Business Presentation

Contents

- Business Problem Overview and Solution Approach
- Data Overview
- EDA
- Model Building
- Model Performance Comparison – choosing 3
- Model Performance after tuning
- Productionize the model
- Business Insights and Recommendations

Business Problem Overview and Solution Approach

- Core business idea

  Improve operational efficiency of Wind turbines through predictive maintenance.

- Problem to tackle

  Identify potential component failures at the right time for replacement such that costs of operation and maintenance are reduced.

- Financial implications

  - Component failure results in Increased operations and Maintenance cost.
  - Prematurely replacing components would increase replacement costs.
  - Incorrect prediction of component failures will result in High Inspectioon cost.

  Predicting component failures at right time helps cut down all Maintenance costs to the minimum possible cost.

- How to use ML model to solve the problem

  Using sensor data and previous failures, predict component failures using various classification models.

Data Overview

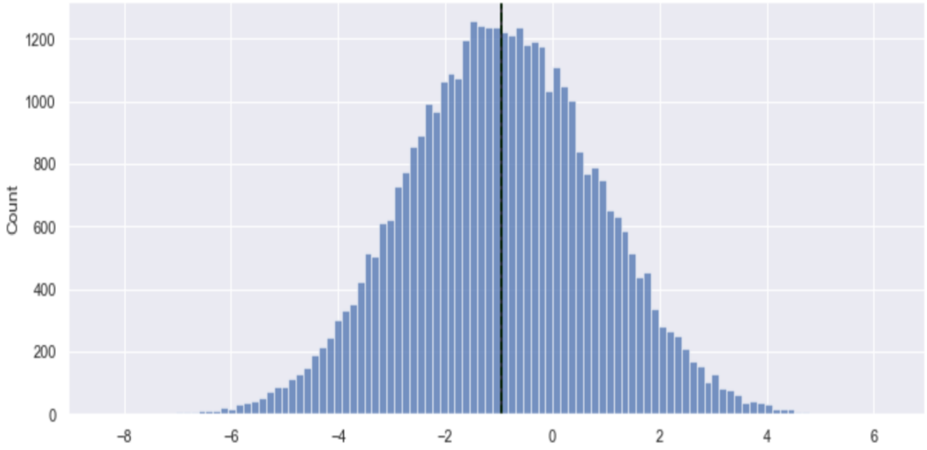
- The data is present in two csv files, Train.csv and Test.csv.
- The Train.csv contains 40000 rows and 41 columns while the Test.csv contain 10000 rows and 41 columns. Both the datasets contain 40 predictor variables and 1 target variable

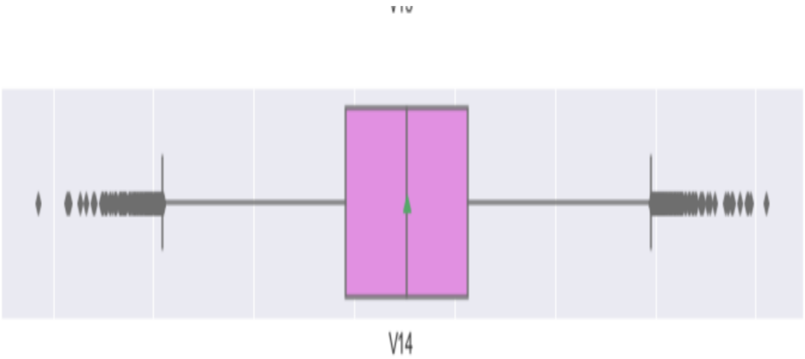
- All the variables are float except the Target variable which is integer containing 0 and 1.
- V1 and V2 consists of missing values. These need to be treated.

- Almost all the variables consists of negative values. Without any domain knowledge or actual names of variables, treating them might reduce the real variability in the given data.

- Maintenance cost = TP*(Repair cost) + FN*(Replacement cost) + FP*(Inspection cost)
- minimum possible maintenance cost = Actual failures*(Repair cost) = (TP + FN)*(Repair cost)
- maintenance cost associated with model = TP*(Repair cost) + FN*(Replacement cost) + FP*(Inspection cost)
- Our objective is to reduce the maximize the ratio of minimum possible maintenance cost and the maintenance cost associated with the model

EDA

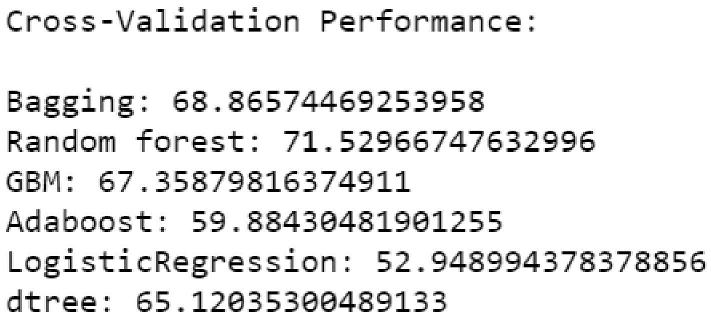




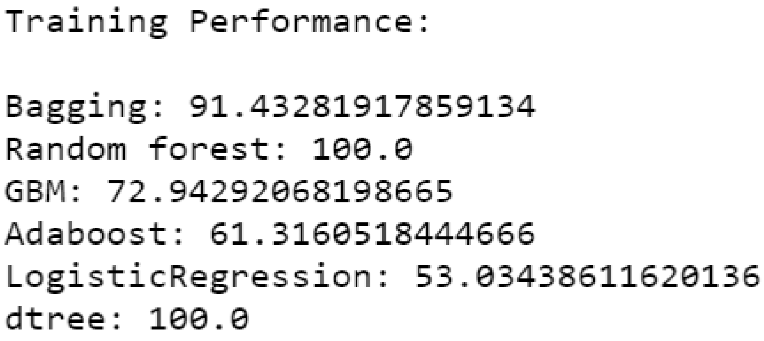
1. The distribution for all the variables is bell-shaped(normal distribution).
2. For all the Variables V1 to V39, the boxplot shows outliers on both the end.
3. Since, we do not know what these columns names are. Without any domain knowledge or actual names of variables, treating them might reduce the real variability in the given data.
4. We will not treat these outliers as they me represent the real market trend.

Model Performance with Original data

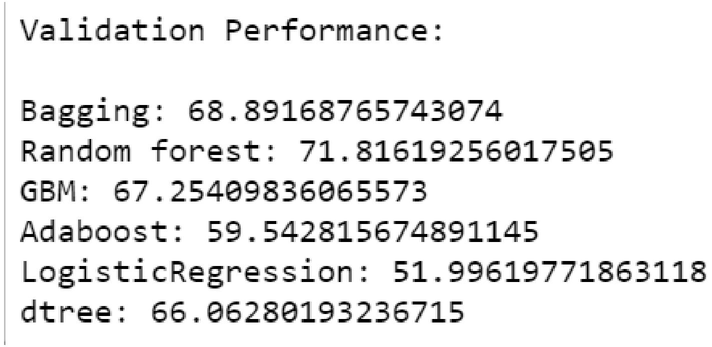
- Chosen 6 different models to calculate the cross validation performance and validation performance on original data



```
Cross-Validation Performance:

Bagging: 68.86574469253958
Random forest: 71.52966747632996
GBM: 67.35879816374911
Adaboost: 59.88430481901255
LogisticRegression: 52.948994378378856
dtree: 65.12035300489133
```

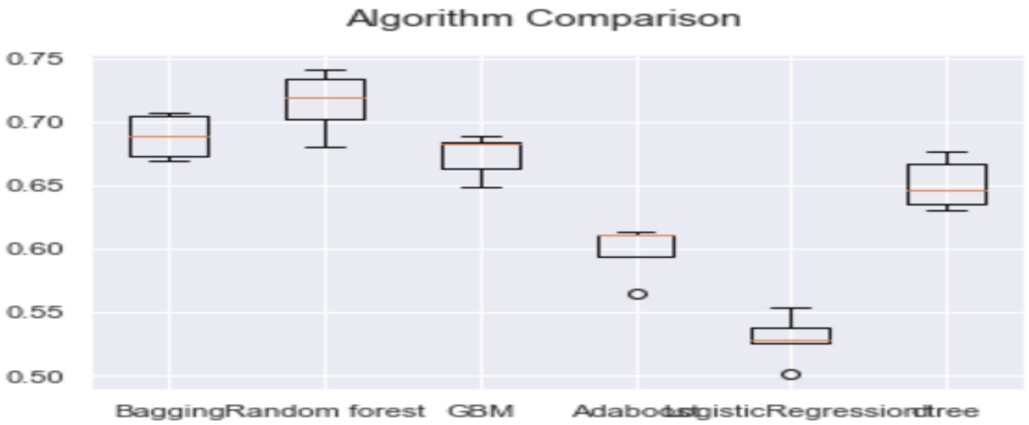


```
Training Performance:

Bagging: 91.43281917859134
Random forest: 100.0
GBM: 72.94292068198665
Adaboost: 61.3160518444666
LogisticRegression: 53.0343861162O136
dtree: 100.0
```



```
Validation Performance:

Bagging: 68.89168765743074
Random forest: 71.81619256017505
GBM: 67.25409836065573
Adaboost: 59.542815674891145
LogisticRegression: 51.99619771863118
dtree: 66.06280193236715
```

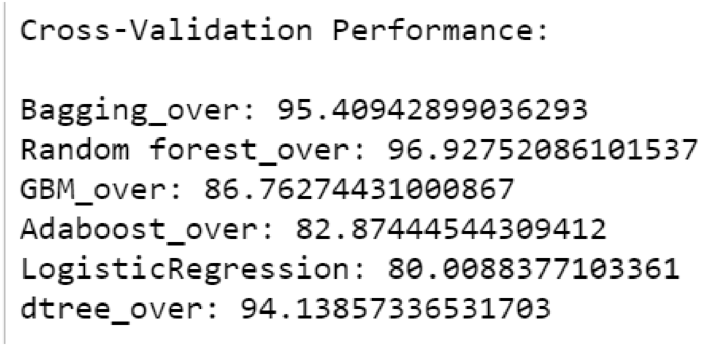
Model Performances – Original Data



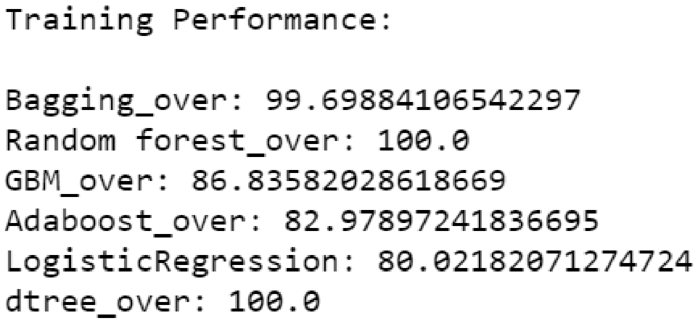
1. We can see that the Random Forest is giving the highest cross-validated scorer followed by Bagging and GBM.
2. The boxplot shows that the performance of these models are consistent and their performance on the validation set is also good.
3. AdaBoost and Logistic Regression have one outlier each.

Model Performance with Over Sampled data

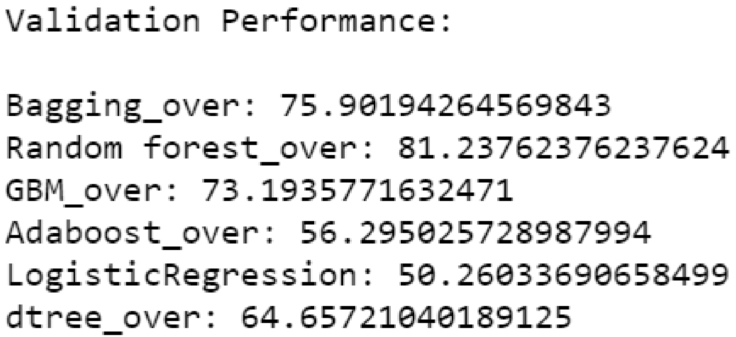
- After oversampling the data, the data has 56720 rows and 41 columns
- the model performances of these 6 models are as follows



```
Cross-Validation Performance:

Bagging_over: 95.40942899036293
Random forest_over: 96.92752086101537
GBM_over: 86.76274431000867
Adaboost_over: 82.87444544309412
LogisticRegression: 80.0088377103361
dtree_over: 94.13857336531703
```

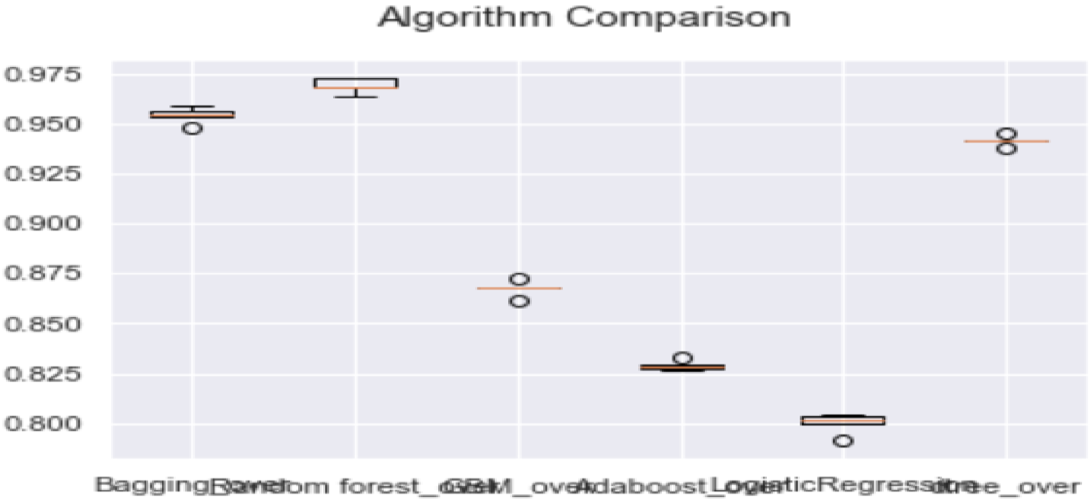


```
Training Performance:

Bagging_over: 99.69884106542297
Random forest_over: 100.0
GBM_over: 86.8358202861869
Adaboost_over: 82.97897241836695
LogisticRegression: 80.02182071274724
dtree_over: 100.0
```



```
Validation Performance:

Bagging_over: 75.90194264569843
Random forest_over: 81.23762376237624
GBM_over: 73.1935771632471
Adaboost_over: 56.29502572898799 4
LogisticRegression: 50.26033690658499
dtree_over: 64.65721040189125
```

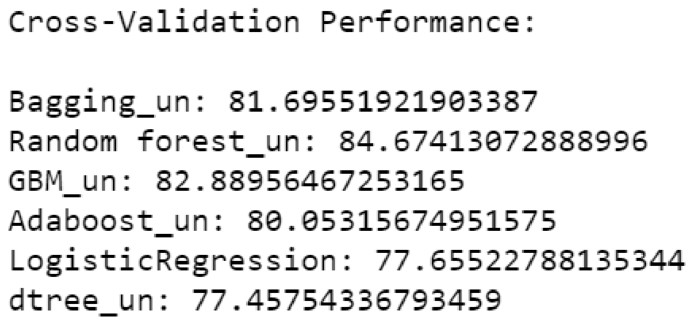
Model Performance – Over Sampled



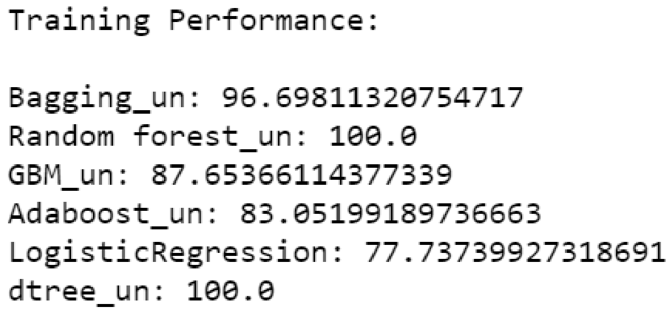
1. We can see that the Random Forest – over sampled is giving the highest cross-validated scorer followed by GBM – over sampled and Bagging – over sampled.
2. The boxplot shows that the performance of these models are consistent and their performance on the validation set is also good.
3. Random Forest has one outlier.

Model Performance with Under Sampled data

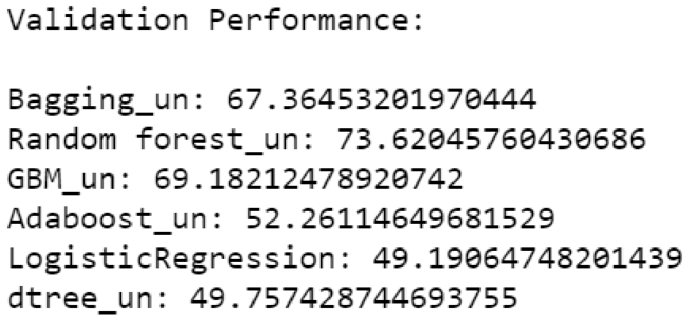
- After under sampling, the data has 3280 rows and 41 columns.



```
Cross-Validation Performance:

Bagging_un: 81.69551921903387
Random forest_un: 84.67413072888996
GBM_un: 82.88956467253165
Adaboost_un: 80.05315674951575
LogisticRegression: 77.65522788135344
dtree_un: 77.45754336793459
```

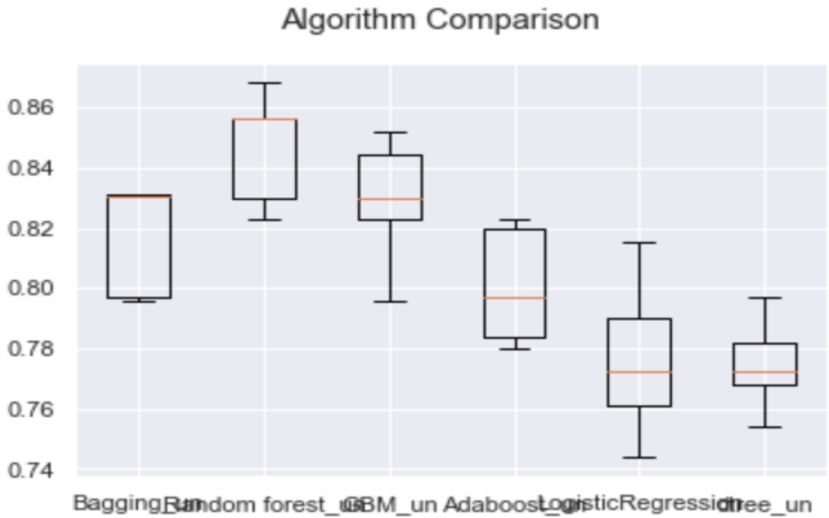


```
Training Performance:

Bagging_un: 96.69811320754717
Random forest_un: 100.0
GBM_un: 87.65366114377339
Adaboost_un: 83.05199189736663
LogisticRegression: 77.73739927318691
dtree_un: 100.0
```



```
Validation Performance:

Bagging_un: 67.36453201970444
Random forest_un: 73.62045760430686
GBM_un: 69.18212478920742
Adaboost_un: 52.26114649681529
LogisticRegression: 49.19064748201439
dtree_un: 49.757428744693755
```

Model Performance – UnderSampled.



- 1. We can see that the Random Forest – under sampled is giving the highest cross-validated scorer followed by Gradient Boosting Model – under sampled and Bagging under sampled.
- 2. The boxplot shows that the performance of the above three models are consistent and their performance on the validation set is also good.

Model Selection

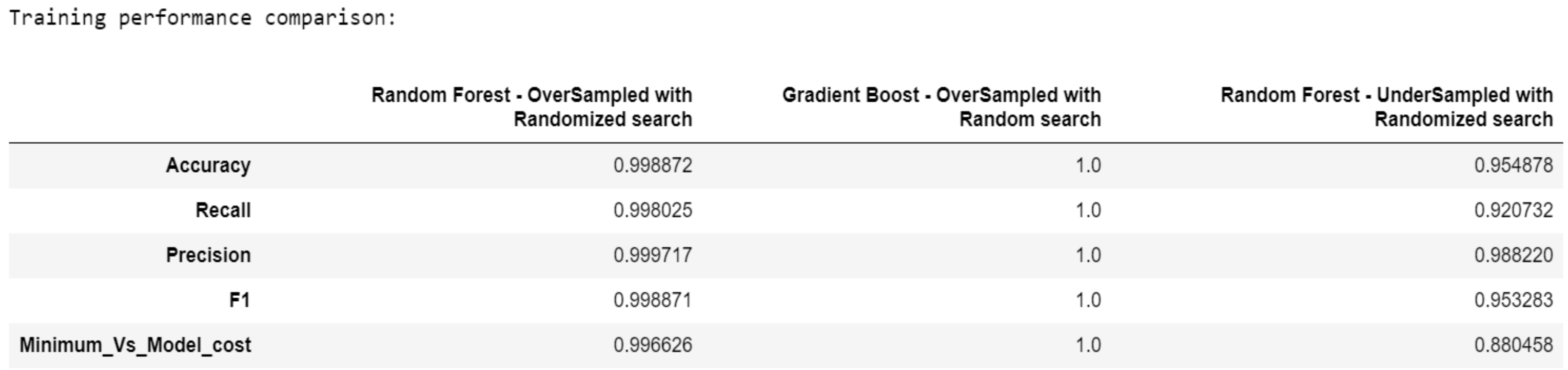
- **From the above 9 models built, we will choose 3 models to further tune them to improve the performance.**

1. Random Forest – over sampled
2. Gradient Boosting Model - over sampled
3. Random Forest – under sampled

These models have high score value. Since, our aim is to increase the scorer value.

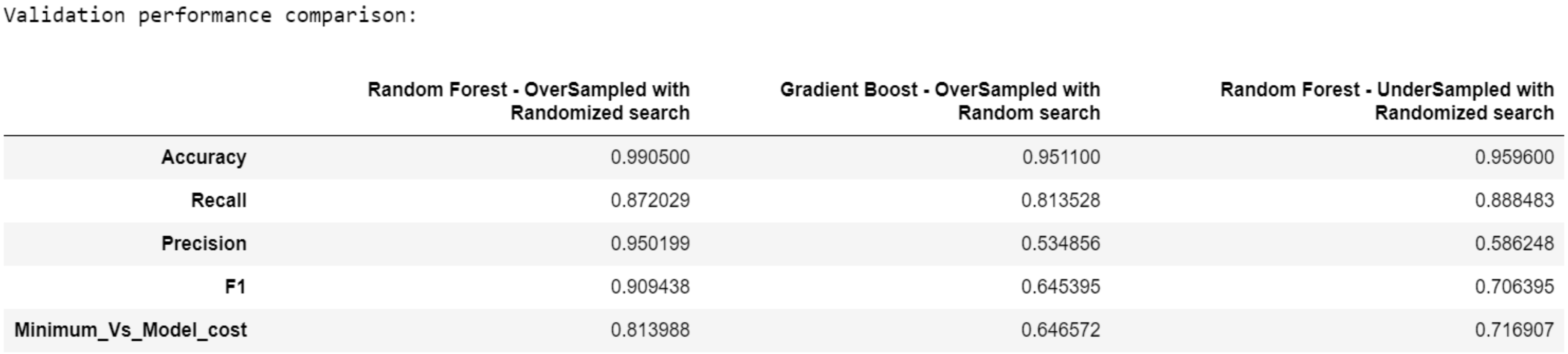
1. Even though, these models have high score value and are overfitting. We have chosen models with high score value and also with the difference between their respective Cross Validation scores and Validation Performance scores being lesser.
2. We can see that the Random Forest – over sampled is giving the highest cross-validated scorer followed by Gradient Boosting Model - over sampled and Random Forest – under sampled.
3. The boxplot shows that the performance of Random Forest –over sampled, GBM – over sampled and Random Forest – under sampled are consistent and their performance on the validation set is also good.
4. Hence, we will tune Random Forest – over sampled , Gradient Boosting Model - over sampled and Random Forest – under sampled models and see if we can reduce the overfitting and also difference between their respective Cross Validation and Validation Performance Scores.

Model Performance – After Hypertuning



Training performance comparison:

	Random Forest - OverSampled with Randomized search	Gradient Boost - OverSampled with Random search	Random Forest - UnderSampled with Randomized search
Accuracy	0.998872	1.0	0.954878
Recall	0.998025	1.0	0.920732
Precision	0.999717	1.0	0.988220
F1	0.998871	1.0	0.953283
Minimum_Vs_Model_cost	0.996626	1.0	0.880458

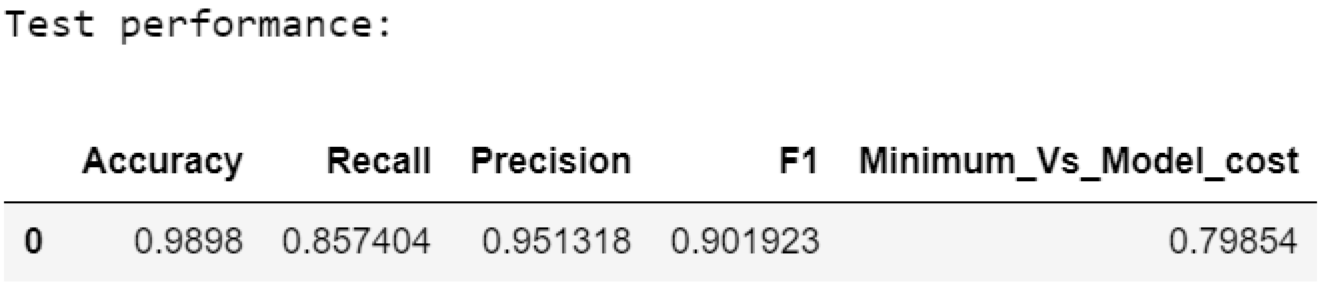


Validation performance comparison:

	Random Forest - OverSampled with Randomized search	Gradient Boost - OverSampled with Random search	Random Forest - UnderSampled with Randomized search
Accuracy	0.990500	0.951100	0.959600
Recall	0.872029	0.813528	0.888483
Precision	0.950199	0.534856	0.586248
F1	0.909438	0.645395	0.706395
Minimum_Vs_Model_cost	0.813988	0.646572	0.716907

Model Performance – Final Model

- Final Model chosen after hypertuning is Random Forest – Over Sampled. The performance of the model after running on test data.



```
Test performance:
```

	Accuracy	Recall	Precision	F1	Minimum_Vs_Model_cost
0	0.9898	0.857404	0.951318	0.901923	0.79854

Business Insights and Recommendations

- Recommendations based on interpretation of the model input variables

  - V18, V36, V39, V15, V26, V16, V3 are the top features that influence component wear and Tear.
  - Data is mostly normally distributed as the data is auto fetched from sensors. It is vital that the sensors work all the time providing real time data so that the models can be kept up to date.
  - We need to ensure that more sensors be added to collect accurate information of top features V18, V36, V39, V15, V26, V16, V3
  - Invest in sensors for other components of the wind turbines so that overall operation cost is completely brought down.
  - The Generator sensor data could also be used to turn on or off the generators so that efficiency be further improved. If conditions are averse for turbines operation, then the turbines can be shutoff to save them from further wear.

Business Insights and Recommendations

- Comments on additional data sources for model improvement, model implementation in real world, and potential business benefits from the model.

    - Consistent and clean sensor data has a ensured a stable model. Almost all models have performed consistently on train, validation and test.

    - Without background of variables V1 – V39, it is difficult to decide if some outliers in data are real anomalies affecting data or if they are consistent, repeating indicators.

    - Given the data is imbalanced with approximately 6% of failures, the initial model on raw data could achieve scores only around the range of 60%

    - Upsampling of data has given a big boost to the overall scores. Howver, there seems to be some overfitting.

    - Undersampling of data has also improved performance of the model. Howver the scores were less compared to upsampling due to possible loss of data.

    - Our aim was to pick the model with high scores and also keep the variability of scores across train, test and validation sets consistent in order to have a predictable score in production.

Business Insights and Recommendations

- Comments on additional data sources for model improvement, model implementation in real world, and potential business benefits from the model.

  - Random Forest – over sampled is giving the highest cross-validated scorer followed by Gradient Boosting – over sampled and Random Forest – under sampled.

  - Out of all the models, Random Forest with oversampling, Gradient boost with oversampling and Random Forest with under sampling gave good results.

  - These models were tuned further using RandomSearchCV and “Random forest with Oversampling” gave the best results with scores close to 80%

greatlearning
Power Ahead

Happy Learning !