MODELLING AND OPTIMISING THE FAULT-HANDLING PROCESS OF NETWORK ENGINEERS AT ARQIVA

Methodology

- Aggregating the dataset from different JSON files and converting it into a CSV file
- Preprocessing the attributes/features for ML
- Spot checking with different ML algorithms
- Using metrics to compare model performance
- Performing feature selection

Aggregating the dataset

- The dataset came in the form of JSON
- Total instances 18827 (success=10000)
- Wrote a function to combine the JSON and convert it to CSV
- Necessary to visually check the number of columns, types of dataset (binary, continuous or categorical)

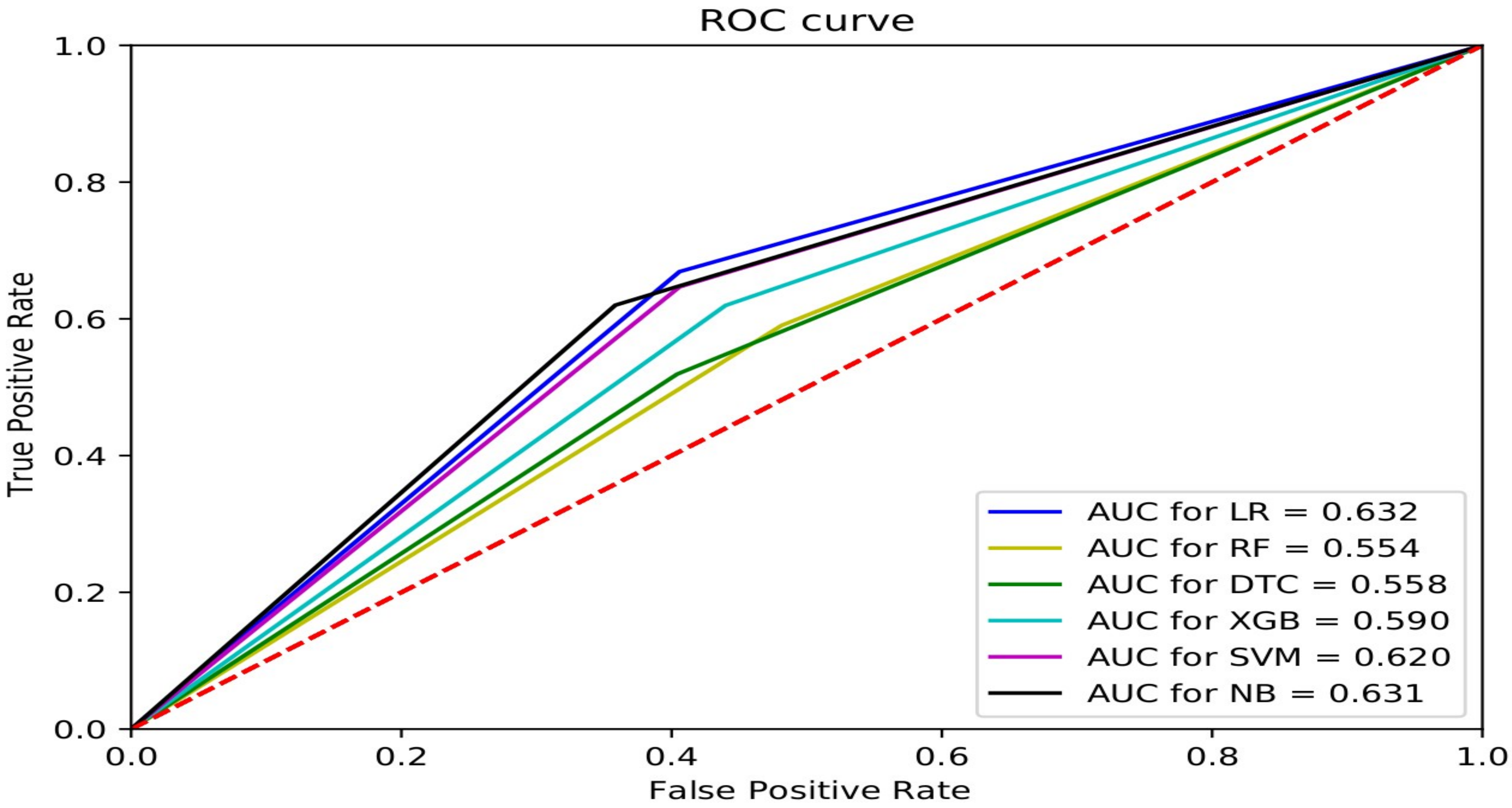
Preprocessing

- The dataset had to be preprocessed before giving it to the ML algorithms
- First, features not relevant such as ID, columns with constant entries, dates, engineer_note, and so on were removed
- One-hot encoding was used in converting categorical dataset to a usable format
- Normalise the columns with continuous entries to (0, 1)

Modelling

- Logistic regression, Decision Tree Classifier, Random Forest, Support Vector Machine, Naive Bayes and XGBoost were tried on dataset
- Classification report and receiver operation characteristic curve was then used to determine the best performing model
- Logistic regression had the overall best performance with AUC of 0.63

ROC curve for models



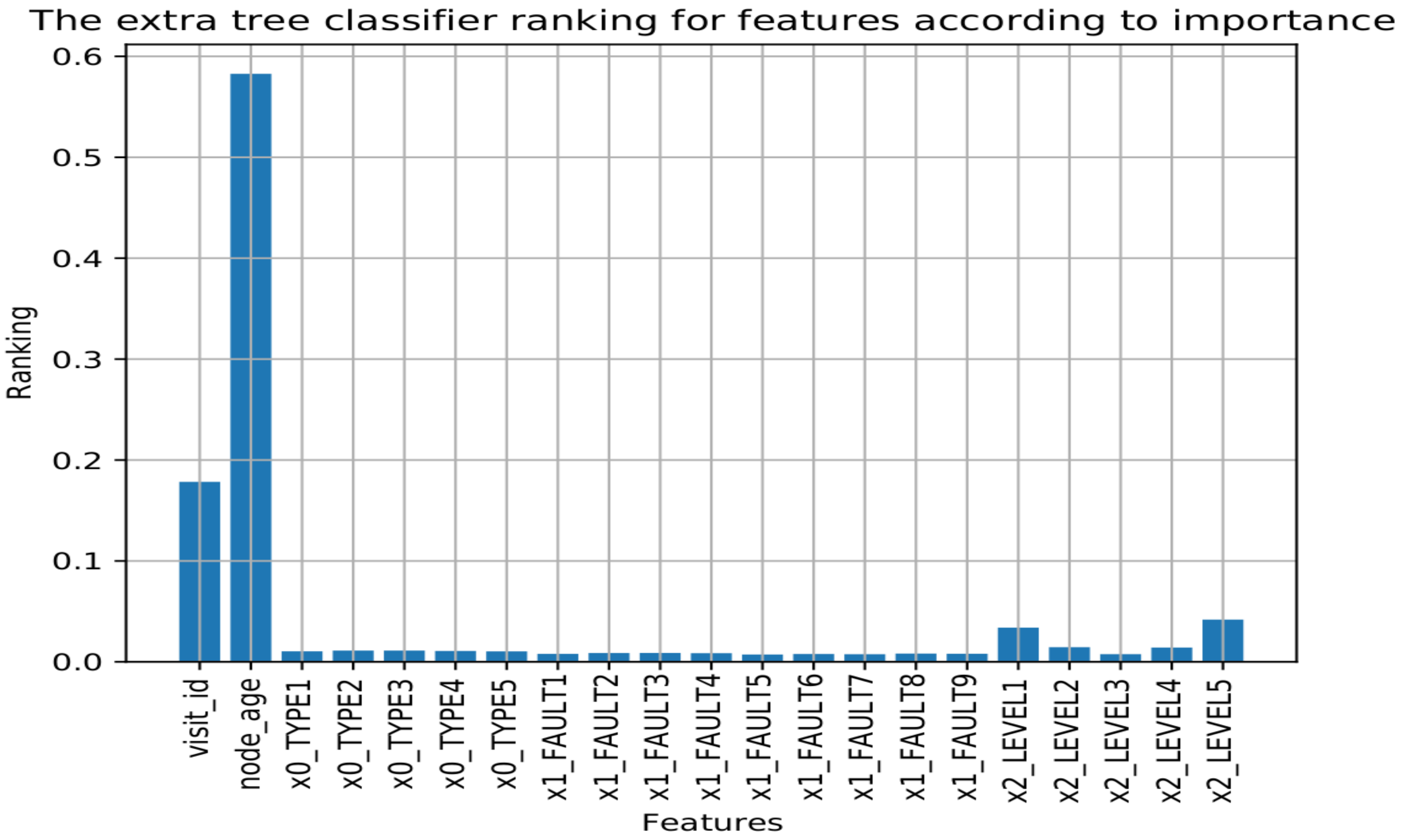
Feature selection

- There are various ways of performing feature selection which includes wrapper method, filtering method and the statistic method
- All three were tried using different algorithms i.e RFE, ETC, RTC and US.
- Features/attributes were ranked according to their contribution to the model or in conjunction with other attributes to improve models

Tree based feature selections

- Tree algorithms are one of the best techniques for feature selections. A tree can be easily explained as an if/else statement.
- A feature that contributes the most to the division of the dataset in the solution space becomes more important
- Extra Tree Classifier and the Random Forest Classifier had a similar performance
- They jointly favoured features such as node_age, visit_id, entry_level_engineer and expert_level_engineer. US favoured competency more than node_age and visit_id

Tree ranking diagram



Conclusion and future work

- I have created a model that can classify the performance of Engineers based on features. I have also determined the features that positively or negatively affects the performance of the model
- Future work that can be carried out on the dataset is to optimise the performance by tuning the hyper-parameters of the model and performing feature engineering