13th - 17th May 2019 – Kuala Lumpur

Fraud Model Development and Deployment in SAS FM

Session 5: Model Development

Model Development

Topics

- Overview of model development process
- Appending signatures
- Exclusions
- Risk tables
- Segmentation
- Feature generation
- Feature selection
- Model training and selection

Modeling Development

Overview of Modeling Process

Model Development

Overview of Model Development Process

- SAS FM is a realtime scoring engine
- As each transaction arrives, several pre-processing, scoring and post-processing actions are executed on it before it exits the engine
- All actions run within a SAS datastep

```
data _null_;
  /* Read transaction from an incoming message stream */
  input stdin lrecl = &MAXLEN;
  %recode;
  %update_signatures;
  /* Subsequent steps within the %score module */
  %calculate_segment;
  %lookup_global_risk_factors;
  %generate_features;
  %impute_transform_features;
  %generate_model_score;
  %align_score;
  %reason_codes;
  %additional_processing;
run;
```

Model Development

Overview of Model Development Process

- Therefore it is best that the model development process is also setup as a series of steps that emulate the production process
- Various benefits in doing this:
  - WYSIWYG – from a code point of view, you can be guaranteed that the model will work the way as it did during development with minimal tweaks
  - Lets the developer keep a pulse of technical performance from the get go
    - Any abnormally time consuming steps can be borne out during the development phase itself
  - Allows you to have a reusable framework for model development

Model Development

Key Steps in the Model Development Process

- Setup basic scoring engine
- Introduce signatures - lookup, initialize, recode, update and persist
- Segmentation
- Generate data for risk table generation
- Generate risk tables
- Generate features
- Variable reduction / selection
- Model training and selection
- Score alignment
- Final model assembly

Modeling Development

Modeling Steps

Model Development

Step 1: Setup Basic Scoring Engine

- Skeleton code
- Error checks
- Scoring exclusions
- Replacement window
- Tag based filtering
- Load essential reference datasets as hashes
- Basic regression testing

Model Development

Step 2: Introduce Signatures

- Signature lookups
- Signature initialization
- Basic recodes
- Signature updates
- Signature persistence
- Advanced recodes

Model Development

Step 3: Segmentation

- Hard vs soft segmentation
- Soft segmentation
  - Feature generation
  - Clustering and segment selection
- Plugging the final segmentation scheme into the scoring engine
- A no segment model can also be developed in parallel
  - Can serve as a baseline for measuring performance
  - Can be used in score alignment

Model Development

Feature Generation

- Model features can be grouped into 2 main classes:
  - Global features - features based on the entire modeling population
  - Behavioral features - features based on individual behavior
- Global features involve taking an aggregate level view of the training data
  - Computed quantities are generally static, stored in datasets and looked up by the model
  - Generally referred to as risk tables
- Behavioral features involve iterating through the signature for each entity
  - Represent the individual entities' behavior
- Can have variables that combine both concepts
  - E.g. average spend of customers at a given MCC; this involves computing the average spend at a MCC for each customer and then averaging these averages

Model Development

Step 4: Generate Risk Table Data

- Risk in its purest form is a classical technique to convert a categorical variable into a numeric variable
- E.g. relative risk of different POS entry modes, MCC, hour-of-day etc.
  - $hour\_of\_day\_risk(h) = \frac{no.of\ fraud\ transactions\ during\ hour\ h}{no.of\ transactions\ during\ hour\ h}$
- Various other global quantities can also be computed
  - E.g. different percentiles for fraud amounts at different hours of the day
  - Can be directly used as features or further processed to generate features
    - E.g. ratio of current amount / median fraud amount during this hour
- Quantities can be generated at the entire population level or at a different segment level (not necessarily the model segments)
  - E.g. hour of day risk for payments within the bank vs payments outside the bank

Model Development

Step 4: Generate Risk Table Data

- This step creates the necessary variables that are aggregated together in the risk table creation step

Model Development

Step 5: Generate Risk Tables

- Is a process done outside the modeling step
- Result is a set of datasets later referenced by the model

Model Development

Preprocessing Prior to Feature Generation

- We will perform some additional processing on the dataset at this stage:
  - Truncation of fraud window
  - Down-sampling the goods
- When training the model, it is preferable to place emphasis on the early part of the fraud episode
- To achieve this we truncate the fraud episode to contain only the early parts of the fraud
- Review statistics on the run length of fraud episodes to determine criteria
  - For e.g. if median length of fraud episodes is 5 events, then truncating to the first 5 events will result in keeping the entire fraud episode for 50% of the fraud cases and up to 5 events for the remaining 50% of the fraud cases
  - Can be done by segment for soft segmentation schemes

Model Development

Preprocessing Prior to Feature Generation

- The goods : bads ratio at a transaction level will still be very high
  - Remember, our prior downsampling was done at an entity level
  - Truncating the frauds will make this worse
- However a good model cannot be produced with this level of imbalance
- Therefore we will further down sample the data to bring the transaction level goods : bads to a fixed ratio (usually > 10:1 but < 30:1 depending on the problem)
- Goods are chosen randomly at a transaction level to achieve this ratio
- However the goods are not thrown out of the dataset
  - We still need them to populate the signatures correctly
  - Just set a marker or use a hash to remember which transactions to keep in the final dataset

Model Development

Preprocessing Prior to Feature Generation

- Why perform this preprocessing prior to feature generation and not prior to model training?
- Feature generation is a time consuming process
- We can skip the feature generation step on the good transactions that were not sampled in and the bad transactions that were dropped
  - Provides enormous savings in processing since feature generation is the most time consuming step
- We will use some flags to indicate that the dropped transactions should update the signature, but need not generate the features during the feature generation step

Model Development

Step 6: Generate Features

- Risk tables created in the previous step are loaded as hashes in the scoring engine
- Various global features are created from these tables
- Also signatures are iterated through to created behavioral features
- Resulting datasets with the appended features should be written out by segment
  - Since we will perform feature selection by segment

Model Development

Step 7: Feature Selection

- Feature selection performed separately for each segment
  - De-duplication
  - Transformations
  - Imputation
  - Normalization
  - Correlation
  - Information gain
  - KS distance
  - Regression
  - PCA
- Final outcome is a dataset per segment containing the chosen features and target values ready for training the model

Model Development

Step 8: Model Training

- Pick your favorite method
- We need the model to produce a score between 1 and 999
  - Scale probabilities appropriately
- It is OK to sacrifice some model performance for better generalization
  - Overfitting should be avoided at all costs
  - Model can very quickly degrade in production

Model Development

Step 9: Score Alignment

- Scores from each segment model won't have the same 'meaning'
- We need to 'align' the scores such that a given score means the same according to some definition
  - False positive rate, detection rate, outsort rate etc.
- In order to align the scores, the entire training dataset should be scored
  - Not the downsampled & truncated set we created prior to feature generation
- Also entity level sampling rates should be factored when generating the tables used for alignment
- Usually align to the no segment model
  - If it was not generated, then use the largest segment

Model Development

Step 10: Final Model Assembly

- Trimming unused variables
  - A hard problem
- Breaking the flow into individual modules
- Package