Metro Interstate Traffic Volume Prediction

High Level Design

Domain: Machine Learning

Creator: Jaiprakash

Date: 10.07.2022

Document Version Control

Date issued	Version	Description	Author
July 28, 2022	1.1	First Draft	Muhammad Ojagzada
July 29, 2022	1.2	Added unit test cases	Muhammad Ojagzada
August 1, 2022	1.3	Correcting font sizes	Muhammad Ojagzada

Contents

Abstract

Nowadays, traffic is a major issue for everyone, and it is a source of stress for anyone who has to deal with it on a daily life. The growth of the population delays traffic and makes it worse day by day. The settlement of modern civilization looks at it, but it is unable to act in such a way as to protect people. We can watch traffic, collect data, and anticipate the next and subsequent observations using a variety of approaches and patterns. The observation agency then makes observations, which are then required out and predictions are made. Being stuck in a cosmopolitan city's traffic is the most common occurrence in one's life.

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 forecast the value of a continuous variable with the Metro Interstate Traffic Volume dataset, which is a regression issue.

Introduction

What is High-Level Design Document?

The goal of this HLD or a high-level design document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

- Present all of design aspects and define them in detail
- Describe all user interfaces being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and architecture of the project

- List and describe the non-functional attributes such as security, reliability, maintainability, portability, reusability, application compatibility. resource utilization, serviceability

Scope

The HLD documentation presents the structure of the system, such as database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly technical terms which should be understandable to the administrators of the system.

General Description

Definitions

Term	Description
IMTVP	Interstate Metro Traffic Volume Predictor
Database	Collection of the Information
Cloud	A data center full of services connected to the internet performing service
IDE	Integrated Development Environment
UI	User Interface
Anvil	A Python based UI builder
Heroku	A cloud service

Product Description

IMTVP is a Machine Learning based regression model which helps us to do predictive analysis of the traffic volume using certain features and parameters.

Problem Statement

To create an ML based solution for predictive analysis of a person's annual income and also deploy it in the form of a UI.

The aim to predict the traffic volume on specific weekday of month. This is basically regression problem.

Proposed solution

Using all the standard techniques used in the life cycle of a Data Science project starting from Data Exploration, Data Cleaning, Feature Engineering, Model Selection, Model Building and Model Testing and also building a frontend where a user can fill their information in the form input and get the output instantly.

Further improvements

This IMTVP can be easily embedded inside any website or an application and everybody can get quick answer by inputting required data on friendly user interface.

This can be further improved by training more data in the model. Data can be acquired from MN Department of Transportation

Data requirements

Data requirement completely depend on our problem statement. We need the dataset from MN Department of Transportation to improve accuracy of the model. Required dataset should contain the following features:

- Date time. Weekday, month can be extracted from it.
- We should know do we have holiday on that particular date.
- Weather condition like cloudy, rain, smoke and etc.
- The numeric percentage of cloud cover on specific date time.
- Temperature of weather on specific date time.

These are the required parameters to feed into model.

Tools used

Python programming language and frameworks such as NumPy, Pandas, Scikit-learn, Flask, Anvil, and a few other libraries were used to build the whole model.

- For visualization tasks, matplotlib, seaborn and plotly were used
- For visualization tasks, matplotlib, seaborn and plotly were used
- Anvil and Flask were used for building the web application and server to run the code
- Apache Cassandra was used to storage and retrieval of data
- GitHub is used as version control system
- NumPy and Pandas were used to clean and interpret data
- Scikit-learn was used to cross validate and compare different models
- CatBoost Regressor was used to build the final model

Hardware Requirements

- Windows Server, Linux, or any operating system that can run as a webserver, capable of delivering HTML5 content.
- Minimum 1.10 GHz processor or equivalent.
- Between 1-2 GB of free storage
- Minimum 512 MB of RAM
- 3 GB of hard-disk space

Constraints

The front-end must be user friendly and should not need any one to have any prior knowledge in order to use it.

Assumptions

The main objective of this project is to implement the use case as previously mentioned (2.3 problem statement) for new dataset that comes through the UI. It is assumed that all aspects of this project have the ability to work together as the designer is expecting and also the data on which our model is trained is as correct as possible

Design Details

Process Flow

For accomplishment of the task, we will use a trained Machine Learning model. The process flow diagram is shown below:
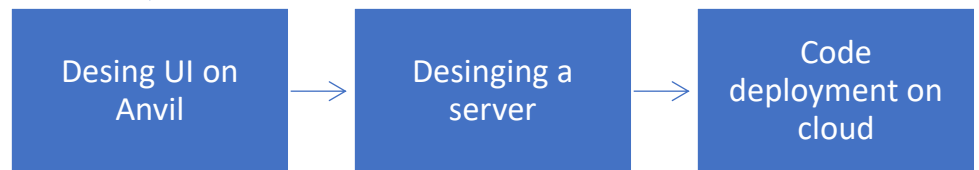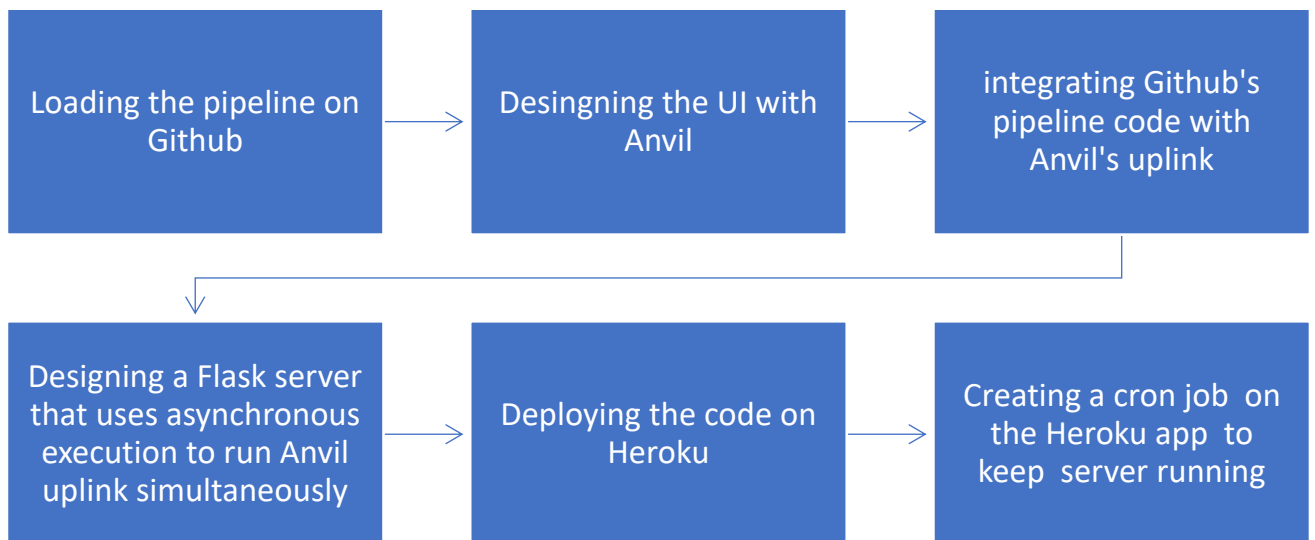
Data Preparation

Data Preprocessing	→	Exploratory Data Analysis	→	Feature Engineering

Model Development

Model implementation	→	Hyper-parameter Tuning	→	Model Evaluation

Deployment

Desing UI on Anvil	→	Desinging a server	→	Code deployment on cloud

Deployment

Loading the pipeline on Github	→	Desingning the UI with Anvil	→	integrating Github's pipeline code with Anvil's uplink

Designing a Flask server that uses asynchronous execution to run Anvil uplink simultaneously	→	Deploying the code on Heroku	→	Creating a cron job on the Heroku app to keep server running

Event Log

The system should log every event so that the user will know what process is running internally. Initial step-by-step description: 1. The system identifies at what level logging is required 2. The system should be able to log each and every system flow 3. Developer can choose logging method. You can choose database logging/ File logging as well 4.

System should not hang even after so many loggings. Logging just because we can easily debug issues, so logging is mandatory to do.

Error Handling

Errors should be encountered, an explanation will be displayed as to what went wrong ? An error will be defined as anything that falls outside the normal intended usage.

Performance

The MITVP tool is used to predict whether traffic condition is bad or good by providing numeric traffic volume. It can be used by various governmental/ non-governmental/ private agencies then it is supposed to be as accurate as possible. So that it doesn't mislead authorities. Also, model retraining is very important to further enhance its performance.

Reusability

The code written and the components used should have the ability to be reused with no problems.

Application Compatibility

The different components for this project will be using Python as an interface between them, each component will have its own task to perform, and it is the job of Python to ensure proper transfer of information.

Resource Utilization

When any task is performed, it will likely use all the processing power available to it until finished.

Dashboards

As and when, the system starts to capture the historic/ periodic data for a user, the dashboards will be included display charts over time with progress on various indicators or factors.

KPIs (Key Performance Indicators)

- Key Performance Indicators of MITVP
- Latency or the amount of time the application takes to display results for some specific input.
- The processing power our application takes to run
- The memory and RAM our application takes to run on a web server.

Conclusion

All in all, overall project architecture, design details, used technologies and performance were explained in detail. The MITVP will give the traffic volume predictions instantly and has the potential to help various government organizations, agencies and etc.