BackOder Prediction

Objective:

Development of a predictive model for forecast backorder. The model will tell whether a product will go into a backorder or not.

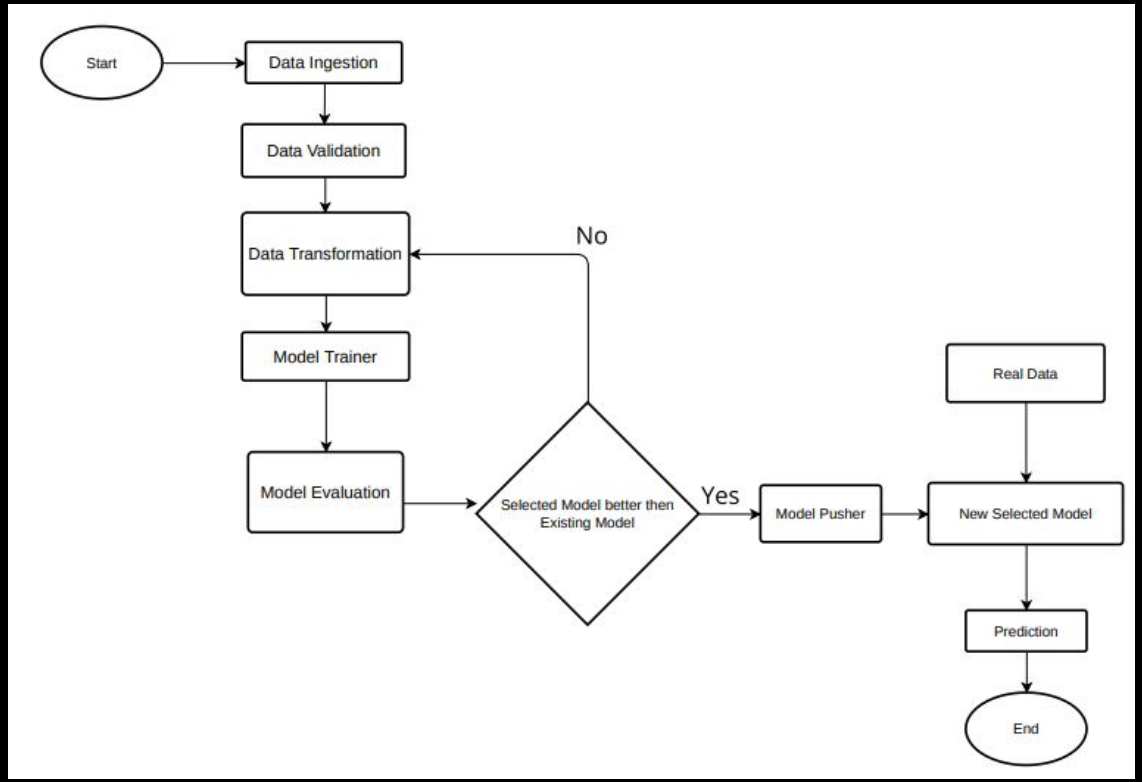
Benefits:

- Detect product which went to backorder.
- Preventing unexpected stain on production, logistic and transportation.
- Planning and streamlining product manufacturing.

Data Sharing Agreement:

- Separate Files for Training and Testing
- Sample Train file name (ex Kaggle_Training_Dataset_v2.csv)
- Sample Test file name (ex. Kaggle_Test_Dataset_v2.csv)
- Number of Columns
- Column names
- Column data type

Architecture:



Data Validation and Data Transformation:

- Name Validation - Validation of files name as per the DSA. We have a Json file containing the name of training and testing file.
- Number of Columns - Validation of number of columns present in the files, and if it isn’t in the DSA don’t use the the column
- Name of Columns- The name of columns is validated and should be the same as given in the schema file.
- Data type of Columns- The data type of columns is given and should be the same as in the schema file.
- Null values in Columns- if any of the columns in a file have all the values as null or missing we can drop them

Model Training:

- Data Preprocessing
    - Performing EDA to get insight of data like identifying distribution, outliers, trend among data etc.
    - Check for null values in the columns. If present impute the null values.
    - Encode the categorical values with numeric values
    - Perform PCA for dimensionality reduction
    - Perform Standard Scaler to scale down the values

Model Selection:

We find the best model and parameter. For that we are using GridSearchCV for hyperparameter tuning. We calculate the accuracy docre for both models and select the model with best core.

Model Pusher:

The selected model is compared with an existing deployed model. If the model has a better score then existing model then new model will be deployed

Prediction:

- We perform data pre-processing techniques on it
- The current deployed model is loaded for the preprocessed data
- Prediction are made on the preprocessed data
- The prediction is shown to the user who requested it.

Q&A:

Q1) What’s the source of data?

The data for training and testing is provided by the client as url for a .rar file containing separate training and testing files

Q2) What was the type of data?

The data was the combination of numerical and categorical values

Q3) What’s the complete flow you followed in this Project ?

Refer slide 4th for better Understanding

Q4) How logs are managed?

We are using a fog file with all the log statements based on the stage of training, timestamp and file name.

Q5) What technique were you using for pre-processing?

- Removing unwanted attributes
- Checking and changing distribution of continuous value
- Cleaning data and imputing if null values are present
- Converting categorical data into numeric values
- Doing Dimensionality reduction
- Scaling the data