the data fetcher and stores in a file with csv format
- ★ If the directory for the commodity and state exists, then it stores the CSV file in the directory.
- ★ If the directory is not found, creates a directory for the specific commodity or state and saves the file.
- ★ We store the data in a year-wise manner, as retrieving all data at once may result in data loss due to network traffic.

- ★ Data merger merges the CSV files together after fetching and storing the data in the directories.
- ★ Since it is best practice for any Deep Learning model to have all data in a single place rather than multiple places to train the model,
- ★ Once the CSV files are merged into one, the operation is successful. if not, it throws parsing error

Ctrl K

6.3 K.J. V - 1 MODEL EXPERIMENTATION

SMART INDIA HACKATHON 2024

- ★ We preprocess the data after receiving it and merging it into a single csv file.
- ★ We conducted an EDA to identify the features that influence the price of the commodities.
- ★ Then, select the features that influence the price of the commodities.
- ★ Then, we experimented with various models, such as NHITS, NBEATS, SARIMA, and ARIMA, among others, to determine which model could accurately identify the trend and seasonality of the prices.
- ★ Dart's NHITS performed well on the dataset, so we implemented it.

ACTUAL vs PREDICTED VALUES

NHITS	y
1135.288606	1128.114754
1132.345438	1117.956522
1128.98599	1124.852113
1126.017908	1105.476821
1126.135892	1106.873786
1111.310141	1100.593548
1109.129628	1104.006536
1105.274109	1133.906667
1101.431367	1111.716814
1096.856821	1128.019231
1090.696146	1122.314961
1090.937538	1112.274336
1080.566726	1130.253333
1078.60099	1127.961039
1080.597291	1139.980645
1070.447006	1138.196078
1060.712392	1123
1060.132075	1117.20625
1063.291848	1089.690909
1061.269809	1113.59375
1064.504254	1111.53125
1058.348634	1107.233766
1060.157422	1097.388535
1055.802965	1097.578947

NHITS	y
832.9281409	823.6507937
823.3529353	844.3571429
831.2108929	857.3642384
838.6940361	845.7409639
846.4337988	860.9642857
850.7569643	862.251497
858.2554757	863.5636364
863.3217211	883.378882
862.7111911	874.4705882
860.5232289	878.8461538
861.5055906	877.5507246
864.8232168	892.0081967
869.8753217	884.7530864
875.4296584	886.2982456
876.681939	887.9112426
876.0349167	898.5092025
875.8802909	902.0935673
877.5193017	901.8295455
879.5750162	903.3770492
880.9697603	911.035503
883.4254685	907.127907
884.8636015	910.3372781
885.2823844	917.1165644
882.8046736	918.0697674
879.0494175	917.6646707
875.9757786	920.8099174

NHITS	y
2623.887968	2581.857143
2601.893673	2681.118421
2611.833685	2662.755102
2623.976961	2594.227273
2608.818573	2559.982456
2592.751539	2648.548673
2585.202495	2645.140187
2604.085138	2586.7
2591.944252	2604.42029
2602.003608	2640.438596
2606.298145	2655.246753
2596.780818	2679.827586
2584.61038	2655.892857
2583.608959	2655
2608.799085	2608.761905
2584.136427	2662.127273
2599.885318	2650.5
2597.248037	2681.719298
2589.681436	2579.181818
2571.841172	2643.858491
2579.090359	2599.295238
2600.703656	2607.556604
2581.907457	2640.721154
2590.708756	2641.271186
2586.104766	2661.168224
2591.356804	2577.735849

NHITS	y
2156.997648	2136.945455
2159.390575	2137.309278
2165.609106	2136.661417
2170.408441	2137.986577
2169.14252	2135.271605
2169.800115	2141.432432
2169.179214	2142.041379
2170.394672	2141.709924
2172.257257	2141.278351
2173.82279	2145.255172
2173.149536	2145.313725
2176.889371	2143.047059
2180.137231	2149.459459
2173.77642	2152.741497
2175.883579	2155.690141
2173.568452	2158.642857
2181.392263	2167.126623
2179.931771	2168.822368
2174.298694	2166.398876
2174.012263	2175.510067
2173.798674	2178.449324
2178.900808	2186.059028
2179.261812	2192.055944
2177.639135	2201.904459
2182.060142	2202.852349
2183.752766	2199.556962

6.3 K.J. V - 1 MODEL IMPLEMENTATION & STREAMLIT DEPLOYMENT

SMART INDIA HACKATHON 2024

- ★ We load the required features like modal_rs_quintal and datetime from the processed dataset and train the NHITS model for each state-commodity combination.
- ★ We save each trained model locally in a pickle-torch format .
- ★ During prediction, for each state and commodity we load the corresponding saved pickle file to predict the price for given horizon.
- ★ As we are caching the trained model in the disk, we are optimally using the resources available to us.

- ★ User can choose the state and commodity from the dropdown.
- ★ After the input selected by the user, the streamlit app undergoes the life cycle as denoted in the flowchart beside.
- ★ We chose plotly and go libraries as they are the easiest means to have interactive charts on websites and also they are light weight.
- ★ Streamlit is one of the best platforms that allow hosting websites requiring data loading and data visualization.
- ★ This version of the app cannot be deployed because it involves data from 2 sources namely the model and the data source.

7 K.J. V - 2 DESCRIPTION

- ★ Update the data fetching and processing pipeline periodically (weekly)
- ★ Model pipeline runs by default after data fetching pipeline but can be run at will
- ★ Entire solution is hosted in the cloud and is automated
- ★ The client can take ownership with just a transfer of the billing account and voila, it will be yours

AWS EC2

AWS EBS

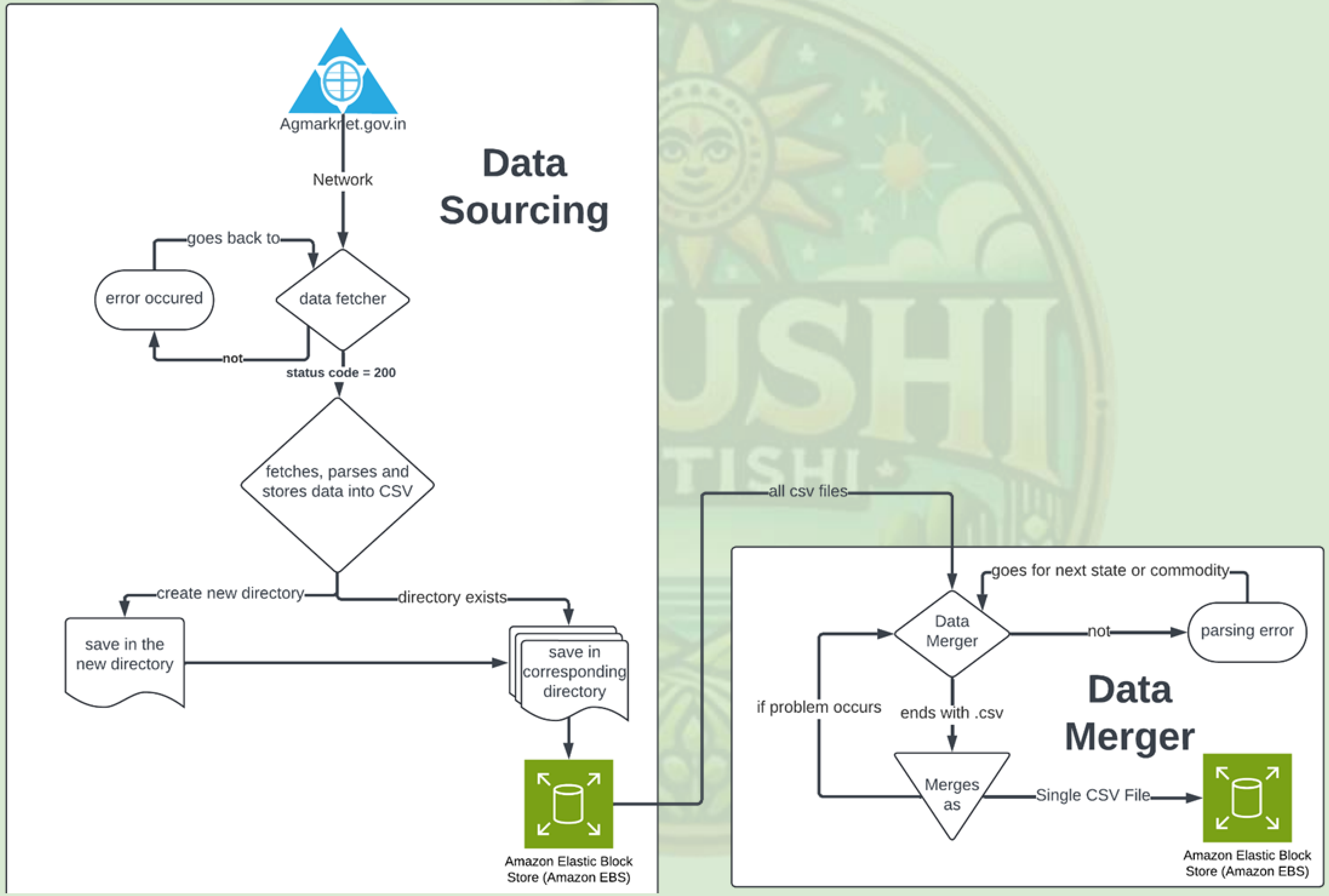
7.1 K.J. V - 2 DATA FETCHING & MERGING

Data Sourcing:

- ★ **Network Request:** The process begins by fetching data from Agmarknet.gov.in via a network request.
- ★ **Error Handling:** If a network error occurs (status code != 200), the process the says 'data not found' and goes to next commodity or state.
- ★ **Data Fetching and Parsing:** Upon a successful network request, a data fetcher retrieves and parses the data. This step transforms the raw data into a CSV (Comma Separated Value) file format.
- ★ **CSV File Saving:** The parsed CSV file is saved to the appropriate directory (either the existing or newly created one). The data is saved into an Amazon Elastic Block Store (EBS) for persistence.

Data Merger:

- ★ **Input:** All generated CSV files are collected as input to the data merger.
- ★ **Merging Process:** The Data Merger combines all the individual CSV files into a single, unified CSV file.
- ★ **Error Handling:** If any problems occur during the merging process (e.g., file format inconsistencies), an error is reported then it goes for next commodity or state.
- ★ **Output:** The merged, single CSV file is saved to Amazon EBS.



Ctrl K

7.2 K.J. V - 2 MODEL IMPLEMENTATION

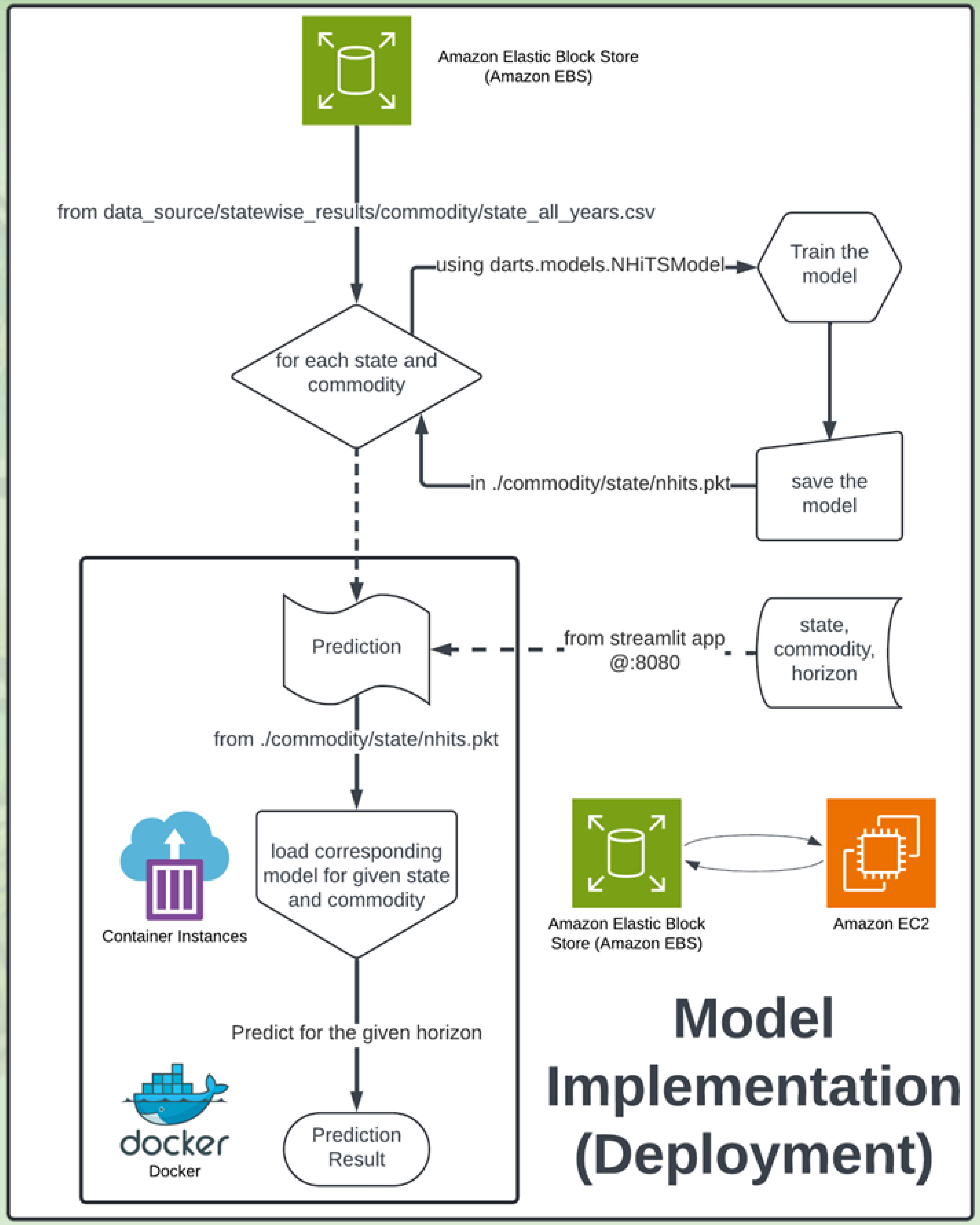
SMART INDIA HACKATHON 2024

Model Training and Saving:

- ★ **Data Input:** The process starts by loading data from data_source/statewise_results/commodity/state_all_years.csv. This CSV file contains the preprocessed features needed for model training.
- ★ **Iterative Training:** The system iterates through each state and commodity combination.
- ★ **Model Training:** For each state-commodity pair, the darts.models.NHiTSModel is trained using the corresponding data from the input CSV.
- ★ **Model Saving:** After training, the trained model is saved locally in a .pkt file (presumably a pickled PyTorch file) within a directory structure reflecting the state and commodity (./commodity/state/nhits.pkt). The trained models are stored in Amazon Elastic Block Store (EBS) for persistence.

Model Deployment and Prediction:

- ★ **Streamlit App Request:** A Streamlit application receives requests (state, commodity, and prediction horizon) via port 8080.
- ★ **Model Loading:** Based on the received state and commodity information, the application loads the appropriate pre-trained model from the corresponding .pkt file stored in Amazon EBS.
- ★ **Prediction:** The loaded model predicts the price for the specified horizon using the input data.
- ★ **Prediction Result:** The prediction result is sent back to the Streamlit application.
- ★ **Containerization:** The prediction process runs within Docker containers, allowing for consistent and reproducible execution. The application itself is likely hosted on an Amazon EC2 instance.



7.3 K.J. V - 2 STREAMLIT APP DEPLOYMENT

Ctrl K

SMART INDIA HACKATHON 2024

Data Acquisition:

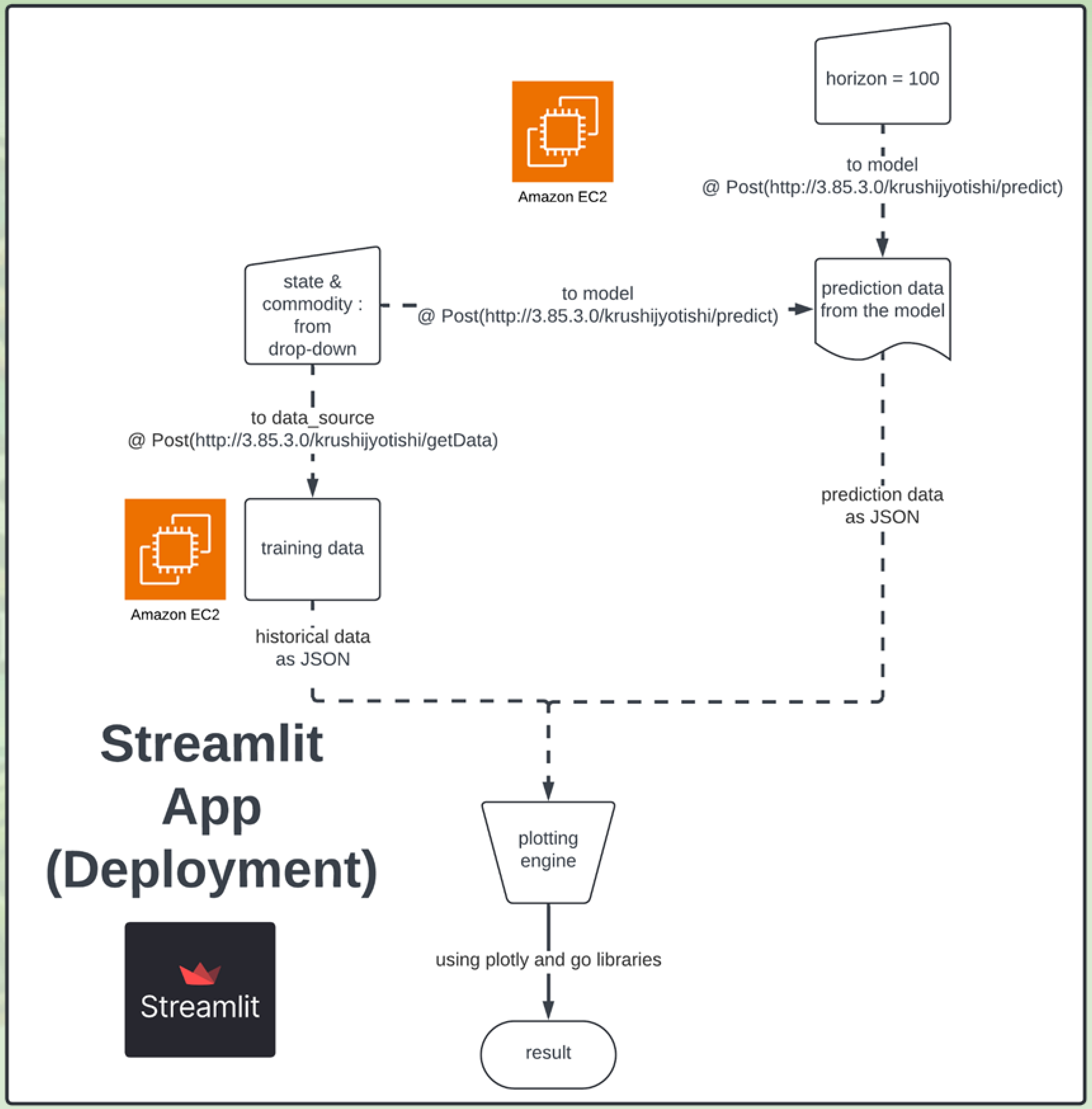
- ★ **User Input:** The user selects a state and commodity from dropdown menus in the Streamlit interface.
- ★ **Data Source Request:** This selection triggers a POST request to /krushijyotishi/getData (at http://3.85.3.0/krushijyotishi/getData). This endpoint retrieves historical data (as JSON) for the selected state and commodity from a data source (likely hosted on an Amazon EC2 instance).

Prediction:

- ★ **Prediction Request:** A POST request is sent to /krushijyotishi/predict (at http://3.85.3.0/krushijyotishi/predict), specifying the selected state, commodity, and a prediction horizon of 100. This request is directed to a model endpoint (also likely hosted on an Amazon EC2 instance).
- ★ **Model Prediction:** The model processes the request, making a prediction for the future price based on the data.
- ★ **Prediction Response:** The model returns the prediction results as JSON data.

Visualization and Output:

- ★ **Plotting Engine:** The application receives the prediction data (as JSON) and uses a plotting engine (leveraging Plotly and Go libraries) to generate a visualization of the results.
- ★ **Display:** The generated chart, showing the predicted prices, is displayed in the Streamlit application.



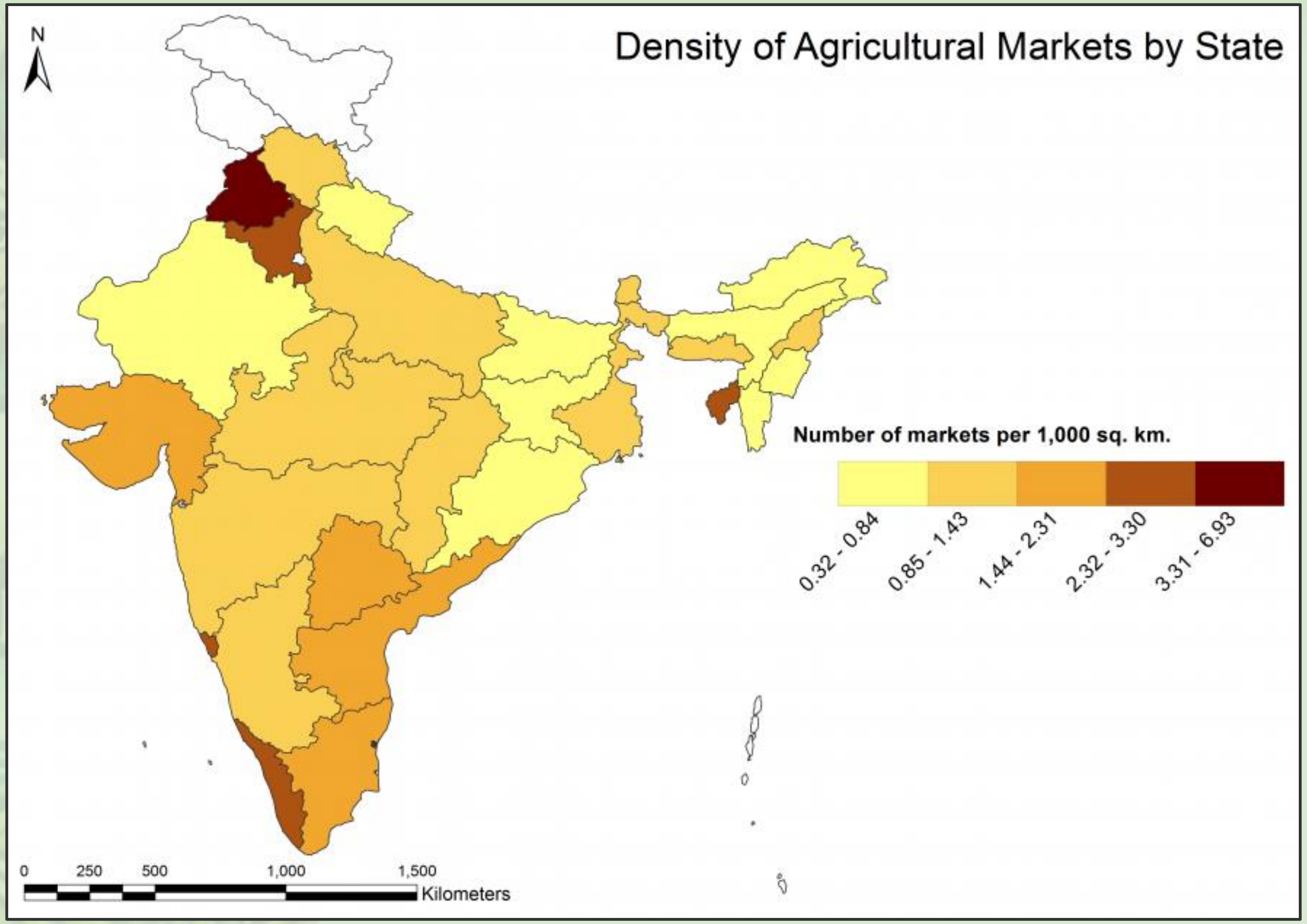
8 K.J. V - 3 DESCRIPTION

SMART INDIA HACKATHON 2024

★ The price for predictions analysed in different ways using
  - dynamic line plots(adjustable prediction range) and choropleth maps with todays prediction

★ This showcases the optimal dynamic data loading algorithms that are able to load large quantity of data with limited resources

★ Example of choropleth map of india given beside



source: https://tci.cornell.edu/?blog=mapping-mandis-a-spatial-exploration-of-agricultural-markets-in-india

Ctrl K

K.J.V - 3 CRON JOB

SMART INDIA HACKATHON 2024

What is Cron Job?

A cron job is a scheduled task that runs automatically at specified intervals or times on Unix-like operating systems. It is managed by the cron daemon (crond), which is a background service that executes commands or scripts as per the instructions defined in a crontab (short for "cron table") file.

Cron Jobs present for:

- ★ data fetcher
- ★ data merger
- ★ model implementation

This is image shows a crontab file, used to schedule tasks on a Linux system. The last line schedules a script to run every Friday (5) at 6:30 AM (30 6). The comments explain the format of each line: minute, hour, day of month, month, day of week, then the command to execute. * means "every".

```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
30 6 * * 5 /home/ubuntu/cron_job_files/fetch_merge_model.sh
```

K.J.V - 3 AUTO-UPDATE PIPELINE

SMART INDIA HACKATHON 2024

Part 1: Data Fetching and Merging

- ★ **Activate Python Environment:** The following script initially activates the corresponding python virtual environment, ensuring the correct Python version and libraries are used.
- ★ **Run Data Pipeline:** python data_fetcher.py fetches the data. python data_merger.py then merges the fetched data and saves into the Elastic Block Storage connected with EC2 instance.

Part 2: Model Training

- ★ **Run Model Pipeline:** the script with deploy_model_state_commodities.py runs the main model training script . This trains the model using the previously processed data and saves it in pickle-pytorch format in corresponding directory in the EBS in EC2 instance

```
# sourcing python environment with preinstalled libraries
source /home/ubuntu/.venv/bin/activate
# changing directory to run  Fetching & Merging pipeline
cd /home/ubuntu/sih_2024_project/sih_2024_data_source/
# Run fetch followed by merge
python data_fetcher.py
python data_merger.py
# Change directory to train models pipeline on new data
cd /home/ubuntu/sih_2024_project/sih_2024/
# Run the model pipeline
python deploy_model_state_commodities.py
```

FUTURE SCOPES

- Deploy more commodities into the application like seasonal commodities, exported and imported food commodities.
- Choropleth mapping can be incorporated.
- Include commercial crops such as cotton, jute etc as well.
- AgriTech and Fintech Innovations
- Climate adaptation and soil features also to be incorporated.