ASSETTE

Detecting Anomalies

Team B2

Boston University Questrom School of Business

BOSTON UNIVERSITY

Our Team

Aash Gohil

Tommy Yang

Zixing Li

Phyllis Cao

Zihan cui

Agenda

- *Project and Progress Background*
- *Datasets Overview & EDA*
- *Optimized Detection Method 1 - Modified Z-score*
- *Optimized Detection Method 2 - KNN*
- *Models Performance Measurement and Pipeline Designs*
- *Project Summary*

Project Background - Data Outlier Detection

- Assette data warehouse refreshes its client databases on a batch process.

- Possibility that Assette receives data that has not been properly curated.

- Detecting errors after data ingestion has a high cost on time and it can also cause issues with the compliance if it is not handled properly in time

- **Solution**: ***Develop a series of statistical validations to detect anomalies in data submitted by asset managers (Assettes's clients) during ingestion process.***

Progress on Z-score Model along with Building New

- Understood datasets structure, explored key data features in main datatables
- Utilize Z-score model on data but still needs to develop the optimized function
- Develop researches on a set of **secondary statistical validation rules** including cluster analysis and other appropriate methods, needs to build the function and implement on data

Data Features Overview

- There are **502 accounts** from 1979 to 2021 and **244 benchmarks** from 1969 to 2019.
- We heavily use **'MonthlyValue'** in each table to do the anomaly detection, since it's easy to see if the value is inside or outside the normal range.
- The original 'MonthlyValue' is **left skewed**, but we look for normal distribution (see next page).

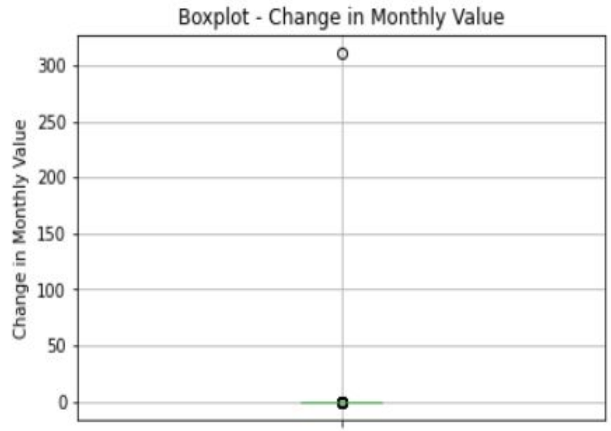
BenchMark History		
Number of benchmark		244
	Num Benchmark	Year
oldest account	1	1969
newest account	2	2019
maximum accounts	98	1994
median accounts	98	1994

AccountPerformanceFactors		
Number of accounts		502
	Num Account	Year
oldest account	1	1979
newest account	18	2021
maximum accounts	36	2015
median accounts	19	2011

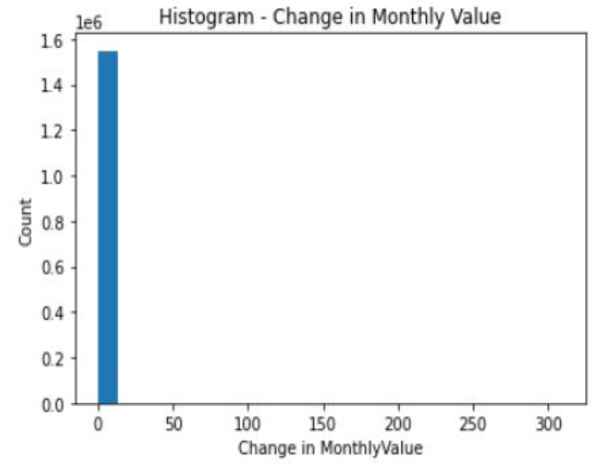
Relationship between BMH and APF	
Num of benchmark	Num of Accounts
1	168
2	293
3	25
4	19
5	2
6	5
7	1

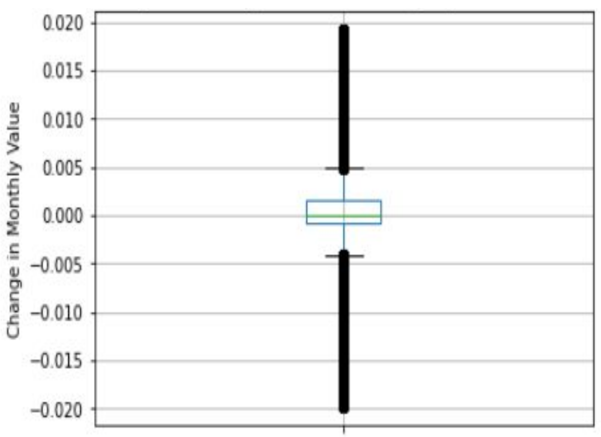
Subsetting Outliers to Generate Normal Distribution

Account performance - ‘monthly value’

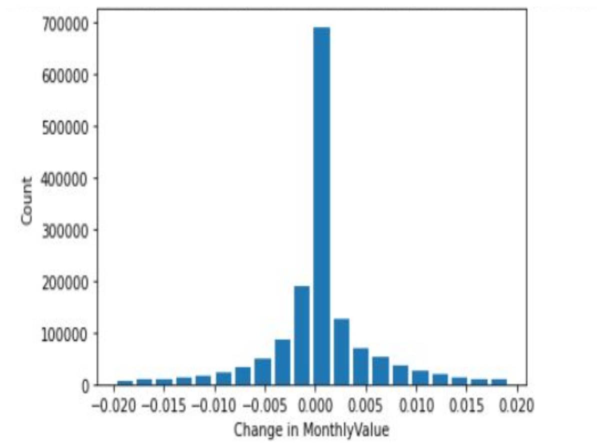


count	1548221
mean	0.00049841
std	0.25027337
min	-0.99679540
25%	-0.00092048
50%	0.00007890
75%	0.00166912
max	311.21399605



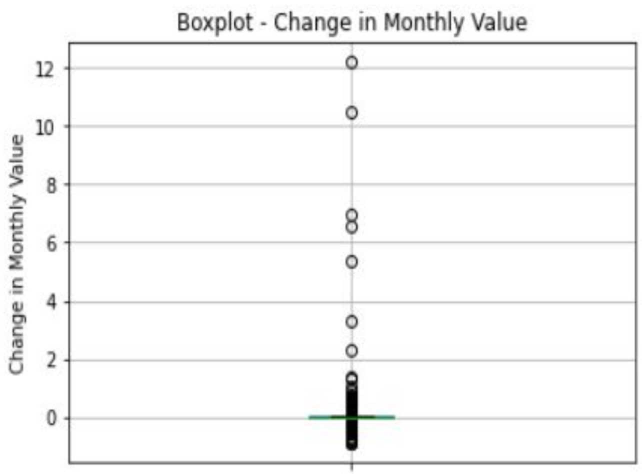


count	1488342
mean	0.00033709
std	0.00507803
min	-0.01984254
25%	-0.00077853
50%	0.00007892
75%	0.00151997
max	0.01920008

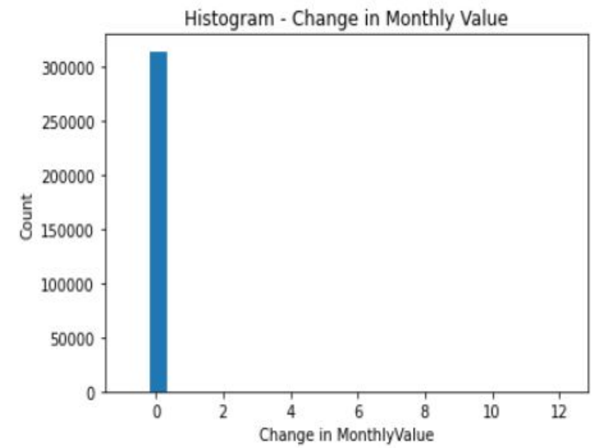


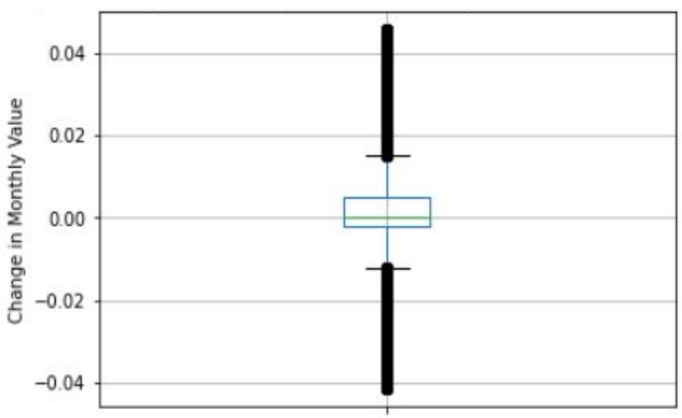
Subsetting Outliers to Generate Normal Distribution

Benchmark - ‘monthly value’

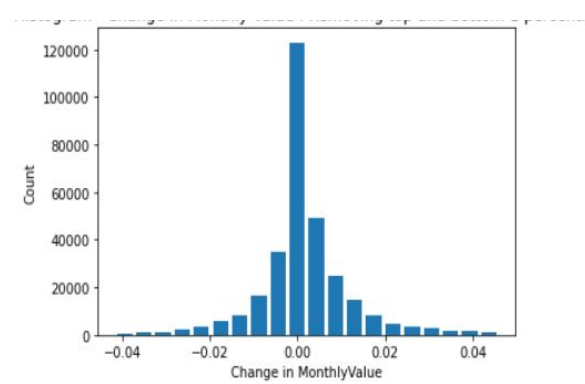


count	314888
mean	0.00155766
std	0.03898630
min	-0.92339600
25%	-0.00214400
50%	0.00036400
75%	0.00499100
max	12.24417700



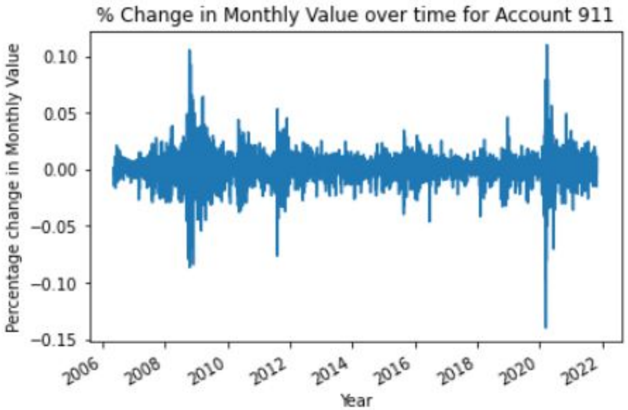


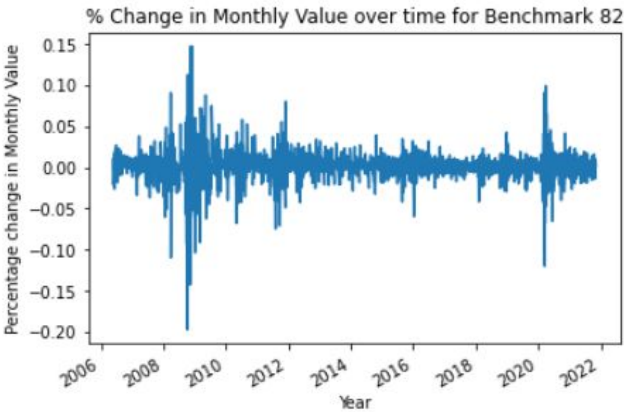
count	308660
mean	0.00142636
std	0.01018846
min	-0.04163900
25%	-0.00203000
50%	0.00036500
75%	0.00485100
max	0.04584000

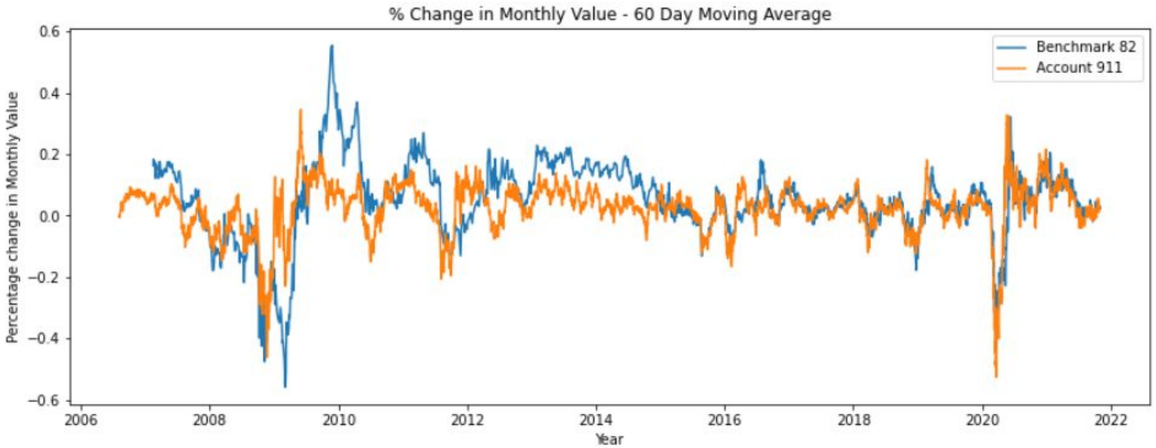


Comparing Sample Account and Corresponding Benchmark

- Expect to find the anomalous data for an account that differs from its pegged benchmark.
- Most of the monthly values and monthly value percentage are between **[-0.05, 0.05]**.







Two Methods Implementing on Primary & Comparison Datasets

- 2 Methods - Modified Z-score & KNN
  - Utilize Modified Z-score model from mid-term and bring out according function
  - Create the KNN function
- 2 Datasets - Primary Dataset & Comparison Dataset
  - Implement methods on original Account Performance Factor datasets
  - Also Implement on the created Comparison between Account and Benchmark Dataset
- In result it would have 4 functions with 4 anomalies detection result output

Method 1: Modified Z-Score Model

- The standard z-score are limited when the data are not normally distributed or the data/sample size is small, also sensitive to extreme values

- The modified z-score is a standardized score that measures outlier strength or how much a particular score differs from the typical score
  - It is less influenced by outliers when compared to the standard z-score because it relies on the median for calculating the z-score
  - The modified z-score is calculated from the mean absolute deviation (MeanAD) or median absolute deviation (MAD)
  - The values are multiplied by a constant to approximate the standard deviation

$$\text{Z-score} = \frac{x - mean}{\text{Standard Deviation}}$$

If MAD = 0:

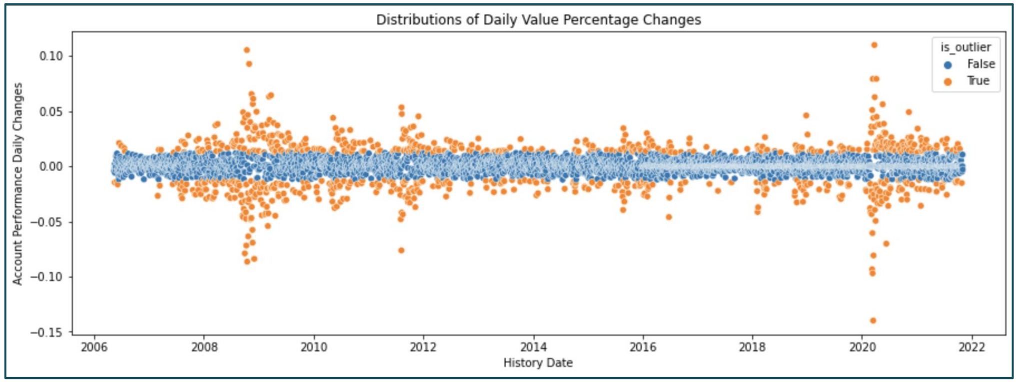
*Modified Z-score = (X - MED) / (1.253314*MeanAD)*

If MAD ≠ 0:

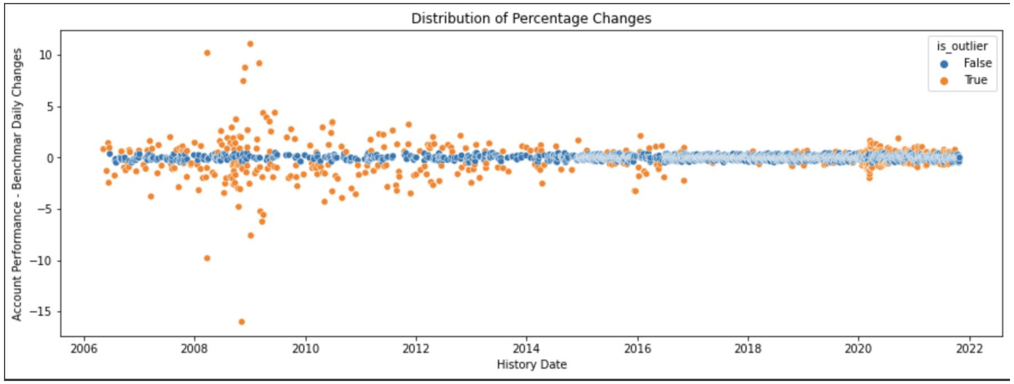
*Modified Z-score = (X - MED) / (1.486*MAD)*

Method 1 Implementation on 2 Datasets

- Model implements on sample account 911 percentage of monthly values changes
- Modified Z-score implementation on primary dataset got 824 anomalies
- Modified Z-score implementation on comparison dataset got 363 anomalies



- Modified Z-score on primary dataset of percentage of monthly values changes
- Orange points are detected outliers

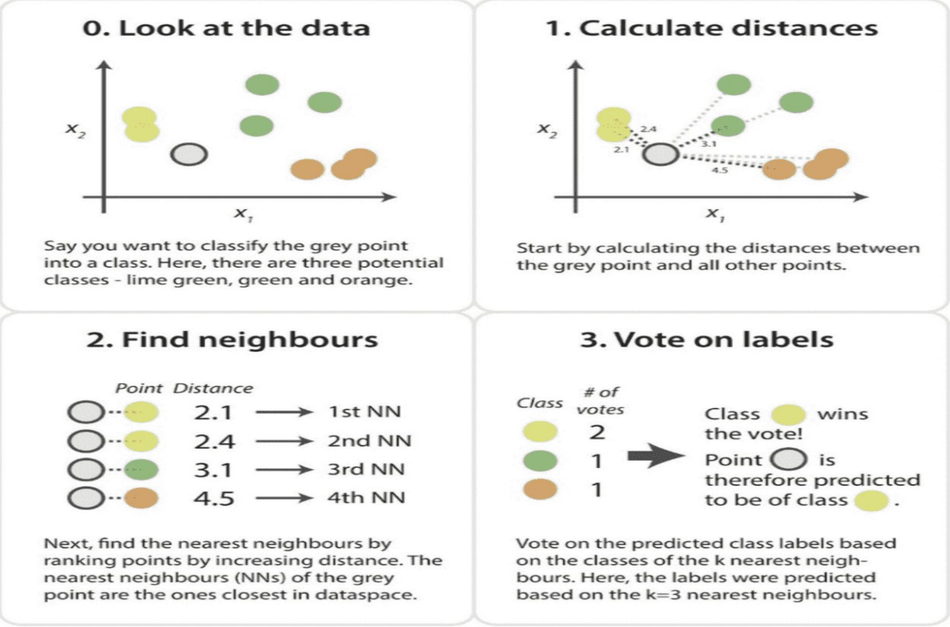


- Modified Z-score on comparison dataset
- Y label represents the difference between account performance and benchmark

Method 2: K-Nearest Neighbours algorithm Model

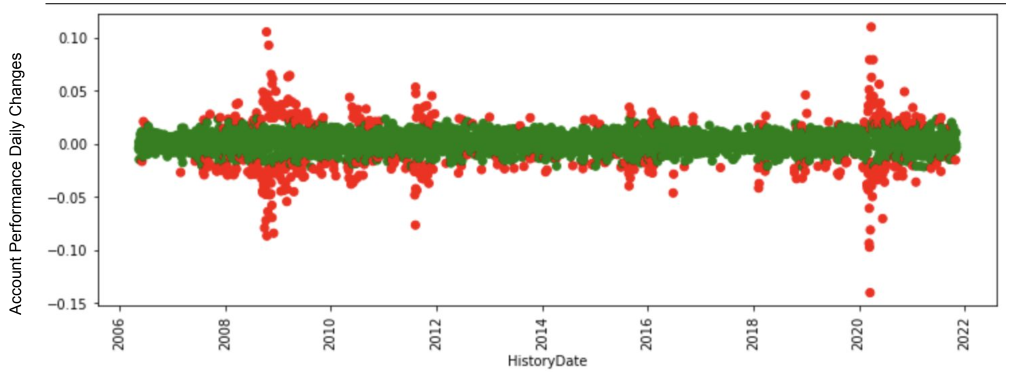
- K-Nearest Neighbours algorithm detects anomalies using the distances of k-nearest neighbors as anomaly scores.
  - If an observation is much far from the other observations then that observation is considered to be an anomaly.

- The key parameter in KNN is N_neighbors, which determines the number of neighbors to use for calculating distances from the point of measurement

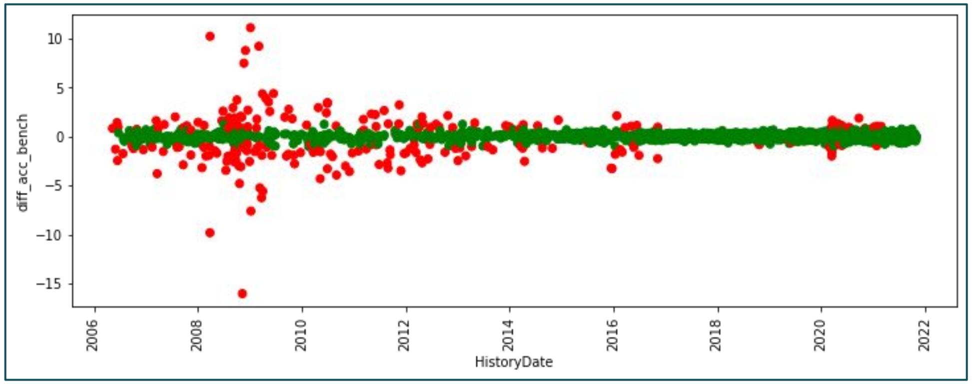


Method 2 Implementation on Two Datasets

- Model implements on sample account 911 percentage of monthly values changes
- KNN model implementation on primary dataset got 460 anomalies
- KNN model implementation on difference dataset got 222 anomalies



- KNN on primary dataset of percentage of monthly values changes
- Red points are detected outliers



- KNN on comparison dataset
- Y label represents the difference between account performance and benchmark

Anomalies Measurement with 4 Model Outputs

- After implementing 4 methods, establish a vote function that combines 4 outputs and the total votes would decide whether anomalies or not
  - For example, for the single datapoint, if model 1,3&4 decided it anomaly and model 2 not, the total votes for this point is 3
  - Then the return statement would define if total votes >= 3 then return anomalies, the datapoint would be detected as an anomaly
  - The vote parameter could be decided and entered by the Team

Sample Output of Process

```
             Model 1 | Model 2 | Model 3 | Model 4 | total Votes
03/31/2022 -   1         0         1         1          3
```

Sample Return Statement

```
return rows where df.totalvotes >= 3
```

Final Vote Outputs

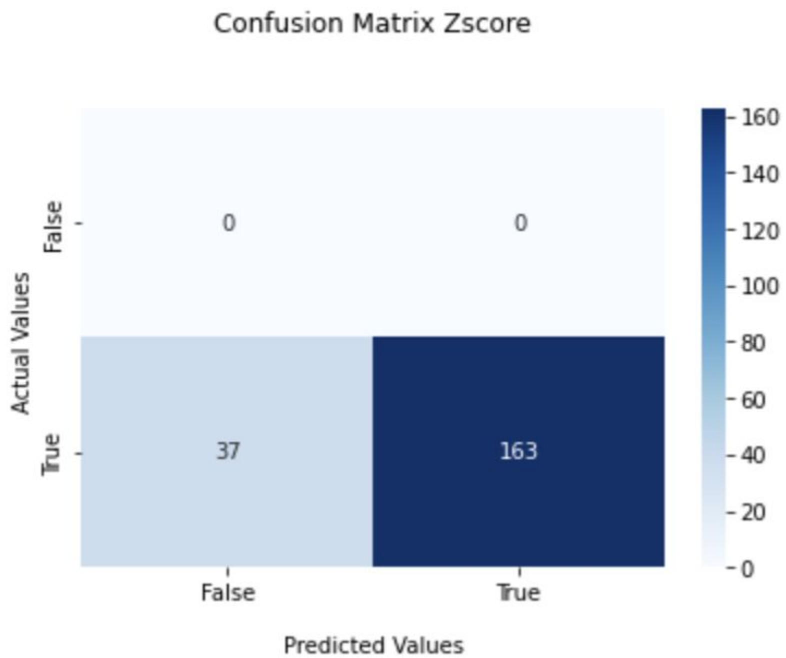
- The final vote outputs for the sample account shows that 440 detected anomalies got 1 vote, 516 anomalies got 2 votes, 51 anomalies got 3 votes and 61 anomalies got 4 votes
- Final decision will base on the voting parameter that Business Team choose

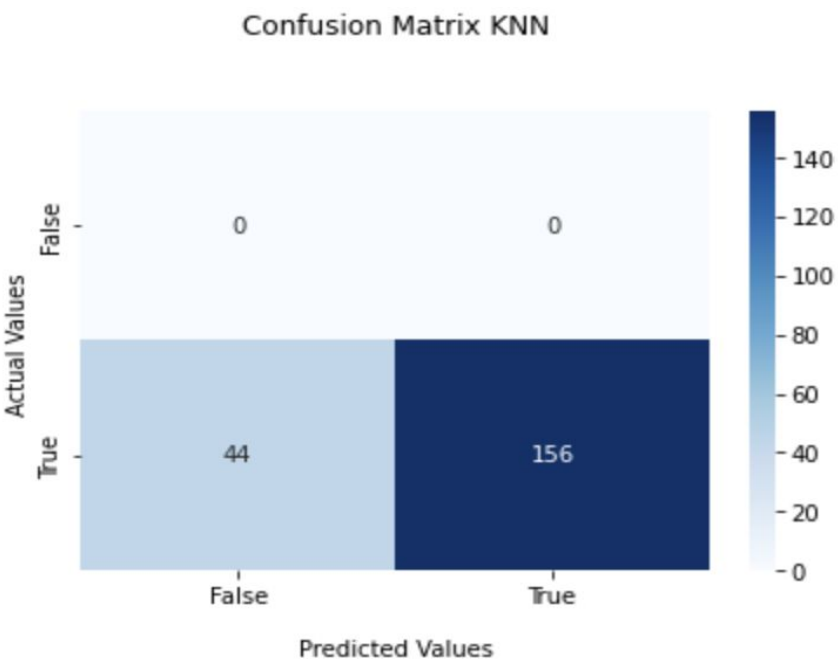
	date	model1_modified_zscore	model2_modified_zscore_diff	KNN	KNN_diff	final_sum
0	2006-05-12	0	0	0.0	0.0	0.0
1	2006-05-15	0	0	0.0	0.0	0.0
2	2006-05-16	0	0	0.0	0.0	0.0
3	2006-05-17	1	0	0.0	0.0	1.0
4	2006-05-18	0	0	0.0	0.0	0.0
...	...	...	...	...	...	...
4733	2015-11-15	0	0	0.0	0.0	0.0
4734	2015-11-22	0	0	0.0	0.0	0.0
4735	2015-11-26	0	0	0.0	0.0	0.0
4736	2015-11-29	0	0	0.0	0.0	0.0
4737	2015-12-06	0	0	0.0	0.0	0.0

4738 rows × 6 columns

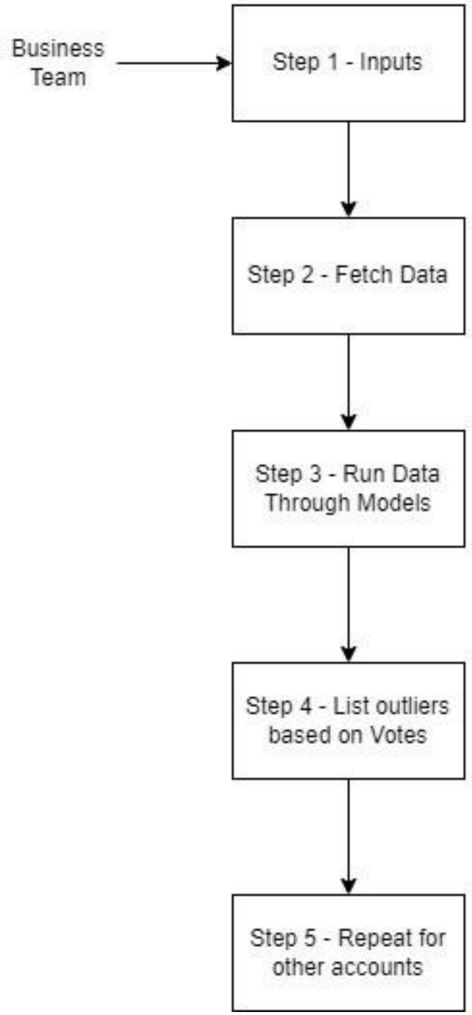
Manually Create Anomalies to Test Model Accuracy

- Conduct litmus test in order to test model accuracy
  - Randomly take 200 monthly-change-percentage values, multiply by 10, as human error and label those data as anomalies
  - Run through Modified Z-score and kNN models to test whether they could detect out the anomalies we created
- The Modified Z-score performs better for getting more True Positive results





Project Implementation in Actual Business Practice



- Step 1: The Business Team enters 3 inputs about the account they want to examine, the total vote parameter and the confidence interval

- Step 2: Coming from the pipeline to DB, fetch data for the input account

- Step 3: Running through 4 models - modified Z-score & KNN methods respectively on Account & Account Benchmark Difference Data

- Step 4: With 4 outputs and the defined vote parameter, list out corresponding outliers

- Step 5: Repeat the whole process for other accounts

Project Summary

1. Litmus test shows that the modified z-score method is more accurate than the kNN method
2. The validity of methods can also be verified by detecting out larger proportions of anomalies during economic events
3. Final detection outputs for the sample account are 824 & 363 outliers for modified z-score, detecting more outliers than the 460 & 222 for kNN
4. For limitations, there were no labelled data so anomalies need to be manually verified, so that the next steps can focus on experimenting with other different methods
5. Our project provides the automated outliers detection from pipeline to database which can help gain more effectiveness in daily business operations

Thank You

References

- https://towardsdatascience.com/statistical-techniques-for-anomaly-detection-6ac89e32d17a
- https://www.geeksforgeeks.org/dbscan-clustering-in-ml-density-based-clustering/
- https://www.analyticsvidhya.com/blog/2021/06/univariate-anomaly-detection-a-walkthrough-in-python/#:~:text=K%2DNearest%20Neighbours%20algorithm,-K%2DNearest%20Neighbours%20algorithm
- https://towardsdatascience.com/k-nearest-neighbors-knn-for-anomaly-detection-fdf8ee160d13