ReneWind

Travis Bruns

Table of Contents

- Business Problem
- Objective
- Model Evaluation and Performance
- Business Insights and Recommendations

Business Problem Overview

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. The U.S Department of Energy has put together a guide to achieving operational efficiency using predictive maintenance practices.

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.).

An opportunity exists to leverage the sensor data to build machine learning models to decrease maintenance costs.

Objective

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. They have shared a ciphered version of the data, as the data collected through sensors is confidential (the type of data collected varies with companies).

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

Data Overview

Data has 40 predictors, 40000 observations in the training set and 10000 in the test set..

- 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

Predictive Values

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

- True positives (TP) are failures correctly predicted by the model.
- False negatives (FN) are real failures in a wind turbine where there is no detection by model.
- False positives (FP) are detections in a wind turbine where there is no failure.

So, the maintenance cost associated with the model would be:

Maintenance cost = TP(Repair cost) + FN(Replacement cost) + FP*(Inspection cost) where,

- Replacement cost = $40,000
- Repair cost = $15,000
- Inspection cost = $5,000

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

- The minimum possible maintenance cost = Actual failures(Repair cost) = (TP + FN)(Repair cost)
- The maintenance cost associated with model = TP(Repair cost) + FN(Replacement cost) + FP*(Inspection cost)

So, we will try to maximize the ratio of minimum possible maintenance cost and the maintenance cost associated with the model.

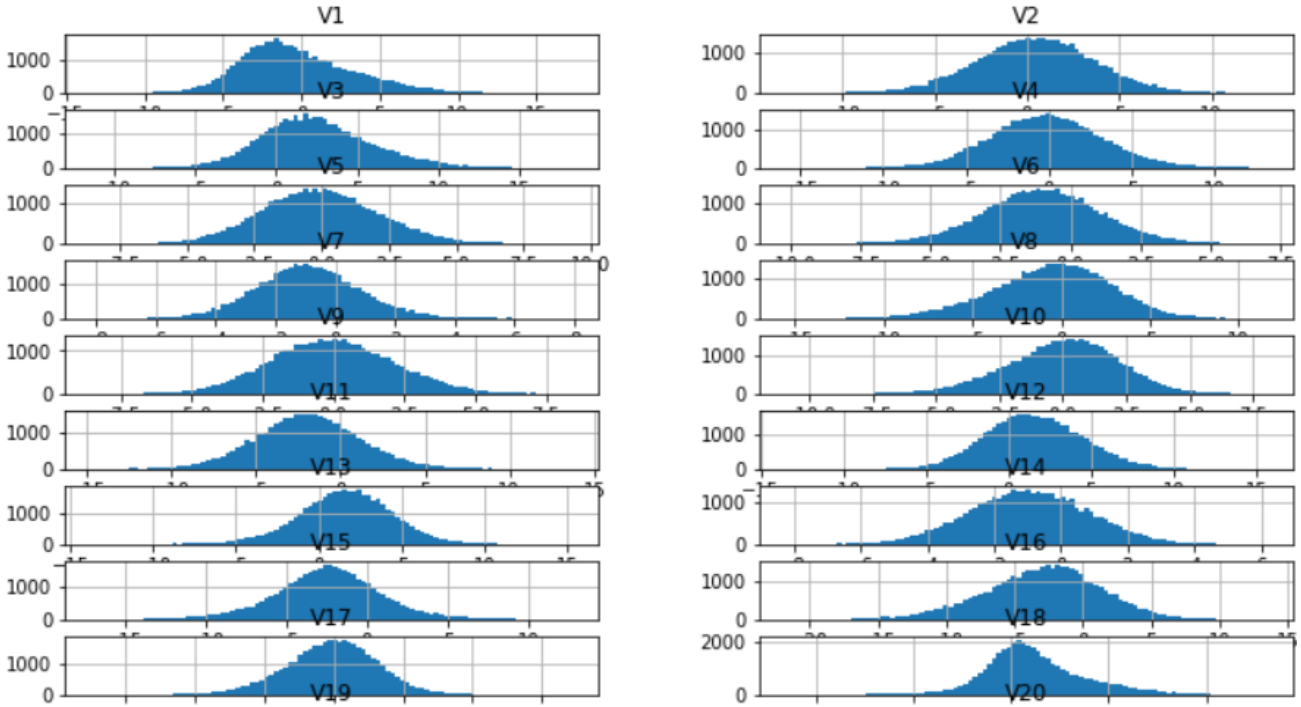
The value of this ratio will lie between 0 and 1, the ratio will be 1 only when the maintenance cost associated with the model will be equal to the minimum possible maintenance cost.

Raw Data Summary

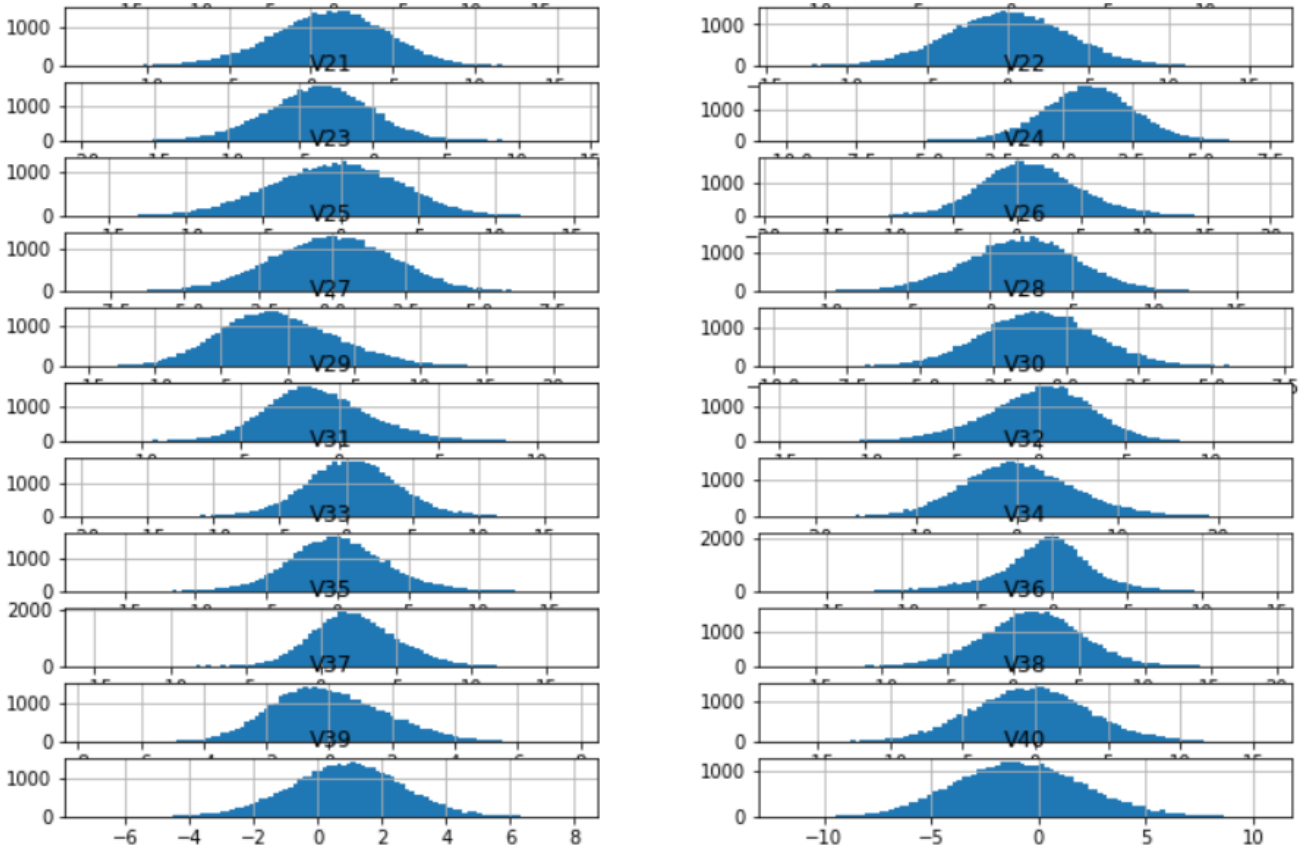
- The Train Data includes 40,000 rows and 41 columns.
- The Test Data includes 10,000 rows and 41 columns
- The Data types are 40 float and 1 integers.
  - Float: Sensor Data
  - Integer: Target
- Both datasets include missing values.
- The Dependent Variable is Target.

Data Summary

- Sensor Data: Appears to be normally distributed for all sensors.

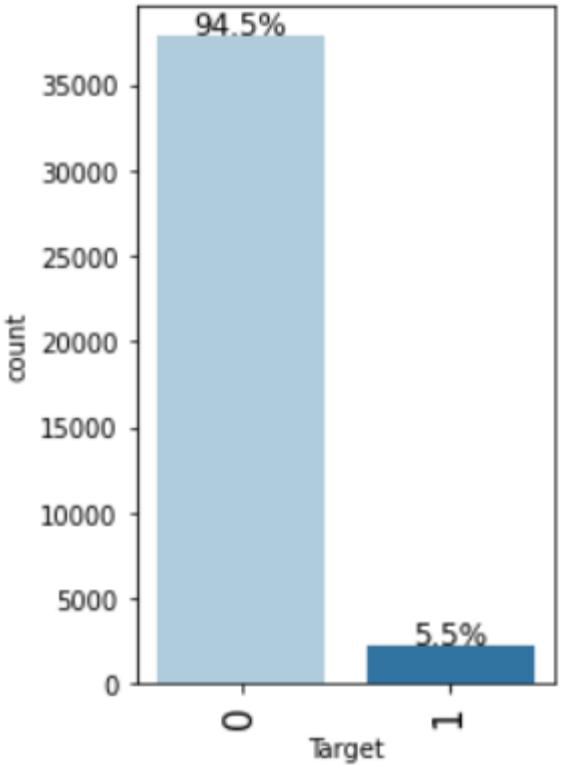


Data Summary (cont.)



Univariate Data Analysis

- Training Data:
  - Wind turbines fail 5.5% of the time.
  - 37,813 wind turbines do not require maintenance.
  - 2,187 wind turbines require maintenance.
  - Sensors V1 and V2 include missing Data
  - V1 missing 46 values
  - V2 missing 39 values



Exploratory Data Analysis Observations

- The sensor data appears to be normally distributed.
- The train data set includes null values.
- Wind turbines fail 5.5% of the time
- There are 37,813 no failures and 2,187 failures present in the training data.

Model Evaluation

3 types of cost are associated with the provided problem

- Replacement cost - False Negatives - Predicting no failure, while there will be a failure
- Inspection cost - False Positives - Predicting failure, while there is no failure
- Repair cost - True Positives - Predicting failure correctly

How to reduce the overall cost?

- We need to create a customized metric, that can help to bring down the overall cost.
- The cost associated with any model = TP 15000 + FP 5000 + FN * 40000
- And the minimum possible cost will be when, the model will be able to identify all failures, in that case, the cost will be (TP + FN) * 15000
- So, we will try to maximize Minimum cost/Cost associated with model

Model Performance

Normal Data						
Training Performance						
	Logistic Regression	Decision Tree	Random Forest	Bagging Classifier	Adaboost	Gradient Boosting
Accuracy	0.966633	1.000000	0.999967	0.997767	0.975267	0.987267
Recall	0.473301	1.000000	0.999393	0.960558	0.611044	0.782767
Precision	0.854326	1.000000	1.000000	0.998738	0.908845	0.981735
F1	0.609137	1.000000	0.999697	0.979276	0.730769	0.871033
Minimum vs Model Cost	0.520080	1.000000	0.998990	0.937962	0.599273	0.731577
Validation Performance						
	Logistic Regression	Decision Tree	Random Forest	Bagging Classifier	Adaboost	Gradient Boosting
Accuracy	0.967000	0.971300	0.988600	0.984800	0.975300	0.984600
Recall	0.480519	0.756957	0.795918	0.753247	0.615955	0.751391
Precision	0.838188	0.723404	0.990762	0.955294	0.892473	0.952941
F1	0.610849	0.739801	0.882716	0.842324	0.728869	0.840249
Minimum vs Model Cost	0.527225	0.665980	0.744818	0.702738	0.600669	0.700910

*Green represents selected for hypertuning.

Model Performance

Oversampled Data						
Training Performance						
	Logistic Regression	Decision Tree	Random Forest	Bagging Classifier	Adaboost	Gradient Boosting
Accuracy	0.867152	1.000000	1.000000	0.999012	0.899372	0.942262
Recall	0.867593	1.000000	1.000000	0.998378	0.886392	0.913939
Precision	0.866829	1.000000	1.000000	0.999647	0.910016	0.968818
F1	0.867211	1.000000	1.000000	0.999012	0.898049	0.940579
Minimum vs Model Cost	0.790447	1.000000	1.000000	0.997186	0.820639	0.867122
Validation Performance						
	Logistic Regression	Decision Tree	Random Forest	Bagging Classifier	Adaboost	Gradient Boosting
Accuracy	0.824000	0.949700	0.990600	0.981500	0.975300	0.964000
Recall	0.871985	0.803340	0.881262	0.829314	0.615955	0.897959
Precision	0.280262	0.521687	0.940594	0.827778	0.892473	0.613435
F1	0.424188	0.632579	0.909962	0.828545	0.728869	0.728916
Minimum vs Model Cost	0.510256	0.635613	0.822064	0.745161	0.600669	0.736004

*Green represents selected for hypertuning.

Model Performance

Undersampled Data						
Training Performance						
	Logistic Regression	Decision Tree	Random Forest	Bagging Classifier	Adaboost	Gradient Boosting
Accuracy	0.851942	1.000000	1.000000	0.989988	0.902306	0.947816
Recall	0.849515	1.000000	1.000000	0.981189	0.885922	0.913228
Precision	0.853659	1.000000	1.000000	0.998765	0.915935	0.981095
F1	0.851582	1.000000	1.000000	0.989899	0.900679	0.945946
Minimum vs Model Cost	0.769614	1.000000	1.000000	0.969222	0.728869	0.869198
Validation Performance						
	Logistic Regression	Decision Tree	Random Forest	Bagging Classifier	Adaboost	Gradient Boosting
Accuracy	0.866600	0.844100	0.964700	0.950800	0.975300	0.944000
Recall	0.877551	0.873840	0.905380	0.873840	0.615955	0.892393
Precision	0.271683	0.240061	0.617720	0.526257	0.892473	0.489318
F1	0.414912	0.376649	0.734387	0.656904	0.728869	0.632063
Minimum vs Model Cost	0.502955	0.468968	0.743790	0.679126	0.600669	0.671233

*Green represents selected for hypertuning.

Model Performance – Hypertuned Models

The models with the highest Minium_Vs_Model_cost scores for the validation data sets were selected for hypertuning:

- Random Forest OverSampling
- Bagging Classifier Oversampling
- Random Forest Normal

Model Performance – Hypertuned Models

Random Forest with normal data and without hypertuning is the best model and will be selected as the final model and used with the Test.csv dataset.

Hypertuned Models			
Training Performance			
	Random Forest - Oversampled Data	Bagging Classifier - Oversampled Data	Random Forest - Normal Data
Accuracy	0.910694	0.999929	0.965467
Recall	0.888332	0.999859	0.373786
Precision	0.929922	1.000000	0.993548
F1	0.908651	0.999929	0.543210
Minimum vs Model Cost	0.827522	0.999765	0.489118
Validation Performance			
	Random Forest - Oversampled Data	Bagging Classifier - Oversampled Data	Random Forest - Normal Data
Accuracy	0.928700	0.986700	0.963900
Recall	0.871985	0.871985	0.335807
Precision	0.421903	0.880150	0.983696
F1	0.568663	0.876048	0.500692
Minimum vs Model Cost	0.620491	0.798124	0.474194

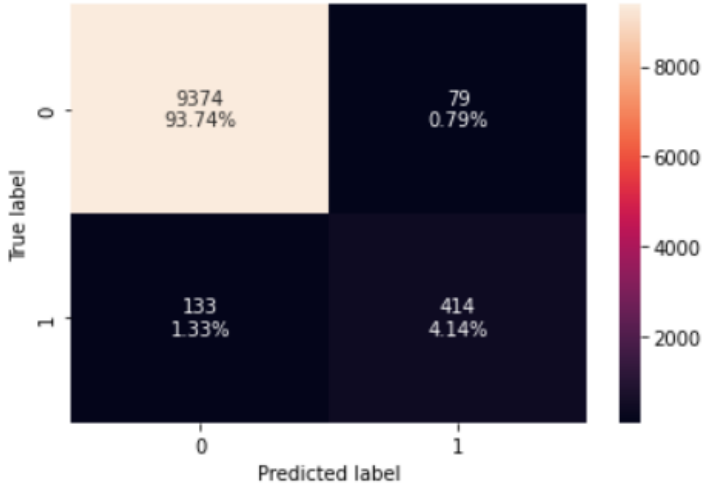
Final Model Performance Summary

Observations

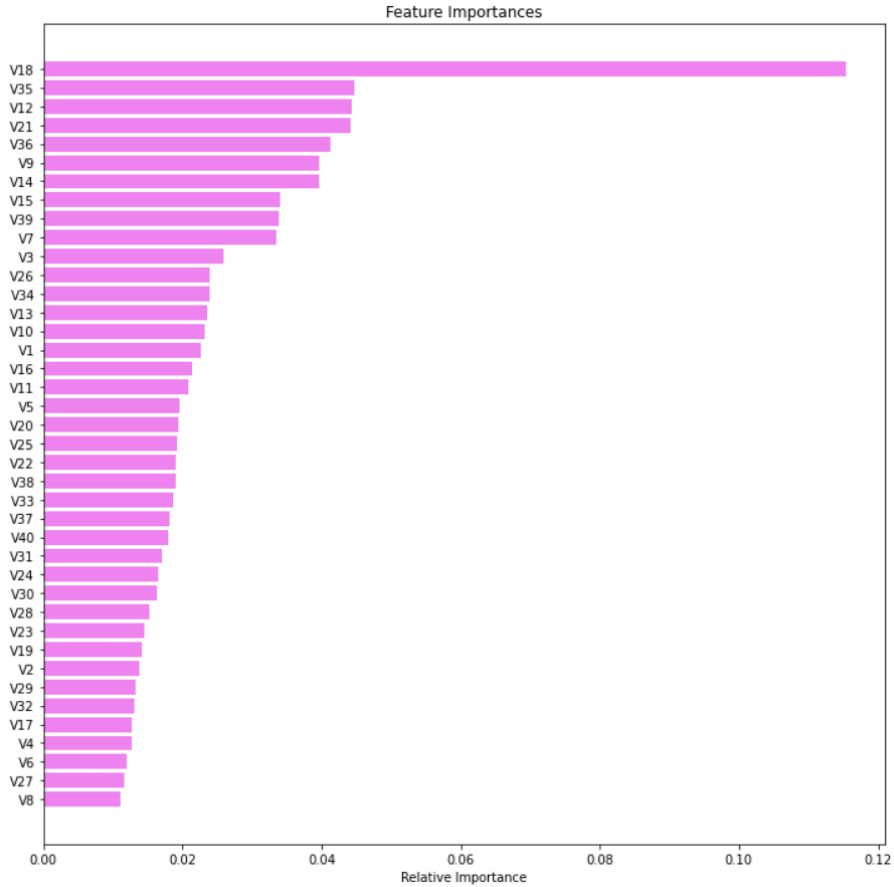
- The model does not perform well on the test data.
- The results reveal further hypertuning needs to be conducted prior to implementing into production as a predictor to decrease maintenance cost.
- Further review reveals Bagging Classifier with Oversampled Data should have been selected for the final model.

Testing performance:

	Accuracy	Recall	Precision	F1	Minimum_Vs_Model_cost
0	0.9034	0.061224	0.066937	0.063953	0.350988



Model Feature Importance



Business Insights and Recommendations

- The model needs to continue to be hypertuned before being implemented into production.
- The hypertuned Bagging Classifier trained with oversampled data should be tested to determine if it is viable to be placed into production to accurately predict maintenance needed to decrease costs.
- Sensor V18 is the sensor to focus on when predicting maintenance costs.
- Sensors V35, V12, V21, V36, V9, and V14 should all be further investigated as their feature importances are all over .04.

greatlearning
Power Ahead

Happy Learning !