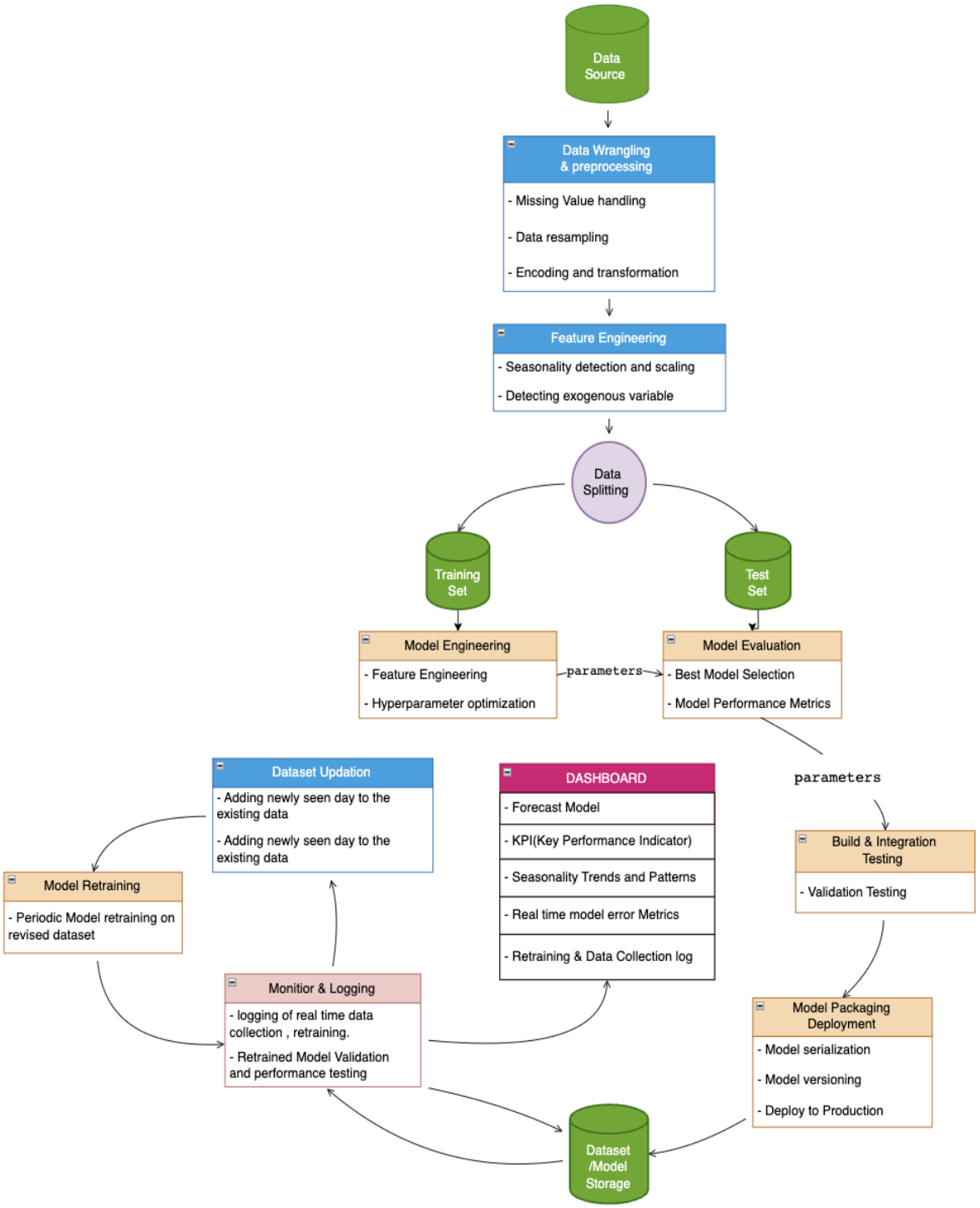
Academic Grand Challenge on

Climate Challenge

Solution Architecture



Solution Architecture -

1. Model Life Cycle:

a. Data Wrangling and Preprocessing: Missing values are imputed, variables are expressed as numerical values, and data is resampled, transformed, and scaled.

b. Feature Engineering: Seasonality is identified and encoded, and data is scaled.

c. Split the Data: The data is split into training, validation, and test sets.

d. Hyperparameter Tuning: Cross-validation is used for hyperparameter tuning using fbprophet.

e. Model Training: Baseline models, linear models, deep neural networks, LSTMs, CNNs, autoregressive LSTMs and Prophet models are trained.

f. Forecast and Evaluate: The models are used to forecast over the test set, evaluated using various error metrics, and compared to the baseline performance; the best-performing model is selected.

Libraries Used - Numpy ,Pandas ,MatPlotLib ,Seaborn ,TensorFlow ,Keras ,SkLearn ,fbprophet.

A typical forecasting model uses past-predicted data to make future predictions, but this method has a drawback. With each subsequent prediction, the accuracy decreases due to the accumulation of marginal error, leading to significant inaccuracies in the long term.
To mitigate this issue, we propose periodically inserting newly collected weather data into the original dataset and re-training the best model. This will restore the accuracy of the predictions. To automate this process, we plan to integrate it into the existing classic architecture using the CI/CD process or MLOps.

2. Continuous Integration (CI): On each commit to the code repository, the CI system automatically builds the model and runs the unit tests. This ensures that the code is always working and catches any bugs early in the development cycle.

3. Continuous Deployment (CD): After the code is successfully built and tested, the CD system automatically deploys the code to a testing environment.

4. Model Packaging, Build, & Integration Testing: The machine learning model is packaged into a reusable form and integrated into the overall system. Automated tests are run to ensure that the model is working correctly and that it integrates with the rest of the system seamlessly.

5. Model Versioning: The model is versioned to preserve previous versions for easy retrieval in cases where the retrained model performs less effectively.

6. Deployment to Production: After the model is tested and evaluated in a testing environment, it is deployed to a production environment where it is used by end-users through the dashboard.

7. Monitoring & Logging: The system is monitored for performance, accuracy, and reliability. Logs are captured to help diagnose any issues that may arise.

8. Model Evaluation: The model's performance is regularly evaluated using various error metrics to ensure that it is still accurate and relevant.

9. Dashboards: A Dashboard integrates various metrics, data sources, APIs, and services to provide companies and organisations with a centralised platform to access and analyse critical information. Thus, our dashboard presents every process from deployment, and Performance monitoring to Model Retraining, along with the forecast model.