Architecture
Back Order Prediction

Written By	Vikas , Chaudhary Shubham R. , Bijayalaxmi Kar
Document Version	0.1
Last Revised Date	11 Sept 2021

Document Control

Change Record :

Version	Date	Author	Comments
0.1	10-Sept-2021	Vikas , Chaudhary Shubham R. , Bijayalaxmi Kar	Introduction & Architecture defined

Approval Status :

Version	Review Date	Reviewed By	Approved By	Comments
0.1				

Content :

1. Introduction

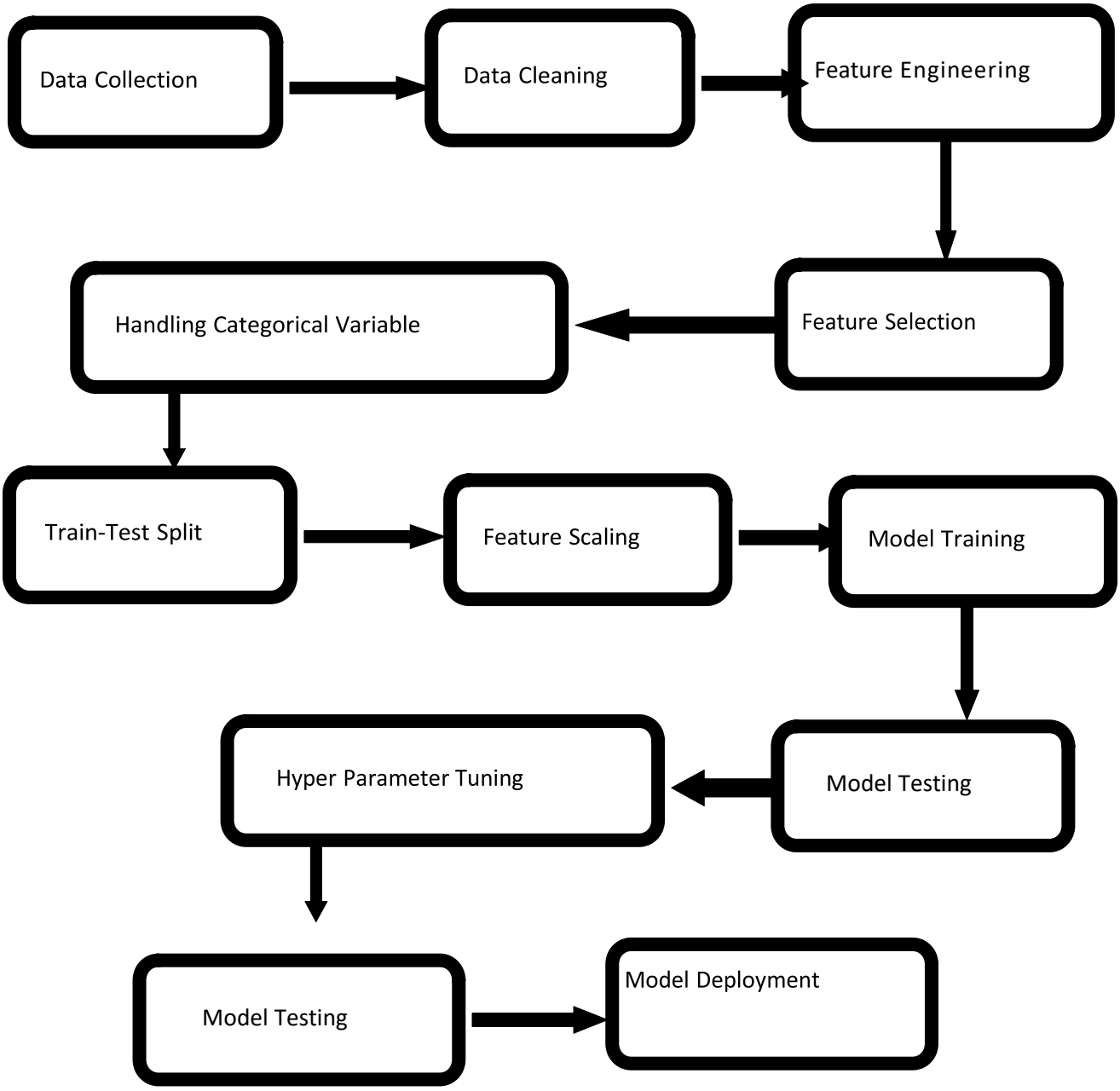
1.1 Why this Architecture Design Document?

The purpose of this document is to provide a detailed architecture design of the Back Order Prediction Project by focusing on four key quality attributes:

usability, availability, maintainability, testability.

This document will address the background for this project, and the architecturally significant function requirements. The intension of this document is to help the development team to determine how the system will be structured at the highest level. Finally, the project coach can use this document to validate that the development team is meeting the agreed-upon requirements during the evaluation of the team's efforts.

2. Architecture



3. Architecture Description

3.1. Data Description

We have 2 lakh Dataset row columnar data includes sku , national_inv , lead_time , in_transit_qty , forecast_3_mon th , forecast_6_month , forecast_9_month , sales_3_month , sales_6_month , sales_9_month , min_bank , potential_issue , pieces_past_due, perf_6_month_avg , perf_12_month_avg , local_bo_qty, deck_risk , oe_constraint , ppap_risk , stop_auto_buy , rev_stop , went_on_backorder. These is given in the comma separated value format (.csv).These data is Given By Company Team which contains both the test data and train data.

3.2. Data Cleaning / Data Transformation

In the Cleaning process, we have cleaned up all the data because data is present in very bad format which was can not recognized by machine .So data Cleaning is done very first by data validation methods. In which we are create a json file in which name of file , numbers of columns etc information present.
File name is given in “BackOrder_08012020_120000” Format.

3.3. Exploratory Data Analysis

In EDA we have seen various insights from the data so we have selected which column is most important and dropped some of the columns by observing them spearman rank co-relation and plotting their heatmap from seaborn library also we done null value managed in an efficient manner and also implemented categorical to n umerical transfer of column method here.

3.4. Event Log

The system should log every event so that the user will know what process is running internally. Logging is implemented using python's standard logging library. Initial step-by-step description:-

- The system should be able to log each and every system flow.
- System must be able to handle logging at greater scale because it helps debugging the issue and hence it is mandatory to do.

3.4. Data Insertion into Database

- Database Creation and connection - Create a database with name passed. If the database is already created, open the connection to the database.
- Table creation in the database.
- Insertion of files in the table

3.5. Export Data from Database

- Data Export from Database - The data in a stored database is exported as a CSV file to be used for Data Pre-processing and Model Training.

3.6. Data Pre-processing

Data Pre-processing steps we could use are Null value handling, Categorical to Numerical Transformation of columns , Splitting Data into Dependent and Independent Features , Robust Scaling , Remove those columns which are does not participate in model building Processes , Imbalanced data set handling, Handling columns with standard deviation zero or below a threshold, etc.

3.7. Model Creation / Model Building

After cleaning the data and completing the feature Engineering/ data Perprocessing. we have done splitted data in the train data and test data using method build in pre-processing file and implemented various Classification Algorithm like RandomForestClassifier and XgBoostClassifier also calculated their accuracies on test data and train data.

3.8. Hyperparameter Tuning

In hyperparameter tuning we have implemented randomized search cv or grid search cv and from that we also implemented cross validation techniques for that. From that we have choose best parameters according to hyperparameter tunning and best score from their accuracies so we got 99% accuracy in our random forest classifier after hyper parameter tuning.

3.9. Model Dump

After comparing all accuracies and checked all ROC, AUC curve accuracy we have choose hyper parameterized random forest classifier and xgboost classifiers as our best model by their results so we have dumped this model in a pickle file format with the help of load_model and find_best_model method build in

save_model.py python module.

3.10. Data from User

Here we will collect user's requirement to predict whether a product in backorder or not their forecast_3_month, forecast_6_month, forecast_9_month, sales_3_month, sales_6_month, sales_9_month, perf_6_month_avg, perf_12_month_avg .

3.11. Data Validation

Here Data Validation will be done, given by the user.

3.12. Model Call for specific input

Based on the User input will be throwing to the backend in the variable format then it converted into pandas data frame then we are loading our pickle file in the backend and predicting whether product come in backorder or not as an output and sending to our html page.

3.13. Saving Output in .csv file

After calling model Recipe/Output will be recommended, this output will be saved in .csv file and it will be used to show the Output.

3.14. User Interface

In Frontend creation we have made a user interactive page where user can enter their input values to our application. In these frontend page we have made a form which has beautiful styling with CSS and bootstrap. These HTML user input data is transferred in variable format to backend. Made these html fully in a decoupled format.

3.15. Deployment

We will be deploying the model with the help of AWS cloud platforms.

This is a workflow diagram for the Back-Order Prediction Application …….