ReneWind
Business Presentation

Contents

The presentation consists of three Four Sections:

1. Background and Business Problem Overview
2. Data Overview
3. Exploratory Data Analysis
4. Model overview and performance summary
5. Key findings and insights
6. Business recommendations

Background of the Business and Objective

Background

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 guidance to achieve operational efficiency on Wind Energy Machines using predictive maintenance practices.

Predictive maintenance uses sensor information and analysis methods to measure and predict degradation and future component capabilities of Wind Machines using the sensors which are fitted across different machines.

The sensors fitted across different machines involved in the process of energy generation collect data related to various environmental factors (temperature, humidity, wind speed, etc.) and additional features related to various parts of the wind turbine (gearbox, tower, blades, break, etc.).

Objective

The objective is to build various classification models, tune them and find the best one that will help identify failures so that the wind machine generator could be repaired before failing/breaking and the overall maintenance cost of the generators can be brought down.

Maintenance cost = TP x (Repair cost) + FN x (Replacement cost) + FP x (Inspection cost)

Here the objective is to reduce the maintenance cost so, we want a metric that could reduce the maintenance cost.

Data Overview

- The data provided is a transformed version of original data which was collected using sensors.
- Train.csv - To be used for training and tuning of models.
- Test.csv - To be used only for testing the performance of the final best model.
- Both the datasets consist of 40 predictor variables and 1 target variable
- The train dataset has 40000 rows and 41 columns.
- The test dataset has 10000 rows and 41 columns.
- Most of the data is in numerical in nature.
- Predictor V1 has 46 and V2 has 39 missing values in Train data.
- Predictor V1 has 11 and V2 has 7 missing values in Test data.
- There are no duplicate values in train and test datasets.
- The data is normally distributed for majority of the feature columns.

Data description and Model overview

- We split the training data into the 80-20 ratio for train and validation set.
- We did missing value treatment on train data using the imputer method.
- We evaluated the performance on train and validation data using following models.
- Logistic Regression
- Decision Tree Classifier
- Random Forest Classifier
- Bagging Classifier
- Gradient Boosting Classifier
- Ada Boost Classifier

Model evaluation overview and performance summary

Model Performance on Training data with all the models

Model	Accuracy	Recall	Precision	F1 Score	Minimum_Vs_Model_cost
Logistic Regression	0.967	0.479	0.849	0.613	0.527
Decision Tree	1.000	1.000	1.000	1.000	1.000
Random forest	1.000	0.998	0.999	0.999	0.997
Bagging	0.998	0.957	0.998	0.977	0.932
Gradient Boost	0.987	0.781	0.978	0.868	0.730
Adaboost	0.975	0.632	0.879	0.735	0.609

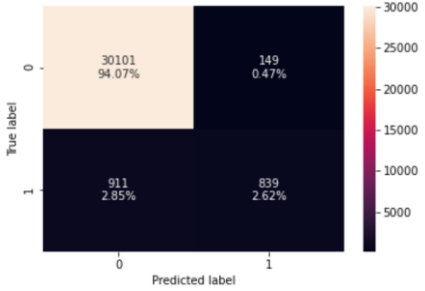
Model evaluation overview and performance summary

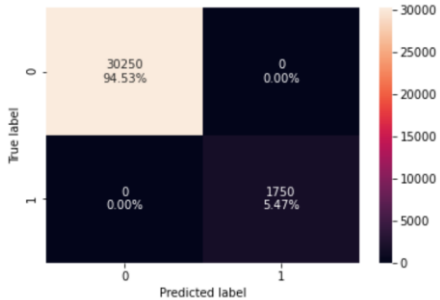
Model Performance on Validation data with all the models

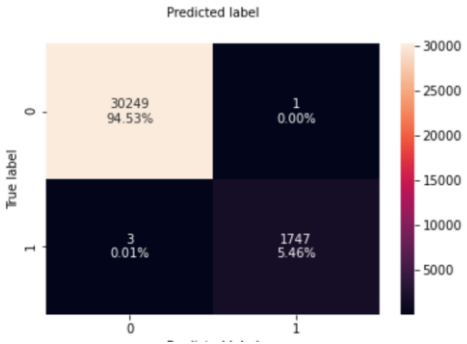
Model	Accuracy	Recall	Precision	F1 Score	Minimum_Vs_Model_cost
Logistic Regression	0.967	0.483	0.854	0.617	0.529
Decision Tree	0.970	0.741	0.714	0.727	0.654
Random forest	0.987	0.776	0.988	0.869	0.726
Bagging	0.984	0.746	0.953	0.837	0.697
Gradient Boost	0.984	0.735	0.961	0.833	0.689
Adaboost	0.971	0.586	0.842	0.691	0.579

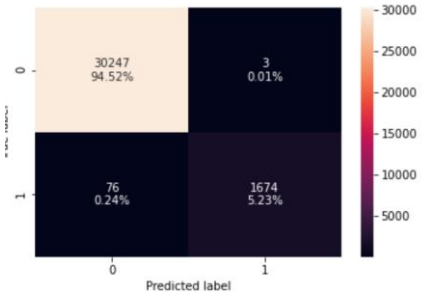
Model evaluation overview and performance summary

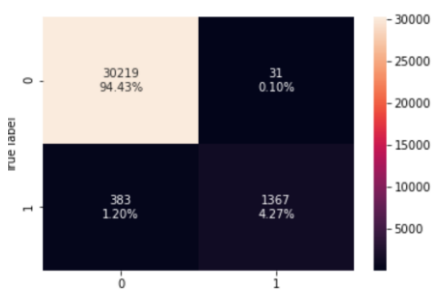
Confusion Matrix for all the models with Training data

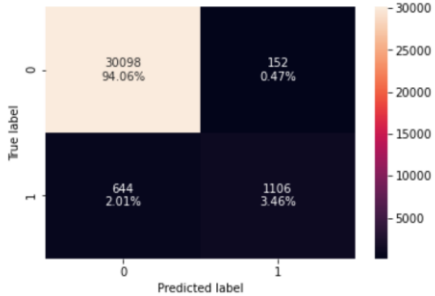






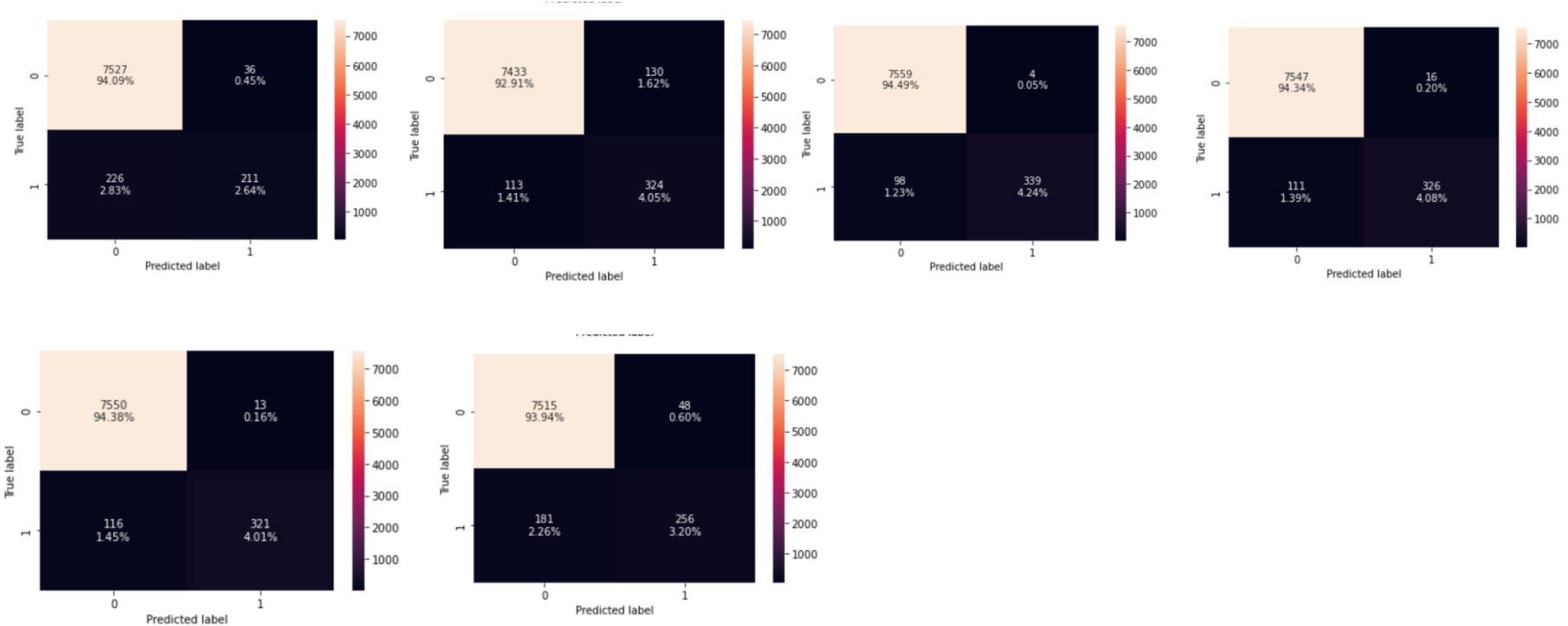






Model evaluation overview and performance summary

Confusion Matrix for all the models with Validation data



Model evaluation overview and performance summary

Model Performance on Oversampled Train data with

Model	Accuracy	Recall	Precision	F1 Score	Minimum_Vs_Model_cost
Logistic Regression	0.873	0.875	0.872	0.874	0.799
Decision Tree	1.000	1.000	1.000	1.000	1.000
Random forest	1.000	1.000	1.000	1.000	1.000
Bagging	0.999	0.998	1.000	0.999	0.997
Gradient Boost	0.944	0.918	0.969	0.943	0.872
Adaboost	0.906	0.897	0.912	0.905	0.833

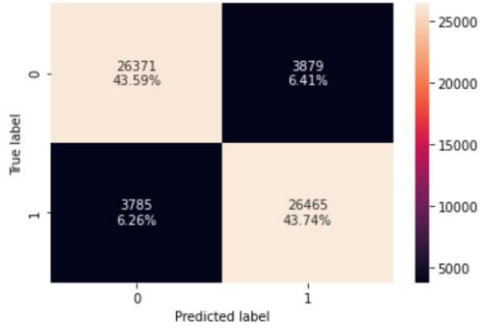
Model evaluation overview and performance summary

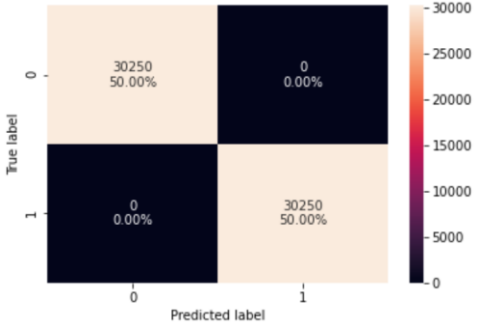
Model Performance on Oversampled Validation data

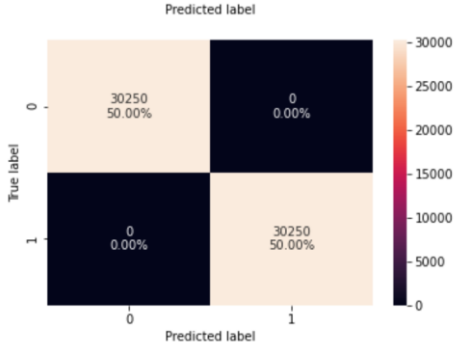
Model	Accuracy	Recall	Precision	F1 Score	Minimum_Vs_Model_cost
Logistic Regression	0.871	0.840	0.276	0.415	0.500
Decision Tree	0.950	0.819	0.527	0.642	0.647
Random forest	0.990	0.863	0.954	0.906	0.805
Bagging	0.985	0.842	0.870	0.856	0.766
Gradient Boost	0.962	0.881	0.604	0.717	0.719
Adaboost	0.905	0.847	0.348	0.493	0.560

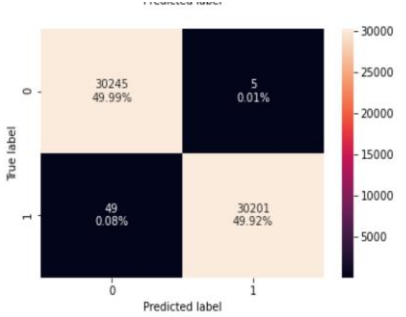
Model evaluation overview and performance summary

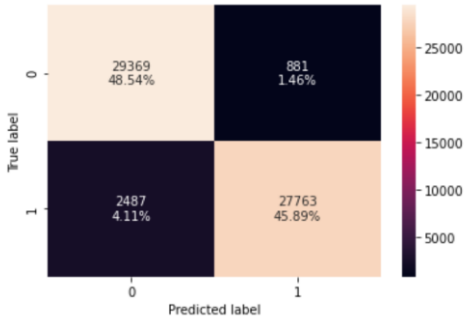
Confusion Matrix for all the models on oversampled trained data

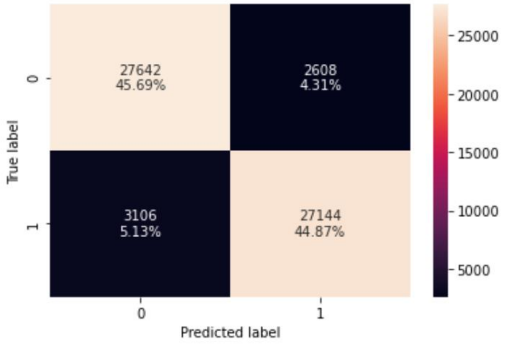






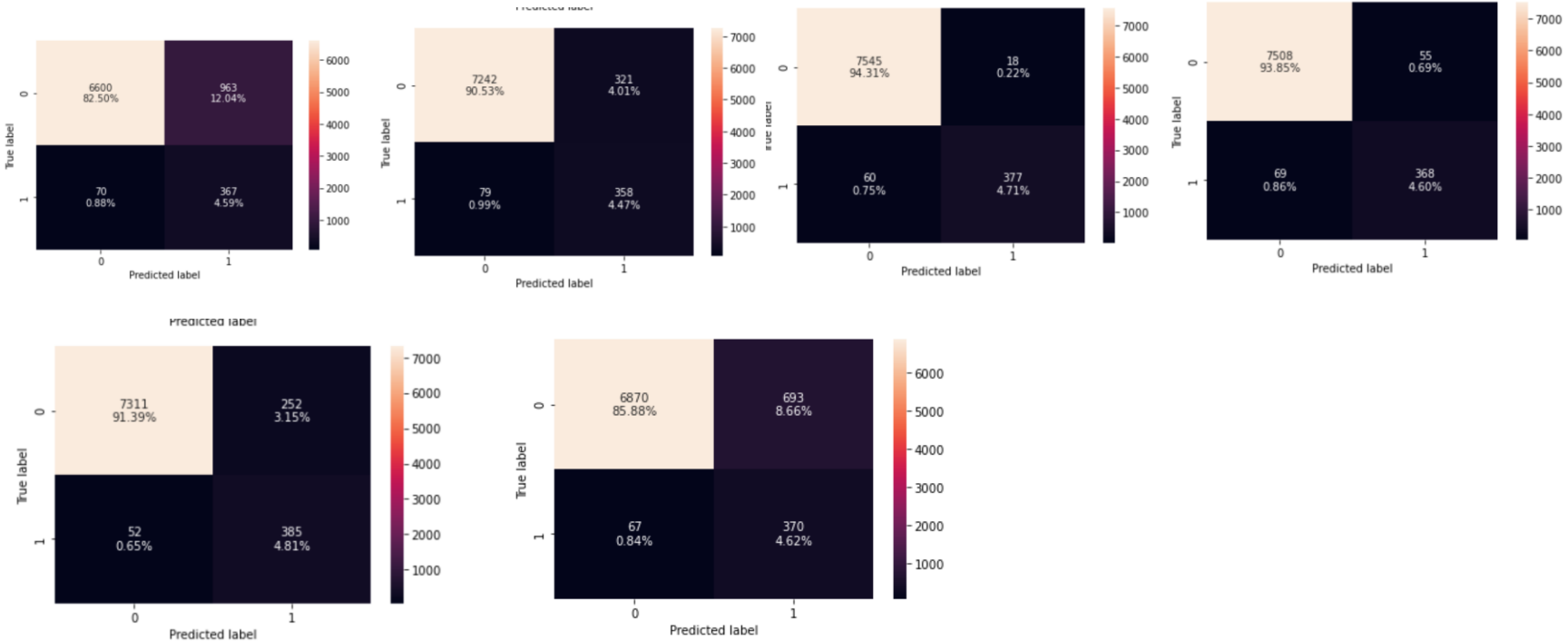






Model evaluation overview and performance summary

Confusion Matrix for all the models on oversampled validation data



Model evaluation overview and performance summary

Model Performance on Under Sampled Train data

Model	Accuracy	Recall	Precision	F1 Score	Minimum_Vs_Model_cost
Logistic Regression	0.862	0.856	0.867	0.861	0.779
Decision Tree	1.000	1.000	1.000	1.000	1.000
Random forest	1.000	1.000	1.000	1.000	1.000
Bagging	0.989	0.979	0.999	0.989	0.966
Gradient Boost	0.953	0.923	0.982	0.951	0.882
Adaboost	0.903	0.887	0.917	0.902	0.823

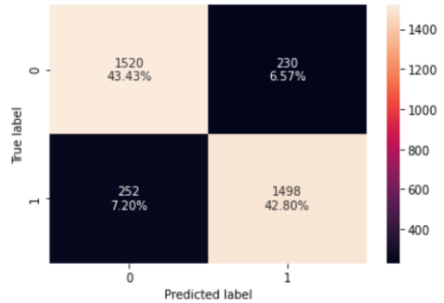
Model evaluation overview and performance summary

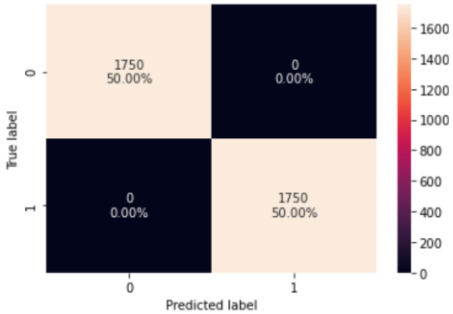
Model Performance on Under Sampled Validation data

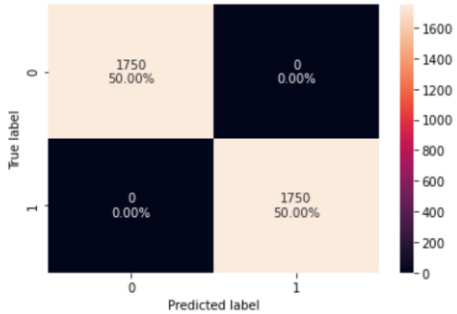
Model	Accuracy	Recall	Precision	F1 Score	Minimum_Vs_Model_cost
Logistic Regression	0.871	0.840	0.276	0.415	0.500
Decision Tree	0.950	0.819	0.527	0.642	0.647
Random forest	0.990	0.863	0.954	0.906	0.805
Bagging	0.985	0.842	0.870	0.856	0.766
Gradient Boost	0.962	0.881	0.982	0.951	0.882
Adaboost	0.905	0.847	0.348	0.493	0.560

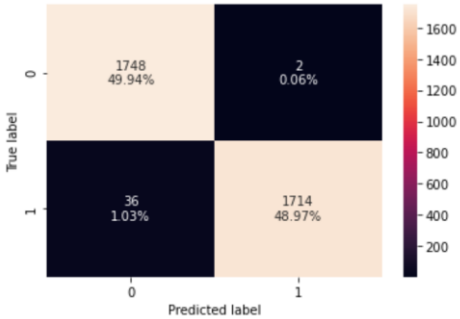
Model evaluation overview and performance summary

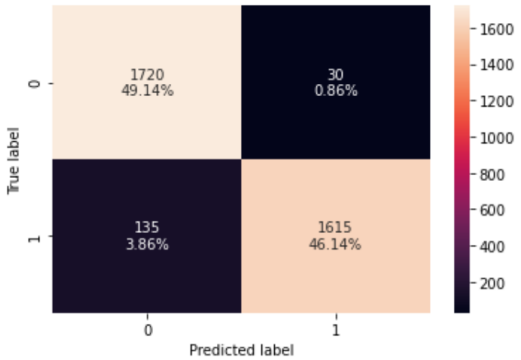
Confusion Matrix for all the models on Under sampled Train data

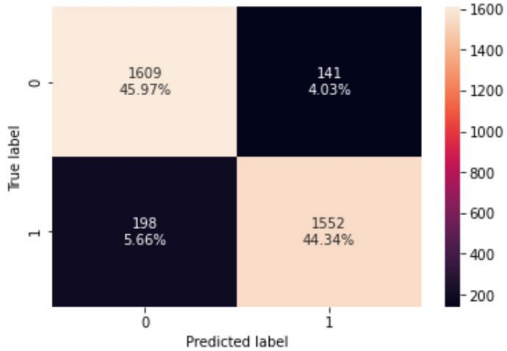






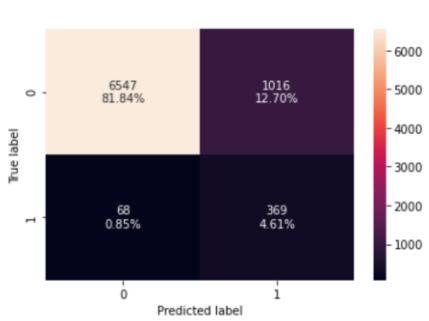


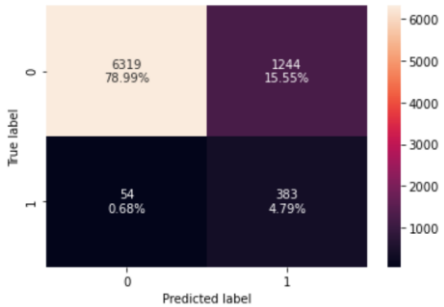


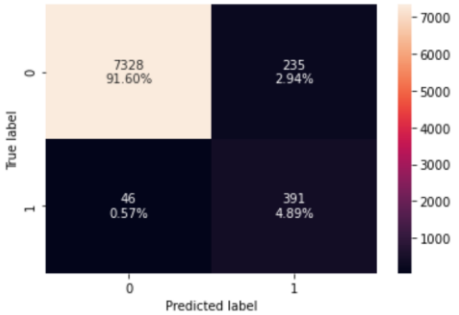


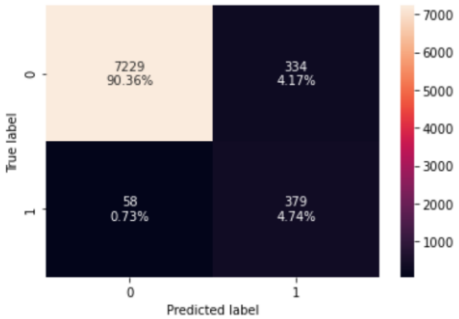
Model evaluation overview and performance summary

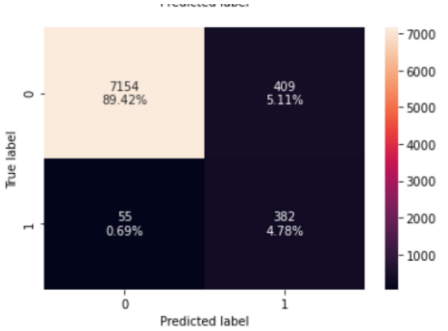
Confusion Matrix for all the models on Under sampled Validation data

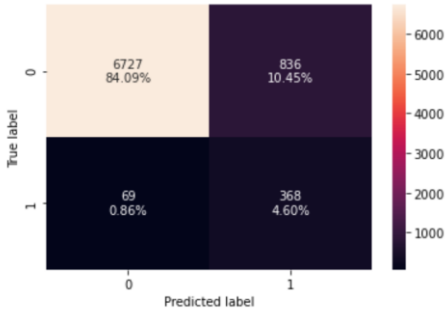












Model evaluation overview and performance summary

Model Comparison for Minimum_VS_Model_Cost value

Model	Original Train Data	Validation Data	OverSampled Train Data	OverSampled Validation Data	UnderSampled Train Data	UnderSampled Validation Data
Logistic Regression	0.527	0.529	0.799	0.500	0.779	0.500
Decision Tree	1.000	0.654	1.000	0.647	1.000	0.647
Random forest	0.997	0.726	1.000	0.805	1.000	0.805
Bagging	0.932	0.697	0.997	0.766	0.966	0.766
Gradient Boost	0.730	0.689	0.872	0.719	0.882	0.882
Adaboost	0.609	0.579	0.833	0.560	0.823	0.560

Model evaluation overview and performance summary

Based on Minimum_Vs_Model_cost values we selected our top 3 models.

Our top 3 models for evaluations are :

Random Forest

Bagging

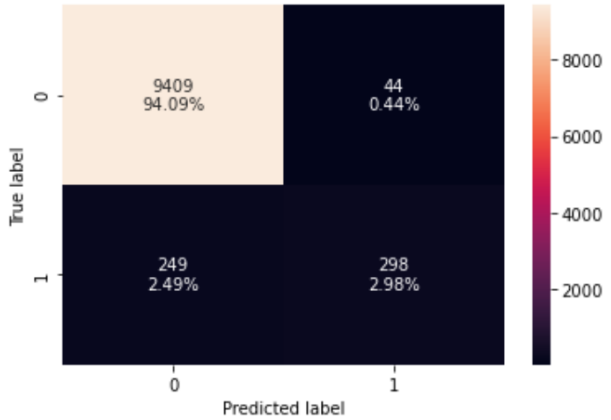
Gradient Boost

Training and Validation performance comparison:

	Training performance for Random Forest Tuned with Grid search	Validation performance for Random Forest Tuned with Grid search	Training performance for Bagging Tuned with Grid search	Validation performance for Bagging Tuned with Grid search	Training performance for GBM Tuned with Grid search	Validation performance for GBM Tuned with Grid search
Accuracy	0.999	0.990	1.000	0.988	0.979	0.977
Recall	0.998	0.867	1.000	0.867	0.923	0.886
Precision	1.000	0.950	1.000	0.909	0.755	0.739
F1	0.999	0.907	1.000	0.888	0.831	0.805
Minimum_Vs_Model_cost	0.997	0.809	1.000	0.800	0.814	0.772

Final model evaluation on test set

- We tuned Random Forest model using Grid search for the best parameters.
- We created the pipeline with the best parameters.
- We applied the model on the test data.



Model	Accuracy	Recall	Precision	F1 Score	Minimum_Vs_Model_cost
Random forest	0.989	0.863	0.935	0.897	0.801

Business Insights & Conclusions

- The objective was to come up with the best tuned model and apply it on the test data to identify failures so that the generator could be repaired before failing/breaking to keep the maintenance cost minimum.
- We picked Random Forest, Bagging ,Gradient Boost classification models among all the models we applied on the training data.
- We tuned these models with the best parameters and applied the tuned models on the validation set of data.
- After collecting all the performance matrices on the tuned classifications models, We chose the RandomClassifier classification model and applied it on the test data.
- We are able to predict 94.22% of Total Positives “No Failures” hence Minimum Maintenance cost for these machines.
- We are able to predict 4.70% of Total Negative i.e. “Failures” so a good estimate of maintenance cost to keep it down.
- We were able to bring down the maintenance cost by minimizing the False Negative and False Positive which is 0.31% and 0.77%.
- The RandomForest Classification model was able to predict the maintenance cost with the 98% Accuracy with 80% of Minimum_Vs_Model_cost.

greatlearning
Power Ahead

Happy Learning !