MAY 2020

RISK & FRAUD ANALYTICS

Master in Business Analytics & Big Data

Diego Cuartas

Madrid, España

Fraud Modeling Challenge

Objective

This document is a logbook that registers the methodology and all the steps I executed during the Fraud Modeling Challenge. I created a Machine Learning Pipeline that covers:

1. Libraries
2. Data Loading
   a. DEV
   b. OOT
3. Data Preparation
   a. OOT Censoring
   b. Dummy Variables
   c. NaN Imputer - OOT
4. Baseline Model
5. Parameter Optimization
6. Optimized Model
7. Genetic Algorithm
8. Final Model - GA Variables
9. Scoring

FIRST SUBMISSION

Executing the baseline document provided with the challenge:

1. Replacing NaN values in DEV & OOT with 0
2. Manually selected 4 variables: 1, 22, 25, 65
3. Fitting and Predicting with LogisticRegression

RESULT SUBMISSION:
KS2 = 0.12548
GINI = 0.11813
GRADE = 2.974

SECOND SUBMISSION

Executing a new model with Random Forest with all the variables:

1. Replacing NaN values in DEV & OOT with 0
2. Kept all the variables, no additional transformations
3. Fitting and Predicting with RandomForest

RESULT SUBMISSION:
KS2 = 0.34441
GINI = 0.47716
GRADE = 8.161

Analysis: As mentioned in **hint #5** We needed to implement a machine learning method that could use its robustness to detect changing behaviour from fraudsters combined with its predictive power to overfit and capture outliers as possible fraud behaviour. Random Forest is an ensemble method of decision trees that seemed to be a good option to try and as expected the KS and GINI scores greatly increased.

THIRD SUBMISSION

Executing the same Random Forest and transforming the DEV & OOT:

1. Replacing NaN values with SimpleImputer(Mean)
2. Solving Censoring problem
3. Dummy creation for 3 Categorical-Nominal variables
4. Kept all the variables
5. Fitting and Predicting with RandomForest

RESULT SUBMISSION:
KS2 = 0.31725
GINI = 0.43448
GRADE = 7.518

Analysis: As mentioned in **hint #6** I implemented the Min-Max Censoring method as a good practice to avoid finding values in OOT that do not occur in DEV. This actually reduced the score obtained in the previous submission but I decided to keep it for future submissions. As mentioned in **hint #7** I imputed the NAN values with the average value for OOT. Then I transformed only the 3 categorical-nominal variables to dummy. Random Forest (Tree base methods) can interpret the ordinality, so I did not apply the method to all the categorical variables.

FOURTH SUBMISSION

Optimizing the Random Forest hyper-parameters with the previous transformations:

1. Replacing NaN values with SimpleImputer(Mean)
2. Solving Censoring problem
3. Dummy creation for 3 Categorical-Nominal variables
4. Kept all the variables
5. Best parameters grid search for RandomForest
6. Fitting and Predicting with Optimized RandomForest

RESULT SUBMISSION:
KS2 = 0.37919
GINI = 0.49477
GRADE = 8.986

Analysis: My first grid search approach considered a small grid of parameters trying to improve the model score. **max_depth** in [70, 80, 90, 100] - **n_estimators** in [5000, 6000, 7000] - **max_features** in ["log2"] - **min_samples_split** in [2,4] - **min_samples_leaf** in [2] - **bootstrap** in [False].
The best combination of parameters after extracting the scores from the competition server: {'max_depth': 80, 'n_estimators': 6000, 'max_features': 'log2', 'min_samples_split': 4, 'min_samples_leaf': 2, 'bootstrap': False}. As expected Hyper-Parameter optimization was key to increase the score.

FINAL SUBMISSION

Applying Genetic Algorithm with the Random Forest hyper-parameters recommended by the professor and the previous transformations:

1. Replacing NaN values with SimpleImputer(Mean)
2. Solving Censoring problem
3. Dummy creation for 3 Categorical-Nominal variables
4. Best parameters for RandomForest (Professor's)
5. Executing the Genetic Algorithm against the server until reaching the maximum grade 10.0 with the best features
6. Fitting and Predicting with Final RandomForest and Feature Selection from Genetic Algorithm

RESULT SUBMISSION:
KS2 = 0.42551
GINI = 0.56024
GRADE = 10.00000

Analysis: In the final submission I executed two major steps that led me to the maximum score: first I tried a new set of recommended parameters in **hint #14:** n_estimators=10000, oob_score=True, random_state=25 ,max_features='log2', bootstrap = 'True', min_samples_split = 2, min_samples_leaf = 1, criterion = 'entropy', n_jobs=5.
Then I executed the advanced feature selection method: Genetic Algorithm(from **hint #11)**. It allowed to remove noisy variables and after several loops I obtained this set of selected features:

['ib_var_2', 'ib_var_3', 'ib_var_6', 'ib_var_7', 'ib_var_8', 'ib_var_9', 'ib_var_10', 'ib_var_11', 'ib_var_12', 'ib_var_13', 'ib_var_14', 'ib_var_15', 'ib_var_17', 'ib_var_18', 'ib_var_19', 'ib_var_20', 'ib_var_21', 'icn_var_22', 'icn_var_23', 'icn_var_24', 'ico_var_26', 'ico_var_28', 'ico_var_30', 'ico_var_32', 'ico_var_33', 'ico_var_34', 'ico_var_35', 'ico_var_36', 'ico_var_37', 'ico_var_38', 'ico_var_39', 'ico_var_40', 'ico_var_43', 'ico_var_44', 'ico_var_45', 'ico_var_46', 'ico_var_48', 'ico_var_49', 'ico_var_51', 'ico_var_53', 'ico_var_54', 'ico_var_55', 'ico_var_56', 'ico_var_57', 'ico_var_58', 'ico_var_61', 'ico_var_63', 'ico_var_64', 'if_var_65', 'if_var_66', 'if_var_67', 'if_var_69', 'if_var_72', 'if_var_73', 'if_var_74', 'if_var_77', 'if_var_78', 'if_var_79', 'if_var_80']

It's also important to highlight that many different feature selections were applied but the best results were with all the features before getting the best combination out of Genetic Algorithm.