Renewind Data Analysis

Python Foundations : PGP-DSBA

May 5th ,2023

Contents / Agenda

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

Executive Summary

- ReneWind is a renewable energy firm that uses machine learning to improve the efficiency of wind energy generation. They have gathered sensitive sensor data on wind turbine generator failures with the goal of developing a classification model to predict failures and cut maintenance costs. The data set contains 40 predictors, 20000 observations in the training set, and 5000 observations in the test set.

- The goal is to create a classification model that can accurately forecast generator failures while minimizing repair, replacement, and inspection expenses. The model will be evaluated using true positive and false positive rates, with the goal of maximizing true positive rate while minimizing false positive rate. For the sake of secrecy, the data was transformed, and the target variable is binary, with 1 signifying a failure and 0 indicating no failure.

- The company will use predictive maintenance practices to predict generator failure patterns and replace components before they fail, resulting in lower maintenance costs. The company plans to tune various classification models to find the best one that can accurately identify failures. The cost of repairing a generator is much lower than replacing it, and inspection costs are lower than repair costs. Therefore, the model should focus on minimizing the number of false negatives (real failures not detected by the model), while keeping false positives to a minimum to reduce inspection costs.

Business Problem Overview and Solution Approach

- **Problem Definition**: The problem is to develop a classification model using machine learning techniques that can accurately predict generator failures in wind turbines, based on sensor data. The aim is to reduce maintenance costs by using predictive maintenance practices to predict failure patterns and replace components before they fail. The target is to minimize false negatives (real failures not detected by the model) while keeping false positives to a minimum to reduce inspection costs.

- **Solution Approa**ch : The solution approach to the problem of predicting wind turbine generator failures involves several steps. Firstly, the data will be preprocessed to remove any missing values, outliers or redundant variables, and feature engineering will be used to extract relevant features from the raw sensor data. Secondly, various classification models such as Logistic Regression, Decision Trees, Random Forest and Gradient Boosting will be evaluated using metrics such as accuracy, precision, recall, and F1-score. The models will be trained on the training set and evaluated on the test set to select the best performing model. Thirdly, the best performing model will be further tuned using hyperparameter tuning techniques such as Grid Search, Random Search or Bayesian Optimization to optimize the model's performance on the test set. Fourthly, the final model will be evaluated on the test set, with a focus on minimizing false negatives while keeping false positives to a minimum. Finally, the selected model will be deployed in production, integrated with the company's predictive maintenance system to predict generator failures in real-time and help reduce maintenance costs while improving wind energy production efficiency. Overall, the solution approach is a comprehensive and iterative process that combines data preprocessing, model selection, tuning, evaluation and deployment to achieve the objective of reducing maintenance costs through predictive maintenance practices.

Data Overview

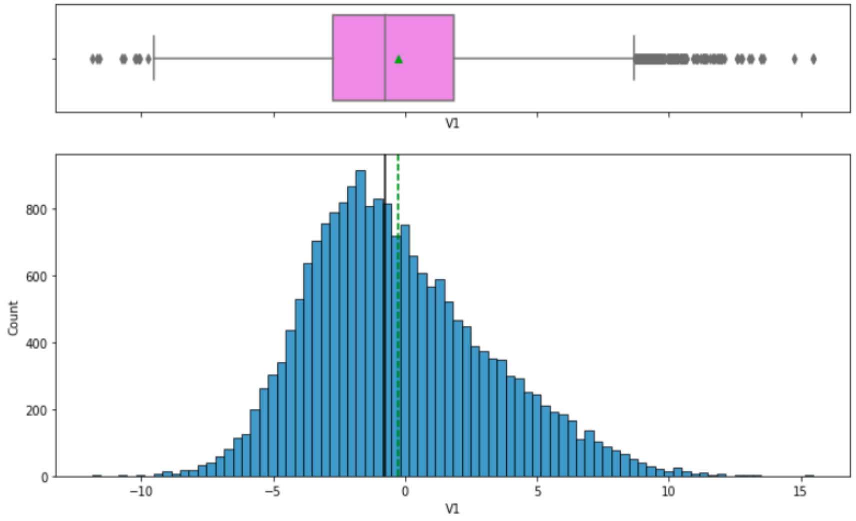
- The data is divided into two, the train and test set
- The train set consist of 20,000 rows, 41 columns
- The test set consist of 5,000 rows, 41 columns
- There were 0 duplicated values
- There are 40 columns with data type float64 and 1 column with data type int64.

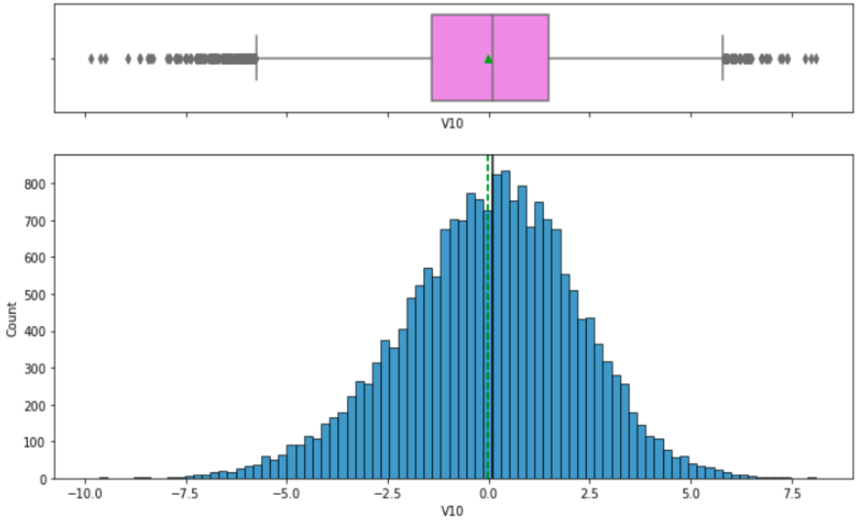
EDA Results

- *The average V1 predictor variable is greater than the median, indicating that the distribution is skewed to the right.*

- *There are outliers in the distribution of the V1 predictor variable, which ranges from -10 to 15.*

- *The average and median V10 predictors are nearly identical, indicating that the median is roughly symmetrical.*
- *The V10 predictor variable has a very uniform distribution between -7.5 and 7.5.*
- *The distribution contains outliers.*

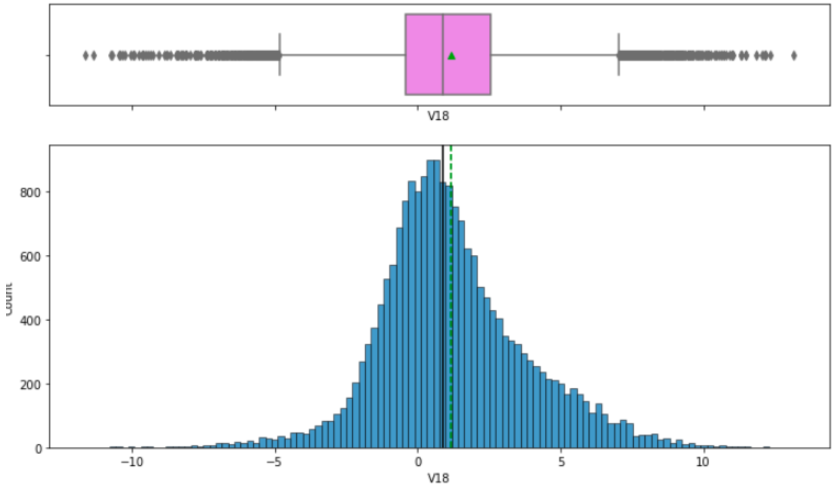


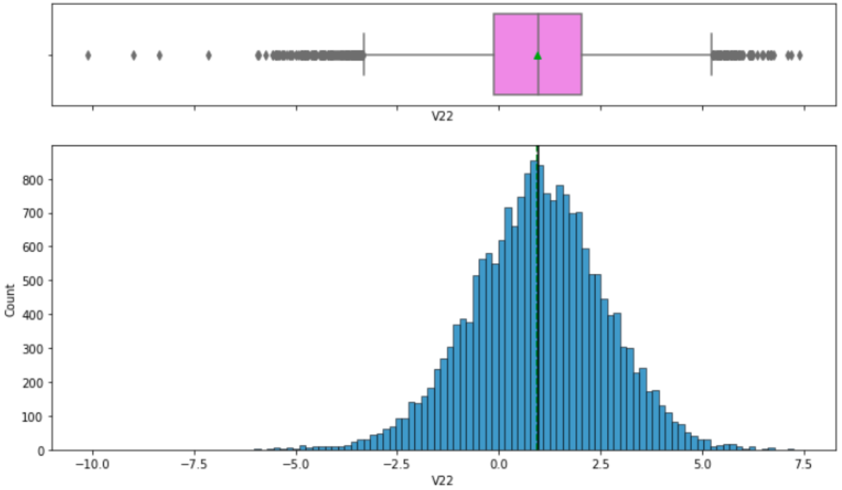


EDA Results

- *The average V18 predictor variable is greater than the median, indicating that the distribution is skewed to the right.*
- *There are outliers in the distribution of the V1 predictor variable, which is approximately uniformly distributed between -10 and 10.*

- *The average and median V22 predictors are the same, indicating that the median is symmetrical and that there are outliers in the distribution.*

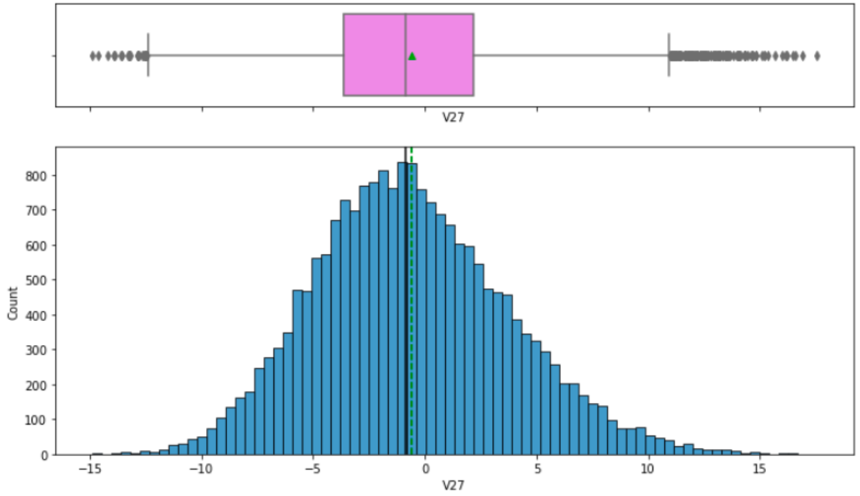


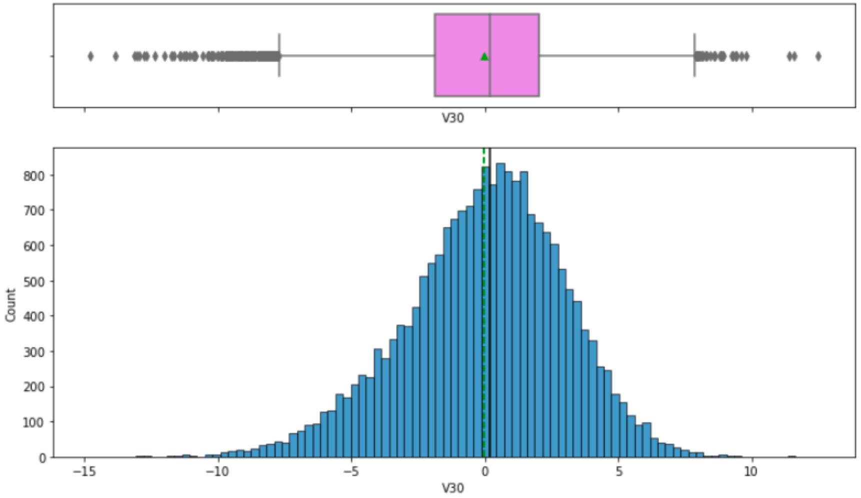


EDA Results

- *The average V27 predictor variable is greater than the median, indicating that the distribution is slightly biased to the right and that there are outliers in the distribution.*

- *The average V30 predictor variable is lower than the median, indicating that the distribution is biased to the left and that there are outliers in the distribution.*

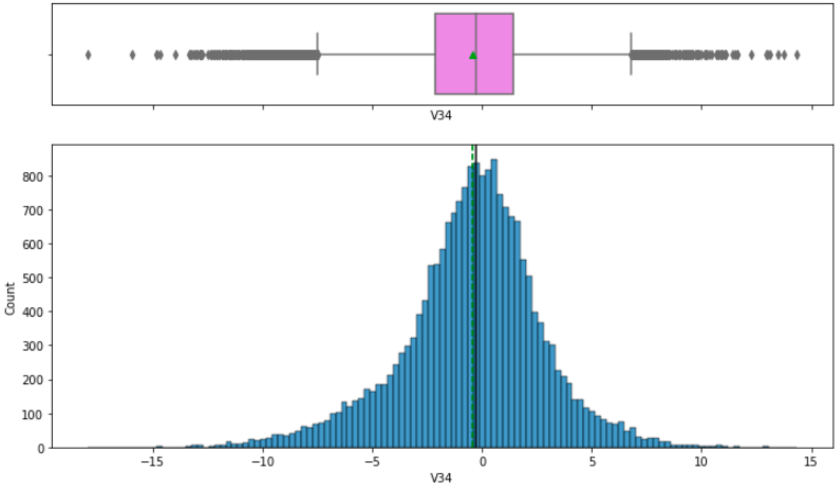


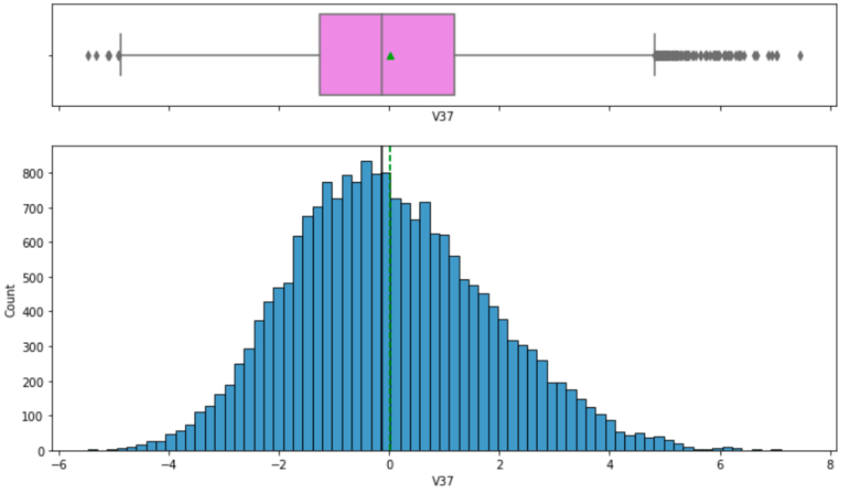


EDA Results

- *The average V34 predictor variable is approximately identical to the median, indicating that the distribution is roughly symmetrical.*
- *There are outliers in the distribution of the V1 predictor variable, which is fairly uniformly distributed between -7.5 and 7.5.*

- *The average V37 predictor variable is greater than the median, indicating that the distribution is biased to the right and that there are outliers in the distribution.*

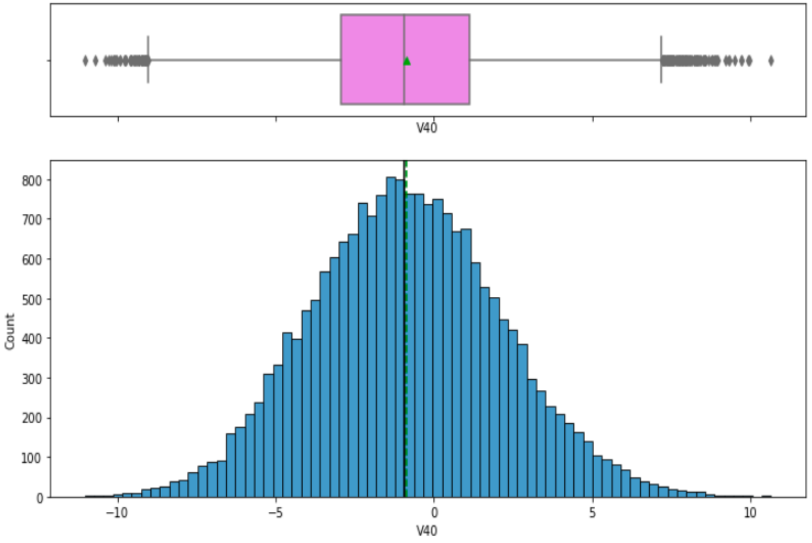


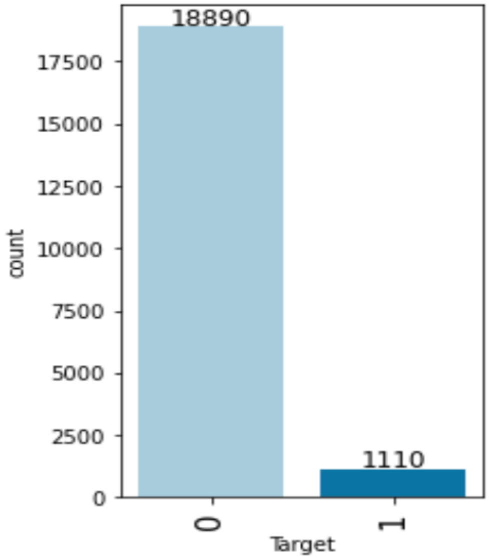


EDA Results

- *The average V40 predictor variable is approximately identical to the median, indicating that the distribution is roughly symmetrical.*
- *There are outliers in the distribution of the V40 predictor variable, which is fairly uniformly distributed between -10 and 10.*

- *The target variable indicates that 18890 predictors are in good condition.*
- *The distribution predicts that 1110 predictors will require repair.*





Data Preprocessing

- There are no duplicates in the data set
- There are missing value in both the train and test data indication predictor V1 and V2 with missing value. we handled the missing values using the SimpleImputer class from the scikit-learn library. We created an instance of the SimpleImputer class and set the strategy to "median". Then, we fit the imputer on the training set using the fit_transform() method to both fit the imputer on the training set and transform the training set at the same time. We then transformed the validation and test sets using the transform() method to impute missing values without data leakage. Finally, we checked that there were no missing values left in any of the sets using the isna() and sum() methods.
- The data preparation will be used 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

Model Performance Summary

- To create several categorization models, adjust them, and discover the optimal one for identifying faults so that generators can be fixed before they fail/break, reducing total maintenance costs.
- Recall will be used as performance metric of evaluation,
  - A high recall indicates that the model successfully predicted failures. These will incur repair costs.
  - The lower the recall, the more real failures that the model does not detect. These will incur replacement charges.
  - False positives (FP) are detections where there is no failure. These will result in inspection costs.
- the higher the Recall, the more the chances of lowering false negatives.
- The most significant predictors variable to recognize failure are:
  - V30
  - V9
  - V18
  - V12
  - V36
  - V3

Model Performance Summary

MODEL	TRAIN ACCURACY	VALIDATION ACCURACY	TRAIN RECALL	VALIDATION RECALL	TRAIN PRECISION	VALIDATION PRECISION	TRAIN F1	VALIDATION F1
Gradient Boosting tuned with oversampled data	0.99	0.97	0.99	0.84	0.99	0.69	0.99	0.76
AdaBoost classifier tuned with oversampled data	0.99	0.97	0.98	0.85	0.99	0.79	0.99	0.82
Random forest tuned with undersampled data	0.96	0.93	0.93	0.88	0.98	0.46	0.96	0.61

Productionize and test the final model using pipelines

Steps taken to create a pipeline for the final model

- Create a pipeline using the best model, which was optimized using an adaboost classifier and oversampled data.
- Then, divide the target variable and other variables into independent variables and target variables for train data.
- We can't oversample/undersample data without first treating missing values in the train and test sets, therefore we have to start there.
- Oversample/undersample the train data and, if necessary, generate variables for them.
- Train data was used to fit the model.
- Examine the performance on the test set

The performance of the model built with pipeline on the test dataset

- The test recall is 79%, which is poor when compared to the precision of 98%; this cannot fully forecast failures and can lead to cost-cutting. This is not a good model for predicting failures.

The most important factors used by the model built with pipeline for prediction

- V30
- V9
- V18.

Model Performance Summary (original data)

- As we can see, xgboost has the highest cross-validated recall, followed by random forest and bagging.
- The boxplot indicates that the performance of xgboost, random forest, and bagging is consistent, and that it is also good on the validation set.

Model	Cross Validation Test	Validation Performance
Logistics regresssion	0.49	0.48
Bagging	0.72	0.73
Random forest	0.72	0.72
Gradient boosting	0.70	0.72
Adabosst	0.63	0.67
Xgboost	0.79	0.82

Model Performance Summary (oversampled data)

- For the gradient boost and ada boost classifiers, we used the random search cv oversampling method.
- In both the cross validation and validation sets, the gradient boost and adaboost classifiers did not have the highest recall.
- After hypertuning the data, overall performance improved, making it a superior option for forecasting failures with the highest recall.

MODEL	TRAIN ACCURACY	VALIDATION ACCURACY	TRAIN RECALL	VALIDATION RECALL	TRAIN PRECISION	VALIDATION PRECISION	TRAIN F1	VALIDATION F1
Gradient Boosting tuned with oversampled data	0.99	0.97	0.99	0.84	0.99	0.69	0.99	0.76
AdaBoost classifier tuned with oversampled data	0.99	0.97	0.98	0.85	0.99	0.79	0.99	0.82

Model Performance Summary (undersampled data)

- For the random forest under sampling approach, we utilized the random search CV.
- In the original data, the random forest performed well in terms of both cross validation cost and validation performance.
- The training performance improved once the data was hyper tuned, however the validation precision was very low, despite having the highest recall.

MODEL	TRAIN ACCURACY	VALIDATION ACCURACY	TRAIN RECALL	VALIDATION RECALL	TRAIN PRECISION	VALIDATION PRECISION	TRAIN F1	VALIDATION F1
Random forest tuned with undersampled data	0.96	0.93	0.93	0.88	0.98	0.46	0.96	0.61

Business Insights and Conclusion

Business Insights

- From the results, it appears that the AdaBoost classifier tuned with oversampled data performed the best in terms of validation recall. This model could be a good option for predicting failures and minimizing replacement costs.
- However, it is important to note that the validation precision of this model is lower than its training precision. This could result in false positives, where the model predicts a failure when there is none, leading to unnecessary inspection costs. It may be worthwhile to further investigate ways to improve the precision of the model while maintaining a high recall.
- The logistic regression shows performance with oversampled data on training set varies between 0.86 to 0.88 recall
- The logistic regression shows performance with undersampled data on training set varies between 0.83 to 0.86 recall

The following will be recommended for RENEWIND

- The best model to help identify failures so that the generators could be repaired before failing/breaking to reduce the overall maintenance cost will be the Adaboost classifier tuned with oversampled data, Which presents the best recall of 85% with overall good performance across all set and this will help minimize false negatives.

Great Learning

Happy Learning !