IDEA PROPOSAL

TEAM 16 - DATADA

INTRODUCTION

In the fast-paced realm of retail, businesses grapple with the ever-present challenge of optimizing sales and pricing strategies. The intricacies of forecasting sales accurately and implementing dynamic pricing models often lead to inefficiencies that hinder growth and profitability.

To address these pain points, our Minimum Viable Product (MVP) provides an interface to forecast and optimize pricing through a user-friendly, simple website application that helps retailers improve their sales the easiest way possible.

PROBLEM STATEMENT

Precise sales forecasting and optimal pricing strategies are critical for long-term business success in the ever-changing retail environment.

Conventional methods frequently fail to deliver real-time insights, resulting in lost chances and less-than-ideal results.

Pricing decisions are frequently made without leveraging advanced analytical techniques, leading to missed revenue opportunities and potential customer dissatisfaction.

PAINT POINTS AND INEFFICIENCES

1. Sale Forecasting

- **Inaccuracies**: Conventional methods struggle to predict sales with precision, resulting in overstock or stockouts.
- **Reactivity**: Businesses face challenges in adapting quickly to changing market trends, leading to missed opportunities.

2. Pricing Strategies

- **Static Approaches**: Fixed pricing models limit the ability to respond dynamically to market fluctuations and competitor actions.
- **Revenue Impact**: Inefficient pricing structures may lead to potential revenue loss or customer dissatisfaction.

SOLUTION OVERVIEW

It is a **web-based platform** that integrates advanced **machine learning model** and **data analytics** to provide real-time insights and recommendations for pricing strategies and sales forecasting.

AI’s IMPACT ON SOLUTIONS

1. Enhanced Predictive Analytics:

- **Sales Forecasting**: AI algorithms, such as hybrid models combining traditional time series analysis and machine learning techniques, significantly improve the accuracy of sales predictions. This allows businesses to proactively manage inventory levels based on precise demand forecasts.
- **Pricing Optimization**: Long Short-Term Memory (LSTM) networks, a type of recurrent neural network (RNN), enable the model to capture complex patterns and dependencies in pricing data. This leads to more accurate price predictions, considering factors such as seasonality, market trends.

2. Continuous Learning and Improvement:

- Machine learning models incorporated in the MVP can continuously learn from new data, enabling them to adapt to changing market dynamics. This ensures that the solutions remain relevant and effective over time.

OVERVIEW OF METHODOLOGIES

1. Architecture – Predict Sale

- Hybrid model (Random Forest – XGBoost – Linear Regression) approach

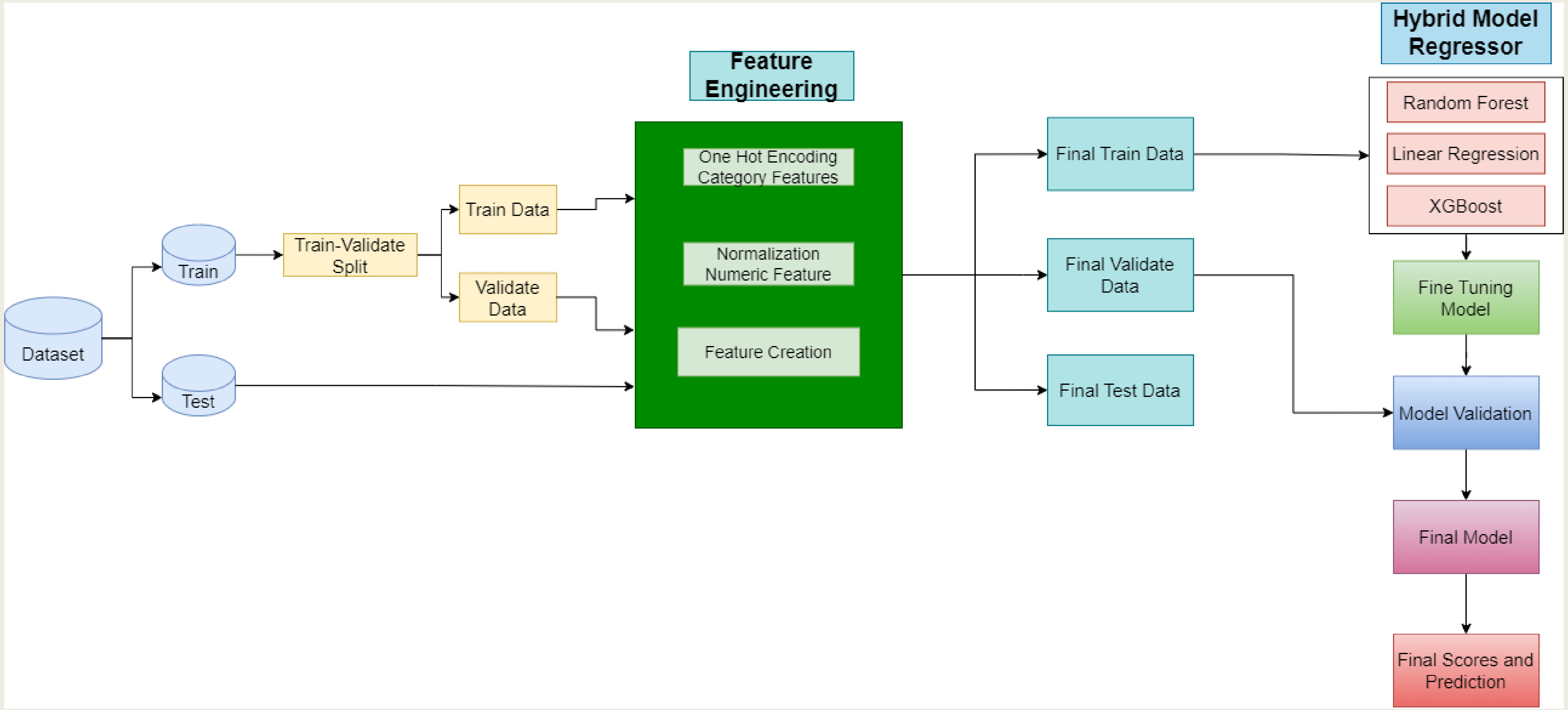
2. Architecture – Optimizing / Strategy Price

- An advanced Long Short-Term Memory (LSTM) approach

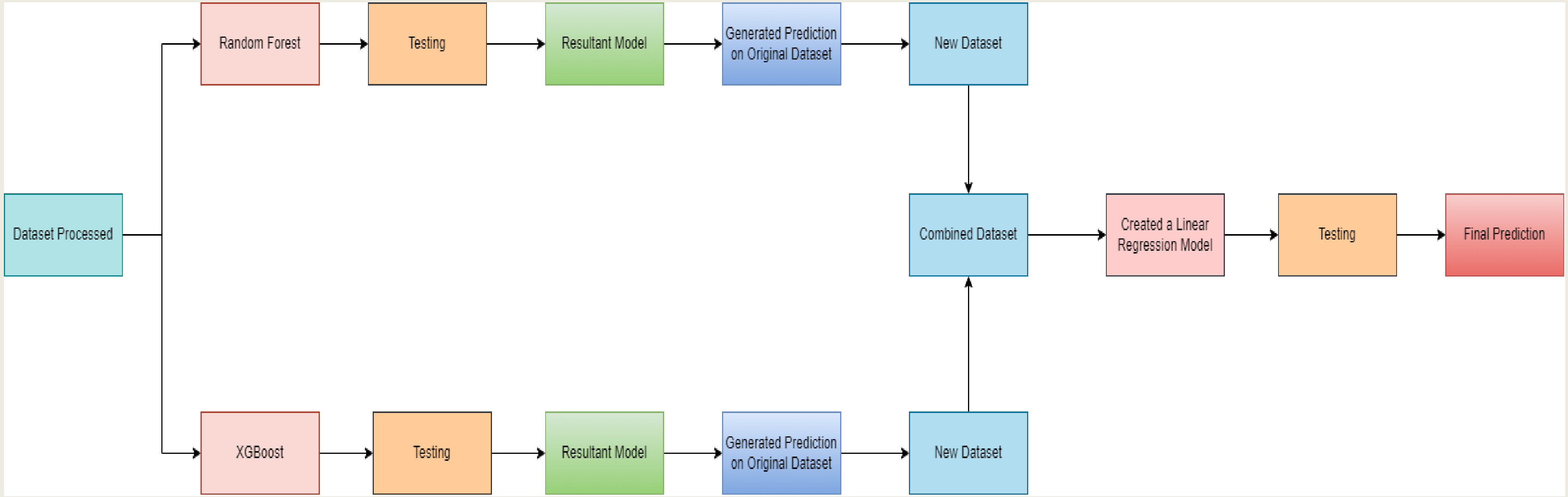
3. Machine Learning Model Deployment Architecture

- Streamlit
- MLFlow

PREDICTION SALE ARCHITECTURE

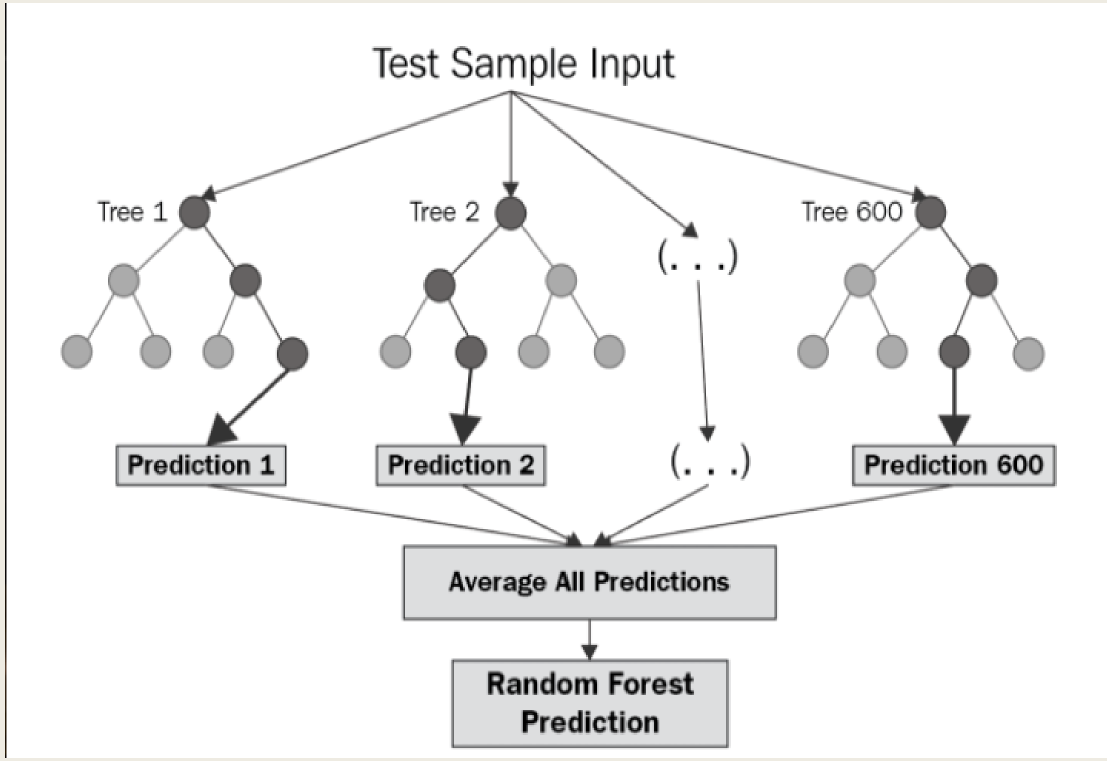


HOW HYBRID MODEL WORK



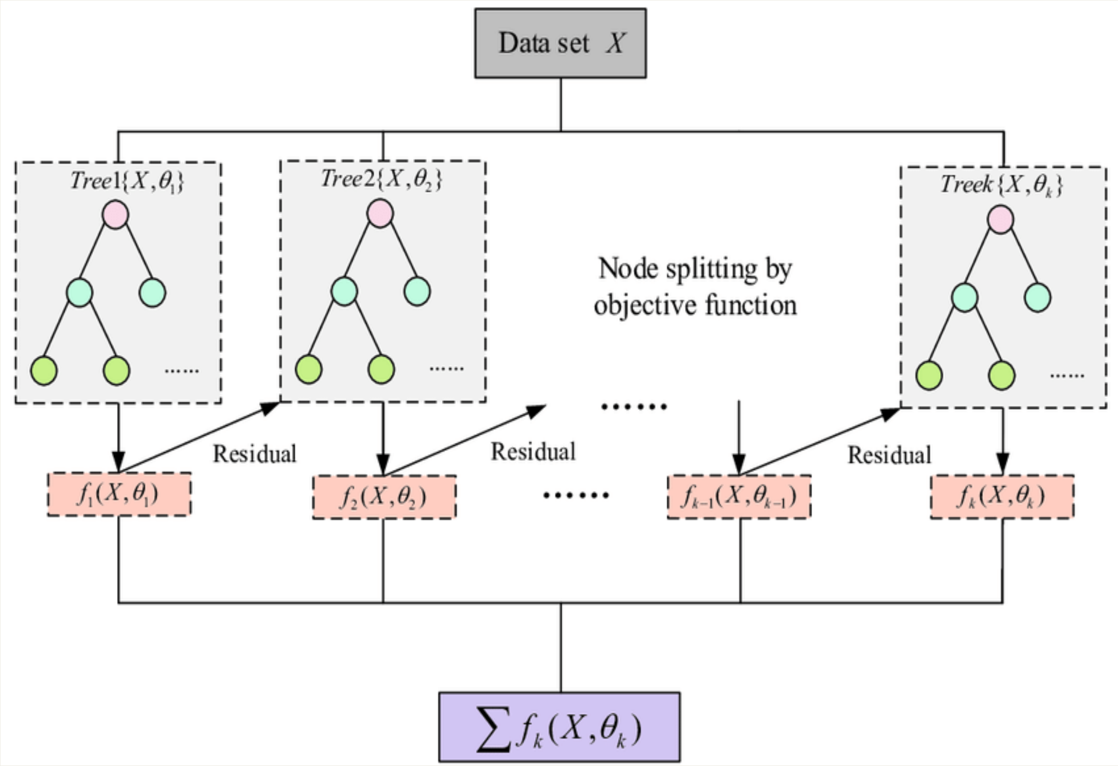
MODEL DESCRIPTION – RANDOM FOREST

- RF is an ensemble technique in which the results of many regression trees are combined to generate a single prediction
- The primary premise is bagging, in which a sample of training data is selected at random and fitted into a regression tree



MODEL DESCRIPTION – XGBOOST

- An abbreviation for ‘extreme gradient boosting’ is XGBoost with potential improvements upon gradient boosting
- XGBoost enhances the performance and is capable of solving problems of real-world scale while making use of a minimum number of resources
- XGBoost is a parallel tree model built upon the gradient boosting model. It utilizes the tree ensemble method, which is made up of a series of CART



WEAKNESS OF RANDOM FOREST AND XGBOOST

- **Random Forest**, while effective at reducing overfitting with parallel decision trees, has limitations
- It uses separate trees for different training data copies, boosting accuracy by reducing variance.

➔ However, its independent tree learning leads to poor adaptive learning, limiting information exchange between trees and reducing overall model performance.

- **XGBoost** is a boosting technique.
- It takes advantage of parallel processing and runs the model on several CPU cores

➔The problem arises when early iterations lack enough trees, significantly affecting the model. Overfitting leads to poor generalization and unreliable performance on new data, causing high variance and low bias

REASON TO DEVELOP A HYBRID MODEL

- The hybrid model combines different individual models to benefit from their strengths and address their weaknesses
- In this MVP, a hybrid machine learning model merges the bagging technique of a random forest regressor with the boosting technique of XGBoost regressor

So, **two main reason** to develop:

➢ To eliminate the risk of an unfortunate prediction of a single forecast in some specific conditions.

➢ To improve upon the performance of the independent models.

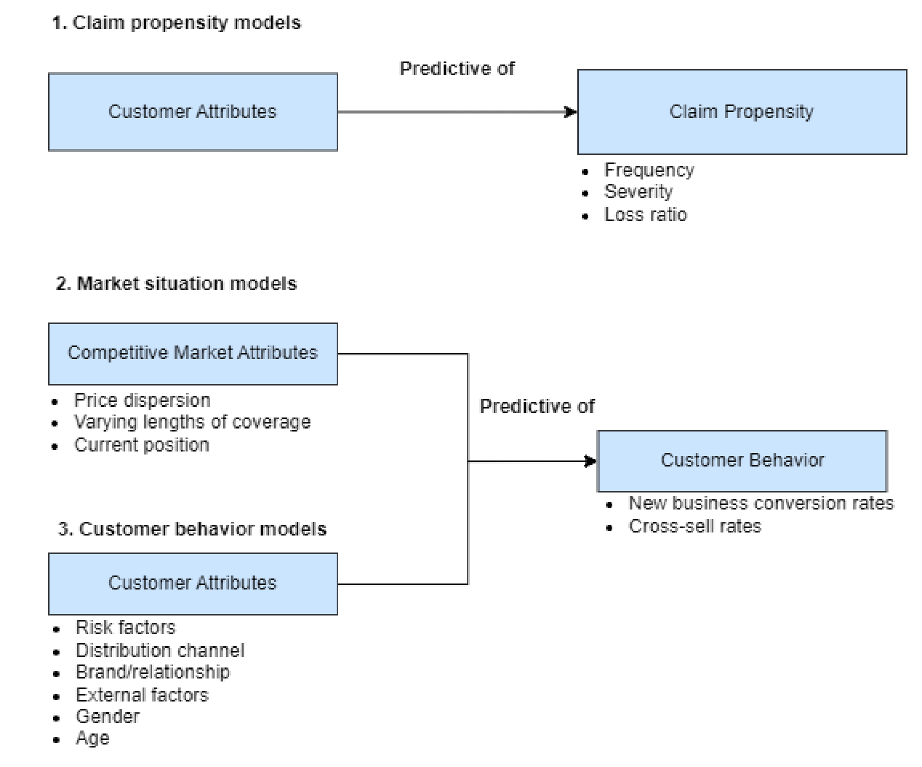
OUTPUT MODEL - LINEAR REGRESSION

- In the proposed framework, Random Forest and XGBoost models are trained separately and predictions of both the models are used as input into an Linear Regression model
- The Linear Regression model processes the final output

The reason why choosing **linear regression as output**:

- The Random Forest and XGBoost model outputs feed into an Linear Regression model for the final predictions due to its simplicity
- Using a complex model in the final layer might cause overfitting in the hybrid model
- This hybrid model addresses the limitations of both Random Forest and XGBoost

OVERVIEW OF PRICE OPTIMIZATION

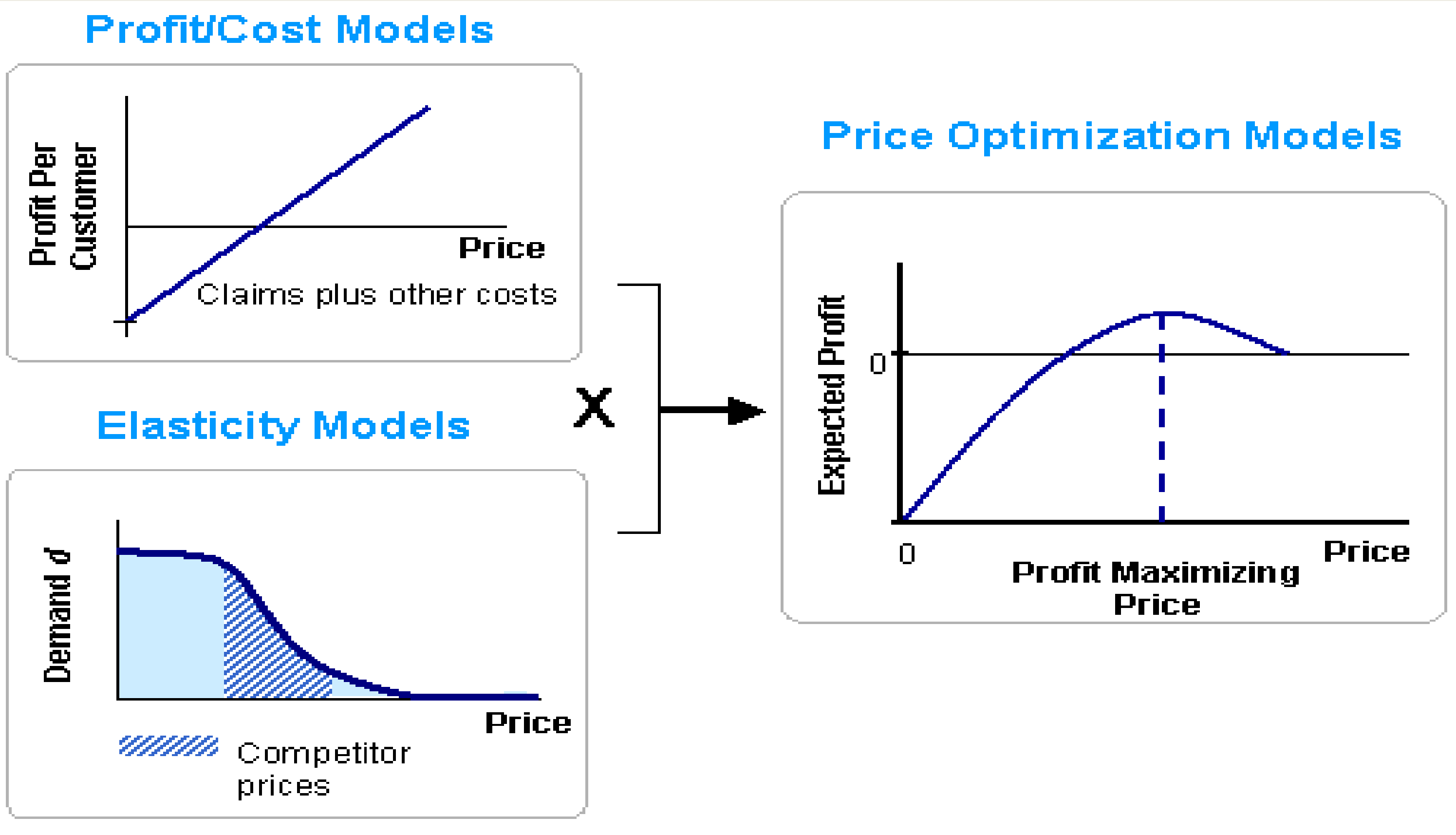


- Price optimization integrates claim propensity, market situation and customer behavior models to predict the impact of price changes on volume and to identify the best price changes for a given financial objective and constraints.

- The goal is to provide a company with the tools necessary to reach a particular strategic objective (e.g., volume or profitability) and to allow for flexibility to adapt to changing business circumstances.

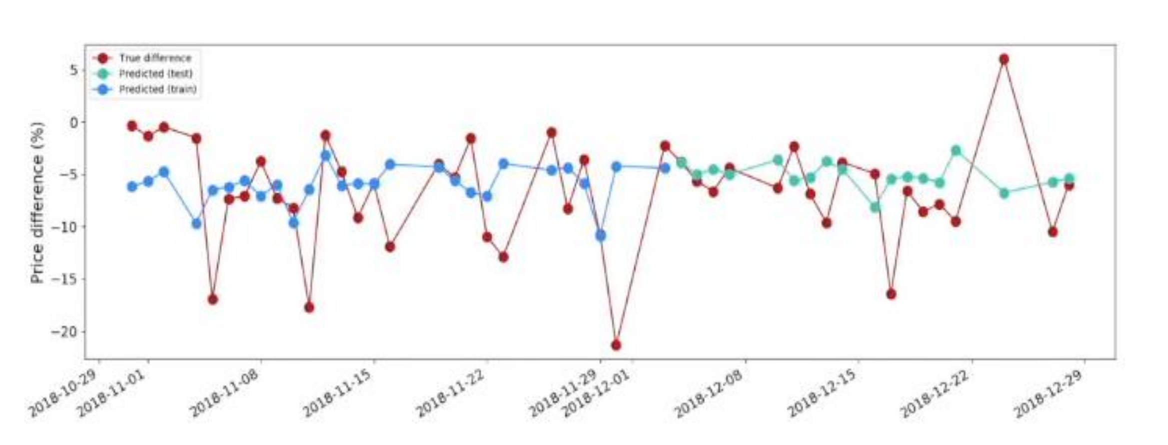
- Note: In this part, crawl or gather information on competitive prices, particularly on the internet through web scraping or using APIs provided by e-commerce platforms or price comparison websites

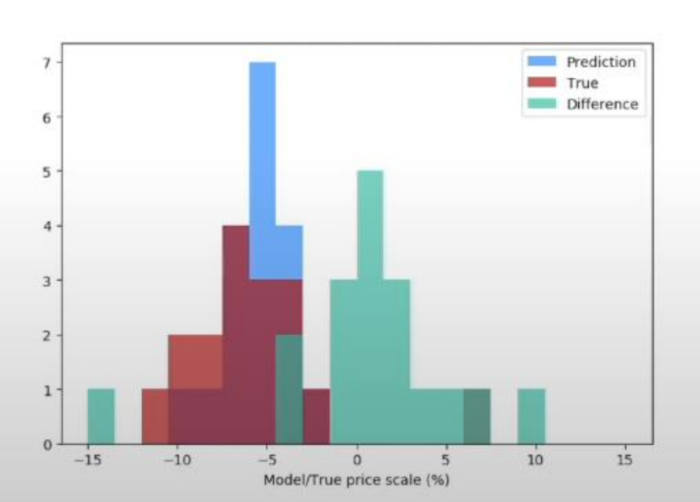
OVERVIEW OF PRICE OPTIMIZATION (cont)



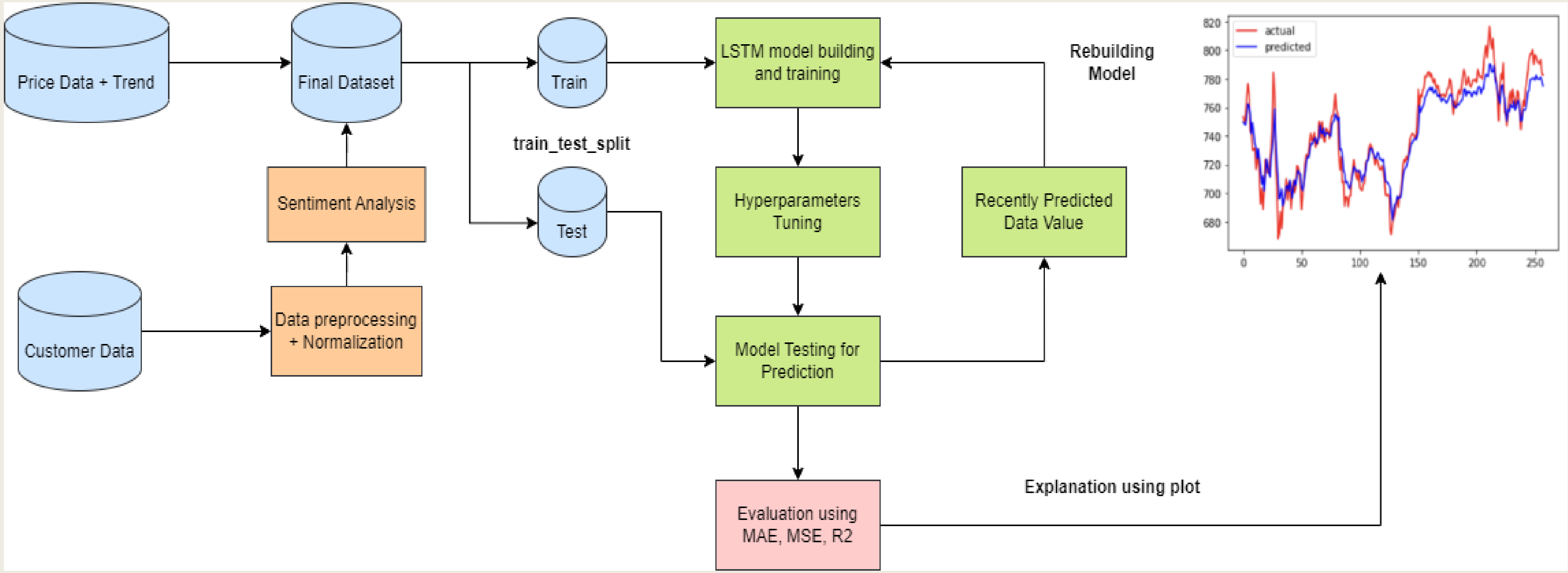
PRICING OPTIMIZATION USING LSTM

- Scaling predictions independently of accuracy
- An advanced Long Short-Term Memory (LSTM) – Fit LSTM to initial phase of test period
- Feature are previous days/months aggregate sales figure
- Output is median difference between model and price
- Features: Total revenue, customer attributes, and pure seasonal sales information





PRICE OPTIMIZATION ARCHITECTURE



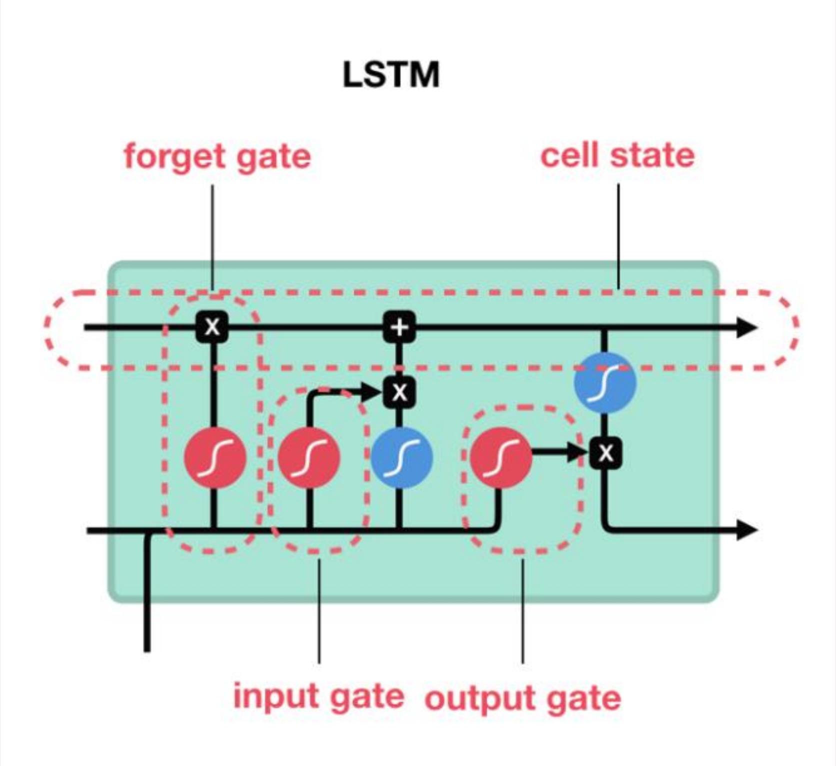
WHY LSTM?

- LSTM networks various state cells. These short and long-term memory cells rely on the state of these cells.
- These memory cells act as an aide for the model to remember historical context as predictions made by the network are influenced by past experiences of inputs to the network.
- This helps us make better predictions.

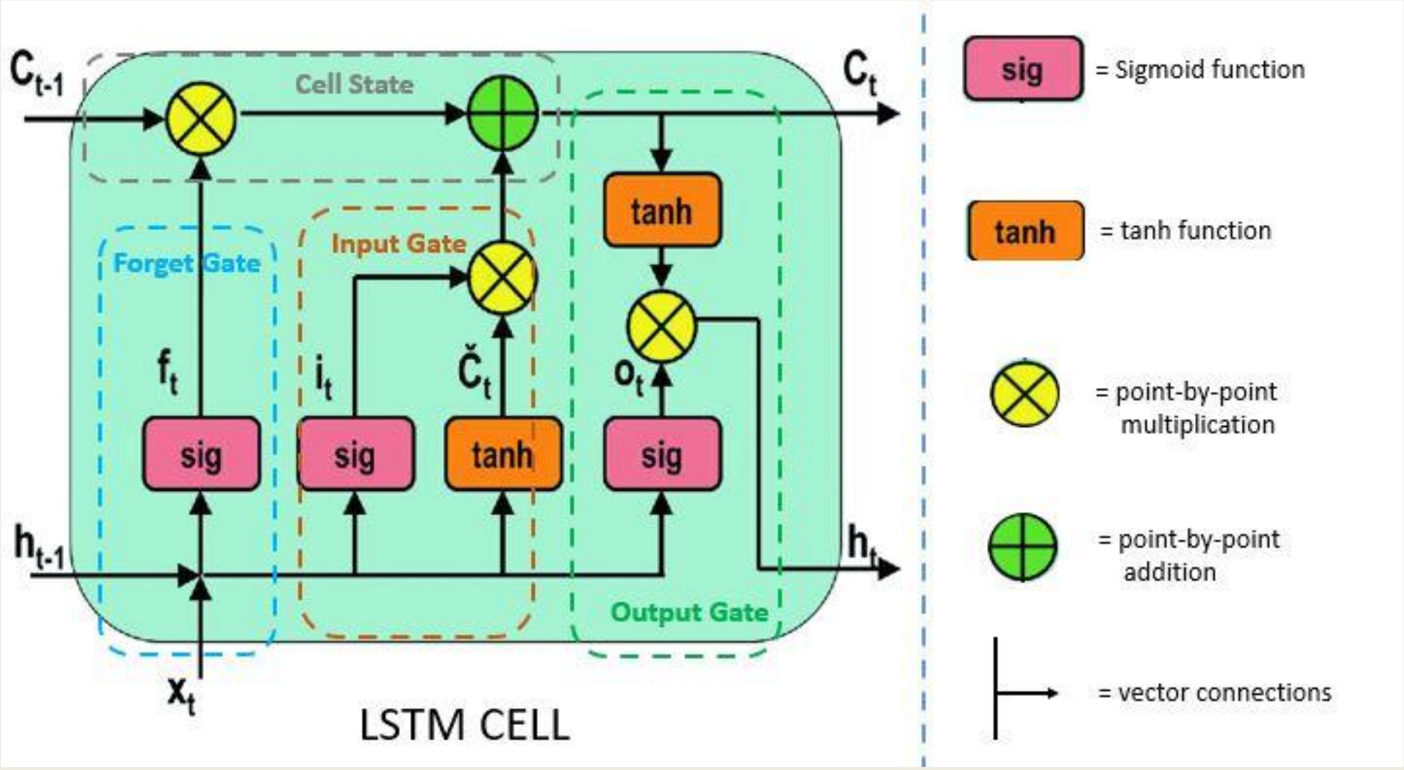
MODEL DESCRIPTION - LSTM

- **Input gate**: i_t
- **Forget gate**: $\mathrm{f_t}$
- **Output gate**: o_t
- **Cell gate**: c_t
- **Weight matrix**:

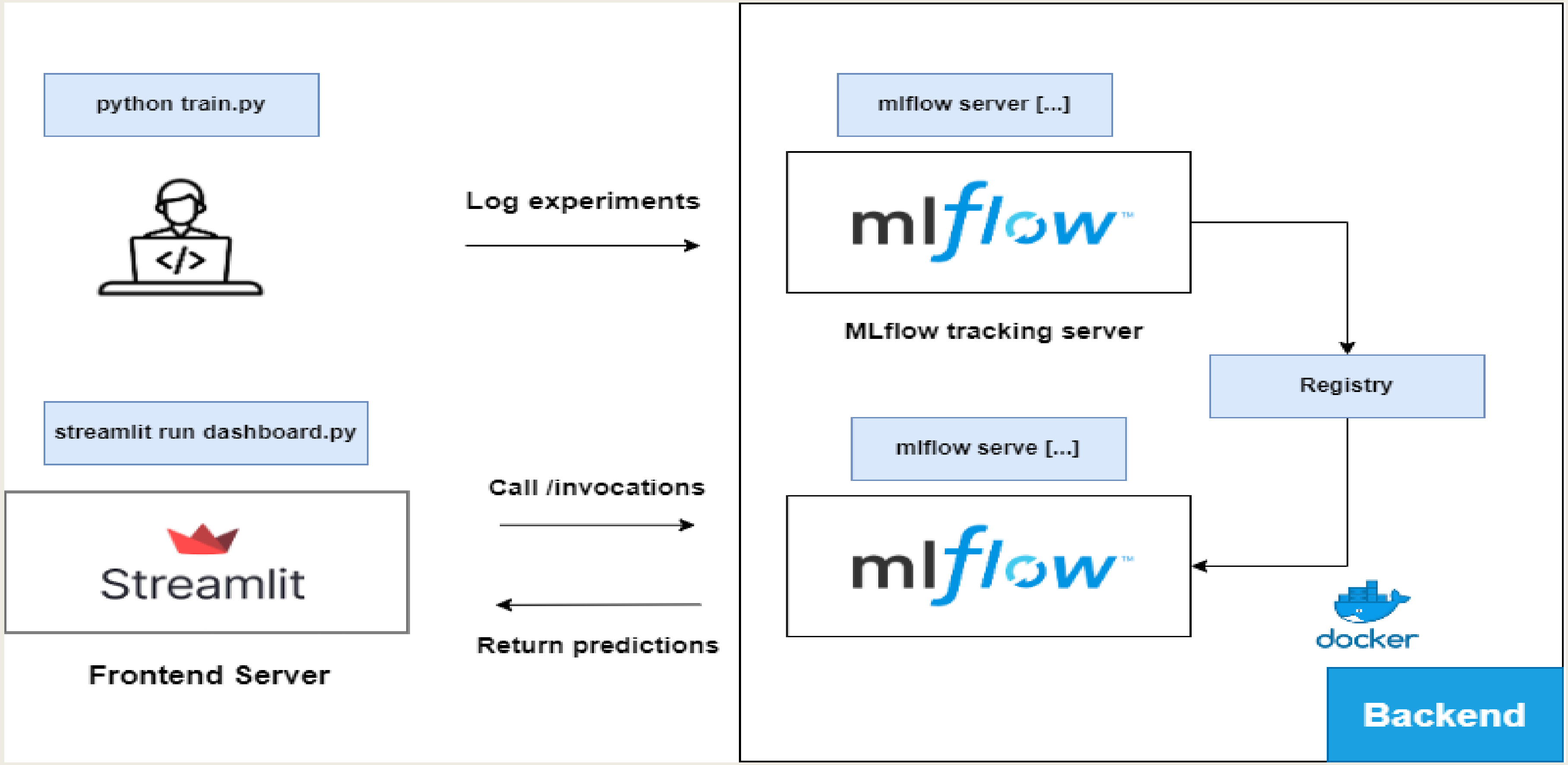
$$W_f\ ,\ W_i\ ,\ W_o\ ,\ W_c$$
$$W_f\ ,\ W_i\ ,\ W_o\ ,\ W_c$$
$$b_f,\ b_i\ ,\ b_o\ ,\ b_c$$



MODEL DESCRIPTION – LSTM (cont)



MACHINE LEARNING MODEL DEPLOYMENT ARCHITECTURE



DESCRIPTION EACH STAGE IN MODEL DEPLOYMENT

Model Development

- Start by creating and training your machine learning model.
- Save the trained model in a serialized form as a file. MLflow is used to log the model, creating a folder called **mlruns** and storing the experiments and data inside it.

Model Tracking

- Run a script with MLflow to track and log the model locally.
- Launch an MLflow tracking server, connecting it to store models remotely.

Model Registry

- Use MLflow's Model Registry to manage the model lifecycle.
- Register models from specific runs, providing names for identification.

DESCRIPTION EACH STAGE IN MODEL DEPLOYMENT (cont)

DEPLOYING MODEL AS A REST API

- Using MLflow's deployment capabilities, convert the registered model into a REST API endpoint that can be queried to get predictions.
- This involves running mlflow serve with additional arguments, including the model to deploy. The API endpoint is created, usually called "**invocations**."

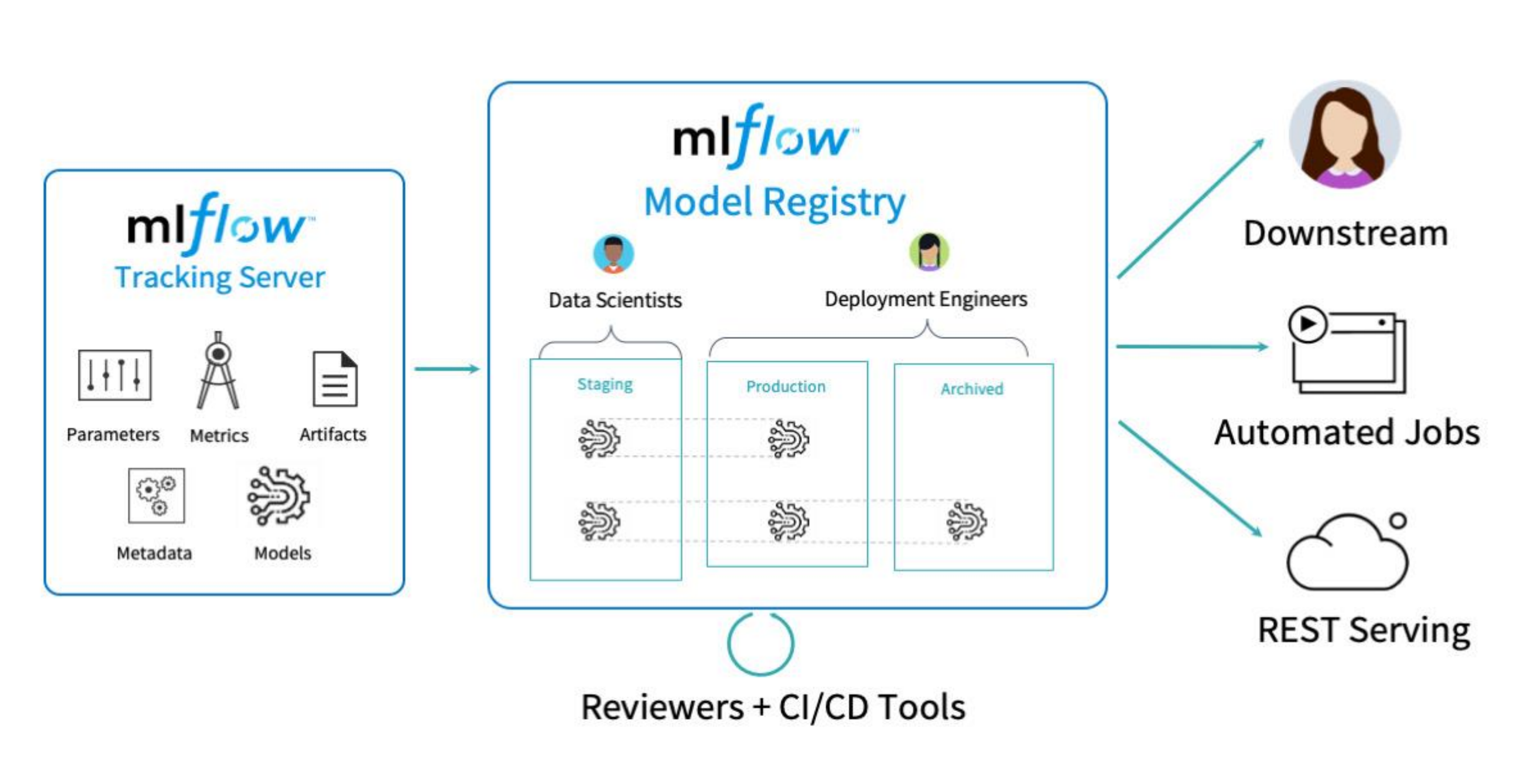
BUILDING THE FRONT END

- Create a user interface with Streamlit, a Python-based dashboarding tool.
- Start the Streamlit server by running “**streamlit run”** followed by the filename containing the dashboard code.
- The Streamlit dashboard allows users to interact with the machine learning model, providing inputs and receiving predictions.

DEPLOYING MODEL AS A REST API

- Users interact with the Streamlit dashboard, providing input data
- The Streamlit application, running on the server, communicates with the MLflow-deployed REST API endpoint to obtain predictions from the machine learning model.

MLFLOW REGISTRY



TECHNICAL STACKS

Libraries

NumPy

Numpy can be used to perform a wide variety of mathematical operation arrays

pandas

Pandas is a fast, powerful, flexible and easy to use open-source data analysis and manipulation tool, built on top of the Python programming language

matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualization in Python

scikit learn

Scikit-learn is a popular open-source machine learning library in Python, providing tools for data analysis, modeling, and predictive data science. (Including Random Forest, XGBoost, and Linear Regression.

K Keras

Keras is an open-source library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library (Use for LSTM)

PostgreSQL is an open-source relational database management system emphasizing extensibility and SQL compliance

Deployment

mlflow

MLflow is an open source platform for the machine learning lifecycle

Streamlit

Streamlit is an open-source python library for creating and sharing web apps for data science & machine learning projects

CORE FUNCTIONALITIES

SALES FORECASTING MODULE

- **Historical Analysis**: Enables retailers to explore and analyze historical sales data with interactive visualizations, allowing for a deep understanding of past performance.
- **Forecasting Tools**: Incorporates a hybrid model for accurate sales predictions, considering seasonality, trends, and external factors. Retailers can customize forecast parameters to align with specific business needs.

PRICE OPTIMIZATION MODULE

- **Product-specific Insights**: Provides detailed insights into each product, including cost price, competitor pricing, and historical sales, aiding in informed pricing decisions.
- **Dynamic Pricing Recommendations**: Utilizes the LSTM model to dynamically recommend optimal prices for products, considering market conditions, competitor strategies, and internal goals.

MODEL PERFORMANCE METRICS

Mean Absolute Error

Measures the average absolute difference between actual and predicted sales values. A lower MAE indicates a more accurate forecasting model.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Mean Squared Error

Quantifies the average squared difference between actual and predicted sales values. MSE penalizes larger errors more heavily, providing insights into the model's overall accuracy.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

R^2 SCORE

Evaluates the proportion of variance in the sales data that is captured by the forecasting model. An R2 score close to 1 signifies a model that effectively explains

$$R^2 = 1 - \frac{SS_{res}}{SS_{total}} = \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \mu)}$$

where γ_i is the true value, $\widehat{\gamma}_i$ is the prediction value, and n is the number of observations.

SS_{res} is the sum of squares of residuals, SS_{total} is the total sum of squares,

PRICE OPTIMIZATION METRICS

PROFIT MARGIN IMPROVEMENT

Measures the increase in profit margins achieved through the optimized pricing strategy. The system aims to enhance profitability while maintaining competitiveness.

PRICE ELASTICITY ANALYSIS

Assesses the sensitivity of demand to price changes. Understanding price elasticity aids in setting optimal price points for maximizing revenue.

TIMELINE AND ROADMAP

Initiation

20/11/2023 – 23/11/2023

- Milestone 1: Kickoff Meeting
- Milestone 2: Data Collection and Cleaning.

Model Development

24/11/2023 – 01/12/2023

- Milestone 3: Sales Forecasting Model
- Milestone 4: Price Optimization Model

Web Development

02/12/2023 – 09/12/2023

- Milestone 5: UI/UX Design
- Milestone 6: Streamlit and MLflow Integration

Testing and Optimization

10/12/2023 – 14/12/2023

- Milestone 7: User Acceptance Testing
- Milestone 8: Model Optimization and Fine-Tuning

Deployment and Monitoring

15/12/2023 – 17/12/2023

- Milestone 9: Streamlit and MLflow Deployment
- Milestone 10: System Integration and Monitoring Setup

USER INTERFACE

The UI encompasses the following key components:

1. Dashboard
2. Sales Forecasting Module
3. Pricing Optimization Module
4. Customization and Settings
5. User Guidance and Help Center

USER INTERFACE – DETAIL

Dashboard

Purpose: Offers a centralized view of critical information, including real-time sales forecasts, pricing recommendations, and performance metrics.

Features: Interactive charts, graphs, and widgets provide a dynamic overview of sales trends, inventory status, and pricing strategies.

Sales Forecasting Module

Purpose: Enables retailers to anticipate future sales trends and plan inventory accordingly.

Features: Intuitive controls for selecting timeframes, interactive visualizations for sales predictions, and detailed breakdowns by product, channel, and region.

Pricing Optimization Module

Purpose: Empowers retailers to set competitive and profit-maximizing prices for their products.

Features: Dynamic pricing tools, price elasticity insights, and competitive benchmarking data aid in formulating effective pricing strategies.

USER INTERFACE – DETAIL (cont)

User Guidance and Help Center

Purpose: Supports users with onboarding, feature explanations, and troubleshooting assistance.

Features: User guides, tooltips, and a comprehensive help center contribute to a smooth user experience.

Customization and Settings

Purpose: Allows users to tailor the application to their specific needs and preferences.

Features: Adjustable parameters for forecasting models, pricing rules, and user-specific settings enhance flexibility and adaptability.

LIMITATION

- Categorical features are often dominant in pricing but it can be challenge to encode categorical features with large numbers of possible values (The type of product, type of sales area, type of customer, ...)
- Lack of some knowledge about MLOps to implement our MVP
- Lack of some knowledge about business domain to evaluate some metrics relevant to price and sales.

FUTURE ENHANCEMENTS

1. Cloud Deployment:

- Deploy containers to a cloud environment (GCP or AWS) for scalable and reliable hosting.

2. Kubernetes Orchestration:

- Utilize Kubernetes to orchestrate containers, managing deployment, scaling, and resource allocation.

3. CI/CD Pipeline:

- Implement CI/CD pipelines and data pipelines to automate testing, build processes, and deployment.

CONCLUSION

In conclusion, our Minimum Viable Product (MVP) represents a revolutionary solution for retailers, leveraging data science and machine learning to optimize sales forecasting and pricing strategies.

By offering a streamlined, data-driven approach, our MVP addresses the challenges hindering growth and profitability in the fast-paced retail landscape. We envision a future where businesses can navigate market complexities with high efficiency, and adaptability.