Insurance Premium Prediction

Low Level Design

Domain: Machine Learning

Creator: Narendran

Date: 25.11.2023

DOCUMENT VERSION CONTROL

Date issued	Version	Description	Author
Nov 25, 2023	0.0.1	First Draft	Narendran M

Chennai, India

+91 9884863650

narendas10@gmail.com

CONTENTS

INTRODUCTION

WHAT IS LOW-LEVEL DESIGN DOCUMENT?

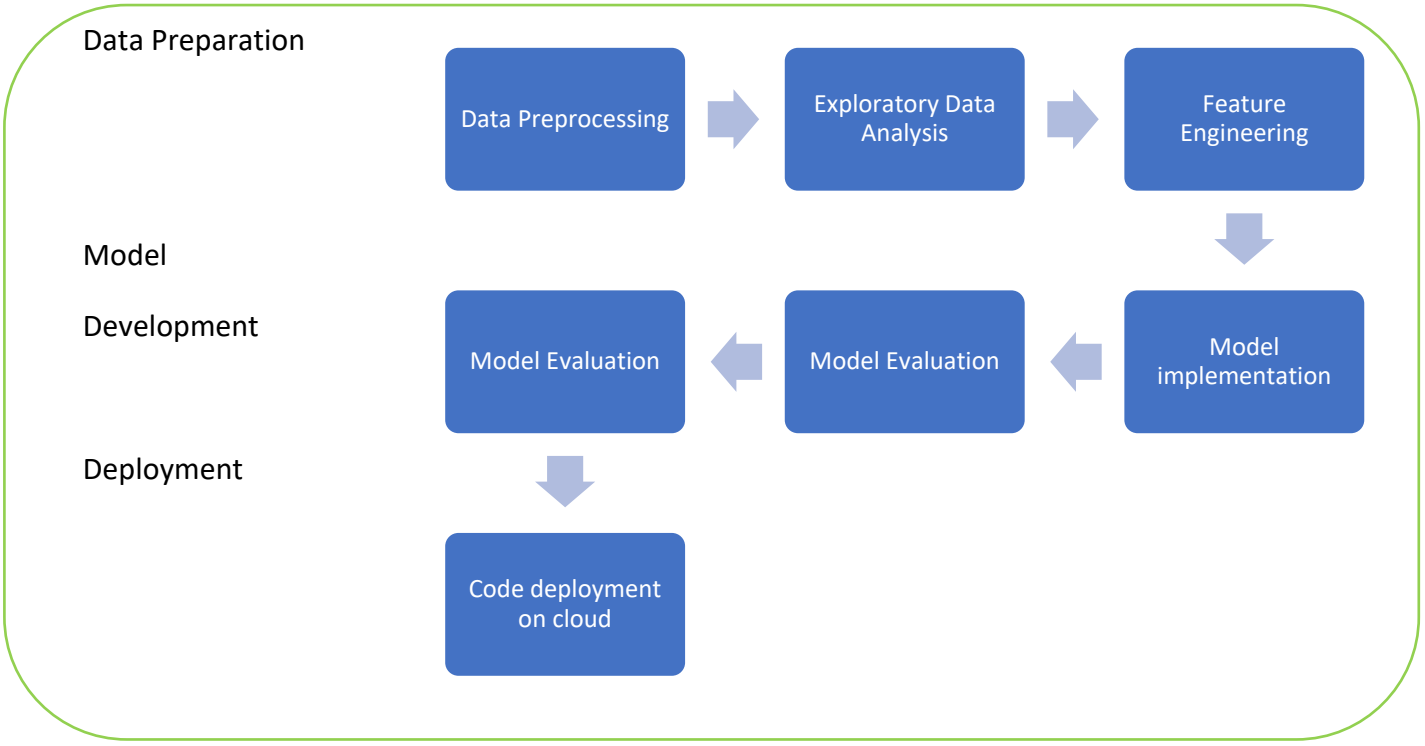
The goal of LLD or a low-level design document is to give the internal logical of the actual program code for Metro Interstate Traffic Volume Prediction. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli.

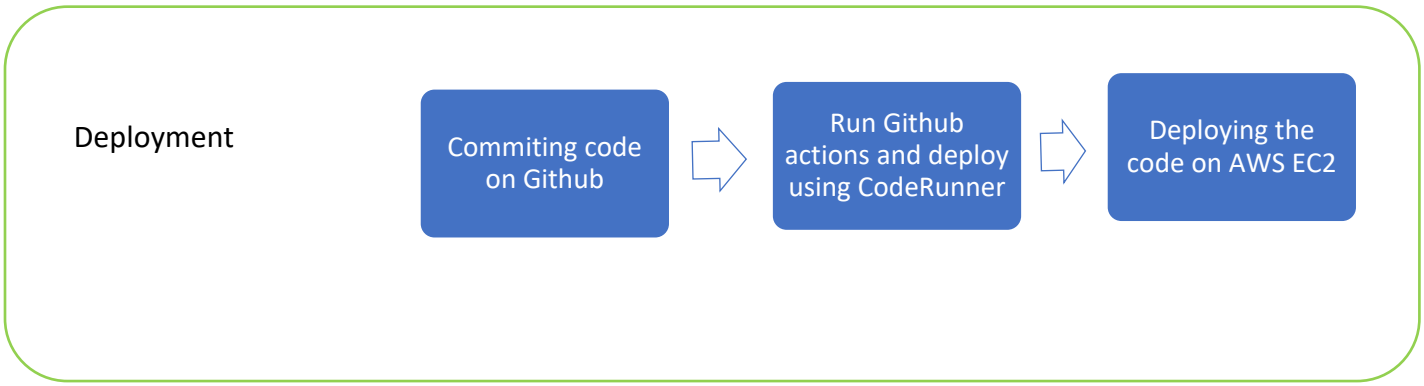
The main objective of the project is to predict if traffic volume is in high or low on particular date. Weather circumstance, special days like holidays, daytime (morning, afternoon, night and etc.), a temperature, a weekday, a numeric percentage of cloud cover are vital attributes for predicting traffic volume.

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 organization may be defined during requirement analysis and then refined during data design work.

ARCHITECTURE





ARCHITECTURE DESCRIPTION

DATA PREPARATION

Data Description

The health condition data from the insurance company. The goal of this project is to build a prediction model using multiple machine learning techniques and to use a template to document the end-to-end stages. We're trying to predict the expenses the client will make on the premium of insurance.

Data Preprocessing

In data preprocessing step, we check if there are missing data, duplicate values, and data types of each feature. In our dataset, there was not any null and duplicate value

MODEL DEVELOPMENT

MODEL IMPLEMENTATION

After train and test splitting, pipeline containing Standard Scaler and One Hot Encoder was fitted to several models such as Linear Regression, SVR, Decision Tree Regressor and Random Forest Regressor. Their R2 score were obtained. And it was determined that Random Forest Regressor performs better than other models.

MODEL EVALUATION

Test dataset is used to evaluate the model. A percentage of dataset was separated for testing. Predicted results of the model are compared with the actual data to check the amount of error.

DEPLOYMENT

DESIGNING FORM WITH HTML

For this project, a form is built on HTML.

DESIGNING A SERVER

A server should be created to run the application continuously. Flask server is built.

CODE DEPLOYMENT ON CLOUD

The codes for this machine learning model should be deployed to the cloud, so that when data is entered into the application, our code runs, and a user gets the result online.

DEPLOYMENT PROCESS

The code was first committed on Github. The pipeline was created between Git and Code Runner. Then the code was deployed to the AWS-EC2. This process is been done using Github Actions. Hence Continuous Integration, Continuous Delivery and Continuous Deployment.