LOW LEVEL DESIGN (LLD)
Backorder Prediction

Written By:- GAURAV MALODE

Contents

1. Introduction:

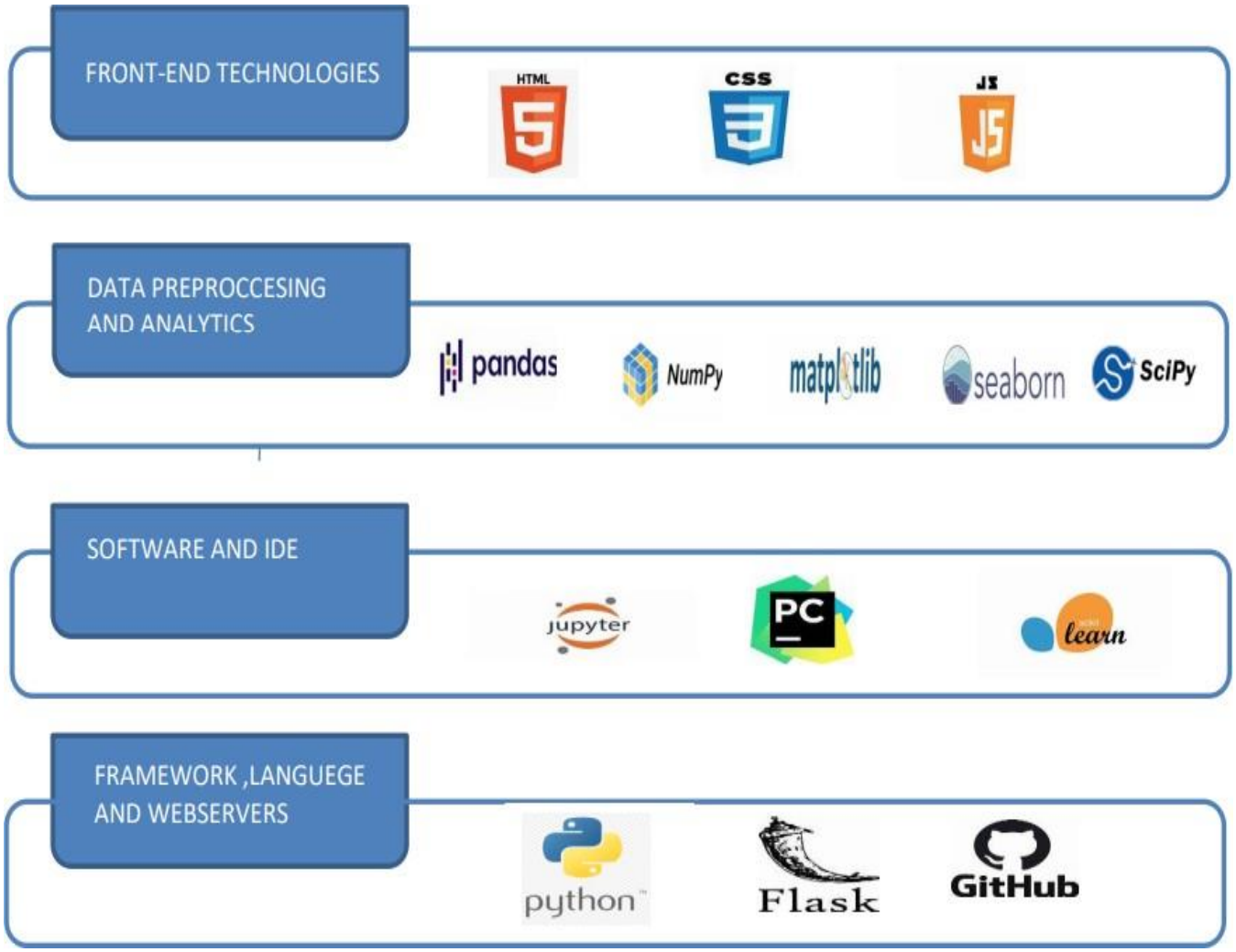
1.1. What is a Low-Level design document?

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for the Food Recommendation System. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

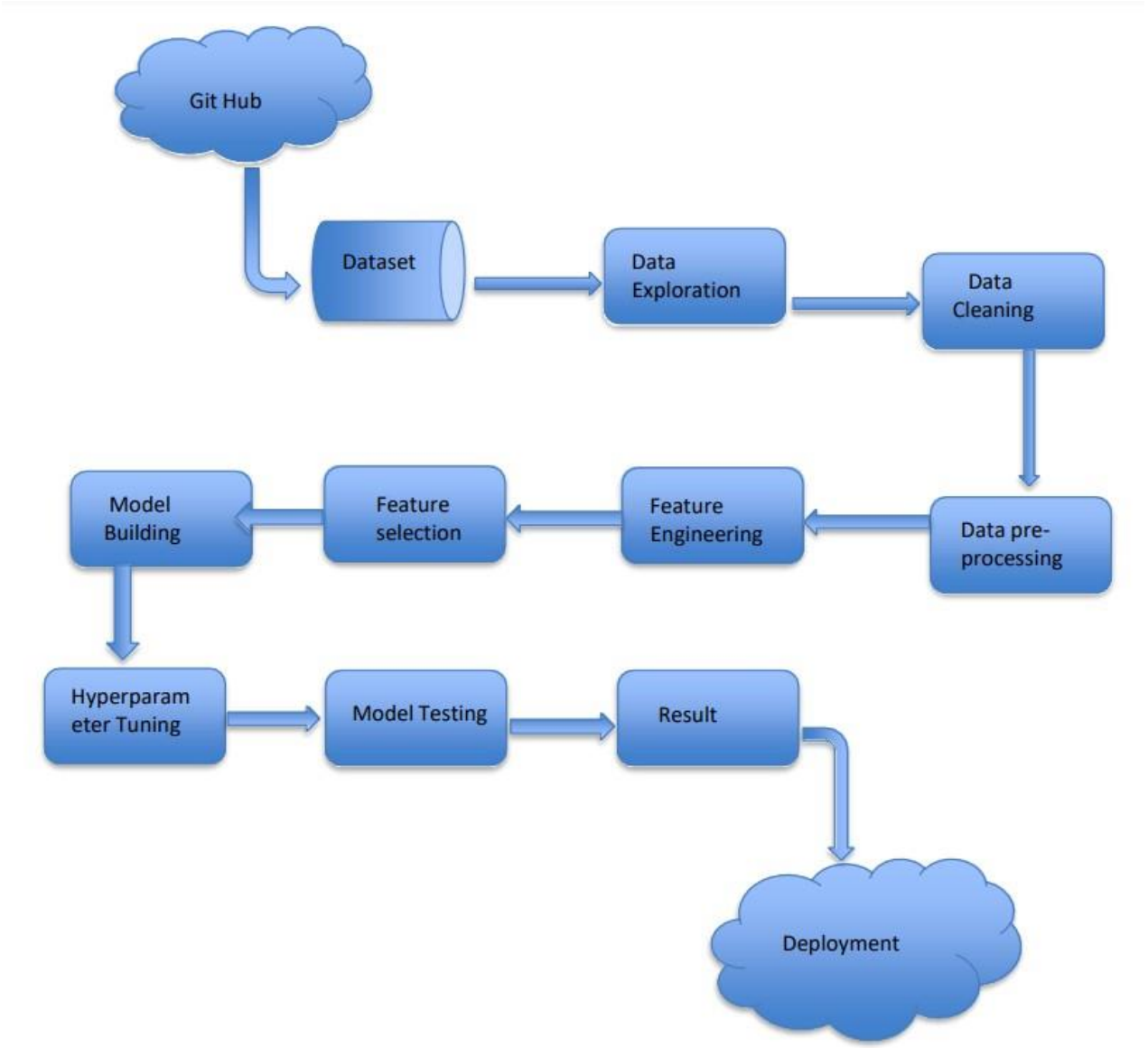
1.1.1 Scope:

Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organisation may be defined during requirement analysis and then refined during data design work

1.1.2 Technology Stack :



2. Architecture:



3. Architecture Description:

3.1. Data Description:

The dataset of this research has been published in Kaggle. It is divided into training and testing datasets. Data Source:
https://github.com/rodrigosantis1/backorder_prediction/blob/master/dataset.rar
The dataset is highly imbalanced which should be addressed for accurate predictions by the model; each dataset contains 23 attributes with 1,687,862 and 242,077 observations for the training and testing sets, respectively.

3.2. Data Transformation:

In the Transformation Process, we will convert our original dataset which is in CSV Format to a pandas data frame. After reading the dataset we have two pandas data frame train and test concatenated into one data frame.

3.3. Exploratory Data Analysis:

Exploratory Data Analysis, or EDA, is an important step in any Data Analysis or Data Science project EDt. A is the process of investigating the dataset to discover patterns, and anomalies (outliers), and form hypotheses based on our understanding of the dataset.

3.4 Data Insertion into Database:

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

3.5. ExportData from Database:

The data in a stored database is exported as a CSV file to be used for Data Preprocessing and Model Training.

3.6. Data Cleaning:

Data cleansing or data cleaning is the process of detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or database and refers to identifying incomplete, incorrect, inaccurate or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarse data.

3.7 Data Pre-processing:

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. When creating a machine learning project, it is not always the case that we come across clean and formatted data.

3.8 Feature Engineering:

Feature engineering is the process of using domain knowledge of the data to create features that make machine learning algorithms work. If feature engineering is done correctly, it increases the predictive power of machine learning algorithms by creating features from raw data that help facilitate the machine learning process.

3.9 Feature Selection:

A feature selection algorithm can be seen as the combination of a search technique for proposing new feature subsets, along with an evaluation measure which scores the different feature subsets. The simplest algorithm is to test each possible subset of features finding the one which minimises the error rate. This is an exhaustive search of the space, and is computationally intractable for all but the smallest of feature sets.

3.10 Model Building:

A machine learning model is built by learning and generalising from training data, then applying that acquired knowledge to new data it has never seen before to make predictions and fulfil its purpose. Lack of data will prevent you from building the model, and access to data isn't enough.

3.11 Random Forest:

A random forest classifier, a random forest is a Meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

3.12 DecisionTreeClassifier:

Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

3.16. Model Validation:

For machine learning systems, we should be running model evaluation and model tests in parallel. Model evaluation covers metrics and plots which summarise performance on a validation or test dataset. Model testing involves explicit checks for behaviours that we expect our model to follow.

3.17 Flask:

Flask is a micro web framework written in python. Here we use flask to create an API that allows us to send data, and receive a prediction as a response. Flask supports extensions that can add application features as if they were implemented in Flask itself.

3.18STREAMLIT:

Streamlit is an open-source Python library. It enables you to create interactive web applications with ease. You can build data-driven dashboards, visualizations, and machine learning models. With a simple and intuitive syntax, Streamlit makes web app development accessible to developers of all levels

Unit test Cases:

Test Case Description	Pre-requisite	Expected Result
Verify the user should able to see their input data	1. Application is accessible 2. Application is responsive	Users can see their data in the form.
Verify the user can edit their information in the form	1.Application is responsive	User should be able to edit all input field
Verify whether the user gets Submit button to submit the inputs.	1.Application is accessible 2.Application is responsive	User should able to submit values
Verify whether the wrong information should not be submitted by the user	Application is secured	Correct information should be submitted