|  |  |
| --- | --- |
|  | **AVM Engineering**  **App Lens Hackathon**  **Theme 12- Failure Prediction Model Design Document** |
| |  |  |  | | --- | --- | --- | |  | **Participant 1** | **Participant 2** | | **Name** | Vinoth S(266251) | Jay Suriya R(672852) | | **Role** | A | PA | | **Signature** | VS | JSR | | **Date** | 07/16/2019 | 07/16/2019 | |
|  |

|  |  |
| --- | --- |
| ***CheckList*** | ***Tick*** |
| Design Approach:(Security/Quality(Exceptions Handled)/Performance) | Yes |
| Scenarios Covered(If Applicable) | Yes |
| Flow Diagram | Yes |
| Implementation Plan | Yes |
| Testing Plan | Yes |
| Challenges / Dependencies / Assumptions if any | Yes |

|  |
| --- |
|  |

Theme - 12 Create a prediction model utilizing APM data, logged data or any other monitoring data correlating outages/failures happened in the past. Algorithm should be portable and Integra table.

**Approach to Machine learning Predictive model for Failure Prediction Scenario:**

1. **Design Approach:**

**Why Machine Learning Model for Predictive Model?**

Machine learning and cloud storage have created a tremendous opportunity to utilize the gamut of data coming from various mode of application and tools, and more to not only monitor equipment health but also predict when something is likely to malfunction or fail.

Data should have hundreds or even thousands of failures. But even in these cases the distribution or the ratio of failure to non-failure data is highly skewed. Additionally, the data should be collected from all of the relevant parts and should capture the complete picture or timeline of events prior to the occurrence of the failure

1. **Scenarios to Covered**

**Business Case:**

Businesses require critical equipment to be running at peak efficiency and Memory utilization to realize connectivity issue and DB Failure

**Business Problem:**

* Detect anomalies in equipment or system performance or functionality.
* Predict whether an asset may fail in the near future.
* Estimate the remaining useful life of an asset.
* Identify the main causes of failure of an asset.
* Identify what maintenance actions need to be done, by when, on an asset.

**Goal:**

* Reduce operational risk of mission critical equipment.
* Increase rate of return on assets by predicting failures before they occur.
* Control cost of maintenance by enabling just-in-time maintenance operations.
* Lower customer attrition, improve brand image, and lost sales.
* Lower inventory costs by reducing inventory levels by predicting the reorder point.
* Discover patterns connected to various maintenance problems.

1. **Flow Diagram**

1. **Implementation Plan:**

**Tools:**

Language - Python

IDE - Jupyter Notebook

ML - Classification Model

Metrics - Accuracy and Precision

Visualization - Tableau

**Data Preprocessing:**

The success of any learning depends on (a) the quality of what is being taught, and (b) the ability of the learner. Predictive models learn patterns from historical data, and predict future outcomes with certain probability based on these observed patterns. A model's predictive accuracy depends on the relevancy, sufficiency, and quality of the training and test data. The new data that is 'scored' using this model should have the same features and schema as the training/test data

* Relevant Data
* Sufficient Data
* Quality Data

1. **Testing Plan**

**Metrics to measure the Model:**

* [**Accuracy**](https://en.wikipedia.org/wiki/Accuracy_and_precision) is the most popular metric used for describing a classifier’s performance. But accuracy is sensitive to data distributions, and is an ineffective measure for scenarios with imbalanced data sets. Other metrics are used instead. Tools like [**confusion matrix**](https://en.wikipedia.org/wiki/Confusion_matrix) are used to compute and reason about accuracy of the model.
* [**Precision**](https://en.wikipedia.org/wiki/Precision_and_recall) of PdM models relate to the rate of false alarms. Lower precision of the model generally corresponds to a higher rate of false alarms.
* [**Recall**](https://en.wikipedia.org/wiki/Precision_and_recall)rate denotes how many of the failures in the test set were correctly identified by the model. Higher recall rates mean the model is successful in identifying the true failures.
* [**F1 score**](https://en.wikipedia.org/wiki/F1_score)is the harmonic average of precision and recall, with its value ranging between 0 (worst) to 1 (best).

1. **Challenges/Dependencies :**

* **Dependencies** 
  + To develop the better Machine learning model will require more historical data as connectivity, DB Failure and connectivity issues, we should collect data not just from incident logs, but also from the history of DB usage the complete record of daily logs when minor or major issues happened, and when last time DB size growth got increased complete history over a long period of time. A model that learns from rich data like this will be able to find patterns and might identify dependencies that would otherwise not be so obvious and correctly predict in advance when a failure will occur.
* **Challenges:**
  + **Modeling Imbalanced Data**

Modelling for Predictive Maintenance falls under the classic problem of modelling with imbalanced data when only a fraction of the data constitutes failure. This kind of data poses several issues. While normal operations data (i.e. non-failure data) which constitutes the majority of the data is similar to one another, failure data may be different from one another. Standard methods for feature selection and feature construction do not work so well for imbalanced data

**Dealing imbalance data set:**

1) As mentioned above, with imbalanced data, the classification algorithms performance is biased against the minority class. Hence the first step is to balance the dataset through resampling. There are various sampling techniques available, each with their own advantages and disadvantages. The module [**SMOTE**](https://msdn.microsoft.com/en-us/library/azure/dn913076.aspx) allows to **upsample** or increase the number of minority (failure) instances by synthesizing new examples. The module [Partition and Sample](https://msdn.microsoft.com/library/azure/a8726e34-1b3e-4515-b59a-3e4a475654b8) allows us to do simple random sampling or stratified random sampling and can be used for down sampling the majority (non-failure) class. The [Split Data](https://msdn.microsoft.com/library/azure/70530644-c97a-4ab6-85f7-88bf30a8be5f) module can also be used to down sample the majority class.

2) For most machine learning algorithms, we need to provide some **hyperparameters** (e.g. for Boosted Decision Tree we will need to provide values for Maximum number of leaves per tree, Minimum number of samples per leaf node, Number of trees constructed etc.). This determines the efficiency of the model. Additionally, we also need to specify the metric (e.g. Accuracy, Recall, AUC etc.) to use for determining the best set of parameters. The [Sweep Parameters](https://msdn.microsoft.com/library/azure/038d91b6-c2f2-42a1-9215-1f2c20ed1b40/) module in Azure ML allows us to do just that. By selecting Recall or Precision as the metric to optimize the parameter set, the resulting model can be tuned for high recall or high precision performance.

3) The classification models in ML, besides predicting a positive or negative, also output a score which is a real number between 0 and 1. Scoring values above 0.5 (the default threshold) are labelled positive and below negative. The choice of this threshold decides the predicted label and thus the related metrics. For example, choosing a threshold or the operating point of 0.7 means all instances with scores greater than 0.7 will be labelled as positive and below negative. By adjusting this threshold, we can tweak the predictions to produce a high recall or high precision