Cap & Floor

Description

- Pricing methodology for removing the outliers of numeric variables of the dataframe.
- This method calculates the highest and the lowest percentile defined by the user, by column and substitute outliers for those values.
- e.g. If the user defines 95% as the highest percentile and 5% as the lowest then the left and right 5% tails will be replaced with those percentiles.

Arguments

- **input_floor:** Is the value that the user defines as the lowest percentile expressed in floating point type.
- **input_cap:** Is the value that the user defines as the highest percentile expressed in floating point type.
- **l_exclusiones:** Key variables that shall not enter to the function (like client number, account number, etc.).
- **sqlContext:** sqlContext defined by the user.
- **pipeline:** Mode of execution:
    - To run in pipeline mode pipeline = 1.
    - To run as a single function pipeline = 0.
- **lib_write:** Name of a library to write intermediate results. All results will be erased at the end of the process.
- **rnd_nbr:** Random integer to avoid collisions between other users. When used in pipeline mode, all functions to be executed must contain the same number.
- **df:** Dataframe(spark type) to be evaluated.

Value

This function returns a spark dataframe.

Details

Percentile can be calculated with the following formula:

$$n = \lceil (P/100) \times N \rceil.$$

“n” throws the position of the value that represents the % of information that we are searching with the percentile. This value then is searched in a list of the sorted data.

In the function only the extreme values that exceed the % of information that we need are replaced.

Before running Cap & Floor function:

Numclient	Balance_0
567356	15’000
438769	9’000
763890	3’000
989341	7’000
650021	8’000
891221	1’000
833297	2’000
993212	6’000
121897	300
121897	5’000

After running Cap & Floor function (with cap = .8 and floor = .2):

Numclie nte	Balance _0
567356	9’000**
438769	9’000
763890	3’000
989341	7’000
650021	8’000
891221	1’000
833297	2’000
993212	6’000
121897	1’000**
121897	5’000