ReneWind - Project

Machine Learning solution for predicting machine failures

Contents

Business Problem Overview and Solution Approach

- ## Context of Business Problem

  Renewable energy sources play an increasingly important role in the global energy mix, as the effort to reduce the environmental impact of energy production increases. The U.S Department of Energy has put together a guide to achieving operational efficiency using predictive maintenance practices. Failure patterns are predictable and if component failure can be predicted accurately and the component is replaced before it fails, the costs of operation and maintenance will be much lower.
  "ReneWind" is a company working on improving the machinery/processes involved in the production of wind energy using machine learning.

- ## Problem to tackle

  For "ReneWind" one of the machinery heavily used in their wind energy business is generator. And this has maintenance cost which includes Repair Cost, Replacement cost, Inspection cost. Replacement cost is ~3x expensive than Repair cost. Company wants to build a Machine Learning solution which can predict the likely failure so that they can lower the overall maintenance cost, by repairing and not have to replace which incurs more cost.

- ## Financial implications

  Not able to predict a failure automatically –
  1. Takes effort to Inspect manually, which will incur additional inspection cost.
  2. Timing of manual inspection can lead to machinery failure and incur additional cost for replacement.
  3. Both of this is considered additional expense, so identifying a likely failure systematically will minimize the maintenance cost

- ## How to use ML model to solve the problem

  With a good robust predictive model where the model can predict likely failures in advance based on the sensor data, it can help business to repair the faulty machinery before it breaks down and need replacement. Also prediction through ML solutions will lower their manual Inspection cost. This will help them lower their overall maintenance cost.

Data Overview

- Data Dictionary

Variables	Description
Data presented as a ciphered version of actual sensor data. – Variables V1 – V40	Represents 40 different Sensor data with no indication which variable represents which sensor
Target	Represents failure (1) and No failure (0)

Type	Observations	Variables
Train	40000	41
Test	10000	41

Duplicate value counts Train & Test
0

Notes:

1. Dataset looked consistent with the data idea provided in data dictionary.
2. There are no duplicate values
3. There are missing values across 2 variables V1 and V2 in both train and test dataset provided.

Missing value counts Train and Test
Train – V1 has **46** and V2 has **39** Test – V1 has **11** and V2 has **7**

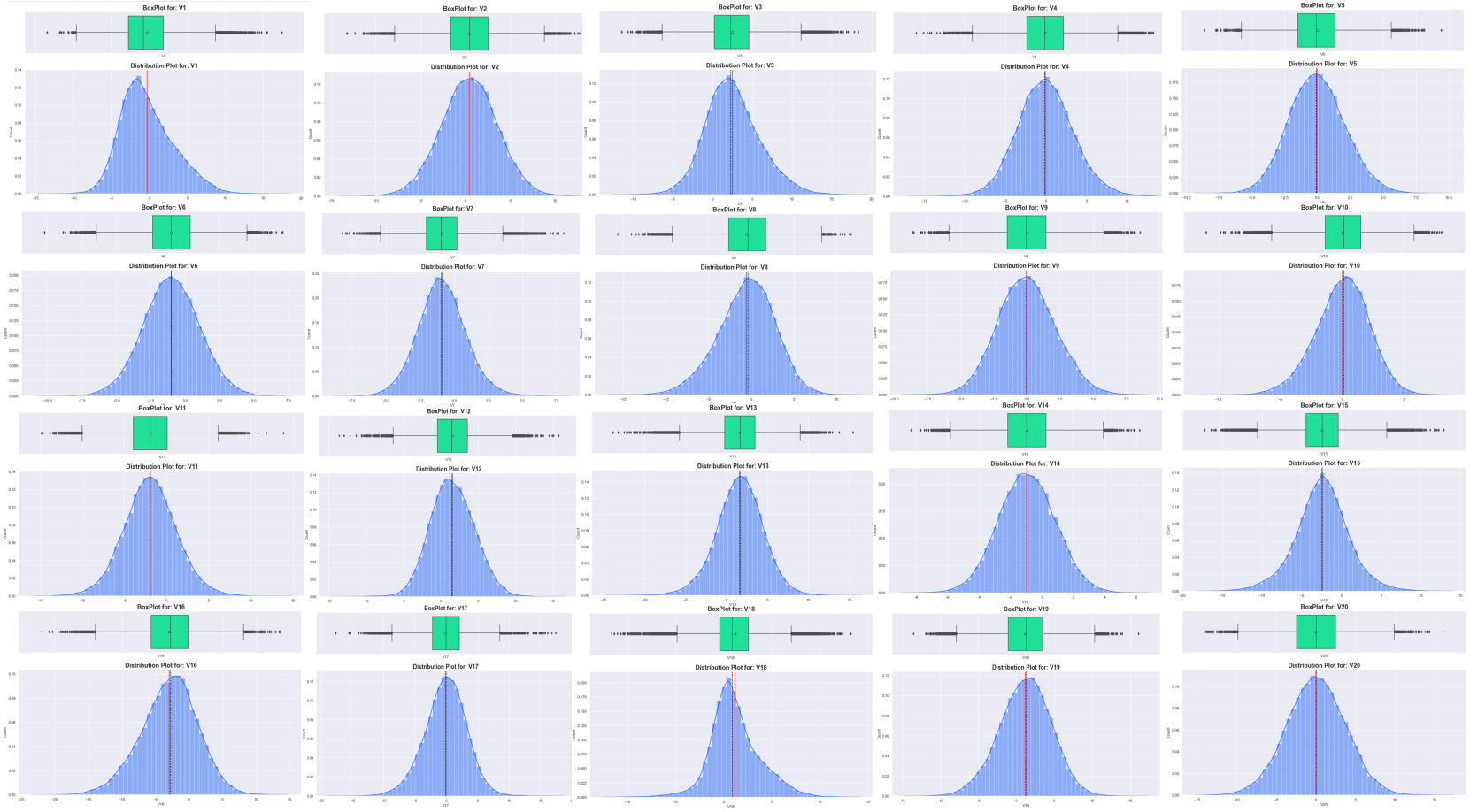
Data Overview contd..

- Brief description of significant manipulations made to raw data

1. Missing values are imputed with simple imputer using KNN and strategy as mean.
2. Outliers' treatment was not done as the models we have built are not sensitive to outliers.
3. As we are dealing with imbalanced class in both dataset, we have used oversampling and under sampling to explore the performance as compared to provided data.
4. No new features were extracted as we do not have enough background on the data variables provided.
5. All variables were used for model preparation, even though we have seen some high correlation among variables, since we do not have much background on the data and models, we are targeting is not affected by it.

Exploratory Data Analysis – Univariate Analysis

- Distribution of dependent variables

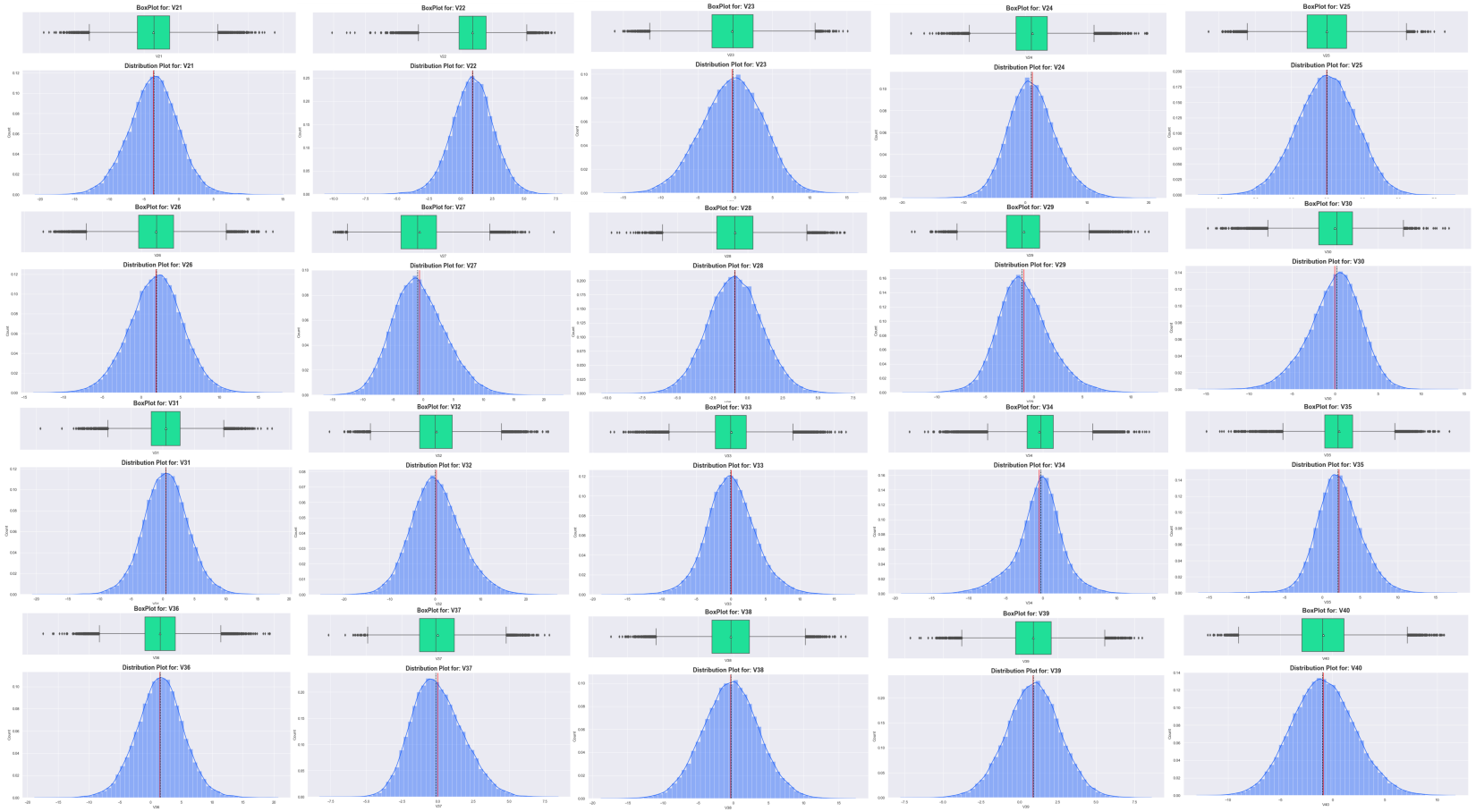


Notes:

1. Majority variables are normally distributed or closer to normally distributed
2. All variables shows outliers on both sides
3. Data for each variables are ranged from –ve to +ve values, where –ve value is as low as -24 and +ve value is as high as +25
4. Following slide will have the distribution for rest of the variables

Exploratory Data Analysis – Univariate Analysis

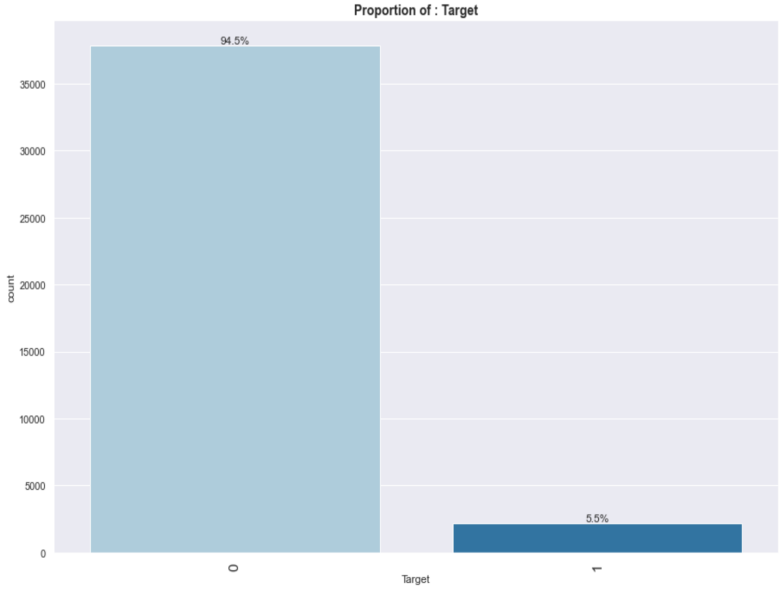
- Distribution of dependent variables



Notes:

1. As noted earlier this slide shows the distribution for other variables and all the observations noted earlier holds good.

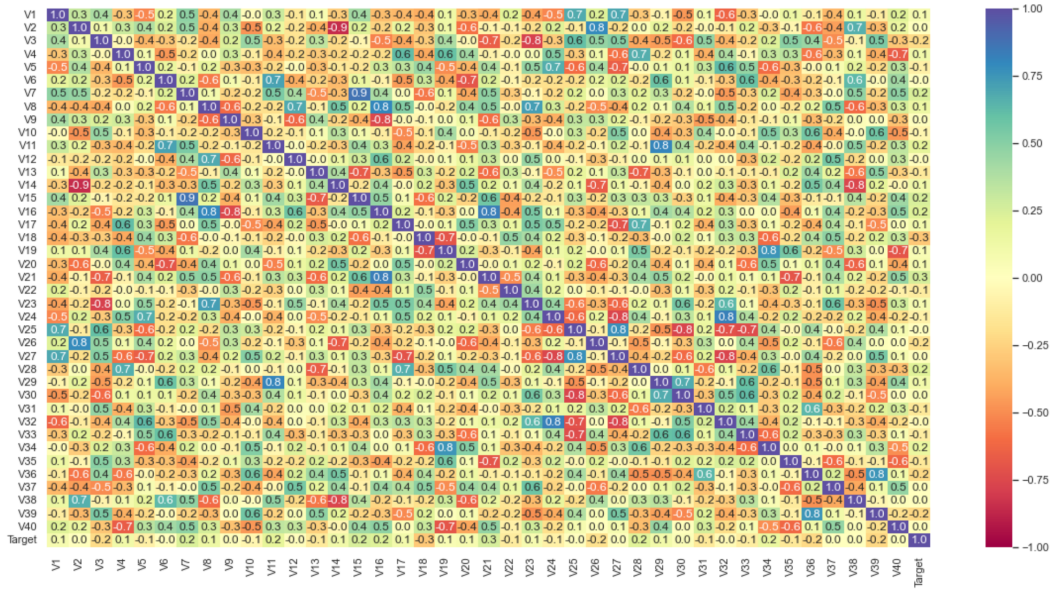
Exploratory Data Analysis – Univariate and Bivariate Analysis



Notes:

1. ~94.5% observations are recorded as not failure
2. ~5.5% are recorded as failure.
3. This confirms we are dealing with imbalanced class for this problem

Heatmap of correlation



Notes:

1. We can notice there are lot of variables which has high correlation with each other.
2. V1 shows high positive correlation with V25, V27 and high negative correlation with V32.
3. V2 shows high negative correlation with V14, V20, V36 and high positive correlation with V26, V38
4. V11 shows high positive correlation with V6 and V29
5. Even though we see there are variables which are highly correlated we could drop them, but we have decided to kept them because of lack of clarity on variables, and models we are targeting wouldn't be affected if we keep them.

Model Performance Summary

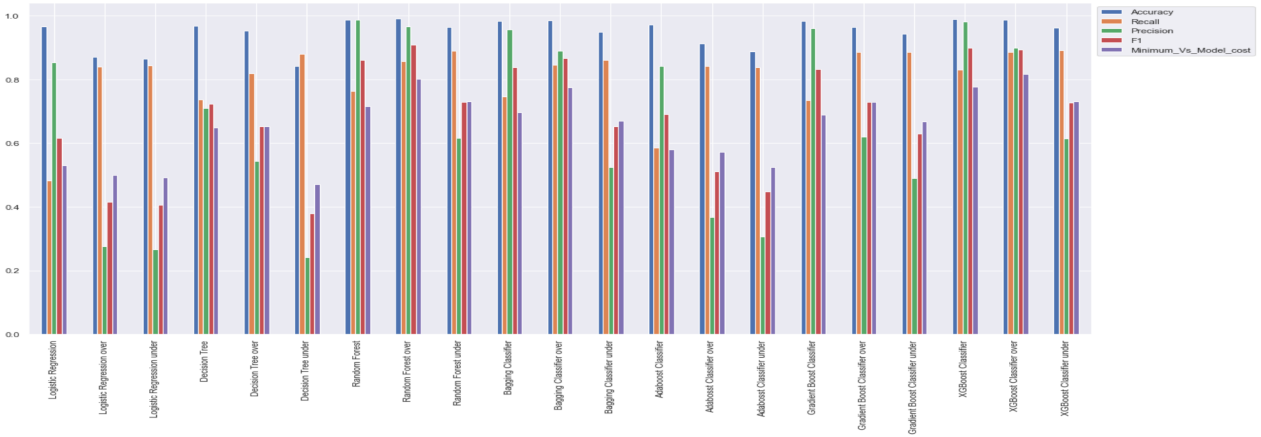
Approach

- Explored Logistic Regression, Decision tree, Random Forest, Bagging, Adaboost, Gradient boosting, XGBoost for creating models for the solution.
- We have been given train and test data separately, test data was kept aside for final model evaluation to avoid data leakage
- 20% of randomly chosen data are kept for validating initial model performances, with 32000 records for training and 8000 records for validation.
- Since we are dealing with imbalanced class, we have explored all models with regular sample (32000) records where each class represent the proportion of original dataset, then by oversampling where additional observations were created to make the proportion of both classes to match, and by under sampling where we have considered less samples where proportion of both classes match.
- Model performance was checked against Recall, and Cost metric which was given to us. For this problem we are trying to bring the model maintenance cost (TP * 15 + FP * 5 + FN * 40) closer to minimum possible maintenance cost (TP * 15 + FN * 40).
- After initial model preparation based on the performance on validation set and K fold cross validation score which is measured on cost metric mentioned above, best 3 models were chosen and further tuned using GridSearch and RandomizedSearch CV.
- Final 3 tuned models were evaluated based on their performance on validation set, and one final model was chosen.
- Final model was evaluated with the Test data based on given success criteria where cost metric (min model maintenance cost/model maintenance cost) > .78
- Final model was prepared using Pipeline for deploying in production.

Model and Parameters

- We have used all 40 dependent variables to build each model for all three scenarios.
- Grid Search and Random search for tuning final 3 models resulted in same best hyper parameters.

Performance of the Models on Validation data

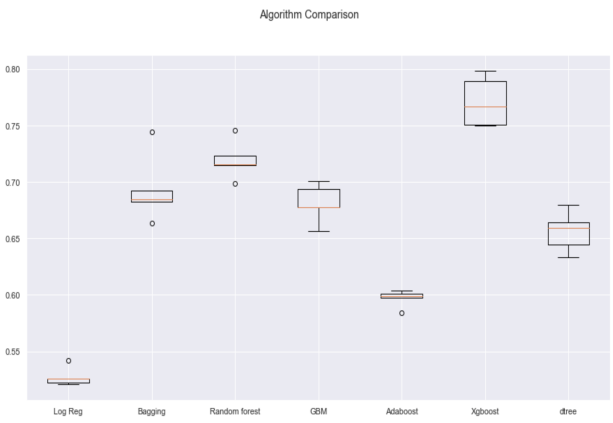


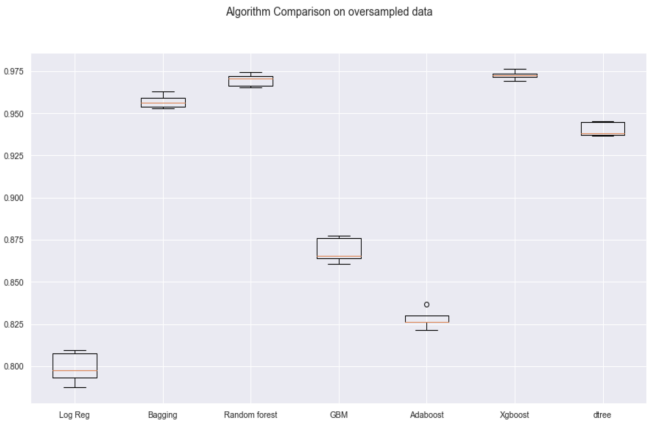
Notes:

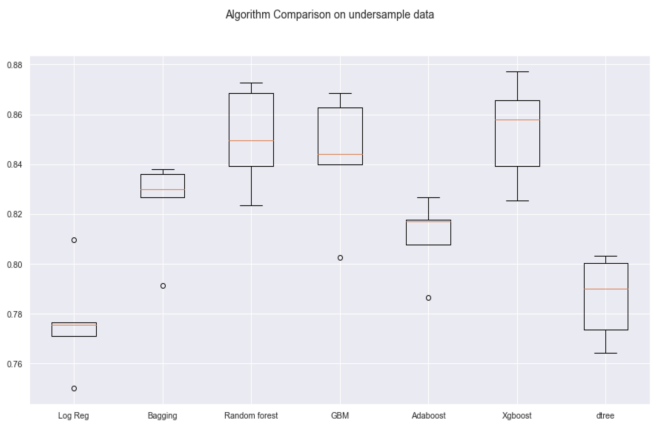
1. Almost all models are overfitted.
2. As we are trying to maximize the cost metric, we will choose 3 models which has validation set performance where all metrics not drastically different from each other, and the cost metrics is very close to .78 or higher
3. With these considerations, we chose below 3 models, however we also did cross validation with cost metric as scorer and take cross validation results also into consideration for finalizing list of 3 models.
   1. Random Forest Classifier on oversampled dataset
   2. Bagging Classifier on oversampled dataset
   3. XGBoost Classifier on oversampled dataset

Model Performance Summary

Cross Validation score summary





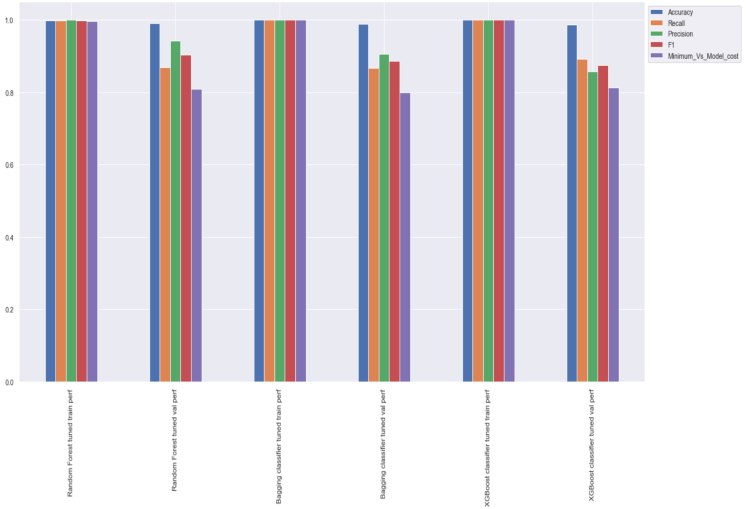


Notes:

1. XGBoost model shows higher cost metric for all 3 approach.
2. In oversample data Bagging, Random Forest and XGBoost shows better performance with respect to cost metric.
3. In under sample data data as well Bagging, Random Forest and XGBoost are the 3 best performing models.
4. As Bagging, Random Forest and XGBoost gives us cost metric > 90% in oversample data as compared to under sample data, we are choosing these models for further tune and generalized if possible.

Model Performance Summary – Tuned model performance

Tuned models performance summary

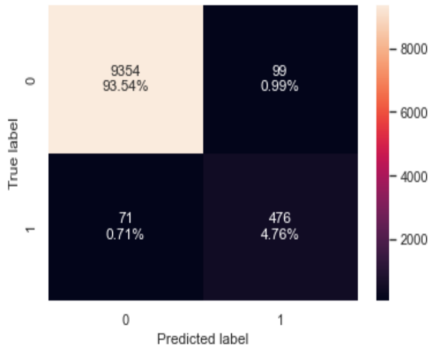


Notes:

1. All tuned models are slightly overfitted on the training data.
2. On validation data all models performed almost similar with XGBoost slightly higher than the Bagging and Random Forest.
3. On validation set XGBoost is giving 81.4% as cost metric and this is > .78, hence we are choosing this as our final model.

Tuned XGBoost performance on test data

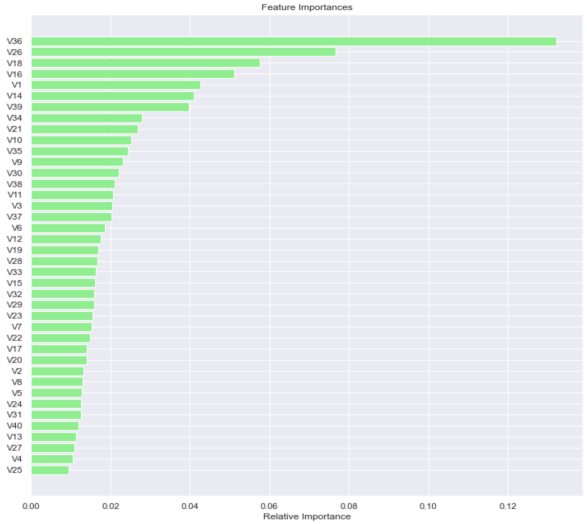
	Accuracy	Recall	Precision	F1	Minimum_Vs_Model_cost
0	0.983	0.870	0.828	0.848	0.783



Notes:

1. Model performed decent on test/unseen data.
2. Only .71% of failure data are not correctly identified, and ~1% of non failure data are not correctly identified
3. Which gives us the cost metric as 78.3% higher than success criteria of 78%, hence we concluded and build pipeline for production deployment of this model.
4. Pipeline consider imputing missing values, and oversample data to represent both class as equal.

Important features for the final model



Notes:

1. All variables shows it has some importance for the model.
2. V36 shows that it's the most important feature for the model, followed by V26, V18, V16, V1 are rest for among the top 5 features.

Business Insights and Recommendations

Insights

- V36 is one of the most important sensor data which has high impact on the failure status of machine.
- Sensor data related to V36 should be closely monitored, if sensor data is <-10 or >+12 risk of failure will be minimal to none, anything in between should be closely looked at
- Next important sensor data is V26 and should be closely monitored, when data is <-12 or >+15 risk of failure will be minimal to none, anything in between should be closely looked at.
- V18 and V16 are the next two important sensor data which should be closely monitored when V18 sensor data is <-8 it is more likely to show sign of failure, and V16 sensor data is >-14 shows sign of failure.
- V1, V14, V39 are next set of 3 sensors for which sensor data should be taken as priority and acted upon, specifically when V1 has data in between -10 to +13, when V14 data is <-7.5, and when V39 data is <+6
- V34, V21, V10 are the last 3 on top 10 important sensor data, and it should be taken with priority when V34 data is in between -12 to +12, V21 data >+10
- The final model prepared are correctly predicting failures ~88% of total failures, in other words approximately 88 failures out of 100 will be correctly predicted by this model.

Recommendations

- Business should plan on using this model to predict failures and look at the above important sensor data and take necessary actions to avoid breakdown
- Using this model ~78.3% of the time maintenance cost will be equal to minimum maintenance cost, and only ~21.7% of the time additional spend in terms of Inspection cost. This is within the considerable limit defined by business.
- Business should think about setting up auto alert based on prediction and sensor data value and, kick off maintenance based on identified sensors and their likely failure range/threshold to avoid breakdown.

greatlearning
Power Ahead

Happy Learning !