ARCHITECTURE DOCUMENT DESIGN (ADD)
Credit card default Prediction

Written By:- Swaraj Kakade

Document Control:

Change Record:

Version	Date	Author	Comments
0.1	12-May- 2023	Swaraj Kakade	Initial Architecture design Draft.
0.2	13-May-2023	Swaraj Kakade	Architecture Design appended and updated
0.3	15-May-2023	Swaraj Kakade	Final design appended

Reviews:

Version	Date	Reviewer	Comments
1.0	18-May-2023	Swaraj Kakade	Document Content , Architecture Design Updated

Approval Status:

Version	Date	Reviewed By	Approved By	Comments
1.1	20-May-2023	Swaraj kakade		Final

Contents

1. Introduction:

1.1. What is an Architecture design document?

The architecture design document is a technical document describing the components and specifications required to support the solution and ensure that the specific business and technical requirements of the design are satisfied. The architecture design document includes the conceptual, logical, and physical designs.
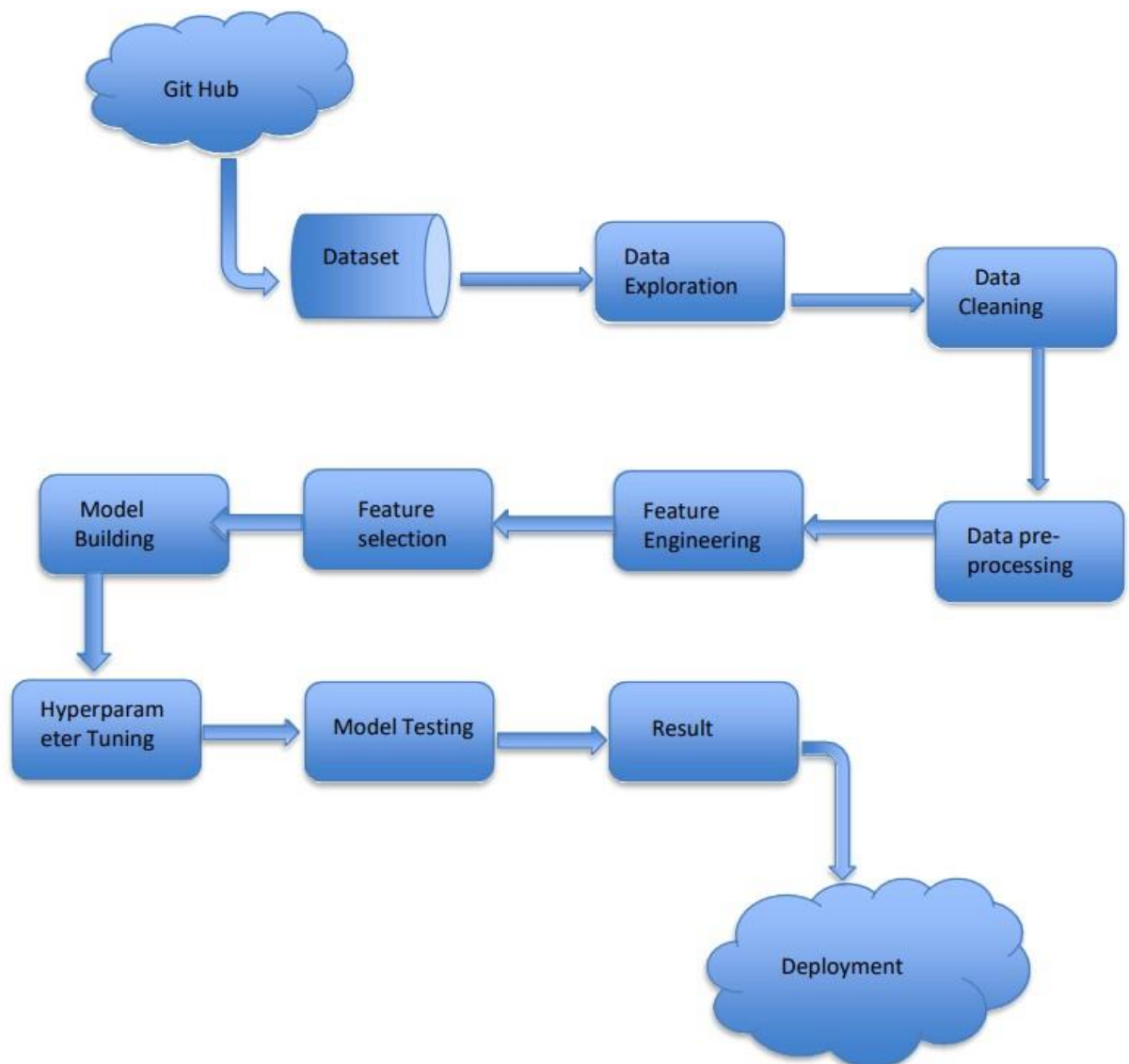
1.2. Scope:

The scope of this document is specific to the implementation of the Predictive Machine Learning Model to deploy on to the cloud. The supporting Libraries are defined in the Architecture Design. These python libraries are used to mining the data to understand the anomalies and frequent patterns from the row data. The resultant data can be used for predicting future sales volume with the help of different machine learning techniques.

1.1.2 Credit card default prediction Page :

Credit Card Default Prediction

MARRIAGE
Married

SEX
Male

EDUCATION
Graduate School

PAY_0
Zero credit

PAY_2
Zero credit

PAY_3
Zero credit

PAY_4
Zero credit

PAY_5
Zero credit

PAY_6
Zero credit

LIMIT_BAL

AGE

BILL_AMT1

BILL_AMT2

BILL_AMT3

2. Architecture:

3. Architecture Description:

3.1. Data Description:
The dataset of this research has been published in Kaggle Data Source:
https://www.kaggle.com/datasets/uciml/default-of-credit-card-clients-dataset
The dataset contains 25 attributes with 30,000 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 EDA. 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 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.

3.5. 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.

3.6.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.

3.7 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.

3.8   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.

3.9   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.

3.10 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.

3.11 Logistic Regression:

Logistic regression is a statistical model used for binary classification. It predicts the probability of an event belonging to one of two classes. It uses the logistic function (sigmoid function) to transform the input variables into a value between 0 and 1, representing the estimated probability. The model provides interpretable coefficients for each input variable, indicating their influence on the predicted probability. Logistic regression is trained using maximum likelihood estimation and can be evaluated using metrics such as accuracy, precision, recall, and the ROC curve.

3.12 Support Vector classifier:

The Support Vector Classifier (SVC), also known as Support Vector Machine (SVM), is a classification algorithm. It finds an optimal hyperplane that separates data points of different classes with the maximum margin. SVC can handle nonlinear data by using the kernel trick, and it aims to find a hyperplane with the largest margin between support vectors. It is trained by solving an optimization problem, and it can classify new data points based on their position relative to the hyperplane.

3.13 KNN Classifier:

KNN is a non-parametric algorithm that classifies data points based on their proximity to other data points. It assigns a class label to an unclassified data point based on the class labels of its nearest neighbors in the feature space. The "K" in KNN represents the number of nearest neighbors considered for classification. When a new data point is to be classified, KNN finds the K closest neighbors based on a distance metric (such as Euclidean distance) and assigns the class label that appears most frequently among those neighbors.

3.14. Model Validation:

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

3.15 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.

4. Database:

Cassandra DB is used to retrieve, insert, delete and update the customer input database. Flask is a micro web framework written in python. Here we use the flask to create an API that allows sending data, and receives a prediction as a response. Flask API's act as an interface between the user, ML- Model and DB. Users interact with the model and the result will be returned to the user by API. The Flask API will collect all the customer input data stored in Cassandra DB.

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