CIS 5560

Professor J. Woo

PREDICTING AD CLICK FRAUD

MACHINE LEARNING PREDICTIVE ANALYSIS USING AZURE ML AND DATABRICKS SPARK ML

Group- F

Hai Anh Le

Neha Gupta

Maria Boldina

TABLE OF CONTENTS

INTRODUCTION

- Click Fraud happens at an overwhelming volume, resulting in misleading click data and wasted money
- Click Fraud occurs when a person, automated script or computer program imitates a legitimate user, clicking on an ad without having an actual interest in the target of the ad's link
- As the largest mobile market in the world, China suffers from huge volumes of fraudulent traffic
- TalkingData is China's largest independent big data service platform
  - covers over 70% of active mobile devices nationwide
  - handles 3 billion clicks per day
  - 90% of which are potentially fraudulent

The goal of our project is to predict whether a user will download an app after clicking on a mobile app ad

To better target the audience, to avoid fraudulent practices and save money

DATASET DETAILS

- Dataset name: TalkingData AdTracking Fraud Detection
- Dataset URL: https://www.kaggle.com/c/talkingdata-adtracking-fraud- detection/data
- Dataset contains 200 million clicks over 4 day period
- Dataset size:
  - Original dataset size: 7GB
  - Implemented Python code to reduce size to1GB (15%)
- Dataset format: .csv

DATASET DETAILS

Final Project - Two Class Decision Jungle › train_sample_15_pct - Copy (2).csv › dataset

rows 27175395 columns 8

view as

ip	app	device	os	channel	click_time	attributed_time	is_attributed
161007	3	1	13	379	2017-11-06T14:35:08		0
172522	3	1	25	379	2017-11-06T14:38:27		0
124979	3	1	18	379	2017-11-06T14:40:16		0
129614	3	1	20	379	2017-11-06T14:49:36		0
28739	3	1	13	379	2017-11-06T14:50:29		0
23550	3	1	13	379	2017-11-06T14:53:39		0
129614	3	1	19	379	2017-11-06T14:57:50		0
128855	3	1	13	379	2017-11-06T14:58:16		0
108942	3	1	19	379	2017-11-06T15:04:28		0
192796	3	1	19	379	2017-11-06T15:07:24		0
32450	3	1	17	379	2017-11-06T15:07:29		0
107899	6	1	13	459	2017-11-06T15:12:43		0
89489	3	1	13	379	2017-11-06T15:13:23		0
164537	58	1	19	120	2017-11-06T15:14:35		0
129614	3	1	15	379	2017-11-06T15:16:47		0
89489	3	1	13	379	2017-11-06T15:20:38		0
3653	3	1	18	379	2017-11-06T15:21:34		0
108942	3	1	13	379	2017-11-06T15:37:28		0
204158	35	1	13	21	2017-11-06T15:41:07	2017-11-07T08:17:19	1

DATASET DETAILS

Data fields

Each row of the training data contains a click record, with the following features.

Features

- `ip` : ip address of click.
- `app` : app id for marketing.
- `device` : device **type** id of user mobile phone (e.g., iphone 6 plus, iphone 7, huawei mate 7, etc.)
- `os` : os version id of user mobile phone
- `channel` : channel id of mobile ad publisher
- `click_time` : timestamp of click (UTC)
- `attributed_time` : if user download the app for after clicking an ad, this is the time of the app download

Label

- `is_attributed` : the target that is to be predicted, indicating the app was downloaded

TECHNICAL SPECIFICATIONS

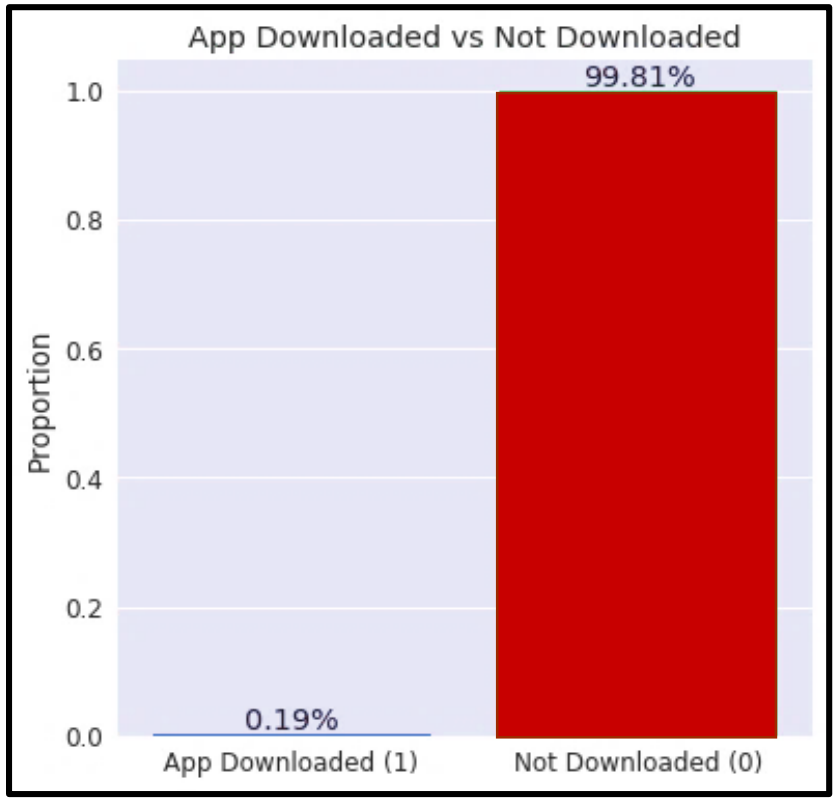
Azure ML Studio

- Free Workspace
- 10GB storage
- Single node
- Region: South Central US

databricks

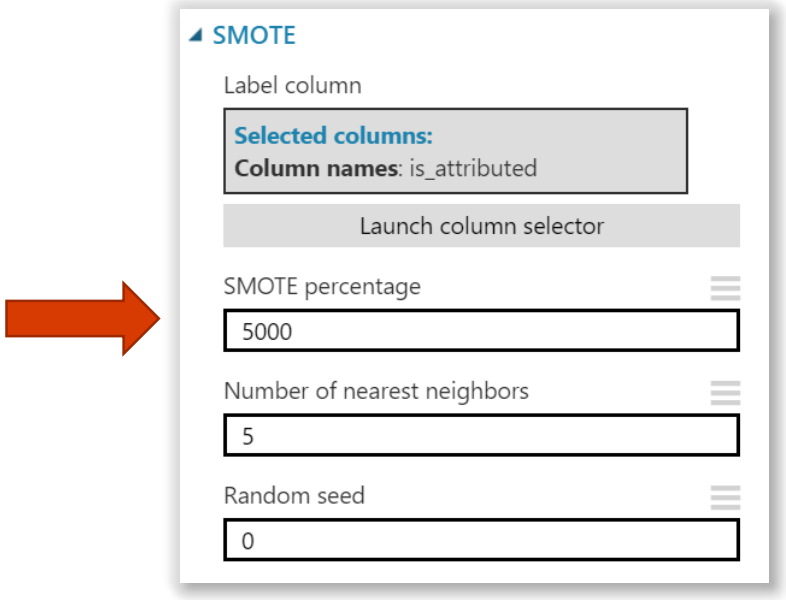
- DataBricks Subscription
- Cluster 4.0 (includes Apache Spark 2.3.0, Scala 2.11)
- 2 Spark Workers with total of 16 GB Memory and 4 Cores
- Python 2.7

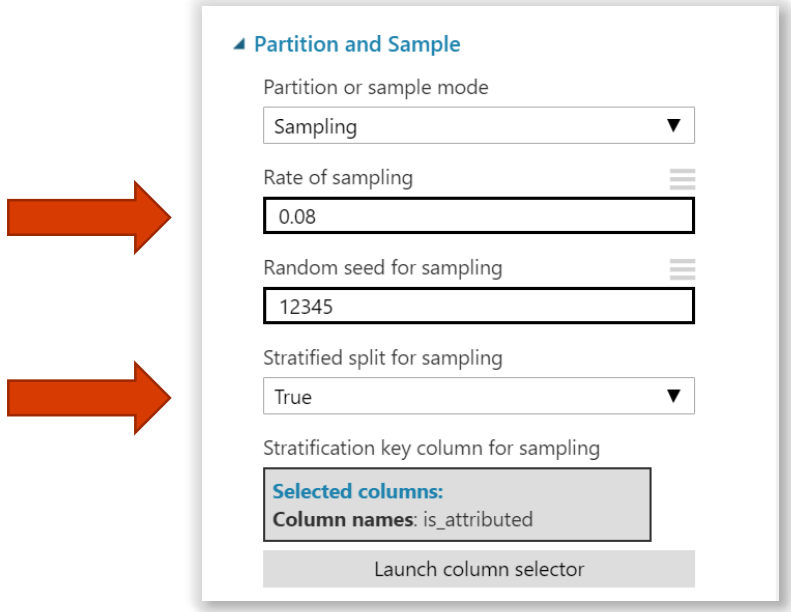
DATA PROCESSING AND CONSTRAINTS



- Unbalanced dataset with number of negative classes (0) far more outweighing the positive class (1)
- 0.19% App downloaded
- 99.81% App not downloaded
- 1GB dataset still too large for Azure ML → sampling was used

WORKING WITH UNBALANCED DATA





- **SMOTE:** Synthetic Minority Over Sampling Technique takes a subset of data from the minority class and creates new synthetic similar instances
- Helps balance data & avoid overfitting
- Increased percent of minority class (1) from 0.19% to 11%

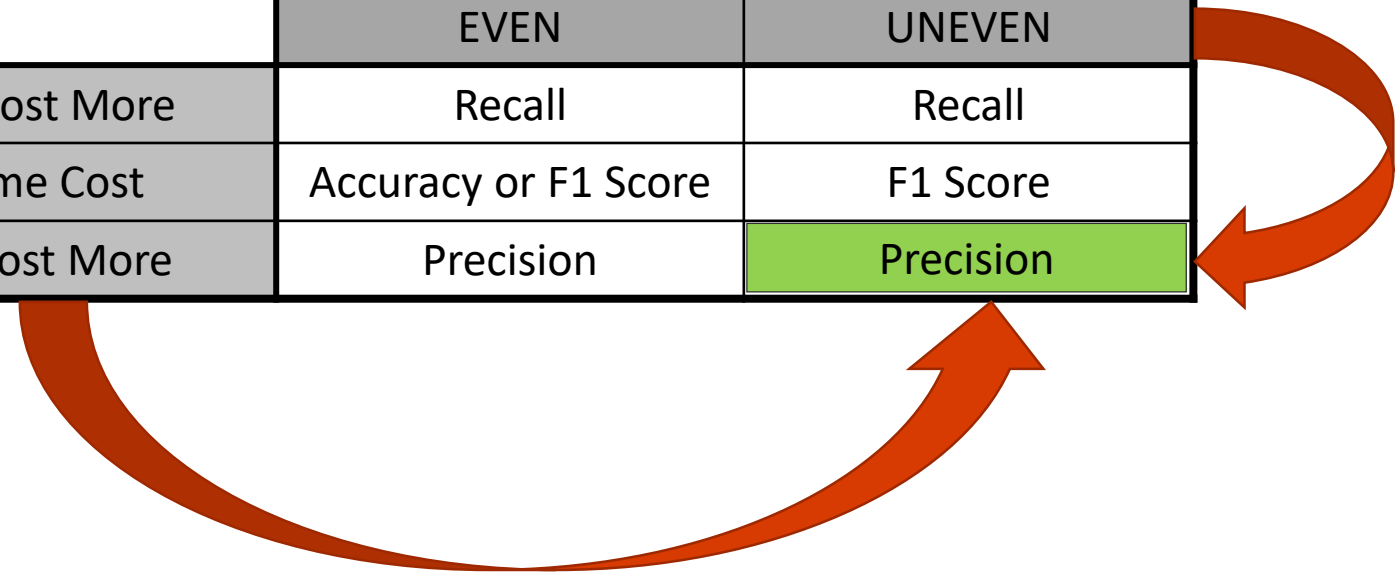
- **Stratified Split** ensures that the output dataset contains a representative sample of the values in the selected column
- Ensures that the random sample does not contain all rows with just 0s
- 8% sample used = 80 MB

SELECTING ALGORITHMS FOR AZURE ML

- **Two-Class Classification:**
  - Binary or binomial classification is the task of classifying the elements of a given set into two groups
  - predicting a category or class, either downloaded (1) or not downloaded (0)
- Decision trees often perform well on imbalanced datasets because their hierarchical structure allows them to learn signals from both classes.
- Tree ensembles almost always outperform singular decision trees ➔
  - Algorithm #1: Two-class Decision Jungle
  - Algorithm #2: Two-class Decision Forest

SELECTING PERFORMANCE METRICS

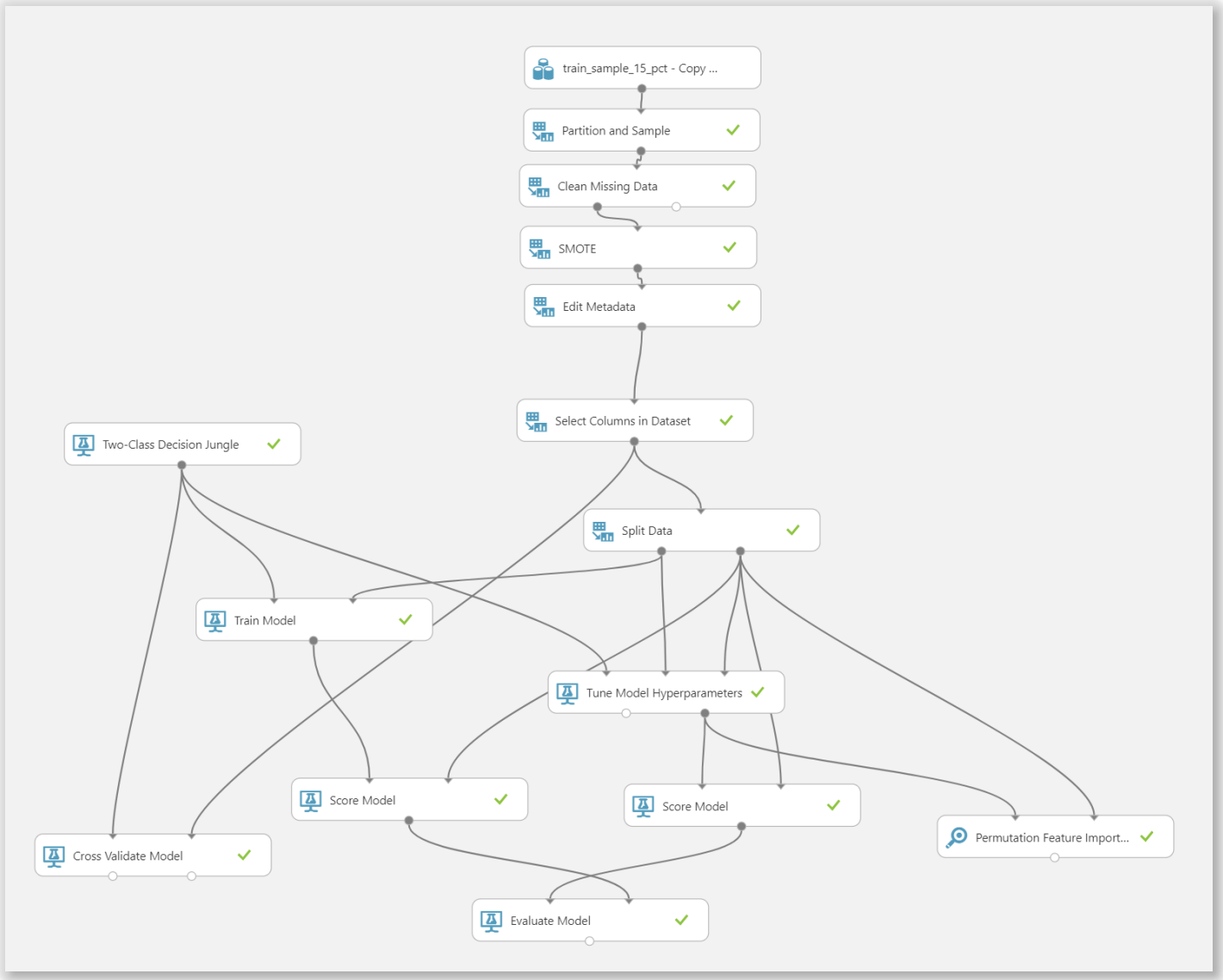
		CLASS DISTRIBUTION	
		EVEN	UNEVEN
COST	FN Cost More	Recall	Recall
	Same Cost	Accuracy or F1 Score	F1 Score
	FP Cost More	Precision	Precision



False Positives indicate the model predicted an app was downloaded when in fact it wasn't
We want to minimize the FP ➔ save $$$

AZURE ML MODEL #1: TWO-CLASS DECISION JUNGLE

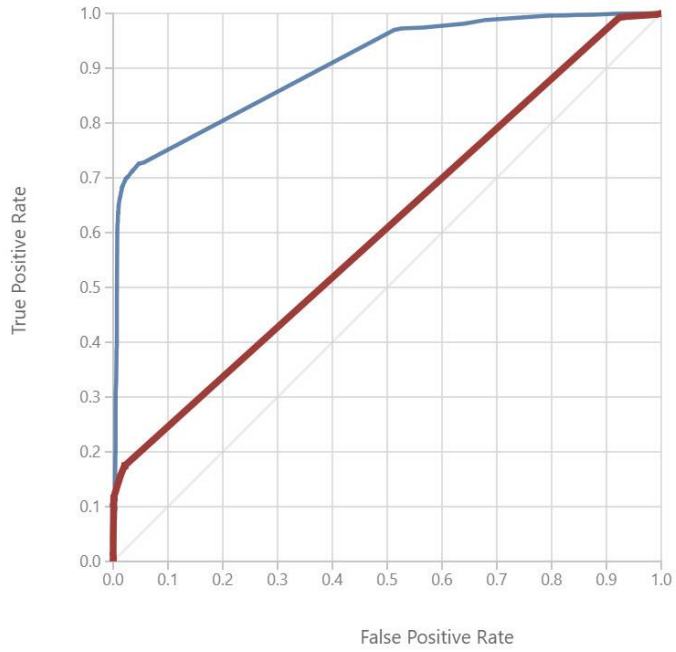
- 8% Sample
- SMOTE 5000%
- 70:30 Split Train/Test
- Cross-Validation
- Tune Model Hyperparameters
- Features used: all 7



AZURE ML MODEL #1: RESULTS

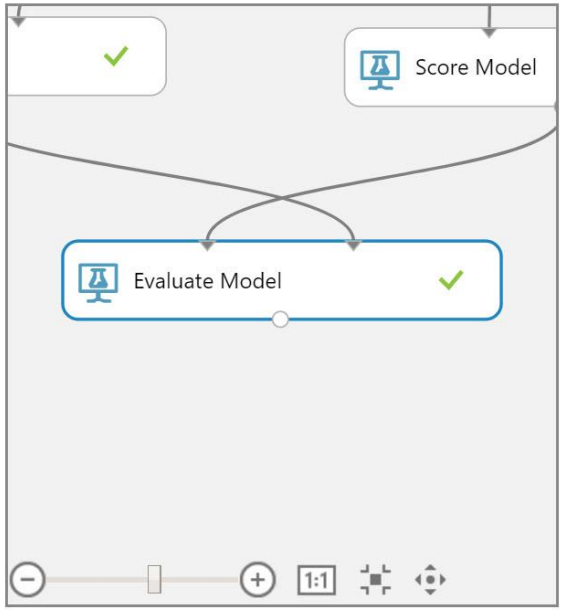
Final Project - Two Class Decision Jungle › Evaluate Model › Evaluation results

ROC PRECISION/RECALL LIFT



Scored dataset

Scored dataset to compare



True Positive	False Negative	Accuracy	Precision	Threshold	AUC
2	52339	0.886	1.000	0.5	0.606

False Positive	True Negative	Recall	F1 Score
0	406605	0.000	0.000

Positive Label	Negative Label
1	0

AUC = 0.606

AZURE ML MODEL #1: Tune Model Hyperparameters

Final Project - Two Class Decision Jungle › Tune Model

Binary Gemini Decision Jungle Classifier

Settings

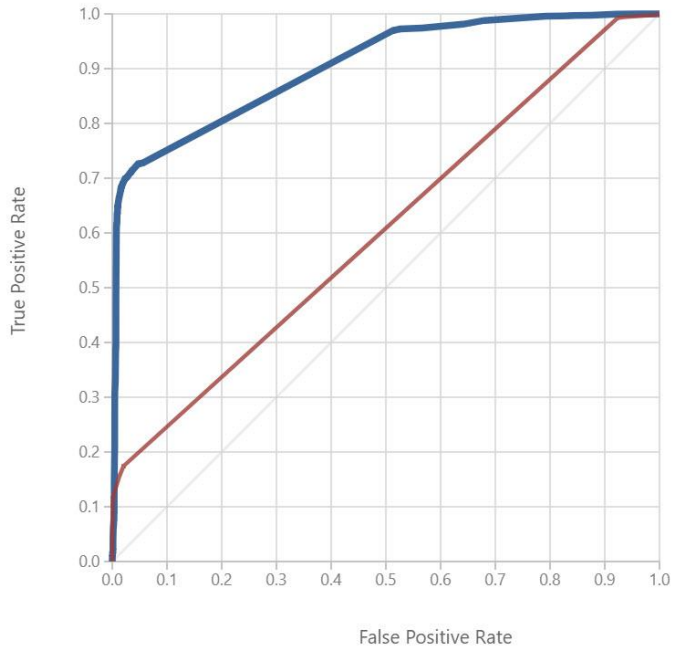
Setting	Value
Optimization Step Count	9937
Max Width	8
Max Depth	67
Ensemble Element Count	12
Class Count	2
Resampling Method	Bagging
Random Number Seed	5
Allow Unknown Levels	True

With Tune Hyperparameters

Without Tune Hyperparameters

Final Project - Two Class Decision Jungle › Evaluate Model › Evaluation results

ROC PRECISION/RECALL LIFT



Scored dataset

Scored dataset to compare

True Positive	False Negative	Accuracy	Precision	Threshold	AUC
35	52306	0.886	1.000	0.5	0.905

False Positive	True Negative	Recall	F1 Score
0	406605	0.001	0.001

Positive Label	Negative Label
1	0

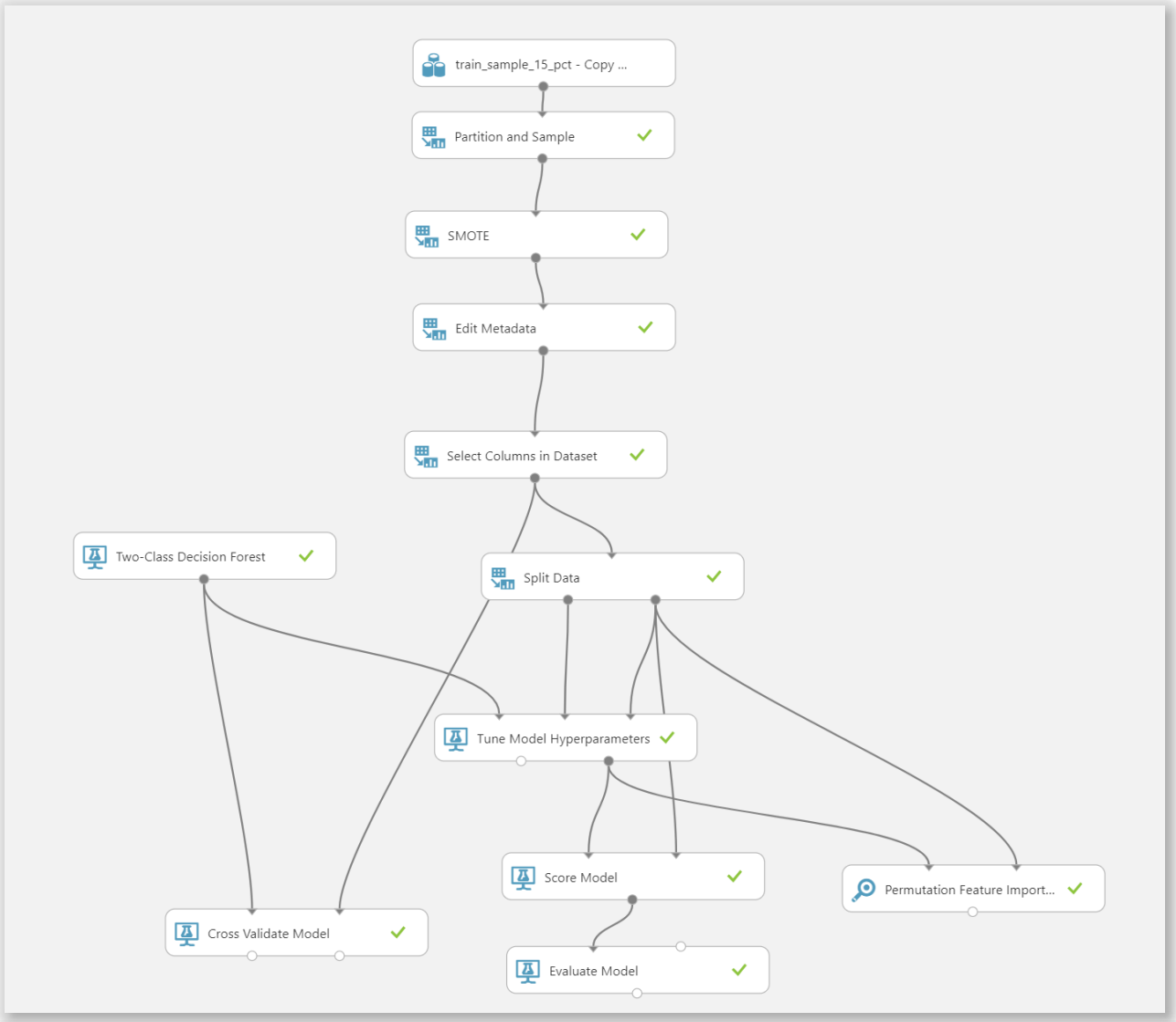
AUC = 0.905 vs 0.606

Precision = 1.0

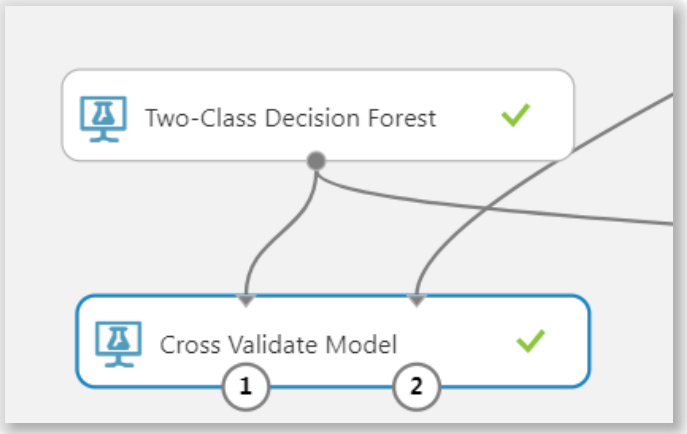
TP = 35, FP = 0

AZURE ML MODEL #2: TWO-CLASS DECISION FOREST

- 8% Sample
- SMOTE 5000%
- 70:30 Split Train/Test
- Cross-Validation
- Tune Model Hyperparameters
- Permutation Feature Importance



AZURE ML MODEL #2: Cross Validation

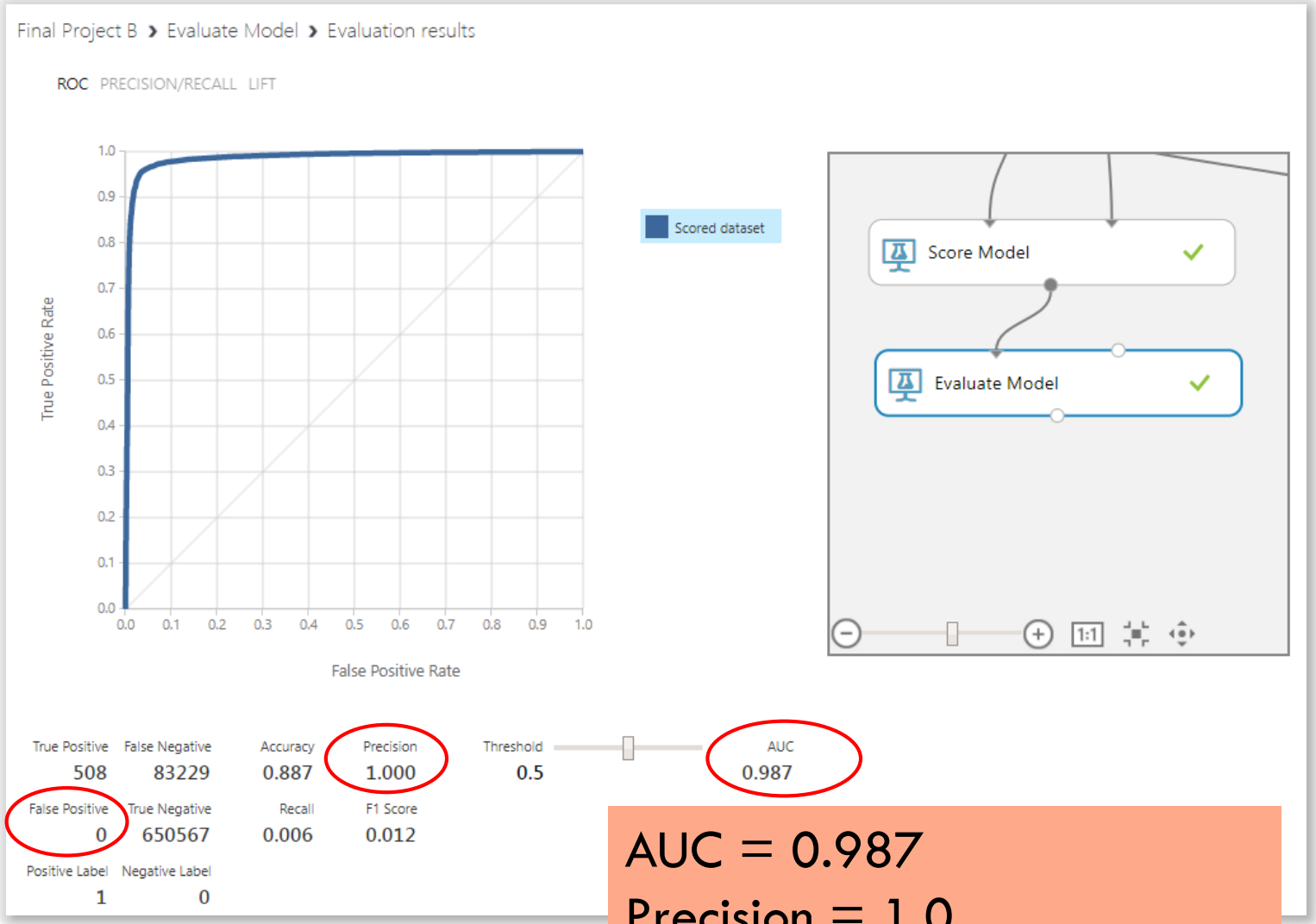


Final Project - Two-Class Decision Forest › Cross Validate Model › Evaluation results by fold

rows 12 columns 10

	Fold Number	Number of examples in fold	Model	Accuracy	Precision	Recall	F-Score	AUC	Average Log Loss	Training Log Loss
view as										
	0	244768	Microsoft.Analytics.Modules.Gemini.Dll.BinaryGeminiDecisionForestClassifier	0.885496	0	0	0	0.952675	0.324843	8.708544
	1	244768	Microsoft.Analytics.Modules.Gemini.Dll.BinaryGeminiDecisionForestClassifier	0.886578	0	0	0	0.950305	0.320513	9.359797
	2	244768	Microsoft.Analytics.Modules.Gemini.Dll.BinaryGeminiDecisionForestClassifier	0.886738	0	0	0	0.960451	0.316589	10.386319
	3	244768	Microsoft.Analytics.Modules.Gemini.Dll.BinaryGeminiDecisionForestClassifier	0.885467	0	0	0	0.951249	0.325006	8.677798
	4	244768	Microsoft.Analytics.Modules.Gemini.Dll.BinaryGeminiDecisionForestClassifier	0.884981	0	0	0	0.941921	0.32598	8.658941
	5	244768	Microsoft.Analytics.Modules.Gemini.Dll.BinaryGeminiDecisionForestClassifier	0.886141	0	0	0	0.957163	0.320884	9.484623

AZURE ML MODEL #2: RESULTS



AUC = 0.987
Precision = 1.0
TP = 508, FP = 0

AZURE ML MODEL #2: Permutation Feature Importance

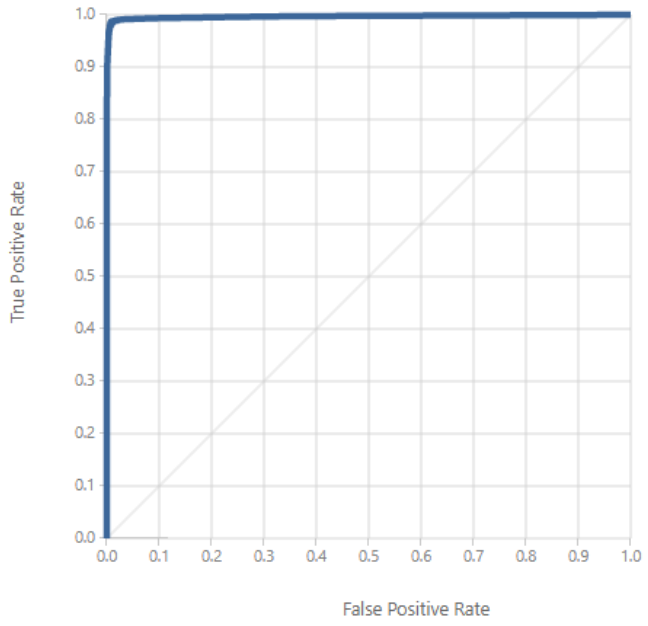
rows 7 columns 2

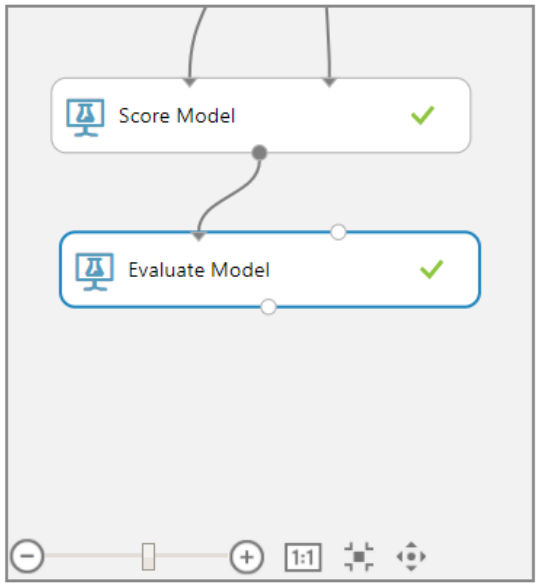
Feature	Score
channel	0.051948
app	0.018519
ip	0
device	0
os	0
click_time	0
attributed_time	0

- Features used:
  - channel
  - app

Running with Permutation decision › Evaluate Model › Evaluation results

ROC PRECISION/RECALL LIFT





True Positive	False Negative	Accuracy	Precision	Threshold	AUC
50507	1834	0.992	0.968	0.5	0.997

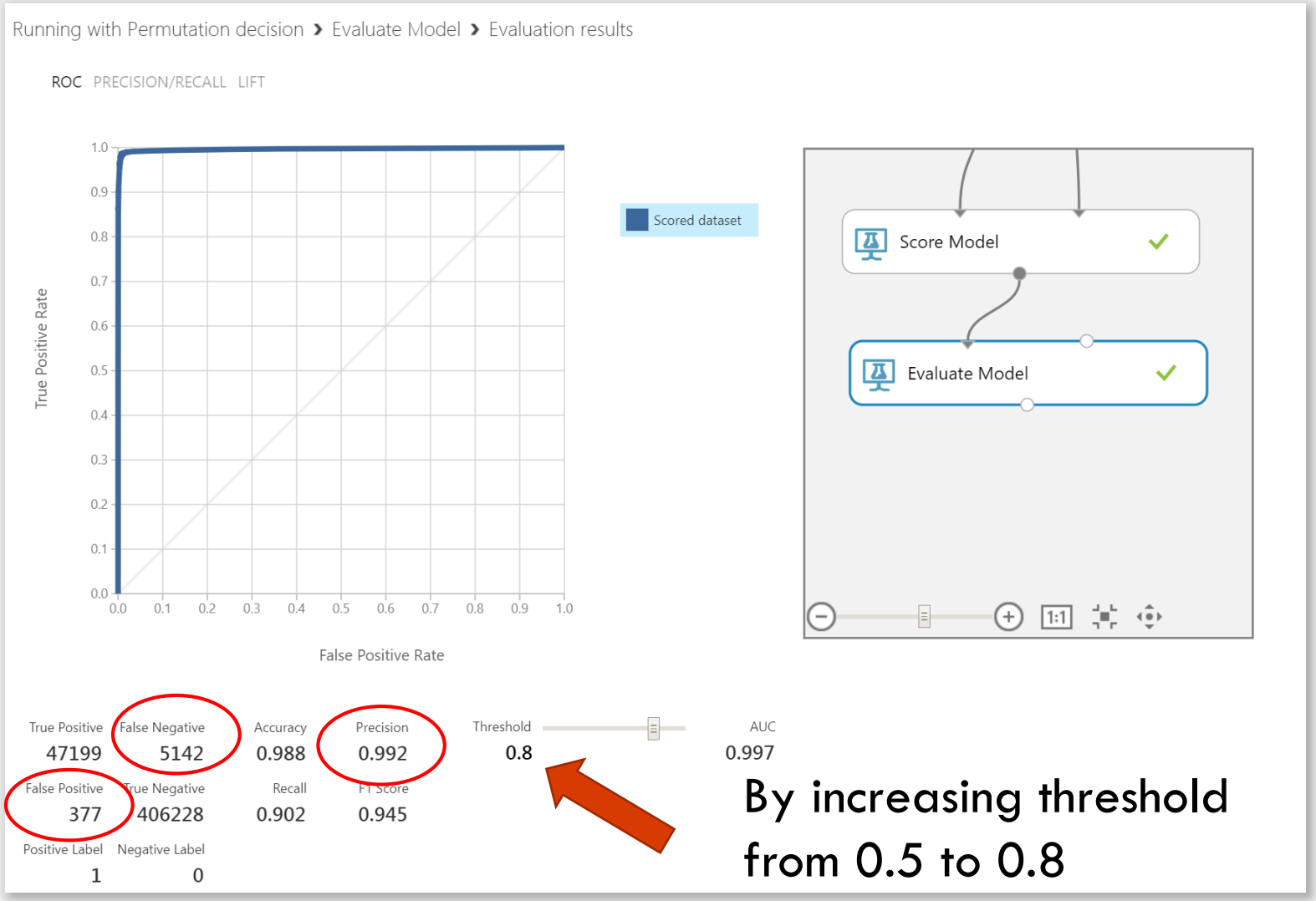
False Positive	True Negative	Recall	F1 Score
1659	404946	0.965	0.967

Positive Label	Negative Label
1	0

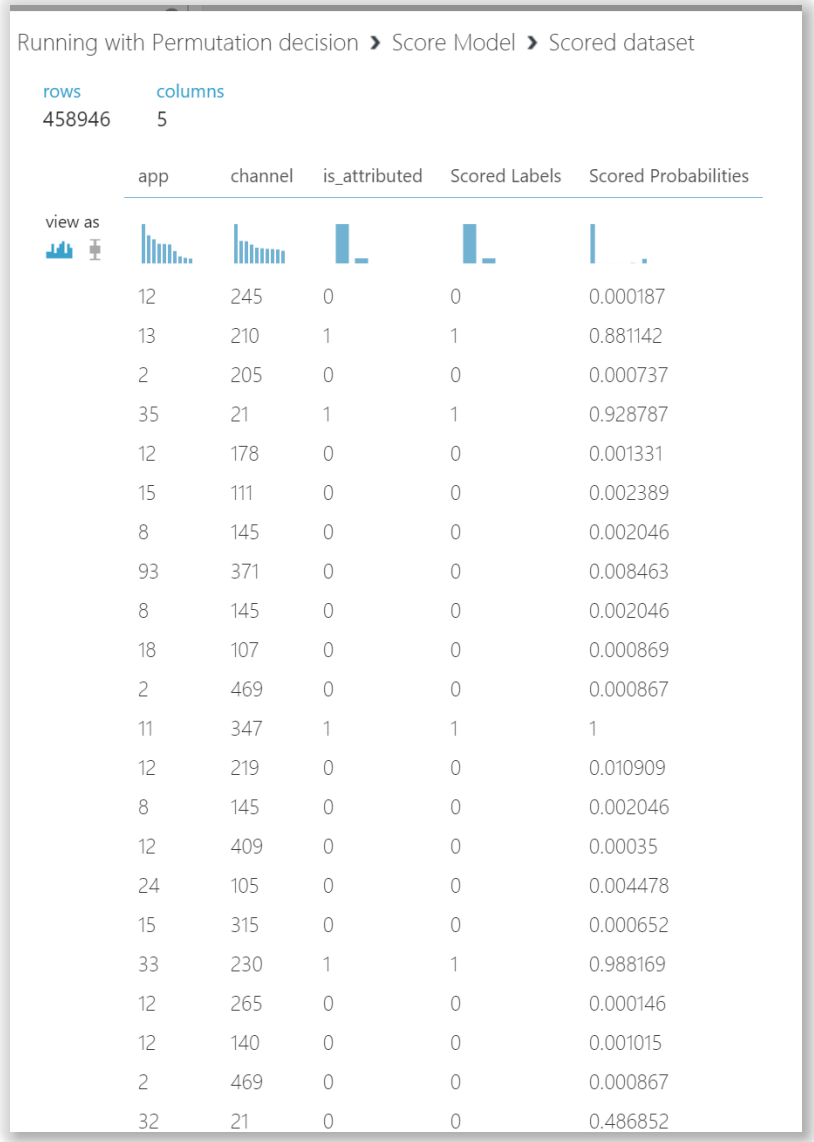
AUC = 0.997 vs 0.987
Precision = 0.968 vs 1.0

AZURE ML MODEL #2: Improving Precision

Precision increased to 0.992
FP decreased from 1,659 to 377
FN increased from 1,834 to 5,142



Permutation Feature Importance Results



Running with Permutation decision › Score Model › Scored dataset

rows 458946 columns 5

view as

app	channel	is_attributed	Scored Labels	Scored Probabilities
12	245	0	0	0.000187
13	210	1	1	0.881142
2	205	0	0	0.000737
35	21	1	1	0.928787
12	178	0	0	0.001331
15	111	0	0	0.002389
8	145	0	0	0.002046
93	371	0	0	0.008463
8	145	0	0	0.002046
18	107	0	0	0.000869
2	469	0	0	0.000867
11	347	1	1	1
12	219	0	0	0.010909
8	145	0	0	0.002046
12	409	0	0	0.00035
24	105	0	0	0.004478
15	315	0	0	0.000652
33	230	1	1	0.988169
12	265	0	0	0.000146
12	140	0	0	0.001015
2	469	0	0	0.000867
32	21	0	0	0.486852

AZURE ML RESULTS COMPARISON

	TWO-CLASS DECISION JUNGLE	TWO-CLASS DECISION FOREST
AUC	0.905	0.997
PRECISION	1.0	0.992
RECALL	0.001	0.902
TP	35	47,199
FP	0	377
TN	52,306	406,228
FN	406,605	5,142

Two-class Decision Forest is the best model!

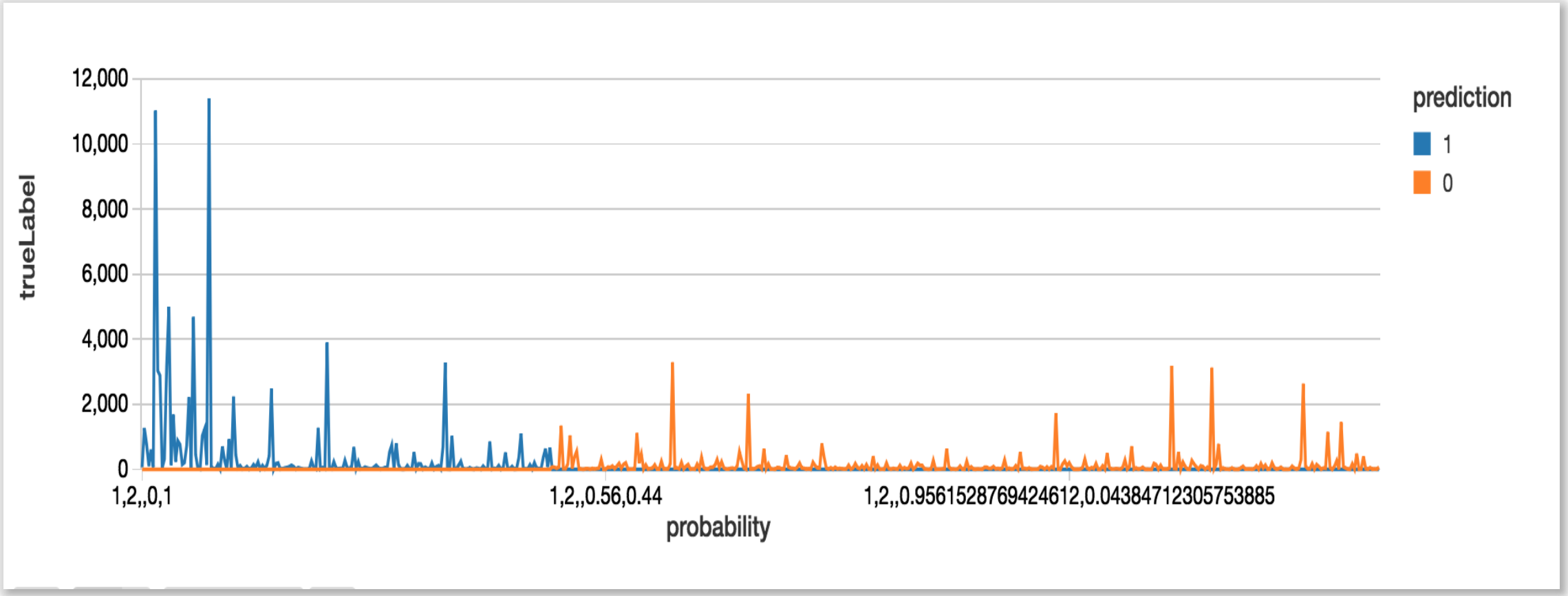
SELECTING ALGORITHMS FOR DATABRICKS SPARK ML

- **Binary Classification:**
  - Predicting a category or class, either downloaded (1) or not downloaded (0)
- Binary or binomial classification is the task of classifying the elements of a given set into two groups (predicting which group each one belongs to) on the basis of a classification rule.
- Algorithm #1: Decision Tree Classifier
- Algorithm #2: Random Forest Classifier

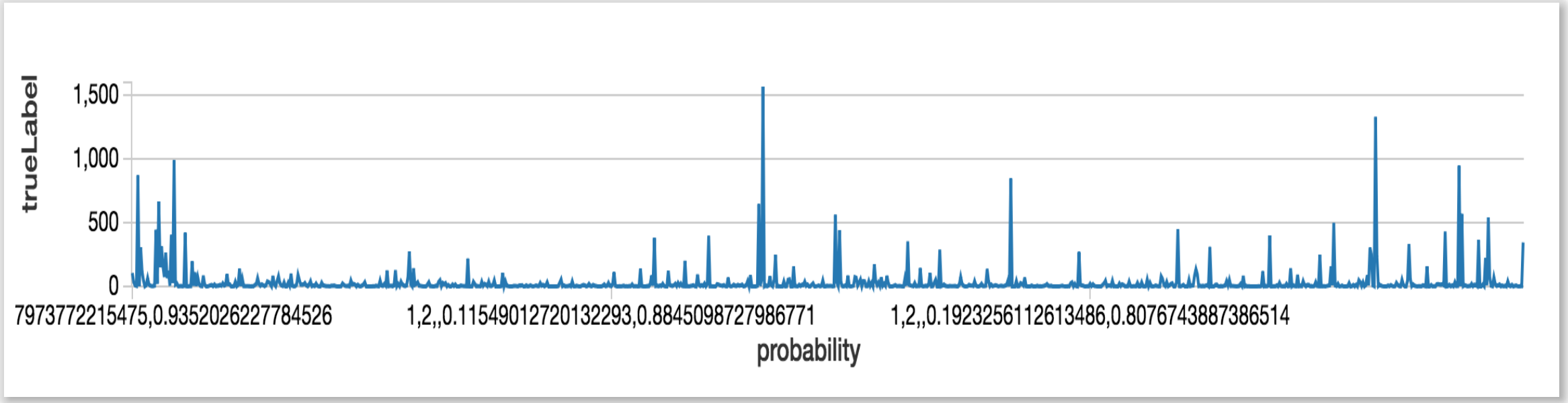
DATABRICKS SPARK ML COMBINATION OF FEATURES

Feature 1: extract day of the week and hour of the day from the click time
Feature 2: group clicks by combination of (Ip, Day_of_week_number and Hour)
Feature 3: group clicks by combination of (Ip, App, Operating System, Day_of_week_number and Hour)
Feature 4: group clicks by combination of (App, Day_of_week_number and Hour)
Feature 5: group clicks by combination of (Ip, App, Device and Operating System)
Feature 6: group clicks by combination of (Ip, Device and Operating System)

DATABRICKS MODEL #1: Decision Tree Classifier TrueLabel and Prediction

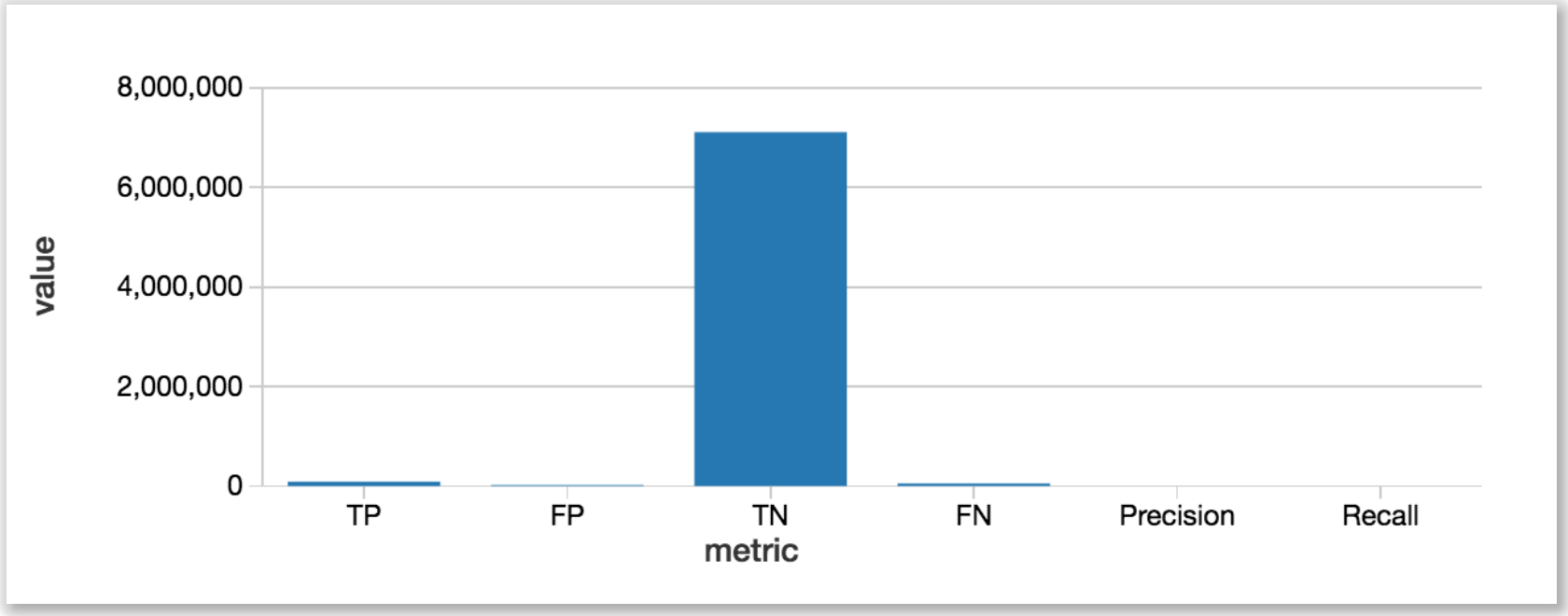


DATABRICKS MODEL #2: Random Forest Classifier TrueLabel and Prediction



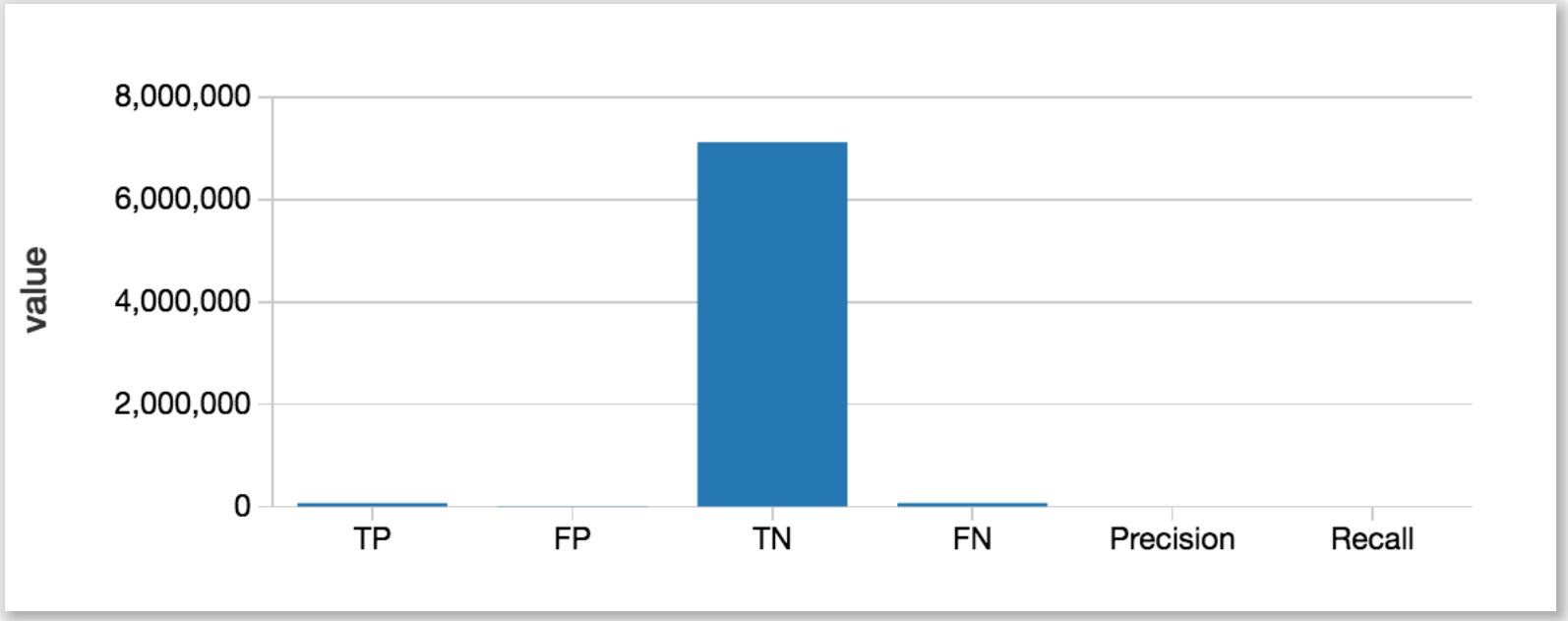
DATABRICKS MODEL #1: Decision Tree Classifier Confusion Matrix

metric	value
TP	86683.0
FP	18727.0
TN	7112961.0
FN	50074.0
Precision	0.8223413338392942
Recall	0.6338468963197497



DATABRICKS MODEL #2: Random Forest Classifier Confusion Matrix

metric	value
TP	67726.0
FP	9408.0
TN	7122280.0
FN	69031.0
Precision	0.8780304405320611
Recall	0.49522876342710065



DATABRICKS SPARK ML RESULTS COMPARISON

	Decision Tree Classifier	Random Forest Classifier
AUC	0.815	0.746
PRECISION	0.822	0.878
RECALL	0.633	0.495
TP	86,683	67,726
FP	18,727	9,408
TN	7,112,961	7,122,280
FN	50,074	69,031
RMSE	0.0972	0.1038

Decision Tree Classifier is the best model!

AZURE ML AND DATABRICKS RESULTS COMPARISON

	TWO-CLASS DECISION JUNGLE	TWO-CLASS DECISION FOREST	DECISION TREE CLASSIFIER	RANDOM FOREST CLASSIFIER
AUC	0.905	0.997	0.815	0.746
PRECISION	1.0	0.992	0.822	0.878
RECALL	0.001	0.902	0.633	0.495
TP	35	47,199	86,683	67,726
FP	0	377	18,727	9,408
TN	52,306	406,228	7,112,961	7,122,280
FN	406,605	5,142	50,074	69,031
Run Time	2 hrs	2-3 hrs	22 mins	50 mins

Azure ML Two-class Decision Forest is the best model!

REFERENCES

- Github link: https://github.com/ngupta8
- https://www.kaggle.com/c/talkingdata-adtracking-fraud-detection/data
- https://blogs.msdn.microsoft.com/andreasderuiter/2015/02/09/performance-measures-in-azure-ml-accuracy-precision-recall-and-f1-score/
- https://docs.microsoft.com/en-us/azure/machine-learning/studio/
- https://docs.microsoft.com/en-us/azure/machine-learning/studio/algorithm-choice
- https://docs.databricks.com/spark/latest/mllib/binary-classification-mllib-pipelines.html

THANK YOU!