Transaction Prediction XAI Model

Solution Document

Document History

Version	Date	Author	Description
1.0	26th August 2020	Ashish Kumar/Dishi Tiwari	

Comments:

Table of Contents

1. Introduction

A Transaction Prediction model is a model that considers the history of previous financial transactions for a customer and predicts if the transaction will occur in the next seven days. This module needs, as a mandatory prerequisite, the Temenos Transaction Classification model to be installed as the model relies on the transaction labels defined by the Transaction Classification model.

The business value for this model lies in predicting the future purchasing behaviour of customers by considering all their accounts. Transaction Prediction can help a

- bank in identifying new business opportunities,
- bank's customers to manage their spending better or,
- bank to analyse/forecast trends in revenues.

The model 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 – leading to its wide adaptability.

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.

- 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 Transaction 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 transaction prediction.
- 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 for future reference onto the XAI platform by tagging individual predictions with a unique transaction identifier.
- Temenos XAI can monitor and analyse historical transactions 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, and identify new approaches that are working, and provide feedback through an inference webpage to test different approaches for a given instance.
- Transactions can be ranked according to a prediction score to prioritise which ones should make it through.
- 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 data can be provided 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 Prediction model on the XAI platform as a standalone solution.	
BR-03	For every valid input, the model gives an output between 0 and 1 (inclusive), where scores closer to 0 indicate that the transaction doesn't occurs in the next 7 days and scores closer to 1 indicate that the transaction occurs in the next 7 days.	
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 transaction happening in next week 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 prediction 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 prediction 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 Transaction Prediction pipelines are available in the Analytics framework and are made up of two components:

- The first component predicts the likelihood of a transaction occurring in the next 'n' days, where 'n' is a variable that is passed to the pipeline. The underlying model used in a logistic regression model and there are separate models for each label.
- The second component predicts the likely value of the next transaction for the same customer/transaction label. The underlying model is a regression model and there are separate models for each label.

Together, these two components consist of hundreds or more underlying models which makes their management/deployment complex. The proposed solution discussed in the next section consists of a single model (classification) and should be simpler to manage.

3. Proposed Solution

3.1 Overview

The model's training requires historical daily transactional data for a period of a few months, in order to provide the prediction algorithm with a sufficient number and variety of transactions. These standard guidelines regarding the size of the training set can vary depending on the size of the financial institution for which the model is configured. Once the training is completed, predictions can be applied to live data, typically considering transactions from the latest business date(s).

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 detail in the next section.

3.2 Run Time

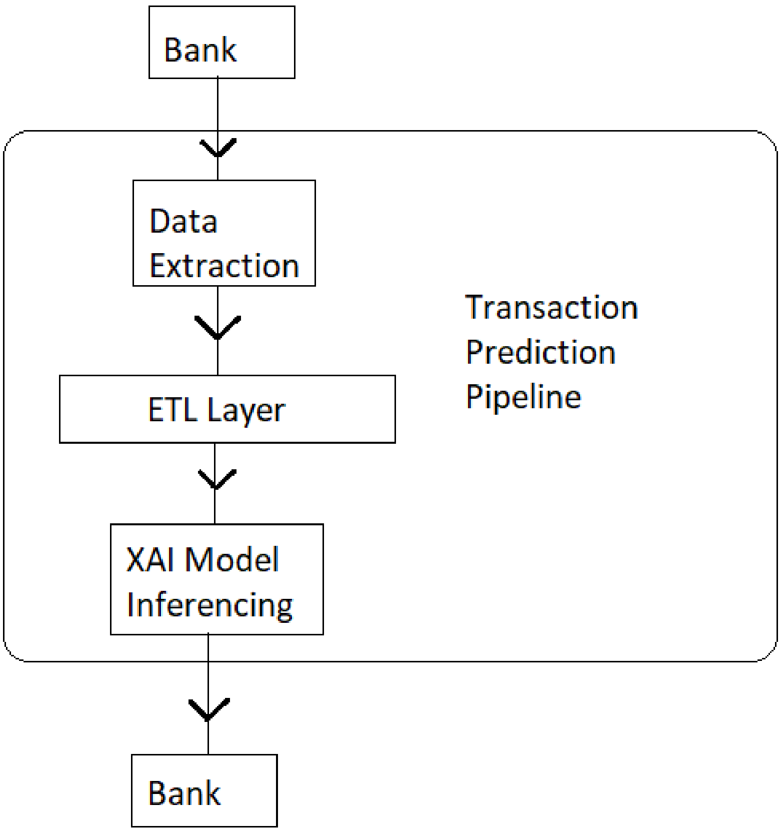

Figure 1 The Model Run Time Flow

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 Temenos ETL layer – creating the features required by the model. The ETL layer is not necessary in every single pipeline, but is available if required.

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 may be unique to information sent by each bank.

Raw feature	Explanation
Customer Debit Transaction Sum (7D)	Sum of debit transaction amount per customer in 7 day window
Customer Debit Transaction Count (7D)	Count of debit transaction amount per customer in 7 day window
Customer Debit Transaction Avg (7D)	Average of debit transaction amount per customer in 7 day window
Customer Debit Transaction Sum (30D)	Sum of debit transaction amount per customer in 30 day window
Customer Debit Transaction Count (30D)	Count of debit transaction amount per customer in 30 day window
Customer Debit Transaction Avg (30D)	Average of debit transaction amount per customer in 30 day window
Customer Debit Transaction Sum (90D)	Sum of debit transaction amount per customer in 90 day window
Customer Debit Transaction Count (90D)	Count of debit transaction amount per customer in 90 day window
Customer Debit Transaction Avg (90D)	Average of debit transaction amount per customer in 90 day window
Customer Debit Transaction Sum (120D)	Sum of debit transaction amount per customer in 120 day window
Customer Debit Transaction Count (120D)	Count of debit transaction amount per customer in 120 day window
Customer Debit Transaction Avg (120D)	Average of debit transaction amount per customer in 120 day window
Customer Debit Transaction Sum (150D)	Sum of debit transaction amount per customer in 150 day window
Customer Debit Transaction Count (150D)	Count of debit transaction amount per customer in 150 day window
Customer Debit Transaction Avg (150D)	Average of debit transaction amount per customer in 150 day window
Customer Credit Transaction Sum (7D)	Sum of credit transaction amount per customer in 7 day window
Customer Credit Transaction Count (7D)	Count of credit transaction amount per customer in 7 day window
Customer Credit Transaction Avg (7D)	Average of credit transaction amount per customer in 7 day window
Customer Credit Transaction Sum (30D)	Sum of credit transaction amount per customer in 30 day window
Customer Credit Transaction Count (30D)	Count of credit transaction amount per customer in 30 day window
Customer Credit Transaction Avg (30D)	Average of credit transaction amount per customer in 30 day window
Customer Credit Transaction Sum (90D)	Sum of credit transaction amount per customer in 90 day window
Customer Credit Transaction Count	Count of credit transaction amount per customer in

(90D)	90 day window
Customer Credit Transaction Avg (90D)	Average of credit transaction amount per customer in 90 day window
Customer Credit Transaction Sum (120D)	Sum of credit transaction amount per customer in 120 day window
Customer Credit Transaction Count (120D)	Count of credit transaction amount per customer in 120 day window
Customer Credit Transaction Avg (120D)	Average of credit transaction amount per customer in 120 day window
Customer Credit Transaction Sum (150D)	Sum of credit transaction amount per customer in 150 day window
Customer Credit Transaction Count (150D)	Count of credit transaction amount per customer in 150 day window
Customer Credit Transaction Avg (150D)	Average of credit transaction amount per customer in 150 day window
Customer Avg Debit to Avg Credit Ratio (7D)	Ratio of average debit and credit done by a customer in 7 day window
Customer Avg Debit to Avg Credit Ratio (30D)	Ratio of average debit and credit done by a customer in 30 day window
Customer Avg Debit to Avg Credit Ratio (90D)	Ratio of average debit and credit done by a customer in 90 day window
Customer Avg Debit to Avg Credit Ratio (120D)	Ratio of average debit and credit done by a customer in 120 day window
Customer Avg Debit to Avg Credit Ratio (150D)	Ratio of average debit and credit done by a customer in 150 day window
Customer Label Debit Transaction Sum (7D)	sum of transaction amount per customer (after removing extra labels and credits) for each label in 7 day window
Customer Label Debit Transaction Count (7D)	count of transaction amount per customer(after removing extra labels and credits) for each label in 7 day window
Customer Label Debit Transaction Max (7D)	max of transaction amount per customer(after removing extra labels and credits) for each label in 7 day window
Customer Label Debit Transaction Min (7D)	min of transaction amount per customer(after removing extra labels and credits) for each label in 7 day window
Customer Label Debit Transaction Avg (7D)	average of transaction amount per customer(after removing extra labels and credits) for each label in 7 day window
Customer Label Debit Transaction Sum (30D)	sum of transaction amount per customer(after removing extra labels and credits) for each label in 30 day window
Customer Label Debit Transaction Count (30D)	count of transaction amount per customer(after removing extra labels and credits) for each label in 30 day window
Customer Label Debit Transaction Max (30D)	max of transaction amount per customer(after removing extra labels and credits) for each label in 30 day window

Customer Label Debit Transaction Min (30D)	min of transaction amount per customer(after removing extra labels and credits) for each label in 30 day window
Customer Label Debit Transaction Avg (30D)	average of transaction amount per customer(after removing extra labels and credits) for each label in 30 day window
Customer Label Debit Transaction Sum (90D)	sum of transaction amount per customer(after removing extra labels and credits) for each label in 90 day window
Customer Label Debit Transaction Count (90D)	count of transaction amount per customer(after removing extra labels and credits) for each label in 90 day window
Customer Label Debit Transaction Max (90D)	max of transaction amount per customer(after removing extra labels and credits) for each label in 90 day window
Customer Label Debit Transaction Min (90D)	min of transaction amount per customer(after removing extra labels and credits) for each label in 90 day window
Customer Label Debit Transaction Avg (90D)	average of transaction amount per customer(after removing extra labels and credits) for each label in 90 day window
Customer Label Debit Transaction Sum (120D)	sum of transaction amount per customer(after removing extra labels and credits) for each label in 120 day window
Customer Label Debit Transaction Count (120D)	count of transaction amount per customer(after removing extra labels and credits) for each label in 120 day window
Customer Label Debit Transaction Max (120D)	max of transaction amount per customer(after removing extra labels and credits) for each label in 120 day window
Customer Label Debit Transaction Min (120D)	min of transaction amount per customer(after removing extra labels and credits) for each label in 120 day window
Customer Label Debit Transaction Avg (120D)	average of transaction amount per customer(after removing extra labels and credits) for each label in 120 day window
Customer Label Debit Transaction Sum (150D)	sum of transaction amount per customer(after removing extra labels and credits) for each label in 150 day window
Customer Label Debit Transaction Count (150D)	count of transaction amount per customer(after removing extra labels and credits) for each label in 150 day window
Customer Label Debit Transaction Max (150D)	max of transaction amount per customer(after removing extra labels and credits) for each label in 150 day window
Customer Label Debit Transaction Min (150D)	min of transaction amount per customer(after removing extra labels and credits) for each label in 150 day window
Customer Label Debit Transaction Avg (150D)	average of transaction amount per customer(after removing extra labels and credits) for each label in

	150 day window
Customer Label Previous Transaction Day	day on which last transaction was done by the customer depending on the label of the current transaction
#Days from Customer Label Previous Transaction	days between the current transaction and last transaction depending on the label of the current transaction
#Days from Customer Previous Transaction	the date difference in days between the current and previous transaction per customer for any label
Customer Debit Transaction Sum (7D) - Selected Labels	sum of transaction amount per customer(after removing extra labels and credits) in 7 day window
Customer Debit Transaction Count (7D) - Selected Labels	count of transaction amount per customer(after removing extra labels and credits) in 7 day window
Customer Debit Transaction Max (7D) - Selected Labels	max of transaction amount per customer(after removing extra labels and credits) in 7 day window
Customer Debit Transaction Min (7D) - Selected Labels	min of transaction amount per customer(after removing extra labels and credits) in 7 day window
Customer Debit Transaction Avg (7D) - Selected Labels	average of transaction amount per customer (after removing extra labels and credits) in 7 day window
Customer Debit Transaction Sum (30D) - Selected Labels	sum of transaction amount per customer(after removing extra labels and credits) in 30 day window
Customer Debit Transaction Count (30D) - Selected Labels	count of transaction amount per customer (after removing extra labels and credits) in 30 day window
Customer Debit Transaction Max (30D) - Selected Labels	max of transaction amount per customer (after removing extra labels and credits) in 30 day window
Customer Debit Transaction Min (30D) - Selected Labels	min of transaction amount per customer(after removing extra labels and credits) in 30 day window
Customer Debit Transaction Avg (30D) - Selected Labels	average of transaction amount per customer (after removing extra labels and credits) in 30 day window
Customer Debit Transaction Sum (90D) - Selected Labels	sum of transaction amount per customer (after removing extra labels and credits) in 90 day window
Customer Debit Transaction Count (90D) - Selected Labels	count of transaction amount per customer (after removing extra labels and credits) in 90 day window
Customer Debit Transaction Max (90D) - Selected Labels	max of transaction amount per customer (after removing extra labels and credits) in 90 day window
Customer Debit Transaction Min (90D) - Selected Labels	min of transaction amount per customer(after removing extra labels and credits) in 90 day window
Customer Debit Transaction Avg (90D) - Selected Labels	average of transaction amount per customer (after removing extra labels and credits) in 90 day window
Customer Debit Transaction Sum (120D) - Selected Labels	sum of transaction amount per customer (after removing extra labels and credits) in 120 day window
Customer Debit Transaction Count (120D) - Selected Labels	count of transaction amount per customer (after removing extra labels and credits) in 120 day window
Customer Debit Transaction Max (120D) - Selected Labels	max of transaction amount per customer (after removing extra labels and credits) in 120 day window
Customer Debit Transaction Min (120D) - Selected Labels	min of transaction amount per customer(after removing extra labels and credits) in 120 day window
Customer Debit Transaction Avg	average of transaction amount per customer (after

(120D) - Selected Labels	removing extra labels and credits) in 120 day window
Customer Debit Transaction Sum (150D) - Selected Labels	sum of transaction amount per customer (after removing extra labels and credits) in 150 day window
Customer Debit Transaction Count (150D) - Selected Labels	count of transaction amount per customer (after removing extra labels and credits) in 150 day window
Customer Debit Transaction Max (150D) - Selected Labels	max of transaction amount per customer (after removing extra labels and credits) in 150 day window
Customer Debit Transaction Min (150D) - Selected Labels	min of transaction amount per customer(after removing extra labels and credits) in 150 day window
Customer Debit Transaction Avg (150D) - Selected Labels	average of transaction amount per customer (after removing extra labels and credits) in 150 day window
Customer Label Transaction Density (7D)	Average of date difference between the transactions for each customer per label in 7 day window
Label Transaction Density (7D)	Average of date difference between the transactions for each label in 7 day window
Customer Label Transaction Density (30D)	Average of date difference between the transactions for each customer per label in 30 day window
Label Transaction Density (30D)	Average of date difference between the transactions for each label in 30 day window
Customer Label Transaction Density (90D)	Average of date difference between the transactions for each customer per label in 90 day window
Label Transaction Density (90D)	Average of date difference between the transactions for each label in 90 day window
Customer Label Transaction Density (120D)	Average of date difference between the transactions for each customer per label in 120 day window
Label Transaction Density (120D)	Average of date difference between the transactions for each label in 120 day window
Customer Label Transaction Density (150D)	Average of date difference between the transactions for each customer per label in 150 day window
Label Transaction Density (150D)	Average of date difference between the transactions for each label in 150 day window
Customer Label ProportionCount (7D)	proportion count of each label compared to total count of all labels for a customer in 7 day window
Customer Label ProportionCount (30D)	proportion count of each label compared to total count of all labels for a customer in 30 day window
Customer Label ProportionCount (90D)	proportion count of each label compared to total count of all labels for a customer in 90 day window
Customer Label ProportionCount (120D)	proportion count of each label compared to total count of all labels for a customer in 120 day window
Customer Label ProportionCount (150D)	proportion count of each label compared to total count of all labels for a customer in 150 day window
Local Ratio of Label Density (7D)	ratio of number of days difference between the current and previous transaction and Average of date difference between the transactions for each customer per label in 7 day window
Global Ratio of Label Density (7D)	ratio of number of days difference between the current and previous transaction and Average of date difference between the transactions for each

	customer in 7 day window
Local Ratio of Label Density (30D)	ratio of number of days difference between the current and previous transaction and Average of date difference between the transactions for each customer per label in 30 day window
Global Ratio of Label Density (30D)	ratio of number of days difference between the current and previous transaction and Average of date difference between the transactions for each customer in 30 day window
Local Ratio of Label Density (90D)	ratio of number of days difference between the current and previous transaction and Average of date difference between the transactions for each customer per label in 90 day window
Global Ratio of Label Density (90D)	ratio of number of days difference between the current and previous transaction and Average of date difference between the transactions for each customer in 90 day window
Local Ratio of Label Density (120D)	ratio of number of days difference between the current and previous transaction and Average of date difference between the transactions for each customer per label in 120 day window
Global Ratio of Label Density (120D)	ratio of number of days difference between the current and previous transaction and Average of date difference between the transactions for each customer in 120 day window
Local Ratio of Label Density (150D)	ratio of number of days difference between the current and previous transaction and Average of date difference between the transactions for each customer per label in 150 day window
Global Ratio of Label Density (150D)	ratio of number of days difference between the current and previous transaction and Average of date difference between the transactions for each customer in 150 day window

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 the bank’s database (shown in the pre-processing table above). A sample payload for a single inference is shown below:


```
{
  "data": {
    "credit_debit_flag": "debit",
    "transaction_type": "wire_transfer",
    "functional_transaction_amount": 0.01,
    "original_transaction_currency": "GBP",
    "salary_or_turnover": 10744,
    "individual_age": 13.11,
    "day_of_the_week": "Sat",
    "is_weekend": "False",
    "outgoing_total_tx_value_1D": -53.77,
```

```
    "outgoing_num_tx_value_1D": 0,
    "incoming_total_tx_value_1D": -72.41,
    "incoming_num_tx_value_1D": 0,
    "outgoing_total_tx_value_3D": -58.05,
    "incoming_total_tx_value_3D": -41.08,
    "outgoing_total_tx_value_7D": -41.16,
    "outgoing_num_tx_value_7D": 0,
    "incoming_total_tx_value_7D": -12.25,
    "incoming_num_tx_value_7D": 0,
    "outgoing_total_tx_value_1M": 0,
    "outgoing_average_tx_value_1M": 1.5079999999999998,
    "outgoing_num_tx_value_1M": 7,
    "incoming_total_tx_value_1M": 0,
    "incoming_num_tx_value_1M": 0,
    "outgoing_total_tx_value_3M": 0,
    "incoming_total_tx_value_3M": 0,
    "incoming_average_tx_value_3M": 475.07,
    "incoming_max_tx_value_3M": 938.83,
    "outgoing_total_tx_value_6M": 0,
    "incoming_average_tx_value_6M": 543.3858333333333,
    "incoming_max_tx_value_6M": 938.83,
    "outgoing_total_tx_value_12M": 0,
    "outgoing_average_tx_value_1M_ratio": 4.165175698983442e-7,
    "outgoing_max_tx_value_1M_ratio": 3.4180734301677805e-9,
    "incoming_average_tx_value_1M_ratio": 1.190872027832013e-7,
    "incoming_max_tx_value_1M_ratio": 5.961902252227815e-8,
    "outgoing_average_tx_value_6M_ratio": 0.000002369046089283111,
    "outgoing_max_tx_value_6M_ratio": 3.173925328012939e-9,
    "incoming_average_tx_value_6M_ratio": 7.090623725534245e-7,
    "incoming_max_tx_value_6M_ratio": 5.961902252227815e-8,
    "risk_level": 0,
    "cash_withdrawals_30d": 0,
    "cash_deposits_30d": 0,
    "transaction_sum_15d_>=_cash_withdrawal_sum_30d": "True",
    "credit_country_risk": "Low",
    "functional_transaction_amount_>_1000": "False",
    "cust_max_local_amount_avg_1m_>=_acc_max_local_amount_avg_6m_x1.5": "False",
    "functional_transaction_amount_>_10000": "False",
    "acc_num_tx_sum_1m_>=_acc_num_tx_avg_12m": "",
    "num_tx_sum_1m_>_num_tx_sum_6m_shift1m_x1.5": "False",
    "cash_withdrawals_sum_7d": 0,
    "cash_withdrawals_count_7d": 0,
    "num_tx_1d_>_num_tx_30d_shift1d": "False",
    "outgoing_tx_sum_1m_>_incoming_tx_sum_6m": "False",
    "functional_transaction_amount_>_1000000": "False",
    "debit_sum_tx_1m_>_credit_sum_tx_6m_shift1m": "False",
    "functional_transaction_amount_>_250000": "False",
    "functional_transaction_amount_>_100000": "False",
    "original_transaction_amount_>_10000": "False",
    "transaction_amount_OUTSIDE_cust_debits_per_counterparty_std": "False",
    "new_currency_for_customer": "False",
    "debits_count_1d_>_max_daily_debits_50d_shift1d_x1.2": "False",
    "new_transaction_type_for_customer": "False",
    "num_tx_count_1m_>_1000": "False",
    "new_3rd_party_payment": "False",
    "swift_num_tx_avg_2m_>_swift_num_tx_avg_10m_shift2m": "",
    "swift_num_tx_1d_>_swift_num_tx_avg_20d_shift1d": "False",
    "cust_num_tx_1m_>=_cust_num_tx_3m_shift1m": "False",
    "cash_deposits_max_14d": 0,
    "cash_withdrawals_count_14d": 0
  },
  "modelKey": "f0281b27-f875-4d21-9c49-3645c6e824d1"
}
```

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": [{
    "credit_debit_flag": "debit",
    "transaction_type": "wire_transfer",
    "functional_transaction_amount": 0.01,
    "original_transaction_currency": "GBP",
    "salary_or_turnover": 10744,
    "individual_age": 13.11,
    "day_of_the_week": "Sat",
    "is_weekend": "False",
    "outgoing_total_tx_value_1D": -53.77,
    "outgoing_num_tx_value_1D": 0,
    "incoming_total_tx_value_1D": -72.41,
    "incoming_num_tx_value_1D": 0,
    "outgoing_total_tx_value_3D": -58.05,
    "incoming_total_tx_value_3D": -41.08,
    "outgoing_total_tx_value_7D": -41.16,
    "outgoing_num_tx_value_7D": 0,
    "incoming_total_tx_value_7D": -12.25,
    "incoming_num_tx_value_7D": 0,
    "outgoing_total_tx_value_1M": 0,
    "outgoing_average_tx_value_1M": 1.5079999999999998,
    "outgoing_num_tx_value_1M": 7,
    "incoming_total_tx_value_1M": 0,
    "incoming_num_tx_value_1M": 0,
    "outgoing_total_tx_value_3M": 0,
    "incoming_total_tx_value_3M": 0,
    "incoming_average_tx_value_3M": 475.07,
    "incoming_max_tx_value_3M": 938.83,
    "outgoing_total_tx_value_6M": 0,
    "incoming_average_tx_value_6M": 543.3858333333333,
    "incoming_max_tx_value_6M": 938.83,
    "outgoing_total_tx_value_12M": 0,
    "outgoing_average_tx_value_1M_ratio": 4.165175698983442e-7,
    "outgoing_max_tx_value_1M_ratio": 3.4180734301677805e-9,
    "incoming_average_tx_value_1M_ratio": 1.190872027832013e-7,
    "incoming_max_tx_value_1M_ratio": 5.961902252227815e-8,
    "outgoing_average_tx_value_6M_ratio": 0.000002369046089283111,
    "outgoing_max_tx_value_6M_ratio": 3.173925328012939e-9,
    "incoming_average_tx_value_6M_ratio": 7.090623725534245e-7,
    "incoming_max_tx_value_6M_ratio": 5.961902252227815e-8,
    "risk_level": 0,
    "cash_withdrawals_30d": 0,
    "cash_deposits_30d": 0,
    "transaction_sum_15d_>=_cash_withdrawal_sum_30d": "True",
    "credit_country_risk": "Low",
    "functional_transaction_amount_>_1000": "False",
    "cust_max_local_amount_avg_1m_>=_acc_max_local_amount_avg_6m_x1.5": "False",
    "functional_transaction_amount_>_10000": "False",
    "acc_num_tx_sum_1m_>=_acc_num_tx_avg_12m": "",
    "num_tx_sum_1m_>_num_tx_sum_6m_shift1m_x1.5": "False",
    "cash_withdrawals_sum_7d": 0,
    "cash_withdrawals_count_7d": 0,
    "num_tx_1d_>_num_tx_30d_shift1d": "False",
    "outgoing_tx_sum_1m_>_incoming_tx_sum_6m": "False",
    "functional_transaction_amount_>_1000000": "False",
    "debit_sum_tx_1m_>_credit_sum_tx_6m_shift1m": "False",
    "functional_transaction_amount_>_250000": "False",
```

```
    "functional_transaction_amount_>_100000": "False",
    "original_transaction_amount_>_10000": "False",
    "transaction_amount_OUTSIDE_cust_debits_per_counterparty_std": "False",
    "new_currency_for_customer": "False",
    "debits_count_1d_>_max_daily_debits_50d_shift1d_x1.2": "False",
    "new_transaction_type_for_customer": "False",
    "num_tx_count_1m_>_1000": "False",
    "new_3rd_party_payment": "False",
    "swift_num_tx_avg_2m_>_swift_num_tx_avg_10m_shift2m": "",
    "swift_num_tx_1d_>_swift_num_tx_avg_20d_shift1d": "False",
    "cust_num_tx_1m_>=_cust_num_tx_3m_shift1m": "False",
    "cash_deposits_max_14d": 0,
    "cash_withdrawals_count_14d": 0
  },, {
    "credit_debit_flag": "debit",
    "transaction_type": "wire_transfer",
    "functional_transaction_amount": 0.01,
    "original_transaction_currency": "GBP",
    "salary_or_turnover": 10744,
    "individual_age": 13.11,
    "day_of_the_week": "Sat",
    "is_weekend": "False",
    "outgoing_total_tx_value_1D": -53.77,
    "outgoing_num_tx_value_1D": 0,
    "incoming_total_tx_value_1D": -72.41,
    "incoming_num_tx_value_1D": 0,
    "outgoing_total_tx_value_3D": -58.05,
    "incoming_total_tx_value_3D": -41.08,
    "outgoing_total_tx_value_7D": -41.16,
    "outgoing_num_tx_value_7D": 0,
    "incoming_total_tx_value_7D": -12.25,
    "incoming_num_tx_value_7D": 0,
    "outgoing_total_tx_value_1M": 0,
    "outgoing_average_tx_value_1M": 1.5079999999999998,
    "outgoing_num_tx_value_1M": 7,
    "incoming_total_tx_value_1M": 0,
    "incoming_num_tx_value_1M": 0,
    "outgoing_total_tx_value_3M": 0,
    "incoming_total_tx_value_3M": 0,
    "incoming_average_tx_value_3M": 475.07,
    "incoming_max_tx_value_3M": 938.83,
    "outgoing_total_tx_value_6M": 0,
    "incoming_average_tx_value_6M": 543.3858333333333,
    "incoming_max_tx_value_6M": 938.83,
    "outgoing_total_tx_value_12M": 0,
    "outgoing_average_tx_value_1M_ratio": 4.165175698983442e-7,
    "outgoing_max_tx_value_1M_ratio": 3.4180734301677805e-9,
    "incoming_average_tx_value_1M_ratio": 1.190872027832013e-7,
    "incoming_max_tx_value_1M_ratio": 5.961902252227815e-8,
    "outgoing_average_tx_value_6M_ratio": 0.000002369046089283111,
    "outgoing_max_tx_value_6M_ratio": 3.173925328012939e-9,
    "incoming_average_tx_value_6M_ratio": 7.090623725534245e-7,
    "incoming_max_tx_value_6M_ratio": 5.961902252227815e-8,
    "risk_level": 0,
    "cash_withdrawals_30d": 0,
    "cash_deposits_30d": 0,
    "transaction_sum_15d_>=_cash_withdrawal_sum_30d": "True",
    "credit_country_risk": "Low",
    "functional_transaction_amount_>_1000": "False",
    "cust_max_local_amount_avg_1m_>=_acc_max_local_amount_avg_6m_x1.5": "False",
    "functional_transaction_amount_>_10000": "False",
    "acc_num_tx_sum_1m_>=_acc_num_tx_avg_12m": "",
    "num_tx_sum_1m_>_num_tx_sum_6m_shift1m_x1.5": "False",
    "cash_withdrawals_sum_7d": 0,
    "cash_withdrawals_count_7d": 0,
    "num_tx_1d_>_num_tx_30d_shift1d": "False",
```

```
    "outgoing_tx_sum_1m_>_incoming_tx_sum_6m": "False",
    "functional_transaction_amount_>_1000000": "False",
    "debit_sum_tx_1m_>_credit_sum_tx_6m_shift1m": "False",
    "functional_transaction_amount_>_250000": "False",
    "functional_transaction_amount_>_100000": "False",
    "original_transaction_amount_>_10000": "False",
    "transaction_amount_OUTSIDE_cust_debits_per_counterparty_std": "False",
    "new_currency_for_customer": "False",
    "debits_count_1d_>_max_daily_debits_50d_shift1d_x1.2": "False",
    "new_transaction_type_for_customer": "False",
    "num_tx_count_1m_>_1000": "False",
    "new_3rd_party_payment": "False",
    "swift_num_tx_avg_2m_>_swift_num_tx_avg_10m_shift2m": "",
    "swift_num_tx_1d_>_swift_num_tx_avg_20d_shift1d": "False",
    "cust_num_tx_1m_>=_cust_num_tx_3m_shift1m": "False",
    "cash_deposits_max_14d": 0,
    "cash_withdrawals_count_14d": 0
  },],
  "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.

The definitions of the keys are as follows:

‘externalId’: A reference ID for the inference, if provided

‘bucketName': Predicted bucket the model score places the transaction into. A higher bucket represents a higher chance that a Transaction will take place in the next 7 days.

‘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': None,
 'bucketName': 'Bucket #1',
 'inferenceUri': 'api/audit/model/2946ffbb-56ef-4ac0-a3dc-31fd094c65dc?time=1593772175671',
 'score': 0.0020491166334455357,
 'detailedInformation': {'results': {}},
 'inferenceResultEncoded': -2147483445.0,
 'inferenceResultDecoded': '0',
 'mapOutputRatios': {'0': 0.9979508833665545, '1': 0.002049116633445487},
 'warnings': [{'featureName': 'incoming_total_tx_value_3M',
   'currentValue': '0',
   'minValue': '1891.16',
   'maxValue': '1685617.47'}],
 'status': 'SUCCESS'}
```


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 the transaction. 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.


```
{'externalId': None,
 'bucketName': 'Bucket #1',
 'inferenceUri': 'api/audit/model/2946ffbb-56ef-4ac0-a3dc-31fd094c65dc?time=1593772717202',
 'score': 0.0020491166334455357,
 'detailedInformation': {'results': {'compactResults': [{'rank': 1,
     'name': '0',
     'overall': 0.9979508833665545,
     'rules': [{'featureName': 'functional_transaction_amount_>_10000',
       'drivers': ['False']},
      {'featureName': 'transaction_amount_OUTSIDE_cust_debits_per_counterparty_std',
       'drivers': ['False']},
      {'featureName': 'functional_transaction_amount_>_250000',
       'drivers': ['False']},
      {'featureName': 'incoming_num_tx_value_7D', 'drivers': ['low']},
      {'featureName': 'incoming_max_tx_value_3M', 'drivers': ['low']},
      {'featureName': 'incoming_max_tx_value_6M', 'drivers': ['low']},
      {'featureName': 'credit_debit_flag', 'drivers': ['debit']},
      {'featureName': 'individual_age', 'drivers': ['low']},
      {'featureName': 'cust_num_tx_1m_>=_cust_num_tx_3m_shift1m',
       'drivers': ['False']},
      {'featureName': 'outgoing_total_tx_value_6M', 'drivers': ['low']}],
     'color': 'rgb(35,105,159)'},
    {'rank': 2,
     'name': '1',
     'overall': 0.002049116633445487,
     'rules': [{'featureName': 'transaction_sum_15d_>=_cash_withdrawal_sum_30d',
       'drivers': ['True']},
      {'featureName': 'outgoing_num_tx_value_1D', 'drivers': ['0']},
      {'featureName': 'day_of_the_week', 'drivers': ['Sat']},
      {'featureName': 'salary_or_turnover', 'drivers': ['low']}],
     'color': 'rgb(196,39,41)'}],
   'fuzzyDrivers': [{'rank': 1,
     'name': '0',
     'overall': 0.9979508833665545,
     'rules': [{'feature': 'functional_transaction_amount_>_10000',
       'values': ['False'],
       'actualValue': 'False',
       'consonances': [{'firingStrength': 1.0,
         'dominance': 0.00110598355725257,
         'ratioAsPercentage': 0.38918201448428225,
```


Figure 3 Partial Detailed Response

4. Use Cases

4.1 Use Case 1

Description

User Makes an API call to the Transaction Prediction pipeline with a valid payload. The model returns a response with a score between 0 and 1 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 Prediction 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 score 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 Transaction Prediction 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 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 4 and Figure 5.

Figure 4 Rule View

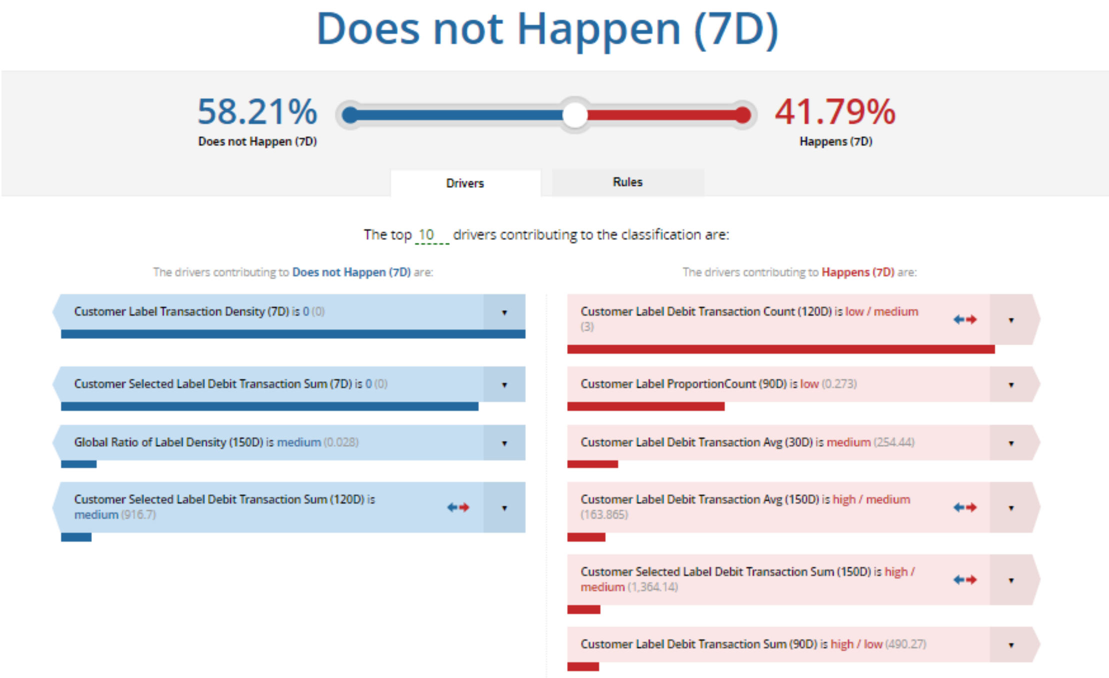

Figure 5 Driver View

4.3 Use Case 3

Description

The model has a transparent global rule base for arriving at individual transaction scores 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 Prediction 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 Prediction model is shown in Figure 6.

Rule Id	Antecedents	Result	Statistics
237	Customer Label Transaction Density (7D) is *0*	Does Not Happen (7D)	Dominance ∧ ★★★★☆ 2 Frequency 29,921
120	Customer Label Debit Transaction Count (120D) is *medium* Customer Label ProportionCount (90D) is *low*	Happens (7D)	Dominance ∧ ★★★★☆ 3 Frequency 17,272
280	Customer Selected Label Debit Transaction Count (30D) is *high* Customer Selected Label Debit Transaction Count (120D) is *high* Global Ratio of Label Density (30D) is *medium*	Happens (7D)	Dominance ∧ ★★★★☆ 4 Frequency 6,416
196	Customer Label ProportionCount (120D) is *high*	Happens (7D)	Dominance ∧ ★★★★☆ 5 Frequency 15,311
213	Customer Label ProportionCount (150D) is *high*	Happens (7D)	Dominance ∧ ★★★★☆ 6 Frequency 15,239

Figure 6 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 Prediction 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 7) will create a new scenario with a user specified name – which by default is the same as the underlying rule base.

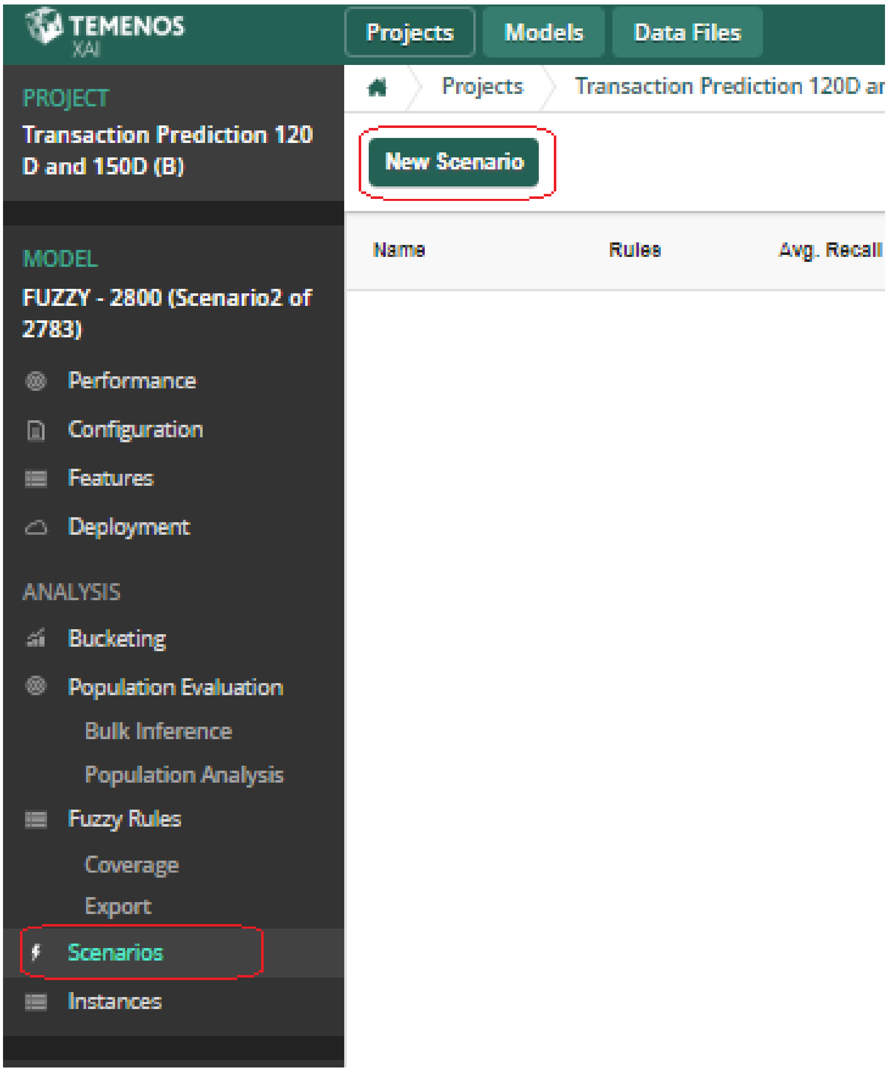

Figure 7: Create a new Scenario

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 8 and Figure 9:

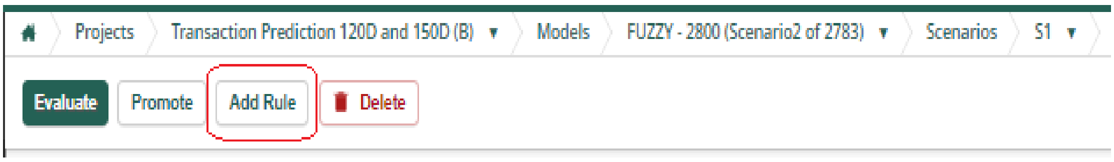

Figure 8 Add Rule Button

- 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.

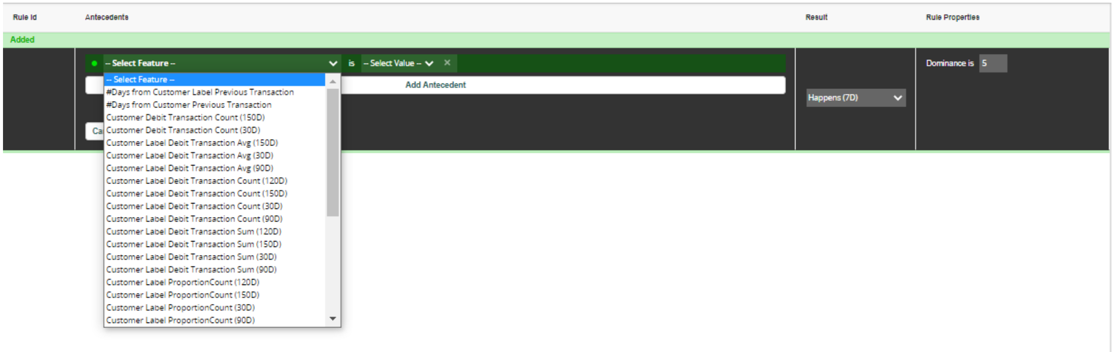

Figure 9 Adding a Rule

- 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 10)

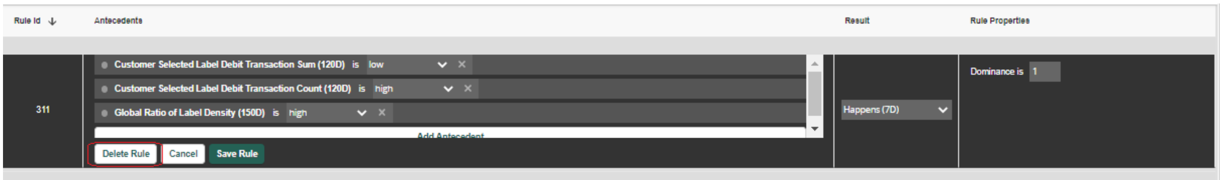

Figure 10 Deleting a Rule

- 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 11) with the performance change shown in a pop-up on the screen (Figure 12).

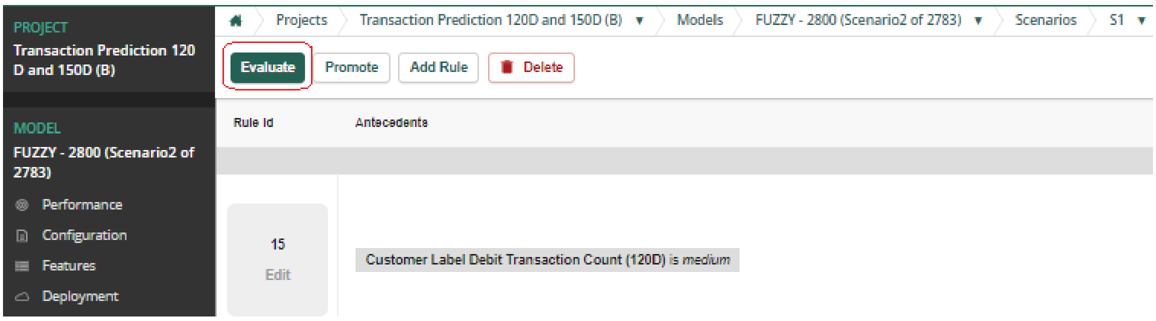

Figure 11 Evaluate Rule Button

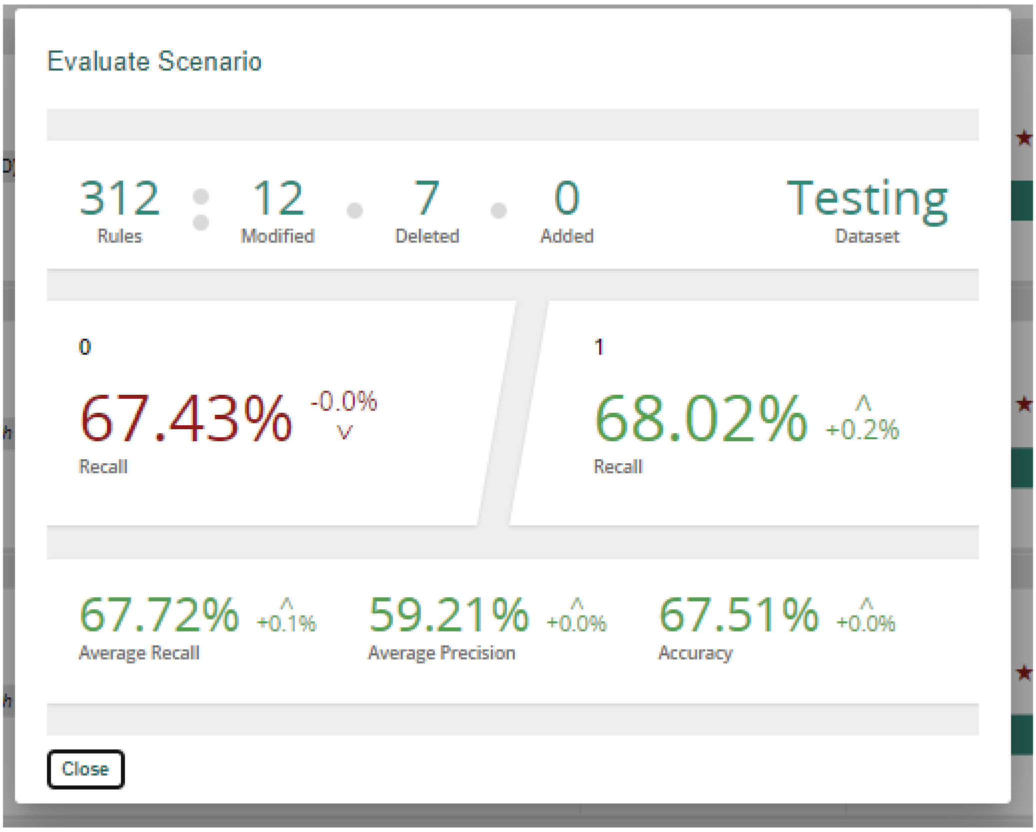

Figure 12 Evaluating the Scenario Performance

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

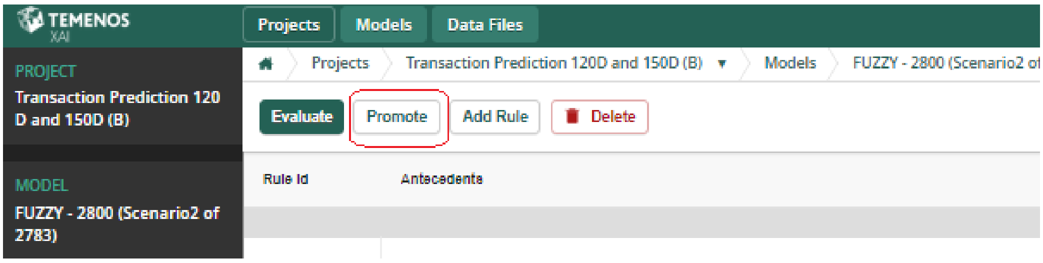

Figure 13 Promote a Scenario to a Model

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	The platform gives a warning that the rule can be derived from other rules – but still allows the user to add

combination of other rules in the rule base	the rule to the rule base.

4.5 Use Case 5

Description

Understanding the likelihood of occurrence of a transaction in next 7 days based on the behaviour of historical transactions allows the bank to consider and employ different strategies to retain transactions in different buckets.

Business Application

The bucketing tool allows a user to segment customers with known outcomes to be segmented in buckets and helps in understanding how each bucket performed. The instances all scored by the Transaction prediction model and are assigned scores between 0 and 1. The bucketing widget then groups the customers falling within a certain range – say 0 to 0.25 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. The platform has various goal seek tools like setting the number of buckets, equalizing the number of instances within buckets, setting bucket boundaries based on number of instances /minority instances required in each bucket etc.

Bucketing can be accessed clicking on the model from the model results list in the Builder page and then visiting 'Analysis → 'Bucketing' as shown in Figure 14.

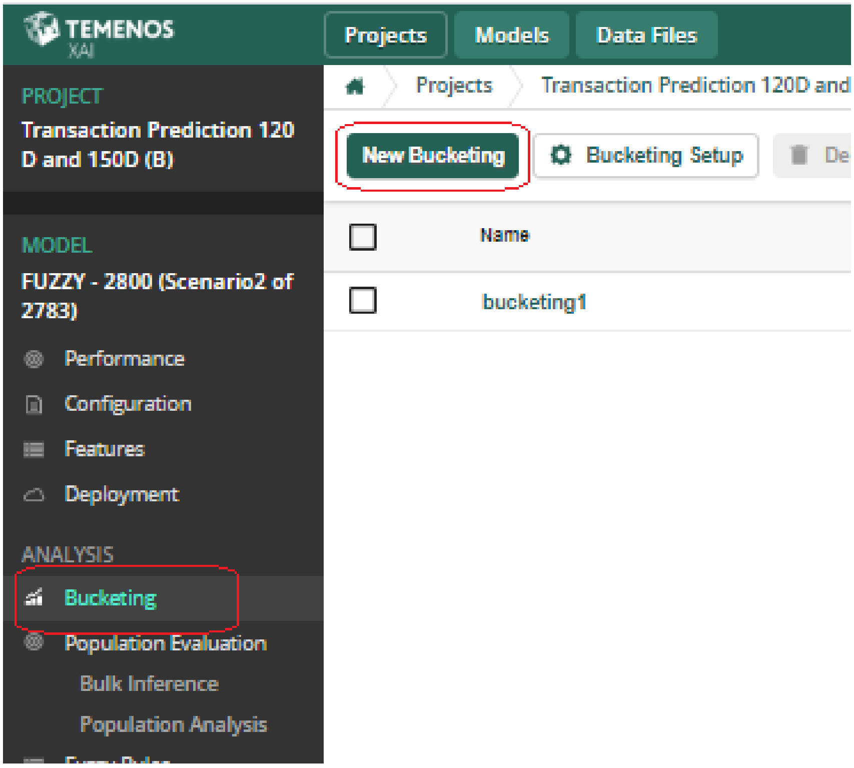

Figure 14 Create Bucketing

A screenshot of a bucketing is shown in Figure 15. The highlighted area shows where the options for changing bucket counts and using various goal seek scenarios are listed.

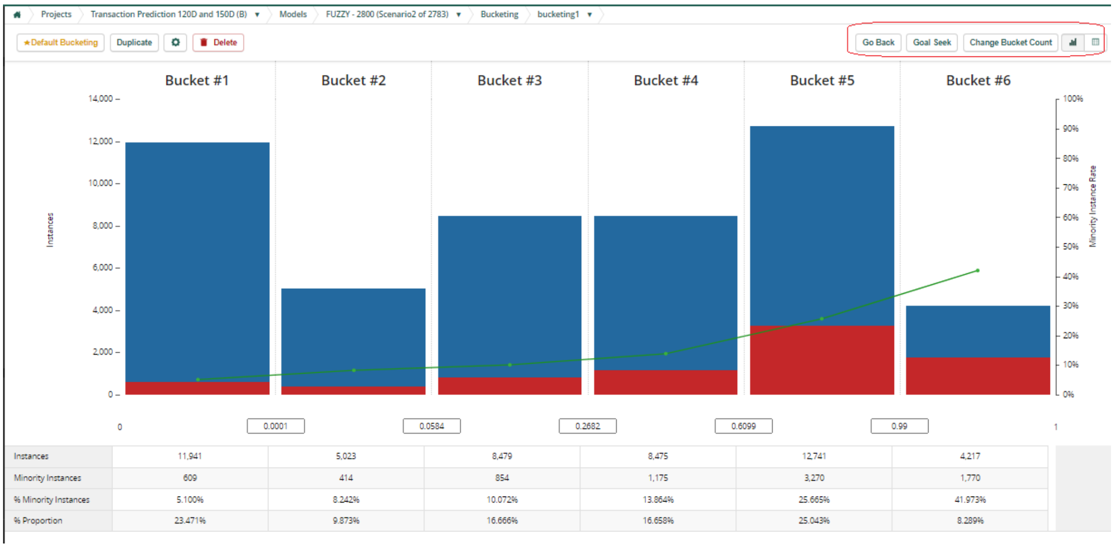

Figure 15 Sample Bucketing

Custom Statistics can be generated as follows:

On the Bucketing page go to ‘Bucketing Setup’ → ‘Aggregate Columns’. Let us assume that the user wants to see Average Score for a bucket added as a custom statistic on the bucketing output. Press ‘Add New Column’ to add a new Aggregate – followed by the type of aggregate desired – which is ‘Average’ in our case. Finally choose the desired feature from the ‘Feature’ drop down list and press ‘Save’ to add this aggregate column (Figure 16). The new setup will apply to any bucketing created for this model. Note that all old buckets will be deleted if the setup is changed.

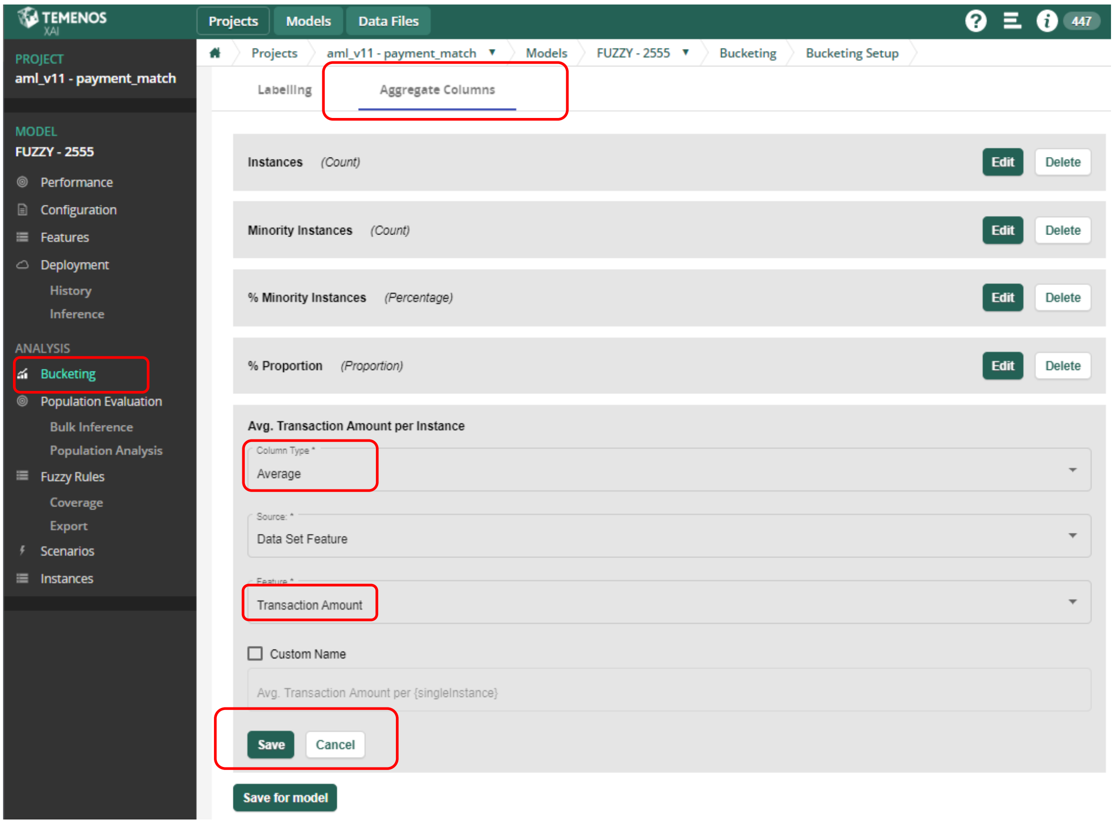

Figure 16 Adding Custom Statistics to a Bucketing

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."

4.6 Use Case 6

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.

The usage is shown in the screenshots below:

To start – load the file with the data on transactions 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 Evaluation→ Bulk Inference as shown in Figure 17.

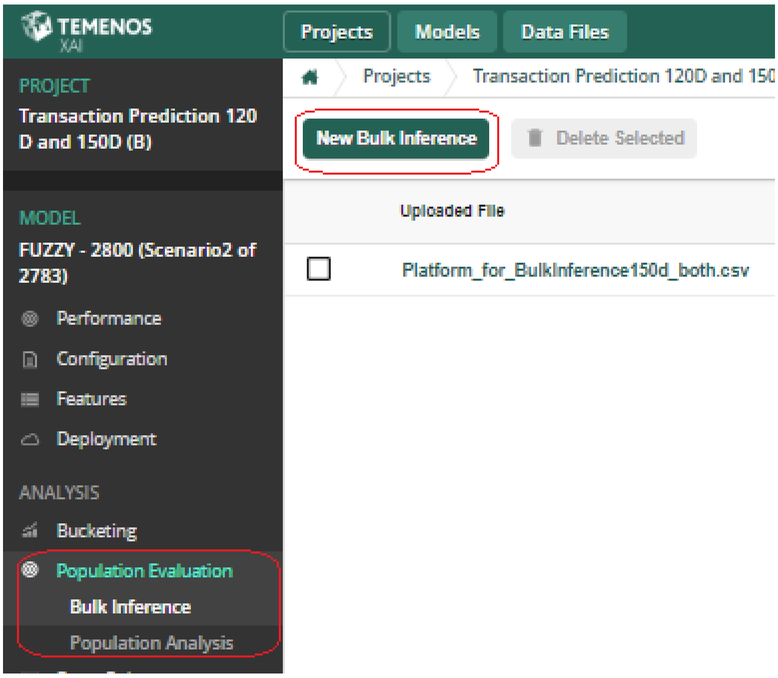

Figure 17 Create A Bulk Inference

Select a dataset of recent inferences to compare against the original modelling population, and upload it to the platform for analysis.

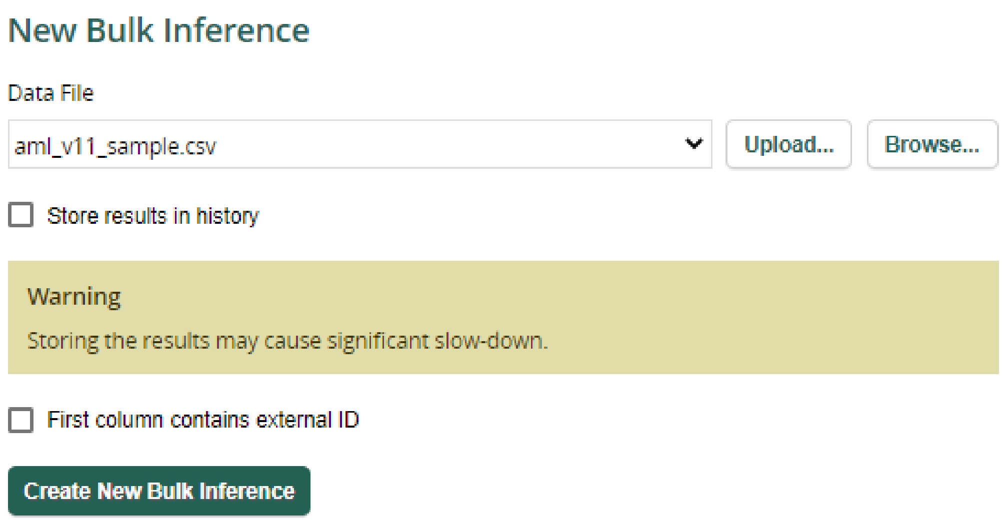

Figure 18 Upload bulk inference dataset

Once the bulk inference is complete, the summary results page will be displayed (Figure 19). It shows the shift in all the standard metrics between the original and uploaded populations.

Results | Population Stability | Characteristic Stability | Rejections

Summary

Data Set	No. Instances	Accuracy %	Average Precision %	Average Recall %
Original	50,876	67.319	60.006	67.712
Upload with outcome	4,728	68.253	60.041	67.959
Upload without outcome	0			
Rejections	0			

Confusion Matrix (uploaded data)

Actual	Predicted: Does not Happen (7D)	Predicted: Happens (7D)	Total	Recall %
Does not Happen (7D)	2,734	1,264	3,998	68.384
Happens (7D)	237	493	730	67.534

Confusion Matrix (original model build)

Actual	Predicted: Does not Happen (7D)	Predicted: Happens (7D)	Total	Recall %
Does not Happen (7D)	28,723	14,061	42,784	67.135
Happens (7D)	2,566	5,526	8,092	68.290

Gini

Uploaded	Original
0.431	0.450

KS-Stat

Uploaded	Original
0.362	0.355

Figure 19 – Results of bulk inference

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. Features, where the distribution has shifted significantly are highlighted in red, as shown in **Error! Reference source not found.**

An example of another models stability analysis is shown below.

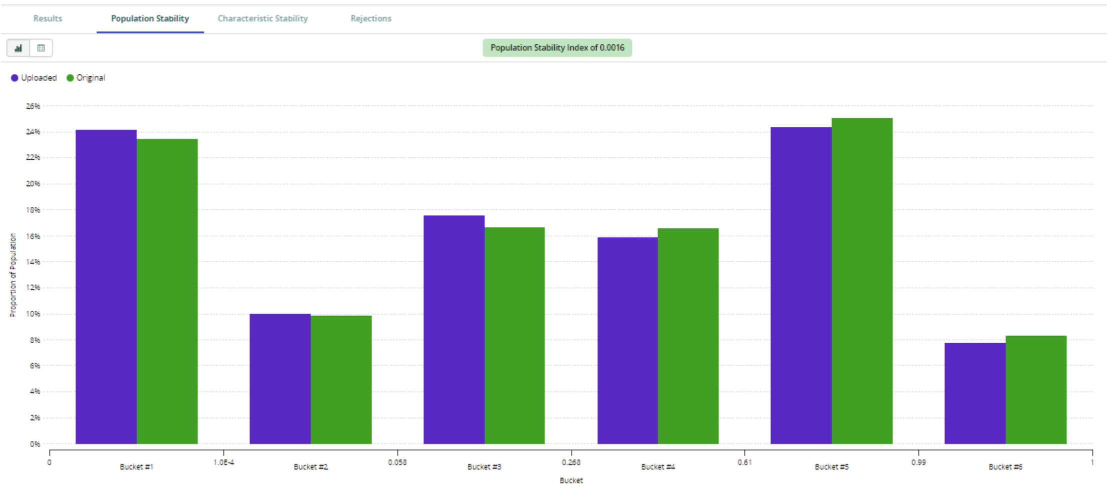

Figure 20 – Population Stability

Results | Population Stability | **Characteristic Stability** | Rejections

		Unique Values		PSI	IV			Gini		
Feature Name	**Type**	**Original**	**Upload**	**PSI**	**Original**	**Upload**	**Difference**	**Original**	**Upload**	**Di**
Customer Debit Transaction Cou…	Continuous	28	22	0.004	0.223	0.160	0.063	0.258	0.221	
Customer Debit Transaction Cou…	Continuous	95	73	0.014	0.261	0.232	0.029	0.284	0.263	
Customer Label Debit Transactio…	Continuous	> 1000	> 1000	0.006	0.344	0.323	0.021	0.294	0.284	
Customer Label Debit Transactio…	Continuous	19	11	8.1E-4	0.450	0.408	0.042	0.346	0.331	
Customer Label Debit Transactio…	Continuous	> 1000	> 1000	0.008	0.351	0.370	-0.019	0.273	0.247	
Customer Label Debit Transactio…	Continuous	> 1000	> 1000	0.007	0.406	0.370	0.036	0.339	0.311	
Customer Label Debit Transactio…	Continuous	48	28	0.002	0.635	0.562	0.073	0.429	0.407	
Customer Label Debit Transactio…	Continuous	> 1000	> 1000	0.011	0.334	0.358	-0.024	0.205	0.201	
Customer Label Debit Transactio…	Continuous	> 1000	> 1000	0.011	0.419	0.420	-0.001	0.343	0.336	
Customer Label Debit Transactio…	Continuous	56	33	0.002	0.684	0.621	0.063	0.445	0.426	
Customer Label Debit Transactio…	Continuous	> 1000	> 1000	0.008	0.421	0.433	-0.012	0.344	0.329	
Customer Label Debit Transactio…	Continuous	61	38	0.003	0.713	0.664	0.050	0.456	0.437	
Customer Label Debit Transactio…	Continuous	> 1000	> 1000	0.014	0.300	0.375	-0.075	0.156	0.153	
#Days from Customer Label Pre…	Mixed	382	307	0.003	0.505	0.490	0.015	0.385	0.368	

Figure 21 – Characteristic Stability

4.7 Use Case 7

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 Prediction 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.

4.8 Use Case 8

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
.

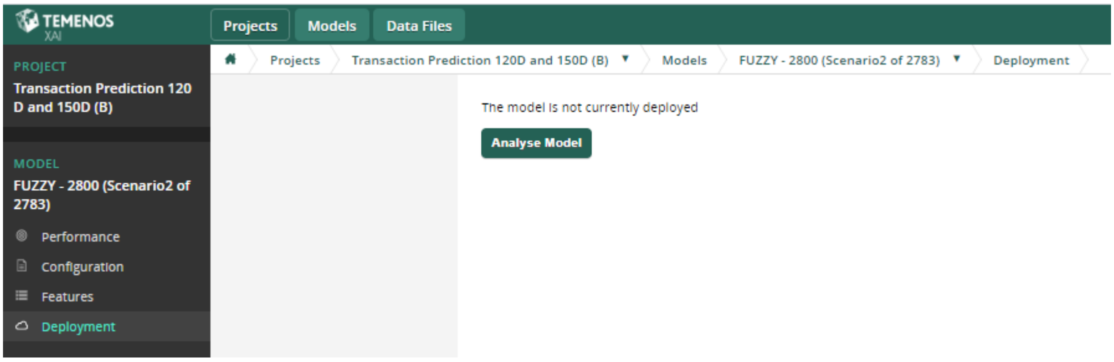

Figure 18 Simple Model Deployment

Upon making this change a reason needs to be provided for audit purposes (Figure 23).

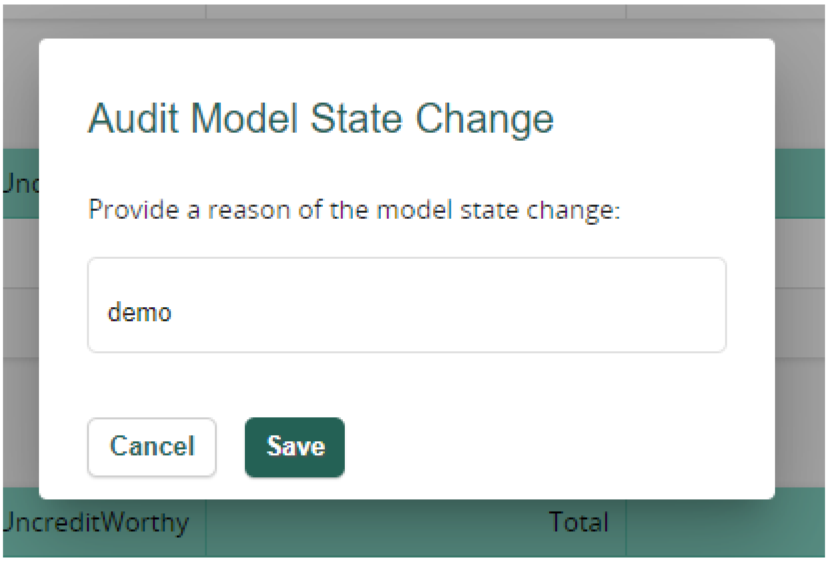

Figure 23: Audit reason

4.9 Use Case 9

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 24. 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 24, 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.

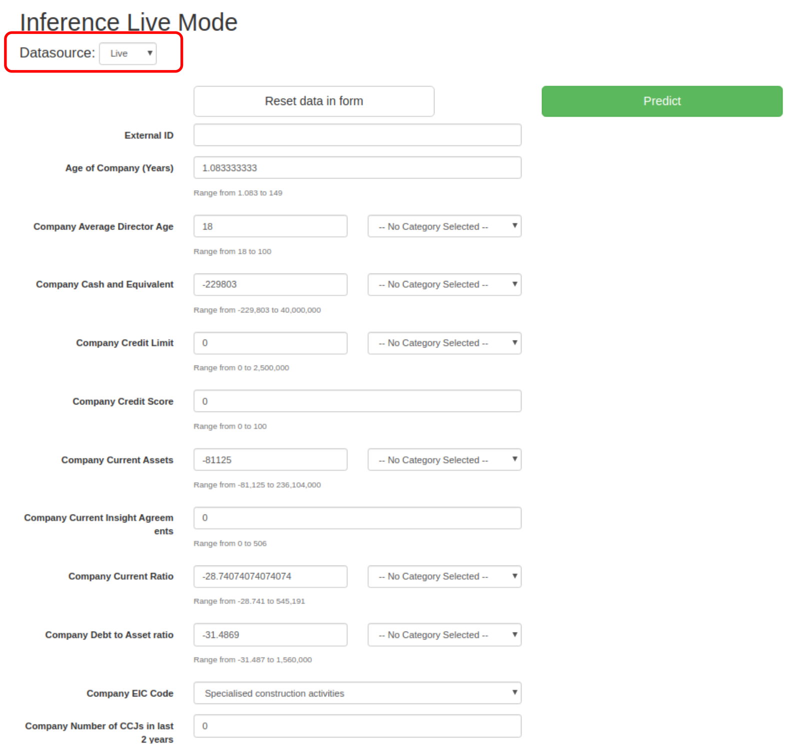

Figure 24 UI based Inference Page

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 Prediction 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.