SHIPMENT PRICING PREDICTION

By

SHINY

SRIRAM

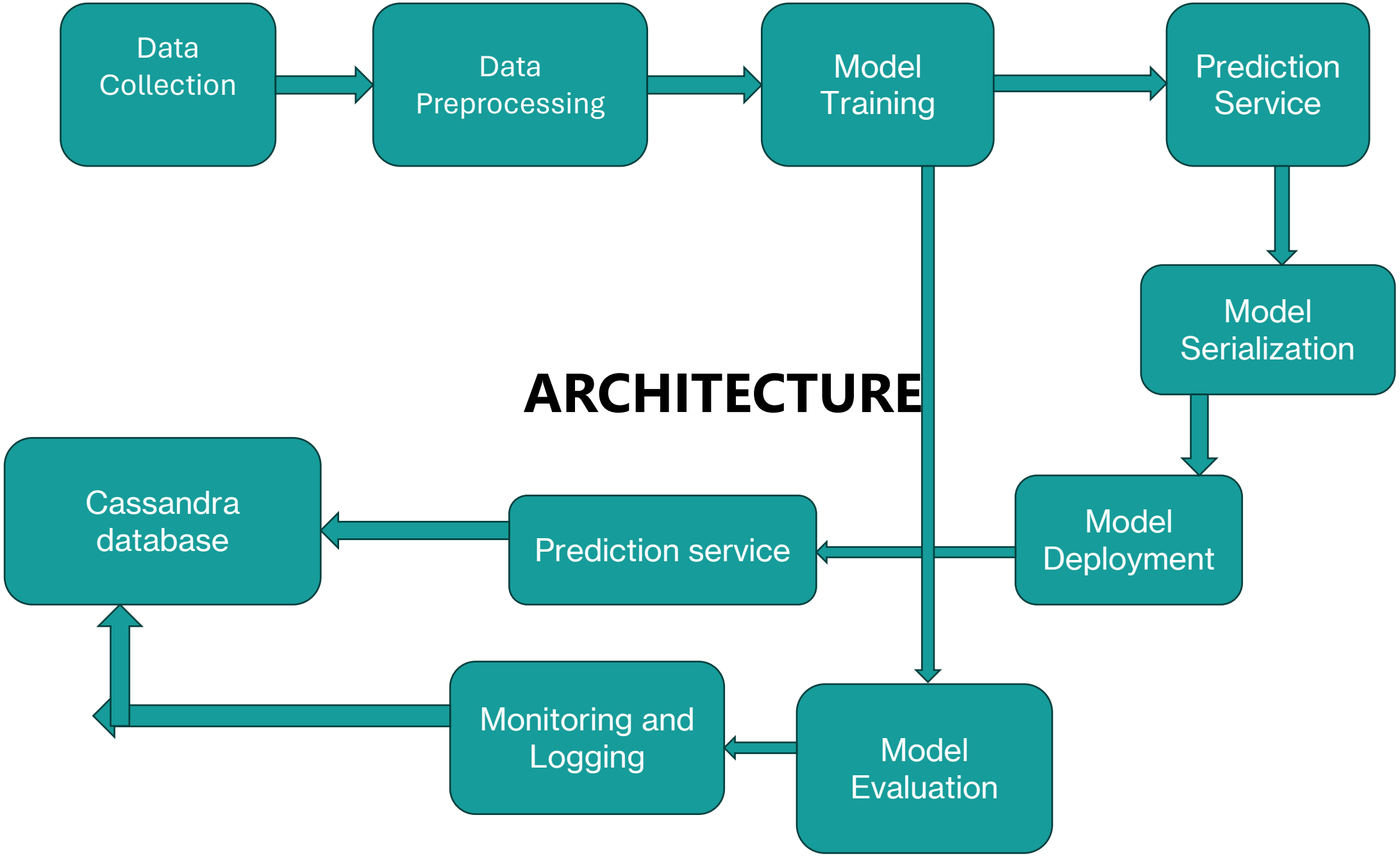
GOBIKRISHNAN

OBJECTIVE

The primary objective of the Shipment Pricing Prediction project is to develop a robust predictive model that accurately estimates the shipping costs based on various influencing factors such as distance, weight, dimensions, shipping method, and destination. This model aims to enhance operational efficiency and cost-effectiveness for logistics companies by providing real-time pricing insights. By leveraging historical shipment data and advanced machine learning techniques, the project seeks to minimize pricing discrepancies, improve customer satisfaction by offering transparent and competitive pricing, and optimize resource allocation. Ultimately, this project aims to facilitate better decision-making processes, drive profitability, and maintain a competitive edge in the logistics and transportation industry**.**

Data Sharing Agreement

- Sample file name: "SCMS_Delivery_History_Dataset (1).csv"
- Length of date stamp: 8 digits (yyyymmdd)
- Number of Columns: 8
- Column names and data type:
  - Shipment_ID (Object)
  - Date_of_Shipment (Date)
  - Origin (Object)
  - Destination (Object)
  - Shipping_Route (Object)
  - Arrival_Time (Time)
  - Dep_Time (Time)
  - Price (Integer)


Data Collection
Data Preprocessing
Model Training
Prediction Service
Model Serialization
ARCHITECTURE
Cassandra database
Prediction service
Model Deployment
Monitoring and Logging
Model Evaluation

Data Validation and Transformation

Name Validation: Check file names against DSA regex. Move to "Good_Data_Folder" if valid, else to "Bad_Data_Folder".

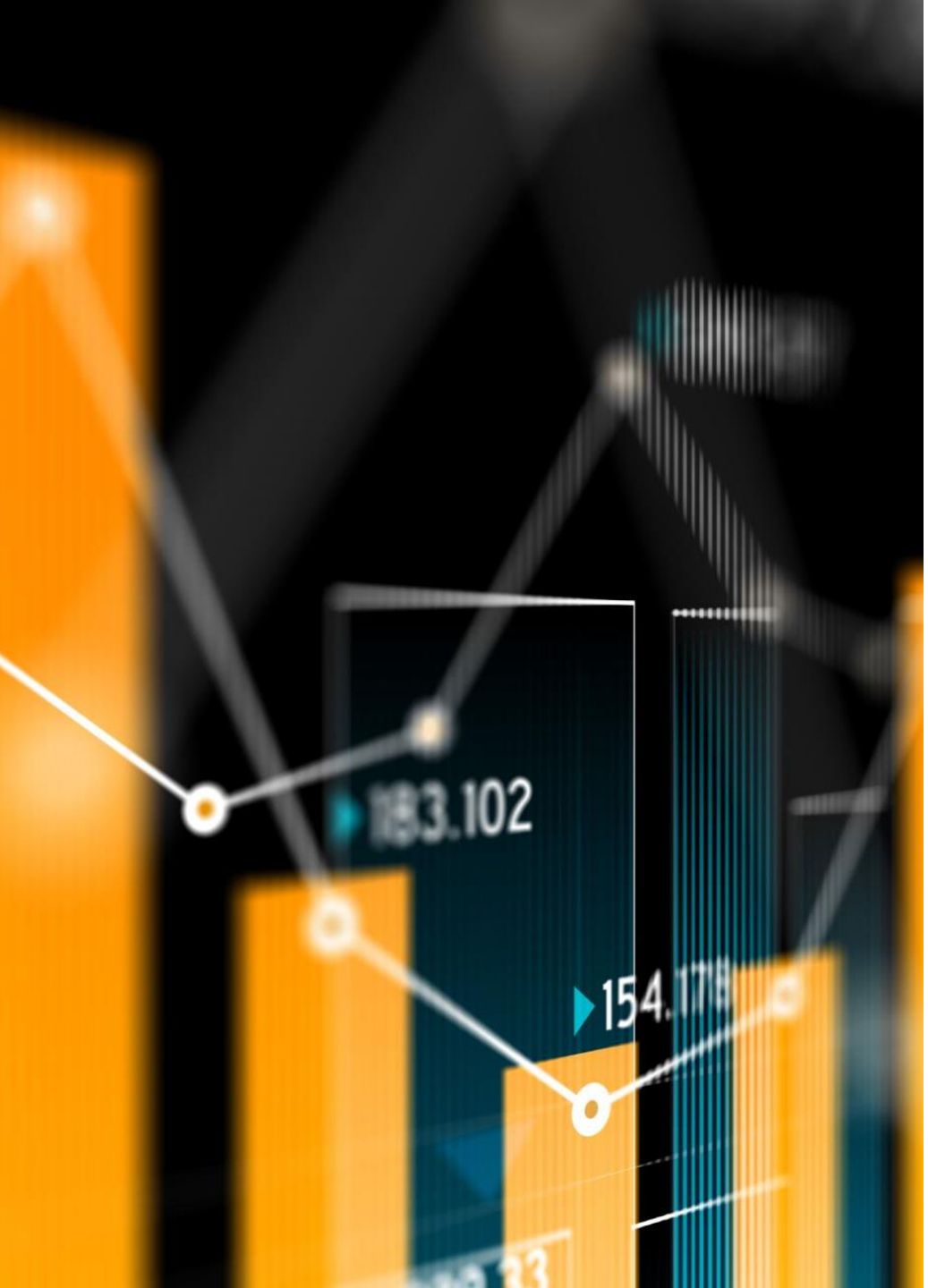
Number of Columns: Validate against schema. Move to "Bad_Data_Folder" if mismatched.

Name of Columns: Match against schema. Move to "Bad_Data_Folder" if not identical.

Data Type of Columns: Validate data types against schema. Move to "Bad_Data_Folder" if incorrect.

Null Values: Discard files with all NULL values. Move to "Bad_Data_Folder".

Data Transformation: Convert data types and preprocess. Move to "Good_Data_Folder" for further processing.

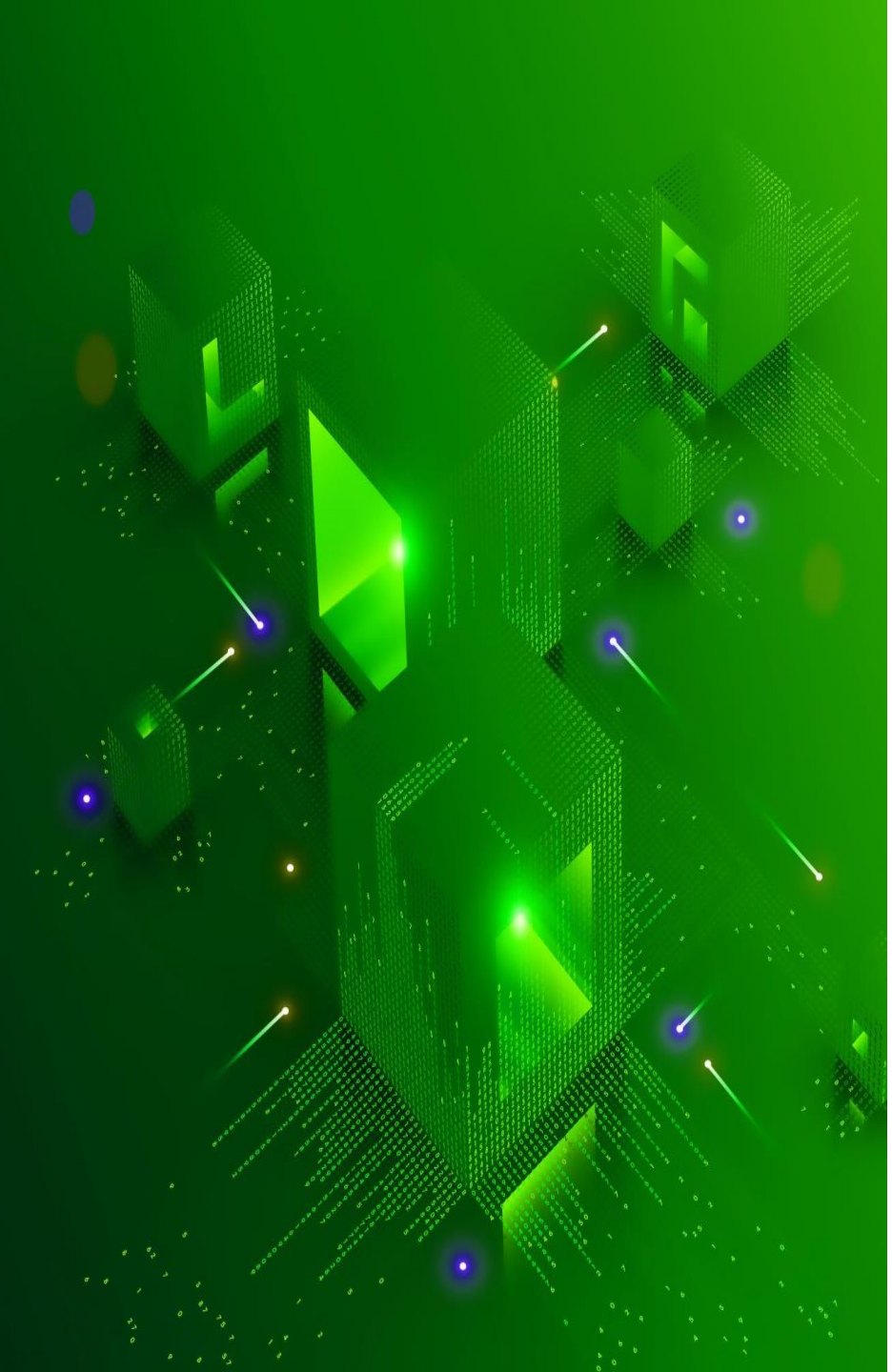


Data Insertion in Database

To manage and insert data into a Cassandra database, first, install the cassandra-driver and import necessary modules such as pandas, Cluster, and PlainTextAuthProvider. Define the connection configuration using a secure connect bundle and an authentication token. Connect to the Cassandra cluster and set the keyspace (keyspp) and table (shipment). Create the table if it doesn't exist, specifying columns like id, date_of_shipment, origin, destination, shipping route, arrival time, dep_time, and price. Insert a sample record into the table to verify functionality. Finally, query the table to retrieve data and convert the result set into a Pandas Data Frame for easy manipulation and display. This process ensures efficient data management and accessibility for further operations.

Model Training

- Split the data into training and testing sets.
- Initialize various regression models including RandomForestRegressor, LinearRegression, KNeighborsRegressor, SVR, DecisionTreeRegressor, AdaBoostRegressor, and XGBRegressor.
- Evaluate each model using cross-validation and print evaluation metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared Score.
- Train a RandomForestRegressor on the training data and evaluate it on both training and testing data.
- Train a DecisionTreeRegressor on the training data and evaluate it on both training and testing data.
- Train an XGBRegressor on the training data and evaluate it on both training and testing data.

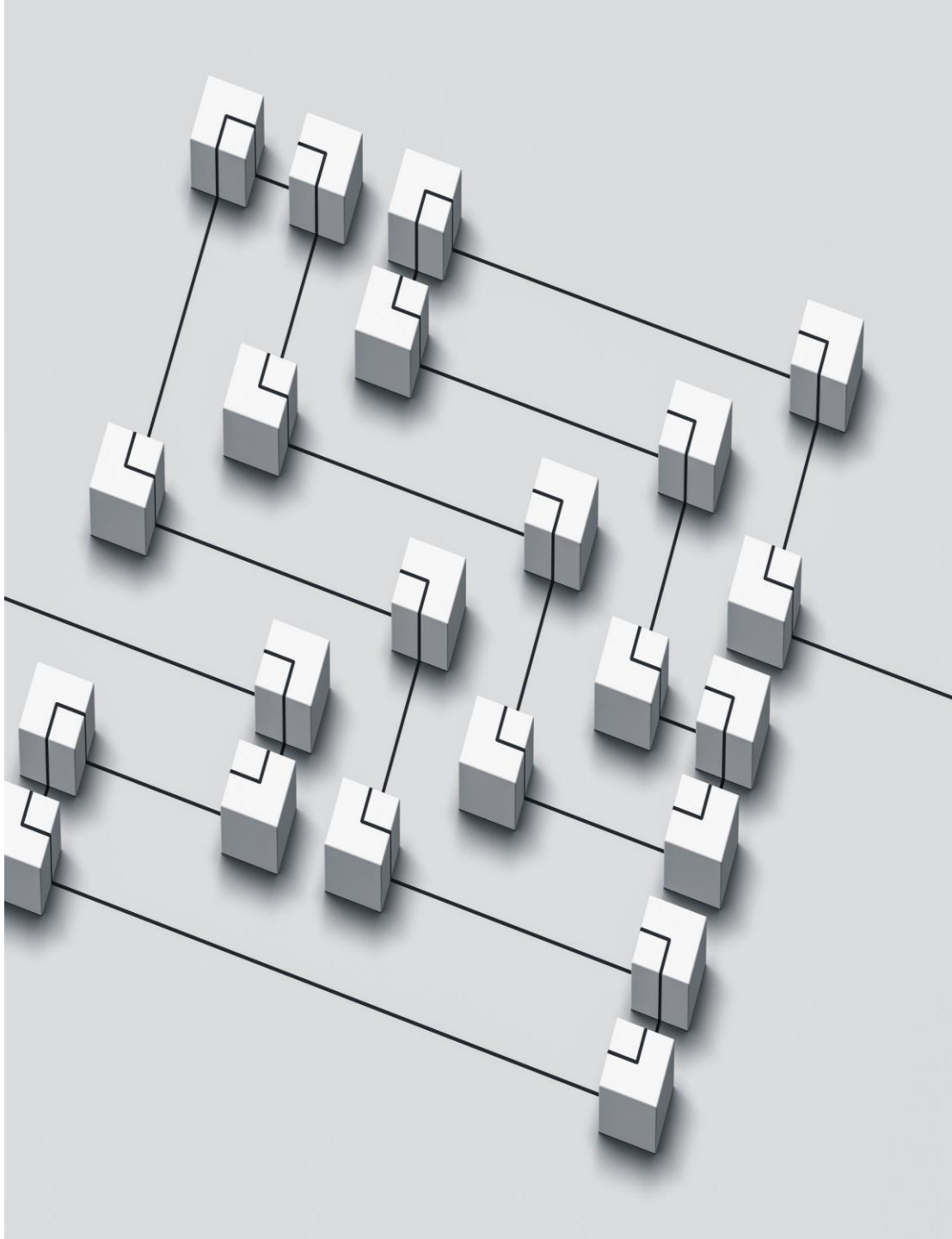


Model Selection

The model selection process involves initializing and evaluating three regression algorithms: Random Forest Regressor, Decision Tree Regressor, and XGBoost Regressor. Each model is trained on the training data and then used to make predictions on the testing data. Evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared score are calculated for each model. The model with the best performance based on these metrics can then be selected for further refinement or deployment.

Prediction

In the final stage of your model-building process, predictions are made using the trained models on the testing data. Specifically, you utilize the RandomForestRegressor and XGBRegressor models to predict pack prices based on the features in the testing dataset. These predictions provide insights into how well the models generalize to unseen data. Following the predictions, you can assess the models' performance using various metrics, such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared score, which were calculated earlier in the code. These evaluation metrics help gauge the accuracy and effectiveness of the models in predicting pack prices.



Q & A

Q1) What's the source of data?

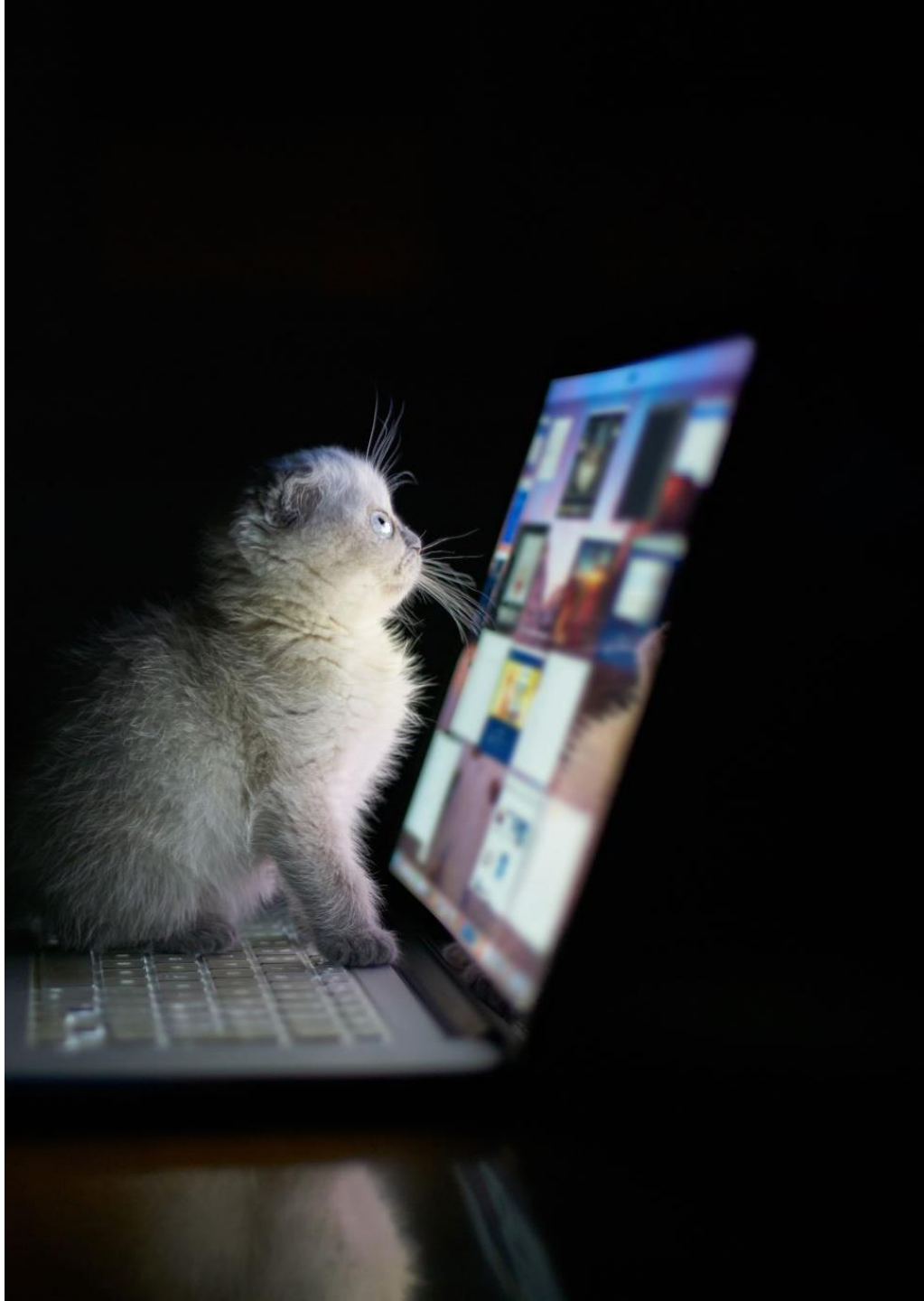
The data for shipment pricing prediction is sourced from the client, typically provided in multiple batches, each comprising multiple files.

Q2) What was the type of data?

The data for shipment pricing prediction consists of a combination of numerical and categorical values, including information such as country, shipment mode, product group, brand, and pricing details.

Q3) What's the complete flow you followed in this project?

The complete flow for the shipment pricing prediction project involves several steps, including data preprocessing, feature engineering, outlier detection, model training, evaluation, and model saving. For detailed understanding, please refer to slide 5 of the project documentation.



THANK YOU