**Initial Model for Tookitaki Test Dataset**

The initial model I created unfortunately performed far from the required benchmark, achieving only a AUC of 0.610 and a Gini score of 0.224. I believe if I could devote more resources to it, reaching / exceeding the benchmark would be achievable.

**1. Data Exploration Insights**

1.1 The data is imbalanced with only 4.29% of the entire data falling into the Bad\_flag\_worst6. As such we used techniques were used to offset this imbalance of training data. The techniques of SMOTE and randomized split will be explained in the AI Pipeline.

1.2 Twelve suggested features were provided, of which my model used only eight of the features. Unfortunately the choice of these eight features were chosen purely for the interest of time / ease of calculation.

1.3 The data exploration / modeling process was done without the use of domain knowledge. I had wanted to explore the data first without the use of domain expertise because I believe personally that we should at least try to obtain additional insights from the data first before the application of domain expertise.

1.4 AI Pipeline

1.4.1 Rather than using the test and train datasets verbatim, I combined the datasets into one large dataset and used randomized split algorithm 70-30 split algorithm. I normally would split the dataset into three portions, the train, test and validation set. However here I used only test and training datasets.

1.4.2 The data is then prepared based on the suggested features.

1.4.3 I used a simple label encoder to replace categorical data. (I tried one hot encoding for this dataset but it does not seem to work well)

1.4.4 As training data was insufficient, I used a technique known as SMOTE to increase the number of default training data. SMOTE is a technique that has been known to out-perform both over-sampling and under-sampling.

1.4.5 Finally, I fitted a simple logistic regression model based on RFE feature selection of 50 features.

**2. Feature Matrix**

A variety of methods were used on feature selection, including RFE, chi square, gain / entropy. In the model supplied, we used RFE as it outperforms the other metrics of chi square. However as requested, I have calculated both the information gain as well as the entropy in the notebook supplied.

Entropy is defined as the amount of uncertainty of a data set. One being a totally random event and 0 being an event which we have full certainty about. Hence a two sided coin has an entropy of 1 and a two headed coin has an entropy of 0.

Entropy_1

H(S) depicts the entropy of the attribute S, p(x) refers to the probability of S having the value of x.

The gain is defined as the difference in entropy before and after a data set is split on an attribute.

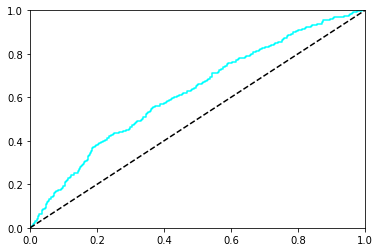
Information_Gain_1

IG refers to the information gain on attribute S, H refers to the entropy before the split while the summation refers to the information of the data after splitting on attribute S.

When you run the notebook, you will be able to find the various gains calculated on the various variables.

**3. Model Evaluation**

The data was tested on a variety of models, such as AdaBoost (with Decision Trees), xgboost and Balanced Bagging Classifier. The final pipeline I used was using Adaboost for feature selection, followed by Logistic Regression. The results yielded the following ROC curve:



The AUC score is 0.612 and the gini score is 0.224.

The precision / recall scores are:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **precision** | **recall** | **f1-score** | **support** |
| **0** | 0.97 | 0.62 | 0.76 | 10809 |
| **1** | 0.06 | 0.60 | 0.11 | 456 |
| **avg / total** | 0.94 | 0.62 | 0.73 | 11265 |

The confusion matrix is:

|  |  |
| --- | --- |
| 6715 | 4094 |
| 181 | 275 |

**The model trained has many additional areas for improvement. Here are some ways that can be further used to improve the model.**

The amount of feature engineering was at a minimal. I used a blanket wide labeler to convert all categorical data. Many features such as dates / categorical data can be further transformed to entities that are more meaningful.

Other than the suggested features provided, little to no domain knowledge was being used in the modelling process. If I had incorporated additional domain knowledge, it will most likely improve the model. However, I do believe in using latent / hidden variables in machine learning and would probably be applying other techniques such as CNN applied here to get better modeling results (as an intellectual exercise).