HUIZHE (SUNNY) ZHU

NEURAL NETWORK ON CREDIT CARD FRAUD DETECTION

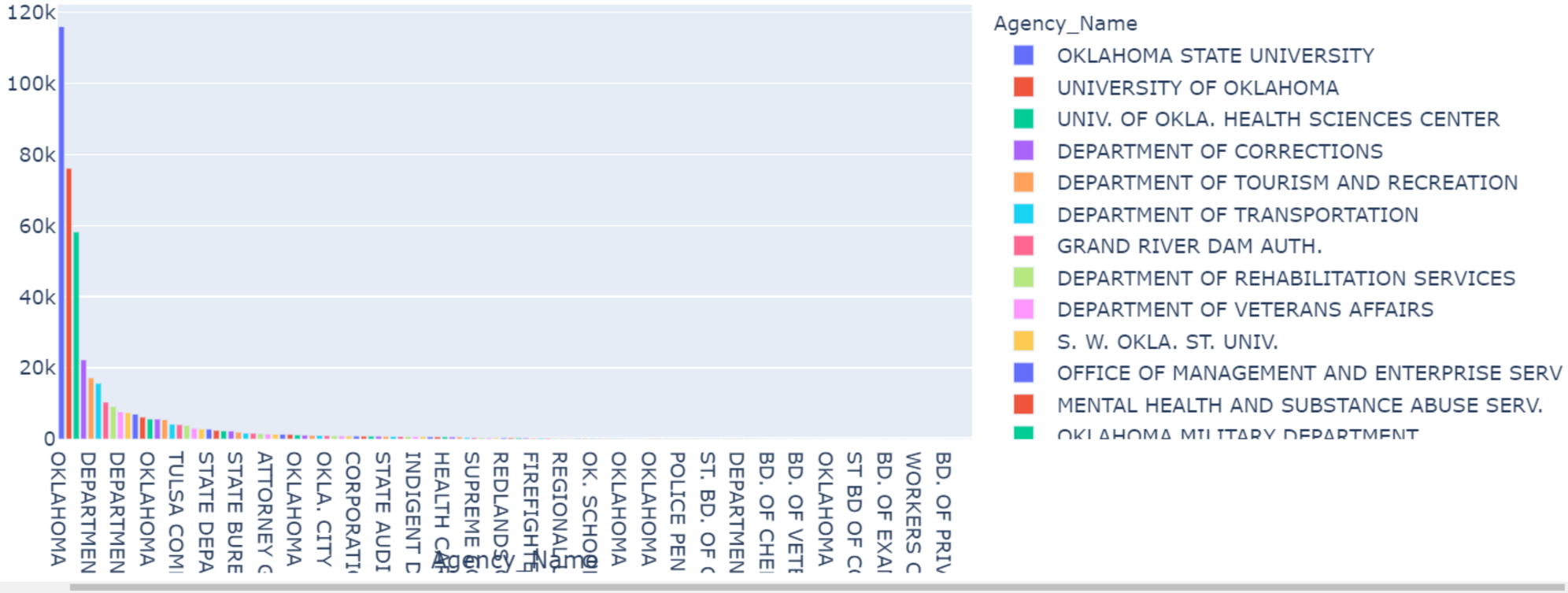
TAKE-AWAYS AND INSIGHTS WEEK 7

1.EDA

	Year-Month	Agency Number	Agency Name	Cardholder Last Name	Cardholder First Initial	Description	Amount	Vendor	Transaction Date	Posted Date	Merchant Category Code (MCC)
0	201307	1000	OKLAHOMA STATE UNIVERSITY	Mason	C	GENERAL PURCHASE	890.00	NACAS	7/30/2013 0:00	7/31/2013 0:00	CHARITABLE AND SOCIAL SERVICE ORGANIZATIONS
1	201307	1000	OKLAHOMA STATE UNIVERSITY	Mason	C	ROOM CHARGES	368.96	SHERATON HOTEL	7/30/2013 0:00	7/31/2013 0:00	SHERATON

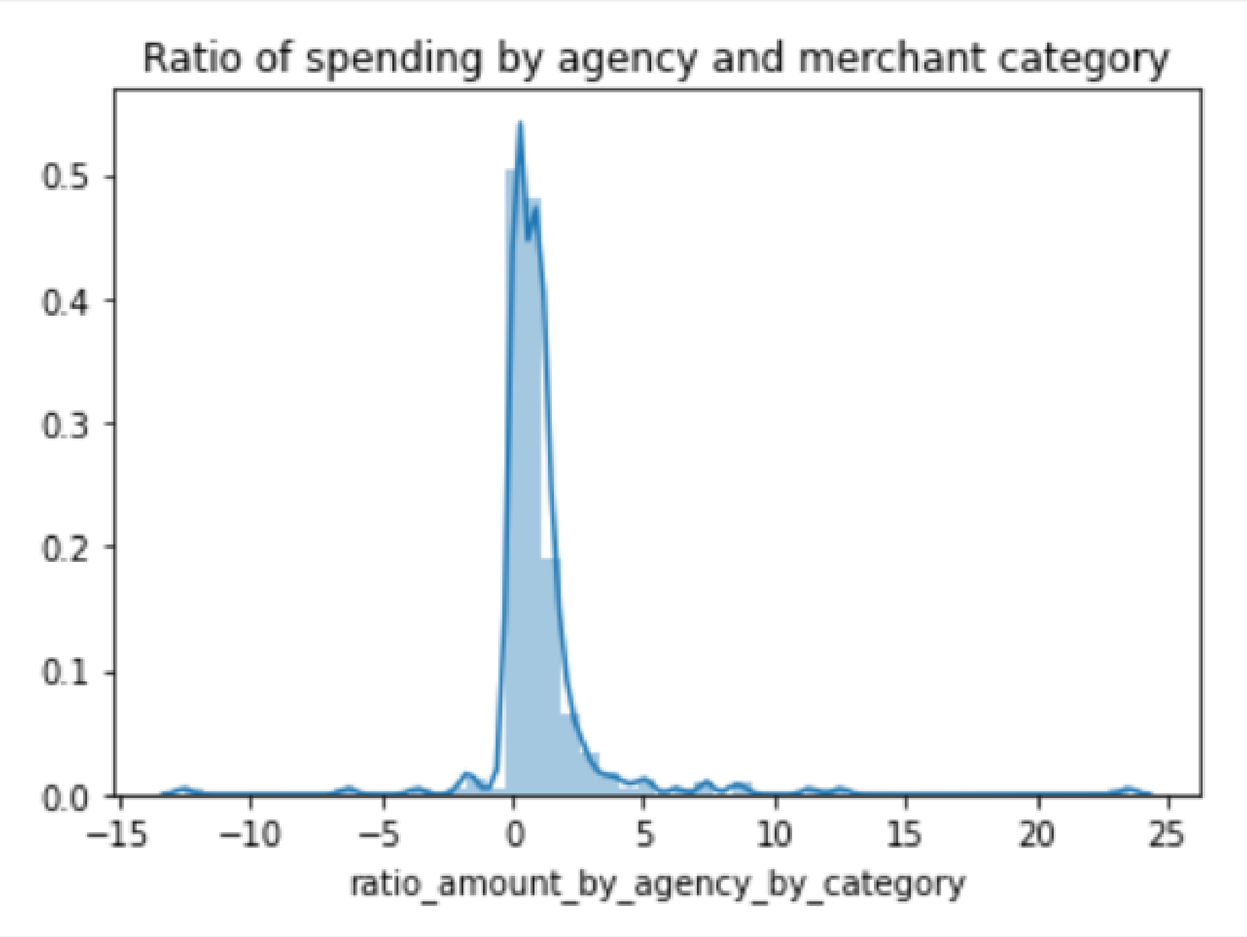
AGENCY LEVEL

CARDHOLDER LEVEL



2. CONSTRUCT FEATURES

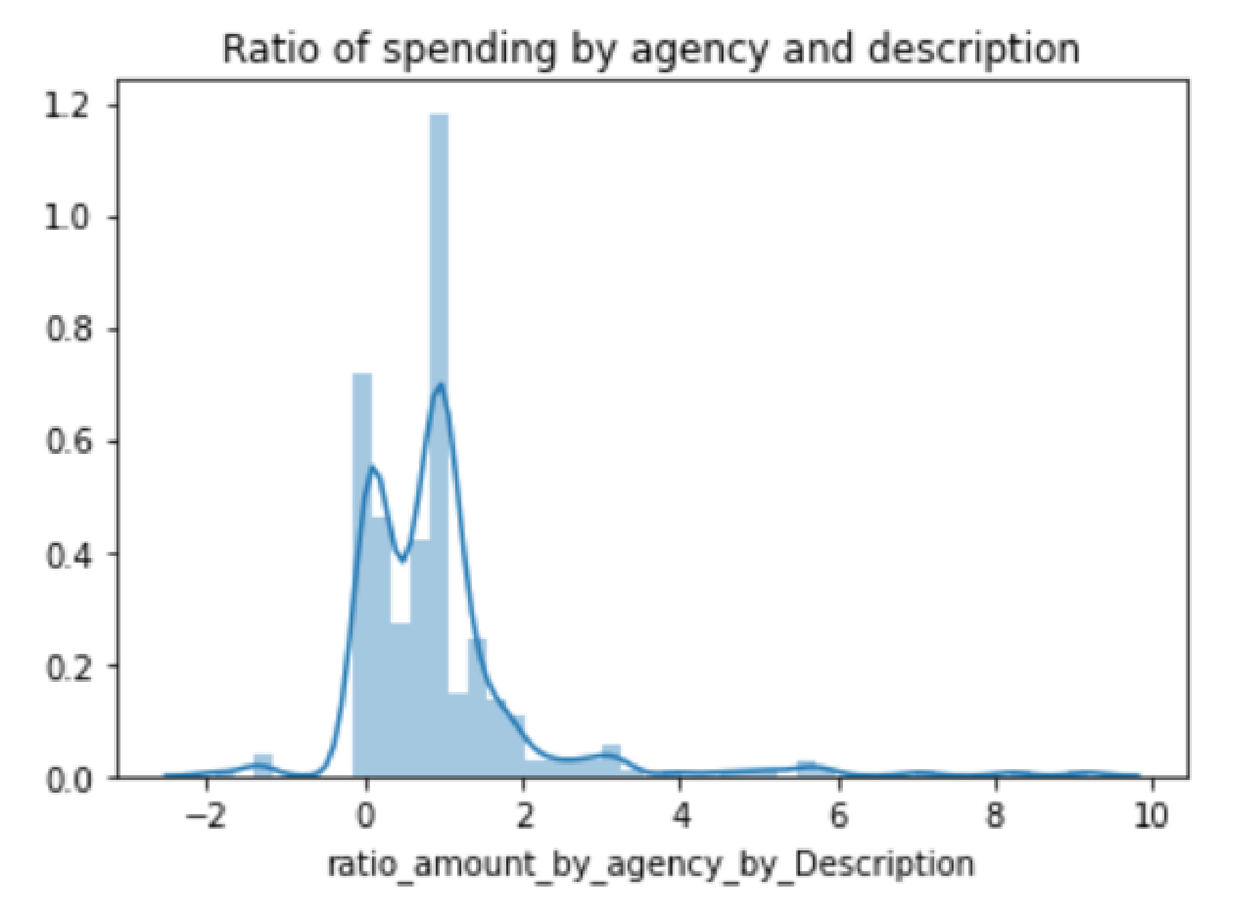
FEATURE 1: RATIO OF AMOUNT SPEND AND AVERAGE BY AGENCY AND MERCHANT CATEGORY



- Higher ratio means more spending compared with the average level
- negative values are refund
- inf values: Denominator = 0 , average spending in that category is 0. never purchased in that category before. replace with 1.

2. CONSTRUCT FEATURES

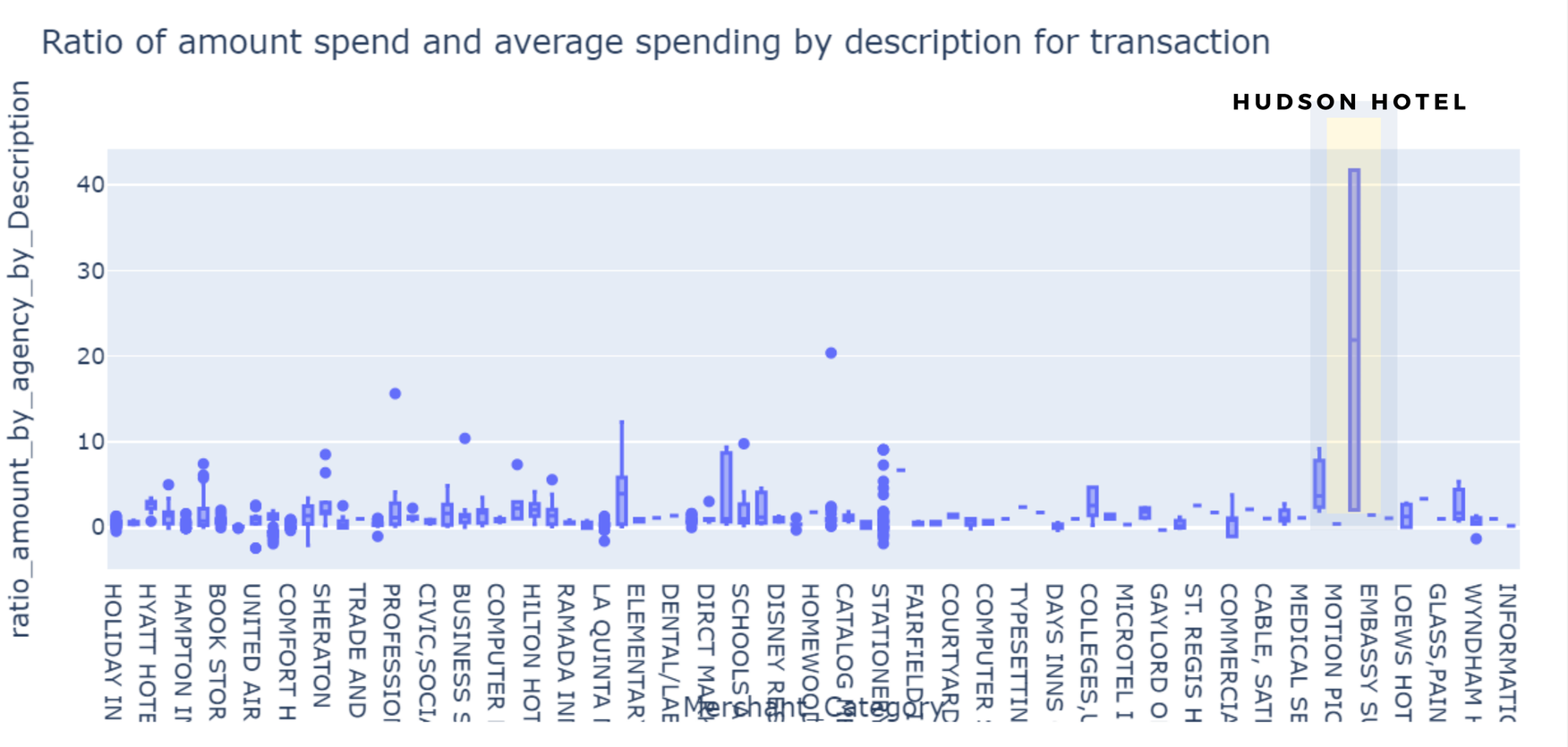
FEATURE 2: RATIO OF AMOUNT SPEND AND AVERAGE BY AGENCY AND DESCRIPTION



- More variance compared with feature 1
- Higher ratio means more spending compared with the average level
- negative values are refund
- inf values: Denominator = 0 , average spending in that category is 0. never purchased in that category before. replace with 1.

2. CONSTRUCT FEATURES

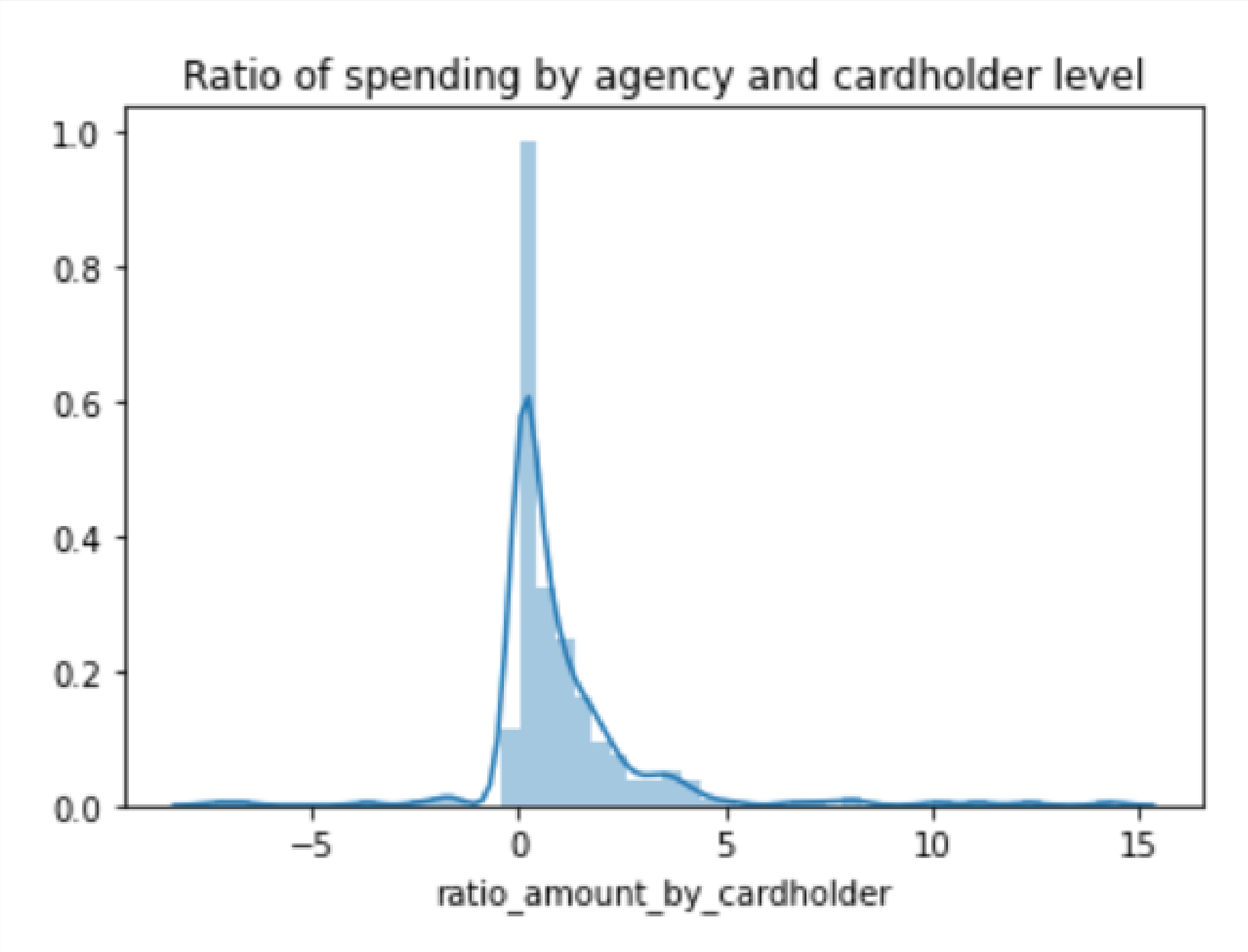
(CONT) FEATURE 2: RATIO OF AMOUNT SPEND BY AGENCY AND DESCRIPTION



- Select an agency, -> check the distribution of ratio among purchased with different descriptions
- Hudson Hotel has largest variance and highest ratio of average spending

2. CONSTRUCT FEATURES

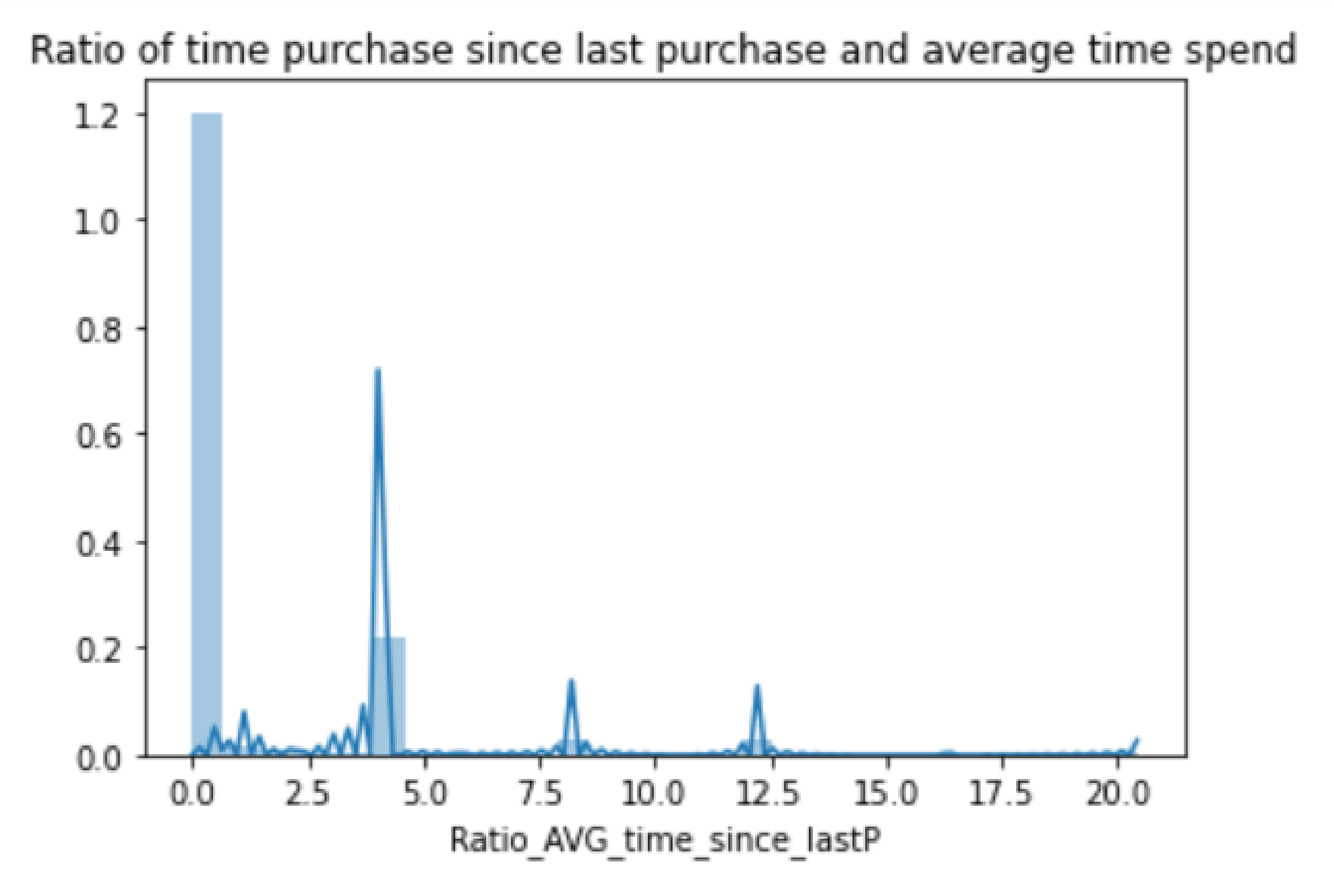
FEATURE 3: RATIO OF AMOUNT SPEND AND AVERAGE SPENDING BY CARD HOLDER



- This feature is based on card holder level, to check if one transaction is significantly higher than what the card holder usually spend
- the variance is similar to feature 1

2. CONSTRUCT FEATURES

FEATURE 4: RATIO OF TIME SINCE LAST TRANSACTION AND AVERAGE TIME SPEND



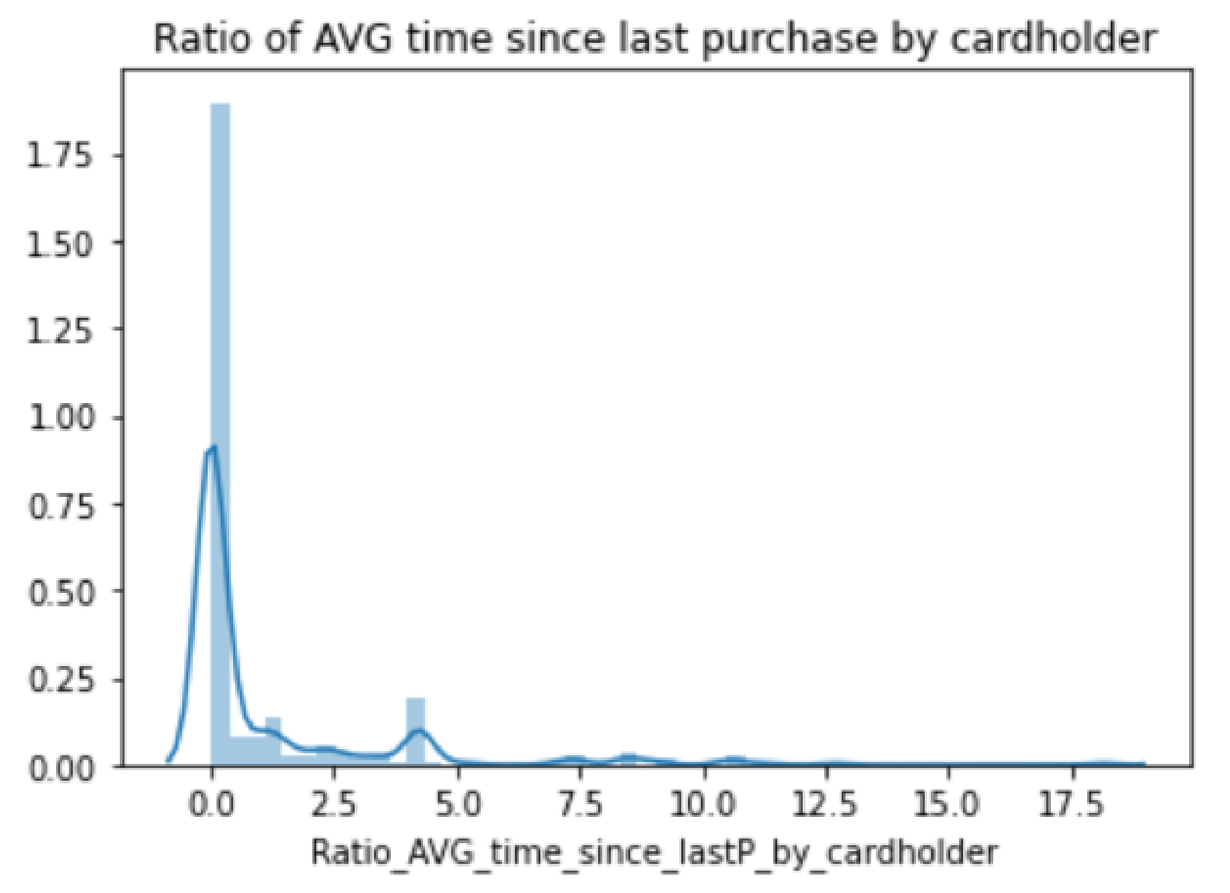
- Create 'time' variable: time since last purchase
- In the first record in an agency, we need to assign the value to 0, otherwise it will be calculated based on the agency 1 row above it, replace it with 0

Agency_Number	Year_Month	Agency_Name	time
1000	201310	OKLAHOMA STATE UNIVERSITY	-428 days
2000	201307	OKLAHOMA ACCOUNTANCY BOARD	-362 days
2200	201309	OKLAHOMA ABSTRACTORS BOARD	-300 days
2500	201307	OKLAHOMA MILITARY DEPARTMENT	-366 days
3900	201307	BOLL WEEVIL ERADICATION ORG.	-213 days

- why inf and NaN value exist: the average time since last transaction is 0. We replace the it with 1.

2. CONSTRUCT FEATURES

FEATURE 5: RATIO OF TIME SINCE LAST TRANSACTION AND AVERAGE TIME BY CARDHOLDER



- This feature is based on card holder level
- The larger ratio, means this transaction takes more time than the card holder's average spending frequency.

3. AUTOENCODER: INTRODUCTION

- Autoencoder uses the same data as both the input values and output values. The goal is to get the middle layer which reduces noises, it can also be considered as a method of dimension reduction.
- By identifying the main pattern in the middle layer, the outliers are revealed.
- Autoencoder uses non-linear transformation (PCA: Linear), more efficient. If there are too many hidden layers or too many neutrons, the model tends to overfit; otherwise the model tends to underfit.

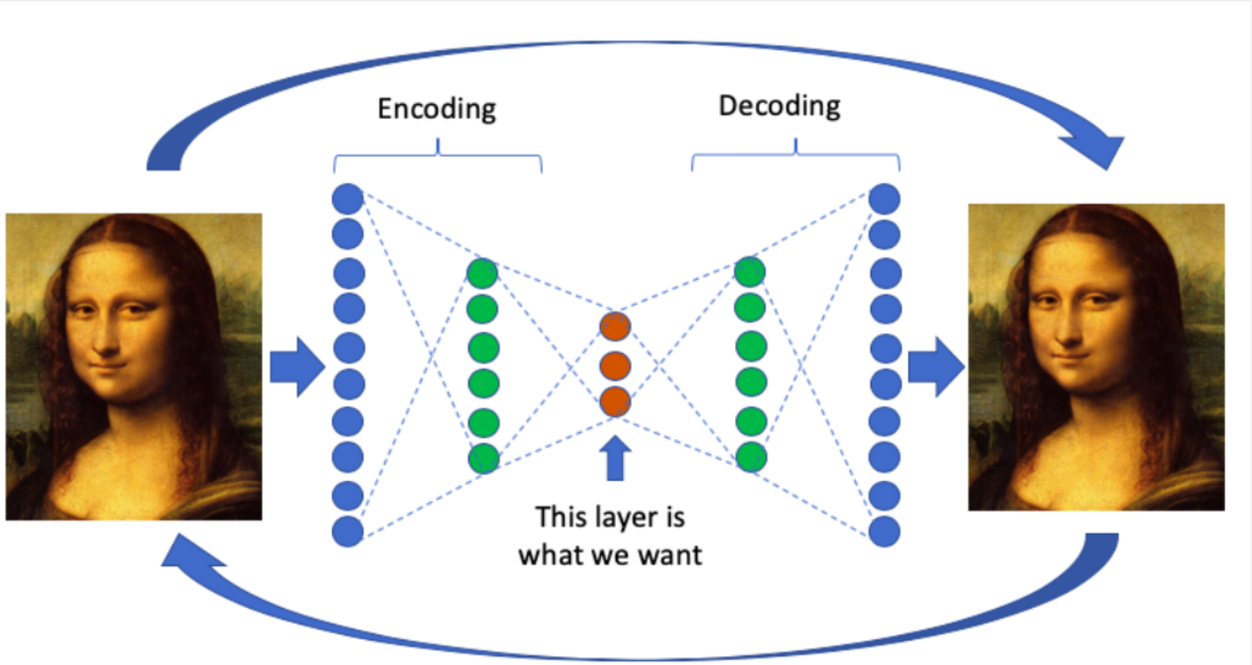
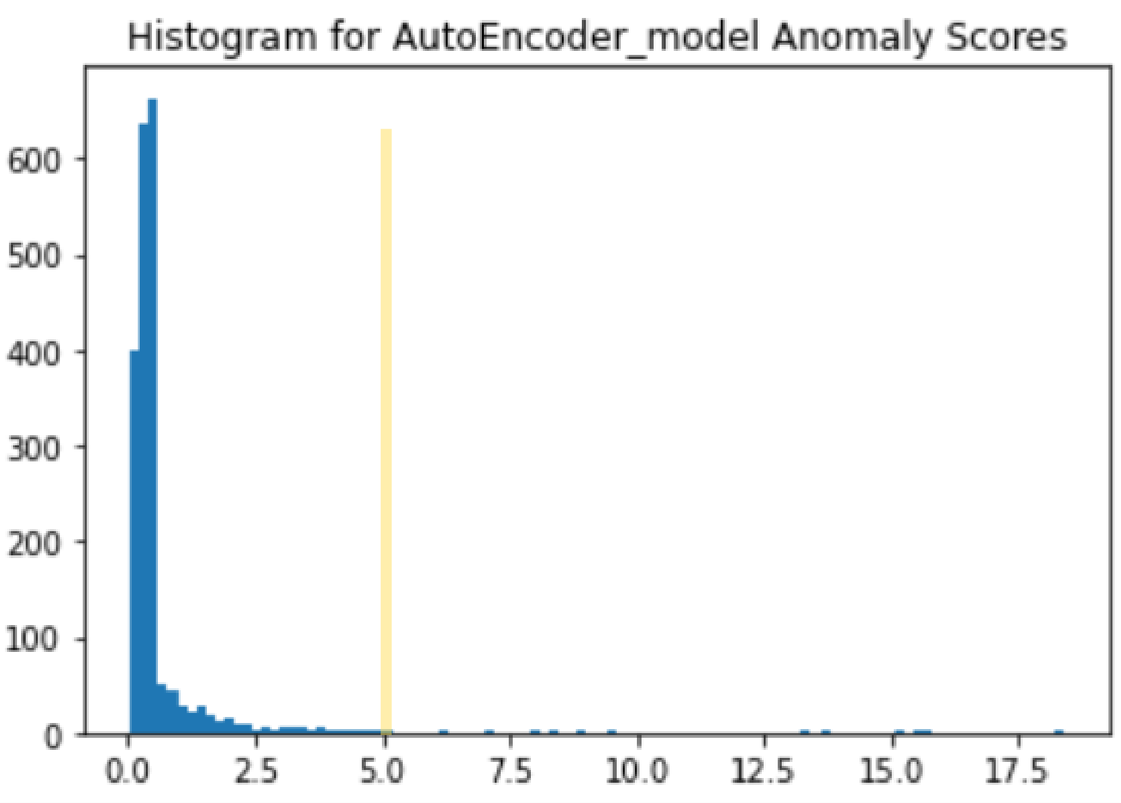


Figure (B): Autoencoders

3. AUTOENCODER: HIDDEN_NEURONS = [5, 2, 2, 5]

- Model: [5, 2, 2, 5], The input layer and the output layer has 5 neurons each. There are two hidden layers, each has two neurons.
- Cut point: 5
- Anomalous cases: 2097 observations, 0.96% of total observations




```
#### Model 1: clf1 has hidden_neurons = [5, 2, 2, 5]
AutoEncoder_model = AutoEncoder(hidden_neurons =[5, 2, 2, 5])
AutoEncoder_model.fit(X_train)
```

```
Train on 199106 samples, validate on 22123 samples
Epoch 1/100
199106/199106 [==============================] - 48s 241us/step - loss: 26.51
Epoch 2/100
199106/199106 [==============================] - 38s 193us/step - loss: 19.57
Epoch 3/100
199106/199106 [==============================] - 40s 200us/step - loss: 15.84
Epoch 4/100
199106/199106 [==============================] - 36s 180us/step - loss: 12.26
Epoch 5/100
199106/199106 [==============================] - 42s 211us/step - loss: 11.18
Epoch 6/100
199106/199106 [==============================] - 47s 236us/step - loss: 10.37
Epoch 7/100
199106/199106 [==============================] - 44s 220us/step - loss: 9.658
Epoch 8/100
199106/199106 [==============================] - 48s 243us/step - loss: 8.945
Epoch 9/100
199106/199106 [==============================] - 64s 321us/step - loss: 8.313
```

3. AUTOENCODER: BUSINESS INSIGHTS

- 0.96% outliers are identified
- Similar result with KNN
- The outliers have higher spending and more time since last purchase compared with average level. It distinguishes the transactions that happen less frequent than usual and have higher spending.
- It is possible that the card is stolen or lost, then being used again with more money each transaction.

cluster	ratio_amount_by_agency_by_category	ratio_amount_by_agency_by_Description	ratio_amount_by_cardholder
0	-1.145794e-16	-2.446088e-16	-0.041776
1	7.244189e-03	1.234511e-01	4.232221

Ratio_AVG_time_since_lastP	Ratio_AVG_time_since_lastP_by_cardholder	anomaly scores
-0.034260	-0.043593	0.518990
3.614706	4.558975	12.733886

4. ISOLATED FOREST: INTRODUCTION

- Isolated Forest randomly select observations and build Itree.
- Each data point in an iTree will have an anomaly score, as there are many iTrees, each data point will have multiple anomaly scores. We take average score for each data point across all the iTrees and get the final anomaly score for that data point.

MUCH FASTER THAN AUTOENCODER

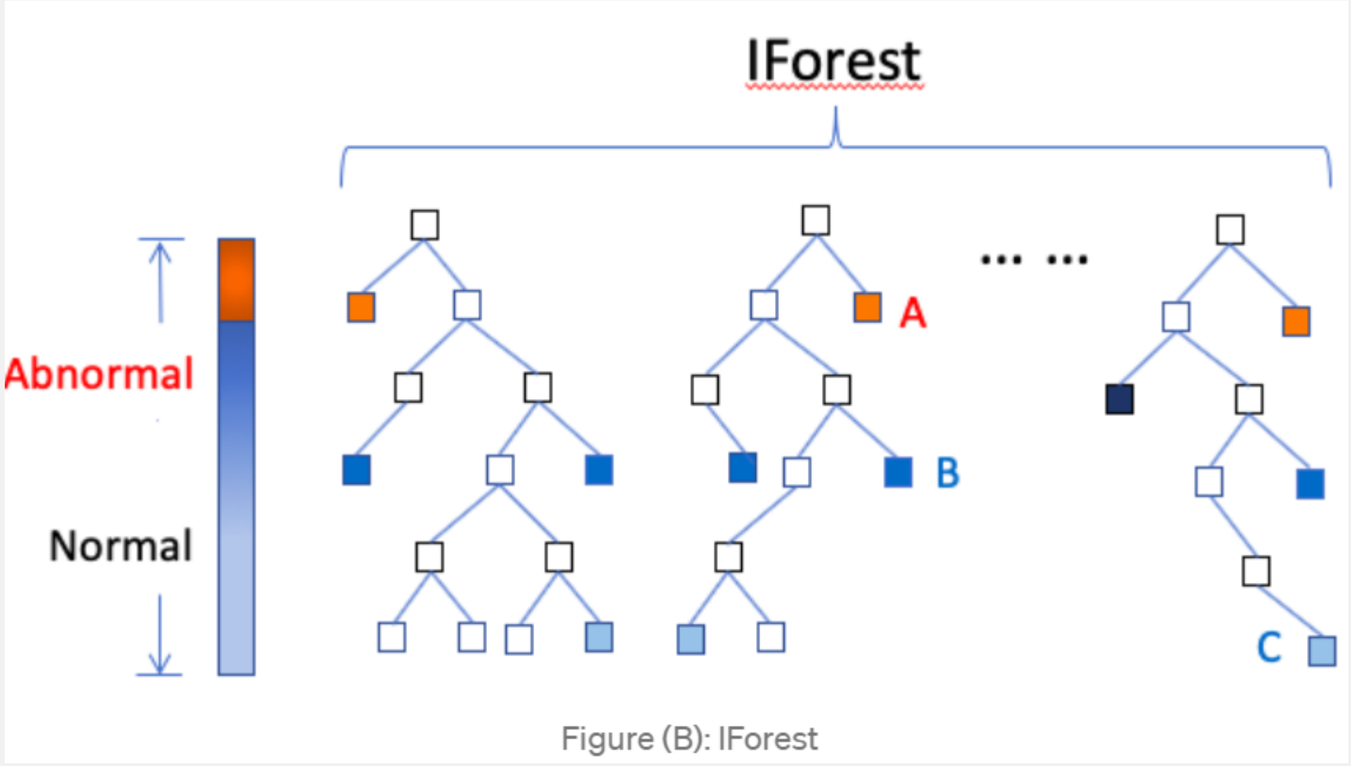


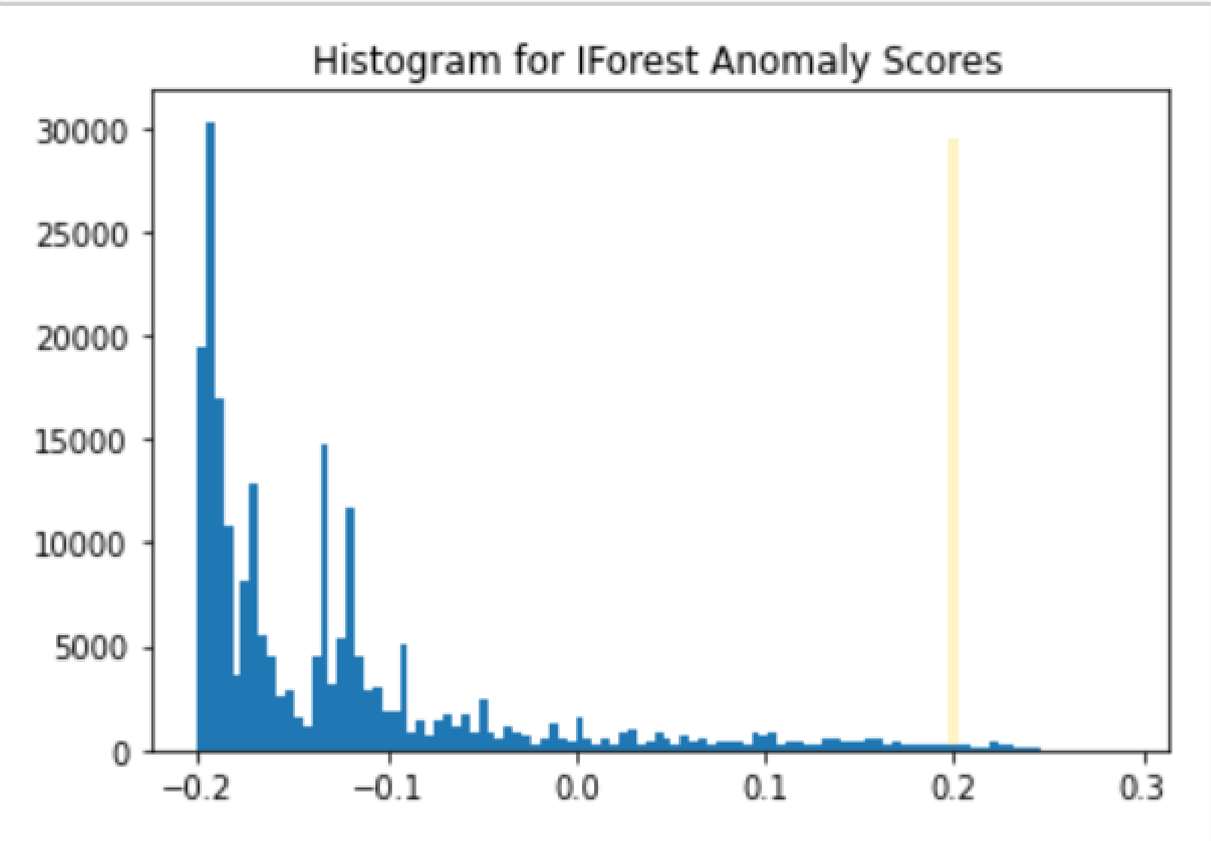
Figure (B): IForest

$$A_{ij} = \begin{pmatrix} 0 & 3 & 1 & 0 & 2 & 3 & 8 & 1 & 1 & 3 \\ 1 & 1 & 0 & 0 & 7 & 1 & 2 & 2 & 3 & 3 \\ 1 & 2 & 2 & 0 & 0 & 6 & 7 & 1 & 2 & 2 \\ 1 & 2 & 3 & 10 & 0 & 4 & 6 & 1 & 0 & 5 \\ 3 & 2 & 2 & 1 & 4 & 3 & 2 & 1 & 6 & 0 \\ 7 & 4 & 4 & 5 & 3 & 9 & 6 & 1 & 6 & 1 \\ 7 & 1 & 1 & 5 & 2 & 8 & 9 & 1 & 3 & 6 \\ 5 & 0 & 1 & 6 & 2 & 0 & 0 & 0 & 1 & 5 \\ 1 & 6 & 3 & 3 & 4 & 6 & 2 & 0 & 1 & 1 \\ 1 & 2 & 2 & 4 & 1 & 1 & 3 & 0 & 8 & 2 \end{pmatrix}$$

(1) (2) (3)

4. ISOLATED FOREST： MAX_SAMPLES = 60

- Determine boundary: 0.2
- Anomalous cases: 2205 observations, 1.01% of total observations



4. ISOLATED FOREST: BUSINESS INSIGHTS

- 1% outliers are identified
- Compared with KNN and Autoencoder, Isolate forest discovered similar patterns for outliers on cardholder level, while KNN and Autoencoder identifies outliers in both agency and cardholder level.
- Cluster 1 has more spending on cardholder level and takes more time for this transaction to happen. It is possible that the card is stolen or lost, then being used again.

cluster	ratio_amount_by_agency_by_category	ratio_amount_by_agency_by_Description	ratio_amount_by_cardholder
0	6.935799e-05	1.181957e-03	-0.006190
1	1.228787e-15	3.068525e-15	0.488076

Ratio_AVG_time_since_lastP	Ratio_AVG_time_since_lastP_by_cardholder	anomaly scores
-0.042388	-0.056681	-0.129635
4.243346	5.633590	0.222902