Soft Credit Scoring with Loan Figure and Rate – XAI Model

Document History

Version	Date	Author	Description
1.0	12/06/2020	Karampreet Bahia	

Comments:

Table of Contents

1. Introduction

A Temenos AI Soft Credit Scoring with Loan Figure and Rate (“Pre-Approved Loans”) solution allows financial institutions to automatically offer loans to their existing customers, at a value determined to be readily affordable to the customer through a derivation of their net disposable income.

The net disposable income is determined by an explainable (XAI) cashflow prediction (fundsflow) model.

A Temenos AI predictive cash flow model allows financial institutions to predict the **monthly disposable income** for each customer, based on the past behaviour the individual and the broader portfolio of customers. The model works on a customer level – for customers with multiple accounts – the model provides a disposable income prediction on an aggregate level.

The information used by the cash flow model is the historical transactions for a customer from all their cash accounts. The model also requires a transaction label associated with each transaction to give an explainable understanding of the customer behaviour.

Temenos XAI leverages the intrinsically explainable nature of rules-based fuzzy models. Unlike opaque “black box” models, this approach to machine learning results in predictions that can be justified to a human being.

The models output gives information on the likely net cash flow over the next month for the customer and the explainable result highlights the drivers behind the prediction.

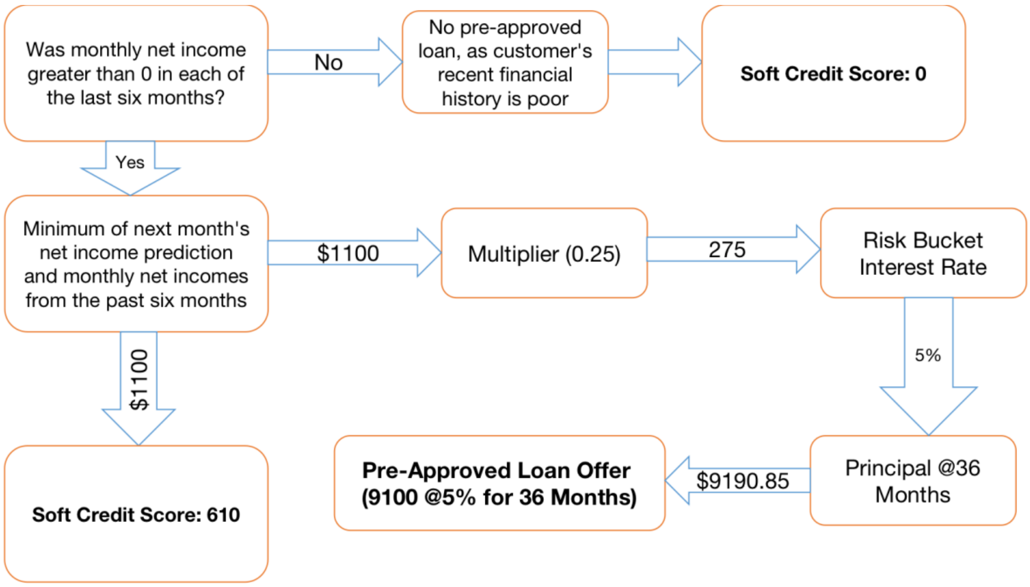
The model gives explainable outputs at the level of full population (model rule base), sub-population groups exhibiting similar behaviour (Rules applicable to individual churn risk buckets) and single customers (interpretable rules/drivers based output for each customer). In addition to being completely transparent – the Temenos AI platform offers easy deployment and real time monitoring of the model performance.

A soft credit score is derived from the regression model’s cashflow prediction score (call this **s**). Let **x** be the minimum monthly disposable income of the customer from the previous sixth months. Then, in terms of dollar values for the net disposable income, the soft credit score is equal to:

(0.1 * (minimum of ***x*** and ***s***)) + 500.

However the soft credit score value is capped to be no less than 0 and no greater than 1000.

The overall pipeline is as follows:



Here the Pre-Approved Loan value is calculated as follows. In the example diagram above, the customer is determined to be able to repay $275 per month (call this **a**), which over 36 months (call this number of months **d**) equates to $9900 in total principal repayment plus interest incurred. However it must be determined what proportion of this will be the principal amount. In the annual interest rate is 1.05 (I.e. 5%) as per the above diagram then the monthly interest rate (call this **i**) is 1.05^(1/12). The principal can then be derived as:

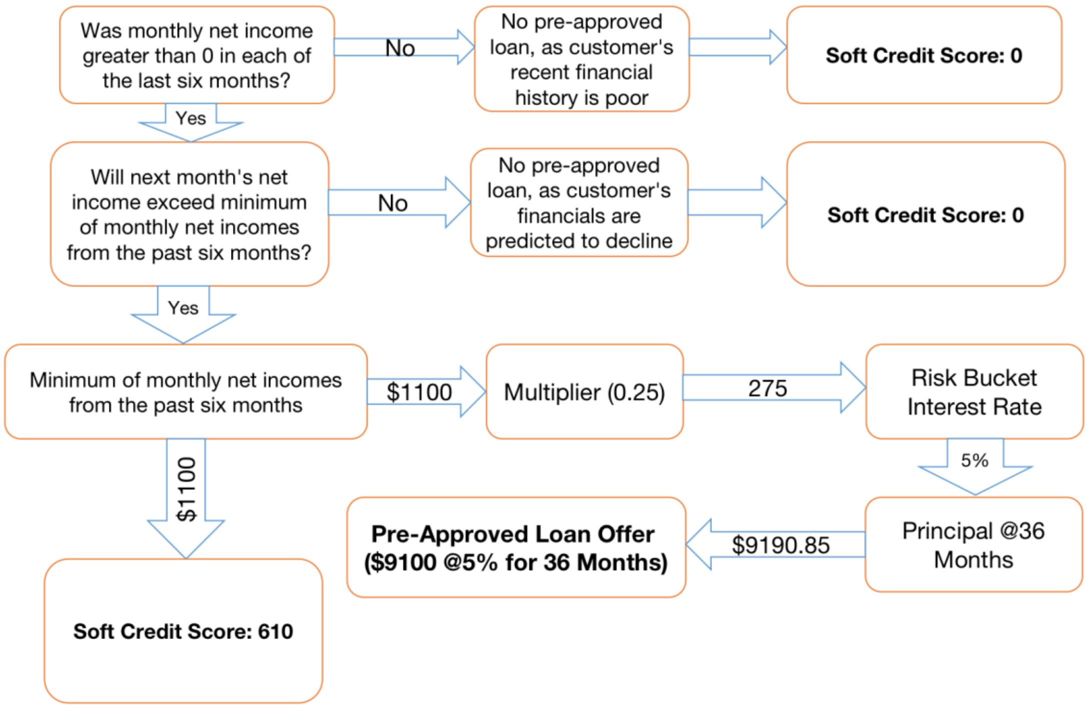
$$\frac{a(i^{-d}-1)}{(1-i)} = \$9190.85$$

The value is then rounded down to the nearest $100.

- The Temenos approach provides centrally created support for decision makers at the front line, using Explainable AI (XAI). This approach has been successfully proven in numerous well known UK financial companies and can provide speedy augmentation of the decision making processes already in place.
- Temenos deploys the system, in the Cloud or On-premise, and works with the bank to fine-tune existing, proven, fundsflow models in short timescales of days or weeks as per requirements.
- The XAI models can be used via a web-interface for the end users or the models can be integrated in the bank's platform as the inferences are exposed via API calls.
- The XAI models allow the bank to add new input features (considerations) to the existing model.
- XAI widgets enable the evaluation of impacts of additional terms and conditions on fundsflow.
- Additional terms and conditions can be suggested for each individual decision by the profiling of the various customer segments extracted from the XAI bucketing tools.
- Decisions can optionally be saved onto the XAI platform for future reference by tagging individual predictions with a customer ID
- Temenos XAI can monitor and analyse your population using XAI analytics dashboards for population and characteristic classification. This allows the user to monitor and change the rules and policies based on changes observed in the overall customer characteristics.

- Temenos XAI can provide actionable insights, and identify new approaches that are working, and provide feedback through an inference webpage to test different approaches for a given customer.
- Customers can be ranked according to their cashflow to prioritise which ones the bank should reach out to, in order to improve their financial situation
- Periodic reviews using actual cashflow outcomes for the customers processed through the XAI models can give insights into the effectiveness of any existing retention strategy. This can then serve as an input into generating better models going forward.

Note that an alternative pipeline is also available; where the model's output gives a binary classification prediction of whether the customer's monthly disposable income next month will exceed the minimum of their monthly net incomes from the past six months, and the explainable result highlights the drivers behind the prediction. However the structure of the final response, after post-processing to give the pre-approved loan offer, remains the same. The available input features and payloads to be sent to the XAI platform also remain the same as for the standard (regression) model. The pipeline for the binary model is as follows:



2. Business Requirements

Requirement Reference (ID)	Requirement Description
BR-01	The model can be adapted to work with any data source, if they can provide data to derive the model features.
BR-02	By default the model integrates the Temenos XAI platform with Temenos Data Warehouse and Temenos Analytics Platform for an end-to-end solution; however banks may also utilise the fundsflow model on the XAI platform as a standalone solution.
BR-03	For every valid inference, the model predicts a continuous value for the next month’s net disposable income.
BR-04	A linguistic label (target class) is associated with each continuous prediction; the possible labels are: extremely low, very low, low, medium, high, very high, extremely high
BR-05	The model output includes an ‘explanation’ in form of drivers and the rules behind every individual prediction.
BR-06	The model has a transparent global rule base for arriving at individual customer scores.
BR-07	The model supports changes in form of addition/modifications/deletions to its rule base conditional on the modifications involving valid input values.
BR-08	The model supports examining the changes in scores arising from a change to any/all the input value/s.
BR-09	The model supports storing of scores and inputs for all historical inferences.
BR-10	The model supports investigating distribution of scores from a ‘Live’ population against the original population used to build the model

BR-11	The model supports adding extra information/features via new/amended rules and generating an updated model to consider new information that was not available historically
BR-12	The model supports one-click deployment on the cloud.
BR-13	A net disposable income bucket analysis can be carried out on a population with known outcomes (Bucketing)
BR-14	The bucketing allows for customization on number of buckets

3. Current Functionality

There is no existing functionality within the Temenos product suite offering similar functionality/scope. At the customer end, existing functionality is specific to each bank, if any such framework exists at all.

4. Proposed Solution

4.1 Overview

The trained model provides responses using an API call with a unique model key for each deployment of the model. The model is called with selected variables from either Temenos Data Warehouse or an alternative datasource provided by the client – these are then passed through a pre-processing pipeline which prepares the final features for the model and sends the same to the model inference engine. The model response is then fed back to the caller. The process is described in detail in the next section.

4.2 Run Time

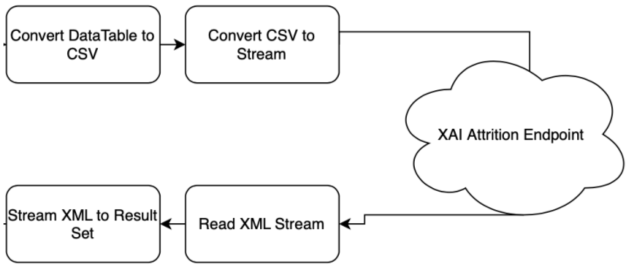


Figure 1: Typical Model Runtime Flow

Figure 1 shows an overview of the model runtime flow, if the bank is using the pre-built solution that integrates the Temenos XAI platform with Temenos Data Warehouse and Temenos Analytics Platform. At run time the request containing the 'raw information' from the data warehouse and other optional information hits the XAI endpoint. The XAI endpoint performs further processing to derive the feature set that is used to obtain inferences from the model. Some post-processing is applied to the model JSON response which is then streamed back to Analytics Platform as XML.

If a client wishes to utilise the XAI platform as a standalone solution, independent of the data warehouse and/or the Analytics Platform, the model runtime flow will change accordingly. The client must have a data source from which either they derive the exact fields used by the model, or Temenos XAI may otherwise implement a custom preprocessing layer to transform client inputs into model features; either way the bank will simply hit the fundsflow endpoint which will be configured accordingly. The structure of the response they receive may then also be modified, e.g. if the client would prefer JSON to be streamed back rather than XML.

Table 1 shows a feature set used for a fundsflow model. This is a subset of the full available feature set, which is listed below the table. Details on the definitions and derivations of each feature may be viewed in the Fundsflow User Guide, under Data Requirements. If a bank generates a project on the XAI platform with its own data, i.e. to generate a bespoke model rather use than a pre-built fundsflow model provided by Temenos, then they will be able to select the most predictive subset of features from the full set using the XAI Platform's feature selection functionality. As per the table, a feature set of size in the range 30-50 features is typically best.

raw	alias
essentials_expenses_1_month_diff_prevmonth	Essential Expenses 1 Month Change ($)
essentials_expenses_3_month_avg	Essential Expenses 3 Month Average
essentials_expenses_3_month_diff_prevmonth	Essential Expenses 3 Month Change ($)
expense_transactions_1_month_diff_prevmonth	Expenses 1 Month Change ($)

expense_transactions_3_month_avg	Expenses 3 Month Average
expense_transactions_3_month_diff_prevmonth	Expenses 3 Month Change ($)
expense_transactions_3_month_max	Expenses 3 Month Max
expense_transactions_3_month_min	Expenses 3 Month Min
expense_transactions_6_month_avg	Expenses 6 Month Average
expense_transactions_6_month_max	Expenses 6 Month Max
expense_transactions_6_month_min	Expenses 6 Month Min
fees_charges_expenses_3_month_avg	Fees/Charges 3 Month Average
fees_charges_expenses_3_month_diff_prevmonth	Fees/Charges 3 Month Change ($)
fees_charges_expenses_6_month_min	Fees/Charges 6 Month Min
income_transactions_1_month_diff_prevmonth	Income 1 Month Change ($)
income_transactions_3_month_avg	Income 3 Month Average
income_transactions_3_month_diff_prevmonth	Income 3 Month Change ($)
income_transactions_3_month_max	Income 3 Month Max
lifestyle_expenses_1_month_diff_prevmonth	Lifestyle Expenses 1 Month Change ($)
lifestyle_expenses_3_month_avg	Lifestyle Expenses 3 Month Average
lifestyle_expenses_3_month_diff_prevmonth	Lifestyle Expenses 3 Month Change ($)
net_transaction_1_month_diff_prevmonth	Monthly Net Transactions 1 Month Change ($)
net_transaction_3_month_avg	Monthly Net Transactions 3 Month Average
net_transaction_3_month_diff_prevmonth	Monthly Net Transactions 3 Month Change ($)
net_transaction_3_month_max	Monthly Net Transactions 3 Month Max
net_transaction_3_month_min	Monthly Net Transactions 3 Month Min
net_transaction_6_month_avg	Monthly Net Transactions 6 Month Average
net_transaction_6_month_diff_prevmonth	Monthly Net Transactions 6 Month Change ($)
net_transaction_6_month_max	Monthly Net Transactions 6 Month Max
net_transaction_6_month_min	Monthly Net Transactions 6 Month Min
online_expenses_3_month_avg	Online Spend 3 Month Average
online_expenses_3_month_diff_prevmonth	Online Spend 3 Month Change ($)
other_income_3_month_avg	Other Income 3 Month Average

other_income_3_month_diff_prevmonth	Other Income 3 Month Change ($)
other_income_3_month_max	Other Income 3 Month Max
regular_bills_expenses_3_month_avg	Bills/Regular Expenses 3 Month Average
regular_bills_expenses_3_month_diff_prevmonth	Bills/Regular Expenses 3 Month Change ($)
salary_income_1_month_diff_prevmonth	Salary/Main Income 1 Month Change ($)
salary_income_3_month_avg	Salary/Main Income 3 Month Average
salary_income_3_month_diff_prevmonth	Salary/Main Income 3 Month Change ($)
salary_income_6_month_avg	Salary/Main Income 6 Month Average
salary_income_6_month_min	Salary/Main Income 6 Month Min
regular_bills_expenses_1_month_diff_prevmonth	Bills/Regular Expenses 1 Month Change ($)
online_expenses_1_month_diff_prevmonth	Online Spend 1 Month Change ($)

Table 1

The full list of potential modelling features is:

expense_transactions_1_month_diff_prevmonth

expense_transactions_3_month_diff_prevmonth

expense_transactions_6_month_diff_prevmonth

income_transactions_1_month_diff_prevmonth

income_transactions_3_month_diff_prevmonth

income_transactions_6_month_diff_prevmonth

lifestyle_expenses_1_month_diff_prevmonth

lifestyle_expenses_3_month_diff_prevmonth

lifestyle_expenses_6_month_diff_prevmonth

essentials_expenses_1_month_diff_prevmonth

essentials_expenses_3_month_diff_prevmonth

essentials_expenses_6_month_diff_prevmonth

fees_charges_expenses_1_month_diff_prevmonth

fees_charges_expenses_3_month_diff_prevmonth

fees_charges_expenses_6_month_diff_prevmonth online_expenses_1_month_diff_prevmonth

online_expenses_3_month_diff_prevmonth

online_expenses_6_month_diff_prevmonth

regular_bills_expenses_1_month_diff_prevmonth

regular_bills_expenses_3_month_diff_prevmonth

regular_bills_expenses_6_month_diff_prevmonth

salary_income_1_month_diff_prevmonth

salary_income_3_month_diff_prevmonth

salary_income_6_month_diff_prevmonth

other_income_1_month_diff_prevmonth

other_income_3_month_diff_prevmonth

other_income_6_month_diff_prevmonth

net_transaction_1_month_diff_prevmonth

net_transaction_3_month_diff_prevmonth

net_transaction_6_month_diff_prevmonth

expense_transactions_3_month_max

expense_transactions_3_month_min

expense_transactions_3_month_avg

expense_transactions_6_month_max

expense_transactions_6_month_min

expense_transactions_6_month_avg

income_transactions_3_month_max

income_transactions_3_month_min

income_transactions_3_month_avg

income_transactions_6_month_max

income_transactions_6_month_min

income_transactions_6_month_avg

lifestyle_expenses_3_month_max

lifestyle_expenses_3_month_min

lifestyle_expenses_3_month_avg

lifestyle_expenses_6_month_max

lifestyle_expenses_6_month_min

lifestyle_expenses_6_month_avg

essentials_expenses_3_month_max

essentials_expenses_3_month_min

essentials_expenses_3_month_avg

essentials_expenses_6_month_max

essentials_expenses_6_month_min

essentials_expenses_6_month_avg

fees_charges_expenses_3_month_max

fees_charges_expenses_3_month_min

fees_charges_expenses_3_month_avg

fees_charges_expenses_6_month_max

fees_charges_expenses_6_month_min

fees_charges_expenses_6_month_avg

online_expenses_3_month_max

online_expenses_3_month_min

online_expenses_3_month_avg

online_expenses_6_month_max

online_expenses_6_month_min

online_expenses_6_month_avg

regular_bills_expenses_3_month_max

regular_bills_expenses_3_month_min

regular_bills_expenses_3_month_avg

regular_bills_expenses_6_month_max

regular_bills_expenses_6_month_min

regular_bills_expenses_6_month_avg

salary_income_3_month_max

salary_income_3_month_min

salary_income_3_month_avg

salary_income_6_month_max

salary_income_6_month_min

salary_income_6_month_avg

other_income_3_month_max

other_income_3_month_min

other_income_3_month_avg

other_income_6_month_max

other_income_6_month_min

other_income_6_month_avg

net_transaction_3_month_max

net_transaction_3_month_min

net_transaction_3_month_avg

net_transaction_6_month_max

net_transaction_6_month_min

net_transaction_6_month_avg

API Request Structure:

The API can be interacted with using a JSON payload, which will include a Model Key. A sample payload for a single inference is shown below:


```
{
  "data": {
    "expense_transactions_1_month_diff_prevmonth": -1000600,
    "expense_transactions_3_month_diff_prevmonth": -858146.03,
    "income_transactions_1_month_diff_prevmonth": -3900000,
    "income_transactions_3_month_diff_prevmonth": -3900000,
    "lifestyle_expenses_1_month_diff_prevmonth": -5111.02,
    "lifestyle_expenses_3_month_diff_prevmonth": -3226.21,
    "essentials_expenses_1_month_diff_prevmonth": -8882.679999999997,
    "essentials_expenses_3_month_diff_prevmonth": -8881.279999999999,
    "fees_charges_expenses_3_month_diff_prevmonth": -2212.5,
    "online_expenses_1_month_diff_prevmonth": -165188.38,
    "online_expenses_3_month_diff_prevmonth": -35650,
    "regular_bills_expenses_1_month_diff_prevmonth": -125000,
    "regular_bills_expenses_3_month_diff_prevmonth": -45847.36,
    "salary_income_1_month_diff_prevmonth": -23843.3,
    "salary_income_3_month_diff_prevmonth": -26541.64,
    "other_income_3_month_diff_prevmonth": -1677491.57,
    "net_transaction_1_month_diff_prevmonth": -3900000,
    "net_transaction_3_month_diff_prevmonth": -3900000,
```

```
    "expense_transactions_3_month_avg": -819.553,
    "income_transactions_3_month_avg": -302.093,
    "lifestyle_expenses_3_month_avg": 0,
    "essentials_expenses_3_month_avg": 0,
    "online_expenses_3_month_avg": 1,
    "regular_bills_expenses_3_month_avg": 0,
    "salary_income_3_month_avg": 0,
    "salary_income_6_month_avg": 1,
    "other_income_3_month_avg": 0,
    "net_transaction_3_month_max": -49321.86,
    "net_transaction_3_month_min": -1426373.43,
    "net_transaction_3_month_avg": -419707.83,
    "net_transaction_6_month_avg": -152478.955    "LG_EXTERNAL_ID": "Customer_123"
  },
  "modelKey": "abc-123-456-789-def",
  "detailed": "false",
  "dataSource": "live"
}
```


The “detailed” flag controls the level of information returned by the response. If set to “true”, information on all triggered rules and their associated weights are returned. This option is useful in case the API response needs to be consumed in a bespoke UI layer.

The model also supports batch inference and a sample batch request payload is shown below:


```
{
  "batch": [{
    "expense_transactions_1_month_diff_prevmonth": -1000600,
    "expense_transactions_3_month_diff_prevmonth": -858146.03,
    "income_transactions_1_month_diff_prevmonth": -3900000,
    "income_transactions_3_month_diff_prevmonth": -3900000,
    "lifestyle_expenses_1_month_diff_prevmonth": -5111.02,
    "lifestyle_expenses_3_month_diff_prevmonth": -3226.21,
    "essentials_expenses_1_month_diff_prevmonth": -8882.679999999997,
    "essentials_expenses_3_month_diff_prevmonth": -8881.279999999999,
    "fees_charges_expenses_3_month_diff_prevmonth": -2212.5,
    "online_expenses_1_month_diff_prevmonth": -165188.38,
    "online_expenses_3_month_diff_prevmonth": -35650,
    "regular_bills_expenses_1_month_diff_prevmonth": -125000,
    "regular_bills_expenses_3_month_diff_prevmonth": -45847.36,
    "salary_income_1_month_diff_prevmonth": -23843.3,
    "salary_income_3_month_diff_prevmonth": -26541.64,
    "other_income_3_month_diff_prevmonth": -1677491.57,
    "net_transaction_1_month_diff_prevmonth": -3900000,
    "net_transaction_3_month_diff_prevmonth": -3900000,
    "expense_transactions_3_month_avg": -819.553,
    "income_transactions_3_month_avg": -302.093,
    "lifestyle_expenses_3_month_avg": 0,
    "essentials_expenses_3_month_avg": 0,
    "online_expenses_3_month_avg": 1,
    "regular_bills_expenses_3_month_avg": 0,
    "salary_income_3_month_avg": 0,
    "salary_income_6_month_avg": 1,
    "other_income_3_month_avg": 0,
    "net_transaction_3_month_max": -49321.86,
    "net_transaction_3_month_min": -1426373.43,
    "net_transaction_3_month_avg": -419707.83,
    "net_transaction_6_month_avg": -152478.955
    "LG_EXTERNAL_ID": "Customer_123"
  },
  {
     "expense_transactions_1_month_diff_prevmonth": -1000600,
    "expense_transactions_3_month_diff_prevmonth": -858146.03,
    "income_transactions_1_month_diff_prevmonth": -3900000,
```

```
    "income_transactions_3_month_diff_prevmonth": -3900000,
    "lifestyle_expenses_1_month_diff_prevmonth": -5111.02,
    "lifestyle_expenses_3_month_diff_prevmonth": -3226.21,
    "essentials_expenses_1_month_diff_prevmonth": -8882.679999999997,
    "essentials_expenses_3_month_diff_prevmonth": -8881.279999999999,
    "fees_charges_expenses_3_month_diff_prevmonth": -2212.5,
    "online_expenses_1_month_diff_prevmonth": -165188.38,
    "online_expenses_3_month_diff_prevmonth": -35650,
    "regular_bills_expenses_1_month_diff_prevmonth": -125000,
    "regular_bills_expenses_3_month_diff_prevmonth": -45847.36,
    "salary_income_1_month_diff_prevmonth": -23843.3,
    "salary_income_3_month_diff_prevmonth": -26541.64,
    "other_income_3_month_diff_prevmonth": -1677491.57,
    "net_transaction_1_month_diff_prevmonth": -3900000,
    "net_transaction_3_month_diff_prevmonth": -3900000,
    "expense_transactions_3_month_avg": -819.553,
    "income_transactions_3_month_avg": -302.093,
    "lifestyle_expenses_3_month_avg": 0,
    "essentials_expenses_3_month_avg": 0,
    "online_expenses_3_month_avg": 1,
    "regular_bills_expenses_3_month_avg": 0,
    "salary_income_3_month_avg": 0,
    "salary_income_6_month_avg": 1,
    "other_income_3_month_avg": 0,
    "net_transaction_3_month_max": -49321.86,
    "net_transaction_3_month_min": -1426373.43,
    "net_transaction_3_month_avg": -419707.83,
    "net_transaction_6_month_avg": -152478.955
    "LG_EXTERNAL_ID": "Customer_456"
  }],
  "modelKey": "abc-123-456-789-def",
  "detailed": "false",
  "dataSource": "live"
}
```


API Response Structure:

The user has the option to choose between a standard response and a detailed response. The standard response contains a model score and a URI to view the explainable output.

The standard response is shown in Figure 2.

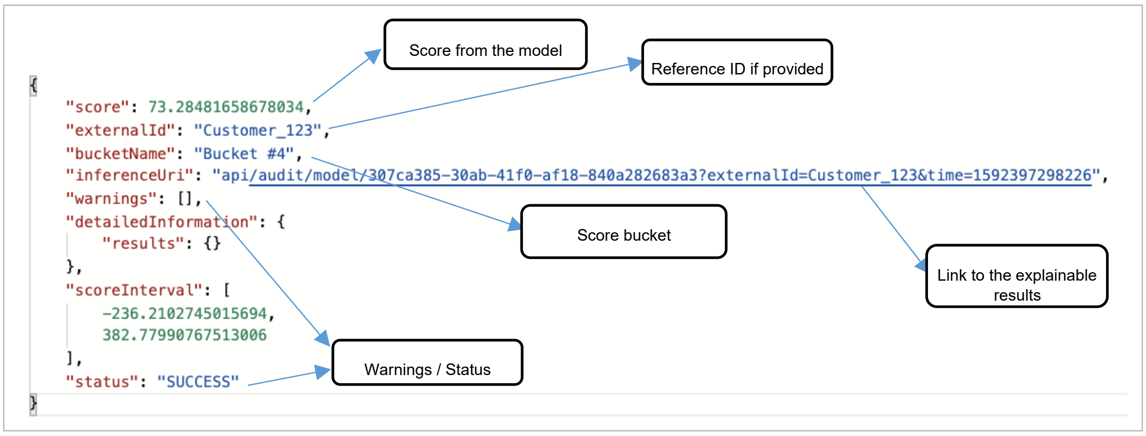


Figure 2 Basic API Response Structure

In Figure 2, the key ‘detailedInformation’ does not contain any results ({‘results’:{}}. Detailed information can be used to format the response in a custom UI or to perform further analysis on individual rules / drivers for customer inferences. Detailed response can be activated by setting the flag 'detailed' to 'True' in the API request payload. An example of a part of results contained within 'detailedInformation' is shown in Figure 3 . All other response information is the same as contained within the standard response.


```
{
    "score": 73.28481658678034,
    "externalId": "Customer_123",
    "bucketName": "Bucket #4",
    "inferenceUri": "api/audit/model/307ca385-30ab-41f0-af18-840a282683a3?externalId=Customer_123&time=1592397691781",
    "warnings": [],
    "detailedInformation": {
        "results": {
            "fuzzyDrivers": [
                {
                    "rank": 1,
                    "name": "extremely low",
                    "overall": -41.517105030757506,
                    "rules": [
                        {
                            "feature": "essentials_expenses_3_month_diff_prevmonth",
                            "values": [
                                "low"
                            ],
                            "actualValue": "-8881.279999999999",
                            "consonances": [
                                {
                                    "firingStrength": 1.0,
                                    "dominance": 2.2767399505533236E-4,
                                    "ratioAsPercentage": -2.385822428410898,
                                    "antecedent": [
                                        {
                                            "field": "essentials_expenses_3_month_diff_prevmonth",
                                            "value": "low"
                                        },
                                        {
                                            "field": "net_transaction_6_month_avg",
                                            "value": "low"
                                        },
                                        {
                                            "field": "net_transaction_3_month_min",
                                            "value": "low"
                                        }
                                    ],
                                    "ruleId": 106,
                                    "dominanceRating": 1.0,
                                    "label": "extremely low"
                                }
                            ],
                            "dissonances": [],
                            "weight": 0.0
                        },
```


Figure 3 Partial Detailed Response

Recall that the results from batch inferencing are then streamed back (either as XML to Temenos Analytics or as XML/JSON to the client directly). The streamed results also contain three additional fields if the bank has also acquired a Temenos Pre-Approved Loans XAI model, namely "approvedLoanValue", "interestRate" and "softCreditScore". This is shown in Figure 4, which shows a single inference extracted from a streamed response (without detailedInformation for brevity). Let **x** be the minimum monthly disposable income of the customer from the previous sixth months. Then, in terms of dollar values for the net disposable income, the softCreditScore is equal to:

(0.1 * (minimum of ***x*** and ***s***)) + 500.

However the softCreditScore value is capped to be no less than 0 and no greater than 1000.

```
▼<result>
  <externalId type="str">Customer_123</externalId>
  <bucketName type="str">Bucket #1</bucketName>
 ▼<inferenceUri type="str">
    api/audit/model/40dcda78-ecd8-42dd-9080-038cd6234d83?externalId=Customer_123&time=1592244188750
  </inferenceUri>
  <score type="float">124.43525</score>
  <approvedLoanValue type="str">4942</approvedLoanValue>
  <interestRate type="float">1.25575</interestRate>
 ▼<detailedInformation type="dict">
    <results type="dict"/>
  </detailedInformation>
  <inferenceResultEncoded type="float">-2147483445.0</inferenceResultEncoded>
  <inferenceResultDecoded type="str">0</inferenceResultDecoded>
 ▼<mapOutputRatios type="dict">
    <n0 type="float">1.0</n0>
  </mapOutputRatios>
  <warnings type="list"/>
  <status type="str">SUCCESS</status>
</result>
```


Figure 4 Single XML Response

5. Use Cases

5.1 Use Case 1

Description

User Makes an API call to the fundsflow pipeline with a valid payload. The model returns a response with a continuous score and a URI where the logged in user can see the explainable results for the response including the top drivers and rules.

Business Application

Banks can integrate the interaction with the fundsflow model API based easily within their pipeline. They can also choose a level of customisation based on their own requirements as the API response contains all necessary information for them to be able to consume the results within their own UI, if needed.

Variations	Expected Results:
“detailed” flag set to true in request	Response contains a detailed set of rules fired for the request
Feature value for continuous features derived from the payload falls outside the ‘seen’ values by the model (see Appendix C for seen range for relevant features)	The response contains warning/s about the values which fall outside the range but will still return a valid response with score and URI
Prediction score	Model returns an error response.
Fields other than prediction score cannot be derived from the payload	Model treats the missing values as ‘Null’ and returns a valid response

5.2 Use Case 2

Description

For every valid response returned by the model – the inference URI shows a detailed ‘Rules’ view of the rules applicable to the customer’s inference and a condensed ‘Driver’ view of the most important features-value combinations applicable to the inference.

Business Application

The fundsflow model’s explainable decisions assist the end-user by presenting its reason for the output score on a driver (feature) and rule basis.

Each *IF-THEN* rule has an underlying dominance, which is derived from

- the prevalence of the rule within all instances used to train the model and,

- the accuracy of the rule

For any given instance, then all the rules that are applicable to that particular case are collated according to their '*firing strength*'. The firing strength of a rule depends on to what degree an instance satisfies the rule condition.

Finally, from all the rules that have '*fired*', all the premise that make up the rules are combined to get a driver level importance.

An example of the two views from the same customer inference is shown in Figure 5 and Figure 6.

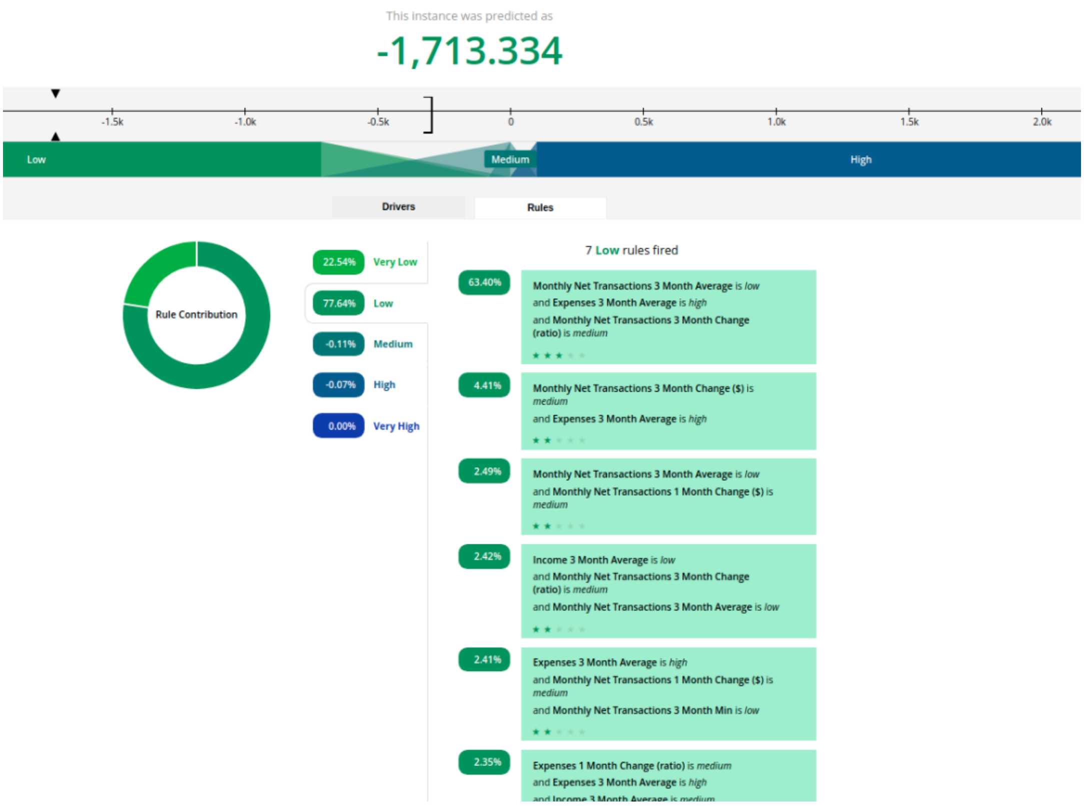


Figure 5 Rule View

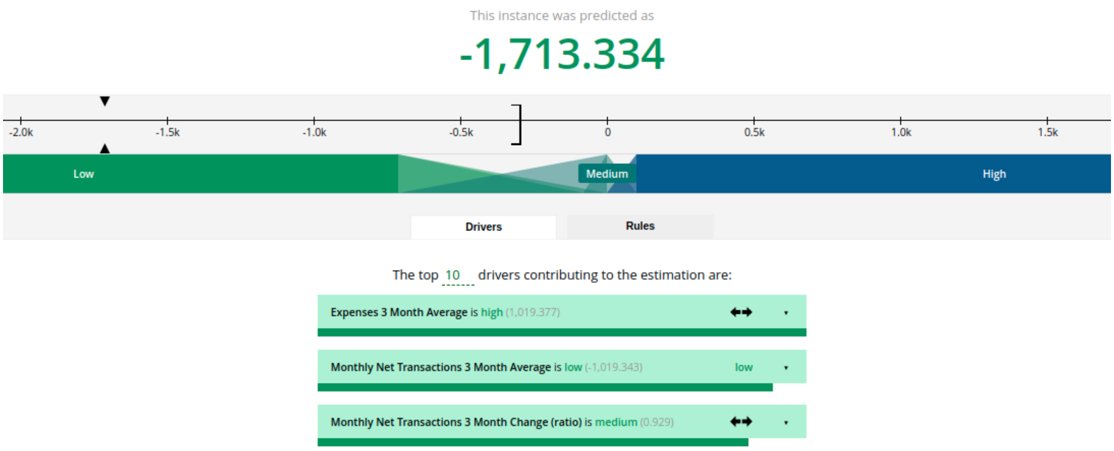


Figure 6 Driver View

5.3 Use Case 3

Description

The model has a transparent global rule base for arriving at individual instance scores and the same can be viewed by going to the model page and then clicking 'Analysis' → 'Rules'. The rules are by default ordered by their importance to the model. Each rule in the rulebase shows:

- The 'Antecedent' part (the premise/s which make up the rule),
- The rule statistics (not meaningful for models generated using expert opinion and synthetic data) and
- The class that the rule points towards.

Business Application

Transparency of model logic allows end-users to have faith in the model actions as well as comply with any regulatory requirements. The fundsflow model's fully transparent rule base captures complex interactions between input features while being explainable.

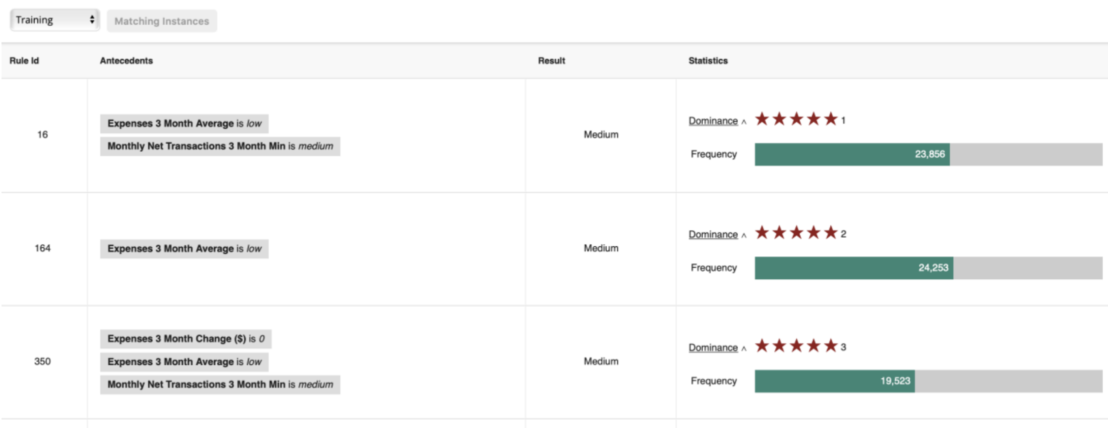


Figure 7 - Example of High Dominance Rules in Rule Base

5.4 Use Case 4

Description

The model supports changes in form of addition/modifications/deletions to its rule base conditional on the modifications involving valid input values.

Business Application

Banks can be in full control of the model by augmenting the initial rule base with their own domain expertise. This also allows them to be fully in control even if the economic or customer landscape has changed and they would like the rule base to reflect the changes before the actual customer data starts picking up on the changes. This allows the banks to be always in sync with the economic and customer cycles by being flexible around their cashflow modelling approach.

Changes to a model's rule base can be made via the scenario functionality offered by the Temenos XAI platform. A new scenario can be created for a fundsflow model by adding/modifying/deleting rules from the existing rule base.

From the model page – going to 'Analysis' → 'Scenarios' → 'Create New Scenario' (Figure 8) will create a new scenario with a user specified name – which by default is the same as the underlying rule base.

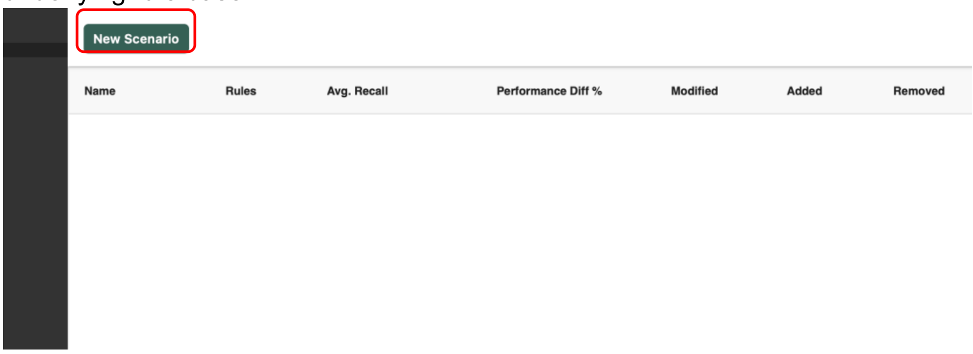


Figure 8 Create a new Scenario

Once a scenario has been created – rules can be added, deleted or modified.

- To add a rule, click 'Add Rule' button as shown in Figure 9 and Figure 100:

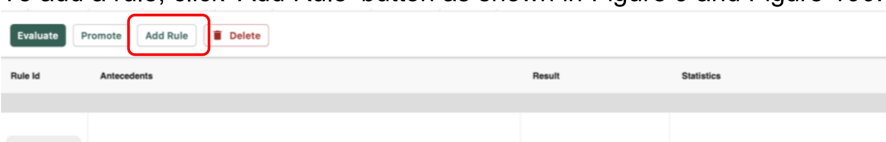


Figure 9 Add Rule Button

- In the dialogue box that appears select the feature along with the value that is applicable for the rule. If needed, select more premises for the Antecedent part of the rule. Select the right Target Class and the rule dominance. The rule dominance is between 1 and 5 with 1 being a weak rule and 5 being a dominant rule. Press Save Rule. This will add the rule to the scenario.

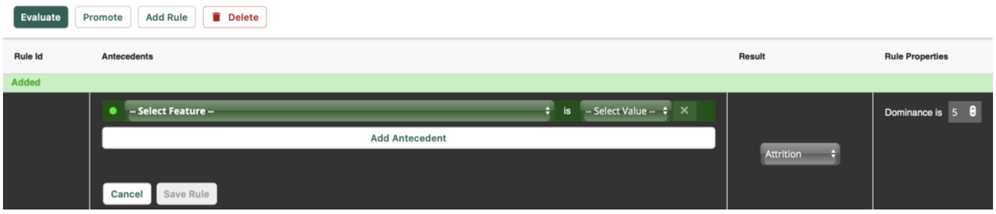


Figure 10 Adding a Rule

- To delete a Rule, select a rule by filtering on Rule Id or by manually finding the rule. Select Edit. On the resulting screen – select 'Delete Rule'. This will delete the rule from the scenario. (Figure 111)

Figure 11 Deleting a Rule

- To edit a rule, select the desired rule to edit as in previous step. Now change any/all of antecedents, associated feature values, rule dominance and Target Class. Once rule has been edited as desired – press 'Save Rule'. This will replace the original rule in the scenario with the edited rule.

Once all the desired changes have been made to a scenario. It can be evaluated for a shift in performance. The evaluation can be performed as often as a user likes – after every rule change or after some/all changes have been made. The evaluation is done on the main scenario screen by pressing the Evaluate button (

- Figure 122) with the performance change shown in a pop-up on the screen (Figure 133).

Figure 12 Evaluate Rule Button

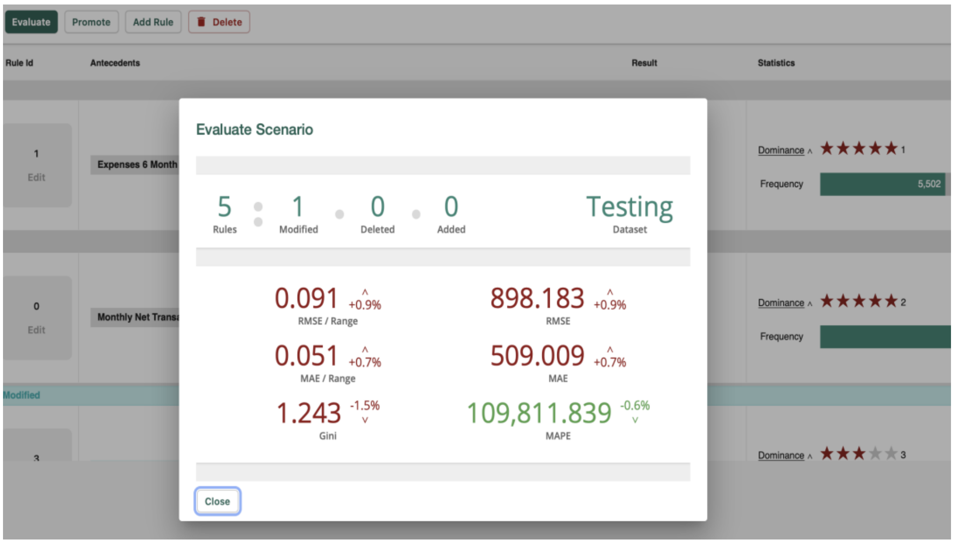


Figure 13 Evaluating the Scenario Performance

- At this stage the scenario has all the desired changes but it is not yet a model. To convert a scenario to a model, press the 'Promote' button on the main scenario screen (Figure 14) and it will be built as a model in the background and will appear in the models list once the build has finished.

Figure 14 Promote a Scenario to a Model

Variations	Expected Results:
Rule being added already exists in the rule base	The platform shows an error message saying the Rule already exists in the Rule Base and gives the Id of the existing Rule. Duplicate / conflicting rules cannot be added to the rule base.
Rule being added can be derived from combination of other rules in the rule base	The platform gives a warning that the rule can be derived from other rules – but still allows the user to add the rule to the rule base.

5.5 Use Case 5

Description

Understanding historical cashflow behaviour of customers allows the bank to consider and employ different strategies to retain customers in different cashflow buckets

Business Application

The bucketing tool allows a user to segment customers with known outcomes to be segmented in prediction score buckets and helps in understanding how each such bucket performed. The instances all scored by the model and are assigned continuous scores predicting net disposable income next month. The bucketing widget then groups the customers falling within a certain range – say 100 to 200 (dollars/currency units), and presents statistics like number of instances falling in this range. Custom statistics can be added by the user based on the information available within the record.

Bucketing can be accessed clicking on the model and then visiting 'Analysis' → 'Bucketing' as shown in Figure 16. The user is free to choose a plotted feature for the y-axis; typically this is left at the default of "Instance Count".

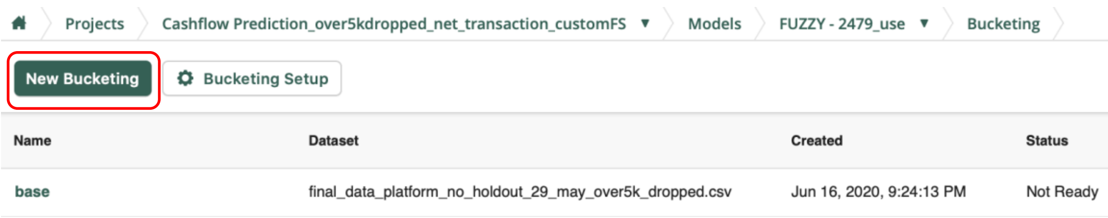


Figure 15 New Bucketing

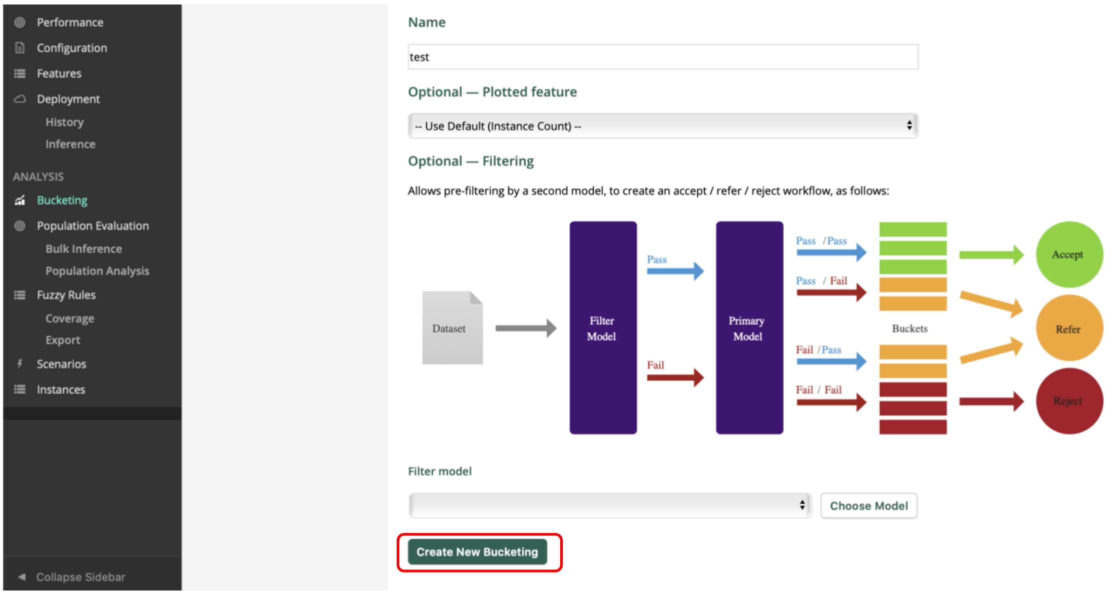


Figure 16 Create Bucketing

A screenshot of a bucketing is shown in Figure 17. The highlighted area shows where the option for changing bucket counts is listed. Note the options to edit the default bucketing are disabled once the model is deployed.

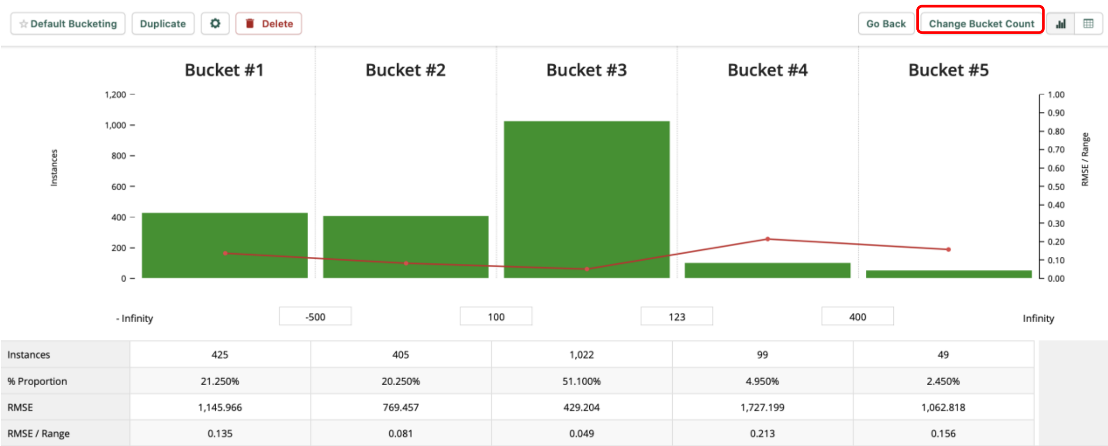


Figure 17 Sample Bucketing

Variations	Expected Results:
The data file for bucketing does not contain the target class / outcome column	The platform shows an error message - “The uploaded file is missing the expected target column, "Outcome".
The data file is missing feature column/s which were used to train the model.	The platform shows an error message – “An unknown error occurred while creating the bucketing.”

5.6 Use Case 6

Description

User can examine the changes in scores arising from a change to any/all the input value/s.

Business Application

The bank might wish to provide understandable actions which will enable the cashflow to be improved at the level of individual customers.

The features or attributes which need improving can be taken from the explanation provided for the score. Once the user knows which attributes for the customer are causing low/negative cashflow, the instance can be opened in the ‘Inference Viewer’ and the user can pass different values for the attributes that need improvement to see how this will affect the score.

As an example, consider the explainable output (driver view) in Figure 18 for an instance which scores -1713 (dollars). From the driver view, it can be seen that high expenses in the last three months is the most significant driver impacting the score.

Note also that if the user clicks on the triangle far to the right of driver name, a drop-down of corresponding rules is shown as per Figure 19.

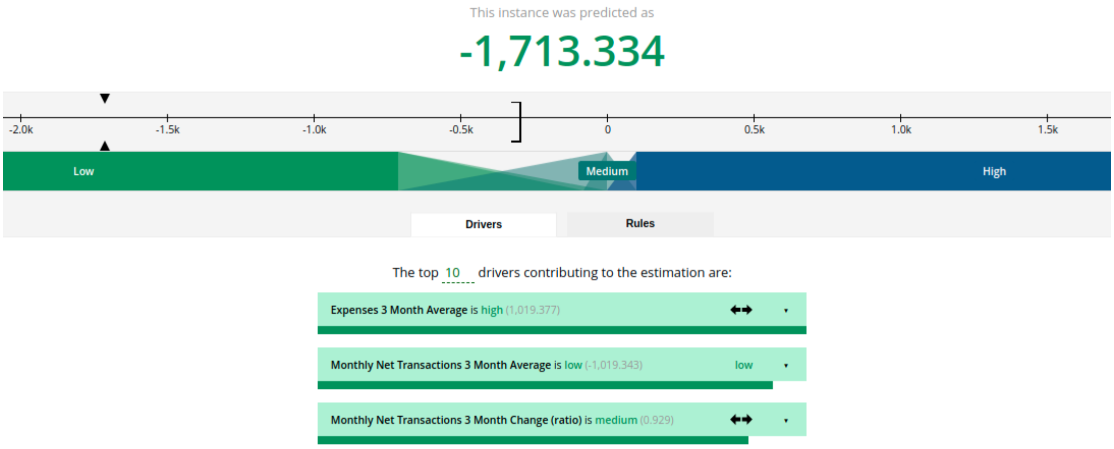


Figure 18 Customer with Poor Predicted Cashflow

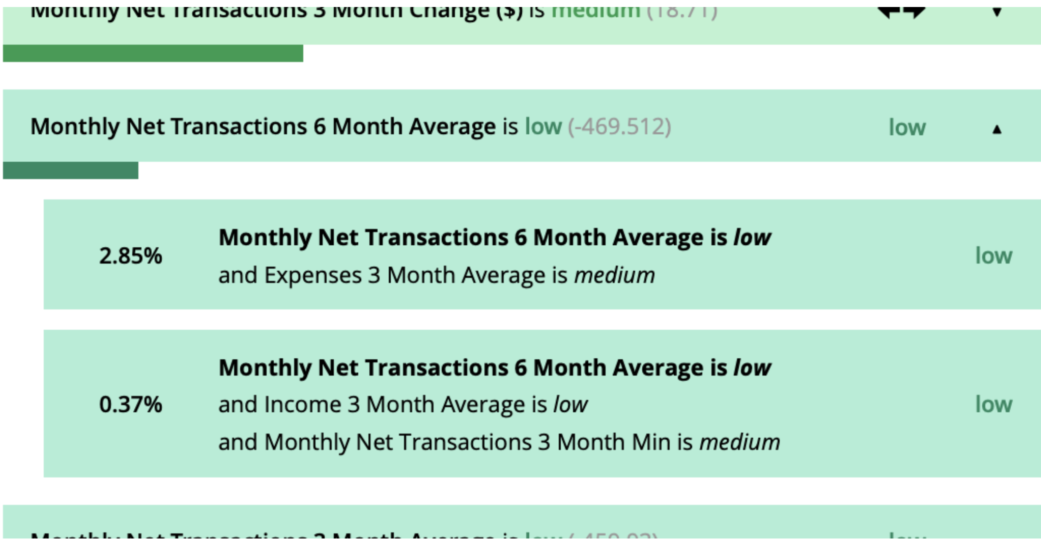


Figure 19 Rules Drop-down from Driver

For any instance the user can click the ‘Open Inference’ button and this takes them to the page shown in Figure 200. On this page, the user can change the value of any selected feature and click the predict button to see the updated score. For example reducing expenses over a 3 month window improves cashflow prospects and the extent of the benefit to the customer’s cashflow can be quantified, if they attempt to improve this going forward.

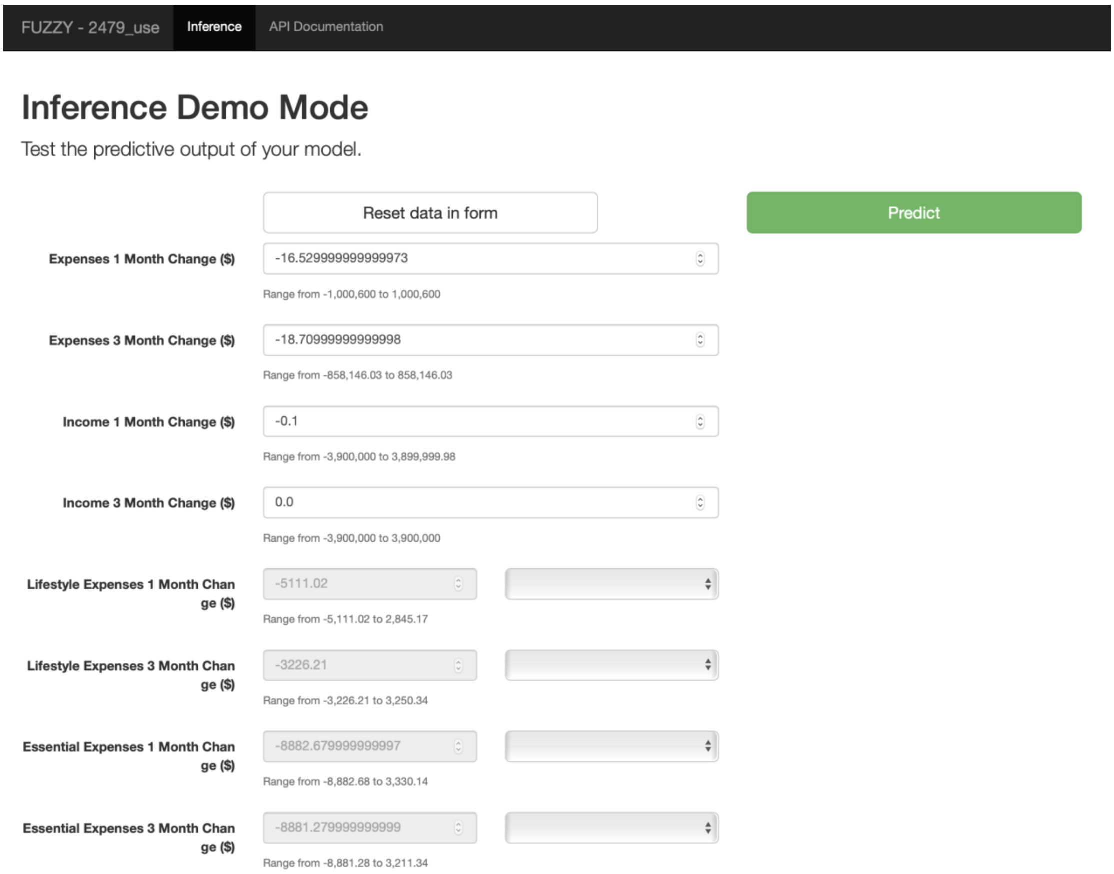


Figure 20 Inference Viewer

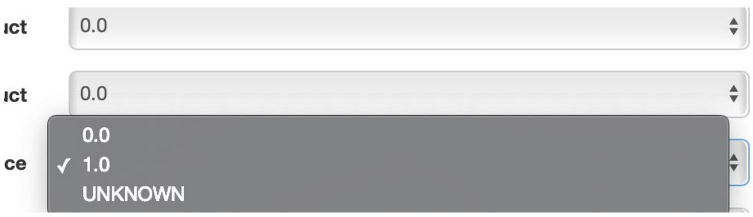


Figure 21 Changing Feature Value

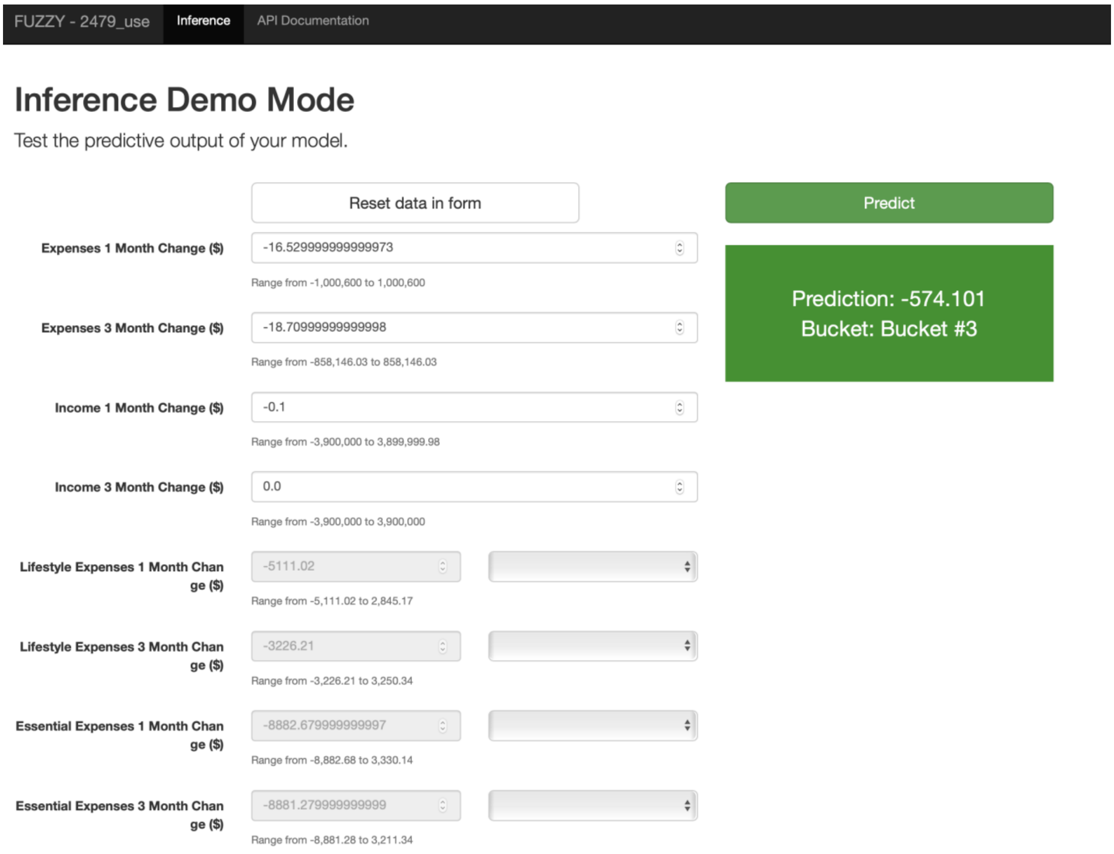


Figure 22 Updated Model Score

5.7 Use Case 7

Description

The user needs to decide on whether the model needs retraining or not, this depends on:

1. Whether the distribution of scores on live instances follows a similar pattern to the training data set.
2. Whether the live instances have similar feature value distributions to the ones used in the training data set

Business Application

Deciding on whether to continue with the current model or train a new one on more recent data is usually a business-critical decision that all clients running predictive models need to make at regular intervals. The need to retrain models can arise from a shift in economic environment and/or shift in the business policies leading to a different customer segments being passed through the model.

The platform allows a user to conduct 'Population Stability Analysis' and 'Characteristic Stability Analysis' to address points 1 and 2 respectively.

The usage is shown in the screenshots below:

To start – load the file with the data on instances that need to be tested for stability analysis. The file should contain all feature columns that were present in the original training data file in the same order. This is done from Analysis → Population → Bulk Inference as shown in Figure 23. View the bulk inference once it has run and press Perform Analysis as per Figure 24.

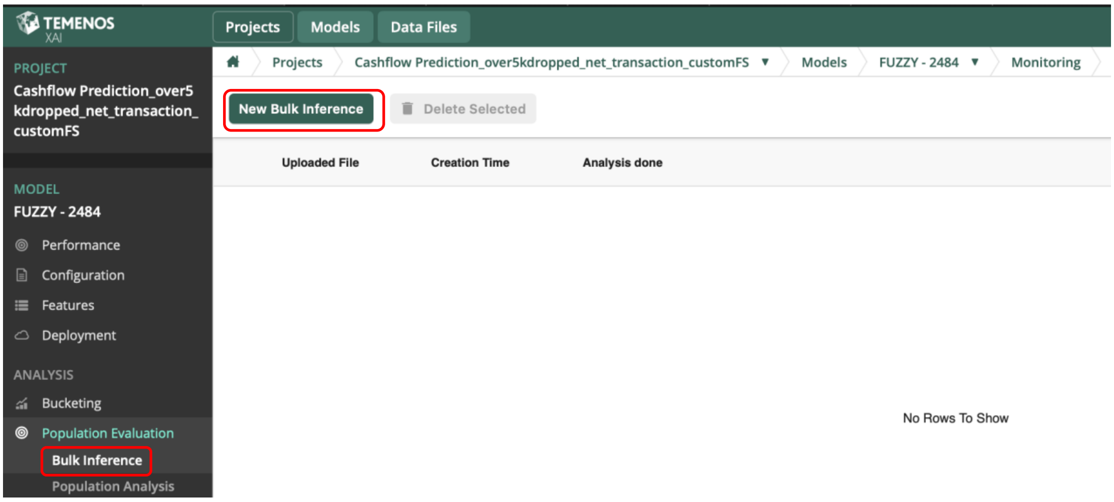


Figure 23 Create A Bulk Inference

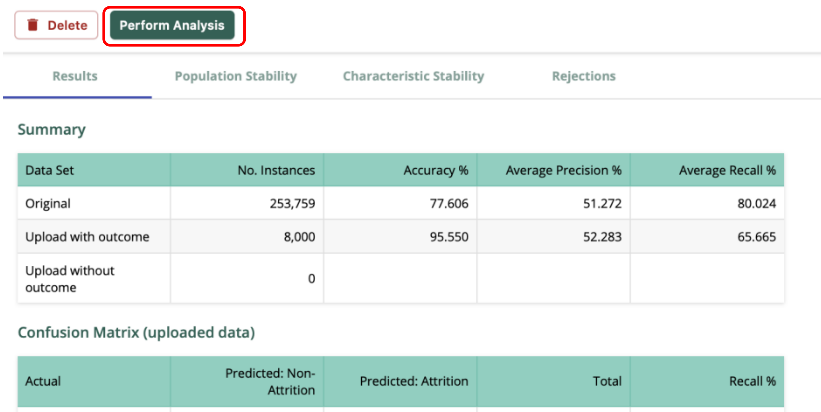
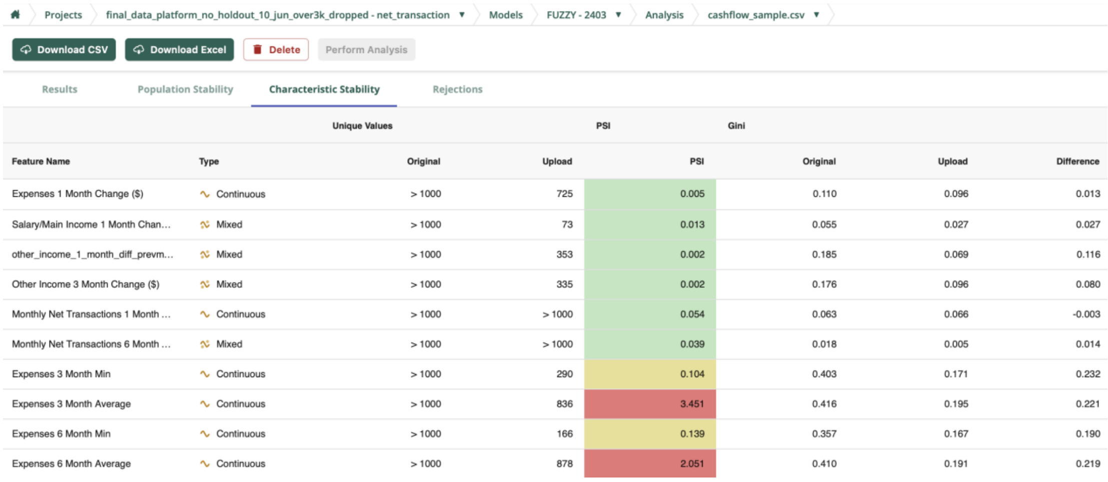


Figure 24 Perform Analysis

Population stability is a useful measure to see how the new instances are scoring compared to the training population. This can be seen visually as proportion of instances falling under each bucket of the default bucketing. The uploaded data refers to the new instances and the original data refers to the data used to build the model. An example of Population Stability is shown in Figure 25.

Temenos XAI platform can also do analysis at the level of each feature. This can be seen under the tab ‘Characteristic Stability’. The information under this tab compares the distribution of each feature in the new instances versus the original instances . This analysis in useful in identifying values which might be entered incorrectly or where the values have shifted due to a change in the economic conditions or the type of customers. Features, where the distribution has shifted significantly are highlighted in red, as shown in Figure 26

Figure 25 Population Stability



		Unique Values		PSI	Gini		
Feature Name	Type	Original	Upload	PSI	Original	Upload	Difference
Expenses 1 Month Change ($)	Continuous	> 1000	725	0.005	0.110	0.096	0.013
Salary/Main Income 1 Month Chan...	Mixed	> 1000	73	0.013	0.055	0.027	0.027
other_income_1_month_diff_prevm...	Mixed	> 1000	353	0.002	0.185	0.069	0.116
Other Income 3 Month Change ($)	Mixed	> 1000	335	0.002	0.176	0.096	0.080
Monthly Net Transactions 1 Month ...	Continuous	> 1000	> 1000	0.054	0.063	0.066	-0.003
Monthly Net Transactions 6 Month ...	Mixed	> 1000	> 1000	0.039	0.018	0.005	0.014
Expenses 3 Month Min	Continuous	> 1000	290	0.104	0.403	0.171	0.232
Expenses 3 Month Average	Continuous	> 1000	836	3.451	0.416	0.195	0.221
Expenses 6 Month Min	Continuous	> 1000	166	0.139	0.357	0.167	0.190
Expenses 6 Month Average	Continuous	> 1000	878	2.051	0.410	0.191	0.219

Figure 26 Characteristic Stability

5.8 Use Case 8

Description

User would like to add expert knowledge on sparsely available attributes/features to the model.

Business Application

In times of sudden changes to the economic landscape – new variables might become relevant for which there is no historical data available. Since the fundsflow model is a rule based approach with rules in human understandable form – it is possible to add rules to the original rule base containing patterns from the newly identified feature. The rules to be added are not a complex as the ones generated from the data as there is typically not much information available to understand complex interactions between features. However – it is a useful and unique starting point to make use of human expertise along with data-based insights. The process involves some complexity and needs to be carried out by Temenos AI data science team to make sure that the resulting model is as robust as the original model.

5.9 Use Case 9

Description

The user would like to deploy models with ease and speed – and monitor the inferences being passed through the deployed model/s.

Business Application

Fast and easy deployment is essential for banks looking to either retraining the models with new data or upgrading current models with expert knowledge.

Temenos AI platform offers a 1-click cloud deployment for models as shown in Figure 27.

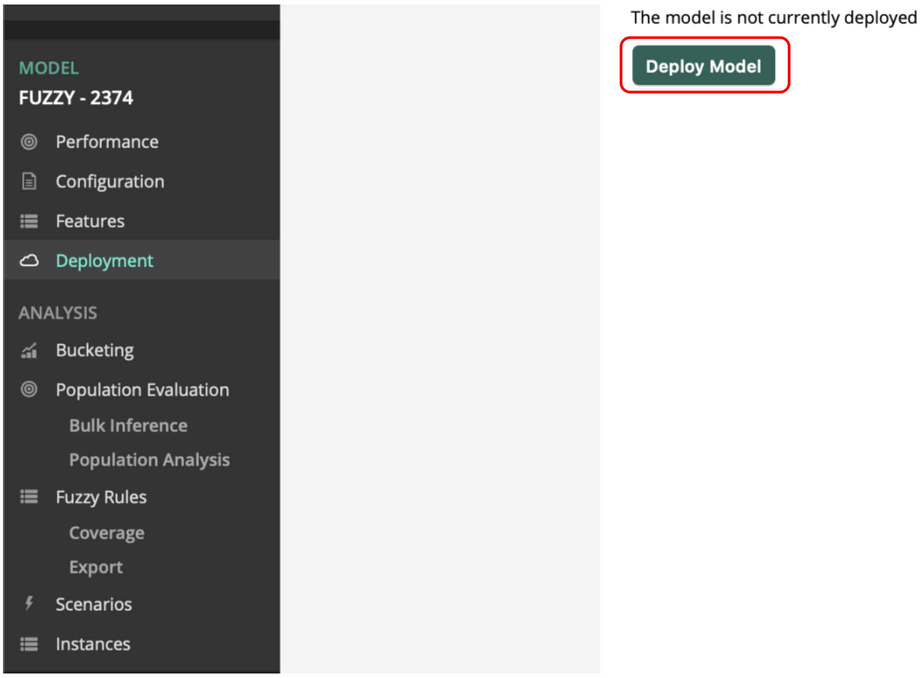


Figure 27 1-Click Model Deployment

5.10 Use Case 10

Description

User can interact with the model using a GUI based inference page, without the need to send an API call to the model. The inference results will have an option to be stored in the database.

Business Application

Some users may not have the need or the technical know-how for integrating API calls within their infrastructure. For such clients - the Temenos AI platform provides an easy way to interact with the model.

The UI based inference page can be opened from 'Deployment' → 'Inference'. The resulting page opens in a new browser tab and is shown in Figure 28. The features that need to be input on this page need to be pre-calculated by the client (using the same logic as in the preprocessing, i.e. ETL, layer) and usually, the Temenos AI team can provide a spreadsheet/script with all necessary calculations for the same. In Figure 28, if the Datasource (highlighted) is set to 'Live', then all results will be stored in the database and can be accessed from the 'History' tab on the 'Deployment' page.

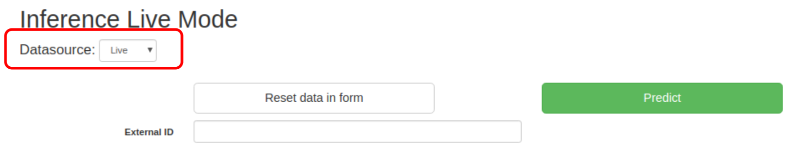


Figure 28 UI based Inference Page

5.11 Use Case 11

Description

Prior to any models being built on a project, the user can define custom fuzzy sets for continuous features. Note at present the user must be in an administrator role to access this functionality.

Business Application

By default, the platform uses a distribution based method to define the fuzzy sets such that, essentially, 'low', 'medium' and 'high' linguistic labels are represented equally often in the uploaded data. The following threshold values are necessary and sufficient to define the trapezoidal interval type-2 fuzzy sets used by the platform, with the usual three linguistic labels:

- 'low' core end/' medium' upper start
- 'low' lower end/ 'medium' lower start
- 'low' upper end/ 'medium' core start
- 'medium' core end/ 'high' upper start
- 'medium' lower end/ 'high' lower start
- 'medium' upper end/ 'high' core start

By default these six thresholds are defined such that they divide the data into heptiles with close to an equal number of instances in each subset.

However the user may wish to define their own fuzzy sets, so that the definitions of low, medium and high are consistent with the conventional understanding of these labels in their own business domain. To do so they are free to redefine these six key threshold values.

The user can therefore reach 'Features' → 'All Features' and click a continuous feature, and then click 'Change Range for MFs' to change the membership functions for the fuzzy sets, as shown in Figure 29 and Figure 30.

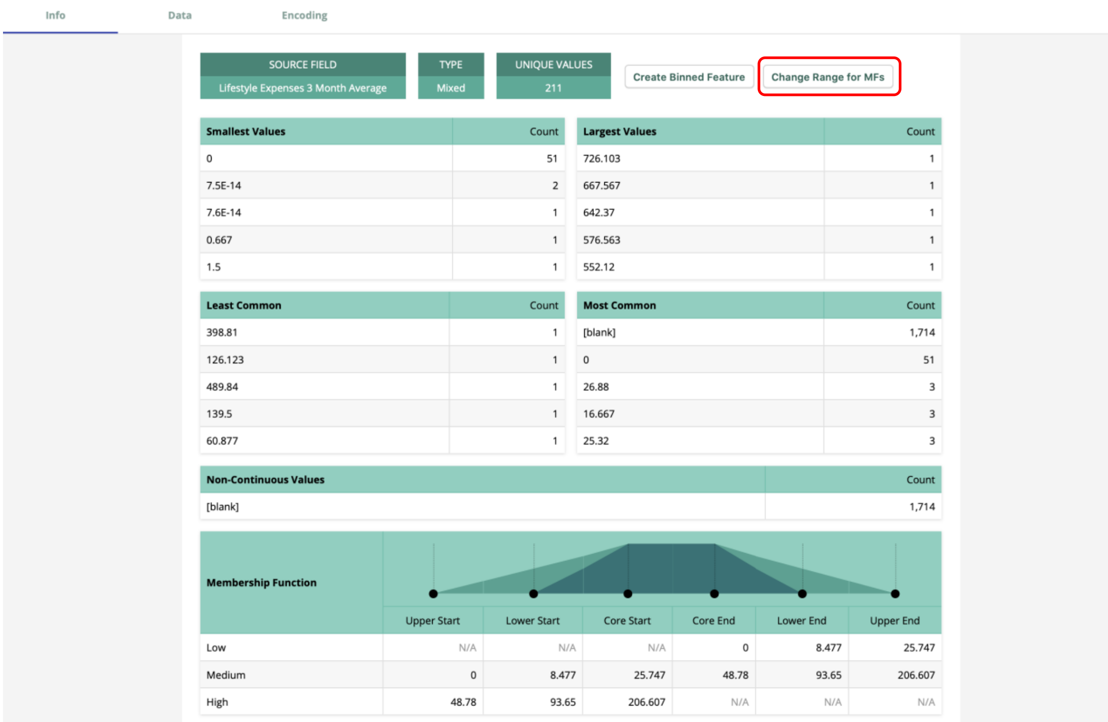


Figure 29

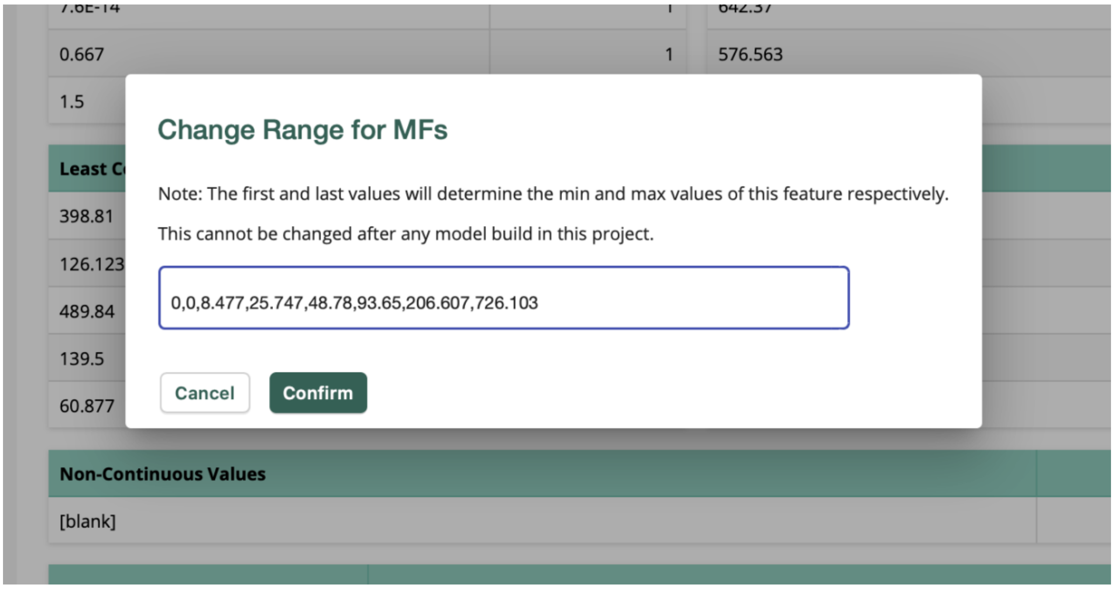


Figure 30

Note there are eight values shown in Figure 30 as the lowest and highest represent the minimum and maximum values for that feature observed in the uploaded data.

6. Configuration / Customization

For each client – the ETL layer needs to be coded if the data source being used is not already supported.

7. Assumptions

This document is based on assumption that clients will use the cloud version of the fundsflow model and Temenos AI platform. For on premise installations, there would be a separate step of configuring /installing the platform and model in client infrastructure.

8. Exclusions

Any requirements arising from on-premise installations are excluded from this document.