RENEWIND

PROJECT 6 – MODEL TUNING

Joshua Willis, PGP-DSBA

6/10/2022

Agenda

- Executive Summary
- Business Problem Overview & Solution Approach
- EDA Results
- Data Preprocessing
- Model performance summary for hyperparameter tuning
- Model building with pipeline

Executive Summary

Key Insights:

- The best model to help identify failures so that generators can be repaired before failing and help reduce cost is XGBoost.
- The top three (3) important features are V36, V26, and V18
- The XGBoost model will identify true positives with a recall score of 0.87

Business Recommendations:

- Research top important features to see if they can be adjusted to reduce likelihood of having to repair generators
- Implement best practices around minimizing costs when repairs are made
- Implement lower cost inspection of generators
- Identify patterns of generator failure and implement quality control process to address accordingly

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. Wind energy is one of the most developed technologies worldwide. The US Department of Energy has put together a guide for achieving operational efficiency using predictive maintenance practices.

Predictive maintenance uses sensor information and analysis methods to measure and predict future component capability. This concept predicts failure patterns; if component failures can be proactively identified accurately, they can be replaced before it fails, thus reducing operation and maintenance costs.

Objective: is to build various classification models, tune them, and find the best one that will help identify failures so that the generators can be repaired before failing in order to reduce the overall maintenance cost.

- Training Data has 20,000 rows and 41 columns
- Testing Data has 5,000 rows and 41 columns
- Both data sets are in float format with target column in integer format
- There are no duplications in either data sets
- There are missing values in both data sets
- The data in both data sets are normally distributed with outliers

EDA Results

- Data was split into train, validation, and testing prior to imputation to prevent data leakage (prevent test data from influencing the training data)
- Impute missing values (with median) into training and testing data sets accordingly
- No duplicates need to be addressed
- No outliers need to be treated since there are no extremes that would severely skew modeling
- No feature engineering done; all features included for analysis

Data Preprocessing

Model Baselines:

- True Positives: correct predictions resulting in repair costs
- False Negative: costly because we repair a component that doesn't need repair
- False Positive: Less costly than False negative; will inspect a component that is believed to be faulty
- Recall metric will be used; higher recall and higher chances of minimizing false negatives

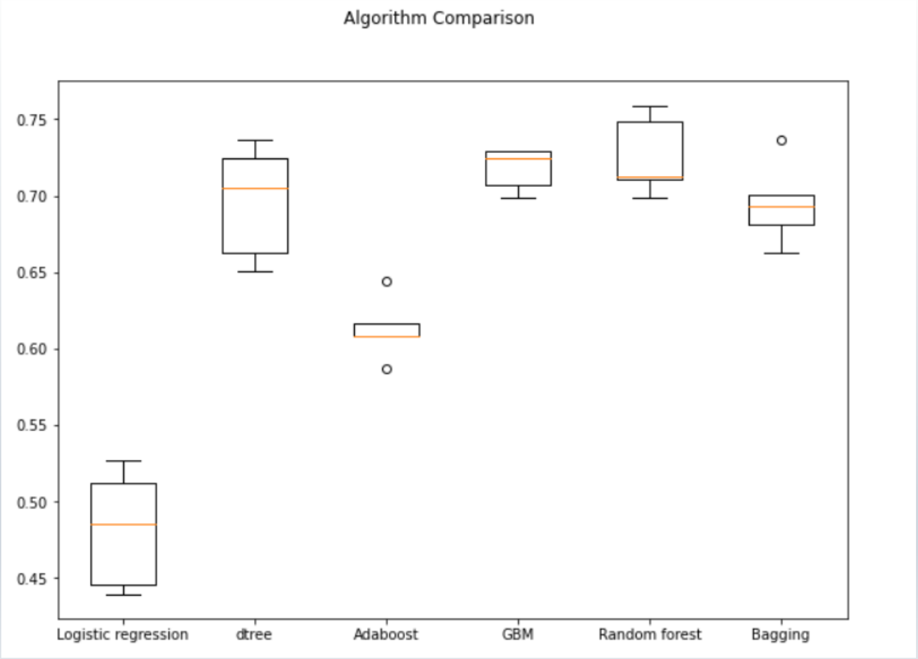
Models considered:

- Logistics Regression, Decision Tree, Adaboost, Gradient Boosting, Random Forest, and Bagging

Model Performance Summary

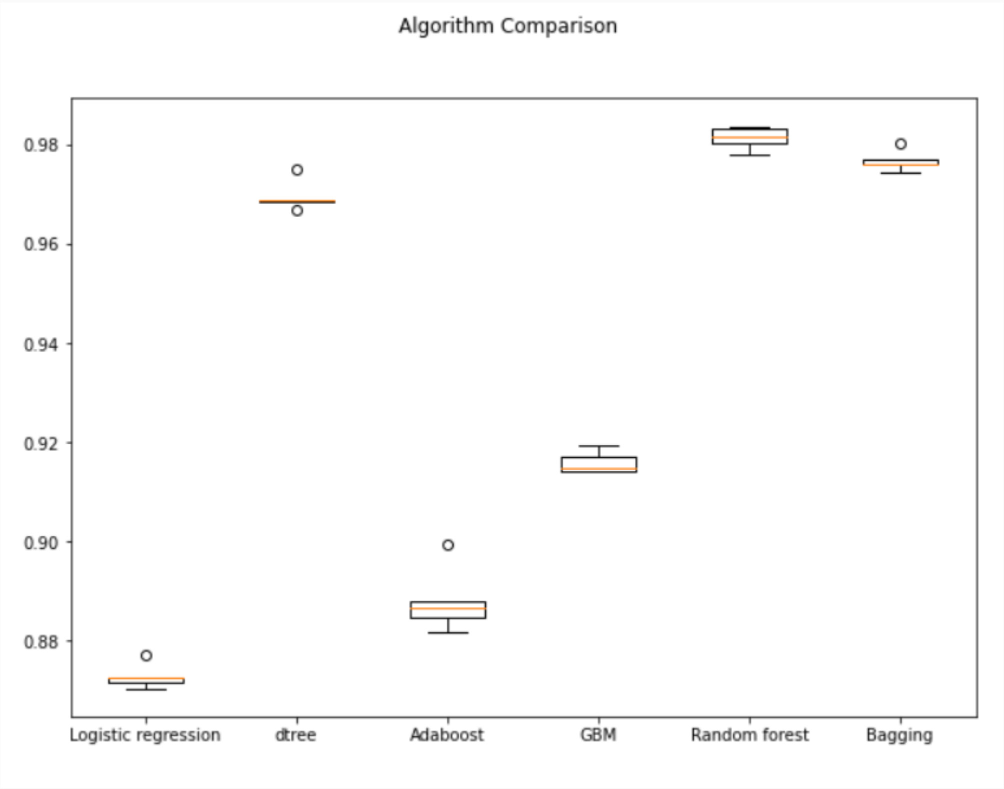
Initial Model Observations:

- Decision Tree, Gradient Boost, Random Forest, and Bagging are best models based upon initial training data

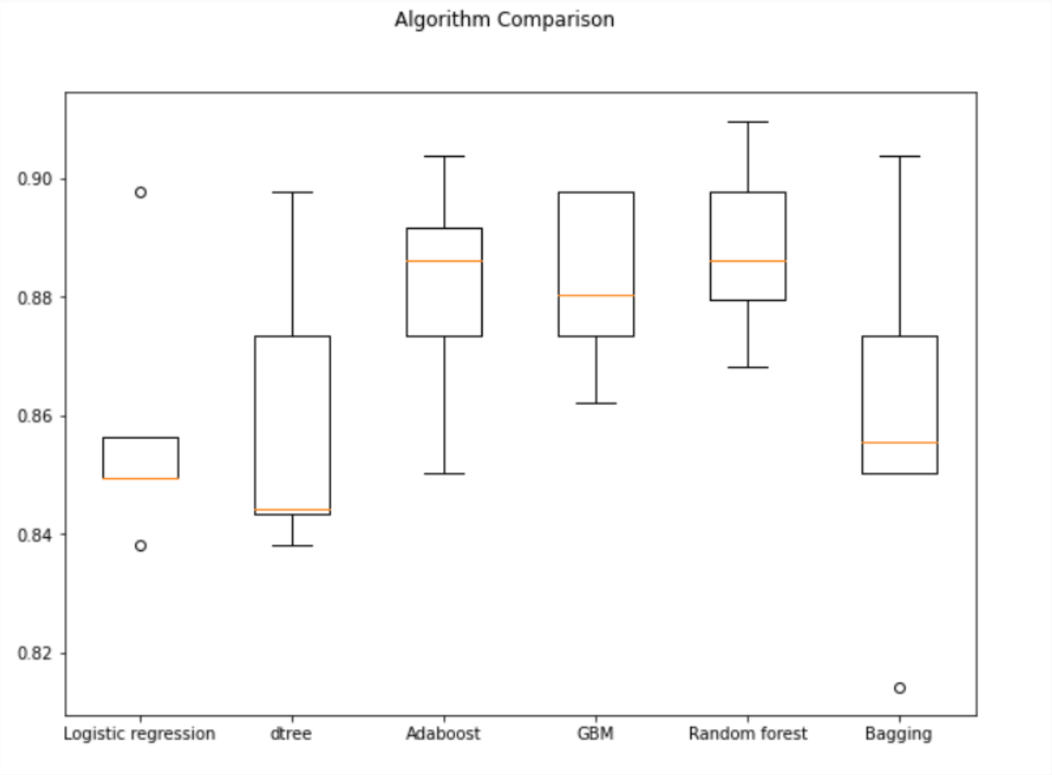


- Since the data is imbalanced (biased towards target value of “0”) oversampling and undersampling analysis was conducted.

Model Performance Summary



Oversampling



Undersampling

Model Performance Summary

Model Performance Summary

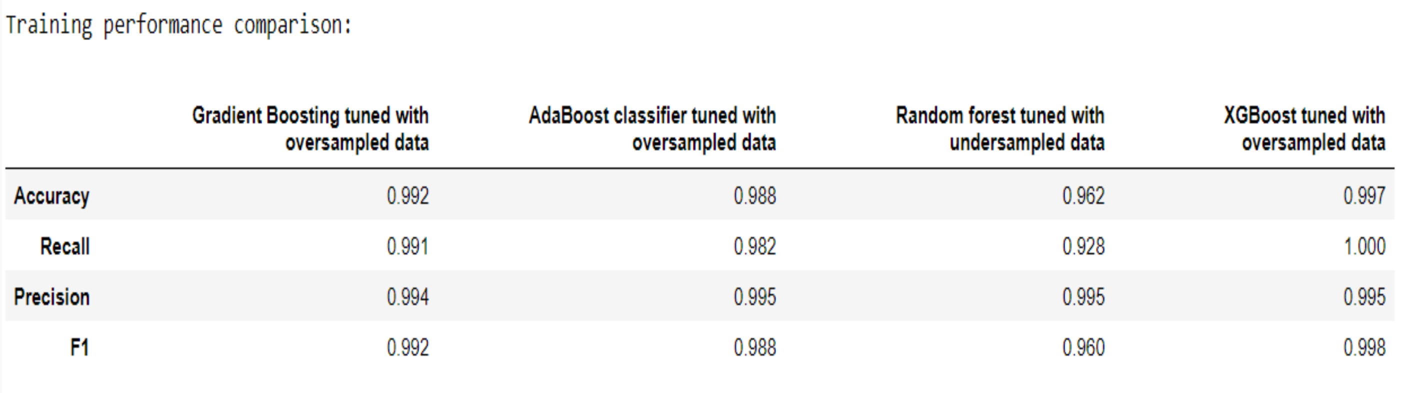
After reviewing all the models, the best performers were:

- AdaBoost with oversampled data
- Random Forest with undersampled data
- Gradient Boosting with oversampled data
- XGBoost with oversampled data

The hyperparameters were tuned for the above models and scored accordingly.

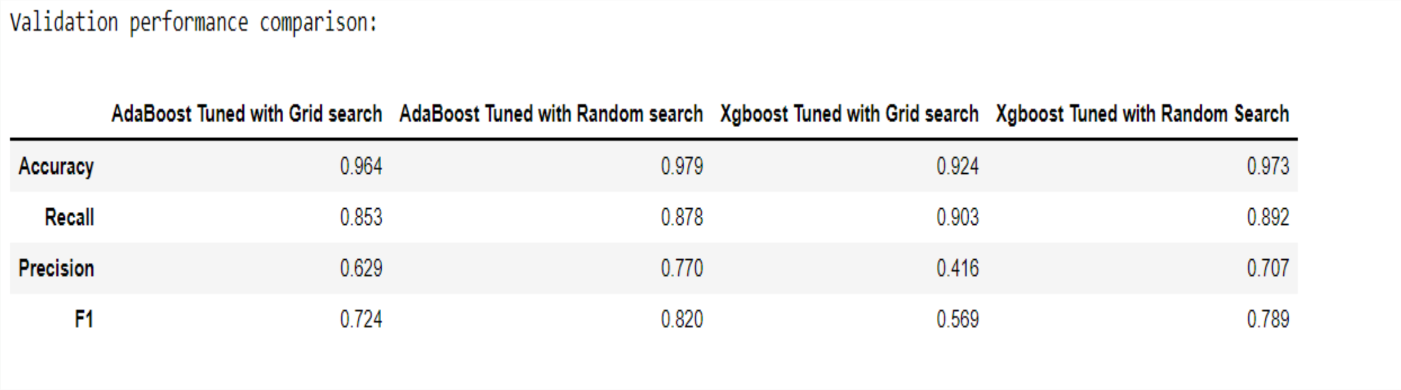
Model Performance Summary

Tuned Model Performance Comparison:



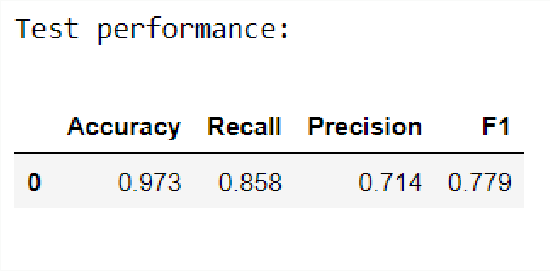
Training performance comparison:

	Gradient Boosting tuned with oversampled data	AdaBoost classifier tuned with oversampled data	Random forest tuned with undersampled data	XGBoost tuned with oversampled data
Accuracy	0.992	0.988	0.962	0.997
Recall	0.991	0.982	0.928	1.000
Precision	0.994	0.995	0.995	0.995
F1	0.992	0.988	0.960	0.998



Validation performance comparison:

	AdaBoost Tuned with Grid search	AdaBoost Tuned with Random search	Xgboost Tuned with Grid search	Xgboost Tuned with Random Search
Accuracy	0.964	0.979	0.924	0.973
Recall	0.853	0.878	0.903	0.892
Precision	0.629	0.770	0.416	0.707
F1	0.724	0.820	0.569	0.789



Test performance:

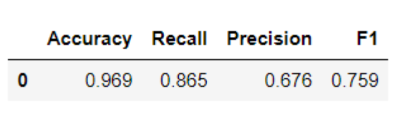
	Accuracy	Recall	Precision	F1
0	0.973	0.858	0.714	0.779

- XGBoost Model performed best with recall score during training and validation.
- XGBoost Model used for test data and has recall score of 0.858

Pipeline Building Process

- Create pipeline with best performance model XGBooster
- No need for column transformer; all data is in the same format
- Separate target from other variables
- Impute missing values into the test set
- Use oversampled data to fit the model
- Use test data to check final performance

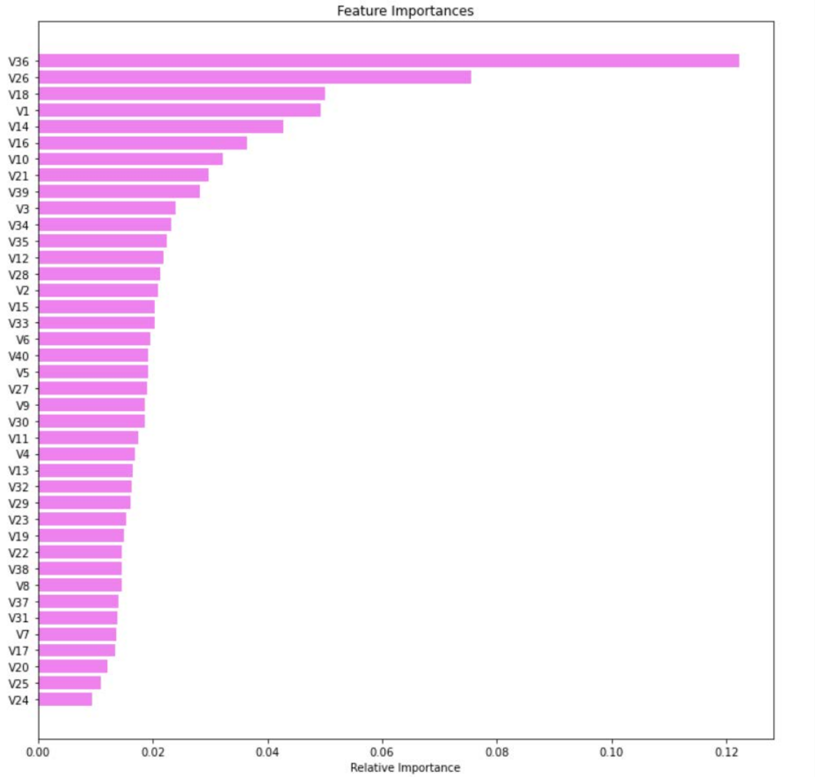
Pipeline Final Model Score



	Accuracy	Recall	Precision	F1
0	0.969	0.865	0.676	0.759

- Pipeline Final Model recall score is 0.865
- Important features are V36, V26, and V18

Feature Importances



Model Performance Summary

THANK YOU