Driverless AI Experiment: `vubirohi`

Generated on: Thu Oct 4 15:51:32 2018

Generated by: ang

Table of Contents

Experiment Overview

Driverless AI built 1 LightGBMModel to predict `default payment next month` given 23 original features from the input dataset `CreditCard_train.csv`. This classification experiment completed in 13 minutes and 46 seconds (0:13:46), using 5 of the 23 original features, and 107 of the 8,117 engineered features.

Performance

Dataset	AUC
Provided Validation Data	0.783
Test Data	0.762

Driverless Settings

Dial Settings	Description	Setting Value	Range of Possible Values
Accuracy	Controls sophistication of the model	7	1-10
Time	Controls duration of the experiment	3	1-10
Interpretability	Controls complexity of the features	6	1-10

System Specifications

System	System Memory	CPUs	GPUs
Docker/Linux	240	32	4

Versions

Driverless AI Version
1.4.0rc7

Data Overview

This section provides information on the datasets used for the experiment.

data	file path	number of rows	number of columns
training	/opt/h2oai/dai/tmp/uploads/CreditCard_train.csv	16,784	25
validation	/opt/h2oai/dai/tmp/uploads/CreditCard_valid.csv	2,387	25
testing	/opt/h2oai/dai/tmp/uploads/CreditCard_test.csv	4,828	25

Training Data

The training data consists of only numeric columns

The summary of the columns is shown below:

Numeric Columns

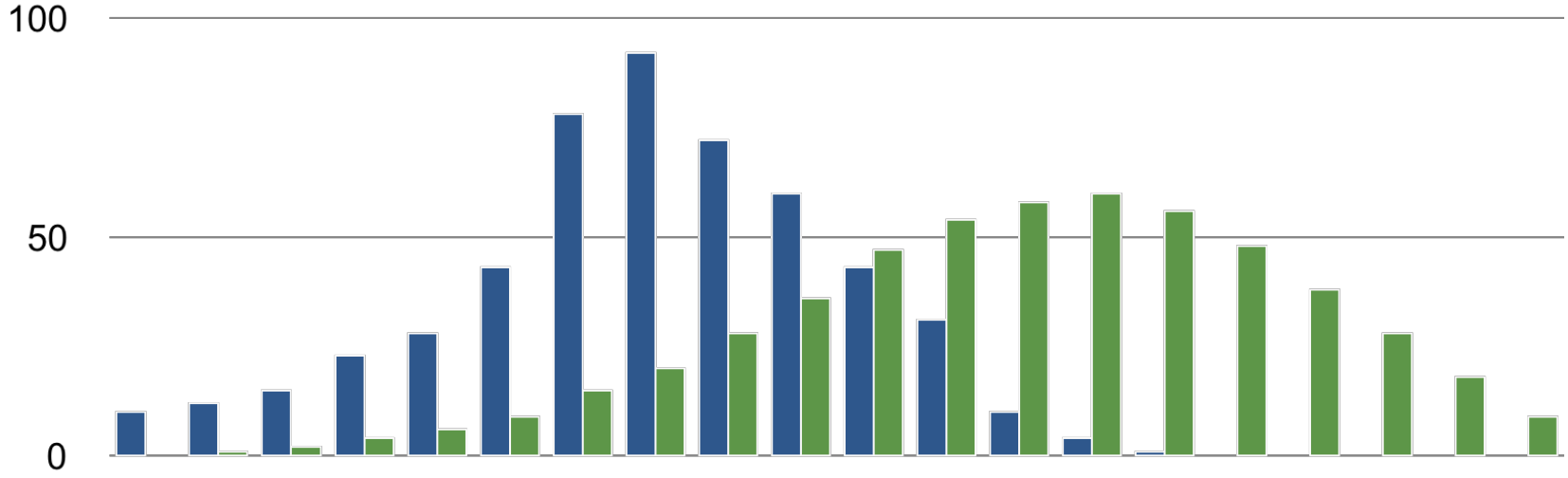
name	min	mean	max	std	unique	freq of mode
ID	1.000	11,984.490	23,999.000	6,921.205	16,784	1
LIMIT_BAL	10,000.000	165,222.693	1,000,000.000	129,386.695	76	1,947
SEX	1.000	1.628	2.000	0.483	2	10,532
EDUCATION	0.000	1.849	6.000	0.779	7	7,971
MARRIAGE	0.000	1.557	3.000	0.523	4	8,989
AGE	21.000	35.394	79.000	9.326	54	876
PAY_0	-2.000	0.011	8.000	1.123	11	8,241
PAY_2	-2.000	-0.117	7.000	1.204	10	8,781
PAY_3	-2.000	-0.151	8.000	1.201	11	8,845
PAY_4	-2.000	-0.204	7.000	1.161	10	9,363
PAY_5	-2.000	-0.249	7.000	1.131	9	9,530
PAY_6	-2.000	-0.273	8.000	1.153	10	9,079
BILL_AMT1	-165,580.000	50,774.611	964,511.000	72,593.456	13,586	1,149
BILL_AMT2	-69,777.000	48,771.125	983,931.000	70,375.208	13,352	1,459
BILL_AMT3	-157,264.000	46,451.671	693,131.000	67,360.391	13,183	1,632
BILL_AMT4	-170,000.000	42,339.368	891,586.000	63,053.499	12,890	1,775
BILL_AMT5	-81,334.000	40,068.033	927,171.000	60,465.380	12,609	1,989

name	min	mean	max	std	unique	freq of mode
BILL_AMT6	-339,603.000	38,626.532	961,664.000	59,479.473	12,359	2,282
PAY_AMT1	0.000	5,490.694	505,000.000	15,049.475	5,412	3,032
PAY_AMT2	0.000	5,765.445	580,464.000	17,860.984	5,388	3,054
PAY_AMT3	0.000	4,894.435	896,040.000	15,820.424	5,083	3,382
PAY_AMT4	0.000	4,714.254	497,000.000	14,505.758	4,775	3,634
PAY_AMT5	0.000	4,748.571	417,990.000	15,217.127	4,655	3,810
PAY_AMT6	0.000	5,231.688	528,666.000	18,097.503	4,685	4,099
default payment next month	False	0.223	True	0.416	2	3,740

Shifts Detected

Driverless AI can perform shift detection between the training, validation and testing datasets. It does this by training a binomial model to predict which dataset a record belongs to. For example, it may find that it is able to separate the training and testing data with an AUC of 0.8 using only the column: `C1` as the predictor. This indicates that there is some sort of drift in the distribution of `C1` between the training and testing data.

An example of a shift distribution between two datasets is shown below:



For this experiment, Driverless AI checked the train, validation, and test data for any shift in distributions but found none. This indicates that all the predictors/columns in the train, validation, and test data are from the same distribution.

Methodology

This section describes the experiment methodology.

Assumptions and Limitations

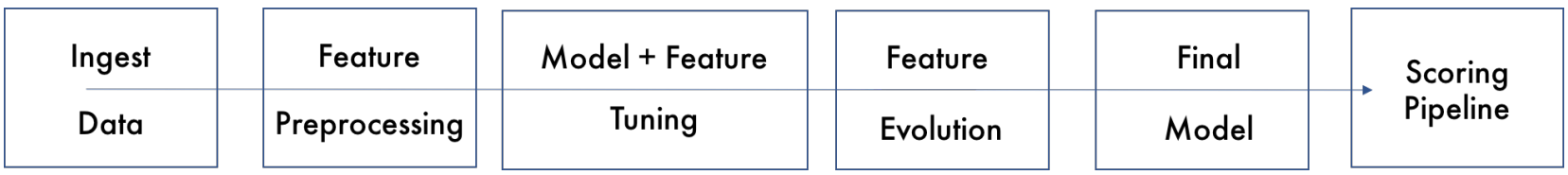
Driverless AI trains all models based on the training data provided (in this case: `CreditCard_train.csv`). It is the assumption of Driverless AI that this dataset is representative of the data that will be seen when scoring.

Driverless AI may perform shift detection between the train, validation, and test data. If a shift in distribution is detected, this may indicate that the data that will be used for scoring may have distributions not represented in the training data.

For this experiment, Driverless AI performed shift detection but found no significant changes in the distribution of the train, validation, and test data.

Experiment Pipeline

For this experiment, Driverless AI performed the following steps to find the optimal final model:



The steps in this pipeline are described in more detail below:

1. **Ingest Data**
    - detected column types
2. **Feature Preprocessing**
    - turned raw features into numeric
3. **Model and Feature Tuning**
    - found the optimal parameters for lightgbm models by training models with different parameters
    - the best parameters are those that generate the greatest **AUC** on the internal validation data
    - trained and scored **41** models to evaluate features and model parameters
4. **Feature Evolution**
    - found the best representation of the data for the final model training by creating and evaluating **8,117** features over **30** iteration s
    - trained and scored **492** models to further evaluate engineered features
5. **Final Model**
    - the final model is the best model from the feature engineering iterations
    - no stacked ensemble is done because a validation dataset was provided by the user
6. **Create Scoring Pipeline**
    - created and exported the Python scoring pipeline (no MOJO Scoring Pipeline created)
    - Python Scoring Pipeline: `h2oai_experiment_vubirohi/scoring_pipeline/scorer.zip`

Driverless AI trained models throughout the experiment in an effort to determine the best parameters, model dataset, and optimal final model. The stages are described below:

Driverless AI Step	Number of Models	Number of Folds/Validation Datasets
Parameter and Feature Tuning	41	1
Feature Evolution	492	1
Final Model	0	None

Experiment Settings

Below are the settings selected for the experiment by ang:

Defined Parameters

Parameter	Value
dataset_key	bimetito
target_col	default payment next month
weight_col	
fold_col	
orig_time_col	
time_col	[OFF]
is_classification	True
cols_to_drop	['ID']
validset_key	hilativi
testset_key	pihofawa
enable_gpus	True
seed	False
accuracy	7
time	3
interpretability	6
scorer	AUC
is_timeseries	False

Config Overrides

Parameter	Value
enable_xgboost	"auto"
enable_lightgbm	"auto"
enable_glm	"auto"
enable_tensorflow	"off"
enable_rulefit	"off"
check_distribution_shift	true
time_series_recipe	true
override_lag_sizes	""
prob_lag_non_targets	0.1
make_python_scoring_pipeline	true
make_mojo_scoring_pipeline	false
use_feature_brain	true
smart_imbalanced_sampling	false
holiday_features	true

Parameter	Value
seed	1234
nfeatures_max	-1
feature_engineering_effort	5
max_feature_interaction_depth	8
max_relative_cardinality	0.95
string_col_as_text_threshold	0.3
enable_tensorflow_force	false
tensorflow_max_epochs	100
tensorflow_max_epochs_nlp	2
max_nestimators	3000
max_learning_rate	0.5
max_cores	0
num_gpus_per_model	1
num_gpus_per_experiment	-1
gpu_id_start	0

These Accuracy, Time, and Interpretability settings map to the following internal configuration of the Driverless AI experiment:

Internal Parameter	Value
data filtered	False
number of feature engineering iterations	30
number of models trained per iteration	8
early stopping rounds	5
monotonicity constraint	False
number of model tuning model combinations	41
number of base learners in ensemble	0
time column	[OFF]

Details

- **data filtered**: Driverless AI may filter the training data depending on the number of rows and the Accuracy setting.
    - for this experiment, the training data was not filtered.
- **number of feature engineering iterations**: the number of iterations performed of feature engineering.
- **number of models evaluated per iteration**: for each feature engineering iteration, Driverless AI trains multiple models. Each model is trained with a different set of predictors or features. The goal of this step is to determine which types of features, lead to the greatest AUC.
- **early stopping rounds**: if Driverless AI does not see any improvement after 5 iterations of feature engineering, the feature engineering step is automatically stopped.

- **monotonicity constraint**: if enabled, the models will only have monotone relationships between the predictors and target variable.
- **number of model tuning combinations**: the number of model tuning combinations evaluated to determine the optimal model settings for the lightgbm models.
- **number of base learners in ensemble**: the number of base models used to create the final ensemble.
- **time column**: the column that provides time column. If a time column is provided, feature engineering and model validation will respect the causality of time. If the time column is turned off, no time order is used for modeling and data may be shuffled randomly (any potential temporal causality will be ignored).

Validation Strategy

Driverless AI used the validation data provided (`CreditCard_valid.csv`) to determine the performance of the model parameter tuning and feature engineering steps.

Model Tuning

The table below shows a portion of the different parameter configurations evaluated by Driverless AI for the lightgbm models and their score and training time. The table is ordered based on a combination of greatest score and lowest training time.

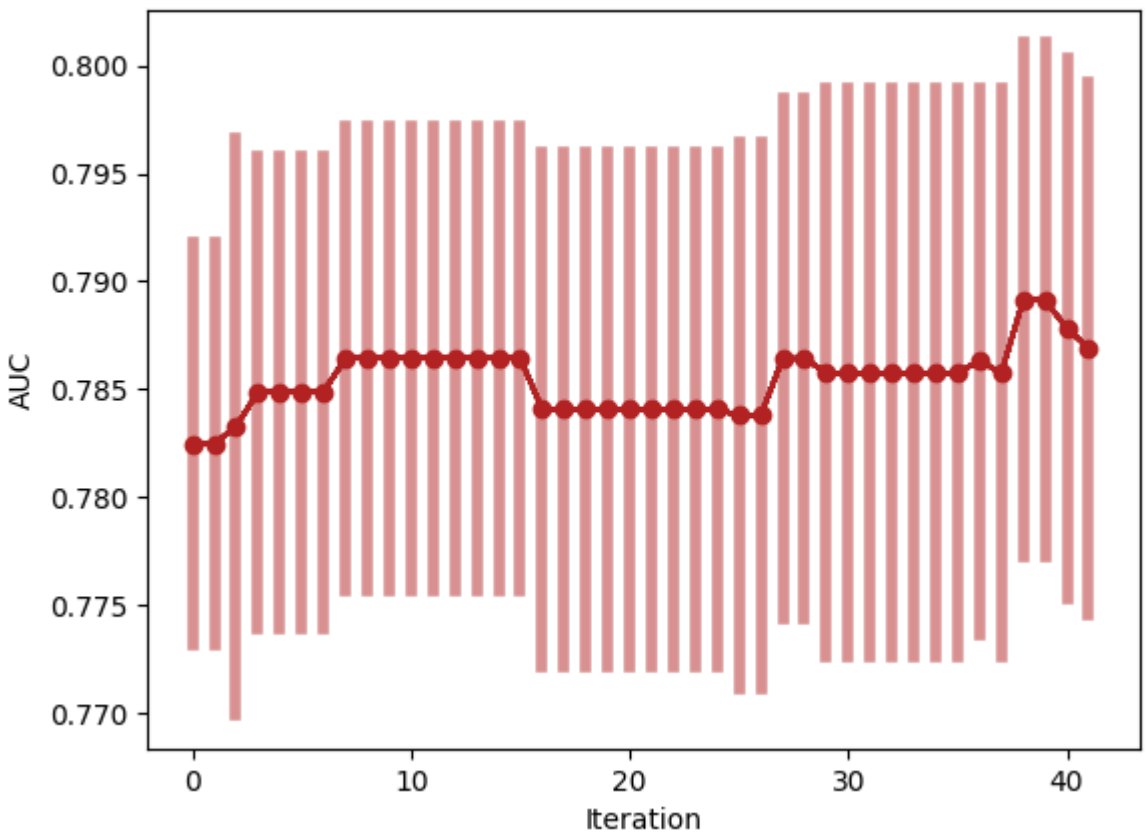
job order	booster	nfeatures	scores	training times
30	lightgbm	188	0.786	1.266
33	lightgbm	204	0.786	2.753
2	lightgbm	23	0.780	1.513
6	lightgbm	23	0.780	1.523
3	lightgbm	23	0.779	2.276
1	lightgbm	23	0.778	2.213
7	lightgbm	23	0.777	2.305
16	lightgbm	164	0.777	2.419
12	lightgbm	45	0.777	0.899
13	lightgbm	107	0.777	1.861
10	lightgbm	46	0.776	1.110
17	lightgbm	121	0.775	1.326
15	lightgbm	108	0.785	1.605
5	lightgbm	23	0.775	2.298
26	lightgbm	167	0.774	1.934
19	lightgbm	161	0.774	1.415
32	lightgbm	254	0.773	2.289
31	lightgbm	249	0.773	2.306
40	lightgbm	236	0.771	6.483
23	lightgbm	162	0.771	1.667
11	lightgbm	46	0.770	1.404

job order	booster	nfeatures	scores	training times
39	lightgbm	252	0.770	4.774
22	lightgbm	167	0.769	2.887
36	lightgbm	210	0.784	1.426
24	lightgbm	172	0.768	3.180
27	lightgbm	171	0.767	3.307
25	lightgbm	172	0.767	6.766
9	lightgbm	46	0.766	5.080
29	lightgbm	246	0.765	7.297
38	lightgbm	238	0.765	4.107
28	lightgbm	170	0.765	4.009
18	lightgbm	169	0.762	2.151
20	lightgbm	167	0.761	5.588
21	lightgbm	169	0.758	7.018
8	lightgbm	23	0.783	1.685
35	lightgbm	248	0.756	5.203
0	lightgbm	25	0.782	2.582
4	lightgbm	23	0.782	1.552
14	lightgbm	109	0.782	1.509
37	lightgbm	250	0.780	4.006
34	lightgbm	238	0.780	4.649

Feature Evolution

During the Model and Feature Tuning Stage, Driverless AI evaluates the effects of different types of algorithms, algorithm parameters, and features. The goal of the Model and Feature Tuning Stage is to determine the best algorithm and algorithm parameters to use during the Feature Evolution Stage. The Feature Evolution Stage trained 492 lightgbm models where each model evaluated a different set of features. The Feature Evolution Stage uses a genetic algorithm to search the large feature engineering space.

The graph belows shows the effect the Model and Feature Tuning Stage and Feature Evolution Stage had on the performance.



Feature Transformation

The result of the Feature Evolution Stage is the final set of features to use for the model. Some of these features were automatically created by Driverless AI. The top 14 features used in the final model are shown belowed ordered by importance. If no transformer was applied, the feature is an original column.

Feature	Description	Transformer	Relative Importance
284_ClusterTE: ClusterID42: BILL_AMT6: PAY_0: PAY_3: PAY_5: PAY_AMT2.0	Out-of-fold mean of the response grouped by: ['ClusterID42:BILL_AMT6:PAY_0:PAY_3:PAY_5:PAY_AMT2'] using 5 folds [internal parameters:(20, 10, 20)] (Clustered into 42 clusters) [internal parameters:(42, True, 20, 10, 20)]	Cluster Target Encoding	1.000
136_ClusterTE: ClusterID91: BILL_AMT6: PAY_0: PAY_3: PAY_4.0	Out-of-fold mean of the response grouped by: ['ClusterID91:BILL_AMT6:PAY_0:PAY_3:PAY_4'] using 5 folds [internal parameters:(10, 3, 20)] (Clustered into 91 clusters) [internal parameters:(91, False, 10, 3, 20)]	Cluster Target Encoding	0.823
139_ClusterTE: ClusterID99: PAY_0: PAY_5.0	Out-of-fold mean of the response grouped by: ['ClusterID99:PAY_0:PAY_5'] using 5 folds [internal parameters:(10, 5, 20)] (Clustered into 99 clusters) [internal parameters:(99, True, 10, 5, 20)]	Cluster Target Encoding	0.708
276_NumToCatTE: PAY_0: PAY_2: PAY_5.0	Out-of-fold mean of the response grouped by: ['PAY_0', 'PAY_2', 'PAY_5'] using 5 folds [internal parameters:(10, 5, 100)] (numeric columns are bucketed into 10 equally populated bins) [internal parameters:(10, 5, 100)]	Numeric to Categorical Target Encoding	0.657

Feature	Description	Transformer	Relative Importance
83_NumToCatWoE: PAY_0: PAY_2.0	Weight of Evidence for columns ['PAY_0', 'PAY_2'] column #0 (numeric columns are bucketed into 250 equally populated bins)	Numeric to Categorical Weight of Evidence	0.371
52_NumToCatWoE: PAY_0: PAY_2.0	Weight of Evidence for columns ['PAY_0', 'PAY_2'] column #0 (numeric columns are bucketed into 100 equally populated bins)	Numeric to Categorical Weight of Evidence	0.280
200_ClusterTE: ClusterID65: PAY_0: PAY_3: PAY_4: PAY_AMT6.0	Out-of-fold mean of the response grouped by: ['ClusterID65:PAY_0:PAY_3:PAY_4:PAY_AMT6'] using 5 folds [internal parameters:(10, 3, None)] (Clustered into 65 clusters) [internal parameters: (65, True, 10, 3, None)]	Cluster Target Encoding	0.257
157_NumToCatTE: PAY_0: PAY_4: PAY_AMT1.0	Out-of-fold mean of the response grouped by: ['PAY_0', 'PAY_4', 'PAY_AMT1'] using 5 folds [internal parameters:(10, 5, 10)] (numeric columns are bucketed into 25 equally populated bins) [internal parameters:(10, 5, 10)]	Numeric to Categorical Target Encoding	0.189
174_ClusterTE: ClusterID27: LIMIT_BAL: PAY_0: PAY_5.0	Out-of-fold mean of the response grouped by: ['ClusterID27:LIMIT_BAL:PAY_0:PAY_5'] using 5 folds [internal parameters: (100, 3, 100)] (Clustered into 27 clusters) [internal parameters:(27, False, 100, 3, 100)]	Cluster Target Encoding	0.187
102_NumToCatWoE: BILL_AMT1: LIMIT_BAL.0	Weight of Evidence for columns ['BILL_AMT1', 'LIMIT_BAL'] column #0 (numeric columns are bucketed into 10 equally populated bins)	Numeric to Categorical Weight of Evidence	0.180
228_ClusterTE: ClusterID35: LIMIT_BAL: PAY_0: PAY_2: PAY_3: PAY_AMT3.0	Out-of-fold mean of the response grouped by: ['ClusterID35:LIMIT_BAL:PAY_0:PAY_2:PAY_3:PAY_AMT3'] using 5 folds [internal parameters:(100, 5, None)] (Clustered into 35 clusters) [internal parameters:(35, False, 100, 5, None)]	Cluster Target Encoding	0.161
283_InteractionMul: LIMIT_BAL: PAY_AMT2	[LIMIT_BAL] * [PAY_AMT2]	Interaction	0.155
178_InteractionAdd: PAY_0: PAY_6	[PAY_0] + [PAY_6]	Interaction	0.150
268_ClusterTE: ClusterID59: BILL_AMT3: BILL_AMT5: BILL_AMT6: PAY_0: PAY_2: PAY_AMT1.0	Out-of-fold mean of the response grouped by: ['ClusterID59:BILL_AMT3:BILL_AMT5:BILL_AMT6:PAY_0:PAY_2:PAY_AMT1'] using 5 folds [internal parameters:(100, 5, 20)] (Clustered into 59 clusters) [internal parameters:(59, False, 100, 5, 20)]	Cluster Target Encoding	0.141

Final Model

Pipeline

Final LightGBMModel pipeline with ensemble_level=0 transforming 23 original features -> 112 features in each of 1 models each fit on external validation set.

Final Model Scores

Scorer	Final ensemble external validation scores +/- standard deviation	Final test scores +/- standard deviation	Optimized	Better score is
GINI	0.56687 +/- 0.023738	0.52466 +/- 0.016573		higher
MCC	0.43559 +/- 0.020192	0.40379 +/- 0.014159		higher
F05	0.60248 +/- 0.019685	0.56838 +/- 0.012599		higher
F1	0.54956 +/- 0.017897	0.53436 +/- 0.010335		higher
F2	0.63717 +/- 0.011447	0.63274 +/- 0.0078875		higher
ACCURACY	0.8274 +/- 0.0069684	0.81048 +/- 0.0055773		higher
LOGLOSS	0.42525 +/- 0.012894	0.45046 +/- 0.009114		lower
AUCPR	0.56373 +/- 0.021806	0.53332 +/- 0.013215		higher
AUC	0.78344 +/- 0.011869	0.76233 +/- 0.0082867	*	higher

Deployment

For this experiment, the Python Scoring Pipeline is available for productionizing the final model pipeline for a given row of data or table of data. The MOJO Scoring Pipeline can be built by clicking the **BUILD MOJO SCORING PIPELINE** button if available.

Python Scoring Pipeline

This package contains an exported model and Python 3.6 source code examples for productionizing models built using H2O Driverless AI. The Python Scoring Pipeline is located here:

- `h2oai_experiment_vubirohi/scoring_pipeline/scorer.zip`

The files in this package allow you to transform and score on new data in a couple of different ways:

- From Python 3.6, you can import a scoring module, and then use the module to transform and score on new data.
- From other languages and platforms, you can use the TCP/HTTP scoring service bundled with this package to call into the scoring pipeline module through remote procedure calls (RPC).