ReneWind - Model Tuning

Michail Mersinias

05/25/2023

Contents / Agenda

- Executive Summary
- Business Problem Overview and Solution Approach
- EDA Results
- Data Preprocessing
- Model Performance Summary
- Appendix

Executive Summary

- The executive summary (insights and recommendations) of the ReneWind Model Tuning presentation is as follows:
  - **Model of Choice:** It is possible to minimize the total maintenance cost of machinery/processes used for wind energy production using a machine learning model. The final tuned model (XGBoost) was chosen after building ~7 different machine learning algorithms & further optimizing for target class imbalance (having few "failures" and many "no failures" in dataset) as well as fine-tuning the algorithm (hyperparameter and cross validation techniques).
  - **Deployment:** A pipeline was additionally built to productionize the final chosen model.
  - **The Main Culprits:** The top 5 attributes of importance for predicting failures vs. no failures were found to be V36, V14, V26, V16 and V18 in order of decreasing importance. This added knowledge can be used to refine the process of collecting more frequent sensor information to be used in improving the machine learning model to further decrease maintenance costs.

Business Problem Overview and Solution Approach

- Problem Definition:
  - 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.
  - The objective 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.
- Solution Approach and Methodology:
  - For EDA, univariate analysis will be performed to find insights.
  - For Data Preprocessing, missing value imputation and feature engineering will be performed.
  - For Model Building, we will use the following classification models: Gradient Boosting, AdaBoost, Random Forest, XGBoost. We will perform oversampling and undersampling techniques to deal with imbalanced data.

EDA Results: Data Overview

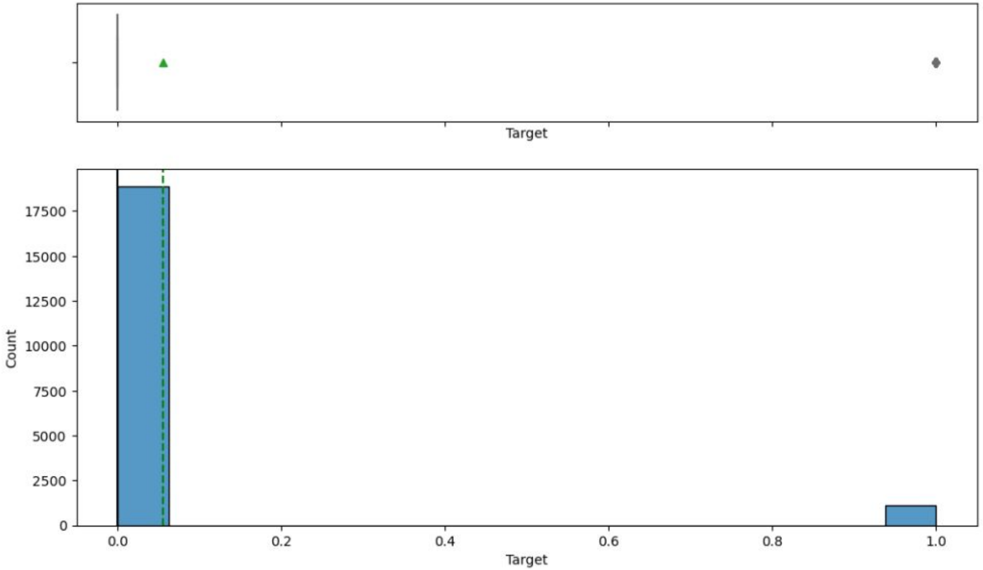
- The dataset is comprised of 25000 rows and 41 columns.
- The columns are as follows: V1, V2, V3, ... , V38, V39, V40, Target.
- All rows represent sensor measurements.
- There are no duplicate values in the dataset.
- There is only a small percentage of missing values, specifically 0.1% for V1 and 0.1% for V2.

EDA Results: Univariate Analysis

- In this section, we perform univariate analysis on the data.
  - For each attribute, we perform a descriptive statistical analysis where only that attribute is involved as a variable.
  - We also analyze the corresponding data and present the distribution of the attribute, with confidence intervals to signify variance and statistical significance.
  - Finally, we write the conclusion based on both quantitative and qualitative observations.

EDA Results: Univariate Analysis

- Analyzing the Target attribute, we report that the vast majority is 0 (no failure) but a small percentage is 1 (failure - needs repair). Thus, the dataset is largely imbalanced. A countplot is presented to depict this more clearly.



EDA Results: Univariate Analysis

- Analyzing the V1, V2, V3, ..., V38, V39, V40 attributes, we report that they all follow a normal distribution with the mean being approximately at zero and the standard deviation being approximately at 3.
- Mean Values: -0.272, 0.440, 2.485, -0.083, -0.054, -0.995, -0.879, -0.548, -0.017, -0.013, -1.895, 1.605, 1.580, -0.951, -2.415, -2.925, -0.134, 1.189, 1.182, 0.024, -3.611, 0.952, -0.366, 1.134, -0.002, 1.874, -0.612, -0.883, -0.986, -0.016, 0.487, 0.304, 0.050, -0.463, 2.230, 1.515, 0.011, -0.344, 0.891, -0.876 => **-0.0328 (Average)**
- Standard Deviation: 3.442, 3.151, 3.389, 3.432, 2.105, 2.041, 1.762, 3.296, 2.161, 2.193, 3.124, 2.930, 2.875, 1.790, 3.355, 4.222, 3.345, 2.592, 3.397, 3.669, 3.568, 1.652, 4.032, 3.912, 2.017, 3.435, 4.369, 1.918, 2.684, 3.005, 3.461, 5.500, 3.575, 3.184, 2.937, 3.801, 1.788, 3.948, 1.753, 3.012 => **2.9827 (Average)**

Data Preprocessing

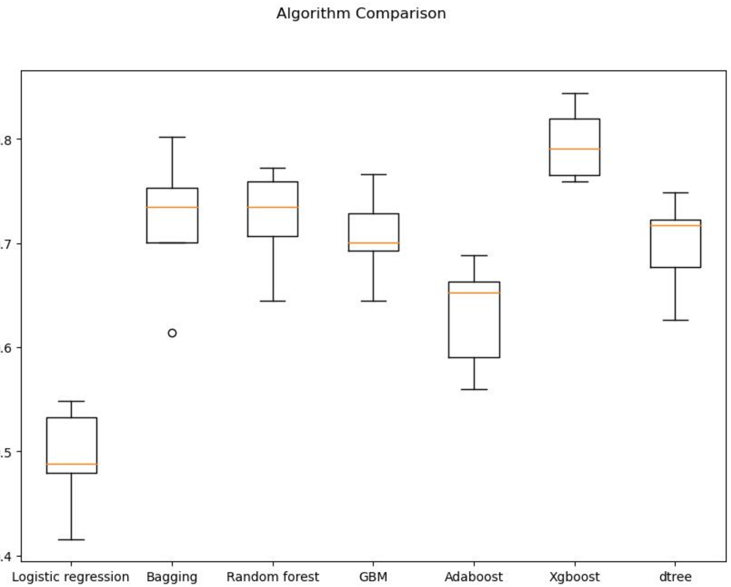
- Outlier Check: There are very few outliers and all of our features follow a normal distribution.
- Missing Value Imputation: We imputed the few missing values for the V1 and V2 features with the median values of those features. As a result, our data no longer has any missing values.
- Preparing Data for Modeling: We use Target as our target variable which we seek to predict accurately, and all the other columns as features to train our model.
- Train/Test Dataset Split: We split the data in 75:25 ratio for train to test data.

Model Performance Summary

- Vanilla Modeling Setup:
  - We use the following classification models: Linear Regression, Bagging, Gradient Boosting, AdaBoost, Random Forest, XGBoost, Decision Tree.
  - For each one of these models, we use GridSearchCV with cv = 5, in order to find the most suitable model parameters based on the training data.
  - Finally, we define functions for the following metrics of performance: Recall (we want to minimize false negatives, thus we aim to maximize recall).

Model Performance Summary

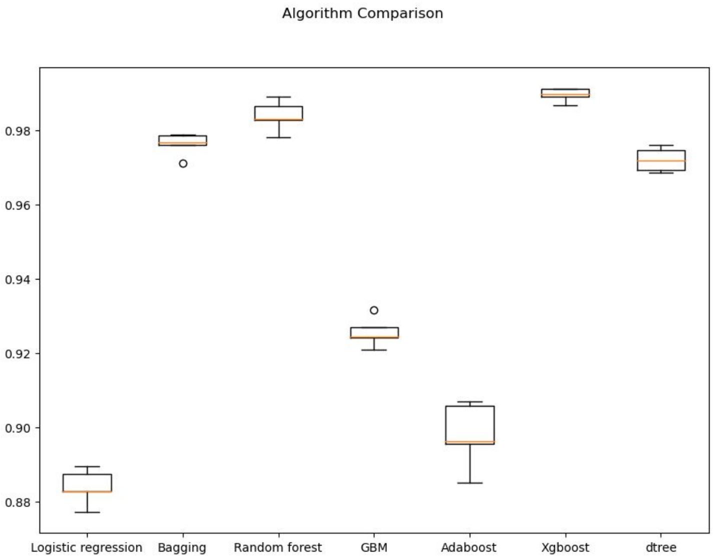
- The cross-validation (cv=5) results on the original validation data are the following:



- Therefore, for the original data, a XGBoost Classifier offers the highest performance. For the test data, that performance is also the highest with 82.01%.

Model Performance Summary

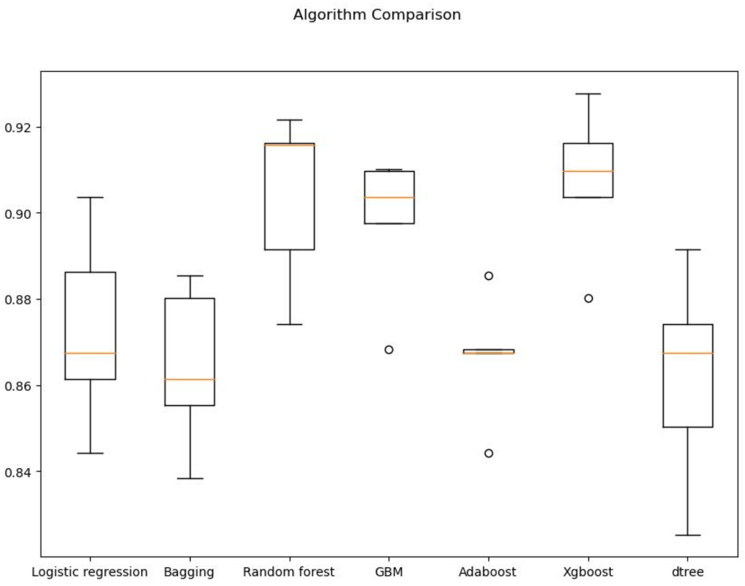
- The cross-validation (cv=5) results on the oversampled validation data are the following:



- Therefore, for the oversampled data, a XGBoost and a Random Forest Classifier offers the highest performance. For the test data, that performance is also the highest with 86.69% and 85.61% respectively.

Model Performance Summary

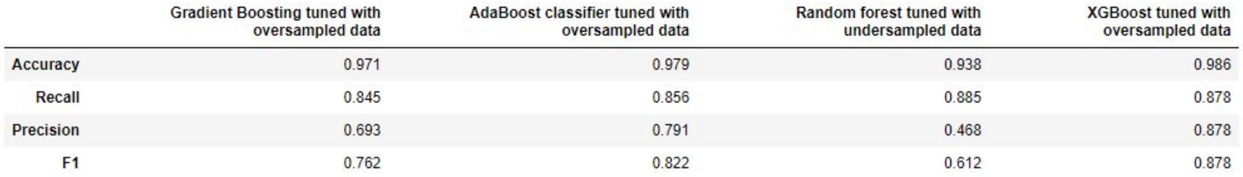
- The cross-validation (cv=5) results on the undersampled validation data are the following:



- Therefore, for the undersampled data, a XGBoost and a Random Forest Classifier offers the highest performance. For the test data, that performance is also the highest with 90.28% and 89.20% respectively.

Model Performance Summary

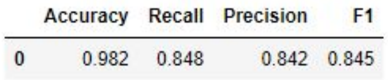
- Tuned Modeling Setup:
  - We use the following classification models: Gradient Boosting, AdaBoost, Random Forest, XGBoost.
  - For each one of these models, we use GridSearchCV with cv = 5, in order to find the most suitable model parameters based on the training data.
  - Finally, we define functions for the following metrics of performance: Accuracy, Precision, Recall and F1 Score.
- The results on the validation data are the following:



	Gradient Boosting tuned with oversampled data	AdaBoost classifier tuned with oversampled data	Random forest tuned with undersampled data	XGBoost tuned with oversampled data
Accuracy	0.971	0.979	0.938	0.986
Recall	0.845	0.856	0.885	0.878
Precision	0.693	0.791	0.468	0.878
F1	0.762	0.822	0.612	0.878

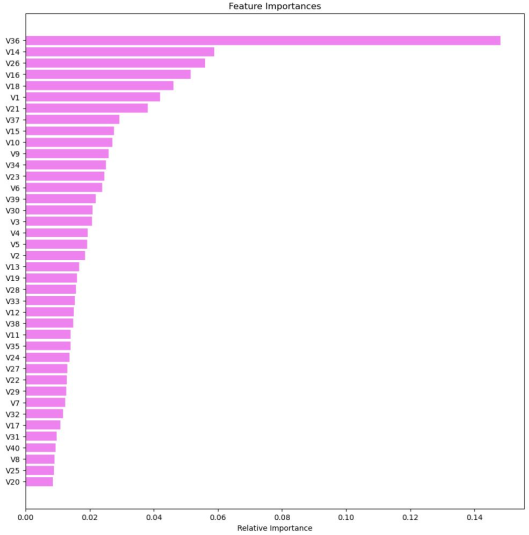
Model Performance Summary

- Thus, we pick the tuned XGBoost Classifier using oversampled data as our final model of choice as it achieves the highest performance of 87.80% in the validation set. For the test set, the performance is as follows:



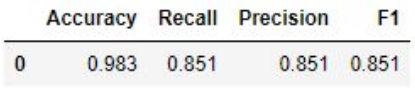
	Accuracy	Recall	Precision	F1
0	0.982	0.848	0.842	0.845

- The feature importance list is as follows:



Model Performance Summary

- Using Pipeline to deploy the tuned XGBoost Classifier using oversampled data as our final model of choice, it achieves a slightly higher performance on the test set as follows:



	Accuracy	Recall	Precision	F1
0	0.983	0.851	0.851	0.851

- Although the performance is high, using undersampled data with the tuned XGBoost Classifier could potentially yield an even higher performance. However, ReneWind instructed us to test specific combinations of models and sampling techniques and from those, the tuned XGBoost Classifier using oversampled data yields the best results.

Executive Summary

- The executive summary (insights and recommendations) of the ReneWind Model Tuning presentation is as follows:
    - **Model of Choice:** It is possible to minimize the total maintenance cost of machinery/processes used for wind energy production using a machine learning model. The final tuned model (XGBoost) was chosen after building ~7 different machine learning algorithms & further optimizing for target class imbalance (having few "failures" and many "no failures" in dataset) as well as fine-tuning the algorithm (hyperparameter and cross validation techniques).
    - **Deployment:** A pipeline was additionally built to productionize the final chosen model.
    - **The Main Culprits:** The top 5 attributes of importance for predicting failures vs. no failures were found to be V36, V14, V26, V16 and V18 in order of decreasing importance. This added knowledge can be used to refine the process of collecting more frequent sensor information to be used in improving the machine learning model to further decrease maintenance costs.

Thank you for your time!

Michail Mersinias

05/25/2023