greatlearning
Power Ahead

# Renewind Project

# Contents

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.).

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

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.

“1” in the target variables should be considered as “failure” and “0” will represent “No failure”.
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.

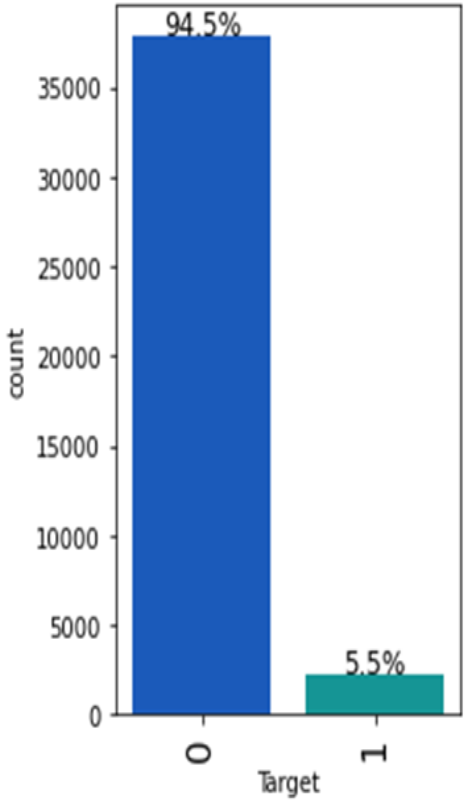
# Data Description

- 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

# EDA

**Observation**

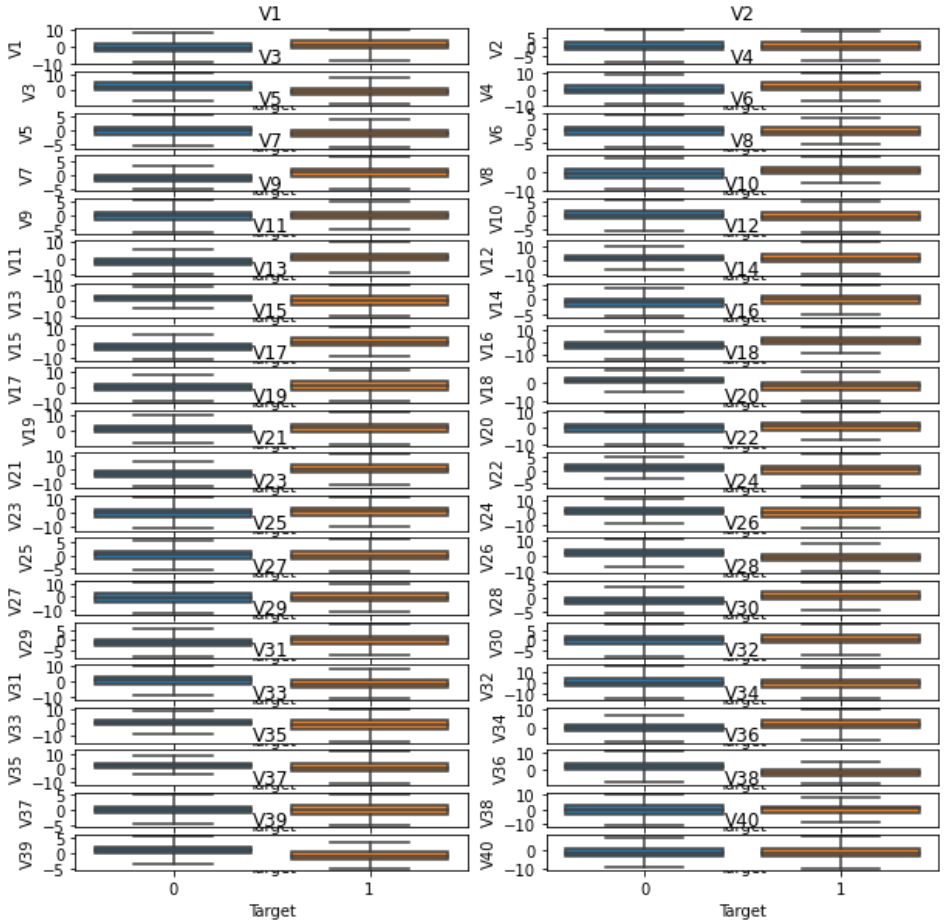
- Majority of the generators in the dataset 94.5% do not have a failure.
- Failure has been found in only 5.5% of the generators.
- Data is heavily imbalanced; it would need some oversampling or under sampling.



# EDA

## Observation

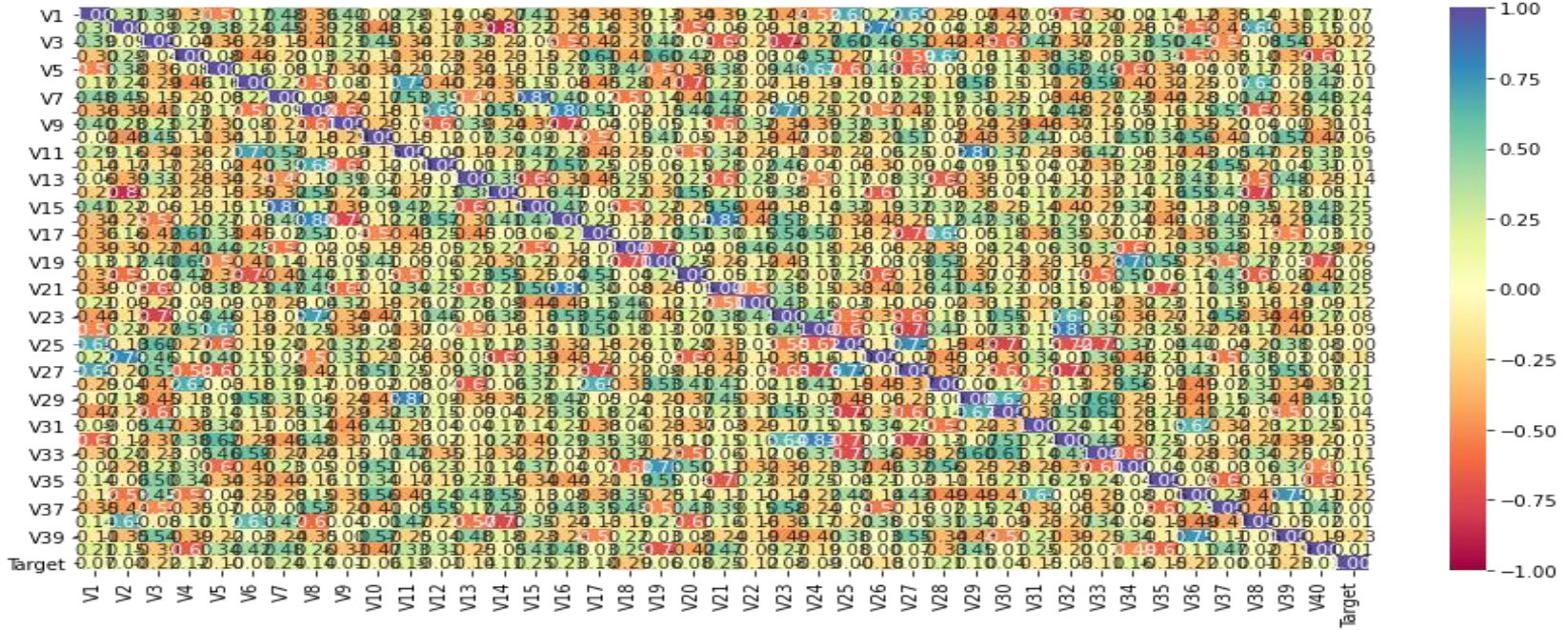
- Almost all the variables have normal distributions.
- Distribution of V1,V18,V27,V37 are a bit right skewed.



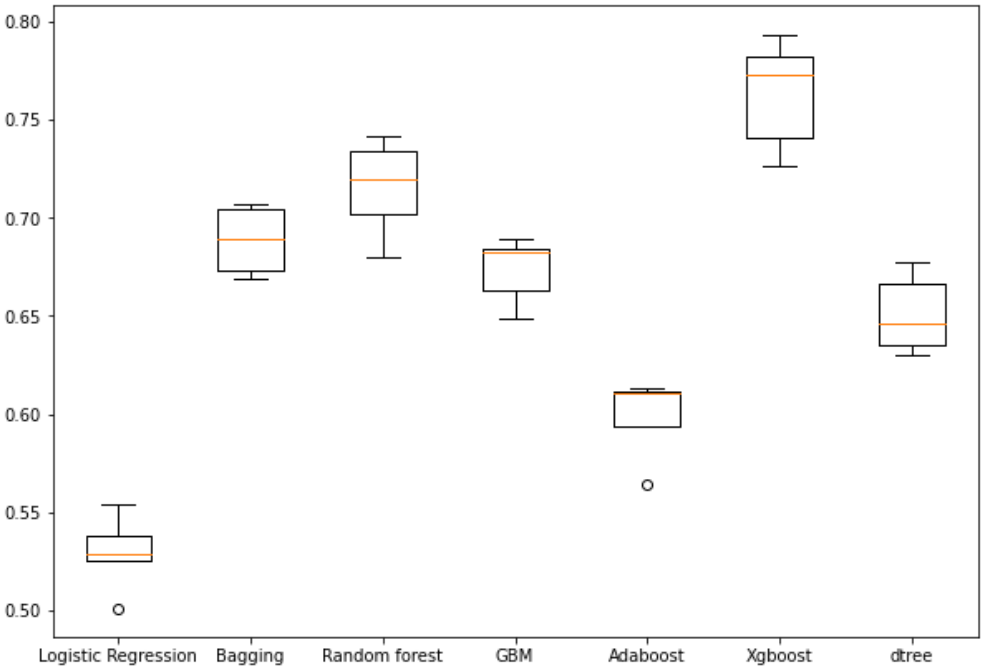
# EDA

## Observation

- There are some variables showing correlation like V7 and V15, V9 and V16, V23 and V32.
- There are too many variables to read the correlation heatmap correctly.



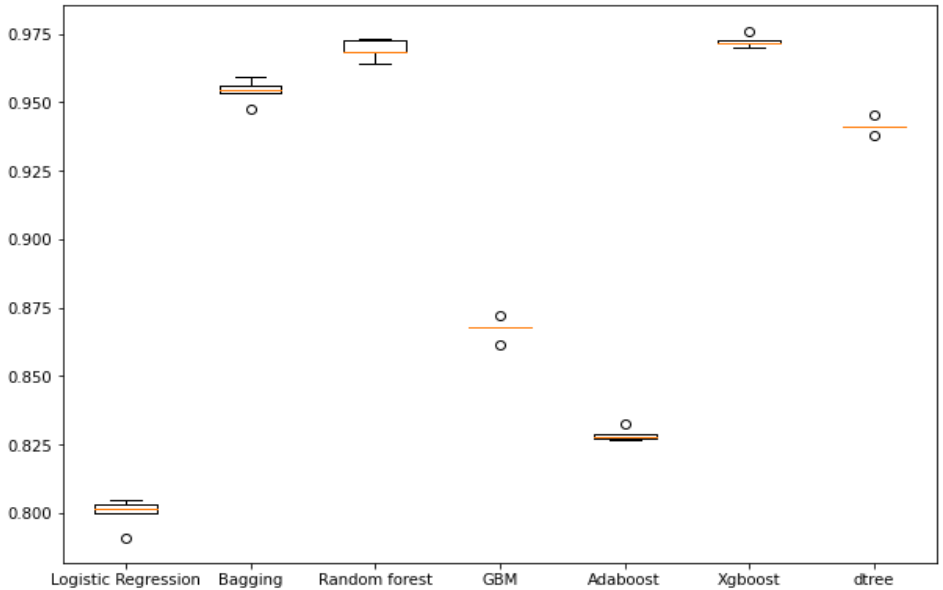
# PERFOMANCE ON TRAINING SET AND CROSS-VALIDATION




```
Cross-Validation Performance:

Logistic Regression: 52.948994378378856
Bagging: 68.86574469253958
Random forest: 71.52966747632996
GBM: 67.35879816374911
Adaboost: 59.88430481901255
Xgboost: 76.32838037155607
dtree: 65.12035300489133

Training Performance:

Logistic Regression: 0.5199619771863118
Bagging: 0.6889168765743073
Random forest: 0.7181619256017505
GBM: 0.6725409836065573
Adaboost: 0.5954281567489115
Xgboost: 0.7733270499528746
dtree: 0.6606280193236715
```

- Performance on training set varies between 0.48 to 0.50 recall.
- Considreable Performance of different models on validation data are: XGBoost:76.33, Random Forest:71.53 and GBM:67.36.

# LOGISTIC, BAGGING, RANDOM FOREST,ADABOOST,DECISION TREE MODELS AFTER SMOTE OVERSAMPLING ON ATRAIN DATASET




```
Cross-Validation Performance:

Logistic Regression: 52.948994378378856
Bagging: 68.86574469253958
Random forest: 71.52966747632996
GBM: 67.35879816374911
Adaboost: 59.88430481901255
Xgboost: 76.32838037155607
dtree: 65.12035300489133

Training Performance:

Logistic Regression: 0.5199619771863118
Bagging: 0.6889168765743073
Random forest: 0.7181619256017505
GBM: 0.6725409836065573
Adaboost: 0.5954281567489115
Xgboost: 0.7733270499528746
dtree: 0.6606280193236715
```

In Oversampled data XGBoost is giving the highest score 97.25 and the Random Forest is giving the highest score 96.93. All models are giving generalised performance.

- We can see that XGBoost is giving the highest cross-validated Minimum_Vs_Model_cost:97.25 followed by RandomForest: 96.92 and then bagging classifier: 95.41
- The boxplot shows that the performance of xgboost is consistent with 1 outlier and with no outlier with Random Forest and one outlier for Bagging classifier.
- The Performance of XGBoost, Random forst and Bagging classifier, is generalised on validation set as well.
- let us check how the different models perform with the undersampled data.

# MODEL PERFORMANCE WITH UNDERSAMPLED DATA

```
Cross-Validation Performance:

Logistic Regression: 77.65522788135344
Bagging: 81.69551921903387
Random forest: 84.67413072888996
GBM: 82.88956467253165
Adaboost: 80.05315674951575
Xgboost: 84.03896599877898
dtree: 77.45754336793459

Training Performance:

----------------------------------------------------
Logistic Regression :
 0.4919064748201439
----------------------------------------------------
Bagging :
 0.6736453201970444
----------------------------------------------------
Random forest :
 0.7362045760430687
----------------------------------------------------
GBM :
 0.6918212478920742
----------------------------------------------------
Adaboost :
 0.5226114649681529
----------------------------------------------------
Xgboost :
 0.7565698478561549
----------------------------------------------------
dtree :
 0.49757428744693755
```

```
Before DownSampling, counts of label 'Yes': 1640
Before DownSampling, counts of label 'No': 28360

After DownSampling, counts of label 'Yes': 1640
After DownSampling, counts of label 'No': 1640

After DownSampling, the shape of train_X: (3280, 40)
After DownSampling, the shape of train_y: (3280,)
```

- All models in undersampled data have given worse result than oversampled data.
- In undersampled data Random Forest is giving the highest score 84.7 and then the XGBoost: 84.04.
- We can see that Random Forest is giving the highest cross-validated recall followed by XGBoost and then Gradient boosting classifier

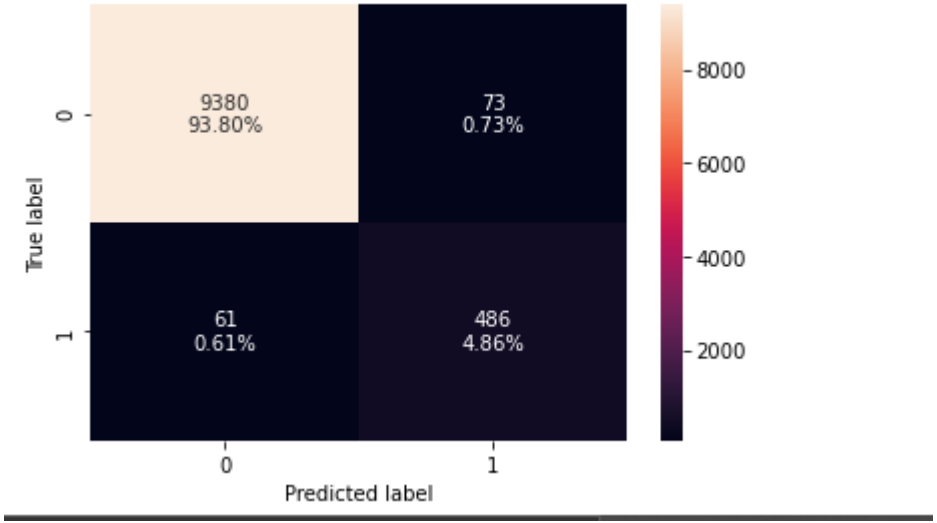
# XGBOOST AFTER GRID SEARCH HYPERTUNNING BEST PARAMETERS

Training performance:

| | Accuracy | Recall | Precision | F1 | Minimum_Vs_Model_cost |
|---|---|---|---|---|---|
| 0 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |

Validation performance:

| | Accuracy | Recall | Precision | F1 | Minimum_Vs_Model_cost |
|---|---|---|---|---|---|
| 0 | 0.987 | 0.888 | 0.869 | 0.879 | 0.813 |




## Observation

XGBoost is giving good minimum vs model cost in validation set. Although it's overfitting

# MODEL PERFORMANCE COMPARISM AND CONCLUSIONS

Training set performance comparison:

| | Xgboost Tuned with Grid search | Xgboost Tuned with Random Search | Random Forest Tuned with Grid search | Random Forest Tuned with Random search | Bagging Classifier Tuned with Grid search | Bagging Classifier Tuned with Random search |
|---|---|---|---|---|---|---|
| **Accuracy** | 1.000 | 0.999 | 0.999 | 0.999 | 1.000 | 1.000 |
| **Recall** | 1.000 | 1.000 | 0.998 | 0.998 | 1.000 | 1.000 |
| **Precision** | 1.000 | 0.999 | 1.000 | 1.000 | 1.000 | 1.000 |
| **F1** | 1.000 | 0.999 | 0.999 | 0.999 | 1.000 | 1.000 |
| **Minimum_Vs_Model_cost** | 1.000 | 1.000 | 0.997 | 0.997 | 1.000 | 1.000 |

Minimum_Vs_Model_cost is 1 in both XGBoost and Bagging classifier and 0.999 in Random forest classifier.

Validation performance comparison:

| | Xgboost Tuned with Grid search | Xgboost Tuned with Random Search | Random Forest Tuned with Grid Search | Random Forest Tuned with Random Search | Bagging Classifier Tuned with Grid Search | Bagging Classifier Tuned with Random Search |
|---|---|---|---|---|---|---|
| **Accuracy** | 0.987 | 0.985 | 0.991 | 0.991 | 0.986 | 0.986 |
| **Recall** | 0.888 | 0.887 | 0.872 | 0.872 | 0.848 | 0.848 |
| **Precision** | 0.869 | 0.841 | 0.950 | 0.950 | 0.884 | 0.884 |
| **F1** | 0.879 | 0.863 | 0.909 | 0.909 | 0.866 | 0.866 |
| **Minimum_Vs_Model_cost** | 0.813 | 0.803 | 0.814 | 0.814 | 0.775 | 0.775 |

- Both XGBoost tuned and Random Forest tuned are giving generalised performance.
- XGBoost grid: 0.813, XGBoost random: 0.803,
- Random Forest grid: 0.814, Random Forest Random: 0.814
- I'll choose the Random Forest random search as it's execution time is comparatively less and it is giving a slightly better performance than XGBoost

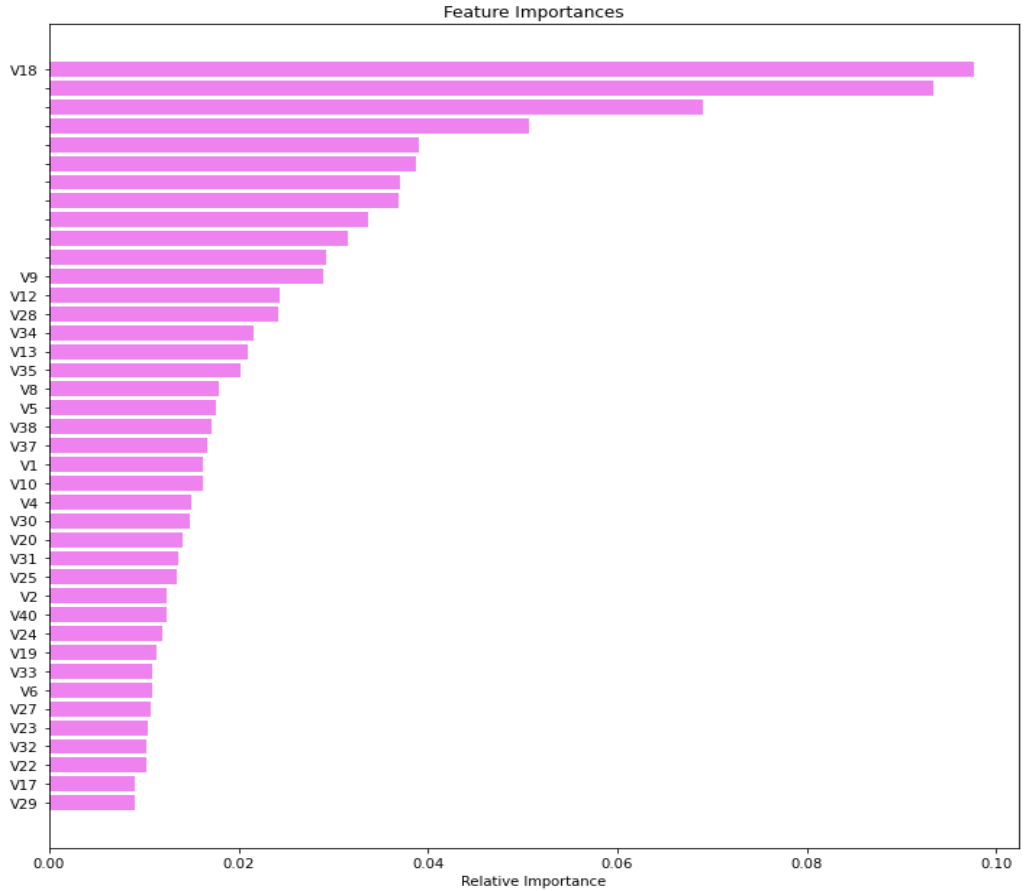
# FINAL MODEL

Test performance:

| | Accuracy | Recall | Precision | F1 | Minimum_Vs_Model_cost |
|---|---|---|---|---|---|
| **0** | 0.990 | 0.857 | 0.951 | 0.902 | 0.799 |

**Observation**

- The 10 most important features out of 40 are V18, V36, V39, V15, V26, V16, V3, V21, V7 and V14.



# Business Insights and Recommendations

- The 10 most important features out of 40 are V18,V36,V39,V16,V26,V15,V3,V21.V7 and V14.

- The input from this features should be considered when making predictions.

greatlearning
*Power Ahead*

# Happy Learning !