Architecture

Scania Truck Failure Prediction System

|  |  |
| --- | --- |
| Written By | Sethu Sai Medamallela |
| Document Version | 1.0 |
| Last Revised Date | 24/01/2022 |

## Document Version Control

Change Record:

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Comments |
| 1.0 | 29-12-2021 | Sethu Sai Medamallela | Architecture |
| 1.1 | 24-02-2022 | Sethu Sai Medamallela | Architecture |
|  |  |  |  |

# Contents

Document Version control

Architecture

Architecture Description

Data Description

Export Data from DB to CSV for training

Data Splitting

Data Preprocessing

Model Training

Model Evaluation

Model saving

Cloud setup

## Architecture

Architecture Description

Data Description-

We will be using scania truck dataset from UCI Machine Learning Repository. This data set consists of 6000 rows and 171 columns in a single csv file. For our convenience, we have split the data into multiple csv files of 3000 rows for batch processing.

Data Storage –

After the data is split into multiple csv files, we have uploaded the data to S3 buckets. from there we shall be picking the data for validation and preprocessing and model training.

Data Validation –

The data stored in S3 buckets are validated based on the filename, columns names, missing values in column. Once the validation is done the data is transferred to good data bucket and other data to bad data bucket.

Data Transformation –

Once the data validation is done, and good data is stored in S3 bucket, we are adding quotes to string data in columns as data transformation. Data transformation takes place in within S3 buckets.

Database Operation-

Now that have we have processed and validated the data, and generated good data which is stored S3 bucket, we need to store them in database which is MongoDB. MongoDB Atlas with M20 cluster is used as configuration. Once the good data is stored in MongoDB we need to export it as csv file for training which will be stored in S3 buckets

## Data Preprocessing

We will be exploring our data set here and do EDA if required and perform data preprocessing depending on the data set. We first explore our data set in Jupyter Notebook and decide what pre-processing and Validation we have to do such as imputation of null values, etc. and then we have to write separate modules according to our analysis, so that we can implement that for training as well as prediction data.

Model Training –

Since we used a customized machine learning approach, we have generate cluster of the data and then train models which in our case is Random Forest and XGBoost with GridSearchCV as hyperparameter tuning technique and evaluated on the basis of accuracy score and AUC score.

Once the models are trained and hyper parameters tuned with model scores.. The trained models are saved to S3 buckets in trained model folders. We are using MLFlow for tracking the experiments models, metrics and parameters. Once it is done, load production model pipeline is triggered.

Load Production Model –

Once the models are trained and logged in MLFlow, now we need to push the best models for the particular cluster to production and staging or staging and archived based on condition.

## Cloud Setup

Here We will do cloud setup for model deployment. Here we also create our flask app and user interface and integrate our model with flask app and UI

## 

## Push app to cloud

After doing cloud setup and checking app locally, we will push our app to cloud to start the application.

Data from client side for prediction purpose

Now our application on cloud is ready for doing prediction. The prediction data which we receive from client side.

Data processing and Prediction

Client data will also go along the same process Data pre-processing and according to that we will predict those data.

Export Prediction to CSV

Finally when we get all the prediction for client data, then our final task is to export prediction to csv file and hand over it to client.