25 Jan 2021

SGUnited Mid-career Pathways Programme - i.am-vitalize - AI0203
Data Science Project - Team 4 - DSBank - Members

- Tan Boon Chuan Alex John
- Tan Hong Sze
- Wong Wei Jie
- Yeong Siew Har

Task Assignment

Task	Assigned
Planning Analytics	Team
Descriptive Analytics	Wei Jie
Diagnostic Analytics	Hong Sze
Predictive Analytics	Siew Har
Prescriptive Analytics	Alex

Schedule

Date		Activity
Tue	12 Jan	Team grouping announced. Team - Initial discussions on project topics.
Thu	14 Jan	Team - AM discussion on topic/ scope. PM discussion with instructor.
Fri	15 Jan	Individual - Data exploration/ experimentation using IBM Cloud and referencing lab 2-5 for guidance. Documentation.
Mon	18 Jan	Team - Online meet up and discussion of issues faced. Data refinery work and AutoAI. Documentation update.
Tue	19 Jan	Team - Online meet up. AutoAI, deployment and environment testing. Documentation update.
Wed	20 Jan	AM - Team discussion with instructor. Documentation update.
Thu	21 Jan	Individual/ Sub-group - experimentation/ personal learning. Documentation update.
Fri	22 Jan	Individual/ Sub-group - experimentation/ personal learning. Documentation update.
Mon	25 Jan	Team - Presentation preparation
Tue	26 Jan	Team - Final Presentation

Planning Analytics - Business Understanding

Introduction

“Customer attrition is a widespread problem that affects firms in a variety of industries. For example, US credit card providers often deal with annual churn rates of about 20 percent, and mobile phone carriers in Europe battle 20 to 38 percent churn, according to the paper.

Lost customers lead to untapped dollars. A McKinsey report estimated that reducing churn could increase earnings of a typical US wireless carrier by as much as 9.9 percent. It's no surprise then that executives in both the United States and Europe say customer retention is their highest marketing priority—and they've been given bigger budgets to fight the battle.” [1]

Background

Mr Chow, a manager at DSBank, is disturbed with more and more customers leaving their credit card services. They would really appreciate, if one could predict for them who is going to get churned so they can proactively go to the customer to provide them better services and turn customers' decisions in the opposite direction.

Problem Statement

How might we use data science to predict potential credit card customer attrition, so that we can provide better loyalty outreach to retain them.

Objectives

- Through data exploration, understand the pattern or find correlations in the data that would lead to customers leaving.
- Producing a model that is able to predict if a credit card customer would leave.
- In choosing a model, the potential loss of income from wrongly predicting an attrition customer (or False Negative) would cost us more than wrongly predicting a staying customer (or False Positive). However, as we continue to strive for more cost savings, we hope to keep a close balance between the two.

Data Gathering

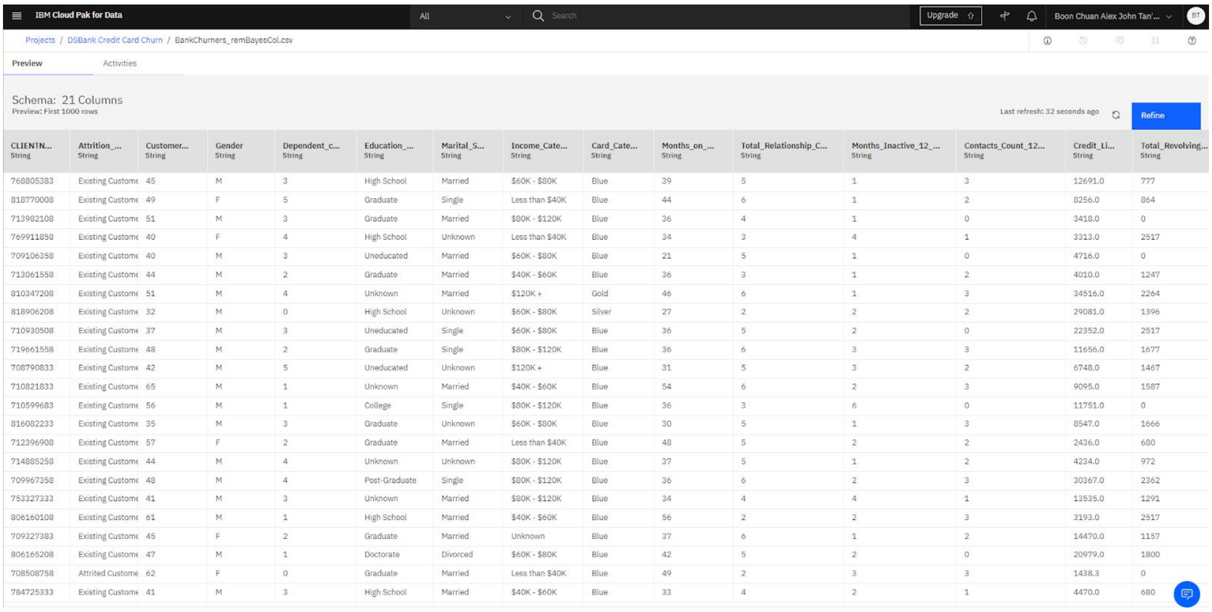
- A subset of data has been extracted from the customer database that also describes if a customer has attrited.
- As we dive deeper into data modeling, training and validation, we may uncover the need for more semi-structured data, such as customer feedback, which is currently unavailable due to being siloed in a different department. Also, feedback from online social networking sources such as the bank’s Facebook or Twitter page, cannot be easily linked to existing customer profiles.

Descriptive Analytics

Data Exploration and Preparation

The data will be a subset extracted from our Bank's customer database. (*For the purpose of this project, it is taken from https://www.kaggle.com/sakshigoyal7/credit-card-customers).

The csv file is downloaded from kaggle and uploaded to IBM Cloud Watson Studio. Previewing the dataset schema gives us an idea of its content.



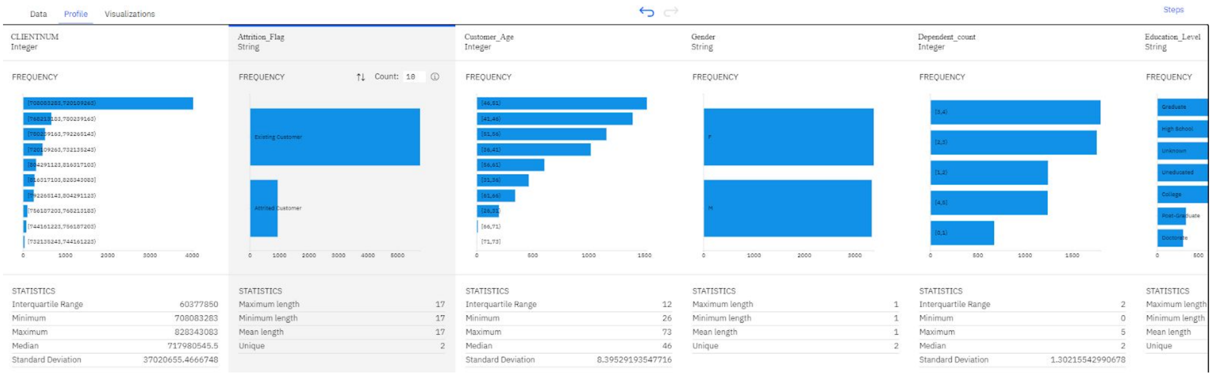
Projects / DSBank Credit Card Churn / BankChurners_remBayesCol.csv

Schema: 21 Columns
Preview: First 1000 rows

Last refresh: 32 seconds ago

CLIENTN... String	Attrition_... String	Customer... String	Gender String	Dependent_c... String	Education_... String	Marital_S... String	Income_Cate... String	Card_Cate... String	Months_on_... String	Total_Relationship_C... String	Months_Inactive_12_... String	Contacts_Count_12... String	Credit_Li... String	Total_Revolving... String
768805383	Existing Custome	45	M	3	High School	Married	$60K - $80K	Blue	39	5	1	3	12691.0	777
818770008	Existing Custome	49	F	5	Graduate	Single	Less than $40K	Blue	44	6	1	2	8256.0	864
713982108	Existing Custome	51	M	3	Graduate	Married	$80K - $120K	Blue	36	4	1	0	3418.0	0
769911858	Existing Custome	40	F	4	High School	Unknown	Less than $40K	Blue	34	3	4	1	3313.0	2517
709106358	Existing Custome	40	M	3	Uneducated	Married	$60K - $80K	Blue	21	5	1	0	4716.0	0
713061558	Existing Custome	44	M	2	Graduate	Married	$40K - $60K	Blue	36	3	1	2	4010.0	1247
810347208	Existing Custome	51	M	4	Unknown	Married	$120K +	Gold	46	6	1	3	34516.0	2264
818906208	Existing Custome	32	M	0	High School	Unknown	$60K - $80K	Silver	27	2	2	2	29081.0	1396
710930508	Existing Custome	37	M	3	Uneducated	Single	$60K - $80K	Blue	36	5	2	0	22352.0	2517
719661558	Existing Custome	48	M	2	Graduate	Single	$80K - $120K	Blue	36	6	3	3	11656.0	1677
708790833	Existing Custome	42	M	5	Uneducated	Unknown	$120K +	Blue	31	5	3	2	6748.0	1467
710821833	Existing Custome	65	M	1	Unknown	Married	$40K - $60K	Blue	54	6	2	3	9095.0	1587
710599683	Existing Custome	56	M	1	College	Single	$80K - $120K	Blue	36	3	6	0	11751.0	0
816082233	Existing Custome	35	M	3	Graduate	Unknown	$60K - $80K	Blue	30	5	1	3	8547.0	1666
712396908	Existing Custome	57	F	2	Graduate	Married	Less than $40K	Blue	48	5	2	2	2436.0	680
714885258	Existing Custome	44	M	4	Unknown	Unknown	$80K - $120K	Blue	37	5	1	2	4234.0	972
709967358	Existing Custome	48	M	4	Post-Graduate	Single	$80K - $120K	Blue	36	6	2	3	30367.0	2362
753327333	Existing Custome	41	M	3	Unknown	Married	$80K - $120K	Blue	34	4	4	1	13535.0	1291
806160108	Existing Custome	61	M	1	High School	Married	$40K - $60K	Blue	56	2	2	3	3193.0	2517
709327383	Existing Custome	45	F	2	Graduate	Married	Unknown	Blue	37	6	1	2	14470.0	1157
806165208	Existing Custome	47	M	1	Doctorate	Divorced	$60K - $80K	Blue	42	5	2	0	20979.0	1800
708508758	Attrited Custome	62	F	0	Graduate	Married	Less than $40K	Blue	49	2	3	3	1438.3	0
784725333	Existing Custome	41	M	3	High School	Married	$40K - $60K	Blue	33	4	2	1	4470.0	680

Looking at the profile tab to get a sense of the data distribution, which would be helpful for further exploration and getting insights on their usefulness to predict customer attrition.



STATISTICS
Interquartile Range 60377850
Minimum 708083283
Maximum 828343083
Median 717980545.5
Standard Deviation 37020655.4666748

STATISTICS
Maximum length 17
Minimum length 17
Mean length 17
Unique 2

STATISTICS
Interquartile Range 12
Minimum 26
Maximum 73
Median 46
Standard Deviation 8.39529193547716

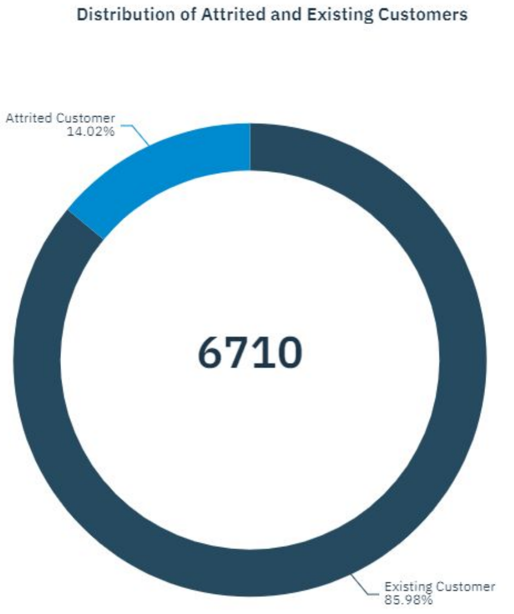
STATISTICS
Maximum length 1
Minimum length 1
Mean length 1
Unique 2

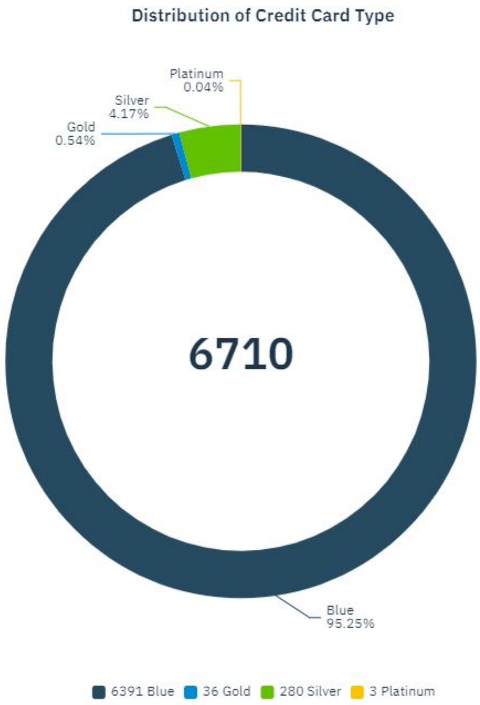
STATISTICS
Interquartile Range 2
Minimum 0
Maximum 5
Median 2
Standard Deviation 1.30215542990678

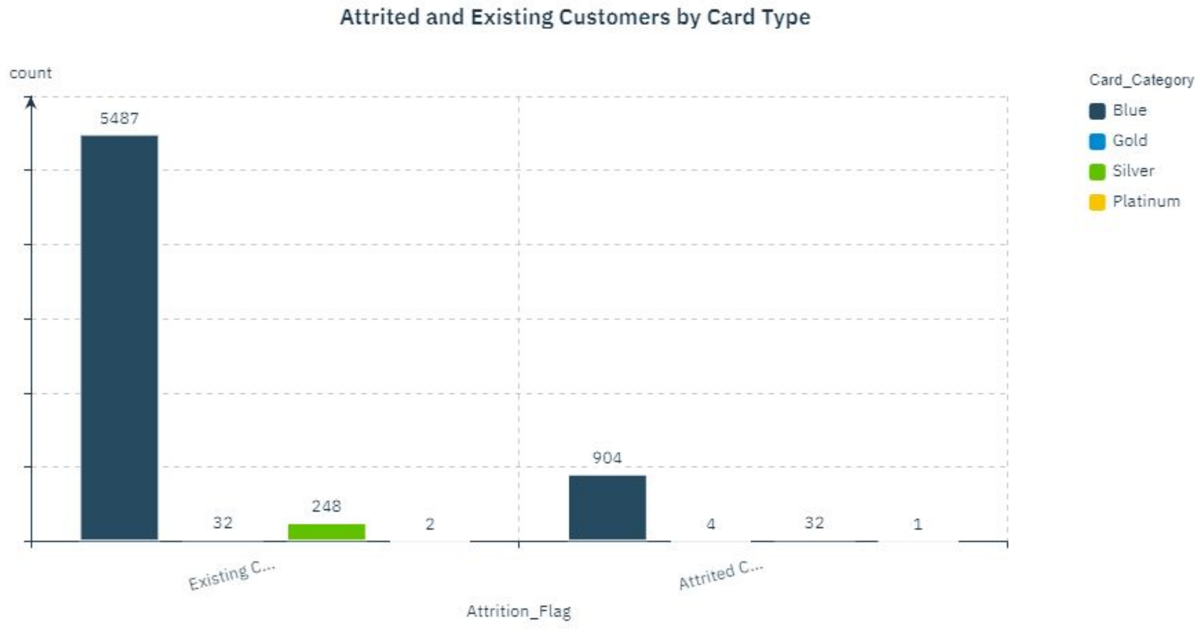
STATISTICS
Maximum length
Minimum length
Mean length
Unique

From the profile tab, a few features are picked and visualized to get a clearer understanding of the data. (In the refinery, the sample size limits us to view 6710 records, or approximately 65% of our dataset).

The number of attrited customers is roughly 14%, which indicates an imbalanced dataset. In the case of card types, an overwhelming 95% are blue card holders, so it should be fine to exclude this feature for now to reduce model calculation time.


Distribution of Attrited and Existing Customers


Distribution of Credit Card Type


Attrited and Existing Customers by Card Type

Data Representation and Transformation

Feature Name	Feature Description	Remove / Keep	Data Transformation
CLIENTNUM	Client number. Unique identifier for the customer holding the account	**Remove**	

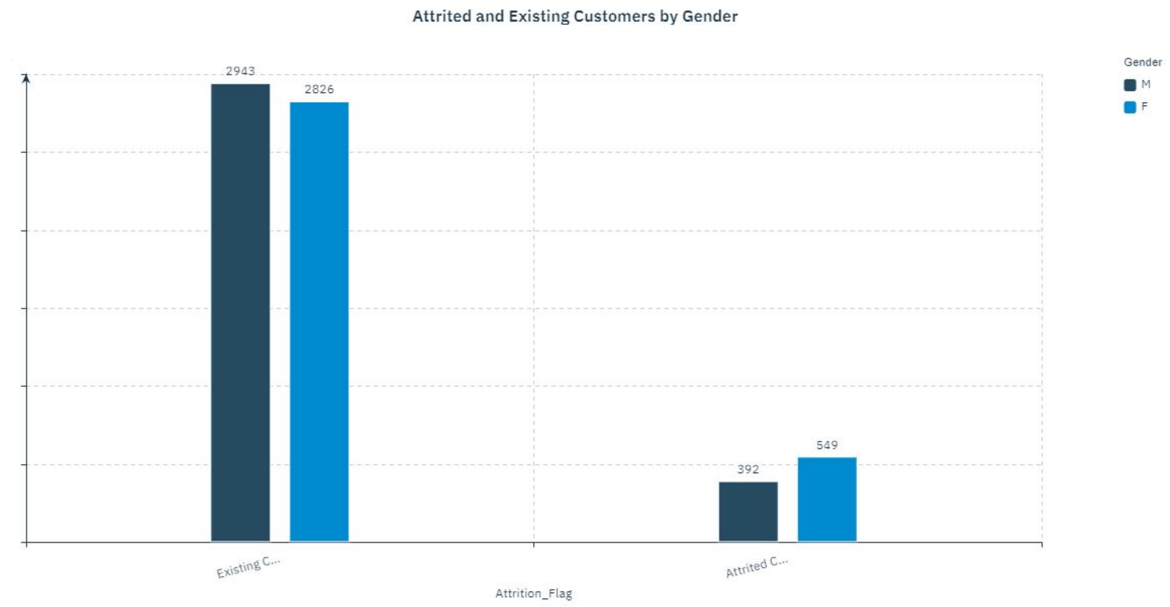
Attrition_Flag	Internal event variable - if the account is closed then 1 else 0
	(Attrited Customer, Existing Customer)
			Attrited... = 1
			Existing...= 0

Customer_Age	Customer's Age in Years		
Gender	M=Male, F=Female		F=1, M=0
Dependent_count	Number of dependents		
Education_Level	Educational Qualification of the account holder (Uneducated, Unknown, High School, College, Graduate, Post-Graduate, Doctorate)		1-Hot Encoding, dropping unknown value
Marital_Status	(Married, Single, Divorced, Unknown)		1-Hot Encoding, dropping unknown value
Income_Category	Annual Income Category of the account holder (Unknown, Less than $40K, $40K - $60K, $60K - $80K, $80K - $120K, $120K +)		1-Hot Encoding, dropping unknown value
Card_Category	Type of Card (Blue, Silver, Gold, Platinum)	**Remove**	
Months_on_book	Period of relationship with bank		
Total_Relationship_Count	Total no. of products held by the customer		
Months_Inactive_12_mon	No. of months inactive in the last 12 months		
Contacts_Count_12_mon	No. of Contacts in the last 12 months		
Credit_Limit	Credit Limit on the Credit Card		
Total_Revolving_Bal	Total Revolving Balance on the Credit Card		
Avg_Open_To_Buy	Open to Buy Credit Line (Average of last 12 months)		
Total_Amt_Chng_Q4_Q1	Change in Transaction Amount (Q4 over Q1)		
Total_Trans_Amt	Total Transaction Amount (Last 12 months)		
Total_Trans_Ct	Total Transaction Count (Last 12 months)		
Total_Ct_Chng_Q4_Q1	Change in Transaction Count (Q4 over Q1)		
Avg_Utilization_Ratio	Average Card Utilization Ratio		

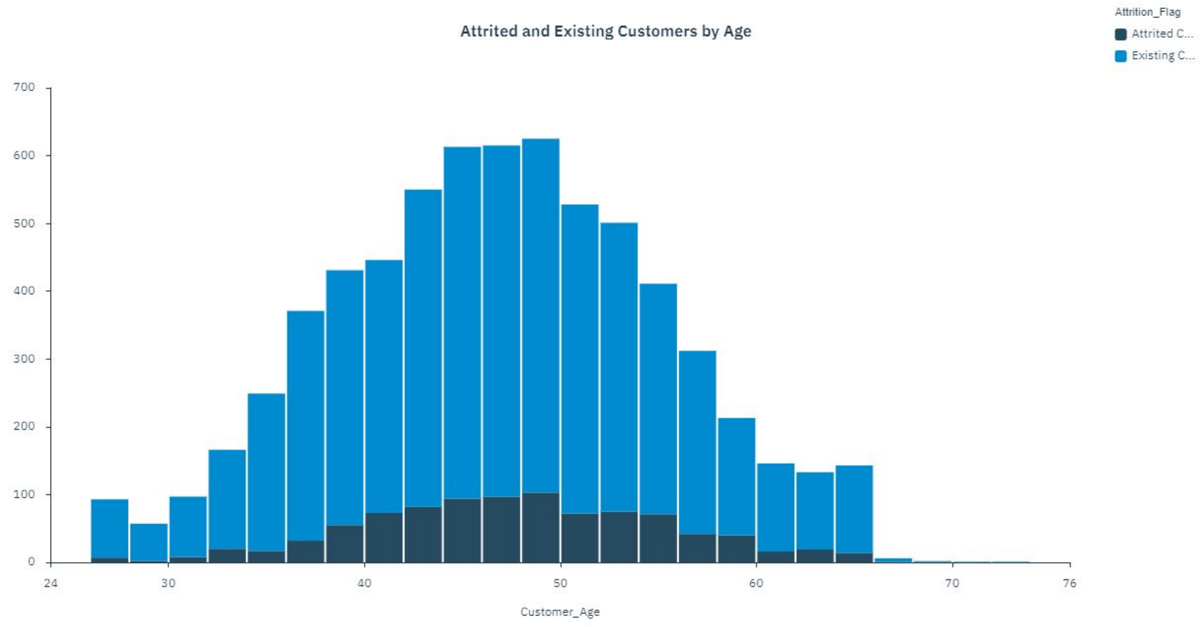
Diagnostic Analytics

Data Visualization and Presentation

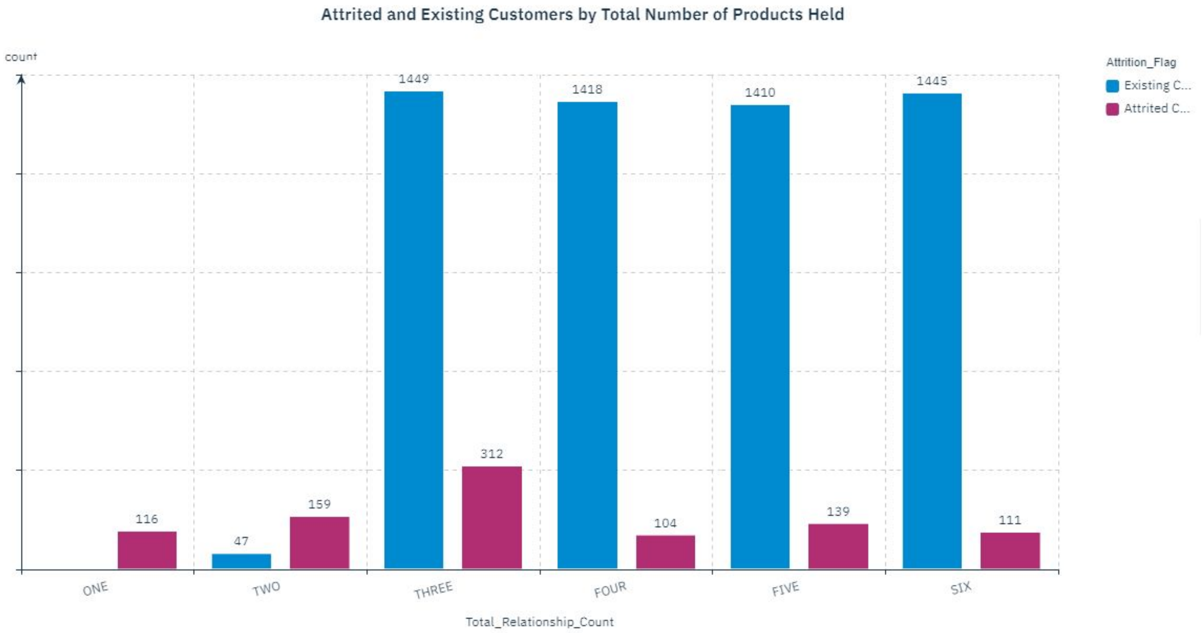
The customer attrition by gender. There might be a slight bias to females leaving the bank’s card service. The model would need further testing if this feature is included in the model training.



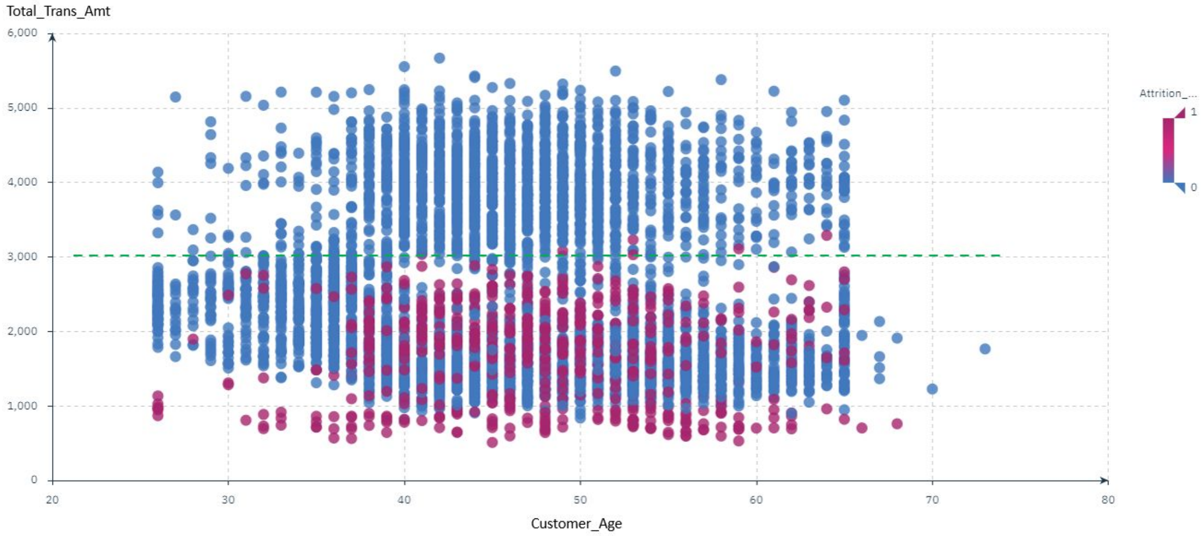
The customer attrition by age.



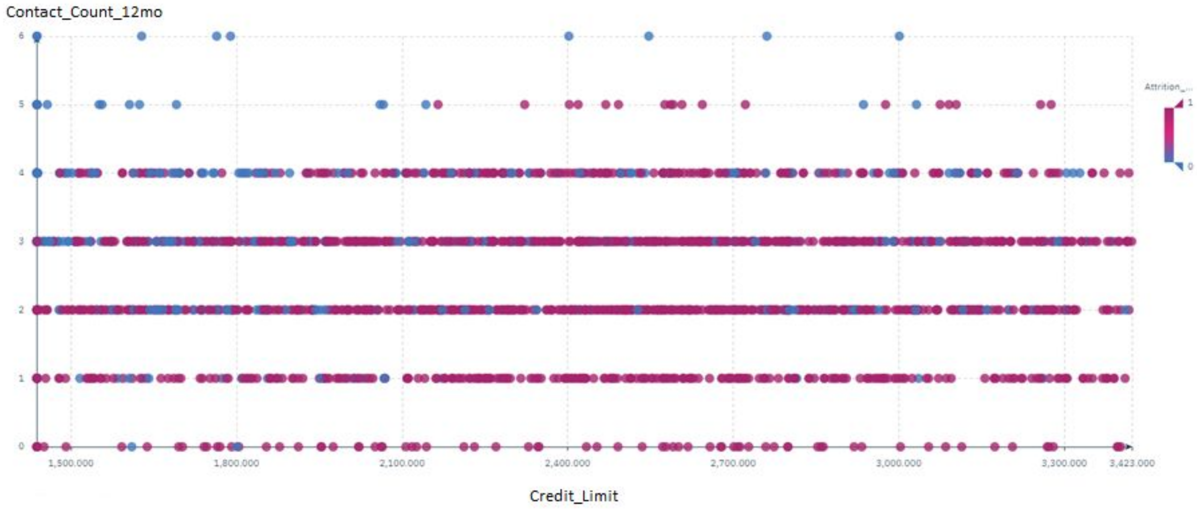
The data suggests a higher tendency for customers holding 1 product to leave, than if they were holding onto 4 or more products.



Across the board, customers who tend to perform greater amounts of transactions (ie more than $3000) are less likely to attrite.

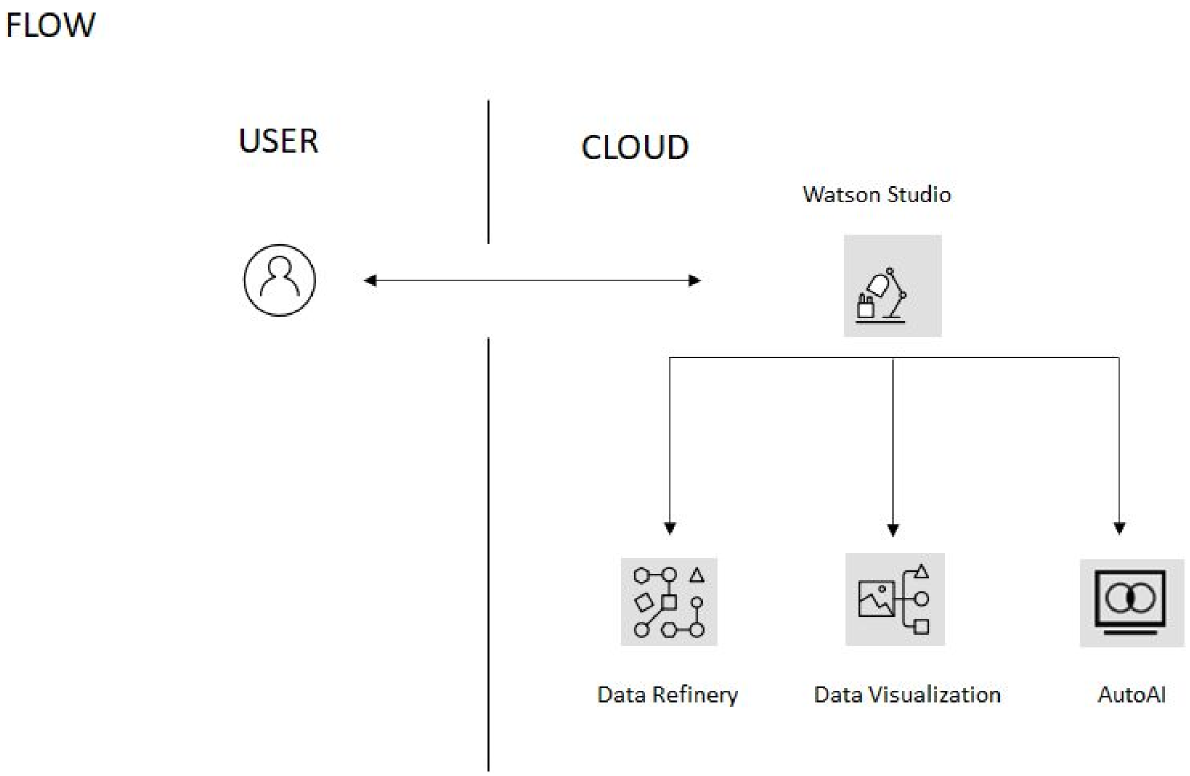


Customers with high service satisfaction, making minimum calls to bank’s contact-center are likely to stay with the bank



Services and Tools Used

The team utilized Watson Studio as well as Data Refinery and Data Visualization tools to work on this project. To help simplify the AI lifecycle, we leveraged on AutoAI graphical tool, which automatically analyzes the data asset and generates candidate model pipelines. A trained model is then selected and deployed for batch processing to generate predictions.



Documentation & Collaborations Tools

Google docs Microsoft Teams zoom

Project limitation

IBM Cloud Watson Studio - Free/ Lite plan - Machine Learning is limited to 20CUH per month - approximately 1 hour of usage.

AutoAI - Each algorithm has to run the 4 pipeline steps - unable to reduce for quicker turnaround testing.

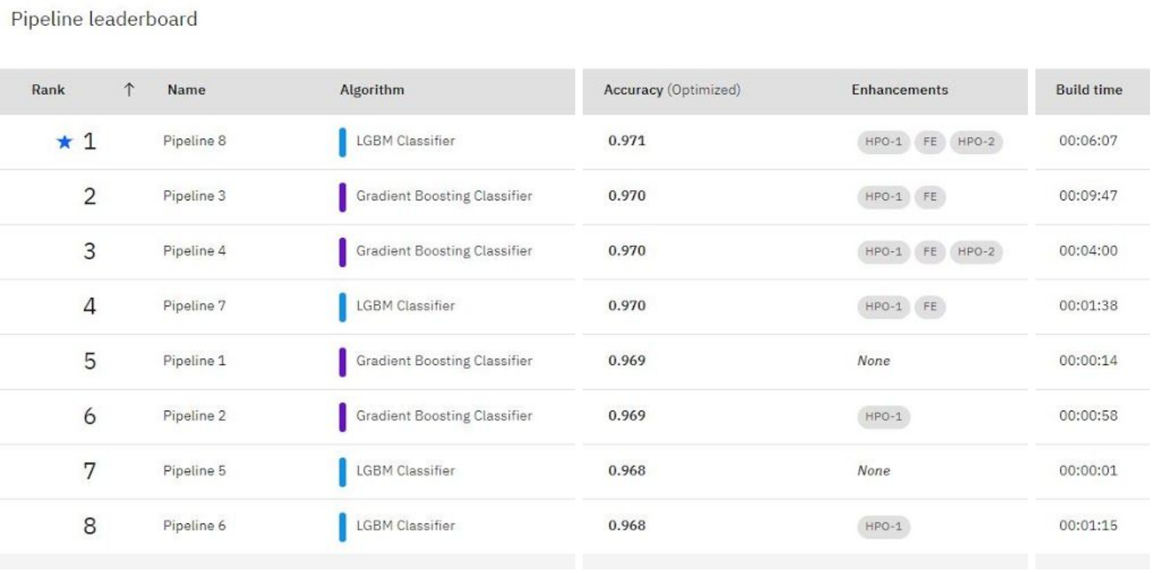
Predictive Analytics

Train Data Models

The models are trained using AutoAI provided in IBM Cloud's Watson Studio.

We kept to AutoAI's default of 90-10 data split for training and hold-out, as well as the default of 2 classifier testing.

After the AutoAI run (approximately 27 minutes), we have the Light Gradient Boosting Machine (LGBM) Classifier with feature engineering and hyper-parameter optimizations - pipeline 8 - coming out at the top. The other classifier picked by AutoAI is the Gradient Boosting Classifier (GBC).



Pipeline leaderboard

Rank ↑	Name	Algorithm	Accuracy (Optimized)	Enhancements	Build time
★ 1	Pipeline 8	LGBM Classifier	0.971	HPO-1 FE HPO-2	00:06:07
2	Pipeline 3	Gradient Boosting Classifier	0.970	HPO-1 FE	00:09:47
3	Pipeline 4	Gradient Boosting Classifier	0.970	HPO-1 FE HPO-2	00:04:00
4	Pipeline 7	LGBM Classifier	0.970	HPO-1 FE	00:01:38
5	Pipeline 1	Gradient Boosting Classifier	0.969	*None*	00:00:14
6	Pipeline 2	Gradient Boosting Classifier	0.969	HPO-1	00:00:58
7	Pipeline 5	LGBM Classifier	0.968	*None*	00:00:01
8	Pipeline 6	LGBM Classifier	0.968	HPO-1	00:01:15

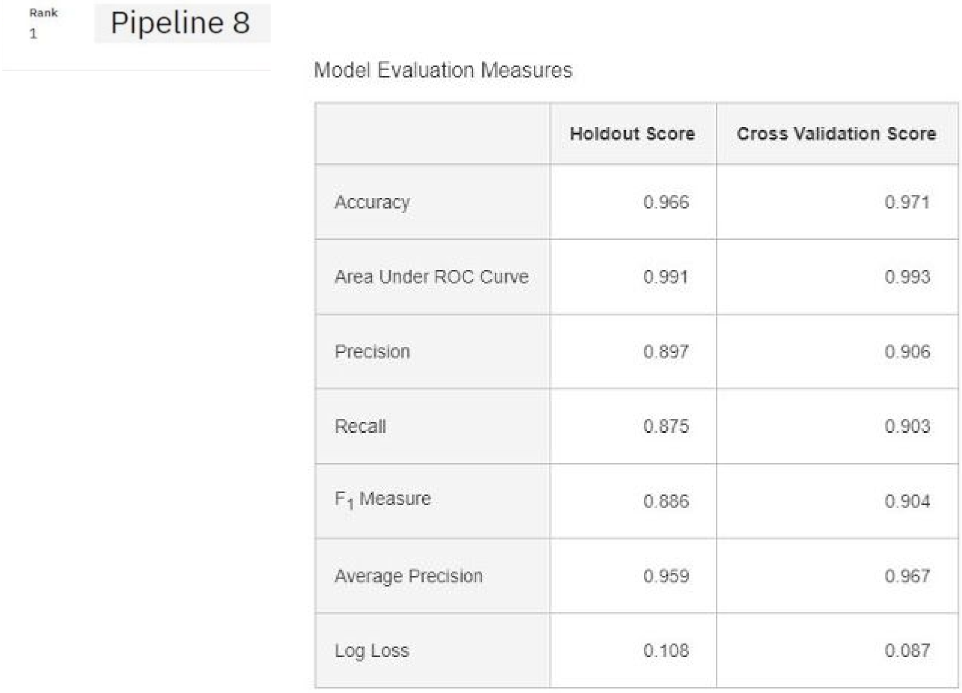
Validate Data Models

Examining the summary results, the 4 pipeline steps in each of the 2 classifiers are quite similar.

Comparing the top ranking pipeline model for each classifier (pipeline 8 for LGBM and pipeline 3 for GBC), the LGBM model has lower false negatives and also has a good balance of false positives to false negatives, which is inline with our objectives.

Hence, the LGBM model generated in pipeline 8 is our chosen model. The following images help to illustrate this.

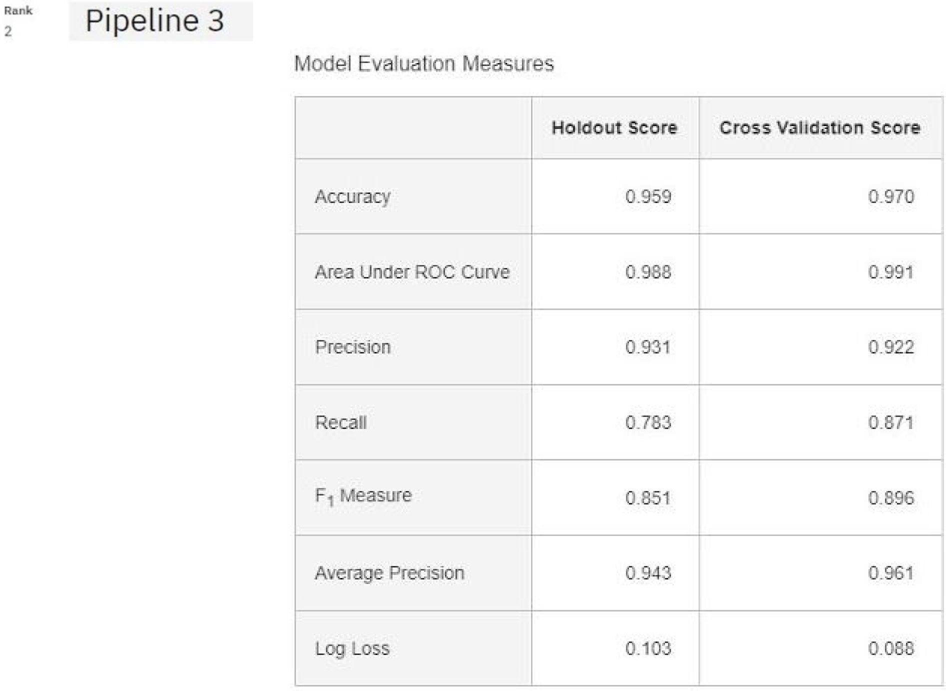
Model evaluation. The precision and recall values are better and more balanced in pipeline 8.



Rank 1 Pipeline 8

Model Evaluation Measures

	Holdout Score	Cross Validation Score
Accuracy	0.966	0.971
Area Under ROC Curve	0.991	0.993
Precision	0.897	0.906
Recall	0.875	0.903
F_1 Measure	0.886	0.904
Average Precision	0.959	0.967
Log Loss	0.108	0.087

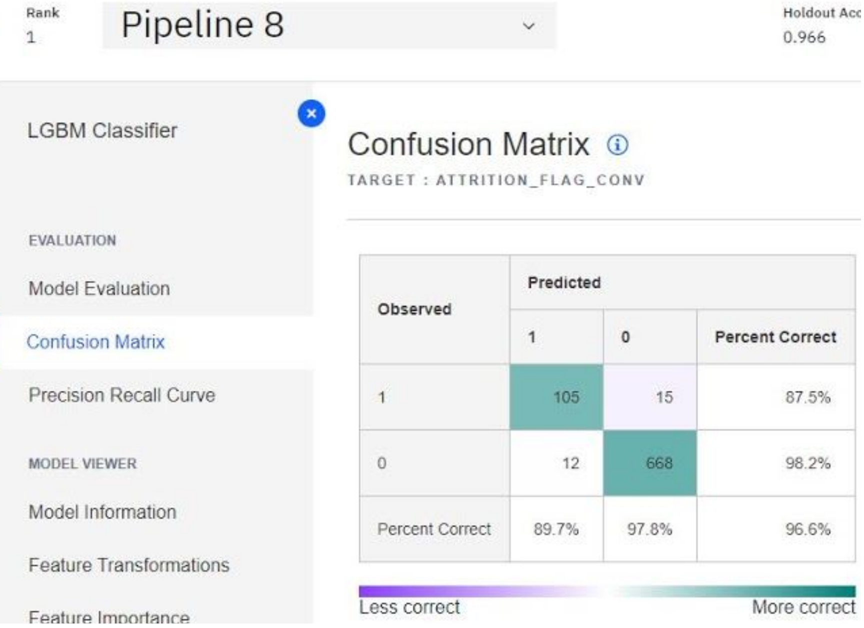


Rank 2 Pipeline 3

Model Evaluation Measures

	Holdout Score	Cross Validation Score
Accuracy	0.959	0.970
Area Under ROC Curve	0.988	0.991
Precision	0.931	0.922
Recall	0.783	0.871
F_1 Measure	0.851	0.896
Average Precision	0.943	0.961
Log Loss	0.103	0.088

Confusion matrix. False negatives are lower in pipeline 8.



Rank 1 Pipeline 8

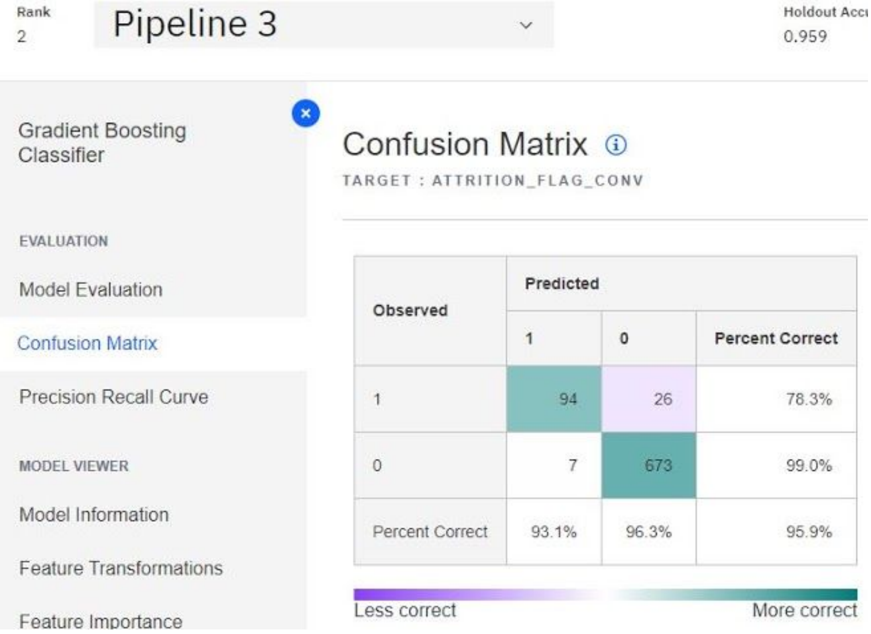
Holdout Acc 0.966

LGBM Classifier

Confusion Matrix

TARGET : ATTRITION_FLAG_CONV

Observed	Predicted		
	1	0	Percent Correct
1	105	15	87.5%
0	12	668	98.2%
Percent Correct	89.7%	97.8%	96.6%



Rank 2 Pipeline 3

Holdout Accı 0.959

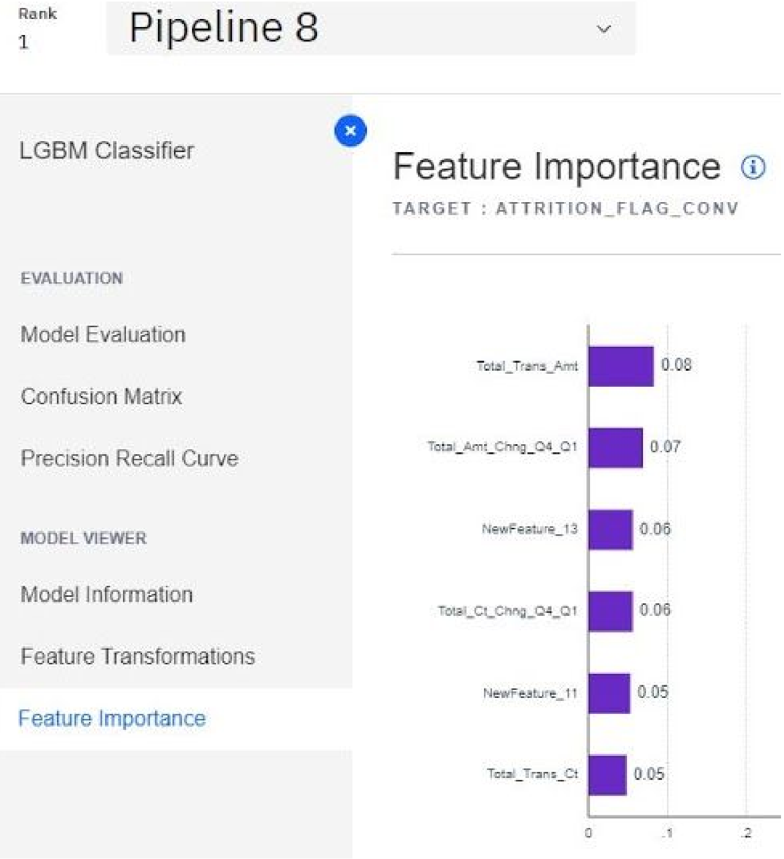
Gradient Boosting Classifier

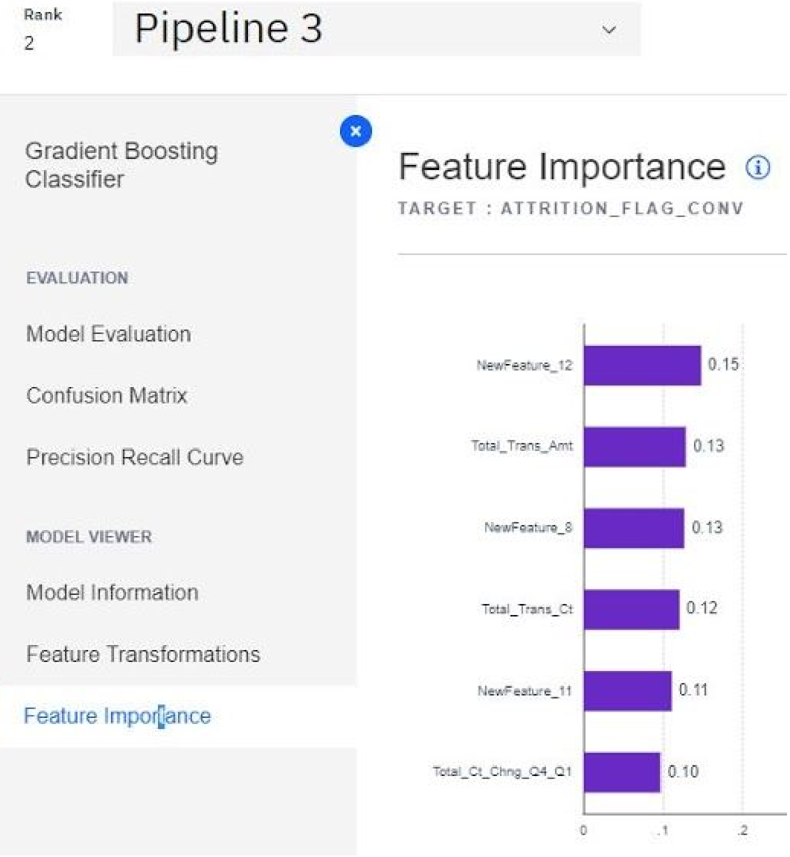
Confusion Matrix

TARGET : ATTRITION_FLAG_CONV

Observed	Predicted		
	1	0	Percent Correct
1	94	26	78.3%
0	7	673	99.0%
Percent Correct	93.1%	96.3%	95.9%

Feature importance. Total_Trans_Amt ranks highly in both models.





Prescriptive Analytics

Deploy Data Models

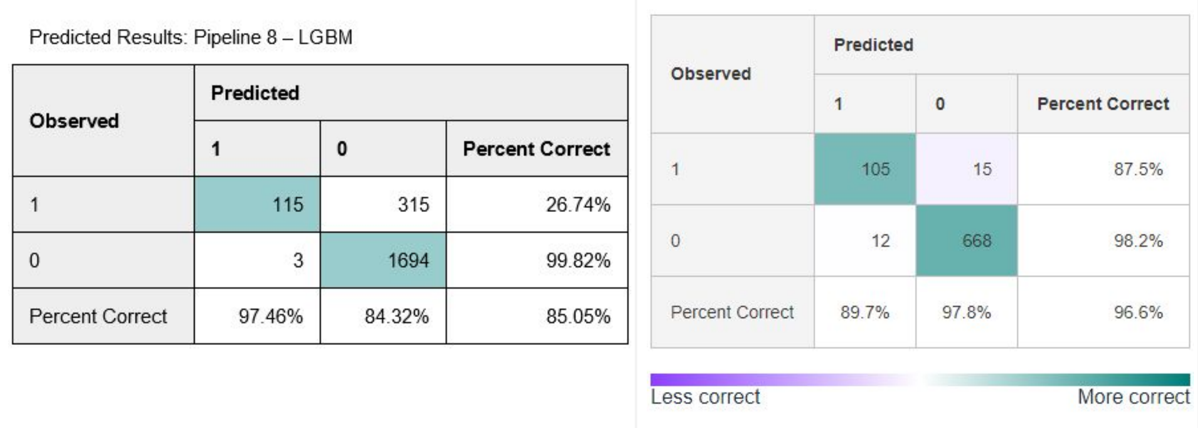
Our chosen model (pipeline 8 - LGBM with feature engineering and optimizations) is deployed into our deployment space for batch processing. This is quite a straightforward process in IBM Cloud.

Environment Feedback

DSBank subsequently provided us with a second batch of data, which contained 2127 rows.

We used the deployed model to process the new data, thereby assessing the usability of the model.

The following figure on the left is a summary of the deployed model's prediction. We noticed that the number of false negatives was particularly high (low recall score), when compared to the holdout results seen from its training, as shown on the right.



Predicted Results: Pipeline 8 – LGBM

Observed	**Predicted**		
	1	**0**	**Percent Correct**
1	115	315	26.74%
0	3	1694	99.82%
Percent Correct	97.46%	84.32%	85.05%

Observed	**Predicted**		
	1	**0**	**Percent Correct**
1	105	15	87.5%
0	12	668	98.2%
Percent Correct	89.7%	97.8%	96.6%

Less correct
More correct

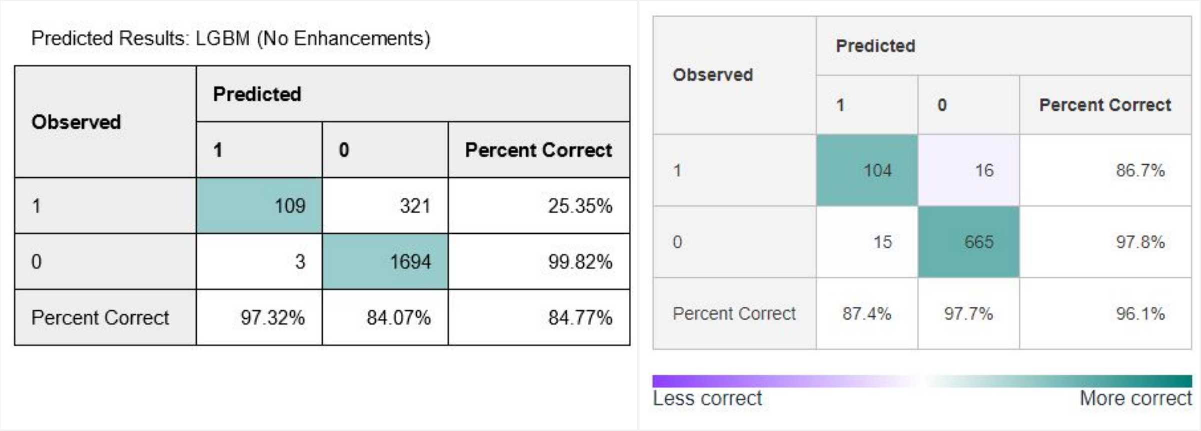
Confusion matrix. Left: Environment Test. Right: AutoAI Training Holdout.
Chosen pipeline 8 - Light Gradient Boosting Machine Classifier, with enhancements.

There are multiple factors that could lead to this high variance, such as the enhancements used in our chosen model, insufficient data for training, too many features or the suitability of our chosen classifier.

Further Testing and Data Science Lifecycle

In order to address this issue of high false negatives, we would need to do further testing, and re-enter the data science lifecycle loop.

We deployed the different data models without enhancements that are generated by AutoAI and found the results to be similar to our initial test. This suggests that our problem most likely lies elsewhere.



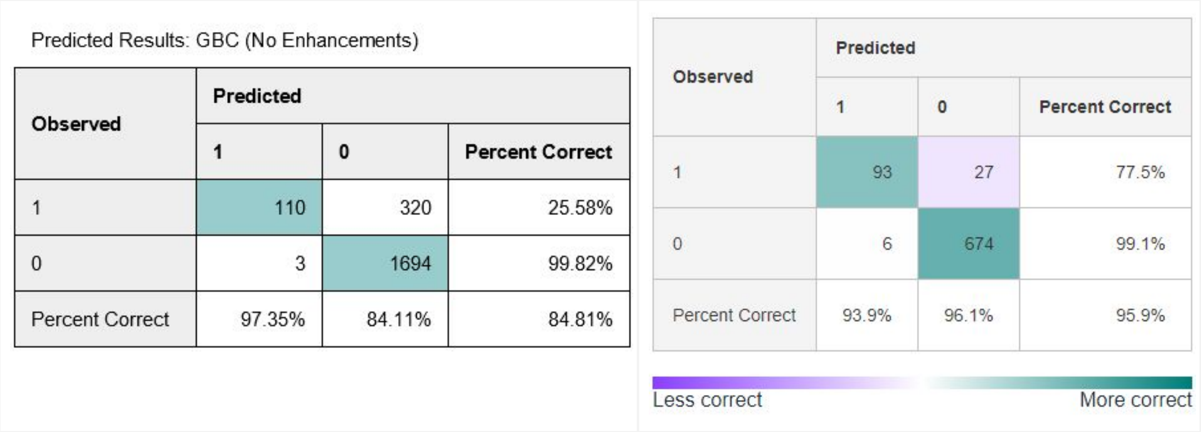
Predicted Results: LGBM (No Enhancements)

Observed	Predicted		
	1	0	Percent Correct
1	109	321	25.35%
0	3	1694	99.82%
Percent Correct	97.32%	84.07%	84.77%

Observed	Predicted		
	1	0	Percent Correct
1	104	16	86.7%
0	15	665	97.8%
Percent Correct	87.4%	97.7%	96.1%

Less correct
More correct

Confusion matrix. Left: Environment Test. Right: AutoAI Training Holdout.
Same Light Gradient Boosting Machine Classifier, but without enhancements.



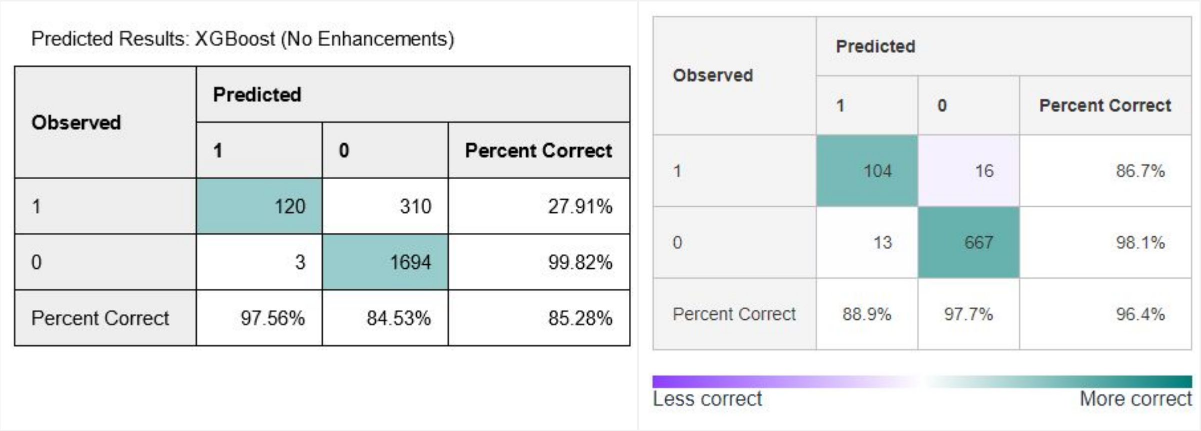
Predicted Results: GBC (No Enhancements)

Observed	Predicted		
	1	0	Percent Correct
1	110	320	25.58%
0	3	1694	99.82%
Percent Correct	97.35%	84.11%	84.81%

Observed	Predicted		
	1	0	Percent Correct
1	93	27	77.5%
0	6	674	99.1%
Percent Correct	93.9%	96.1%	95.9%

Less correct
More correct

Confusion matrix. Left: Environment Test. Right: AutoAI Training Holdout.
Gradient Boosting Classifier (No enhancements).



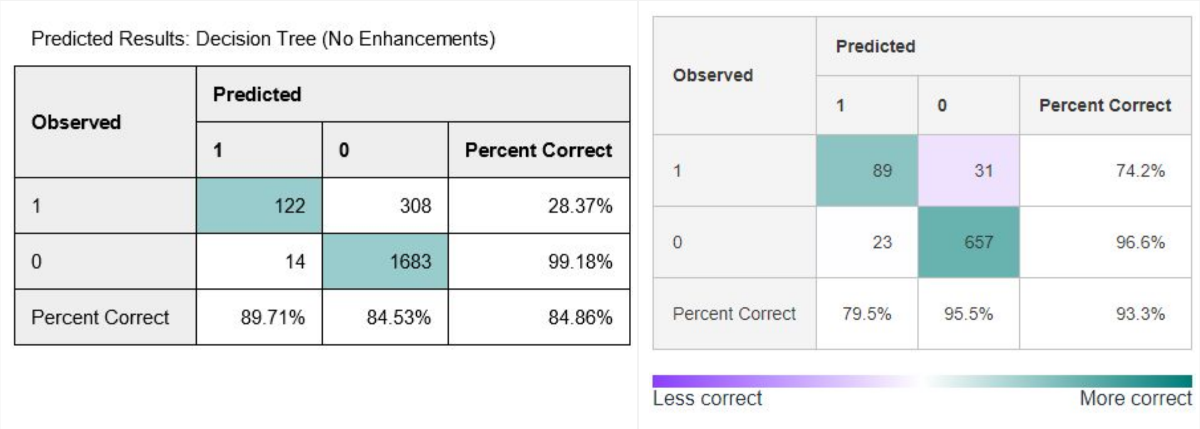
Predicted Results: XGBoost (No Enhancements)

Observed	Predicted		
	1	0	Percent Correct
1	120	310	27.91%
0	3	1694	99.82%
Percent Correct	97.56%	84.53%	85.28%

Observed	Predicted		
	1	0	Percent Correct
1	104	16	86.7%
0	13	667	98.1%
Percent Correct	88.9%	97.7%	96.4%

Less correct
More correct

Confusion matrix. Left: Environment Test. Right: AutoAI Training Holdout.
eXtreme Gradient Boosting Classifier (No enhancements).



Predicted Results: Decision Tree (No Enhancements)

Observed	Predicted		
	1	0	Percent Correct
1	122	308	28.37%
0	14	1683	99.18%
Percent Correct	89.71%	84.53%	84.86%

Observed	Predicted		
	1	0	Percent Correct
1	89	31	74.2%
0	23	657	96.6%
Percent Correct	79.5%	95.5%	93.3%

Less correct
More correct

Confusion matrix. Left: Environment Test. Right: AutoAI Training Holdout.
Decision Tree Classifier (No enhancements).

Our next test would be to return to our initial chosen LGBM model and reduce the features used in its training. Subsequently, if proven unsuccessful, we would have to combine these two sets of data to train a new model, and make a request for a third batch of data and repeat the environment testing.

Once we have a tested model that works, we can then look into incorporating other features like customer complaints/ feedback, which at the moment are semi-structured data in a different department’s database.

Experimentation

Jupyter Notebook

As we delved further into testing, it quickly became apparent that we are traversing beyond what AutoAI is built to solve (automation and ease of use) and more into flexibility. Our preliminary tests exploring feature reduction using AutoAI, showed only marginal improvements and further testings were hampered by the limitations stated earlier. This led to the use of Jupyter Notebook and various python libraries for data science. The notebook output is included in the appendix.

Summary of Tests

With the cleansed csv files outputted from Data Refinery, we focused on eliminating the remaining possibilities - feature reduction and insufficient data.

We noticed that reducing the low importance features only helps marginally, and that without a deeper understanding, further reduction may result in some form of data bias.

We obtained more satisfactory results when we combined the two batches of data and trained a new model. As such, we are unable to continue with environment testing and would need to request for a third batch of data from the bank.

Out of Scope Testing

In the conceptualizing of this project, we premised that the data we obtained (from kaggle) would represent the chronological order of data we received from DSBank - the first 8000 rows of data would be the first batch received, and the remaining 2127 rows, the second batch, which we would then use for environmental testing.

By first shuffling the full data before splitting into the two batches, we found that the resulting tests would be more aligned with each other. We surmised that the distribution of data (with important features) is imbalanced, which resulted in a less than ideal model being trained in our initial run.

In the case of a real world scenario, we would just have to accept that there is insufficient data provided and to combine both sets of data to train a new model, reflecting the need for monitoring the model and updating with the changing patterns in new data.

References

[1]
https://www.forbes.com/sites/hbsworkingknowledge/2013/11/11/a-smarter-way-to-reduce-customer-churn/?sh=a20026c2c0a4

Resources

https://www.kaggle.com/sakshigoyal7/credit-card-customers

Appendix

Jupyter Notebook Output

DSBank Customer Attrition

Using lab 6 as reference

Random Forest Classifier

`train_test_split(X,y,random_state=42)`

train_test_split - find out more at scikit-learn.org

- test_size - default is 0.25
- shuffle - default is True
- stratify - default is None
- random_state - default is None
    - None:
        - Use the global random state instance from numpy.random.
          Calling the function multiple times will reuse the same instance, and will produce different results.

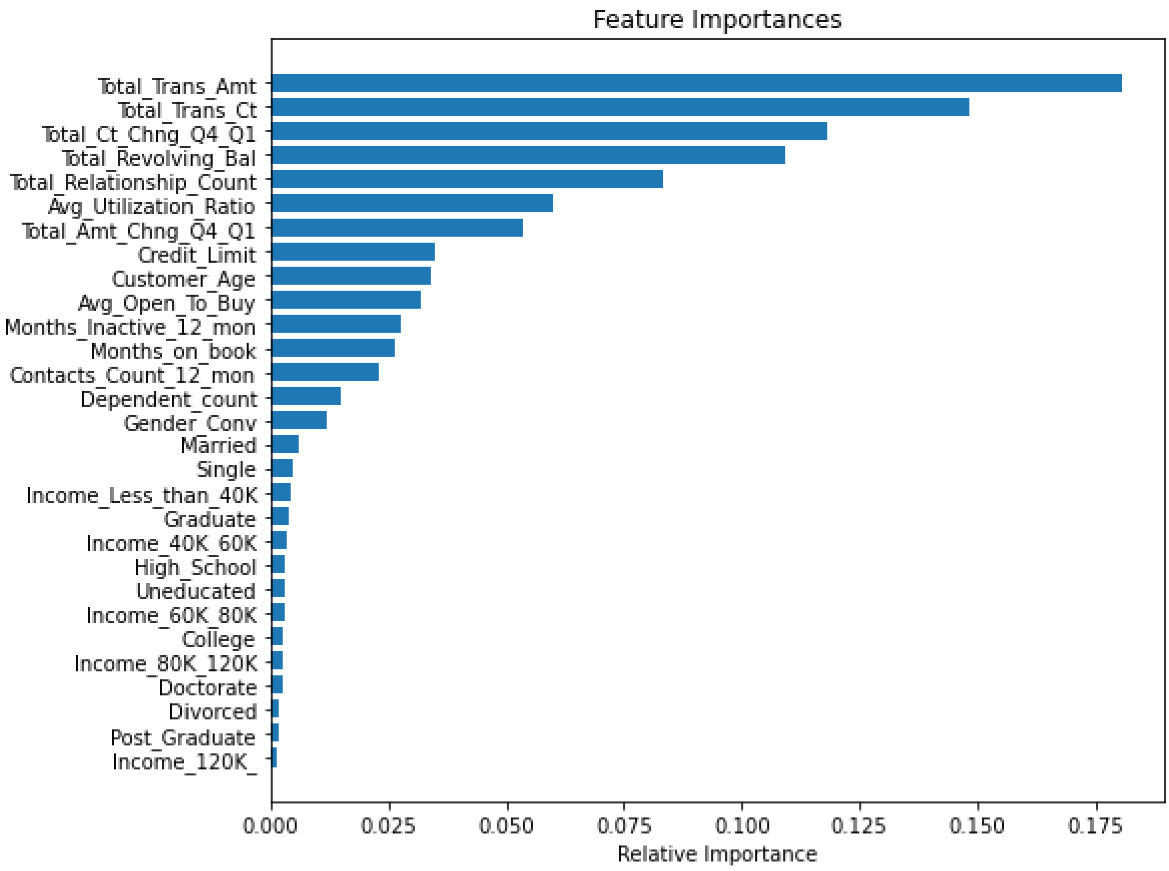
Quick Test

Using the csv file from first batch - 8000 rows.
This is the same file use by AutoAI. The exception being a 75-25 split instead of 90-10.

```
Training Holdout Score: (0.25 of 8000rows:2000)
```

Predicted	0	1
Actual		
0	1667	23
1	55	255

```
Accuracy : 96.10%
Precision: 91.73%
Recall   : 82.26%
```



Similar results are seen (low recall percentage) when doing environment testing using the csv file for deployment.

```
Deploy Test Score: (2127rows)
```

Predicted	0	1
Actual		
0	1694	3
1	316	114

```
Accuracy : 85.00%
Precision: 97.44%
Recall   : 26.51%
```

This follows what we have seen in the various deployed models trained by AutoAI.

Removing Low Weightage Features

The low weighted features related to demographics are removed.

`"Gender_Conv", "Uneducated", "High_School", "College", "Graduate", "Post_Graduate", "Doctorate", "Married", "Single", "Divorced", "Income_Less_than_40K", "Income_40K_60K", "Income_60K_80K", "Income_80K_120K", "Income_120K_"`

The bank related features are kept since we lack domain knowledge to be certain it will not bias the results.

```
Training Holdout Score: (0.25 of 8000rows:2000)
```

Predicted	0	1
Actual		
0	1664	26
1	43	267

```
Accuracy : 96.55%
Precision: 91.13%
Recall   : 86.13%


Deploy Test Score: (2127rows)
```

Predicted	0	1
Actual		
0	1693	4
1	314	116

```
Accuracy : 85.05%
Precision: 96.67%
Recall   : 26.98%
```

The reduction of low weighted features only marginally improved the result.

Combined Data - Training a New Model

A csv file combining both batches (8000 + 2127 rows), is used to train a new model.

```
Train New Model - All Data - Holdout Score: (0.25 of 10127rows:2532)
```

Predicted	0	1
Actual		
0	2089	24
1	95	324

```
Accuracy : 95.30%
Precision: 93.10%
Recall   : 77.33%


With stratify option
Train New Model - All Data - Holdout Score: (0.25 of 10127rows:2532)
```

Predicted	0	1
Actual		
0	2102	23
1	90	317

```
Accuracy : 95.54%
Precision: 93.24%
Recall   : 77.89%
```

With a combined dataset, we achieved a better result with above average recall percentage. The addition of stratify only fractionally improves the result, which shows that the default shuffling is sufficient in this case.

The result below is when shuffle is disabled.

```
Without shuffling
Train New Model - All Data - Holdout Score: (0.25 of 10127rows:2532)
```

Predicted	0	1
Actual		
0	1995	19
1	314	204

```
Accuracy : 86.85%
Precision: 91.48%
Recall   : 39.38%
```

We derived there is an imbalance in the distribution of rows with important features in our data. This might mean there are patterns in the bottom 25% of the dataset that are not present in the top 75%.

Alternate Classifier - Histogram-Based Gradient Boosting

HGBC - find out more at scikit-learn.org

Inspired by LightGBM

```
# explicitly require this experimental feature
from sklearn.experimental import enable_hist_gradient_boosting  # noqa
# now you can import normally from ensemble
from sklearn.ensemble import HistGradientBoostingClassifier
```

Similarly, for this test, we used the combined dataset of 10127 rows.
It is split 75-25 for train and holdout.
Shuffle is enbled and stratify is set to our attrition column.

```
HistGradientBoostingClassifier - All Data - Holdout Score: (0.25 of 10127rows:2532)
```

Predicted	0	1
Actual		
0	2100	25
1	54	353

```
Accuracy : 96.88%
Precision: 93.39%
Recall   : 86.73%
```