



# Retail Customer Next Best Product – XAI Model



## Document History

Version	Date	Author	Description
1.1	27/10/2020	Rayce Rossum	Updated screenshots to multi-class
1.0	24/08/2020	Rayce Rossum	

Comments:

---

---

---

---



## Table of Contents

1.	Introduction .....	4
2.	Business Requirements .....	5
3.	Current Functionality .....	6
4.	Proposed Solution .....	6
4.1	Overview .....	6
4.2	Run Time .....	6
5.	Use Cases .....	16
5.1	Use Case 1 .....	16
5.2	Use Case 2 .....	16
5.3	Use Case 3 .....	18
5.4	Use Case 4 .....	19
5.5	Use Case 5 .....	Error! Bookmark not defined.
5.6	Use Case 6 .....	22
5.7	Use Case 7 .....	Error! Bookmark not defined.
5.8	Use Case 8 .....	23
5.9	Use Case 9 .....	23
5.10	Use Case 10 .....	24
5.11	Use Case 11 .....	24
6.	Configuration / Customization .....	26
7.	Assumptions .....	26
8.	Exclusions .....	26



## 1. Introduction

A Temenos AI next best product model allows financial institutions to predict the most likely product a customer will acquire next, based on the past behaviour of the individual and the broader portfolio of customers. The model works on a customer level.

The information used by the next best product model is the historical product acquisitions for a customer and general customer data such as age and account balances.

The next best product model will be used either by itself to intelligently recommend products to a customer or as a part of a larger pipeline alongside the funds flow model allowing for additional product offerings.

An example of such a pipeline is Urgent Relief Loans – which identifies customers experiencing short term financial distress owing to macroeconomic events such as the Covid-19 outbreak.

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 model gives explainable outputs at the level of full population (model rule base), sub-population groups exhibiting similar behaviour (Rules applicable to individual prediction results 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.

- 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, next best product 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 next best product.
- 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 sorted according to their next best product to prioritise which ones the bank should reach out to, in order to provide more successful marketing campaigns for specific products.
- Periodic reviews using actual outcomes for the customers processed through the XAI models can give insights into the effectiveness of any existing marketing strategy. This can then serve as an input into generating better models going forward.

## 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 next best product model on the XAI platform as a standalone solution.
BR-03	For every valid inference, the model predicts a categorical value for the next best product.
BR-04	A linguistic label (target class) is associated with each prediction; the possible labels are defined by the client with default values of: Product1, Product2, Product3, Product4, Product5, Product6, Product7, Product8, Product9, Product10, Service1, Service2, Service3, Service4, and Service5.
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 recommendations.
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 results arising from a change to any/all the input value/s.
BR-09	The model supports storing of result and inputs for all historical inferences.



BR-10	The model supports investigating distribution of results 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 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

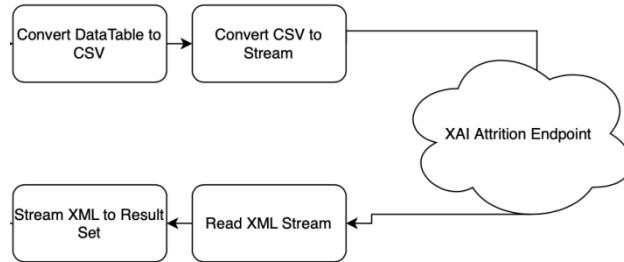
There is no existing functionality within the Temenos product suite offering similar functionality/scope. At the customer end, existing functionality is specific to each bank, if any such framework exists at all.

### 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 pre-processing layer to transform client inputs into model features; either way the bank will simply hit the XAI service 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 next best product 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 Next Best Product 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 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.

*Table 1*

The full list of potential modelling features for Retail is:

- Age
- AgeGroup
- AnnualIncomeGroup
- CreditScore
- CreditScoreGroup
- CustomerIndustry
- CustomerType



CustProfitGroup

CustProfitStatus

CustRegion

DepsBalance

DepsBalGroup

Gender

LoanBalance

LoanBalGroup

NonResident

NumAccounts

NumProdsAndServices

NumProduct1

NumProduct2

NumProduct3

NumProduct4

NumProduct5

NumProduct6

NumProduct7

NumProduct8

NumProduct9

NumProducts

NumProductsGroup

NumService1

NumService2

NumService3

NumService4



NumServices  
Product1Balance  
Product2Balance  
Product3Balance  
Product4Balance  
Product5Balance  
Product6Balance  
Product7Balance  
Product8Balance  
Product9Balance  
Tenure  
TenureGroup  
TotalBalance  
TotalBalGroup  
ChangeInBalanceNextMonth (Funds Flow Augmented)  
TotalBalanceNextMonth (Funds Flow Augmented)  
HasProduct1  
HasProduct2  
HasProduct3  
HasProduct4  
HasProduct5  
HasProduct6  
HasProduct7  
HasProduct8  
HasProduct9  
HasService1



HasService2

HasService3

HasService4

The full list of potential modelling features for SME is:

AnnualIncomeGroup

CreditScore

CreditScoreGroup

CustomerIndustry

CustomerType

CustProfitGroup

CustProfitStatus

CustRegion

DepsBalance

DepsBalGroup

LoanBalance

LoanBalGroup

NonResident

NumAccounts

NumProdsAndServices

NumProduct1

NumProduct2

NumProduct3

NumProduct4

NumProduct5

NumProduct6

NumProduct7



NumProduct8  
NumProduct9  
NumProducts  
NumProductsGroup  
NumService1  
NumService2  
NumService3  
NumService4  
NumServices  
Product1Balance  
Product2Balance  
Product3Balance  
Product4Balance  
Product5Balance  
Product6Balance  
Product7Balance  
Product8Balance  
Product9Balance  
Tenure  
TenureGroup  
TotalBalance  
TotalBalGroup  
ChangeInBalanceNextMonth (Funds Flow Augmented)  
TotalBalanceNextMonth (Funds Flow Augmented)  
HasProduct1  
HasProduct2



HasProduct3

HasProduct4

HasProduct5

HasProduct6

HasProduct7

HasProduct8

HasProduct9

HasService1

HasService2

HasService3

HasService4

**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": {  
    "Age": 0,  
    "AnnualIncomeGroup": "<50K",  
    "CreditScore": 0,  
    "CustomerIndustry": "Regular",  
    "CustomerType": "P",  
    "CustProfitStatus": "Profitable",  
    "DepsBalance": 0,  
    "Gender": "Male",  
    "LoanBalance": -1411.55,  
    "NumAccounts": 0,  
    "NumProdsAndServices": 0,  
    "NumProducts": 0,  
    "NumServices": "1",  
    "CurrentBalance": 0,  
    "SavingsBalance": 0,  
    "RetirementBalance": 0,  
    "SharesBalance": 0,  
    "WealthBalance": 0,  
    "TermLoanBalance": 0,  
    "MortgageBalance": -2.14,  
    "LeasingBalance": -1411.55,  
  }  
}
```



```

    "Tenure": "1Y",
    "TotalBalance": -1378.81,
    "ChangeInBalanceNextMonth": -2626.2578184161,
    "TotalBalanceNextMonth": -2984.92,
    "HasService1_Name": "0",
    "HasService3_Name": "0",
    "HasService4_Name": "0"
},
"modelKey": "f597bfa2-c791-4ba4-b222-9873a435d8a2"
}

```

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": [
    {
      "Age": 0,
      "AnnualIncomeGroup": "<50K",
      "CreditScore": 0,
      "CustomerIndustry": "Regular",
      "CustomerType": "P",
      "CustProfitStatus": "Profitable",
      "DepsBalance": 0,
      "Gender": "Male",
      "LoanBalance": -1411.55,
      "NumAccounts": 0,
      "NumProdsAndServices": 0,
      "NumProducts": 0,
      "NumServices": "1",
      "CurrentBalance": 0,
      "SavingsBalance": 0,
      "RetirementBalance": 0,
      "SharesBalance": 0,
      "WealthBalance": 0,
      "TermLoanBalance": 0,
      "MortgageBalance": -2.14,
      "LeasingBalance": -1411.55,
      "Tenure": "1Y",
      "TotalBalance": -1378.81,
      "ChangeInBalanceNextMonth": -2626.2578184161,
      "TotalBalanceNextMonth": -2984.92,
      "HasService1_Name": "0",
      "HasService3_Name": "0",
      "HasService4_Name": "0"
    },
    {
      "Age": 0,
      "AnnualIncomeGroup": "<50K",
      "CreditScore": 0,
      "CustomerIndustry": "Regular",
      "CustomerType": "P",
      "CustProfitStatus": "Profitable",
      "DepsBalance": 0,
      "Gender": "Male",
      "LoanBalance": -1411.55,
      "NumAccounts": 0,
      "NumProdsAndServices": 0,
      "NumProducts": 0,
      "NumServices": "1",
      "CurrentBalance": 0,
      "SavingsBalance": 0,
      "RetirementBalance": 0,
      "SharesBalance": 0,
      "WealthBalance": 0,
      "TermLoanBalance": 0,
      "MortgageBalance": -2.14,
      "LeasingBalance": -1411.55,
      "Tenure": "1Y",
      "TotalBalance": -1378.81,
      "ChangeInBalanceNextMonth": -2626.2578184161,
      "TotalBalanceNextMonth": -2984.92,
      "HasService1_Name": "0",
      "HasService3_Name": "0",
      "HasService4_Name": "0"
    }
  ]
}

```



```
"DepsBalance": 0,  
"Gender": "Male",  
"LoanBalance": -1411.55,  
"NumAccounts": 0,  
"NumProdsAndServices": 0,  
"NumProducts": 0,  
"NumServices": 1,  
"CurrentBalance": 0,  
"SavingsBalance": 0,  
"RetirementBalance": 0,  
"SharesBalance": 0,  
"WealthBalance": 0,  
"TermLoanBalance": 0,  
"MortgageBalance": -2.14,  
"LeasingBalance": -1411.55,  
"Tenure": "1Y",  
"TotalBalance": -1378.81,  
"ChangeInBalanceNextMonth": -2626.2578184161,  
"TotalBalanceNextMonth": -2984.92,  
"HasService1_Name": "0",  
"HasService3_Name": "0",  
"HasService4_Name": "0"  
}  
],  
"modelKey": "f597bfa2-c791-4ba4-b222-9873a435d8a2"  
}
```

#### 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 **Error! Reference source not found.** . All other response information is the same as contained within the standard response.



## 5. Use Cases

### 5.1 Use Case 1

#### Description

User Makes an API call to the XAI model service with a valid payload. The model returns a response with a categorical result, 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 XAI model service 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 XAI model’s explainable decisions assist the end-user by presenting its reason for the output result on a driver (feature) and rule basis.

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

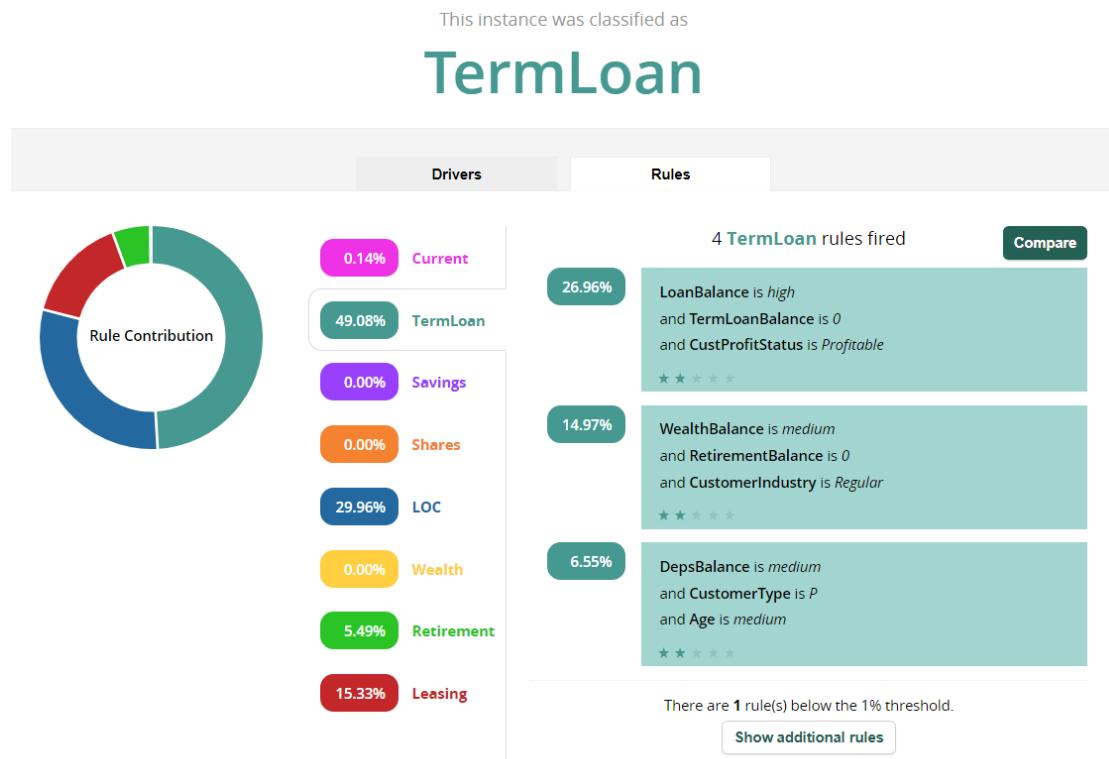
- the prevalence of the rule within all instances used to train the model and,
- the accuracy of the rule



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 3 and 4.





This instance was classified as

## TermLoan

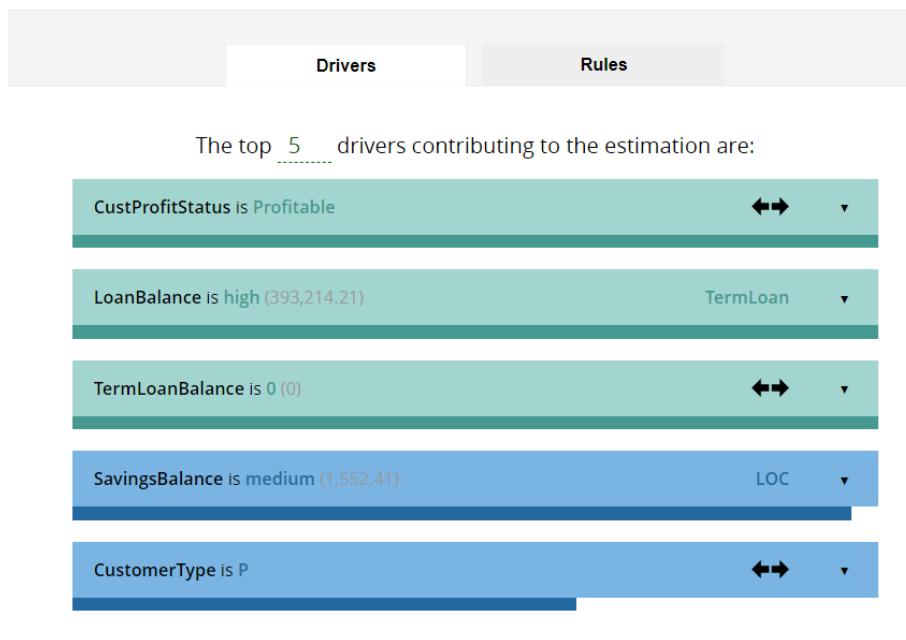


Figure 4 Driver View

### 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 next best product model's fully transparent rule base captures complex interactions between input features while being explainable.



Rule Id	Antecedents	Result	Statistics
218	LoanBalance is high CurrentBalance is 0 MortgageBalance is 0	Current	Dominance ▲ ★★★★ 1 Frequency 7,522
447	LoanBalance is high CurrentBalance is 0 WealthBalance is medium	Current	Dominance ▲ ★★★★ 2 Frequency 6,476
184	AnnualIncomeGroup is <50K CurrentBalance is 0 HasService1_Name is 1	Current	Dominance ▲ ★★★★ 3 Frequency 2,629

Figure 5 - 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 next best product model by adding/modifying/deleting rules from the existing rule base.

The screenshot shows a user interface for managing scenarios. On the left, there is a vertical dark sidebar. In the center, there is a table with columns: Name, Rules, Avg. Recall, Performance Diff %, Modified, Added, and Removed. At the top of the central area, there is a green button labeled "New Scenario" which is highlighted with a red rectangular box. The rest of the table and interface elements are standard grey and white colors.

Figure 6 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 7 and Figure 0:

Rule Id	Antecedents	Result	Statistics

Figure 7 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.

Rule Id	Antecedents	Result	Rule Properties
Added	<input type="radio"/> Select Feature - LoanBalance is high <input type="radio"/> CurrentBalance is 0 <input type="radio"/> MortgageBalance is 0 Add Antecedent	Dominance is 5	

Figure 8 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 1)

Rule Id	Antecedents	Result	Rule Properties
218	<input type="radio"/> LoanBalance is high <input type="radio"/> CurrentBalance is 0 <input type="radio"/> MortgageBalance is 0 Add Antecedent	Dominance is 5	

Figure 9 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 (

Rule Id	Antecedents	Result	Statistics

- Figure 2) with the performance change shown in a pop-up on the screen (Figure 3).

Rule Id	Antecedents	Result	Statistics



Figure 10 Evaluate Rule Button

## Evaluate Scenario

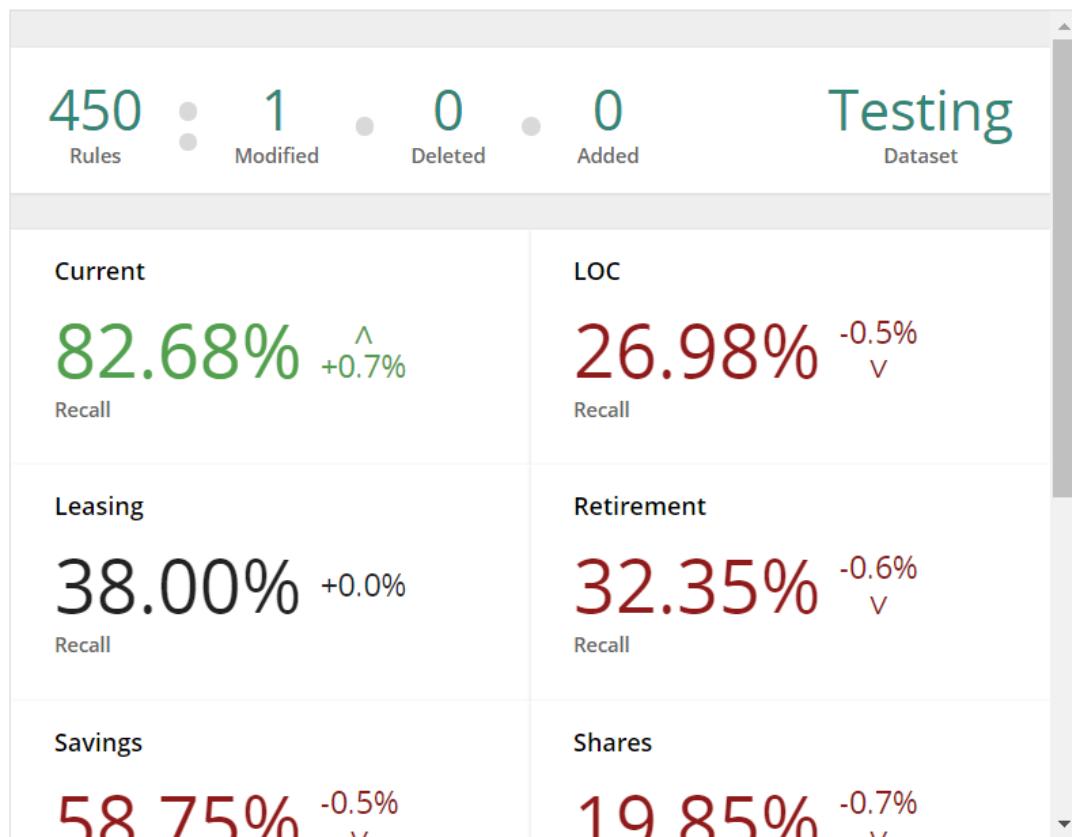


Figure 11 Evaluating the Scenario Performance

- 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 ) and it will be built as a model in the background and will appear in the models list once the build has finished.



Figure 12 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

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

### Business Application

The bank might wish to provide understandable actions which will recommend products based on potential future changes. Perhaps the bank knows a client is going to be depositing a large amount of funds or taking out a large loan in the near future but has yet to actually do so. This provides the opportunity to cross-sell a relevant product to the customer before these actions are taken.

The simulated features or attributes can be taken from the explanation provided for the score. The instance can be opened in the ‘Inference Viewer’ and the user can pass different values for the attributes to see how this will affect the result.

For any instance the user can click the ‘Open Inference’ button and this takes them to the page shown in Figure . On this page, the user can change the value of any selected feature and click the predict button to see the updated result.

## Inference Live Mode

Datasource:

External ID	<input type="text" value=""/>
index	<input type="text" value="8"/>
Range from 8 to 573,117	
Age	<input type="text" value="0"/> <span style="margin-left: 10px;">-- No Category <input style="border: none; border-left: 1px solid #ccc; padding: 0 2px; height: 1em;" type="button" value="▼"/></span>
Range from 0 to 112	
AgeGroup	45-54 <input style="border: none; border-left: 1px solid #ccc; padding: 0 2px; height: 1em;" type="button" value="▼"/>
AnnualIncomeGroup	<50K <input style="border: none; border-left: 1px solid #ccc; padding: 0 2px; height: 1em;" type="button" value="▼"/>
CreditScore	<input type="text" value="0"/> <span style="margin-left: 10px;">-- No Category <input style="border: none; border-left: 1px solid #ccc; padding: 0 2px; height: 1em;" type="button" value="▼"/></span>
Range from 0 to 899	

Predict

Figure 15 Inference Viewer



Range from 0 to 104

AnnualIncomeGroup	<50K
CreditScore	<50K 50K-100K 100K-250K >250K UNKNOWN

*Figure 16 Changing Feature Value*

## 5.6 Use Case 6

### 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 next best product 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.

## 5.7 Use Case 7

### 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 .



The model is not currently deployed

**Deploy Model**

**MODEL**  
FUZZY - 2374

- Performance
- Configuration
- Features
- Deployment**

**ANALYSIS**

- Bucketing
- Population Evaluation**
  - Bulk Inference
  - Population Analysis
- Fuzzy Rules
  - Coverage
  - Export
  - Scenarios
- Instances

Figure 17 1-Click Model Deployment

## 5.8 Use Case 8

### 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 . 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 , 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.

Inference Live Mode

Datasource: **Live**

Reset data in form

Predict

External ID

Figure 18 UI based Inference Page

## 5.9 Use Case 10

### Description

Vx.x/MMM-YY

**Quality Assurance**

**Internal Use**

[To view template history, please click here\\*](#)



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.

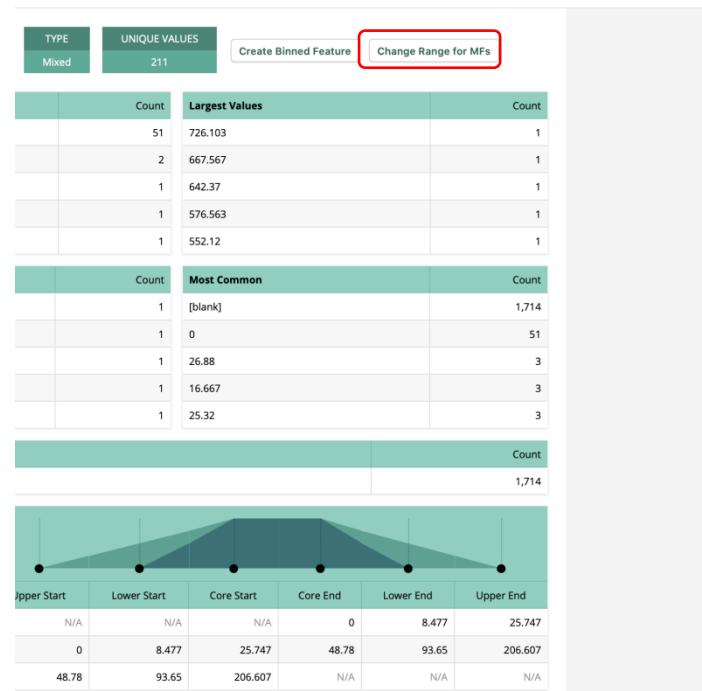




Figure 19

7.0E+14		1	042.37
0.667		1	576.563
1.5			
<b>Least C</b>			
398.81			
126.123			
489.84			
139.5			
60.877			
<b>Change Range for MFs</b>			
Note: The first and last values will determine the min and max values of this feature respectively.			
This cannot be changed after any model build in this project.			
0,0,8.477,25.747,48.78,93.65,206.607,726.103			
<b>Cancel</b>		<b>Confirm</b>	
<b>Non-Continuous Values</b>			
[blank]			

Figure 20

Note there are eight values shown in Figure 20 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 next best product model and Temenos XAI 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.