Predict Air Pressure system failure

Abstract

Identifying issues causing systems failures are critical part of the manufacturing process and having large number of features pose a challenge in identifying the exact components ahead of time. This paper tried to reduce dimensionality of such system failure by first doing a Binary classification of weather the failure happened because of APS or non-APS system. This is achieved by feature extraction, dimensionality reduction and applying Random Forest, SGD, Logistic and Ada boost algorithms.

Business Understanding: Defining the Problem

Manufacturing has evolved over time and now has sensors to capture parameters during the manufacturing process and during operations of the equipment. Such data can be used for predicting malfunctions of devices/equipment when put under certain conditions that are potentially repeatable in real world, and an early intervention could benefit both the manufacturer cost, reputation, and safety of customers if it involves automobiles and heavy-duty equipment. This use case is about Air Pressure systems failure caused during the application of brakes and to predict components failures.

Defining the target variable:

Data set contains a training set and a test set segregated and is available from UCI ML repository. The training set file contains 60k rows and 1,000 of those belong to the positive class and 171 columns determined by the'1' in the class column. All attributes are numeric. 70 of these features came from histograms with ten bins each. The data consists of a subset of all available data, selected by experts. Target is to find if the class is 0 or 1

Data Understanding:

Data columns are.

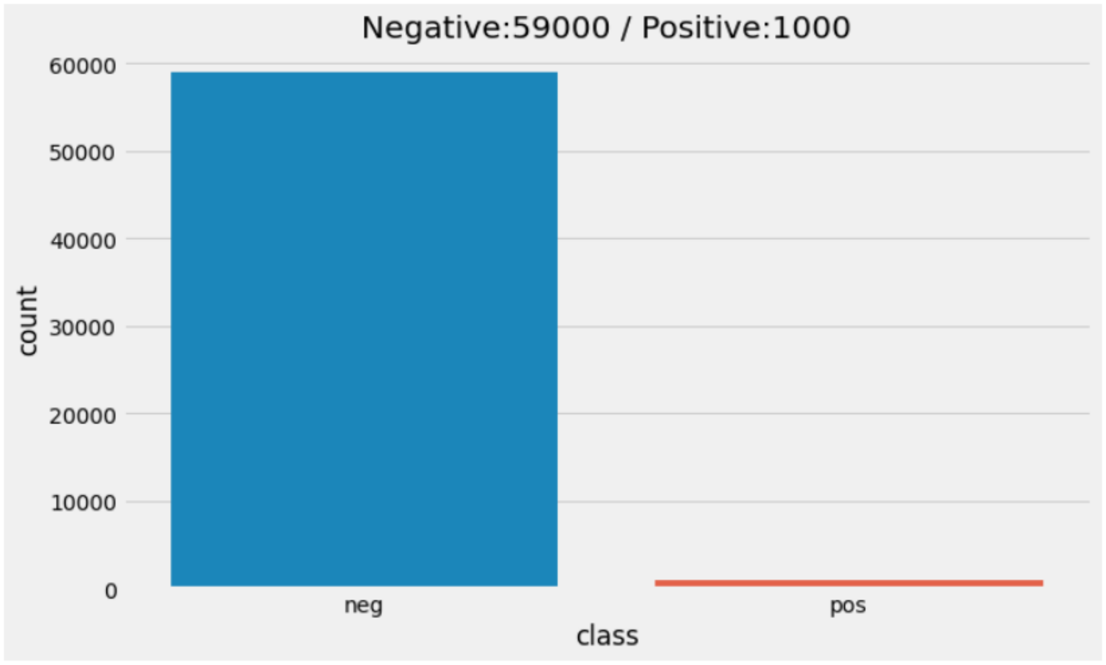
```
class      object
aa_000      int64
ab_000    float64
ac_000    float64
ad_000    float64
           ...
ee_007    float64
ee_008    float64
ee_009    float64
ef_000    float64
eg_000    float64
Length: 171, dtype: object
```

Since the attributes are anonymous there is very little information about what they mean in real world. It consists of both single numerical counters and histograms consisting of bins with different conditions.

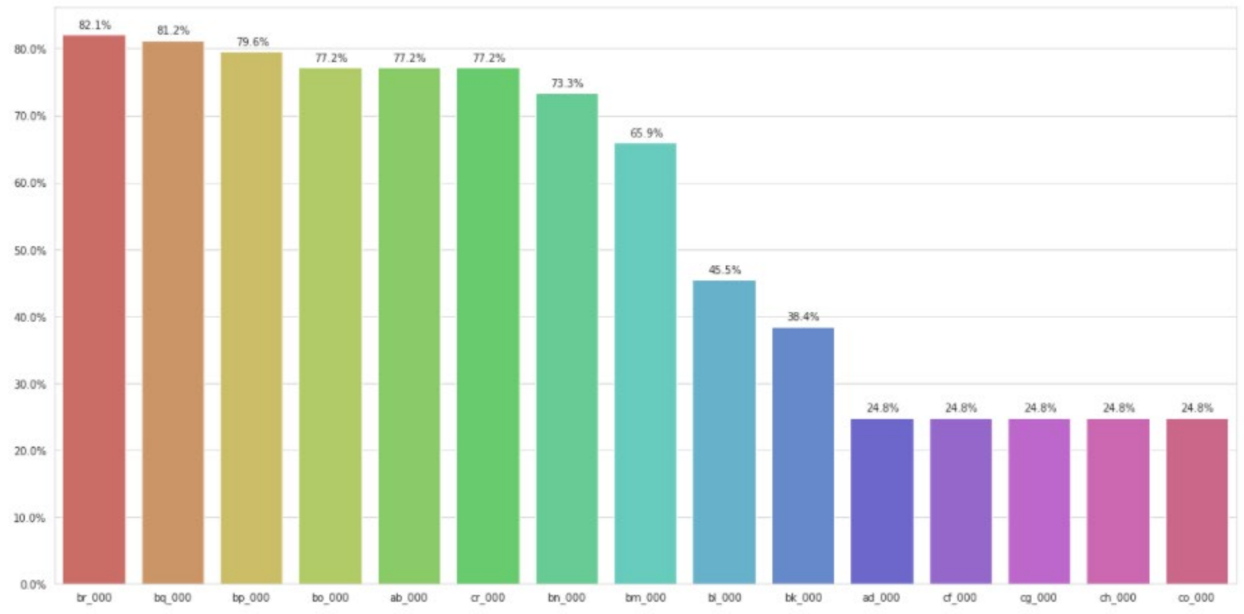
Data Exploration & imputations:

Performing EDA on the data set, the following observation are made against the class, and it explains that we are dealing with highly imbalance data set.

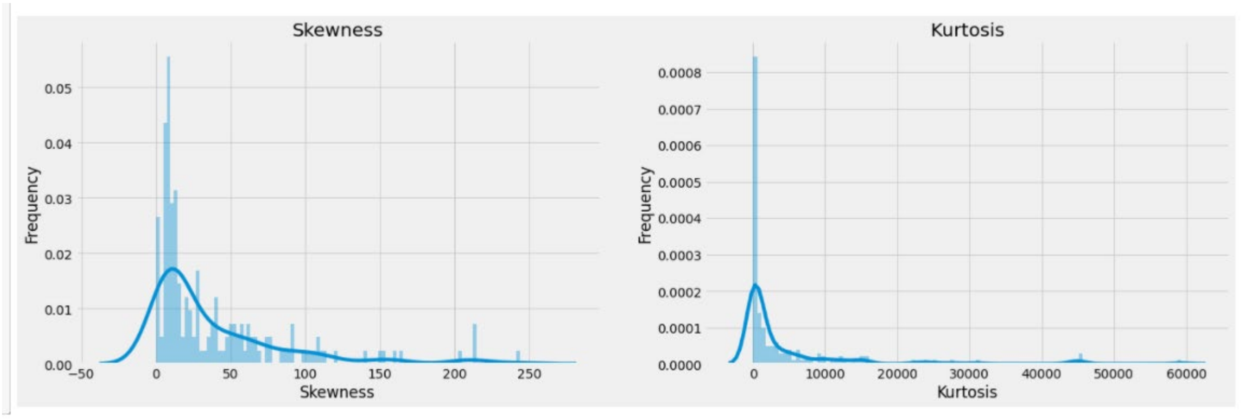
```
Text(0.5, 1.0, 'Negative:59000 / Positive:1000')
```



Quite a bit of missing values is in the features and is evident from the following graph showing the top 15 features with missing values. Some of which are in the 80% missing and while other are close to 50% values missing.



After performing N/A validations, removed features are: ['br_000', 'bq_000', 'bp_000', 'bo_000', 'ab_000', 'cr_000', 'bn_000', 'cd_000']. And 10 features with missing values are imputed with median values.

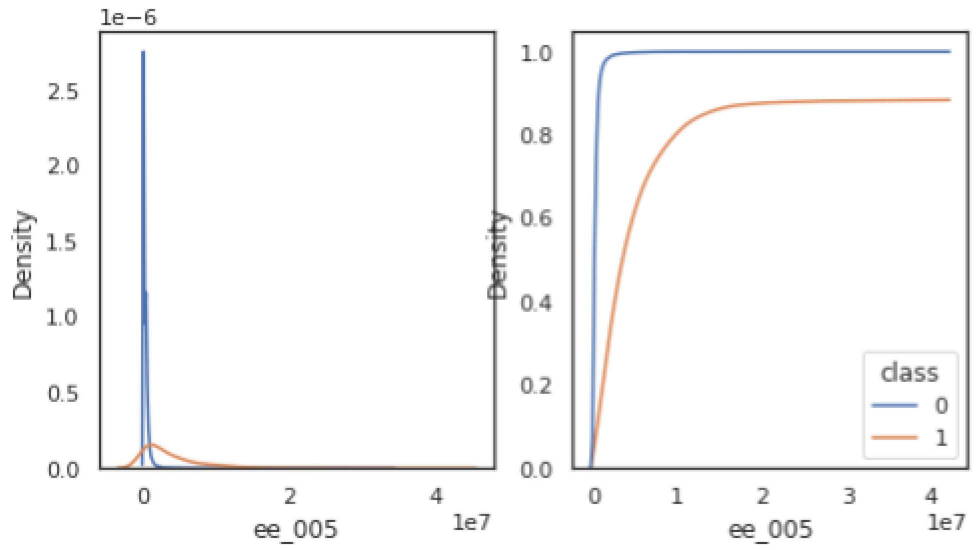


The above graph also shows the data is skewed to the left, while left tailed relative to the normally distributed from the kurtosis graph.

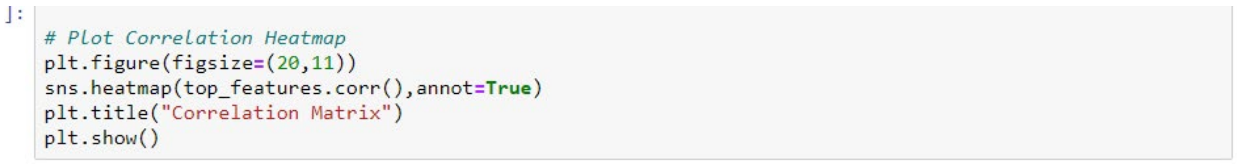
Feature selection and elimination: -

Using Random Forest classifier, the following feature are selected.
['ag_001', 'ag_002', 'ag_003', 'ay_005', 'ay_006', 'ay_008', 'ba_002', 'ba_003', 'ba_004', 'cn_000', 'cn_004', 'cs_002', 'cs_004', 'ee_003', 'ee_005']

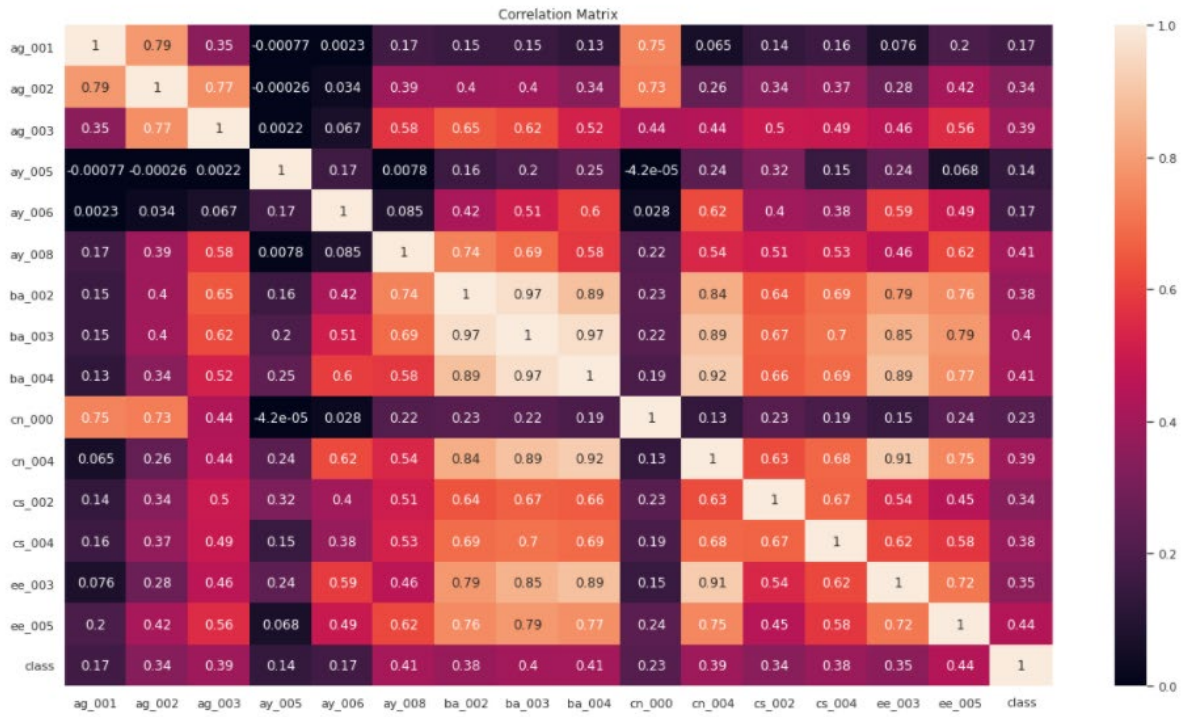
Up on doing the density plots of the features **ag_001 and ay_005** we can see the failure of APS c omponent is set to 0. It also means the higher the values on these features there is high chances of APS failure.



To further evaluate the correlation of the selected features, it also show ee_005 and ba_004,cn_ 004,cs_004 has higher correlation to the class. While ba_004 to cn_004 is at the highest rate of 0 .97 and varying degree of relations among the selected features.
Another observation is with ag_002, ag_001 & cn_000 has high degrees of failure in the APS co mponents



```
]:
    # Plot Correlation Heatmap
    plt.figure(figsize=(20,11))
    sns.heatmap(top_features.corr(),annot=True)
    plt.title("Correlation Matrix")
    plt.show()
```



Data Standardization & transformations:

Unbalanced datasets cause reduction in accuracy biased towards majority class. To counter that, followed –

Under Sampling

Over Sampling

Synthetic minority oversampling technique (SMOTE)

Both Under and Over Sampling

Using these methods tested on logisitc regression to test the best method for further analysis.

Data is fed to MinMax scalar to transform features by scaling feature with a given range.

Since the main challenge is unbalanced data set, fed the data to SMOTE under sampling method to create a balanced data set for the positive class as shown in the first diagram above (0 vs 1).

This resulted in the following data distributions for positive and negative.

```
0       33226
1       16613
```

Modeling:

F1 score and confusion matrix is the measure taken to select a best performing model after hyper parameter turning process. And total cost is based on the type 1 and type 2 errors.

FN is also called as Type 2 error.

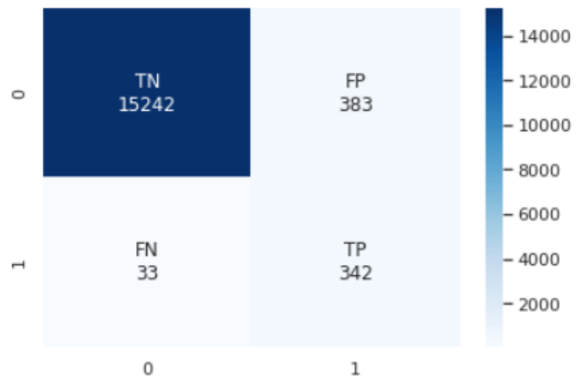
FP is called as Type 1 error.

Total cost = FN * *500 + FP* *10

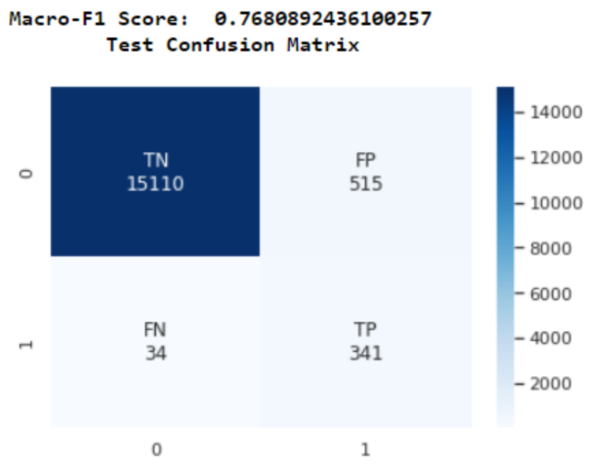
The ideal model is to minimize FN, and to do that hyperparameters are tuned for a better recall, that results in increased number of FP, and FP of a models to have a better precision.

Logistic Regression

```
Macro-F1 Score:  0.8041776993233303
        Test Confusion Matrix
```

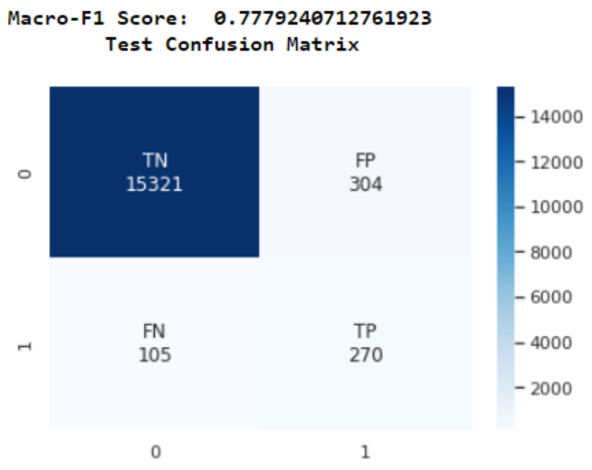


SGDClassifier



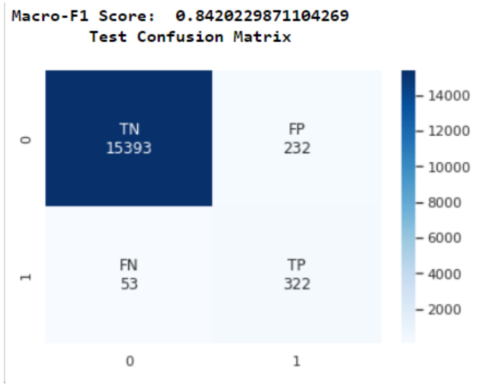
```
Macro-F1 Score:  0.7680892436100257
        Test Confusion Matrix
```

DecisionTreeClassifier



```
Macro-F1 Score:  0.7779240712761923
        Test Confusion Matrix
```

AdaBoost



```
Macro-F1 Score:  0.8420229871104269
        Test Confusion Matrix
```

False negative is 50 times higher than a false positive. To combat this problem one way is to predicted class based on the confidence of our classifier model. For that, a slightly adjusted the function that predicted class variable during the hyper parameter tuning using GridSearchCV. For every feature subset we set a threshold for the prediction confidence and changed it in steps of one percent.

All the models are passed with the same set of 10-fold cross validation as a fixed seed during the parameter passing of params, so cross-validation with many models is at the same standard so overfitting issues would not arise.

Based on these three models, **Adaboost with 0.84 F1 score** performed best. Here in LR parameter C is the hyper parameter to combat the highly imbalanced dataset.

Conclusion:
After cleansing data issues with missing values of highly imbalanced data, smoothening the data, feature engineering has produced a reasonable prediction of positive APS failure or not related to APS brake failure. Detection of a failure in an APS of trucks could save fleet owners quite a bit of repair cost, maintenance, and loss of productivity. Although the meaning of the faults is not known from this data set, a detail decoding of the features would help prediction of a fault only even for the histograms (buckets) that are provided/available.

Next steps: - Still working on other models specially with other decision tres as many papers has claimed, ensemble techniques and use neural network to speed the model building which is the main bottle neck that is taking approximately 20hours for some of the boosting algorithms.

Code files are uploaded to GitHub repo at the following location.

https://github.com/11leven/portfolio/tree/gh-pages/DSC%20680

Q & A

1. Can this data set use for predicting $ amount savings?

   Yes dataset can be used to predict a regression use cases, my analysis is a classification problem to identify if failure caused by APS or not

2. Do the data set features have logic meaning?

   No, the data is anonymized, and the features has no logical meaning as it exist in the UCI ML repository

3. What values are in Histograms bins?

   They represent temp in ranges

   Bin1 T < -20

   Bin2 T >=-20 and T<-20

Bin3 T>=0 and T<20

Bin4 T>20

4. What kind of imputation techniques used for missing values?

   Mean, median and frequent repeating values are used by Sklearn library

5. What standardization techniques used?

   Min/Max scalar and SMOTE

6. What are the challenges in the analysis?

   Imbalanced data set lead to high overfitting, so must cleanse and normalize the data set

7. What kind of hyper parameter turning done?

   GridSearch weight tuning is used depth and weight optimum values

8. Feature engineering methods?

   Since lot of features have values missing, removing them while doing heatmap correlation analysis and only used those features for modeling.

9. What model evaluation techniques used for all models?

   F1 score and confusion matrix

10. Next steps?

    Build more models using Tensor flow and reduce the computation times with auto imputation and parameter tuning.

Acknowledgements & References:

1)Industrial Challenge: Using Machine Learning for Predicting Failures

https://www.researchgate.net/publication/313065916_IDA_2016_Industrial_Challenge_Using_Machine_Learning_for_Predicting_Failures

2) Combining Boosted Trees with Metafeature Engineering for Predictive Maintenance

https://repositorio.inesctec.pt/bitstream/123456789/4388/1/P-00K-WFP.pdf

3)Prediction of Failures in the Air Pressure System of Scania Trucks Using a Random Forest and Feature Engineering

https://www.researchgate.net/publication/309195602_Prediction_of_Failures_in_the_Air_Pressure_System_of_Scania_Trucks_Using_a_Random_Forest_and_Feature_Engineering

4) Air brake

https://www.britannica.com/technology/air-brake
5) https://medium.com/analytics-vidhya/aps-failure-at-scania-trucks-data-set-1eb97b128125)
https://pdfs.semanticscholar.org/b280/216a1d63015afc6a3d1aac9595aeb2b7dd5a.pdf

6) Handling imbalanced datasets and modeling of positive and negative prediction
https://towardsdatascience.com/predicting-a-failure-in-scanias-air-pressure-system-aps-c260bcc4d038

7)A Gradient Boosting Algorithm for Survival Analysis via Direct Optimization of Concordance Index
https://www.researchgate.net/publication/259354509_A_Gradient_Boosting_Algorithm_for_Survival_Analysis_via_Direct_Optimization_of_Concordance_Index

8)An Empirical Comparison of Missing Value Imputation Techniques on APS Failure Prediction
https://app.amanote.com/v3.11.3/note-taking/document/X6B23nMBKQvf0Bhizz2K

9) Improving predictive maintenance classifiers of industrial sensors' data using entropy. A case study.
http://norma.ncirl.ie/3429/1/eleonoraperuffo.pdf

10) Predicting Failures from Sensor Data

https://www.h2o.ai/blog/predicting-failures-from-sensor-data-using-ai-ml-part-1/
https://www.h2o.ai/blog/predicting-failures-from-sensor-data-using-ai-ml-part-2/