**Driverless AI Experiment: *bilefoko***

Generated by: h2oai

Generated on: Mon Dec 17 08:46:49 2018

## Table of Contents

[Table of Contents 1](#_Toc533076764)

[Experiment Overview 1](#_Toc533076765)

[Data Overview 2](#_Toc533076766)

[Methodology 4](#_Toc533076767)

[Validation Strategy 10](#_Toc533076768)

[Model Tuning 10](#_Toc533076769)

[Feature Evolution 12](#_Toc533076770)

[Feature Transformations 14](#_Toc533076771)

[Final Model 15](#_Toc533076772)

[Alternative Models 19](#_Toc533076773)

[Deployment 20](#_Toc533076774)

[Appendix 20](#_Toc533076775)

## Experiment Overview

Driverless AI built a stacked ensemble of 1 LightGBMModel, 1 XGBoostModel to predict *default payment next month* given 23 original features from the input dataset *CreditCard-train.csv*. This classification experiment completed in 19 minutes and 33 seconds (0:19:33), using 6 of the 23 original features, and 476 of the 2,278 engineered features.

### Performance

|  |  |
| --- | --- |
| **Dataset** | **AUC** |
| Internal Validation  | 0.783  |
| Test Data  | 0.802  |

### Driverless Settings

|  |  |  |  |
| --- | --- | --- | --- |
| Dial Settings | Description | Setting Value | Range of Possible Values |
| Accuracy | Controls sophistication of the model | 6  | 1-10 |
| Time  | Controls duration of the experiment | 4  | 1-10 |
| Interpretability | Controls complexity of the model | 1  | 1-10 |

### System Specifications

|  |  |  |  |
| --- | --- | --- | --- |
| **System** | **System Memory** | **CPUs** | **GPUs** |
| Linux  | 125  | 40  | 1  |

### Versions

|  |
| --- |
| Driverless AI Version |
| 1.5.0 |

## Data Overview

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

|  |  |  |  |
| --- | --- | --- | --- |
| **data** | **file path** | **number of rows** | **number of columns** |
| training  | /data/Kaggle/CreditCard/CreditCard-train.csv | 23,999  | 25  |
| validation  | Not provided  | None  | None  |
| testing  | /data/Kaggle/CreditCard/CreditCard-train.csv | 6,000  | 25  |

### Training Data

The training data consists of only numeric columns

The summary of the columns is shown below:

#### Numeric Columns

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **name** | **data\_type** | **min** | **mean** | **max** | **std** | **unique** | **freq of mode** |
| ID  | int  | 1.000  | 12,000.000  | 23,999.000  | 6,928.059  | 23,999  | 1  |
| LIMIT\_BAL  | int  | 10,000.000  | 165,498.716  | 1,000,000.000  | 129,130.743  | 79  | 2,740  |
| SEX  | int  | 1.000  | 1.628  | 2.000  | 0.483  | 2  | 15,078  |
| EDUCATION  | int  | 0.000  | 1.847  | 6.000  | 0.780  | 7  | 11,360  |
| MARRIAGE  | int  | 0.000  | 1.557  | 3.000  | 0.522  | 4  | 12,876  |
| AGE  | int  | 21.000  | 35.381  | 79.000  | 9.271  | 55  | 1,284  |
| PAY\_0  | int  | -2.000  | -0.002  | 8.000  | 1.127  | 11  | 11,732  |
| PAY\_2  | int  | -2.000  | -0.123  | 8.000  | 1.201  | 11  | 12,543  |
| PAY\_3  | int  | -2.000  | -0.155  | 8.000  | 1.204  | 11  | 12,576  |
| PAY\_4  | int  | -2.000  | -0.212  | 8.000  | 1.167  | 11  | 13,250  |
| PAY\_5  | int  | -2.000  | -0.253  | 8.000  | 1.137  | 10  | 13,520  |
| PAY\_6  | int  | -2.000  | -0.278  | 8.000  | 1.158  | 10  | 12,876  |
| BILL\_AMT1  | int  | -165,580.000  | 50,598.929  | 964,511.000  | 72,650.198  | 18,717  | 1,607  |
| BILL\_AMT2  | int  | -69,777.000  | 48,648.047  | 983,931.000  | 70,365.396  | 18,367  | 2,049  |
| BILL\_AMT3  | int  | -157,264.000  | 46,368.904  | 1,664,089.000  | 68,194.720  | 18,131  | 2,325  |
| BILL\_AMT4  | int  | -170,000.000  | 42,369.873  | 891,586.000  | 63,071.455  | 17,719  | 2,547  |
| BILL\_AMT5  | int  | -81,334.000  | 40,002.333  | 927,171.000  | 60,345.728  | 17,284  | 2,840  |
| BILL\_AMT6  | int  | -339,603.000  | 38,565.267  | 961,664.000  | 59,156.501  | 16,906  | 3,258  |
| PAY\_AMT1  | int  | 0.000  | 5,543.098  | 505,000.000  | 15,068.863  | 6,918  | 4,270  |
| PAY\_AMT2  | int  | 0.000  | 5,815.529  | 1,684,259.000  | 20,797.444  | 6,839  | 4,362  |
| PAY\_AMT3  | int  | 0.000  | 4,969.431  | 896,040.000  | 16,095.929  | 6,424  | 4,853  |
| PAY\_AMT4  | int  | 0.000  | 4,743.657  | 497,000.000  | 14,883.555  | 6,028  | 5,200  |
| PAY\_AMT5  | int  | 0.000  | 4,783.644  | 417,990.000  | 15,270.704  | 5,984  | 5,407  |
| PAY\_AMT6  | int  | 0.000  | 5,189.574  | 528,666.000  | 17,630.719  | 5,988  | 5,846  |

#### Boolean Columns

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **name** | **data\_type** | **min** | **mean** | **max** | **std** | **freq of max value** |
| default payment next month  | bool  | False  | 0.2237  | True  | 0.4167  | 5,369  |

### 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.

 For this experiment, Driverless AI checked the train and test data for any shift in distributions but found none. This indicates that all the predictors/columns in the train 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 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 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 light gbm and xgboost models by training models with different parameters
	* the best parameters are those that generate the greatest **AUC** on the internal validation data
	* 25 trained and scored 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 **2,278** features over **60** iterations
	* 140 trained and scored to further evaluate engineered features
5. **Final Model**
	* the final model is a stacked ensemble of **1 LightGBMModel, 1 XGBoostModel**
	* the features of these models are the best features found during the feature engineering iterations
6. **Create Scoring Pipeline**
	* created and exported the Python scoring pipeline (no MOJO Scoring Pipeline automatically created)
	* Python Scoring Pipeline: h2oai\_experiment\_bilefoko/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 Stage | Timing (seconds) | Number of Models |
| Data Preparation | 3.03 | 0 |
| Model and Feature Tuning  | 147.51 | 25 |
| Feature Evolution | 514.61 | 140 |
| Final Pipeline Training  | 493.95 | 10 |

### Experiment Settings

Below are the settings selected for the experiment by h2oai:

**Defined Parameters**

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| dataset\_key  | gobicudo  |
| resumed\_model\_key  |   |
| target\_col  | default payment next month  |
| weight\_col  |   |
| fold\_col  |   |
| orig\_time\_col  |   |
| time\_col  | [OFF]  |
| is\_classification  | True  |
| cols\_to\_drop  | []  |
| validset\_key  |   |
| testset\_key  | sikuvara  |
| enable\_gpus  | True  |
| seed  | False  |
| accuracy  | 6  |
| time  | 4  |
| interpretability  | 1  |
| scorer  | AUC  |
| is\_timeseries  | False  |

 **Config Overrides**

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| enable\_xgboost  | "auto"  |
| enable\_lightgbm  | "auto"  |
| enable\_rf  | "auto"  |
| enable\_glm  | "auto"  |
| enable\_tensorflow  | "off"  |
| enable\_rulefit  | "off"  |
| enable\_ftrl  | "off"  |
| check\_distribution\_shift  | true  |
| drop\_features\_distribution\_shift\_threshold\_auc  | 0.6  |
| enable\_target\_encoding  | true  |
| time\_series\_recipe  | true  |
| override\_lag\_sizes  | ""  |
| prob\_lag\_non\_targets  | 0.1  |
| make\_python\_scoring\_pipeline  | true  |
| make\_mojo\_scoring\_pipeline  | false  |
| rulefit\_max\_num\_rules  | -1  |
| feature\_brain\_level  | 2  |
| smart\_imbalanced\_sampling  | false  |
| holiday\_features  | true  |
| seed  | 1234  |
| force\_64bit\_precision  | false  |
| max\_orig\_cols\_selected  | 10000  |
| nfeatures\_max  | -1  |
| max\_rows\_feature\_evolution  | 1000000  |
| 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  |
| enable\_tensorflow\_nlp  | true  |
| tensorflow\_max\_epochs\_nlp  | 2  |
| min\_dai\_iterations  | 0  |
| 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  |
| compute\_correlation  | false  |
| high\_correlation\_value\_to\_report  | 0.95  |

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  | 40  |
| number of models trained per iteration  | 4  |
| early stopping rounds  | 5  |
| monotonicity constraint  | False  |
| number of model tuning model combinations  | 24  |
| number of base learners in ensemble  | 2  |
| 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 light gbm and xgboost 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 automatically split the training data to determine the performance of the model parameter tuning and feature engineering steps. For the experiment, Driverless AI randomly split the data into **2/3 training** and **1/3 validation**.

## Model Tuning

The table below shows the score and training time of the light gbm and xgboost models evaluated by Driverless AI. The table shows the top 10 parameter tuning models evaluated, ordered based on a combination of greatest score and lowest training time.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **job order** | **booster** | **nfeatures** | **scores** | **training times** |
| 19  | lightgbm  | 242  | 0.7809  | 1.8618  |
| 20  | gbtree  | 268  | 0.7798  | 1.5719  |
| 10  | gbtree  | 76  | 0.7759  | 1.0142  |
| 16  | gbtree  | 205  | 0.7759  | 1.519  |
| 23  | gbtree  | 326  | 0.7735  | 2.6525  |
| 18  | gbtree  | 215  | 0.7733  | 1.7945  |
| 22  | gbtree  | 313  | 0.7722  | 2.6572  |
| 9  | lightgbm  | 80  | 0.7695  | 1.818  |
| 4  | gbtree  | 23  | 0.7694  | 1.9216  |
| 5  | gbtree  | 23  | 0.7694  | 1.8773  |

More detailed information on the parameters evaluated for each algorithm is shown below.

### lightgbm tuning

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **tree method** | **grow policy** | **max depth** | **max leaves** | **colsample bytree** | **subsample** | **nfeatures** | **scores** | **training times** |
| gpu\_hist  | lossguide  | 0.0  | 8.0  | 0.2  | 1.0  | 242  | 0.7809  | 1.8618  |
| gpu\_hist  | depthwise  | 10.0  | 0.0  | 0.3  | 1.0  | 80  | 0.7695  | 1.818  |
| gpu\_hist  | depthwise  | 7.0  | 0.0  | 0.35  | 0.4  | 70  | 0.7792  | 1.1034  |
| gpu\_hist  | lossguide  | 0.0  | 64.0  | 0.65  | 0.4  | 155  | 0.7684  | 1.7384  |
| gpu\_hist  | lossguide  | 0.0  | 512.0  | 0.35  | 0.4  | 237  | 0.7579  | 3.1317  |
| gpu\_hist  | lossguide  | 0.0  | 16.0  | 0.6  | 0.8  | 120  | 0.7775  | 1.114  |

### gbtree tuning

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **tree method** | **grow policy** | **max depth** | **max leaves** | **colsample bytree** | **subsample** | **nfeatures** | **scores** | **training times** |
| gpu\_hist  | depthwise  | 5.0  | 0.0  | 0.7  | 0.9  | 268  | 0.7798  | 1.5719  |
| gpu\_hist  | lossguide  | 0.0  | 16.0  | 0.55  | 0.6  | 76  | 0.7759  | 1.0142  |
| gpu\_hist  | lossguide  | 0.0  | 32.0  | 0.6  | 1.0  | 205  | 0.7759  | 1.519  |
| gpu\_hist  | depthwise  | 7.0  | 0.0  | 0.65  | 1.0  | 326  | 0.7735  | 2.6525  |
| gpu\_hist  | lossguide  | 0.0  | 64.0  | 0.55  | 0.9  | 215  | 0.7733  | 1.7945  |
| gpu\_hist  | lossguide  | 0.0  | 128.0  | 0.6  | 0.9  | 313  | 0.7722  | 2.6572  |
| gpu\_hist  | depthwise  | 10.0  | 0.0  | 0.65  | 0.5  | 23  | 0.7694  | 1.9216  |
| gpu\_hist  | depthwise  | 10.0  | 0.0  | 0.65  | 0.5  | 23  | 0.7694  | 1.8773  |
| gpu\_hist  | depthwise  | 10.0  | 0.0  | 0.65  | 0.5  | 23  | 0.7694  | 1.9014  |
| gpu\_hist  | depthwise  | 10.0  | 0.0  | 0.65  | 0.5  | 23  | 0.7694  | 1.8986  |

## 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 parameters to use during the Feature Evolution Stage. In the Feature Evolution Stage, Driverless AI trained light gbm and xgboost models (140) 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.



Based on the experiment settings and column types in the dataset, Driverless AI was able to explore the following transformers:

* **CVTargetEncodeDT**: calculates the mean of the response column for each value in a categorical column and uses this as a new feature. Cross Validation is used to calculate mean response to prevent overfitting.
* **FrequentTransformer**: calculates the frequency for each value in categorical column(s) and uses this as a new feature.
* **WeightOfEvidenceTransformer**: calculates Weight of Evidence for each value in categorical column(s). The Weight of Evidence is used as a new feature. Weight of Evidence measures the “strength” of a grouping for separating good and bad risk and is calculated by taking the log of the ratio of distributions for a binary response column.
* **OHETransformer**: converts a categorical column to a series of boolean features by performing one-hot encoding. The boolean features are used as new features.
* **BulkInteractionsTransformer**: add, divide, multiply, and subtract two numeric columns in the data to create a new feature.
* **ClusterDistTransformer**: clusters selected numeric columns and uses the distance to a specific cluster as a new feature.
* **ClusterTETransformer**: clusters selected numeric columns and calculates the mean of the response column for each cluster. The mean of the response is used as a new feature. Cross Validation is used to calculate mean response to prevent overfitting.
* **NumToCatTETransformer**: converts a numeric columns to categoricals by binning and then calculates the mean of the response column for each group. The mean of the response for the bin is used as a new feature. Cross Validation is used to calculate mean response to prevent overfitting.
* **NumToCatWoETransformer**: converts a numeric column to categorical by binning and then calculates Weight of Evidence for each bin. The Weight of Evidence is used as a new feature. Weight of Evidence measures the “strength” of a grouping for separating good and bad risk and is calculated by taking the log of the ratio of distributions for a binary response column.
* **TruncSVDNumTransformer**: trains a Truncated SVD model on selected numeric columns and uses the components of the truncated SVD matrix as new features.
* **CVCatNumEncodeF**: calculates an aggregation of a numeric column for each value in a categorical column (ex: calculate the mean Temperature for each City) and uses this aggregation as a new feature.
* **NumCatTETransformer**: calculates the mean of the response column for several selected columns. If one of the selected columns is numeric, it is first converted to categorical by binning. The mean of the response column is used as a new feature. Cross Validation is used to calculate mean response to prevent overfitting.

## Feature Transformations

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | **Description** | **Transformer** | **Relative Importance** |
| 115\_ClusterDist8: LIMIT\_BAL: PAY\_0: PAY\_4: SEX.7  | Distances to cluster center after segmenting columns ['LIMIT\_BAL', 'PAY\_0', 'PAY\_4', 'SEX'] into 8 clusters. Distance to cluster #7 [internal parameters:False]  | Cluster Distance  | 1.0  |
| 141\_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  | 1.0  |
| 115\_ClusterDist8: LIMIT\_BAL: PAY\_0: PAY\_4: SEX.5  | Distances to cluster center after segmenting columns ['LIMIT\_BAL', 'PAY\_0', 'PAY\_4', 'SEX'] into 8 clusters. Distance to cluster #5 [internal parameters:False]  | Cluster Distance  | 1.0  |
| 23\_InteractionAdd: PAY\_0: PAY\_2  | [PAY\_0] + [PAY\_2]  | Interaction  | 1.0  |
| 106\_ClusterDist7: BILL\_AMT2: BILL\_AMT6: PAY\_AMT1: SEX.6  | Distances to cluster center after segmenting columns ['BILL\_AMT2', 'BILL\_AMT6', 'PAY\_AMT1', 'SEX'] into 7 clusters. Distance to cluster #6 [internal parameters:False]  | Cluster Distance  | 1.0  |
| 115\_ClusterDist8: LIMIT\_BAL: PAY\_0: PAY\_4: SEX.6  | Distances to cluster center after segmenting columns ['LIMIT\_BAL', 'PAY\_0', 'PAY\_4', 'SEX'] into 8 clusters. Distance to cluster #6 [internal parameters:False]  | Cluster Distance  | 1.0  |
| 123\_InteractionAdd: PAY\_0: PAY\_2  | N/A  | Interaction  | 0.849  |
| 118\_ClusterTE: ClusterID17: PAY\_0: PAY\_2: PAY\_4: PAY\_AMT4.0  | Out-of-fold mean of the response grouped by: ['ClusterID17:PAY\_0:PAY\_2:PAY\_4:PAY\_AMT4'] using 5 folds [internal parameters:(10, 3, 100)] (Clustered into 17 clusters) [internal parameters:(17, True, 10, 3, 100)]  | Cluster Target Encoding  | 0.8353  |
| 228\_ClusterDist5: PAY\_0: PAY\_2: PAY\_3: PAY\_5.3  | Distances to cluster center after segmenting columns ['PAY\_0', 'PAY\_2', 'PAY\_3', 'PAY\_5'] into 5 clusters. Distance to cluster #3 [internal parameters:False]  | Cluster Distance  | 0.7977  |
| 213\_ClusterDist10: PAY\_3: PAY\_AMT3.8  | Distances to cluster center after segmenting columns ['PAY\_3', 'PAY\_AMT3'] into 9 clusters. Distance to cluster #8 [internal parameters:False]  | Cluster Distance  | 0.7509  |
| 27\_TruncSVD: PAY\_0: PAY\_2.0  | Component #1 of truncated SVD of ['PAY\_0', 'PAY\_2'] into 1 components  | Truncated SVD  | 0.6621  |
| 114\_ClusterDist8: BILL\_AMT1: PAY\_0: PAY\_2: PAY\_3.7  | Distances to cluster center after segmenting columns ['BILL\_AMT1', 'PAY\_0', 'PAY\_2', 'PAY\_3'] into 8 clusters. Distance to cluster #7 [internal parameters:False]  | Cluster Distance  | 0.6571  |
| 109\_NumToCatWoE: PAY\_0: PAY\_4.0  | Weight of Evidence for columns ['PAY\_0', 'PAY\_4'] column #0 (numeric columns are bucketed into 25 equally populated bins)  | Numeric to Categorical Weight of Evidence  | 0.5352  |
| 232\_ClusterDist5: PAY\_0: PAY\_2.3  | Distances to cluster center after segmenting columns ['PAY\_0', 'PAY\_2'] into 5 clusters. Distance to cluster #3 [internal parameters:False]  | Cluster Distance  | 0.4805  |

## Final Model

**Pipeline**

Final StackedEnsemble pipeline with ensemble\_level=2 transforming 23 original features -> 482 features in each of 10 models each fit on 5 internal holdout splits then linearly blended

**Details**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model Index** | **Type** | **Model Weight** | **Num Folds** | **Fitted features** | **Target Transformer** |
| 0  | LightGBMModel  | 0.6667  | 5  | 288  | str  |
| 1  | XGBoostModel  | 0.3333  | 5  | 194  | str  |

* Model Index: 0 has a weight of 0.667 in the final ensemble

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Type** | **max leaves** | **index** | **max depth** | **learning rate** | **colsample bytree** | **subsample** | **grow policy** | **tree method** |
| LightGBMModel  | 16  | 0  | 0  | 0.04  | 0.2  | 1.0  | lossguide  | gpu\_hist  |

* Model Index: 1 has a weight of 0.333 in the final ensemble

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Type** | **max leaves** | **index** | **max depth** | **learning rate** | **colsample bytree** | **subsample** | **grow policy** | **tree method** |
| XGBoostModel  | 16  | 1  | 0  | 0.04  | 0.7  | 0.9  | lossguide  | gpu\_hist  |

For a complete list of the parameters of the final model, see the Appendix.

**Performance of Final Model**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Scorer** | **Optimized** | **Better score is** | **Final ensemble with\_validation scores** | **Final ensemble with\_validation standard deviation** | **Final test scores** | **Final test standard deviation** |
| ACCURACY  |   | higher  | 0.8215  | 0.0057672  | 0.83433  | 0.0057672  |
| AUC  | X  | higher  | 0.78307  | 0.0087279  | 0.80196  | 0.0087279  |
| AUCPR  |   | higher  | 0.561  | 0.017109  | 0.57816  | 0.017109  |
| F05  |   | higher  | 0.58987  | 0.015364  | 0.59902  | 0.015364  |
| F1  |   | higher  | 0.55614  | 0.012855  | 0.55964  | 0.012855  |
| F2  |   | higher  | 0.64871  | 0.0099656  | 0.64909  | 0.0099656  |
| GINI  |   | higher  | 0.56614  | 0.017456  | 0.60391  | 0.017456  |
| LOGLOSS  |   | lower  | 0.43004  | 0.0087455  | 0.40485  | 0.0087455  |
| MCC  |   | higher  | 0.43177  | 0.015607  | 0.4441  | 0.015607  |

 **Validation Confusion Matrix**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Predicted: 0** | **Predicted: 1** | **error** |
| Actual: 0  | 15,906  | 2,724  | 15%  |
| Actual: 1  | 2,317  | 3,052  | 43%  |

 **Test Confusion Matrix**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Predicted: 0** | **Predicted: 1** | **error** |
| Actual: 0  | 4,138  | 596  | 13%  |
| Actual: 1  | 551  | 715  | 44%  |

*Receiving Operator Curve*



*Precision Recall Curve*



*Cumulative Lift*



*Cumulative Gains*



*Kolmogorov–Smirnov*



## Alternative Models

During the experiment, Driverless AI trained 165 alternative models. The following algorithms were evaluated during the Driverless AI experiment:

|  |  |  |  |
| --- | --- | --- | --- |
| **algorithm** | **package** | **version** | **documentation** |
| lightgbm  | lightgbm  | 2.2.3  | LightGBM, Light Gradient Boosting Machine. Contributors: https://github.com/Microsoft/LightGBM/graphs/contributors.  |
| gbtree  | xgboost  | 0.81  | XGBoost: eXtreme Gradient Boosting library. Contributors: https://github.com/dmlc/xgboost/blob/master/CONTRIBUTORS.md  |

Driverless AI is able to evaluate the algorithms: XGBoost GBM, XGBoost GLM, LightGBM, RuleFit, and Tensorflow models. The table below explains why certain algorithms were not selected for the final model, if any.

|  |  |
| --- | --- |
| **algorithm** | **selection** |
| gblinear  | algorithm not evaluated due to experiment configuration  |
| rulefit  | algorithm not evaluated due to experiment configuration  |
| tensorflow  | algorithm not evaluated due to experiment configuration  |
| gbtree  | selected for final model  |
| lightgbm  | selected for final model  |

## 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\_bilefoko/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).

## Appendix

### Final Model Details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model Index** | **Type** | **Model Weight** | **Num Folds** | **Fitted features** | **Target Transformer** |
| 0  | LightGBMModel  | 0.6667  | 5  | 288  | str  |
| 1  | XGBoostModel  | 0.3333  | 5  | 194  | str  |

**Model Index: 0**

|  |  |
| --- | --- |
| **parameter** | **value** |
| num\_classes  | 2  |
| min\_child\_weight  | 1  |
| debug\_verbose  | 0  |
| gpu\_id  | 0  |
| skip\_drop  | 0.4  |
| colsample\_bytree  | 0.2  |
| min\_data\_in\_bin  | 5  |
| objective  | binary:logistic  |
| reg\_lambda  | 5.0  |
| booster  | lightgbm  |
| max\_drop  | 50  |
| monotonicity\_constraints  | False  |
| max\_leaves  | 16  |
| min\_child\_samples  | 1  |
| max\_delta\_step  | 0.0  |
| silent  | True  |
| n\_estimators  | 1800  |
| n\_gpus  | 1  |
| boosting\_type  | gbdt  |
| model\_id  | 0  |
| random\_state  | 212233661  |
| scale\_pos\_weight  | 1.0  |
| subsample\_freq  | 1  |
| early\_stopping\_rounds  | 100  |
| reg\_alpha  | 0.0  |
| early\_stopping\_threshold  | 0  |
| grow\_policy  | lossguide  |
| subsample  | 1.0  |
| tree\_method  | gpu\_hist  |
| learning\_rate  | 0.04  |
| rate\_drop  | 0.1  |
| gamma  | 0.0  |
| max\_depth  | 0  |
| max\_bin  | 32  |
| eval\_metric  | logloss  |
| n\_jobs  | 4  |
| nfolds  | 5  |

**Model Index: 1**

|  |  |
| --- | --- |
| **parameter** | **value** |
| num\_classes  | 2  |
| min\_child\_weight  | 1  |
| debug\_verbose  | 0  |
| gpu\_id  | 0  |
| skip\_drop  | 0.4  |
| colsample\_bytree  | 0.7  |
| min\_data\_in\_bin  | 1  |
| objective  | binary:logistic  |
| reg\_lambda  | 1.0  |
| booster  | gbtree  |
| max\_drop  | 50  |
| monotonicity\_constraints  | False  |
| max\_leaves  | 16  |
| min\_child\_samples  | 1  |
| max\_delta\_step  | 0.0  |
| silent  | True  |
| n\_estimators  | 1800  |
| n\_gpus  | 1  |
| boosting\_type  | gbdt  |
| model\_id  | 1  |
| random\_state  | 212233661  |
| scale\_pos\_weight  | 1.0  |
| subsample\_freq  | 1  |
| early\_stopping\_rounds  | 100  |
| reg\_alpha  | 0.0  |
| early\_stopping\_threshold  | 0  |
| grow\_policy  | lossguide  |
| subsample  | 0.9  |
| tree\_method  | gpu\_hist  |
| learning\_rate  | 0.04  |
| rate\_drop  | 0.1  |
| gamma  | 0.01  |
| max\_depth  | 0  |
| max\_bin  | 256  |
| eval\_metric  | logloss  |
| n\_jobs  | 4  |
| nfolds  | 5  |