# CardFlow

Implement a set of fixed processes (see Competitive Product Comparison - Built-in Fixed Workflow). It divides the scorecard development process into 10 stages: data reading, equal frequency binning, feature pre-filtering, monotonic and U-shaped recommendations, optimal binning, WOE conversion, feature filtering, model building, scorecard making, and report generation. The results of each stage will be automatically saved after execution. You can resume or update the previous results from any stage, and restarting the computer will not lose the results of the steps that have been calculated.

For example: cardflow.start(start\_step=1,end\_step=10) executes all steps. After execution, if you change the configuration file, such as changing the feature screening conditions, which affects the results of step 7 and later, and need to update 7-10, you only need to execute:

Just use cardflow.start(start\_step=7,end\_step=10). The previous results do not need to be re-executed.

For most users, after the configuration file is configured, CardFlow is the only component you need to interact with. The interaction method is very simple:

cardflow.start(start\_step=a,end\_step=b)

# Bins

Calculate the optimal binning split point. The optimal split point calculated by Bins is a global optimal analytical solution with mathematical proof. For categorical variables, including ordered and unordered categories, the global optimal analytical solution with mathematical proof can also be calculated. Its main functions are:

1. The global optimal solution can be found under unconstrained or constrained conditions. Constraints supported: monotonic constraints (automatically determine increasing or decreasing), monotonic decreasing constraints, monotonic increasing constraints, U-shaped constraints (automatically determine convex or concave), and automatically set appropriate constraints (automatically determine monotonic decreasing constraints, monotonic increasing constraints, U-shaped constraints convex, and U-shaped constraints concave).
2. Find the global optimal solution for ordered categorical variables under constraints.
3. Use " minimum difference in event rates between adjacent bins " instead of "information gain" or "chi-square value" to suppress the formation of bins with too small differences. Users can intuitively feel the size of the difference between bins. Categorical variables also support this function.
4. The minimum value of the first bin is not replaced with negative infinity, and the maximum value of the last bin is not replaced with positive infinity. The meaning is: the ignored abnormal values will not be covered up due to the extension of extreme values to infinity. At the same time, ScoreConflow provides a complete mechanism to deal with the problem of online values exceeding the modeling boundary value. It solves the common contradiction between discovering special values as early as possible during data analysis and covering up special values in online applications (but timely warning).
5. The concept of wildcards is introduced to solve the problem that the online values of categorical variables exceed the modeling range.
6. Supports multi-process parallel computing.
7. Supports binning with weighted samples.

In most cases, users do not need to interact directly with Bins components. Since ScoreConflow is designed to be pluggable, advanced users can use Bins modules separately like any python module.

# Reg\_Step\_Wise\_MP

It is a linear two-way stepwise regression and logistic two-way stepwise regression implemented in Python, which adds the following features compared to traditional two-way stepwise regression:

1. When performing stepwise variable selection for logistic regression, AUC, KS, and LIFT indicators can be used instead of AIC and BIC indicators. For some businesses, AUC and KS are indicators that are more suitable for business scenarios. For example, in the sorting business, the model built using the KS indicator has the advantage of using fewer variables but the KS of the model does not decrease on multiple test sets according to past experience.

2. When performing stepwise variable selection, use other data sets to calculate model evaluation indicators instead of using the modeling data set. Especially when the amount of data is large and there is a validation set in addition to the training set and test set, it is recommended to use the validation set to calculate evaluation indicators to guide variable selection. This helps reduce overfitting.

3. Support the use of partial data to calculate model evaluation indicators to guide variable selection. Scenario example: If the business needs to maintain a certain pass rate of N%, then the bad event rate of the first N% samples can be minimized, and all samples do not need to participate in the calculation. According to past experience: In appropriate scenarios, using partial data as evaluation indicators to select fewer variables than using all data, but the indicator that users are concerned about does not decrease in multiple test sets. Because the model only focuses on sample points that are easier to distinguish at the head, business goals can be achieved without too many variables.

4. Supports setting multiple conditions. Variables must meet all conditions at the same time before they can be included in the model. Built-in conditions include: P-Value, VIF, correlation coefficient, coefficient sign.

5. Supports specifying variables that must be entered into the model. If the specified variables to be entered into the model conflict with the conditions in 4, a complete mechanism is designed to solve the problem.

6. The modeling process is output to EXCEL, recording the reasons for deleting each variable and the process information of each round of stepwise regression.

7. When the number of configured CPU cores is greater than 1, multi-process parallel computing stepwise regression is automatically started.

In most cases, users do not need to interact directly with the Reg\_Step\_Wise\_MP component. However, because ScoreConflow is a pluggable component, advanced users can use the Reg\_Step\_Wise\_MP module separately like any other python module.

# Cutter

Perform equal frequency segmentation or segmentation according to specified split points, which has the following enhancements over the built-in segmenter in Python:

1. Mathematically provable analytical solution with minimum global error.
2. All split points are derived from the original values.
3. More humane support for left closed and right open: the last group is right closed. The minimum and maximum values of each interval come from the original data, which is different from the built-in splitter in Python, which changes the extreme values at both ends of each group .
4. The globally optimal segmentation solution can also be given for extremely tilted data.
5. Supports weighted arrays.
6. Supports user-specified special values. Special values are grouped separately, and users can also configure multiple special values to be combined into one group.
7. Special values do not include null values, but if there are null values in the sequence, the null values will be automatically processed into a group.
8. Use the specified split point to cut the sequence. When the maximum or minimum value of the sequence exceeds the split point boundary , the maximum and minimum values of the split point will be automatically extended.

It is recommended to replace Python's built-in equal frequency segmentation component with Cutter.

# Filter

1. There are 6 built-in filters: single value proportion filter, IV filter, IV variation filter, correlation coefficient filter, PSI filter, and missing value filter.
2. Results and intermediate data are saved in Excel and integrated in the model report.
3. Supports user-defined filters, users only need to implement the filtering method. ScoreConflow will automatically integrate the results and intermediate data into the model report according to the interface specification.
4. The single value ratio filter, IV filter, and missing value filter support pre-filtering.
5. Supports users to specify the data sets that participate in filter calculation.
6. Users can specify variables to delete or retain.