METRO INTERSTATE TRAFFIC PREDICTION

Objective:

Development of a predictive model for traffic volume prediction. The model will predict the traffic model based on conditions such as time, weather and occasions.

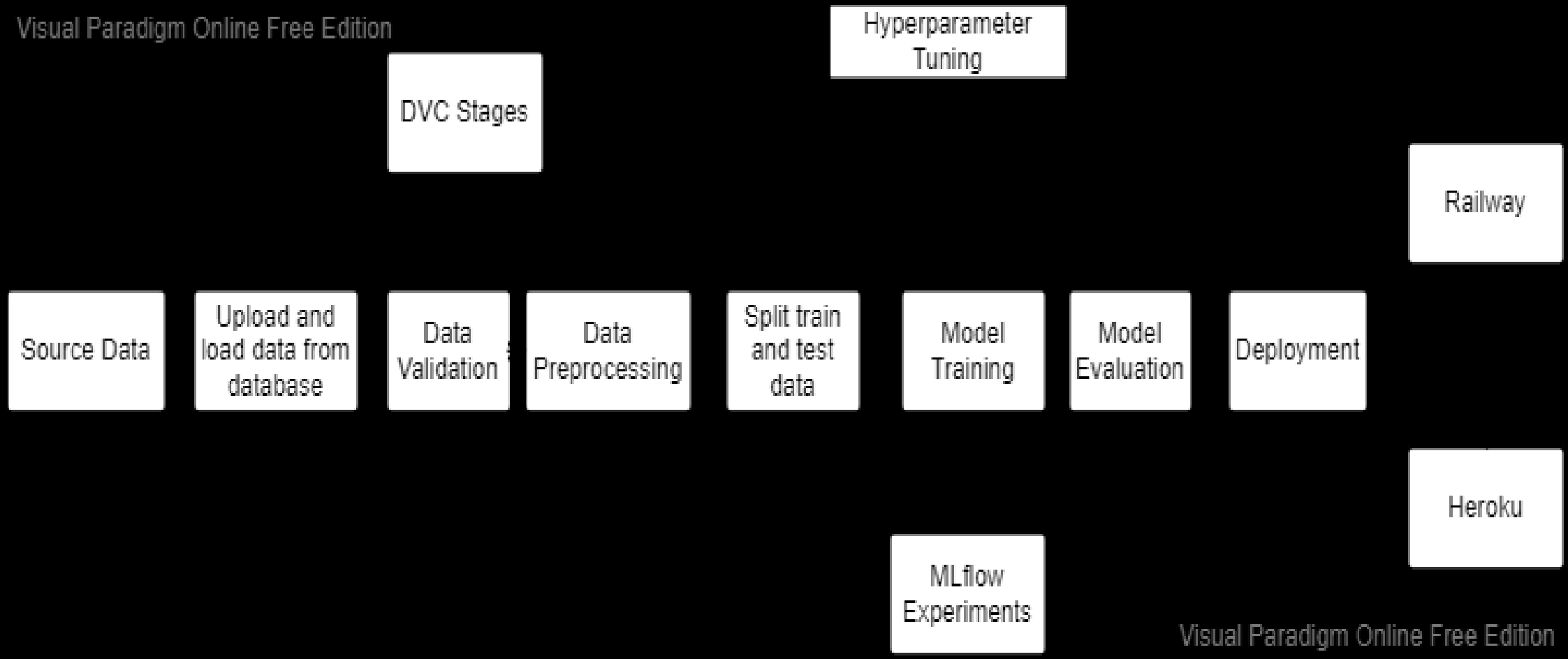
Benefits:

- Detection of behaviour of traffic on different weather conditions.
- Can be used for predicting time when traffic is less.
- Traffic behaviours on holidays.
- Forecasting traffic volume.

Data Sharing Agreement :

- Sample file name (ex fraudDetection_20062021_101010)
- Length of date stamp(8 digits)
- Length of time stamp(6 digits)
- Number of Columns
- Column names
- Column data type

Architecture



Data Description:

The dataset used in this project is a UCI machine learning dataset.

The dataset contains hourly interstate 94 westbound traffic volume for MN DoT ATR station 301. The region of data lies between regions of Minneapolis. It includes features such as holiday, time, weather, etc. which impacts the traffic volume traffic volume directly.

Information of attributes of dataset as follows:

- holiday: Indicates if the date is a holiday and if it specifies the holiday, if not None.
- temp: Indicates the temperature in Kelvin.
- rain_1h: Amount in mm of rain that occurred in the hour.
- snow_1h: Amount in mm of snow that occurred in the hour.
- clouds_all: Percentage of cloud cover.
- weather_main: Short textual description of the current weather.
- weather_description: Longer textual description of the current weather.
- date_time: Hour of the data collected in local CST time.
- traffic_volume: Hourly I-94 ATR 301 reported westbound traffic volume.

Source Data:

- The source data used for this project is collected through UCI Machine Learning Repository - https://archive.ics.uci.edu/ml/datasets/Metro+Interstate+Traffic+Volume

Data Insertion into database:

- MongoDB database is used for this project as the primary source of storing and saving the data. get_data_from_database.py file is responsible for uploading and loading the data.
- Table Creation – ‘traffic_records’ named table or collection is created if not exists in the ‘Metro_Interstate_Traffic_Prediction’ database in mongoDB.
- Data Insertion – The dataset is inserted into traffic records if any data is not found.

Loading Data from database:

- After the source data is uploaded into database, data is loaded from the database and is stored in the raw data path.

Data Validation:

- Validating the data is an important part of the machine learning pipeline.
- Data is validated with respect to no. of columns, no. of rows, column names and its data types.
- If validation is successful, the data is further processed for transformation, otherwise, the stage is failed.

Data Transformation:

- Once data is validated, the data is passed for preprocessing.
- Below transformations are carried out in this stage:
    1. Handeling Null Values
    2. Outliers Detection
    3. Feature Selection
    4. Label and OneHot Encoding
    5. Other feature engineering steps
- After the data is cleaned, the model ready data is transferred to the processed data path.

Splitting the data into training and testing :

- Before model training, the data is separated into training and testing data.
- For our model, the data split is 75% Training data and 25% Testing data..
- train_test_split from sklearn.model_selection is used to split the data.

Model Training:

- As we are solving a regression problem, after testing all the regression models we found that XGBoost Regressor was the best algorithm for our project.
- Through hyperparameter tuning, we tried to achieve best parameters and contribute to model retraining with continuously improved parameters.
- MLFlow experiment tracking is carried out at the same time for model experimentation.

Model Evaluation:

- Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and R2 score are the metric systems used for model evaluation and experiment.
- Detailed report for each experiment metrics can be experimented using MLFlow.
- Metrics and Parameters are also tracked each time the model is deployed using a yaml file.

Hyperparameter Tuning:

- To achieve the most accurate model, it is necessary to tune the model parameters continuously for improvised results.
- We have used RandomSearchCV for fast and continous model improvement.
- XGBoost Regressor parameters are played with different values for experimentation.

DVC:

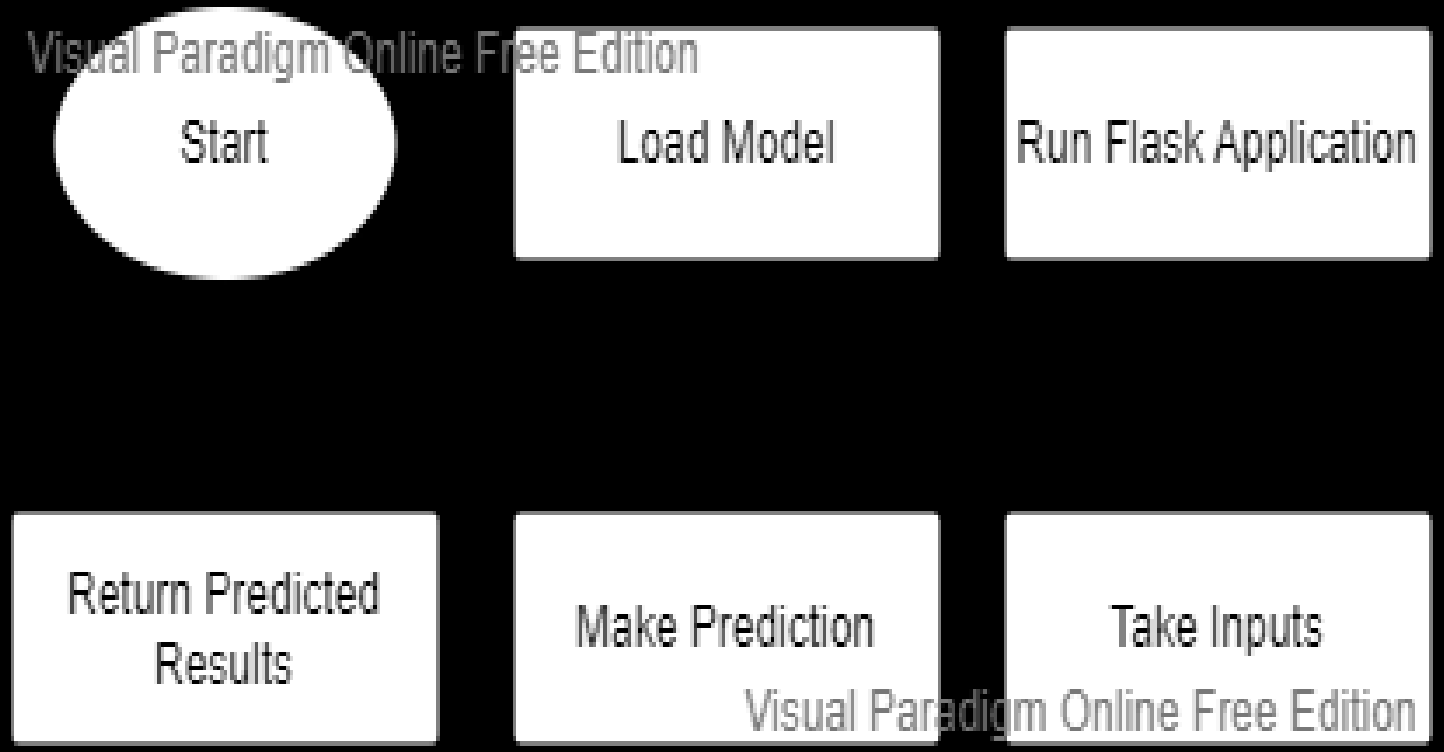
- DVC, which goes by Data Version Control, is essentially an experiment management tool for ML projects.
- DVC represents a complete machine learning pipeline with the help of different DVC stages.
- Stages, dependencies, parameters, metrics and outputs are defined in the dvc.yaml file to execute the DVC.

MLFlow:

- MLflow is a platform to streamline machine learning development, including tracking experiments, packaging code into reproducible runs, and sharing and deploying models.
- Parameters and metrics to log are defined while model training stage.
- Experimentation, model comparision, model version stages, etc. are some of the operation we can perform using MLFlow. Model retraining becomes possible using MLFlow.

Deployment:

- This project is deployed on cloud platforms such as Railway, Heroku and Render.
- Below is the deployment flow chart:



Prediction:

- Our project aims to make prediction via two types of response:
  - Prediction Form
  - API Response
- For Prediction form
  - Enter all the input values in the guided range.
  - Click the prediction button.
  - The prediction is printed in the prediction block.
- For API Response
  - Use the deployed app link as the API for prediction.
  - Provide the input in json format.
  - Get the result in a dictionary format.

Q & A:

Q1) What’s the source of data?

The data is taken from UCI Machine Learning Repository.
https://archive.ics.uci.edu/ml/datasets/Metro+Interstate+Traffic+Volume

Q 2) What was the type of data?

The data was the combination of numerical and Categorical values.

Q 3) What’s the complete flow you followed in this Project?

Refer slide 5th for better Understanding

Q 4) How logs are managed?

We are using custom logs as per the steps that we follow in validation and modeling like File validation log , Data Insertion ,Model Training log , prediction log etc.

Q 5) What techniques were you using for data pre-processing?

- Removing unwanted attributes
- Visualizing relation of independent variables with each other and output variables
- Checking and changing Distribution of continuous values
- Removing outliers
- Cleaning data and imputing if null values are present.
- Converting categorical data into numeric values.
- Scaling the data

Q 6) How training was done or what models were used?

- With the model ready data, we trained the data with all the regression algorithms to find the best model.
- XGBoost Regressor was found the best model.
- Hyperparameter Tuning was performed for finding the best parameters for XGBoost Regression model.
- Parameters and Metrics were logged in the MLFlow for experimentation.

Q 7) How Prediction was done?

The inputs were provided on either the prediction page or through API response. We perform the entire lifecycle till the model is trained and evaluated. In the end we accumulate the data for prediction.

- Q 8) What are the different stages of deployment?
  - Prepare the model to deploy.
  - Validate the ML Model.
  - Deploy the model.
  - Monitor the model.