ReneWind

ReneWind: Model Tuning

Contents / Agenda

Executive Summary

- The AdaBoost Classifier tuned using oversampled data has the best performance
- V30, V9 and V18 are most important features
    - They can be deciphered to determine and analyze the actual variables to understand their impact on the predictive task at hand
- This model can be further used to detect if a wind turbine will fail or not and this will help reduce the cost.
- We also saw that there might be few points near around the classification threshold (0.5 by default) which can be further studied by the engineer and a final call could be made.

Business Problem Overview and Solution Approach

- 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. Out of all the renewable energy alternatives, wind energy is one of the most developed technologies worldwide.

- Predictive maintenance uses sensor information and analysis methods to measure and predict degradation and future component capability. The idea behind predictive maintenance is that 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.

- 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.).

- ReneWind is a company working on improving the machinery/processes involved in the production of wind energy using machine learning and has collected data of generator failure of wind turbines using sensors.

Business Problem Overview and Solution Approach

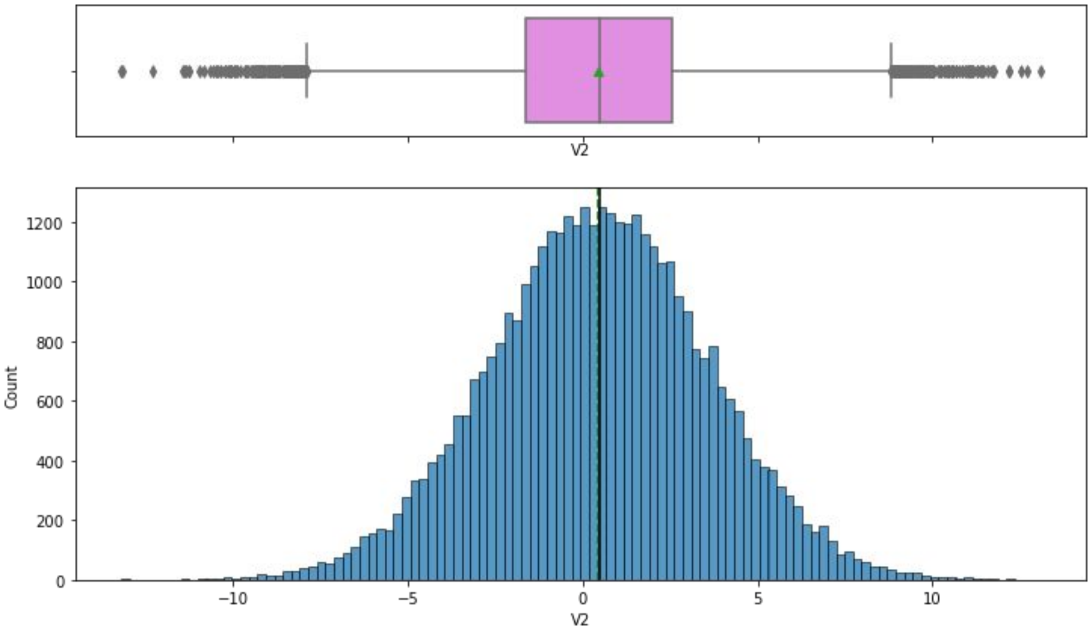
- The task at hand is to build various classification models, tune them, and find the best one that will help identify failures so that the generators could be repaired before failing/breaking to reduce the overall maintenance cost.

- The nature of predictions made by the classification model will translate as follows:

    - True positives (TP) are failures correctly predicted by the model. These will result in repairing costs.
    - False negatives (FN) are real failures where there is no detection by the model. These will result in replacement costs.
    - False positives (FP) are detections where there is no failure. These will result in inspection costs.

- It is given that the cost of repairing a generator is much less than the cost of replacing it, and the cost of inspection is less than the cost of repair

EDA Results



The distribution of all the variables are similar
All the variables are close to normally distributed

Link to Appendix slide on data background check

Data Preprocessing

- The data shared is a ciphered version containing 20000 observations in the train set and 5000 in the test set
- The number of features provided is 40 but the data is ciphered hence, the column names are anonymous
- There were few missing values in V1 and V2, we imputed them using the median and to avoid data leakage we imputed missing values after splitting train data into train and validation sets.
- 94.5% of the observations are negative and only 5.5% observations are a positive representing failure. The dataset is highly imbalanced so we tried undersampling and oversampling techniques to balance the data.

Model Performance Summary - Tuned models

Training Performance

	Gradient Boosting tuned with oversampled data	XGBoost tuned with oversampled data	AdaBoost tuned with oversampled data	Random forest tuned with undersampled data
Accuracy	0.993	0.978	0.992	0.961
Recall	0.992	1.000	0.988	0.933
Precision	0.994	0.959	0.995	0.989
F1	0.993	0.979	0.992	0.960

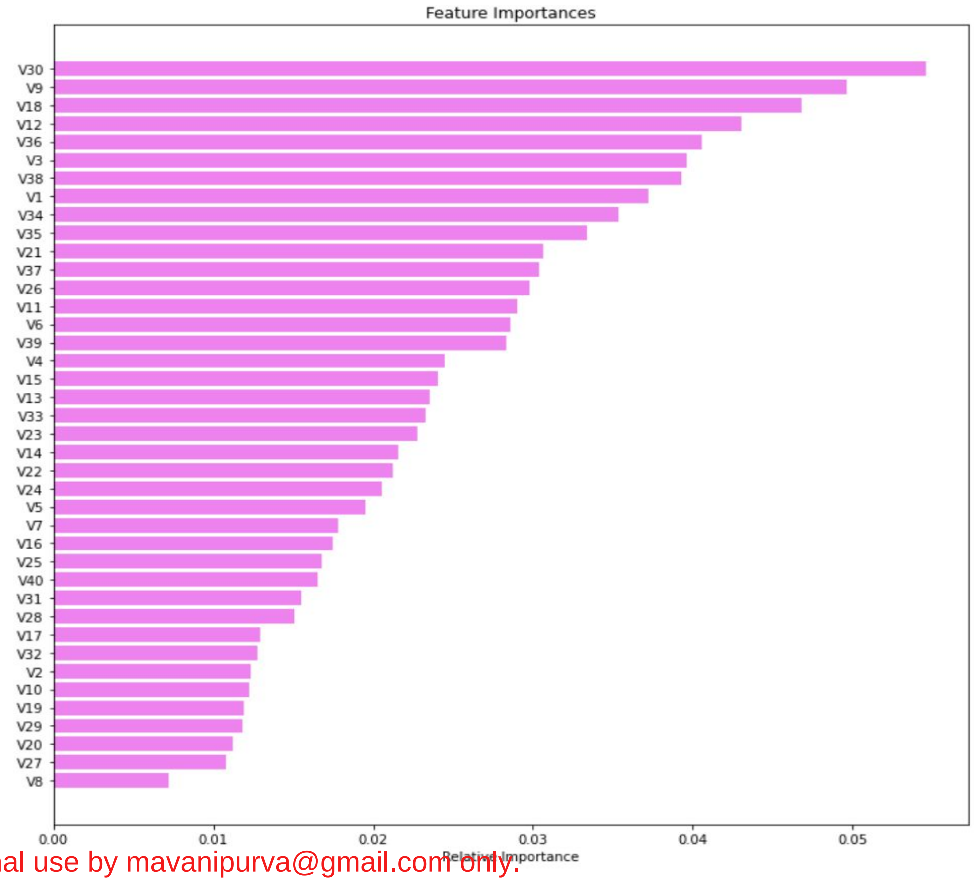
	Gradient Boosting tuned with oversampled data	XGBoost tuned with oversampled data	AdaBoost tuned with oversampled data	Random forest tuned with undersampled data
Accuracy	0.969	0.938	0.979	0.938
Recall	0.856	0.885	0.853	0.885
Precision	0.678	0.470	0.787	0.468
F1	0.757	0.614	0.819	0.612

Validation Performance

AdaBoost model trained with oversampled data has generalised performance, so let's consider it as the best model.

Final Model Feature Importance

- V30, V9 and V18 are most important features. They can be deciphered to determine and analyze the actual variables to understand their impact on the predictive task at hand



Productionize and test the final model using pipelines

- We build the pipeline with the following components:
  - Simple Imputer for imputation
  - AdaBoost model with oversampled data
- AdaBoost model performed well on test data

Data	Accuracy	Recall	Precision	F1
Test	0.978	0.844	0.780	0.811

Link to Appendix slide on model assumptions

APPENDIX

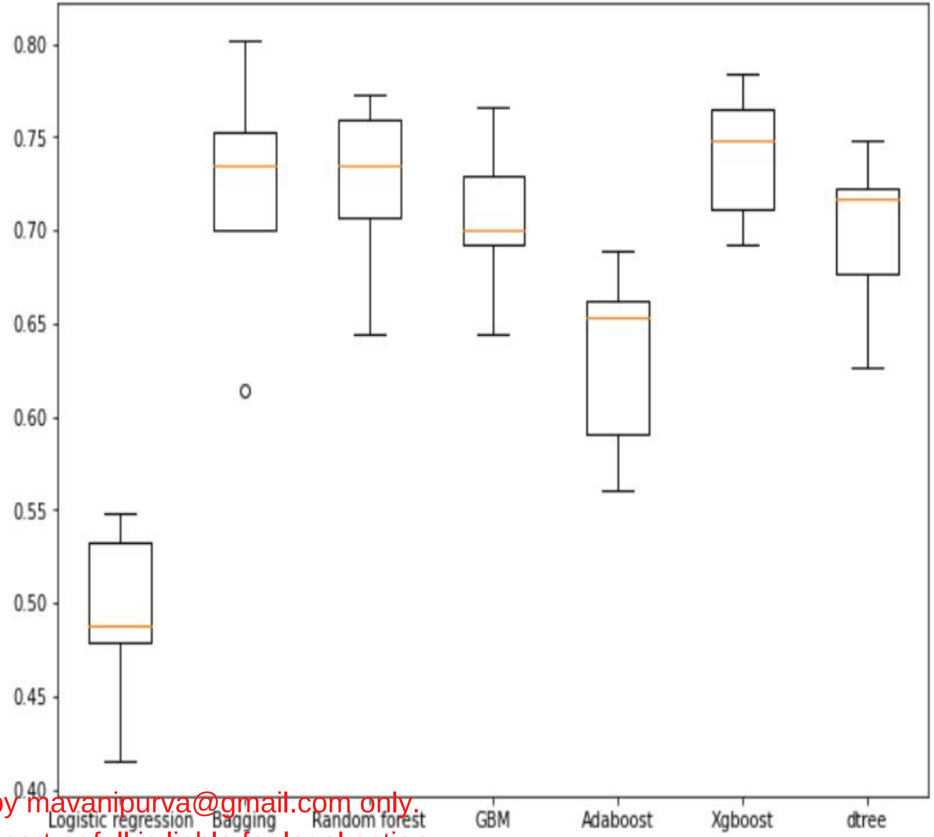
Model Performance Summary - Original data

```
Cross-Validation performance on training dataset:

Logistic regression: 0.4927566553639709
Bagging: 0.7210807301060529
Random forest: 0.7235192266070268
GBM: 0.7066661857008874
Adaboost: 0.6309140754635308
Xgboost: 0.7403217661063415
dtree: 0.698282952167953 2

Validation Performance:

Logistic regression: 0.4820143884892086 3
Bagging: 0.7302158273381295
Random forest: 0.7266187050359713
GBM: 0.7230215827338129
Adaboost: 0.6762589928057554
Xgboost: 0.762589928057554
dtree: 0.7050359712230215
```



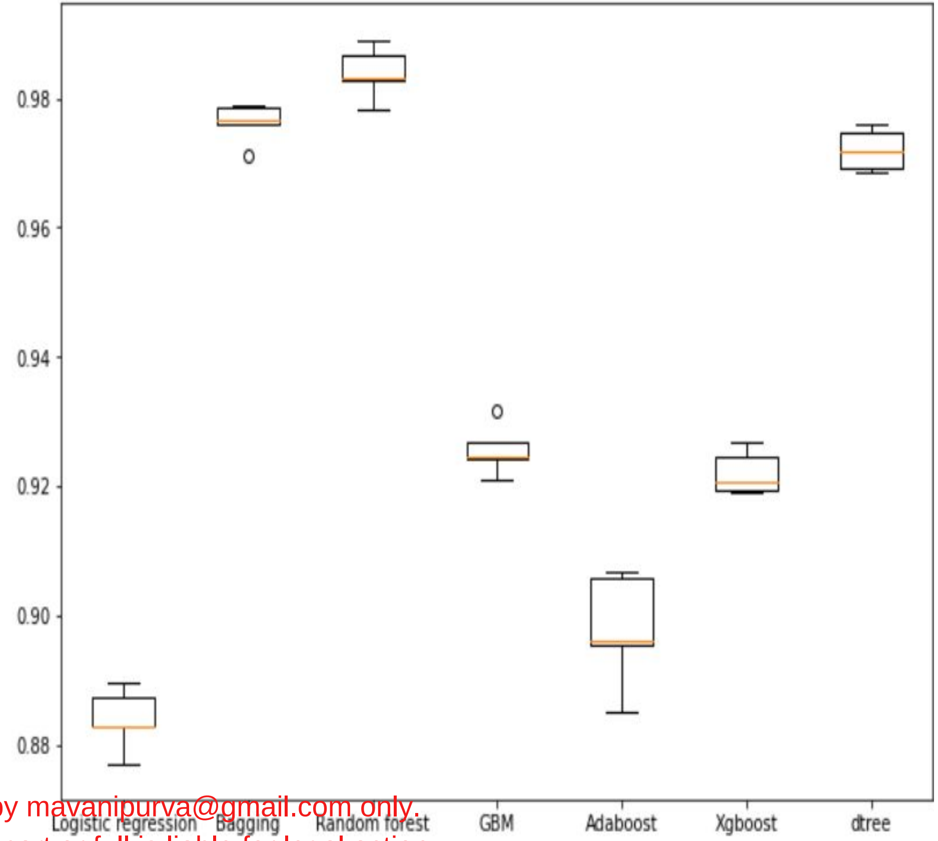
Model Performance Summary - Oversampled data

```
Cross-Validation performance on training dataset:

Logistic regression: 0.883963699328486
Bagging: 0.9762141471581656
Random forest: 0.9839075260047615
GBM: 0.9256068151319724
Adaboost: 0.8978689011775473
Xgboost: 0.922148207398388
dtree: 0.9720494245534969

Validation Performance:

Logistic regression: 0.8489208633093526
Bagging: 0.8345323741007195
Random forest: 0.8489208633093526
GBM: 0.8776978417266187
Adaboost: 0.8561151079136691
Xgboost: 0.8741007194244604
dtree: 0.7769784172661871
```



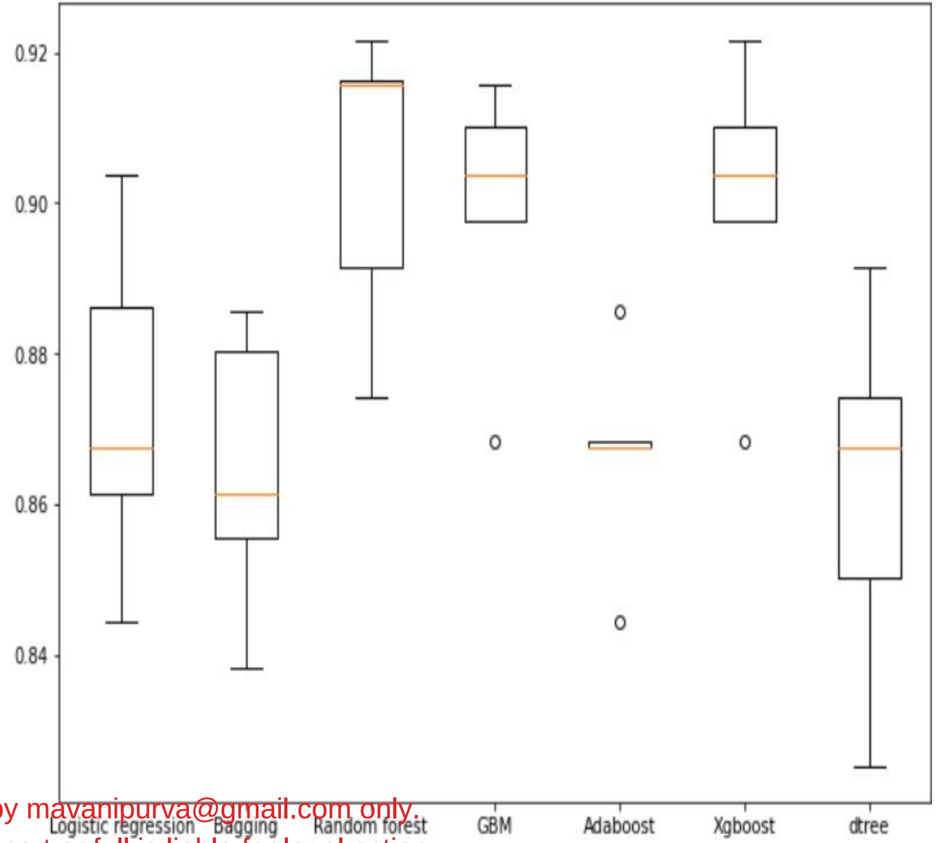
Model Performance Summary - Undersampled data

```
Cross-Validation performance on training dataset:

Logistic regression: 0.8726138085275232
Bagging: 0.8641945025611427
Random forest: 0.9038669648654498
GBM: 0.8990621167303946
Adaboost: 0.8666113556020489
Xgboost: 0.9002669360075031
dtree: 0.8617776495202367

Validation Performance:

Logistic regression: 0.8525179856115108
Bagging: 0.8705035971223022
Random forest: 0.8920863309352518
GBM: 0.8884892086330936
Adaboost: 0.8489208633093526
Xgboost: 0.8884892086330936
dtree: 0.841726618705036
```



greatlearning
Power Ahead

Happy Learning !