HIGH LEVEL DESIGN DOCUMENT FOR SCANIA TRUCK FAILURE PREDICTION

DOCUMENT VERSION CONTROL

|  |  |  |  |
| --- | --- | --- | --- |
| Date Issue | Version | Description | Author |
| 29/12/2021 | 1 | Initial HLD – V 1.0 | Sethu Sai Medamallela |
| 24/01/2022 | 2 | Final HLD - V 2.0 | Sethu Sai Medamallela |

CONTENT

* DOCUMENT VERSION
* ABSTRACT
* INRODUCTION
* WHY THIS HIGH LEVEL DESIGN DOCUMENT
* SCOPE
* DEFINATION
* GENERAL DISCRIPTION
* PRODUCT PERSPECTIVE
* PROBLEM STATEMENT
* PROPOSED SOLUTION
* FURTHER IMPROVEMENTS
* TECHNICAL REQUIREMENTS
* TOOLS USED
* DATA REQUIREMENTS
* CONSTRAINTS
* IMPROVEMENTS
* DESIGN DETAILS
* PROCESS FLOW
* PROPOSED METHODOLOGY
* MODEL TRAINING AND EVALUATION
* MODEL DEPLOYMENT APPROARCH
* EVENT LOG
* ERROR HANDLING
* PERFORMANCE
* REUSEABLITLITY
* APPLICATION COMPATIBILTY
* RESOURCE UTILIZATION
* DEPLOYMENT
* KEY PERFORMANCE INDICATOR
* CONCLUSION

ABSTRACT

In the world of automobiles, one of the important factors that play a key role in automobile efficiency is taking good care and proper maintenance of the automobile components , one such component is in a Truck Automobile is APS (Air Pressure System). The APS generate air pressurized air that is used in various critical functions such as braking and gear changing. The operational cost of the components can be significantly reduced by accurate prediction of the failure based on the measurements of truck mechanical system attributes.

INTRODUCTION

WHY THIS HIGH LEVEL DESIGN DOCUMENT?

The purpose of this High Level Design (HLD) 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 THE DESGIN ASEPCTS AND DEFINE THEM IN DETAIL
* DESCRIBE THE USER INTERFACE BIENG IMPLEMENTED
* DESCRIBE THE HARDWARE AND SOFTWARE INTERFACE
* DESCRIBE THE PERFORMANCE REQUIREMENT
* INCLUDE DEFINE FEATURE AND ARCHITECHTURE OF THE PROJECT
* LIST AND DESCRIBE THE NON FUNCTIONAL ATTRIBUTES
* SECURITY
* RELIABLILTY
* MAINTAINABILITY
* PORTIBILITY
* REUSEABILITY
* APPLICATION COMPATIBILTY
* RESOURCE UTILIZATION
* SERVICEABILITY

SCOPE

The HLD documentation presents the structure of the system, such as the 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.

TECHNICAL TERMS

|  |  |
| --- | --- |
| TERM | DESCRIPTION |
| DATABASE | Collection of Information Monitored by the System |
| IDE | Integrated Development Environment |
| AWS | Amazon Web Services |

GENERAL DESCRIPTION

PROBLEM STATEMENT

The Air Pressure System (APS) is a critical component of a heavy-duty vehicle that uses compressed air to force a piston to provide pressure to the brake pads, slowing the vehicle down. The benefits of using an APS instead of a hydraulic system are the easy availability and long-term sustainability of natural air.

This is a Binary Classification problem, in which the affirmative class indicates that the failure was caused by a certain component of the APS, while PROPOSED SOLUTION

The solution to the above stated problem is creating a UI application. The UI will enable the user to check that the intrusion is genuine or not.

TECHINICAL REQUIREMENTS

Hardware tools like sensors are required for collection of important data from the components. Essentially a Big Data Pipeline, for the extraction of data from the sensors and then putting them into csv file.

1. S3 buckets for cloud storage for the collected data.
2. EC2 instances for running applications
3. EC2 instances for tracking and monitoring experiments
4. ECS cloud deployment.

TOOLS USED –

1. AWS EC2 instance for MLFlow Server
2. MLFlow for Experiment tracking and monitoring metrics, parameters and models
3. Fast-API as Web Server Application
4. MongoDB for CRUD operations
5. Linux Service Management for MLFlow as a service
6. Nginx for MLFlow configuration.
7. GitHub for version control
8. Docker for containerization
9. AWS ECS and ECR for Container Deployment and Management

DATA REQUIREMENTS -

For the sake of the POC we are using the existing data from UCI Machine Learning Repository for which contains encrypted data columns for the original truck data.

CONSTRAINTS

The truck failure based solution system must be completely automated as possible and user should not know any of the internal working, and creating a user friendly interactive frontend for simplicity.

ASSUMPTIONS

The assumption is that there will be Big Data Pipeline, which will created either from the client side or from our side for the data extraction from the sensors and then put them in multiple batches of csv files which will be stored in S3 buckets,

DESIGN DETAIL

PROCESS FLOW

For predicting the failure of engine component., we will machine learning based models.

The process will look like 

PROPOSED METHODOLOGY

For solving the problem statement, we are using a customized machine learning approach, in which we are using KMeans model, to divide the data in to smaller clusters and then for the cluster of data, a machine learning model is created, and the best model is selected for prediction.

MODEL TRAINING AND EVALUATION

As mentioned in the proposed methodology, we are using a customized machine learning approach using KMeans as clustering model. For every cluster, two models are trained which Random Forest and XGBoost models. These trained models are evaluated on the basis of AUC score and Accuracy score

DEPLOYMENT PROCESS

The deployments of models are done on the basis of best model score for the particular cluster data, scoring criteria being accuracy and AUC score. After the training, the models are stored in trained models folder in S3 bucket, now depending upon the best score for the cluster MLFlow will automatically transition the models in MLFlow UI and in S3 buckets also. As for the application, it is first containerized used Dockers and then using GitHub actions, we are deploying them to AWS ECS cluster for end user serving.

EVENT LOG

The system is going to log everything so that the user gets to know which process is running internally.. Logs are stored in MongoDB

ERROR HANDLING

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

PERFORMANCE

The truck failure system solution ise used for predicting the failure of engine components. Whenever there is failure prediction, it will inform concerned parties and authorities. Also model retraining is important to improve the performance.

REUSABLILITY

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

APPLICATION COMPATIBILTY

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 the Python to ensure proper transfer of information.

RESOUCE UTILIZATION

When any task is performed, it will likely use all the process power available until that function is finished.

KEY PERFORMANCE INDICATOR

The only indicator in our application will be the score of the trained model and system performance. This means that whether the application is able to predict correct output for a given input.

CONCLUSION

The application will be providing the banks an interactive platform where they just have to give some required inputs, and based to previous data, model will be able to produce a prediction for credit risk.

This application will be providing Trucker drivers and Automobile manufactures an interactive platform for preventing further accidents to both truck drivers and also predictive maintenance of the components before the damage is too worse or expensive to replace or so.