Transaction Classification XAI Model

Solution Document

Document History

Version	Date	Author	Description
1.0			
1.1	03/11/202		

Comments:

Table of Contents

1. Introduction

This document is an introduction to the XAI Transaction classification pipeline. The pipeline classifies a transaction(s) as belonging to one of the pre-defined labels. The pipeline works with transaction narratives (description strings) - which are then used to create features from a Natural Language Processing (NLP) pipeline and then these features along with other information related to the transaction (amount, time of day etc.) are used as a part of an XAI model to arrive at a final label.

Transaction classification can be used as a part of multiple end-use cases. As an example, the pipeline can be used to create targeted and efficient marketing campaigns and/or customer segmentation.

The pipeline works with basic information provided by the bank about the transaction sender/recipient, and other metadata about the transaction itself. This data can be provided via an API, or provided manually via user interfaces – making the pipeline widely adaptable.

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

Temenos XAI offers many advantages including full explainability, ease of deployment and more:

- 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 and proven Intraday Balance Prediction 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 institutions' 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 XAI predictions.
- Temenos XAI can monitor and analyse historical predictions 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 transaction characteristics.
- Temenos XAI can provide actionable insights, where applicable, and identify new approaches that are working, and provide feedback through an inference webpage to test different approaches for a given instance.
- Periodic reviews using actual outcomes for the transactions processed through the XAI models can give insights into the effectiveness of the existing transaction

prediction strategy. This can then serve as an input into generating better models going forward.

-

1.1 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 Transaction Classification model on the XAI platform as a standalone solution.
BR-03	For every valid input, the model gives an output i.e. the label allocated to that transaction.
BR-04	The scores generated by the model have a monotonic meaning – i.e. a score of 0.8 cannot signify a lower probability of belonging to a class than a score of 0.79 – and the same is true for all numbers between 0 and 1.
BR-05	The model output includes an 'explanation' in form of drivers and the rules behind every individual transaction.
BR-06	The model has a transparent global rule base for arriving at individual transaction 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 risk bucket analysis can be carried out on a population with known outcomes (Bucketing)
BR-14	The bucketing allows for customization on number of buckets
BR-15	The bucketing allows for ‘goal-seek’ analysis on buckets in terms of number of minority class instance in each bucket
BR-16	The bucketing allows for ‘goal-seek’ analysis on buckets in terms of percentage of minority class instance in each bucket

2. Current Functionality

The current Temenos Transaction Classification approach is to allocate labels to the transactions. The labels are allocated based on the Transaction Narrative and the time at which the transaction was done. Based on the patterns of historical transactions, rules are derived based on which different labels are assigned to different types of transactions.

In the current version of the model only one label is allocated to each transaction.

3. Proposed Solution

3.1 Overview

Apart from transaction narrative the new approach can use additional fields to distinguish transaction labels. The fields can be, but are not limited to, as an example amount, time of the day, day of the week, etc. For example, a transaction in the same restaurant in the morning can be regarded as breakfast or coffee and in the afternoon as lunch. Large amount spent in a garage may indicate acquisition of a new car while considerably smaller amount indicated service or repair.

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 data source provided by the client – these are then passed through a pre-processing pipeline if necessary, 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 details in the next section.

3.2 Run Time

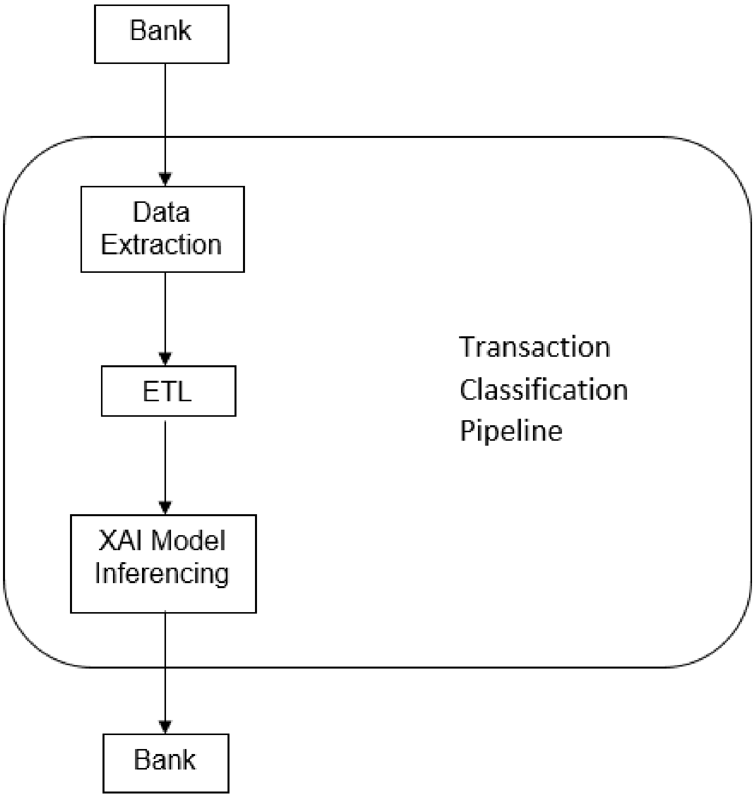

Figure 1 The Model Run Time Flow

Figure 1 shows an overview of the model run time flow. At run time the request containing the raw information from the bank, the model key and other optional information hits the model endpoint.

The raw variables are passed through the Data Extraction layer which extracts the NLP features from the transaction narrative. A host of intermediary features – each featuring TF-IDF scores for various transaction categories are created as part of the NLP pipeline and can be used in later iterations of the XAI model if they are needed.

The ETL layer creates further features useful for XAI models such as 'Max_score' - which determines the category with the highest TF-IDF score from the transaction narrative. In addition, meta-data based features such as transaction amount, day of the week and time of the day are also extracted as a part of the ETL layer. For the training data generation – output labels are mapped to the desired granularity in this layer.

Table 1 shows an example of such transformation for each feature for information received for a given set of features. There could be in practice a separate ETL layer for each financial institution, as the transformation required will be unique to information sent by each bank.

Raw feature	Explanation
TranAmount	Transaction Amount
ToD	Time of the day when the transaction was done. i.e. morning, afternoon, evening or night.
DoW	Day of the week when the transaction was done
Max_score	Label with the max score value based on transaction narrative.

API Request Structure:

The API can be interacted with using a JSON payload, which will include a Model Key. The field names within the “data” field are mapped to corresponding feature names from TransUnion (shown in the pre-processing table above). A sample payload for a single inference is shown below:

```
{
  "data": {
    "TranAmount": "115.8",
    "ToD": "morning",
    "DoW": "Friday",
    "max_score": "Travel"
  },
  "modelKey": "d5dc21b3-bbcc-4472-a00a-c3ce4729debf"
}
```

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": [{
    "TranAmount": "115.8",
    "ToD": "morning",
    "DoW": "Friday",
    "max_score": "Travel"
  },
{
    "TranAmount": "30",
    "ToD": "evening",
    "DoW": "Friday",
    "max_score": "Fast Food"
  }
,],
  "detailed": “false”,
  "modelKey": " f0281b27-f875-4d21-9c49-3645c6e824d1 "
}
```

API Response Structure:

The user has the option to choose between a standard response and a detailed response. The standard response contains all the required information to process a transaction – including a score and a URI to view the explainable output.

The standard response is shown in Figure 2 Basic API Response Structure.

The definitions of the keys are as follows:

externalId: A reference ID for the inference, if provided

bucketName: Risk Bucket the model score places the transaction into

score: Score from the model

detailedInformation: if the “detailed” flag is set to true, contains the inference drivers/rules

inferenceUri: Link to the explainable results

warnings: Any warnings for the inference, such as a continuous input being greater than or less than the range seen by the model in the training data


```
{
    "externalId": null,
    "bucketName": null,
    "inferenceUri": "api/audit/model/0e5d8c82-f97a-4a22-8626-201ce00e0fa7?time=1603964408438",
    "score": 0.05994566565075998,
    "detailedInformation": {
        "results": {}
    },
    "inferenceResultEncoded": -2.147483452E9,
    "inferenceResultDecoded": "Travel",
    "mapOutputRatios": {
        "Travel": 0.94005433434924,
        "Breakfast": 0.05994566565075995
    },
    "warnings": [],
    "status": "SUCCESS"
}
```


Figure 2 Basic API Response Structure

4. Use Cases

4.1 Use Case 1

Description

User Makes an API call to the Transaction Classification pipeline with a valid payload. The model returns a response with one of the target class labels 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 Transaction Classification model API 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 label and URI
Model input/s cannot be derived from the payload	Model returns an error response.

4.2 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 transaction and a condensed ‘Driver’ view of the most important features-value combinations applicable to the transaction.

Business Application

The Temenos Transaction Classification model’s explainable decisions aim to assist the end-user by presenting its reason for the output label 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 records used to train the model and,
- the accuracy of the rule

For any given transaction, 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 a transaction satisfies the rule condition.

Finally, from all the rules that have ‘*fired*’, all the premises 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 3 and Figure 4.

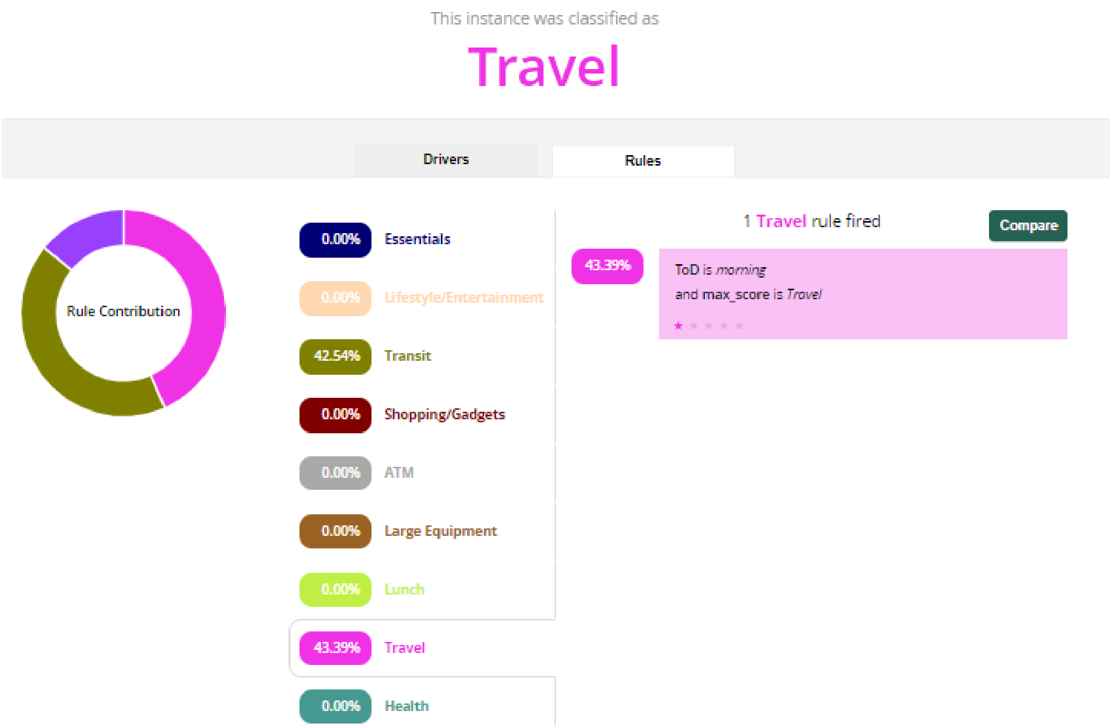

Figure 3 Rule View

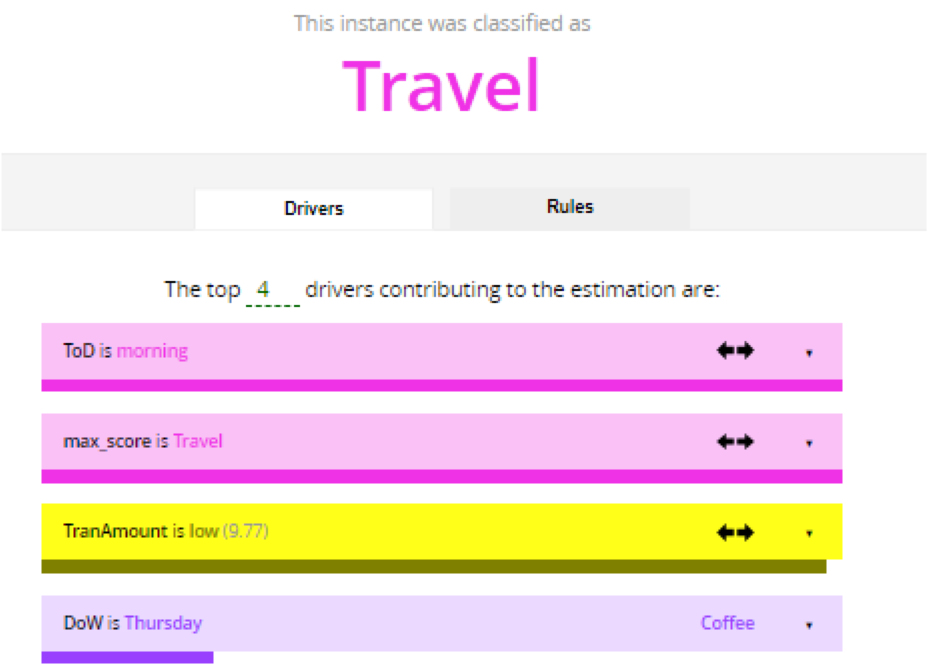

Figure 4 Driver View

4.3 Use Case 3

Description

The model has a transparent global rule base for arriving at individual transaction labels and the same can be viewed by going to the model page and then clicking 'Analysis' → 'Fuzzy Rules'. The rules are by default ordered by their importance to the model. Each rule in the rule base 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 the model logic is necessary for banks to have faith in the model actions as well as comply with regulatory requirements. The Transaction Classification model's fully transparent rule base captures complex interactions between input features while being explainable.

A screenshot for the top 5 Rules for the primary Transaction Classification model is shown in Figure 5.

Rule Id	Antecedents	Result	Statistics
64	max_score is Groceries	Essentials	Dominance ∧ ★★★★★ 1 Frequency 39,295
7	max_score is LOC_Loan Payment	Finance	Dominance ∧ ★★★★★ 2 Frequency 2
6	max_score is Credit Arrangement	Finance	Dominance ∧ ★★★★★ 3 Frequency 36
5	max_score is Money Transfer	Finance	Dominance ∧ ★★★★★ 4 Frequency 5
4	max_score is Term Deposit Closure	Finance	Dominance ∧ ★★★★★ 5 Frequency 3

Figure 5 Rule Base - Top Rules

4.4 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

Financial institutions can be in full control of the Transaction Classification model by augmenting the initial rule base with their own domain expertise. This also allows the banks to be fully in control even if the economic landscape has changed and they would like the rule base to reflect the changes before the actual transaction data starts picking up on the changes. This allows the banks to always be able to adapt rapidly to the ever evolving landscape of financial transactions.

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 model by adding/modifying/deleting rules from the existing rule base.

From the model pane – going to 'Analysis' → 'Scenarios' → 'New Scenario' (Figure 6) will create a new scenario with a user specified name – which by default is the same as the underlying rule base

.

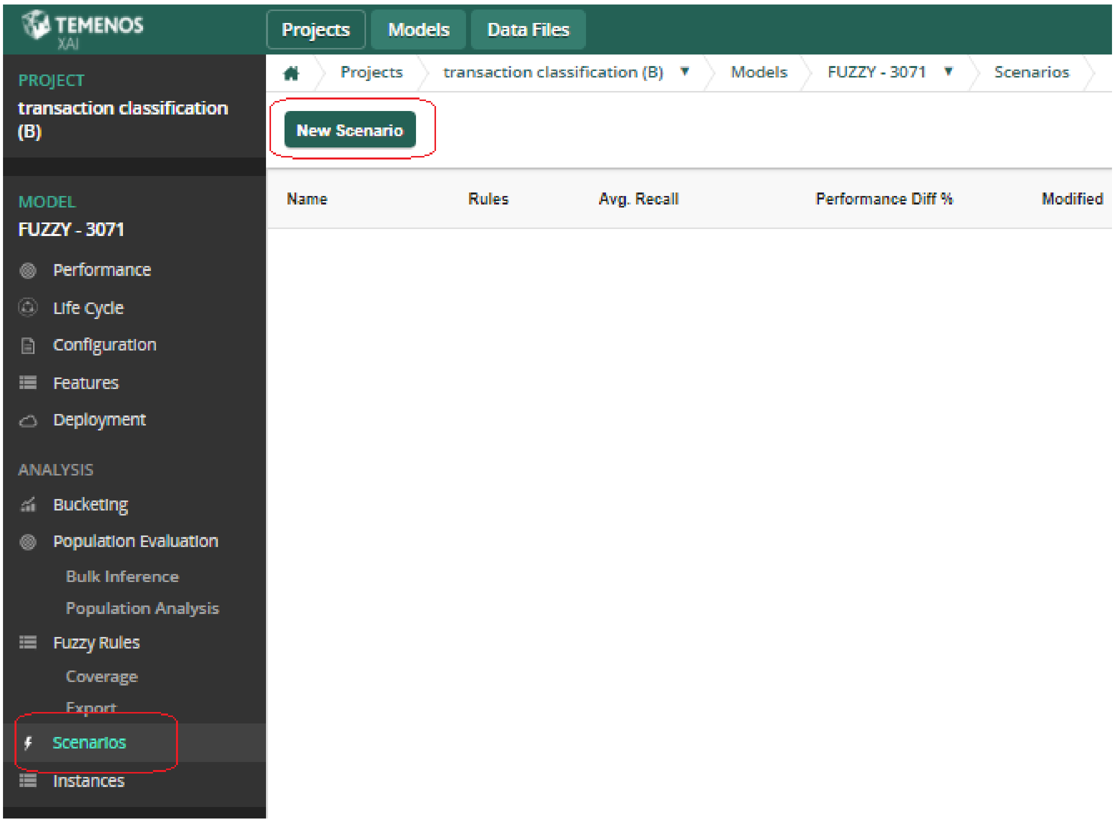

Figure 6: Create a new Scenario

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 8: 22

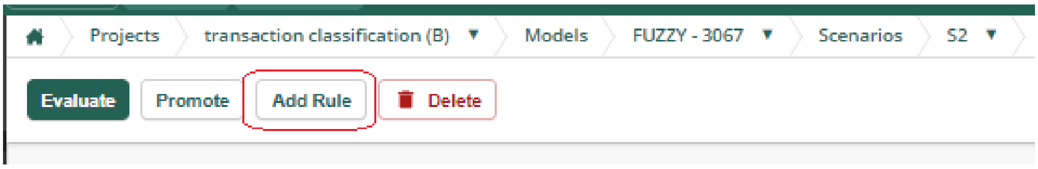

Figure 7 Add Rule Button

- 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.

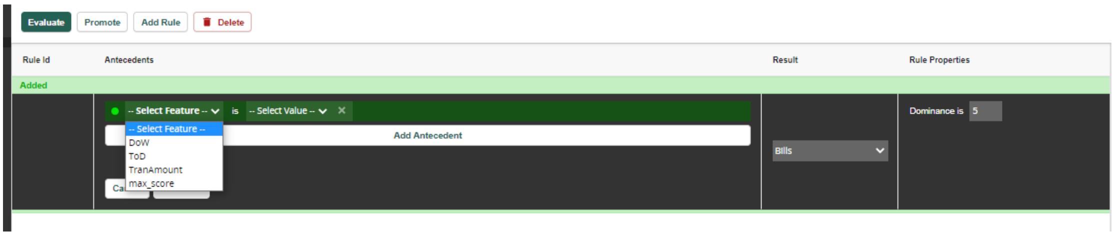

Figure 8 Adding a Rule

- 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 9)

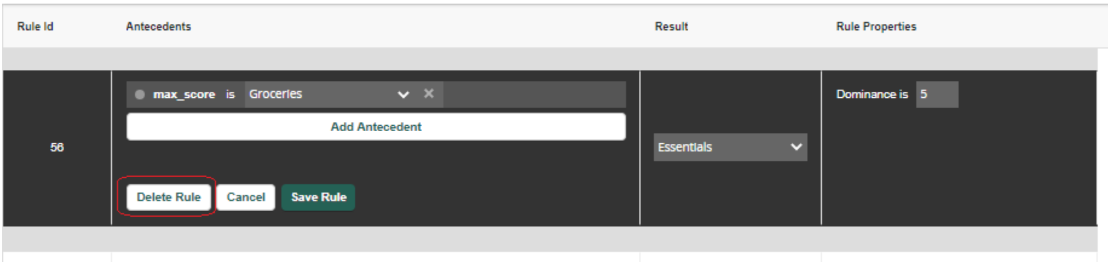

Figure 9 Deleting a Rule

- 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 (Figure 10) with the performance change shown in a pop-up on the screen (Figure 11).

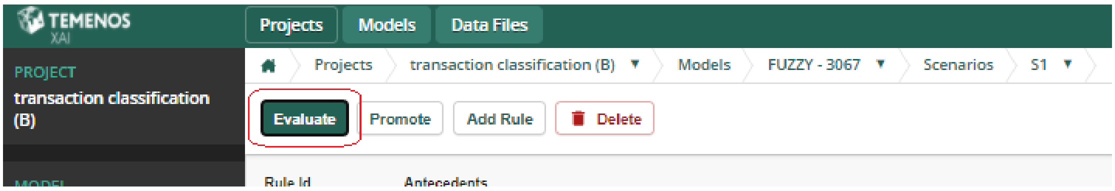

Figure 10 Evaluate Rule Button

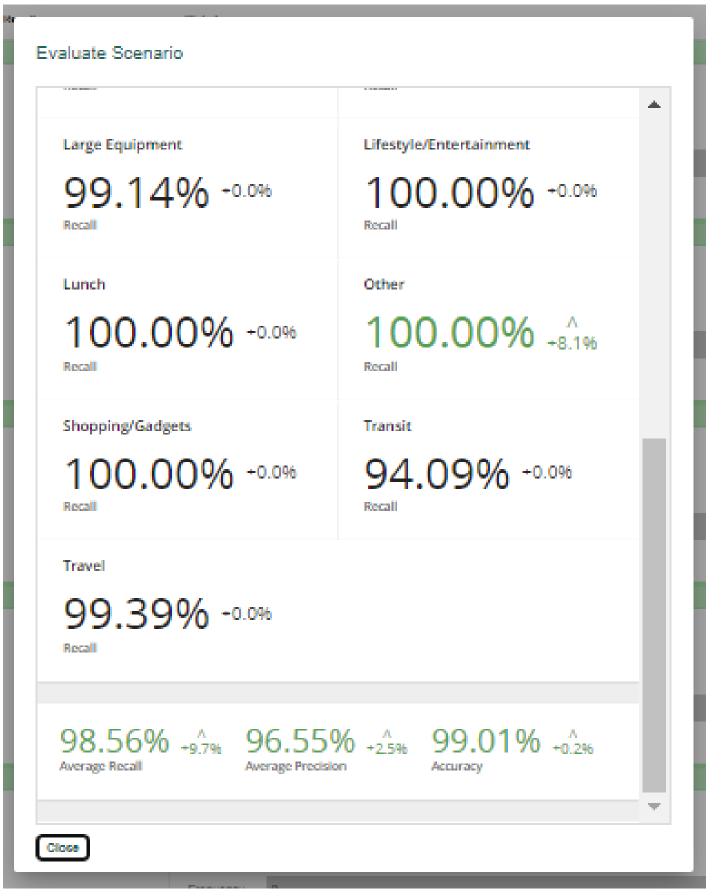

Figure 11 Evaluating the Scenario Performance

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

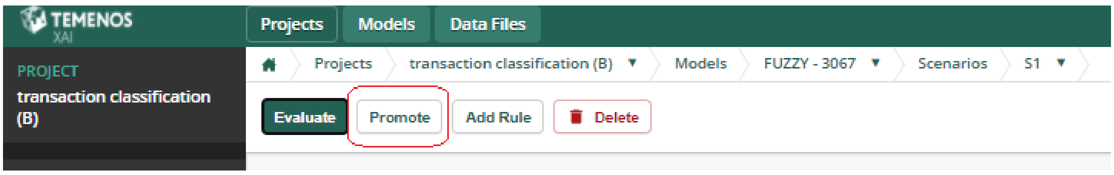

Figure 12 Promote a Scenario to a Model

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.

4.5 Use Case 5

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 transactions follows a similar pattern to the training data set.
2. Whether the live transactions 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.

Population stability is a useful measure to see how the new transactions are scoring compared to the training population. This can be seen visually as proportion of transactions falling under each bucket of the default bucketing. The uploaded data refers to the new transactions and the original data refers to the data used to build the model. An example of Population Stability is shown in **Error! Reference source not found.**.

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 transactions vs the original transactions. 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 payments and transfers made

4.6 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 Transaction Classification 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.

4.7 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 straightforward cloud deployment for models as shown in Figure 14 .

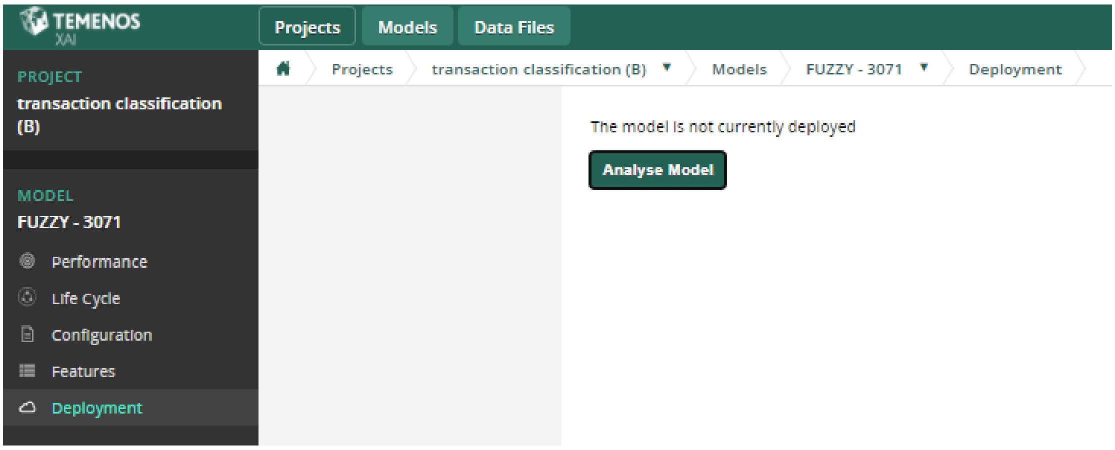

Figure 14 Simple Model Deployment

Upon making this change a reason needs to be provided for audit purposes (Figure 15).

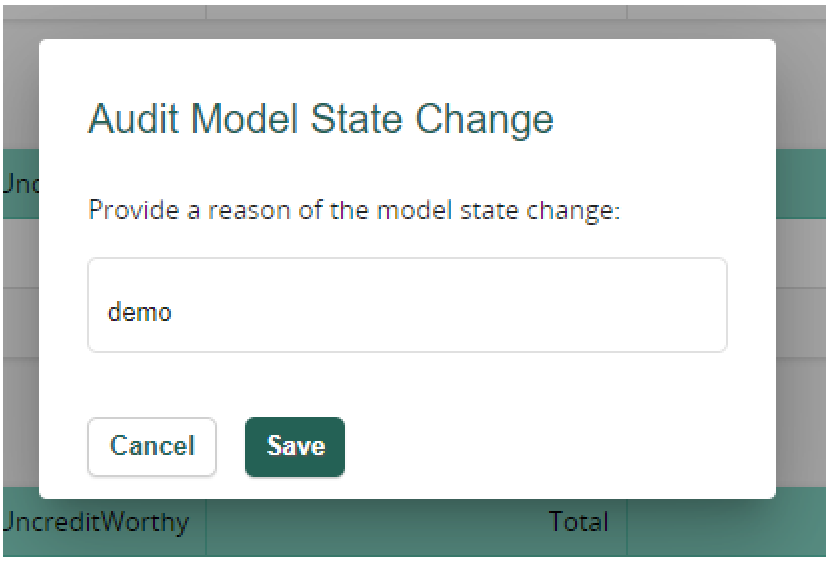

Figure 15: Audit reason

4.8 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 financial institutions 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 16. 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 ETL layer) and usually, the Temenos AI team can provide a spreadsheet/script with all necessary calculations for the same. In Figure 16, 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.

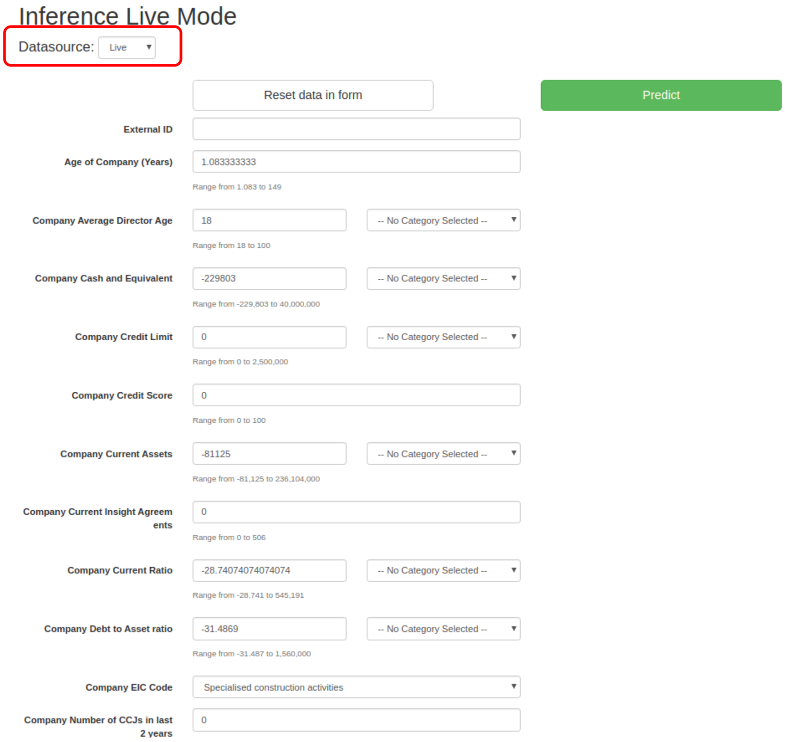

Figure 16 UI based Inference Page

5. Configuration / Customization

For each client – the ETL layer needs to be coded if the bank's information being used is not already supported.

6. Assumptions

This document is based on assumption that clients will use the cloud version of the Transaction Classification 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.

7. Exclusions

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