Renewable energy-Wind

Model Tuning

07/19/2022

Contents / Agenda

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

Executive Summary

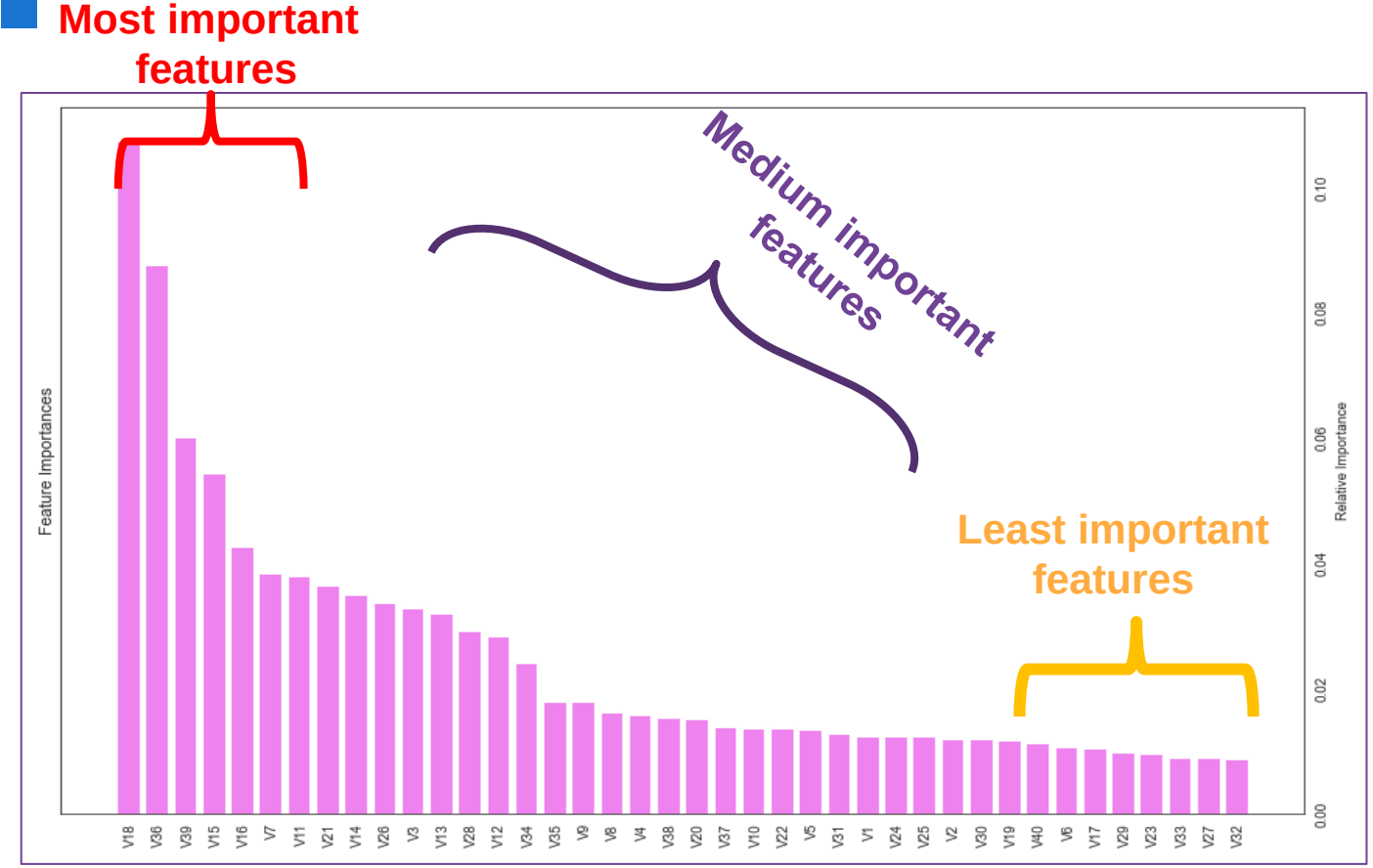
- Actionable insights & recommendations

The most important features are as listed V18,V36,V39,V15,V16

The least import features: V32,V22,V27,V29

- We recommend to keep eyes on the most important feature and tracking the variation in their value over time. Features with distribution skewed to the left toward negative values are more likely to be an indicator for failure of wind turbines
- Keep the most important feature sensors in great condition. Frequently inspect these sensors because any inefficiency in these sensors could cost a lot while make less frequent inspection for medium important features.
- Creating automatic alarm, which will start when values of the most important features becomes negative for following many days.
- Future work: with gathering more data, we recommend rebuild the models and tunning the hyperparameters using cloud or GPU for improved results and higher Recall score.

Executive Summary



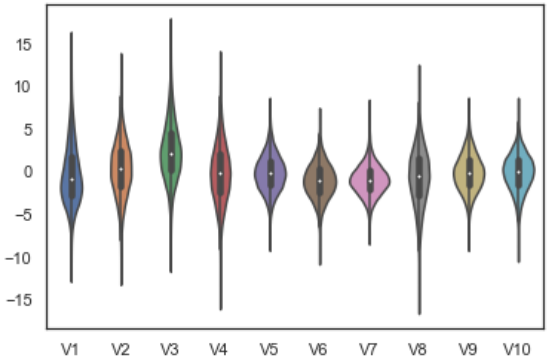
Business Problem Overview and Solution Approach

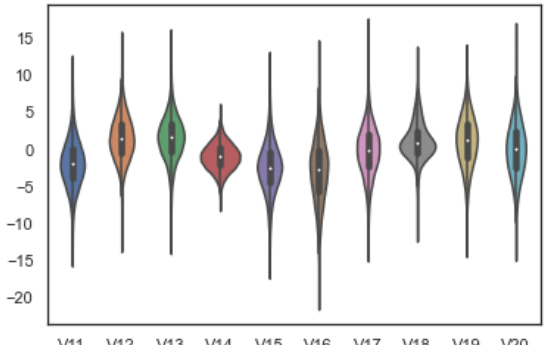
- Problem: The cost of repairing a generator is much less than the cost of replacing it, and the cost of inspection is less than the cost of repair. If component failure can be predicted accurately and the component is replaced before it fails, the costs of operation and maintenance will be much lower.

- Object: Identify component failures so that the generators could be repaired before failing/breaking to reduce the overall maintenance cost.

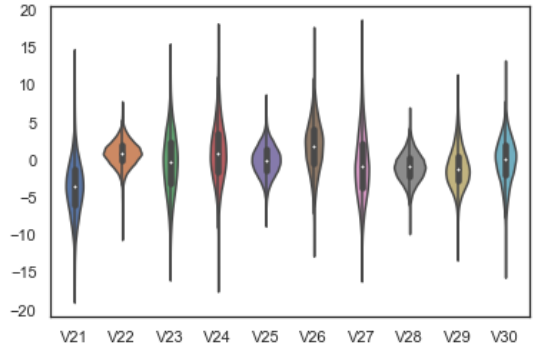
- The solution approach / methodology: Preparing the data; Build various classification models; dealing with inbalanced data set; build model with udersampled dataset and oversampled dataset; Tuning the models; Find the best model that will have generilized result; Pipelines to put the final model to production.

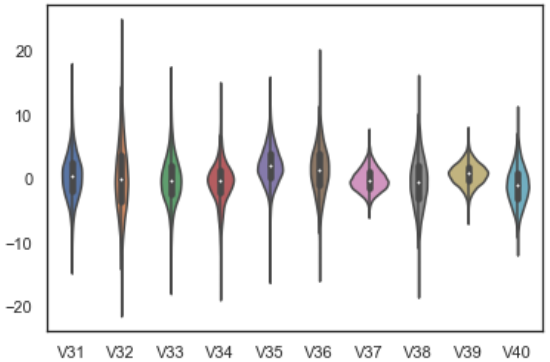
EDA Results

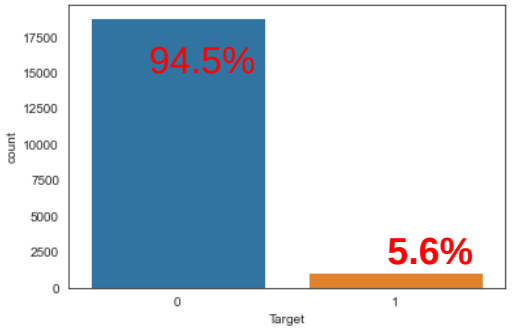




All features have a normal distribution, or they are close to the normal distribution.
Few features are skewed. V16,V8 , and V21 are skewed left while V3,V27,and V32 are skewed right.



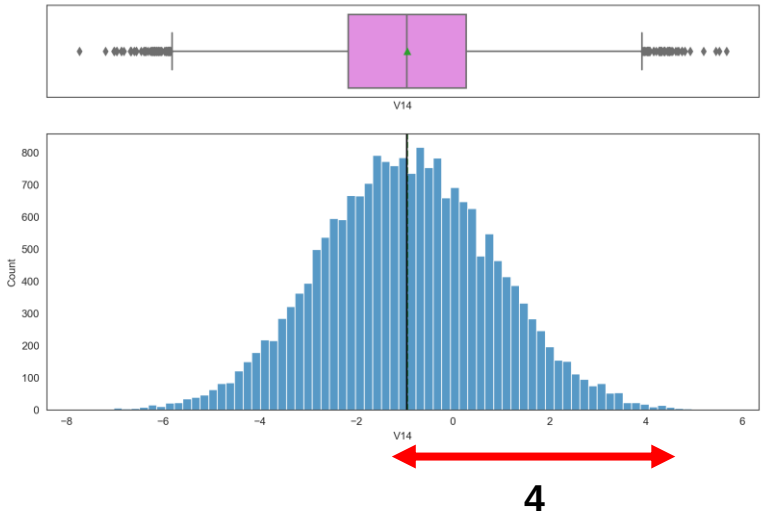


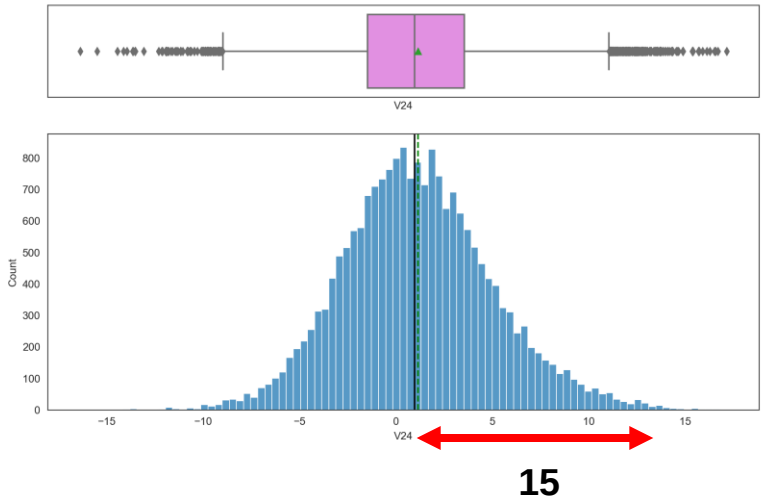


Link to Appendix slide on data background check

EDA Results

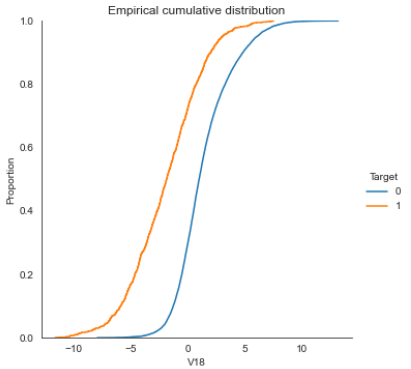
- Please mention the key results from EDA
- The variance of the independed features varies across the featuers .
  - E.g. V18,V14 have small variance while V24,V16,V8 have wide variance

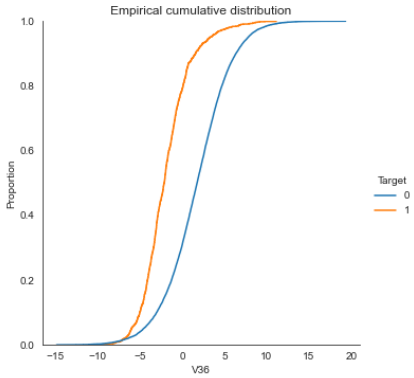


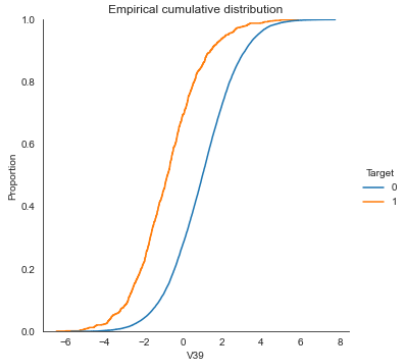


Link to Appendix slide on data background check

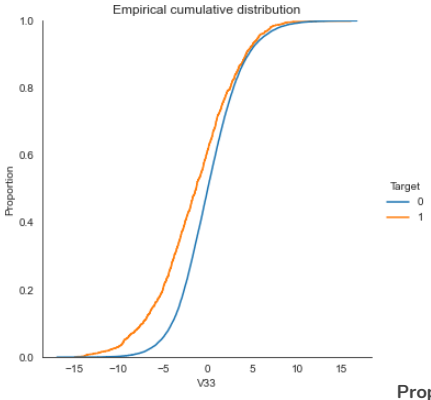
EDA Results

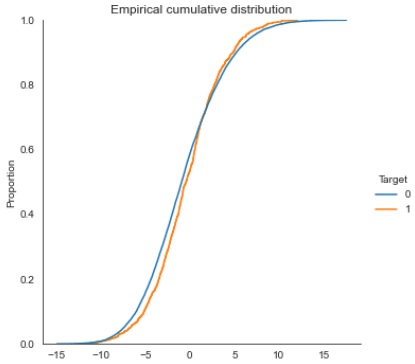


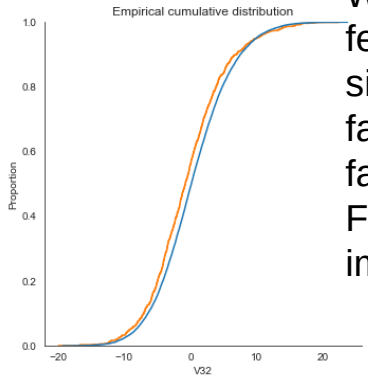




Features with ~90% of the values are negative could be indicators to failed sensors







When the ECDs features are similar for both failed and not failed sensors. Features are not important.

Data Preprocessing

- Duplicate value check
  - No duplication
- Missing value treatment
  - There are missing values in the first two columns
  - We pplied SimpImputer with median on data set
- Outlier check (treatment if needed)
  - No outlier,
- Feature engineering
  - All the feature are numerical values of sensors. No need to feature engineering.
- Data preparation for modeling
  - Split data to three parts: train(15000), validation(5000), and test.(5000)

Model Performance Summary

Which metric to optimize?

* We need to choose the metric which will ensure that the maximum number of generator failures are predicted correctly by the model.
* We would want Recall to be maximized as greater the **Recall**, the higher the chances of minimizing false negatives.
* We want to minimize false negatives because if a model predicts that a machine will have no failure when there will be a failure, it will increase the maintenance cost.

Model Performance Summary

Training	Accuracy	Recall	Precision	F1
Gradient Boosting tuned with oversampled data	0.990	0.989	0.992	0.990
AdaBoost classifier tuned with oversampled data	0.989	0.982	0.995	0.989
Random forest tuned with undersampled data	0.967	0.935	0.999	0.966
XGBoost tuned with oversampled data	0.998	1.000	0.996	0.998

Validation	Accuracy	Recall	Precision	F1
Gradient Boosting tuned with oversampled data	0.963	0.880	0.594	0.709
AdaBoost classifier tuned with oversampled data	0.983	0.873	0.819	0.845
Random forest tuned with undersampled data	0.942	0.899	0.488	0.633
XGBoost tuned with oversampled data	0.974	0.899	0.709	0.793

Winner

Random Forest model with udersampled data has an acceptable Recall score comparing with other models. All the other models show overfit.

Productionize and test the final model using pipelines

- steps taken to create a pipeline for the final model
  - Pipline contains:
    - 1. Imputering the missing values using median strategy,
    - 2. The estimator. To fit and predict the model (Tuned Random Forest model)

- Summary of the performance of the model built with pipeline on test dataset

Test	Accuracy	**Recall**	Precision	F1
Pipeline	0.988	**0.858**	0.931	0.893

- Summary of most important factors used by the model built with pipeline for prediction
  - **SimpleImputer (strategy="median") was applied first.**
  - **For RadnomForest model: max_features='sqrt',random_state=1,max_samples=0.6,n_estimators=200, min_samples_leaf=2)**
  - **We imputed the missing values of the dataset before applying the resampling method that used to training the pipelines model.**

Link to Appendix slide on model assumptions

APPENDIX

Model Performance Summary (original data)

	Logistic regression	Bagging	Random Forest	AdaBoost	GradientBoost	XGB
training	0.493	0.683	0.720	0.599	0.704	0.787
validation	0.490	0.712	0.732	0.593	0.745	0.811

- Random Forest model has the best result in training and validation data, although the model are generlized, the recall score shows 73%, which is not enough to build predictive model.

Link to Appendix slide on model assumptions

Model Performance Summary (oversampled data)

- Over sampling method:

For oversampling method we apply SMOTE with KNN=5, the sample number is 14167 + 14167

	Logistic regression	Bagging	RandomForest	AdaBoost	GradientBoost	XGB
training	0.886	0.975	0.981	0.895	0.925	0.989
validation	0.825	0.844	0.851	0.828	0.871	0.861

Models applied on Oversampled data shows high recall values on training set; However, on validation set are much smaller. All the model are overfit and failed to generalize. AdaBoost, GB and XGB has higher recall values and the difference between validation and training scores are less than other models. They're having the potential to become generalized model by tunning the hyperparameters.

Link to Appendix slide on model assumptions

Model Performance Summary (undersampled data)

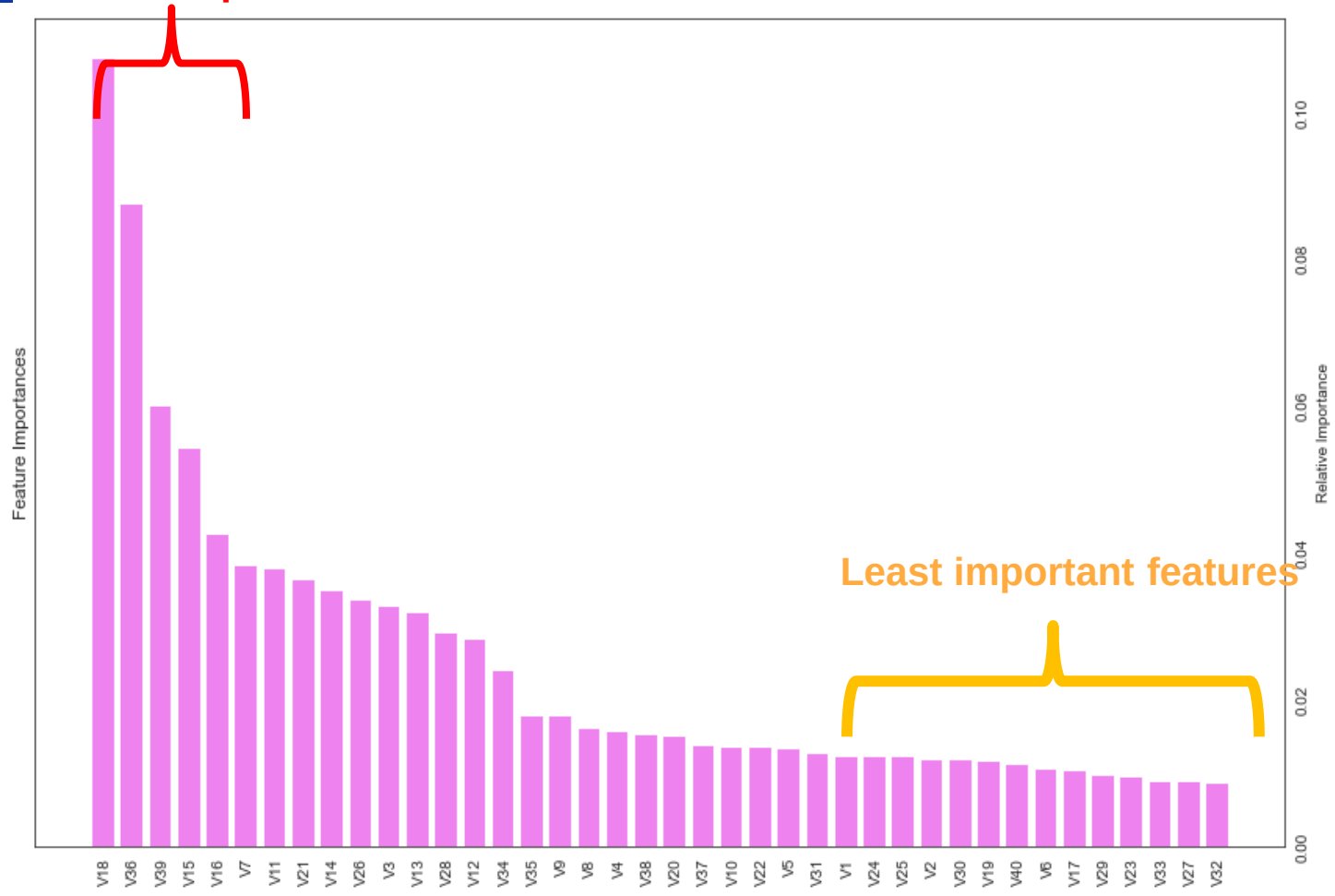
- Under sampling method:
- The undersampling method is RandomUnderSampler with sampling strategy 1. The total sample number 813+813.

	Logistic regression	Bagging	RandomForest	AdaBoost	GradientBoost	XGB
training	0.858	0.842	0.879	0.860	0.871	0.889
validation	0.844	0.877	0.901	0.871	0.891	0.907

The models applied on undersampled data show high training and validation recall results, However, applying udersampling method will make our data set smaller. We will choose RandomForest to tunning model.

Link to Appendix slide on model assumptions

Most important features



Least important features

Great Learning

Happy Learning !