

BANK CHURN PREDICTION & PROFILING

PRESENTED BY

CHAIWAT PREMRIDIKUL

6 5 1 0 4 1 4 0 0 4



PROJECT'S AGENDA

00

OVERVIEW

Data flow & Dataset

02

PART 2

Defined focus group and prediction model

05

PART 5

Business Impact

01

PART 1

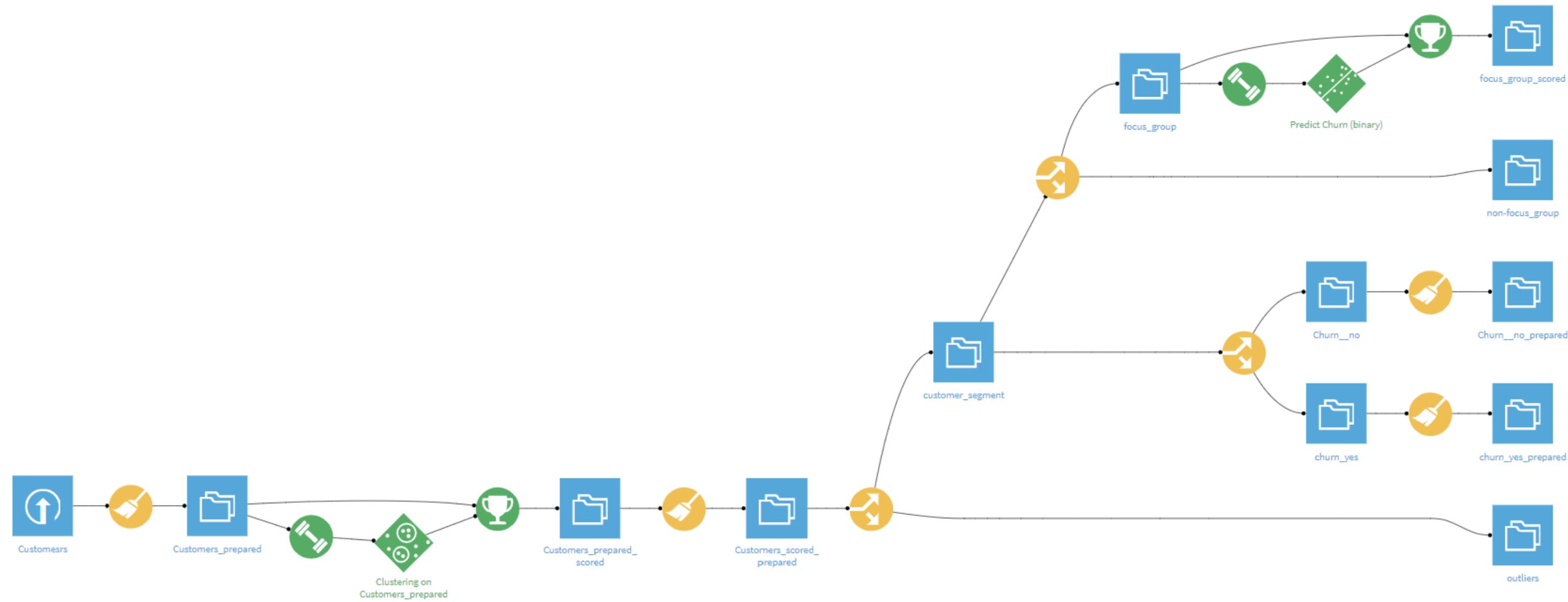
Data preparation and customer segmentation

03

PART 3

Customers and Churners Profiling

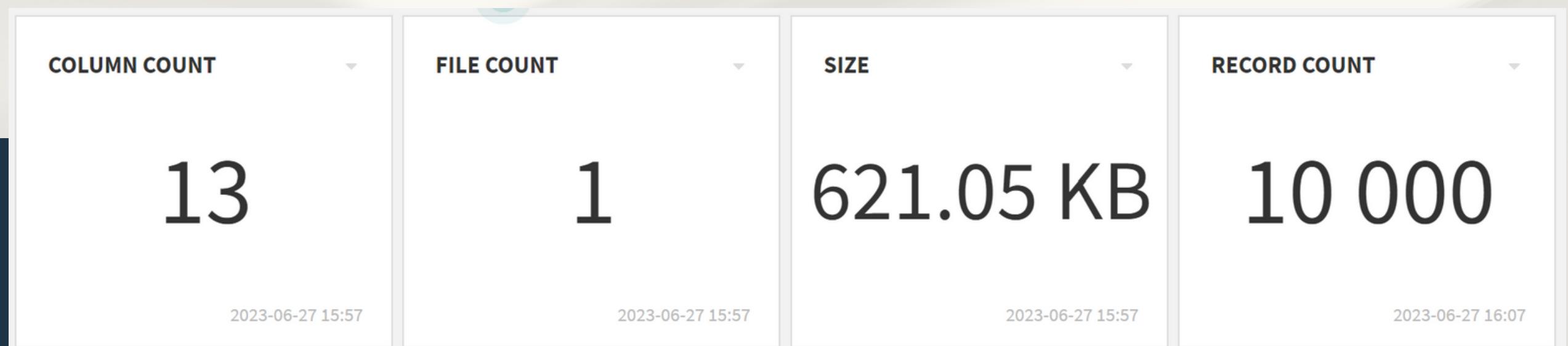
DATA FLOW



DATASET

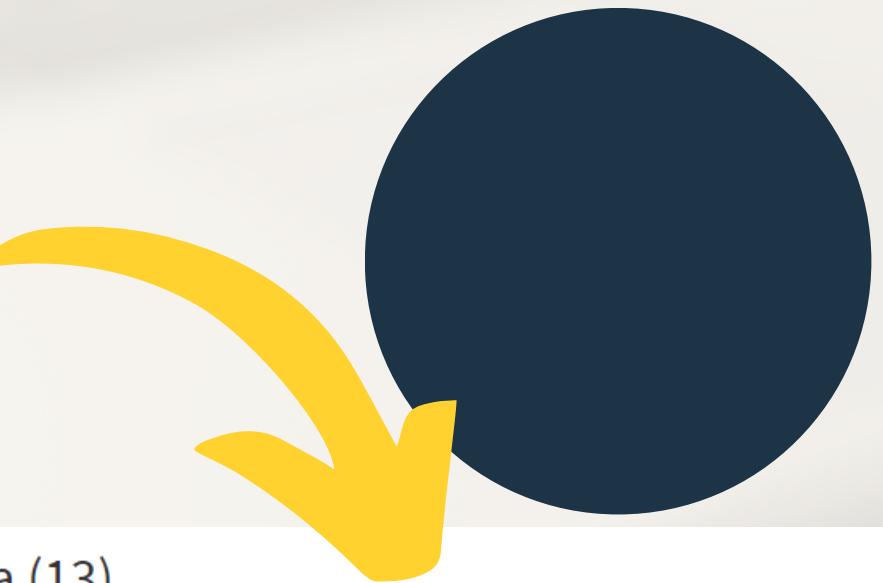
Bank churn customers dataset contained these 13 schema

CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
string Integer	string Text	string Integer	string Country	string Gender	int Integer	int Integer	float Decimal	int Integer	boolean Boolean 	boolean Boolean 	float Decimal	boolean Boolean 
15634602	Hargrave	619	France	Female	42	2	0.0	1	1	1	101348.88	1
15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
15619304	Onio	502	France	Female	42	8	159660.8	3	1	0	113931.57	1
15701354	Boni	699	France	Female	39	1	0.0	2	0	0	93826.63	0
15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.1	0
15574012	Chu	645	Spain	Male	44	8	113755.78	2	1	0	149756.71	1
15592531	Bartlett	822	France	Male	50	7	0.0	2	1	1	10062.8	0
15656148	Obinna	376	Germany	Female	29	4	115046.74	4	1	0	119346.88	1
15792365	He	501	France	Male	44	4	142051.07	2	0	1	74940.5	0
15592389	H?	684	France	Male	27	2	134603.88	1	1	1	71725.73	0
15767821	Bearce	528	France	Male	31	6	102016.72	2	0	0	80181.12	0
15737173	Andrews	497	Spain	Male	24	3	0.0	2	1	0	76390.01	0
15632264	Kay	476	France	Female	34	10	0.0	2	1	0	26260.98	0
15691483	Chin	549	France	Female	25	5	0.0	2	0	0	190857.79	0
15600882	Scott	635	Spain	Female	35	7	0.0	2	1	1	65951.65	0
15643066	Geforth	616	Germany	Male	45	3	143120.41	2	0	1	64327.26	0

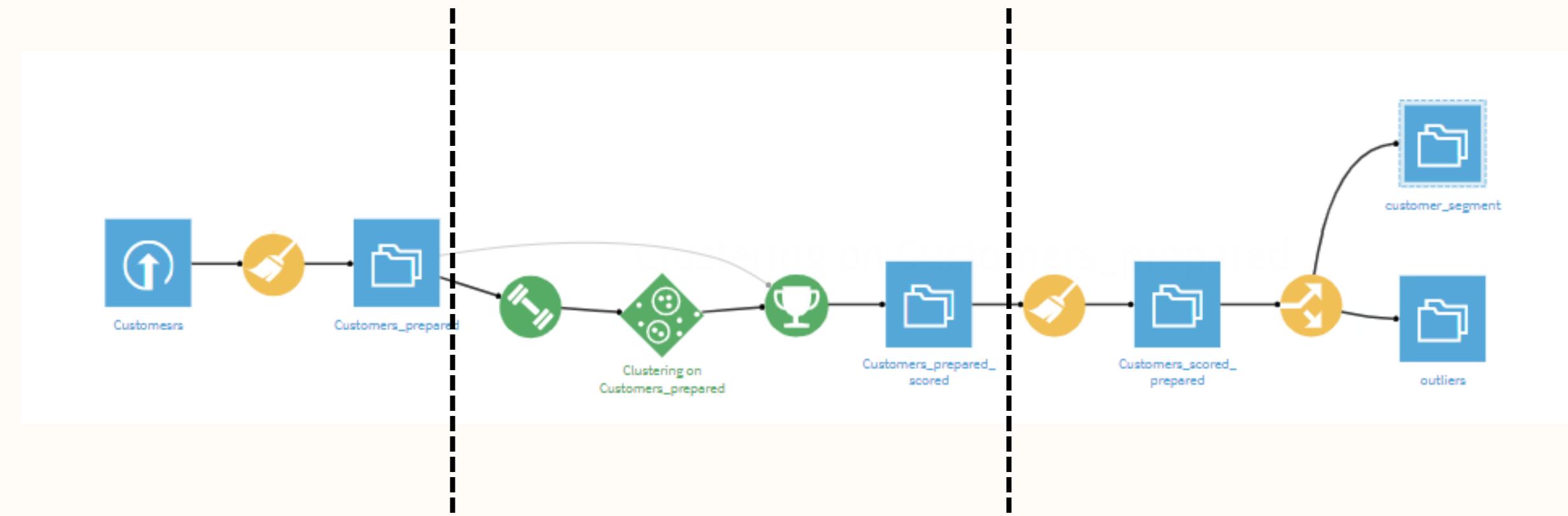


Schema (13)

CustomerId	string
Surname	string
CreditScore	string
Geography	string
Gender	string
Age	int
Tenure	int
Balance	float
NumOfProducts	int
HasCrCard	boolean
IsActiveMember	boolean
EstimatedSalary	float
Exited	boolean



PRAT 1



STEP 01
Data preparation

STEP 02
Cluster modeling

STEP 03
Separate outlier

STEP 1: DATA PREPARATION

- CHANGE COLUMN TYPE
- REPLACE VALUE
- CLEAN OUTLIER

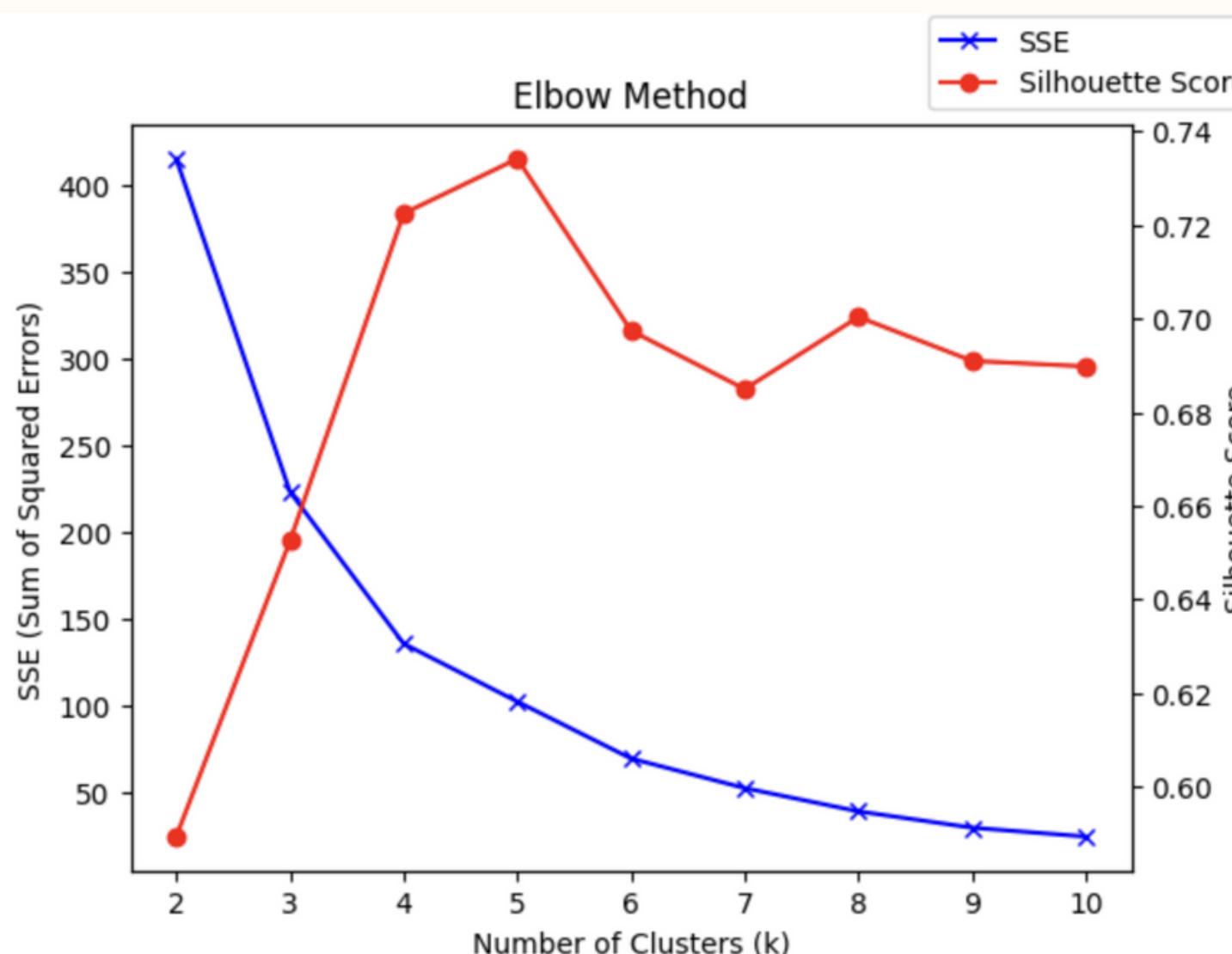
The screenshot shows a data preparation interface with the following details:

- Title:** 6004 churn prediction
- Step:** compute_Churn_Modelling_prepared
- Type:** Script (2 steps)
- Sample settings:** 10,000 first rows
- Description:** Step preview on whole data. Replace 2 values in columns Exited, HasCrCard, IsActiveMember.
- View:** View modified rows (switched off), View all rows, 10,000 rows.
- Columns:** (13) (dropdown), DISPLAY (dropdown), COLUMNS (13) (button), DISPLAY (button).
- Table Preview:** Shows a sample of 10,000 rows with columns: Gender, Age, Tenure, Balance, NumOfProd..., HasCrCard, IsActiveMember, EstimatedSalary, Exited.
- Script Editor (Left Panel):**
 - Replace 2 values in columns Exited, HasCrCard, IsActiveMember:** 10,000 rows.
 - Column:** single | multiple | pattern | all
 - Replacements:**
 - 0 → no
 - 1 → yes
 - Add Column:** + ADD A COLUMN
 - Add Replacement:** + ADD REPLACEMENT
 - Raw text edit:** checkbox
 - Match mode:** dropdown
- Run Button:** RUN (green button)

STEP 2: CLUSTER

DESIGN MODEL

Elbow Method & Silhouette Score



to find number of K for clustering
K = 4, 5

Feature Handling & algorithms

The screenshot shows a user interface for configuring a machine learning model, specifically for the 'Balance' feature. The interface is divided into several sections:

- BASIC**: General settings, Debugging.
- FEATURES**: Features handling (selected), Algorithms, Dimensionality reduction, Outliers detection.
- MODELING**: Algorithms, Dimensionality reduction, Outliers detection.
- ADVANCED**: Runtime environment.

Features Handling section details for 'Balance':

- Role: Input (radio button selected).
- Numerical handling: Keeping as a regular numerical.
- Rescaling: Standard rescaling.
- Make derived feats.: Generate sqrt(x), x^2, ... features (checkbox unchecked).

Distribution statistics for 'Balance':

- Minimum: 0, Maximum: 250898.
- Mean: 76486, StdDev: 62397, Median: 97199.
- Distinct values: 6382, Mode: 0.
- Empty cells: 0.0%, Invalid cells: 0.0%.

Algorithms section:

- KMeans (ON), Gaussian Mixture (ON), Mini-Batch KMeans (OFF), Agglomerative clustering (ON), Spectral clustering (OFF), DBSCAN (OFF), Interactive clustering (OFF), Isolation Forest (OFF).
- CHANGE ALGORITHM PRESETS dropdown.

KMeans configuration:

- ON switch (selected).
- Description: The KMeans algorithm clusters data by separating samples into several clusters, characterized by their centers ("centroids"). The algorithm tries to group the data a...
- Show more... link.

Number of clusters input field: 4, 5.

Seed input field: 1337.

Parallelism input field: 2.

Runtime environment section: ADD CUSTOM PYTHON MODEL button.

STEP 2: CLUSTER

TRAINING RESULT

Previously trained

	SESSION 32	Score	Star
<input type="checkbox"/>	KMeans (k=4) (s32)	0.780	☆
<input type="checkbox"/>	KMeans (k=5) (s32)	0.718	☆
<input type="checkbox"/>	Agglomerative Clustering (K=...)	0.708	☆
<input type="checkbox"/>	Agglomerative Clustering (K=...)	0.767	☆
<input type="checkbox"/>	Gaussian Mixture (k=5) (s32)	0.524	☆
<input type="checkbox"/>	Gaussian Mixture (k=4) (s32)	0.595	☆

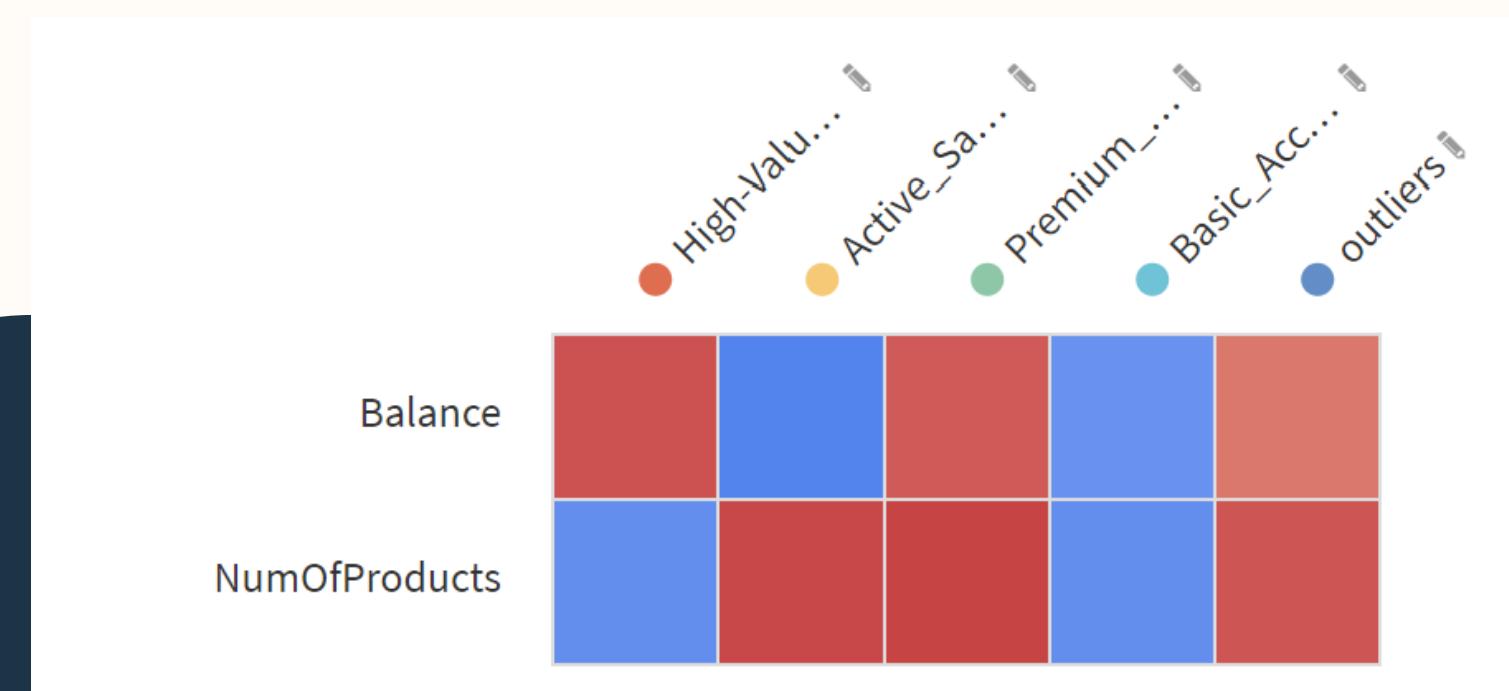
SESSION 32 Started Today at 16:19, ended Today at 16:19 6 models 2 / 13 features

KMeans (k=4) (s32) 0.780 ✓ Done just now (2023-06-27 16:19:09)

Clusters sizes

cluster_0	cluster_1	cluster_2	cluster_3	cluster_outliers
10000	1000	1000	1000	1000

Train set 10000 rows
Train time about a second



01

BASIC ACCOUNT HOLDERS

- low balance
- low no. of product

03

HIGH-VALUE CUSTOMERS

- high balance
- low no. of product

02

ACTIVE SAVERS

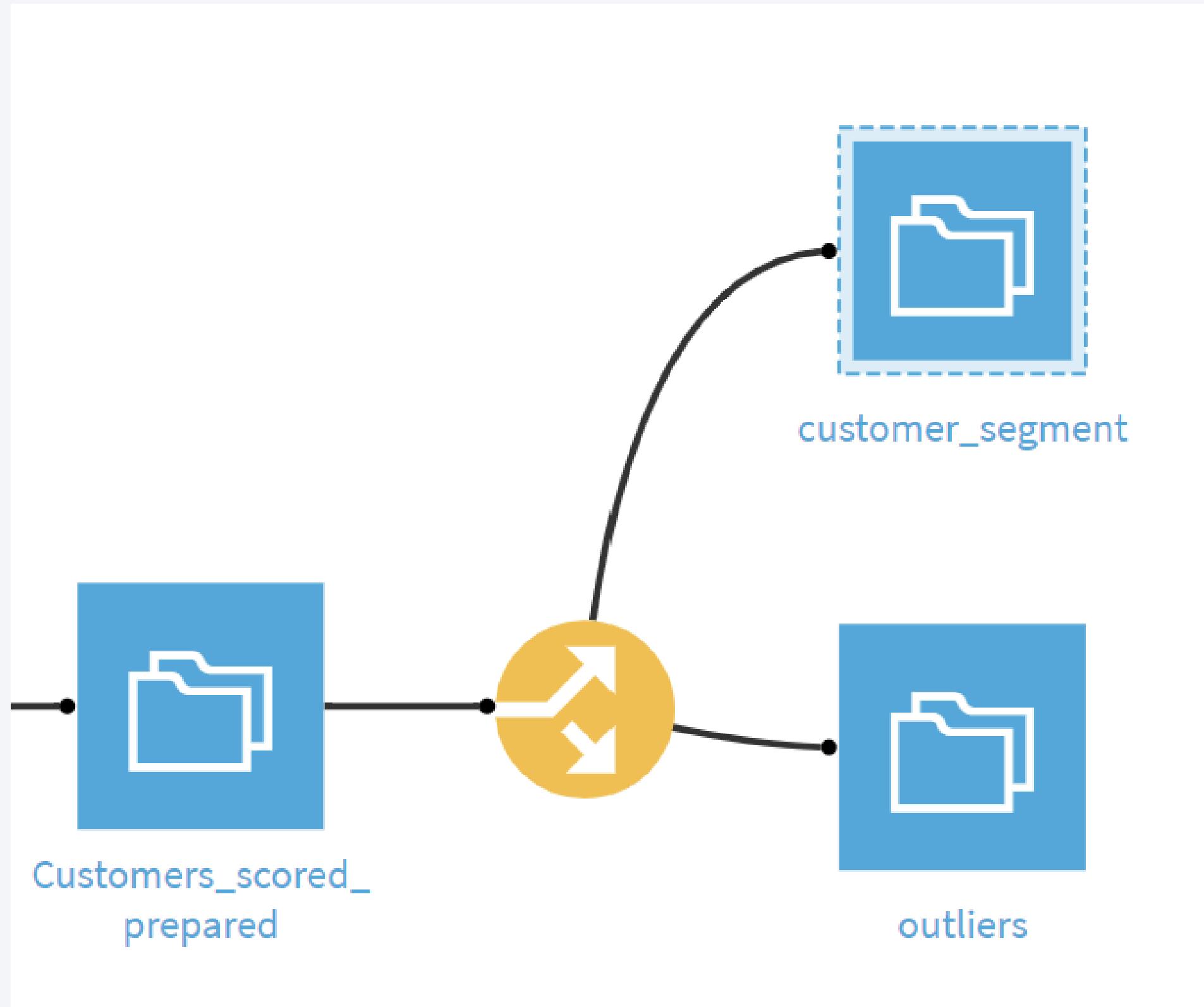
- low balance
- high no. of product

04

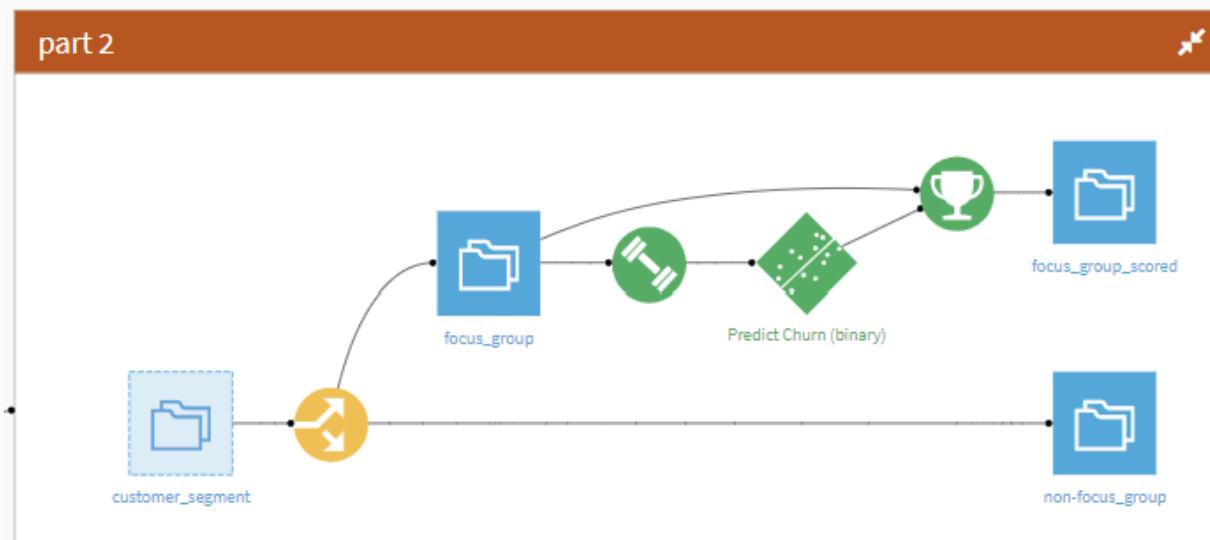
PREMIUM CLIENTS

- high balance
- high no. of product

STEP 3: SEPARATE CLUSTER OUTLIER



PART 2



DEFINED FOCUS
GROUP



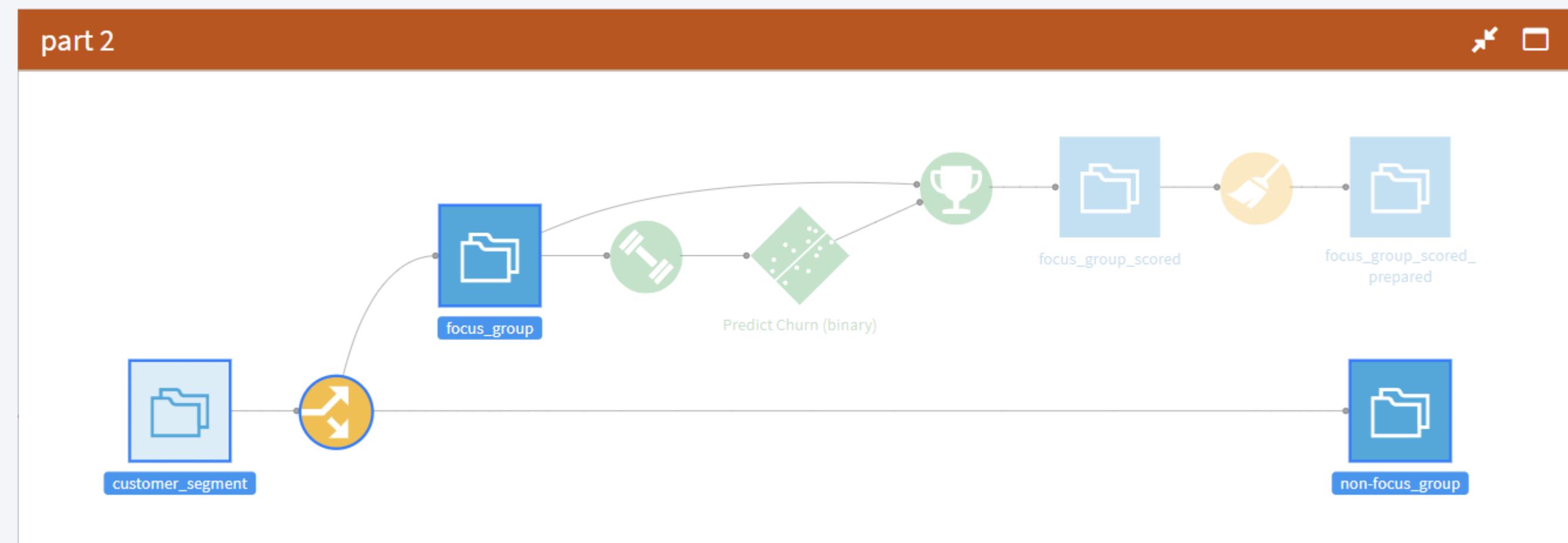
CHURN
PREDICTION

FOCUS TO PREDICT CHURN IN BASIC ACCOUNT HOLDERS & HIGH-VALUE CUSTOMERS WHICH HAVE HIGH CHURN RATIO

Count by Churn and Customer_types

Customer_types	Value	Churn	
		no	yes
Active_Savers	Count of records	94%	6%
Basic_Account_Holders	Count of records	64%	36%
High-Value_Customers	Count of records	74%	26%
Premium_Clients	Count of records	87%	13%

SPLIT DATA INTO FOCUS & NON-FOCUS GROUP



DESIGN MODEL

PREDICT CLASS IMBALANCE

RESAMPLING & K-FOLD CROSS TEST

Target

Prediction type: Two-class classification

Target: Churn

Partitioned models

Partitioning: Not available: input dataset is not partitioned.

Target classes

Proportions of classes in the guess sample

Class	Proportion
no	72%
yes	28%



Train / test set for final evaluation

Policy: Split the dataset

Time ordering

Enabled: OFF

Sampling & Splitting

If your dataset does not fit in your RAM, you may want to subsample the set on which splitting will be performed.

Sampling method: Class rebalance (approx. nb. recor ▾)

Nb. records: 2800

Column: Churn

Split

Randomly: For more advanced splitting, use a split recipe, and then use "Explicit extracts from two datasets" policy

K-fold cross-test: Gives error margins on metrics, but greatly increases training time

Number of folds: 5 Number of folds to divide the dataset into

Random seed: 22422 Using a fixed random seed allows for reproducible result

Stratified: Preserve target variable distribution within every split

Grouped: Rows with the same value for the group column are assigned to only one fold

DESIGN MODEL

METRIX EVALUATION: AUC & F1 SCORE

Metric

Hyperparameter optimization and model evaluation

Optimize model hyperparameters for

AUC

See [the documentation](#)

Optimize threshold for

F1 Score

See [the documentation](#)

Evaluate Model on

+ NEW CUSTOM METRIC

See [the documentation](#)

Lift

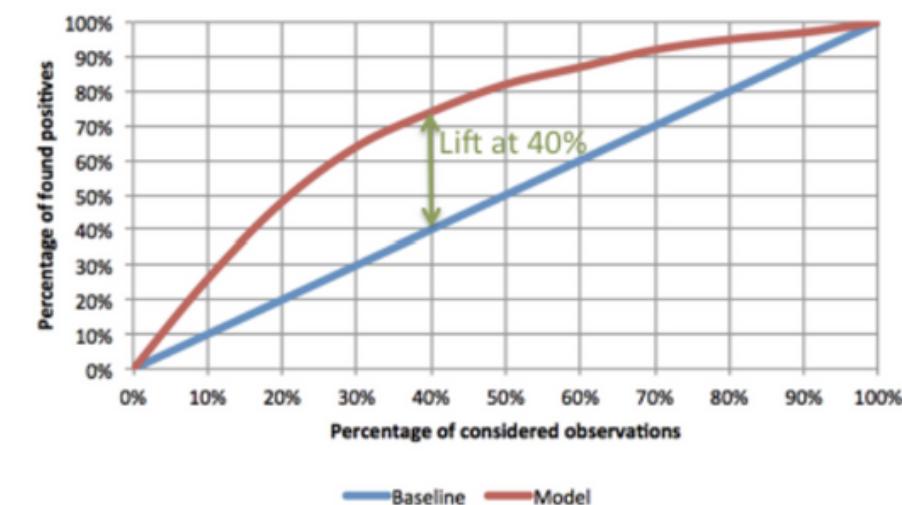
The cumulative lift of a model is the additional proportion of positive records found by a model on a given proportion of the test set, compared to a random model.

DSS computes the cumulative lift for a given proportion that you need to specify.

Compute lift at

40

%



Cost matrix

The cost matrix evaluates a "gain" brought by this model based on the following hypothesis

If the model predicts that **Churn** is true

and it is indeed true, the gain is

-3

but it is not true, the gain is

-3

If the model predicts that **Churn** is false

and it is indeed false the gain is

0

but it is actually true, the gain is

-10

DESIGN MODEL

FEATURE HANDLING

Features Handling [COPY TO...](#) [COPY FROM...](#)

Feature	Action	Status
Surname	Reject	OFF
CreditScore	Reject	OFF
Geography	Dummy encoding	ON
Gender	Reject	OFF
Age	Avg-std rescaling	ON
Tenure	Reject	OFF
Balance	Avg-std rescaling	ON
NumOfProducts	Avg-std rescaling	ON
HasCrCard	Reject	OFF
IsActiveMember	Dummy encoding	ON
EstimatedSalary	Reject	OFF
Churn	Target variable	◎
Customer_types	Dummy encoding	ON

ALGORITHMS SELECTION

Algorithms [CHANGE ALGORITHM PRESETS](#) [COPY TO...](#) [COPY FROM...](#)

Random Forest	ON
Gradient tree boosting	ON
Logistic Regression	ON
LightGBM	ON
XGBoost	ON
Decision Tree	OFF
Support Vector Machine	ON
Stochastic Gradient Descent	OFF
KNN	OFF
Extra Random Trees	ON
Single Layer Perceptron	OFF
Lasso Path	OFF
Deep Neural Network	OFF

+ ADD CUSTOM PYTHON MODEL

Random Forest

A **Random Forest** is made of many decision trees. Each tree in the forest predicts a record, and each tree "votes" for the final prediction.

Show more...

Number of trees

100

Number of trees in the forest.

Default

Adjusts the number of features to sample at each split.

Maximum depth of tree

6 13

Maximum depth of each tree in the forest. Higher values generally increase the quality of the predictions.

Minimum samples per leaf

1

Minimum number of samples required in a single tree node to split this node. Lower values result in increased training and prediction time.

Parallelism

4

Number of cores used for parallel training. Using more cores leads to faster training but at the expense of memory usage.

TRAINING RESULT

Quick modeling of Churn on focus_group

Predict Churn (Binary classification)

DESIGN RESULT

SEARCH... FILTER Metric: ROC AUC

SESSIONS MODELS TABLE

SESSION 1 Started Yesterday at 15:21, ended Yesterday at 15:21 7 models 6 / 14 features

ROC AUC score Time (s)

Gradient Boosted Trees (s1) (no xval.) 0.823
XGBoost (s1) (no xval.) 0.823
Random forest (s1) 0.822 (± 0.039)
Gradient Boosted ... 0.823 (± 0.041)
Logistic Regression (s1) 0.786 (± 0.045)
LightGBM (s1) 0.821 (± 0.033)
XGBoost (s1) 0.823 (± 0.043)
Extra trees (s1) 0.813 (± 0.038)
SVM (s1) 0.816 (± 0.021)

Gradient Boosted Trees ... 0.823 (± 0.041) Done 1 day ago (2023-06-26 15:21:46)

Trees: 100
Learning rate: 0.1
Max depth: 3

Most important variables: Age, Balance, Geography is Germany, IsActiveMember is no, IsActiveMember is yes, Geography is France

Train set: 2775 rows
Train time: about a second

TRAINING RESULT

SUMMARY

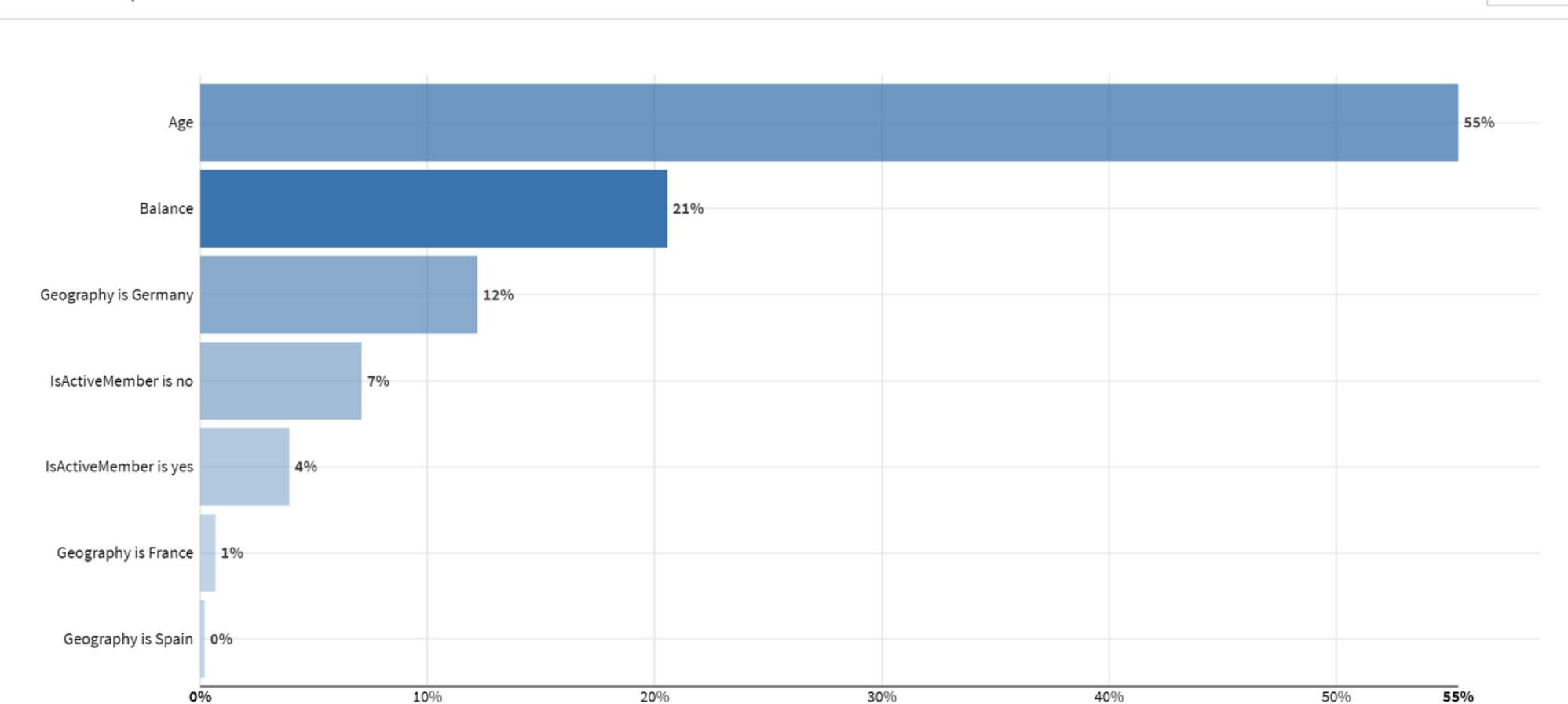
ROC AUC: 0.823

Model

Model ID	A-6004CHURNPREDICTION-wuOVSOh0-ew8AX0Ww-s1-pp5-m1
Model type	Two-class classification
Target	Churn
Classes	no yes
Backend	Python (in memory)
Algorithm	Gbt classification
Trained on	2023/06/26 15:21
Columns	14
Data set rows	2775
Number of folds	5
Calibration method	No calibration
Code Env	DSS builtin env
Python version	3.7.13

VARIABLE IMPORTANT

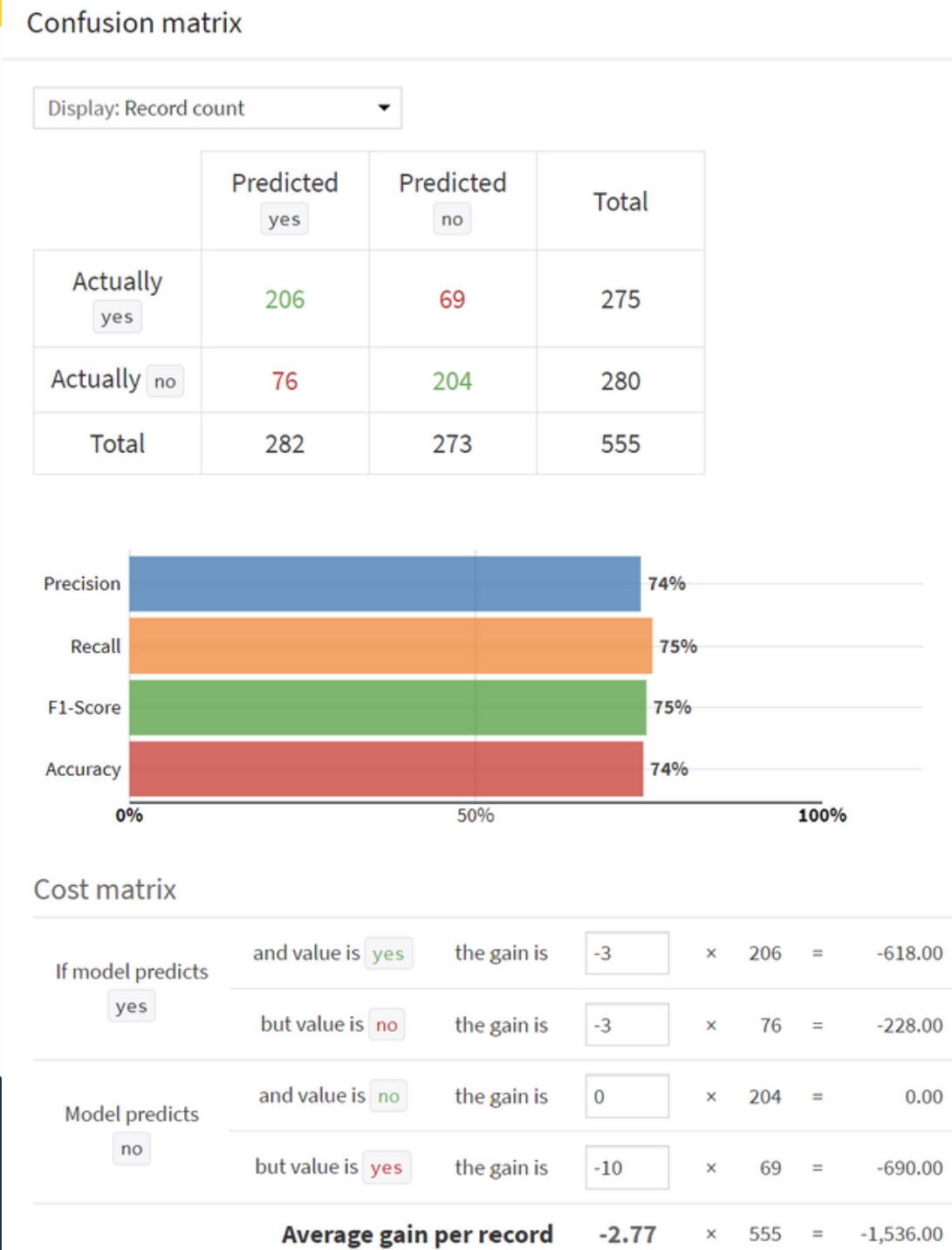
Variable importance



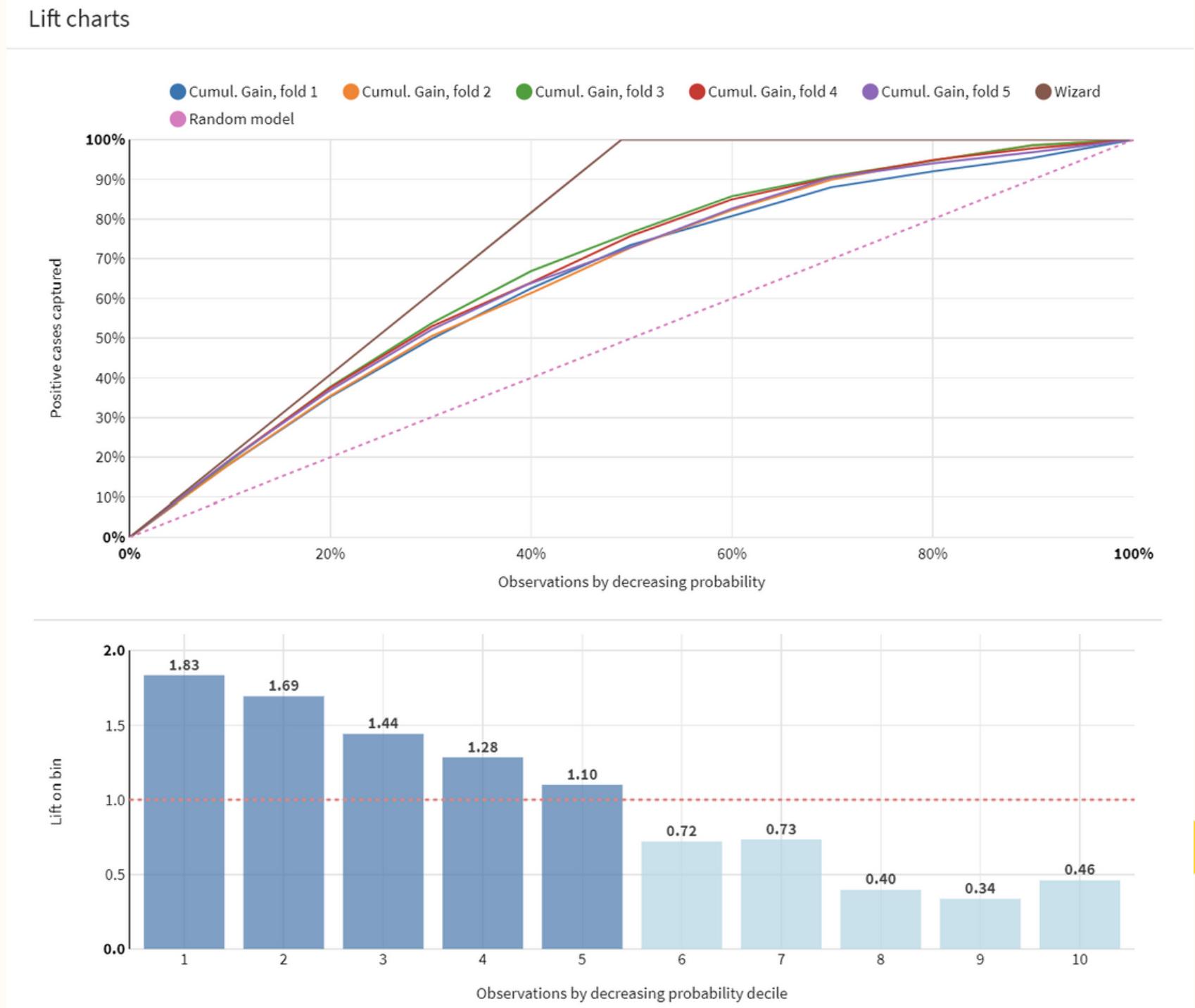
EXPORT

TRAINING RESULT

CONFUSION & COST METRIX



LIFT CHART



TRAINING RESULT

PREDICTION IN TRAINING
DATASET, CORRECT = 74.4%

The screenshot shows a data processing interface with a central preview window and various toolbars and panels around it.

Script Step Details:

- Name:** compute_focus_group_scored_prepared
- Type:** Script
- Sample settings:** 10,000 first rows
- Description:** Step preview on whole data. Create column `score` with formula `if(prediction==Churn,1,0)`.
- Step ID:** 5084
- Output column:** score
- Expression:** `if(prediction==Churn,1,0)`

Preview Window:

A modal dialog titled "score" - (2 distinct) displays the distribution of the "score" column. The summary table shows:

	Count	%	Cum. %
1	3782	74.4	74.4
0	1302	25.6	100.0

The preview also shows a list of 5,084 rows from the dataset, which includes columns: proba_yes, prediction, and score. The prediction column values are mostly 1 (yes) and some 0 (no).

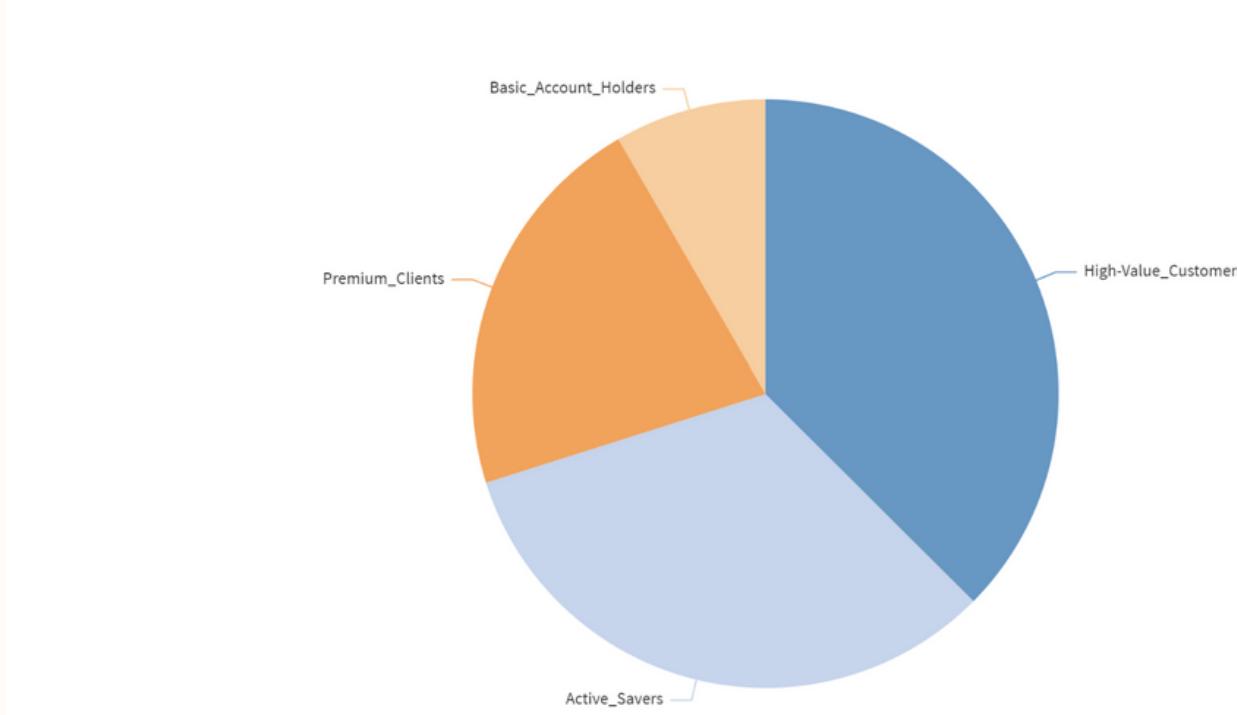
PART 3: CUSTOMER & CHURNER PROFILE





CUSTOMER PROFILE

Customer Type Ratio



7381 records

Run on DSS



Customer by Geography and Types

7381 records

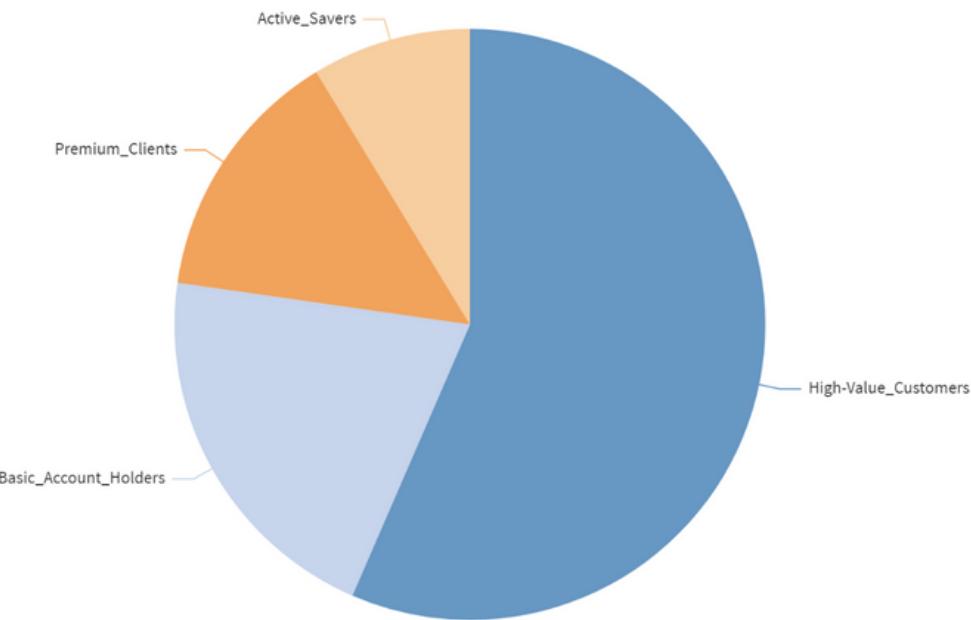
Run on DSS





CHURNER PROFILE

Churner Type Ratio

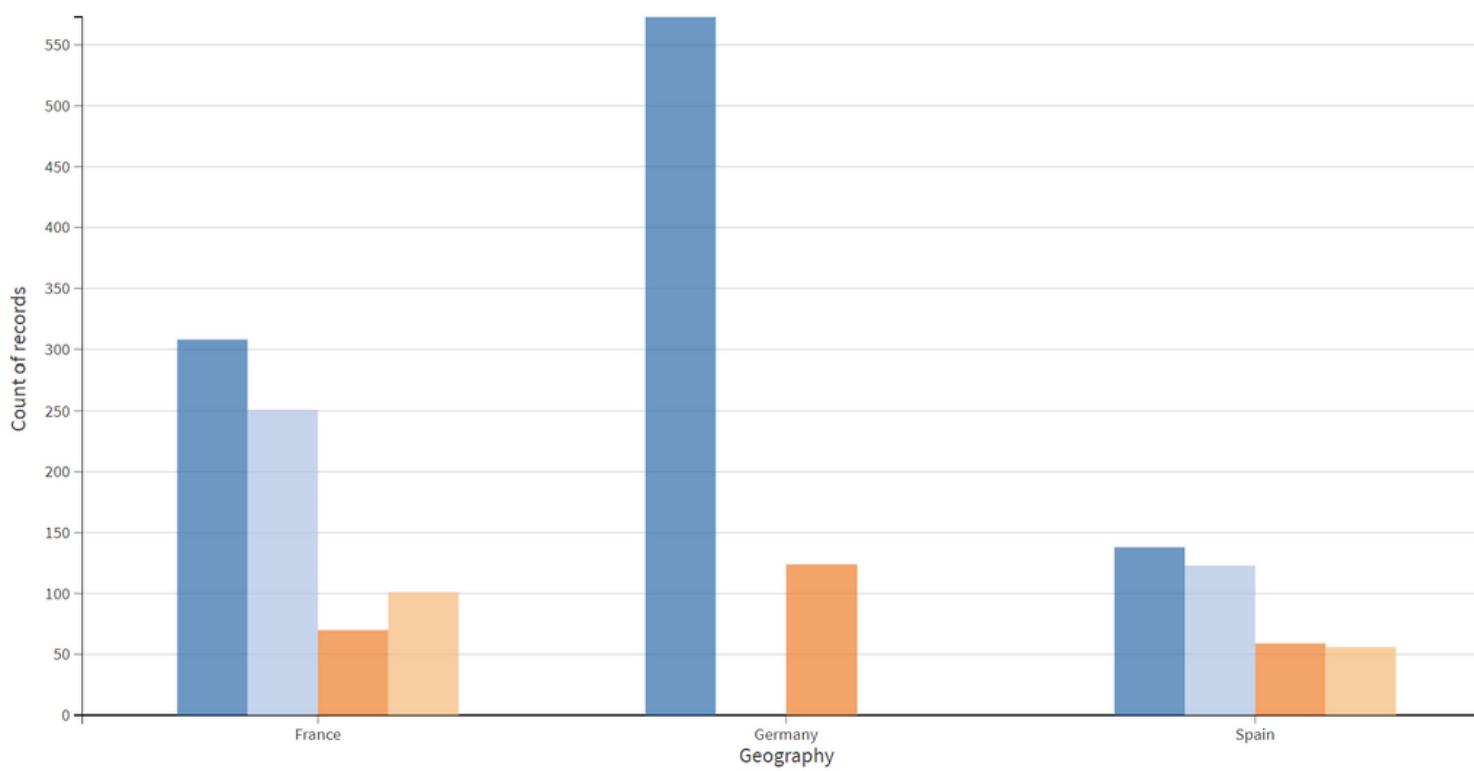


1803 records

Run on DSS



Churner by Geography and Type

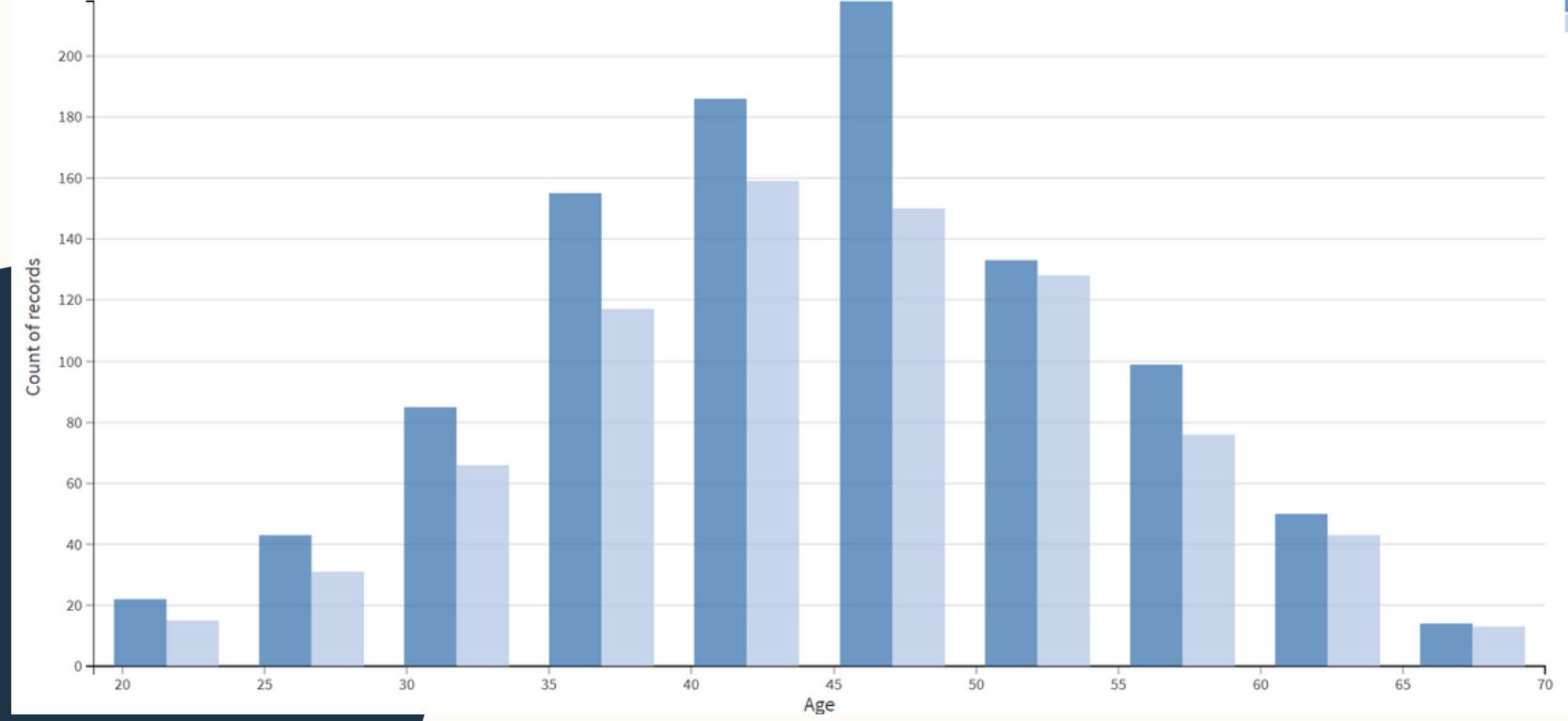


1803 records

Run on DSS



Churner Age and Gender

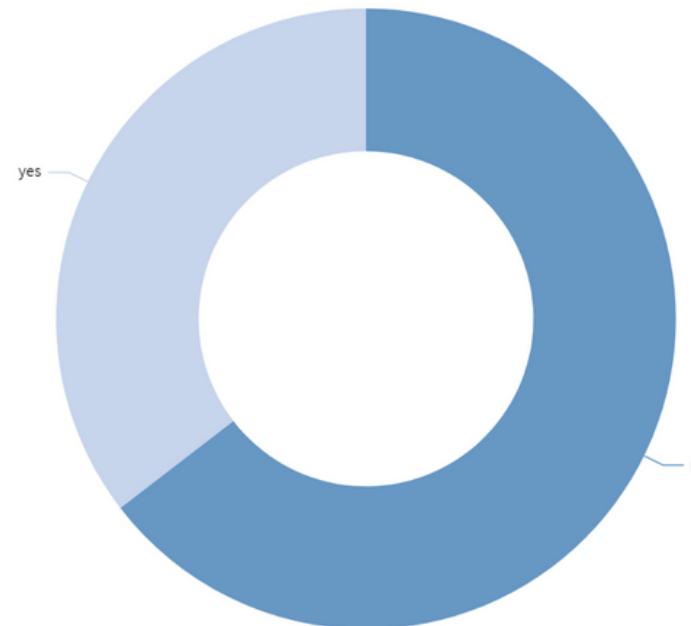


1803 records

Run on DSS



Churner Active Status Ratio



1803 records

Run on DSS



PART 5: BUSINESS IMPACT



Impact 1

Based on customer group, Active Savers has low churning ratio which is 6% only. Business may upslling or crosss-selling to Basic Account Holders to increase no. of product and turn them into Active Savers group.



Impact 2

Retention program to reduce churning rate in focus group which are High-Value Customers and Basic Account Holders



Impact 3

Cost saving and reduce opportunity loss from confusion & cost matrix (explaination next slide)

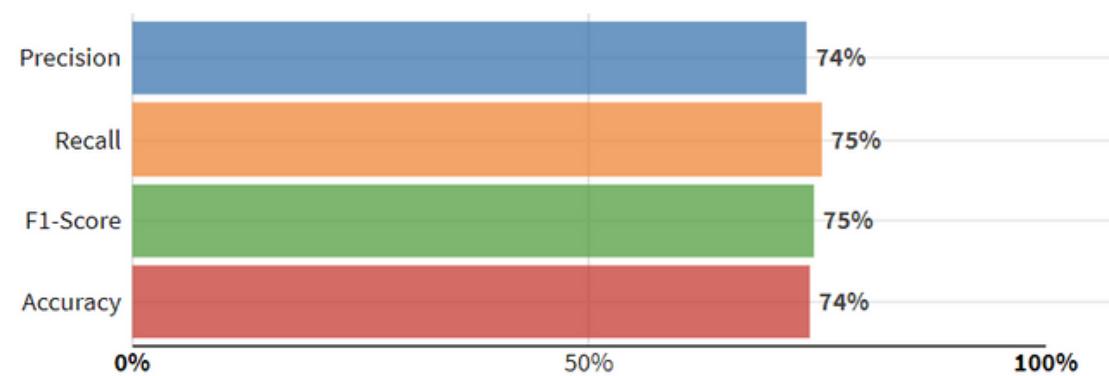
IMPACT 3 EXPLANATION

CONFUSION & COST METRIX

Confusion matrix

Display: Record count

	Predicted yes	Predicted no	Total
Actually yes	206	69	275
Actually no	76	204	280
Total	282	273	555



Cost matrix

If model predicts yes	and value is yes	the gain is -3	\times	206	=	-618.00
	but value is no	the gain is -3	\times	76	=	-228.00
Model predicts no	and value is no	the gain is 0	\times	204	=	0.00
	but value is yes	the gain is -10	\times	69	=	-690.00
Average gain per record					$-2.77 \times 555 = -1,536.00$	

Assumption

Using ML to predict churn

- Cost to maintain customers which model predict churn = yes (Cost = 3)
- Opportunity loss from losing customers = 10

in this case of using ML to predict churn all cost & opportunity loss occur 1,536.00

Not using ML to predict churn and believe all there are loyalty customers, opportunity loss occur = 275(churned customers) * 10(opportunity loss/churn) = 2,750.00

Conclusion if use ML to predict churn will reduce opportunity loss around $2,750.00 - 1,536.00 = 1,214.00$



*Thank
You*