Bike Bandits

An Analysis on the Bicycle Thefts in Toronto

Table of Contents

[Table of Contents 2](#_o15c1ga5dhm6)

[Introduction 3](#_onxrwaxpwfjy)

[Business Understanding 3](#_k0ghydxc4npu)

[Data Understanding 4](#_l91675epsp6f)

[Data Preparation 7](#_w2mx96kc98y6)

[Modeling 16](#_bn34p6a63s4p)

[Outlier Detection 17](#_or9hb61kp6hh)

[Clustering 18](#_s1izz9dzwwse)

[Classification(Decision Tree) 21](#_jacmzrmxxz3c)

[Association Rule Mining 24](#_kqlytirm804a)

[Discussion of Results 28](#_i0z4jxlldrwy)

# Introduction

Toronto is no stranger to thievery. As the largest city in Canada, bikes are always highly susceptible to theft, despite our efforts to prevent it. These frustrating events not only impact individual citizens in their day-to-day lives, but also pose a challenge to local law enforcement and urban planners.

Our project aims to delve into this issue by analyzing a comprehensive dataset of bicycle theft occurrences reported to the Toronto Police Service **from 2014 to 2020.**

# Business Understanding

**Project plan**

We aim to determine the **predominant factors and key associative rules that contribute to bicycle thefts** in Toronto. Additionally, we seek to identify which **factors increase the likelihood of bike recovery post-theft**.

**Hypotheses for Analysis**

The following attributes may have a significant impact on both the likelihood of a bicycle being stolen and the chances of its recovery:

* Bike Characteristics: Color, cost, and make/model of the bicycle.
* Environmental Factors: Location where the bike is parked/locked.
* Temporal Factors: Time of the day and day of the week and season of the year when the bike is most frequently parked.

To achieve our objectives, we will leverage the Cross-Industry Standard Process for Data Mining (CRISP-DM) framework. The analysis will be conducted using RapidMiner, a robust tool for data processing and mining.

**Data Analysis Strategies**

Outlier Detection:

Employ techniques such as Local Outlier Factor (LOF) and isolation forests to identify and manage outliers, ensuring the robustness of our analysis.

Clustering:

Utilize clustering algorithms to segment the data, which may reveal patterns in the geographical and temporal aspects of bike thefts and recovery.

Classification:

Apply decision tree classifiers to sort the data into categories of risk, clarifying the profile of high-risk scenarios for bike thefts.

Association Rule Mining:

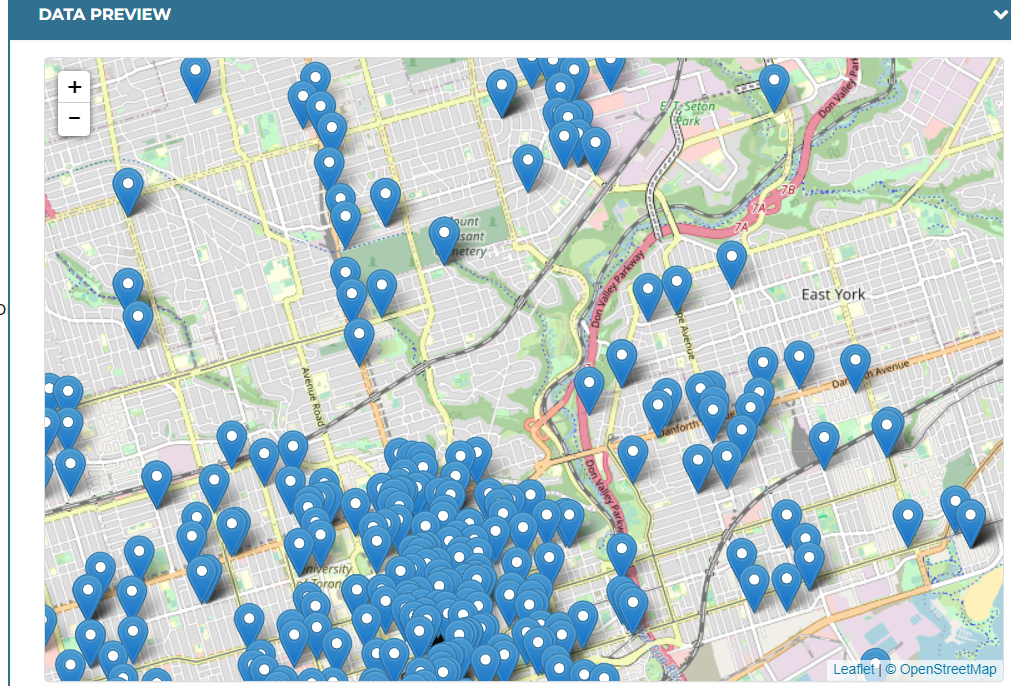
Perform association rule mining to detect strong rules that might indicate the conditions under which thefts are most likely to occur.

**Expected Outcomes**

* A ranked list of factors based on their influence on theft and recovery.
* A set of associative rules that describe the conditions under which thefts are more likely to occur.
* Recommendations on measures that could be implemented to reduce the risk of theft and improve recovery rates.

# Data Understanding

The dataset, sourced from the Toronto Police Service, includes 34291 instances of reported bicycle thefts. It encompasses a wide array of data points, such as the date of the theft, specifics of the stolen bicycle, and the location of the theft location.



However, it's important to note that some entries lack precise coordinates, and in certain cases, the reported locations extend beyond the City of Toronto. Additionally, the dataset aligns with both the old (140) and new (158) City of Toronto Neighbourhood structures.

The dataset contains 28 attributes, which include two identification labels, offering a detailed perspective of each theft occurrence.

Attribute information table:

| Column | Description | datatype | quality | description |
| --- | --- | --- | --- | --- |
| \_id | Unique row identifier for Open Data database, shows this dataset has 34290 instances | int | no missing |  |
| EVENT\_UNIQUE\_ID | Offence Number | nominal | no missing | not much use for this analysis |
| PRIMARY\_OFFENCE | Primary Offence Type | nominal | 0 | The majority of the data is concentrated in the top two categories:Theft under and theft under-bicycle, This label attribute is not useful for data analysis purposes. |
| OCC\_DATE | Date of Offence | Date | 28564 Missing | too much missing, plus we already have individual year month and day record so we will drop this column |
| OCC\_YEAR | Year Offence Occurred | int | no missing | from 1975 to 2023 |
| OCC\_MONTH | Month Offence Occurred | nominal | no missing |  |
| OCC\_DOW | Day of the Month Offence Occurred | nominal | no missing | Monday to Sunday |
| OCC\_DAY | Day of the Year Offence Occurred | int | no missing |  |
| OCC\_DOY | Day of the Week Offence Occurred | int | no missing | not much use for this analysis |
| OCC\_HOUR | Hour Offence Occurred | int | no missing |  |
| REPORT\_DATE | Date Offence was Reported | Date | 28544 missing | too much missing, plus we already have individual year month and day record so we will drop this column |
| REPORT\_YEAR | Year Offence was Reported | int | no missing |  |
| REPORT\_MONTH | Month Offence was Reported | nominal | no missing |  |
| REPORT\_DOW | Day of the Month Offence was Reported | nominal | no missing |  |
| REPORT\_DAY | Day of the Year Offence was Reported | nominal | no missing |  |
| REPORT\_DOY | Day of the Week Offence was Reported | int | no missing |  |
| REPORT\_HOUR | Hour Offence was Reported | int | no missing |  |
| DIVISION | Police Division where Offence Occurred | nominal | no missing | 18 divisions, can narrow down to fewer types as data concentrated on top 8 divisions |
| LOCATION\_TYPE | Location Type of Offence | nominal | no missing | This attribute is almost identical in meaning to PREMISES\_TYPE attribute |
| PREMISES\_TYPE | Premises Type of Offence | nominal | no missing | 7 types |
| BIKE\_MAKE | Make of Bicycle | nominal | no missing | 1142 types, should narrow down to top N |
| BIKE\_MODEL | Model of Bicycle | nominal | 12824 missing | will remove the missing value instances, also there is 1000+types, should narrow it down to top N |
| BIKE\_TYPE | Type of Bicycle | nominal | no missing | 13 types, could also narrow down to topN |
| BIKE\_SPEED | Speed of Bicycle | real | 381 missing | The data needs to be binned and divided into several broad categories. also would try to predict the missing value according to classification result |
| BIKE\_COLOUR | Colour of Bicycle | nominal | 2952 missing | 282 types, should narrows down to topN |
| BIKE\_COST | Cost of Bicycle | real | 2335 missing | The data needs to be binned and divided into several broad categories. also would try to predict the missing value according to classification result |
| STATUS | Status of Bicycle | nominal | no missing | 3 types: stolen, recovered, unknown |
| geometry | location coordinate | nominal | no missing | would extract the coord X and Y out and use them for division clustering |

# Data Preparation

**Attributes Dropped**

Not all the attributes are particularly helpful when deciphering causation of theft. As such, here are all the attributes that were dropped:

\_id, EVENT\_UNIQUE\_ID: In a dataset analysis, the id of the instances have no significance.

PRIMARY\_OFFENCE: Since the dataset consists of all thefts, the type of offence has no purpose within our calculations.

OCC\_[DATE, DOY, DAY]: The DATE is a redundant value, since we already have attributes defining date details. DOY, though irrelevant for direct analysis, can be used to compare with the REPORT\_DOY in order to calculate how many days it took for someone to report the incident. Similarly to DOY, DAY seems useful for exterior calculations, but is not really of use for direct analysis.

REPORT\_[DATE, YEAR, MONTH, DOW, DAY, DOY, HOUR]: We will utilize the report and occurrence dates to compute the time gap between when the incident was reported and when it actually occurred. Otherwise, we will discard the additional report date information.

LOCATION\_TYPE: Location\_type contains useful information, but it is highly cluttered and awfully specific. In this analysis the general premises of the theft is enough, which is why we chose PREMISES\_TYPE over LOCATION\_TYPE.

**Attribute Selection and Preparation**

we selected specific attributes from the dataset, including three primary groups:

| **Column** | **datatype** | **data clean & construct** |
| --- | --- | --- |
| ReportDayGap (calculated & dropped) | int | generated attributes, use occ date and report date to calculate this value: if(OCC\_YEAR == REPORT\_YEAR , REPORT\_DOY - OCC\_DOY, if(OCC\_YEAR != REPORT\_YEAR, (((REPORT\_YEAR - OCC\_YEAR) \* 365) - OCC\_DOY) + REPORT\_DOY, 0))  This attribute will be used for the ReportHourGap. It is being dropped since we already have a value to use to gauge how long it has been since the occurrence. |
| ReportHourGap to binnedReportHourGap | int | Generated attributes, use occ hour and report hour to calculate the difference in hour plus the day difference \* 24: if(ReportDayGap==0,REPORT\_HOUR-OCC\_HOUR,((ReportDayGap \* 24)-OCC\_HOUR)+REPORT\_HOUR)  Then binned it:if( ReportHourGap < 5, "fast",  if(ReportHourGap < 12, "half day",  if(ReportHourGap < 24, "same day",  if(ReportHourGap < 48, "two day", "long time")))) |
| Season | nominal | generated based on month:if(OCC\_MONTH == "November" || OCC\_MONTH == "December" ||OCC\_MONTH == "January", "Winter",  if(OCC\_MONTH == "February" || OCC\_MONTH== "March" || OCC\_MONTH == "April", "Spring",  if(OCC\_MONTH == "May" || OCC\_MONTH == "June" || OCC\_MONTH== "July", "Summer",  if(OCC\_MONTH == "August" || OCC\_MONTH == "September" || OCC\_MONTH == "October", "Fall",  "Unknown")))) |
| OCC\_MONTH(dropped) | nominal | According to the diagram,it is clear that month tendency follows the pattern that the warmer weather is, the more thefts happen, so we will bin this attribute values to four seasons for a new attribute then drop this attribute |
| OCC\_DOW, OCC\_YEAR | nominal | keep it as it is |
| HoursOfTheDay |  | generated based on hour of the day: if( OCC\_HOUR >= 5 && OCC\_HOUR< 12, "Morning",  if(OCC\_HOUR >= 12 && OCC\_HOUR < 17, "Afternoon",  if(OCC\_HOUR >= 17 && OCC\_HOUR < 21, "Evening",  if(OCC\_HOUR >= 21 || OCC\_HOUR < 5, "Night", "Undefined")))) |
| OCC\_HOUR(dropped) | int | Bin this attribute values to five period then drop this attribute |
| Geometry\_X (calculated & dropped), Geometry\_Y (calculated & dropped),  NewDivision | real | extract the coord X and Y out and use them for division clustering: split the geometry by comma (,), which will result in 3 geometries. Remove the surrounding information around geometry\_2 and geometry\_3, and rename the 2 values to “Geometry\_X” and “Geometry\_Y”. Convert values to real using parse numbers.    When selecting attributes, geometry\_1 will be excluded.  The next step is to cluster the values into new divisions. For this, we will use k-Mean to cluster the geometry.    The elbow graph shows a sharper edge on k=3, however the subsequent k values did not truly flatten until k=5; we will be using k=5, as it also adds more detail to the division clustering.  We verified clustering with k = 7 and 3, but it seems like clustering with 5 is the most necessary.  7:, 3:  5:  Lastly, a new attribute (NewDivision) was made as an nominal attribute, where each cluster dictates a division 1 through 5.  if(cluster\_division == "cluster\_4", 5,  if(cluster\_division == "cluster\_3", 4,  if(cluster\_division == "cluster\_2", 3,  if(cluster\_division == "cluster\_1", 2,  if(cluster\_division == "cluster\_0", 1,  0))))) |
| geometry, DIVISION (dropped) | nominal | Since Geometry\_X and Geometry\_Y will be used to cluster geographical orientation, DIVISION will be of no use to us, since it would also be used to calculate location clusters, with less accuracy. |
| PREMISES\_TYPE | nominal | 7 types, keep it as it is |
| BikeMake | nominal | There were 1142 unique bike makes, therefore we narrowed it down to the top 4, since these 4 were the most common by a long shot:  if(BIKE\_MAKE == "OT", "OT",  if(BIKE\_MAKE == "UK", "UK",  if(BIKE\_MAKE == "GI", "GI",  if(BIKE\_MAKE == "TR", "TR",  if(BIKE\_MAKE == "NO", "NO",  "Others"))))) |
| BikeModel to filledBikeModel | nominal | 12824 missing instances are replaced with "UNKNOWN, narrow the model types down to 6 types:  if(contains( BIKE\_MODEL, "MOUNTAIN") || contains(BIKE\_MODEL, "ROCKHOPPER") || contains(BIKE\_MODEL, "AGGRESSOR") || contains(BIKE\_MODEL, "HARDROCK") || contains(BIKE\_MODEL, "CROSSTRAIL") || contains(BIKE\_MODEL, "AVALANCHE") || contains(BIKE\_MODEL, "MARLIN"), "Mountain Bikes",  if(contains(BIKE\_MODEL, "ROAD") || contains(BIKE\_MODEL, "SIRRUS") || contains(BIKE\_MODEL, "ALLEZ") || contains(BIKE\_MODEL, "VITA") || contains(BIKE\_MODEL, "SYNAPSE") || contains(BIKE\_MODEL, "ROUBAIX"), "Road Bikes",  if(contains(BIKE\_MODEL, "HYBRID") || contains(BIKE\_MODEL, "ESCAPE") || contains(BIKE\_MODEL, "DEW") || contains(BIKE\_MODEL, "CYPRESS") || contains(BIKE\_MODEL, "FX2"), "Hybrid Bikes",  if(contains(BIKE\_MODEL, "URBAN") || contains(BIKE\_MODEL, "CITY") || contains(BIKE\_MODEL, "INDIE"), "Urban/Commuter Bikes",  if(contains(BIKE\_MODEL, "CRUISER") || contains(BIKE\_MODEL, "BMX") || contains(BIKE\_MODEL, "SINGLE SPEED") || contains(BIKE\_MODEL, "CLASSICO") || contains(BIKE\_MODEL, "HEART"), "Specialty Bikes",  "Other Models")))))  Filled in missing values:if(missing(BikeModel),"OTHER",BikeModel) |
| BikeType | nominal | Most of the distinct bike types were rather niche and appeared very rarely in the dataset. As such, we decided to reduce the bike types to the 6 most common types, with the remaining bike types being listed as “Other”:  if(BIKE\_TYPE == "MT", "MT",  if(BIKE\_TYPE == "RG", "RG",  if(BIKE\_TYPE== "OT", "OT",  if(BIKE\_TYPE == "RC", "RC",  if(BIKE\_TYPE == "EL", "EL",  if(BIKE\_TYPE == "TO", "TO",  "Others")))))) |
| BIKE\_SPEED | real | The data needs to be binned and divided into several broad categories. also would try to predict the missing value according to classification result    First: select these three attributes and use One hot encoding operator on them  Use K-means on them and calculate the Elbow point, seems k=7  Chart  Then I use operator to combine the output dataset with the bigger dataset    Then filled in the missing Bike\_Speed values with the cluster’s average value:  if(missing(BIKE\_SPEED),  if(cluster == "cluster\_6", 18.32,  if(cluster == "cluster\_2", 16.49,  if(cluster == "cluster\_3", 15.26,  if(cluster == "cluster\_1", 14.46,  if(cluster == "cluster\_5", 14.20,  if(cluster == "cluster\_4", 13.40,  if(cluster == "cluster\_0", 10.31,  BIKE\_SPEED))))))),  BIKE\_SPEED) |
| BIKE\_COLOUR | nominal | This attribute contains a wide range of colours, shades of colours, and colour combinations. Many of the values are written in short form (ex. Silver Red is written SILRED), and with the wide variety of colours, we decided to trim down these colours to be more consistent and correlated. First, each colour was cut to the first 3 letters, being the base colour (SILRED -> SIL).  We then selected the top 8 most relevant and frequent colours to keep as is, while the remaining niche colours were renamed as “Other”:  if(Bike\_Color == "BLK", "BLK",  if(Bike\_Color == "BLU", "BLU",  if(Bike\_Color == "GRY", "GRY",  if(Bike\_Color == "WHI", "WHI",  if(Bike\_Color== "RED", "RED",  if(Bike\_Color == "SIL", "SIL",  if(Bike\_Color == "GRN", "GRN",  if(Bike\_Color== "ONG", "ONG",  "Other"))))))))  This way, only 8 colours are focused on, which makes the dataset much more digestible. |
| BikeCost | real | Similarly to speed, cost must be binned into categories. To do this, BikeMake, BikeType, and BikeModel were clustered using k-Mean to predict BIKE\_COST by cluster.    From there, missing cost values are assigned based on the average of the predicted clusters:  if(missing(BIKE\_COST),  if(cluster == "cluster\_6", 1002.147,  if(cluster == "cluster\_5", 1023.908,  if(cluster == "cluster\_4", 967.665,  if(cluster == "cluster\_3", 982.215,  if(cluster == "cluster\_2", 962.203,  if(cluster == "cluster\_1", 997.911,  if(cluster == "cluster\_0", 996.582,  BIKE\_COST))))))), BIKE\_COST) |
| STATUS | nominal | 3 types: stolen, recovered, unknown |

Lastly, the data is highly string-based. Many of the attributes are in a string format, despite being represented nominally. While this is useful for association rules, we would still need a numeric dataset for outlier detection and clustering. As such, 2 datasets were created:

* Dataset 1: The first dataset will remain mostly consistent with the original, the difference being BikeSpeed and BikeCost will be binned into new nominal values.
  + BIKE\_SPEED: min=0, max=90
    - Slow: [0 - 15]
    - Medium: [16 - 30]
    - Fast: [31 - 45]
    - Very Fast: [46 - 60]
    - Performance: [61 - 90]

Formula:

if( filledBikeSpeed <= 15, "Slow",

if( filledBikeSpeed <= 30, "Medium",

if(filledBikeSpeed <= 45, "Fast",

if(filledBikeSpeed <= 60, "Very Fast",

if(filledBikeSpeed <= 90, "Performance", "Unknown")))))

* + BIKE\_COST: min=1, max=120000
    - Low: [1 - 500]
    - Medium: [501 - 1000]
    - High: [1001 - 2000]
    - Very High: [2001 - 3000]
    - Luxury: [3001 - 120000]

Formula:

if(BIKE\_COST <= 500, "Low",

if(BIKE\_COST <= 1000, "Medium",

if(BIKE\_COST <= 2000, "High",

if(BIKE\_COST <= 3000, "Very High",

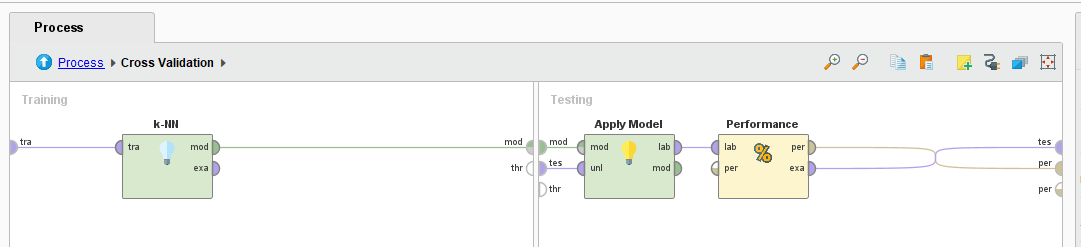
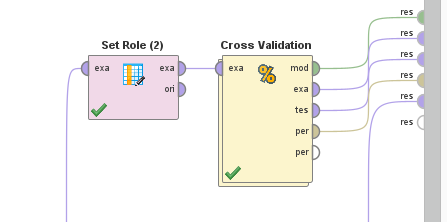
if(BIKE\_COST <= 120000, "Luxury", "Unknown")))))

* Dataset 2: , The second dataset is a little more complicated. Since we will need a dataset containing numeric values, we need to convert most nominal values to binary or ordinal. As such, there would be a significant number of binary attributes in this model.The dataset would also have to be normalized prior to establishing binary values. Here are all the new attributes that would be created:
  + STATUS: converted to binary [1s and 0s]
    - [Recovered, Stolen, Unknown]
  + Season: converted to ordinal [1-4]
    - [Spring, Summer, Fall, Winter]
  + OCC\_DOW: converted to ordinal [1-7]
    - [Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday]
  + HoursOfTheDay: converted to ordinal [1-4]
    - [Morning, Afternoon, Evening, Night]
  + PREMISES\_TYPE: converted to binary [1s and 0s]
    - [Apartment, Commercial, Educational, House, Other, Outside, Transit]
  + BikeType: converted to binary [1s and 0s]
    - [MT, RG, OT, RC, EL, TO, Others]
  + BikeModel: converted to binary [1s and 0s]
    - [Mountain Bikes, Road Bikes, Hybrid Bikes, Urban/Commuter Bikes, Speciality Bikes, Other Models]
  + BikeColour: converted to binary [1s and 0s]
    - [BLK, BLU, GRY, WHI, RED, SIL, GRN, ONG, Other]

# Modeling

**kNN**

The purpose of kNN is to compare data with its nearest neighbours, and depending on its similarities, kNN will predict the class based on the majority class around it. With this in mind, what we’re after is to predict whether a bike gets recovered based on the neighbours around it. For this, I will start off by setting my class label as “Status”. For this calculation, Status will remain nominal, since it will be predicted.



The current data will be cross validated with a test dataset for performance. This way we can see how accurate the program predicts the class.

Confusion Matrix:

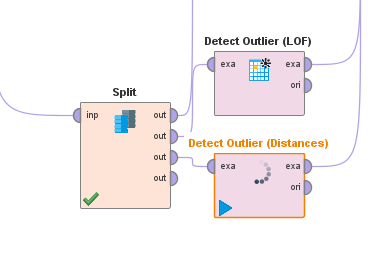
|  | true STOLEN | true RECOVERED | true UNKNOWN | class precision |
| --- | --- | --- | --- | --- |
| pred. STOLEN | 33200 | 356 | 537 | 97.38% |
| pred. RECOVERED | 16 | 4 | 2 | 18.18% |
| pred. UNKNOWN | 52 | 5 | 107 | 65.24% |
| class recall | 99.80% | 1.10% | 16.56% |  |

This is a moment where we can observe the class imbalance within Status. Because of the overwhelmingly high amount of stolen bikes, the model is compelled to predict stolen bikes for the majority of the dataset, because after all 97% precision is quite good. However, predicting 1% of the dataset can be tricky, and the model only managed to correctly predict 4 of the recovered bikes, with the remaining going to the other 2 values. This presents a challenge in predictive machine learning, and though we haven’t touched on handling class imbalances, it is important to recognize it none-the-less.

Despite the rather poor prediction for recovery, there were still rather accurate predictions across stolen and unknown. This means that the data is highly correlated with each other.

## Outlier Detection

The goal here is to find a subset of values within this dataset that do not confine to the consistency of the rest of the data. For this calculation, LOF and Distances were used as our method of Outlier Detection.



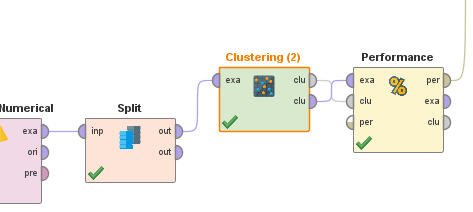
Outlier detection by far took the longest for the program to process.

It seems that the ReportHourGap had a higher likelihood of being predicted as an outlier. This is most likely due to the proportionally small number of report cases being declared multiple years after the incident.

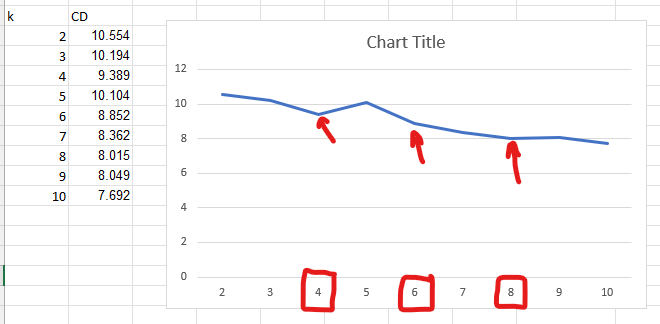
Some instances have a bike stolen in 1975, and only reported in 2023. Of course, this can also tell us about the correlation with ReportHourGap and the bike getting recovered, since after some period of time, it is unlikely the bike will ever be found again.

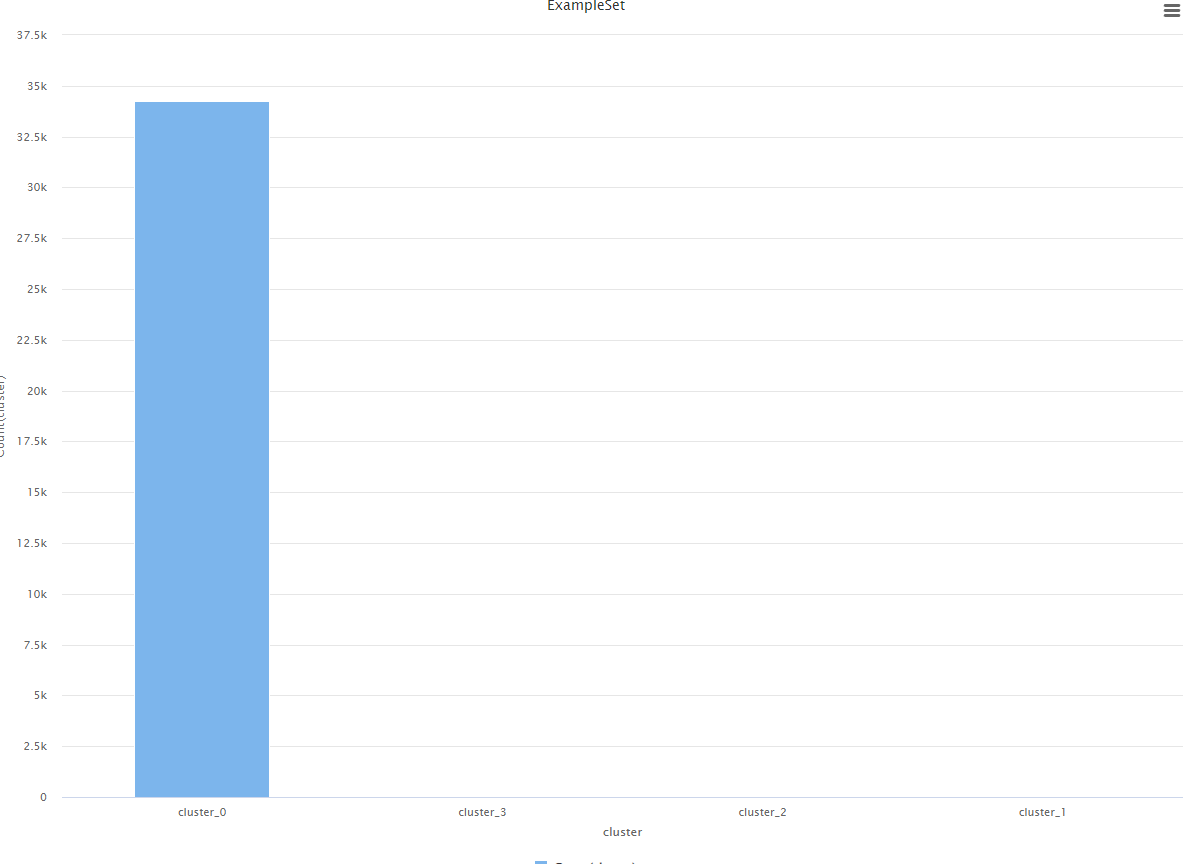
## Clustering

The first step in clustering was finding out the appropriate k value. For this, The numeric dataset was ran through a distance performance check under different numbers of clusters to see what the best fit for the k value would be



Using the elbow technique the plotted average centroid distances with k values from 2-10 suggested a strange assortment. The average within centroid distance seemed to spike upwards at k=5, and then proceed to drop back down at k=6. Due to the uncertainty of the elbow, I’ve decided to run 3 clusters with k=4, k=6, and k=8.



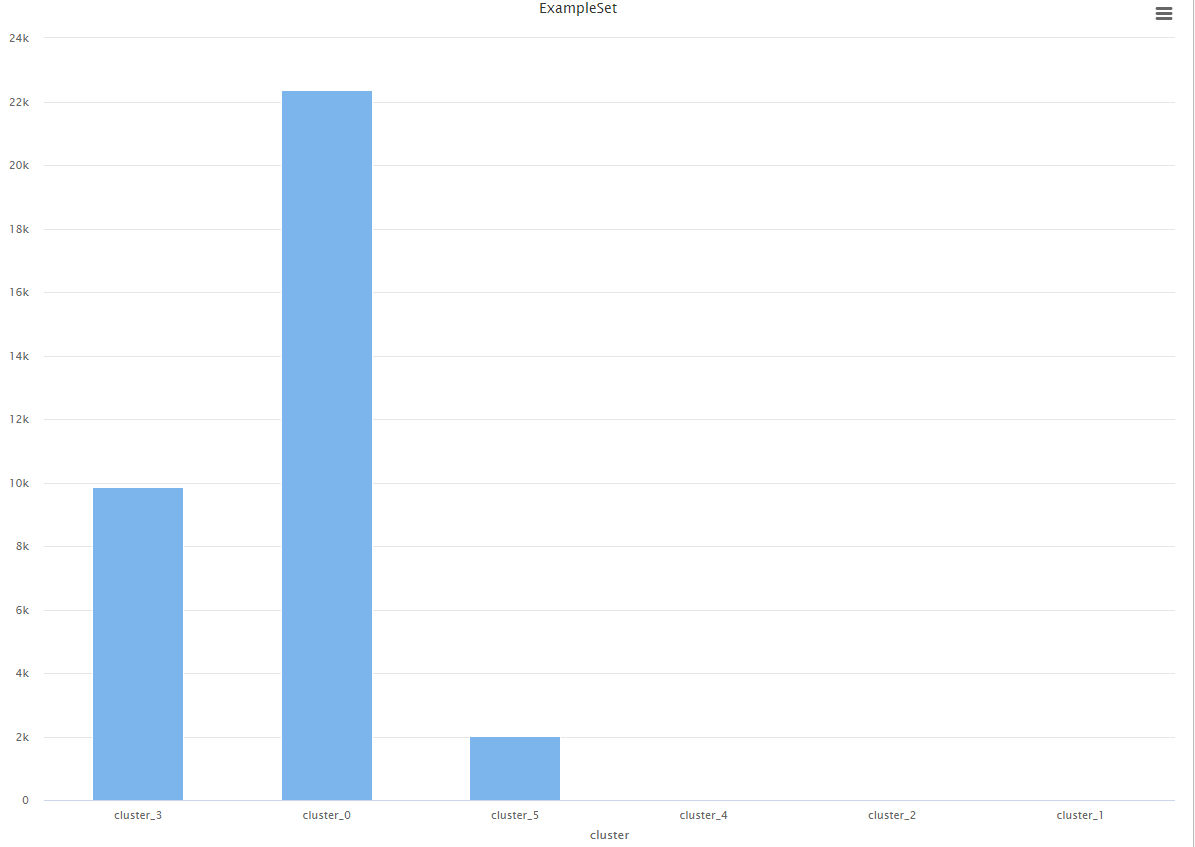
k=4: 

Cluster\_0: 34261

Cluster\_1: 2

Cluster\_2: 4

Cluster\_3: 12

k=6: 

Cluster\_0: 22373

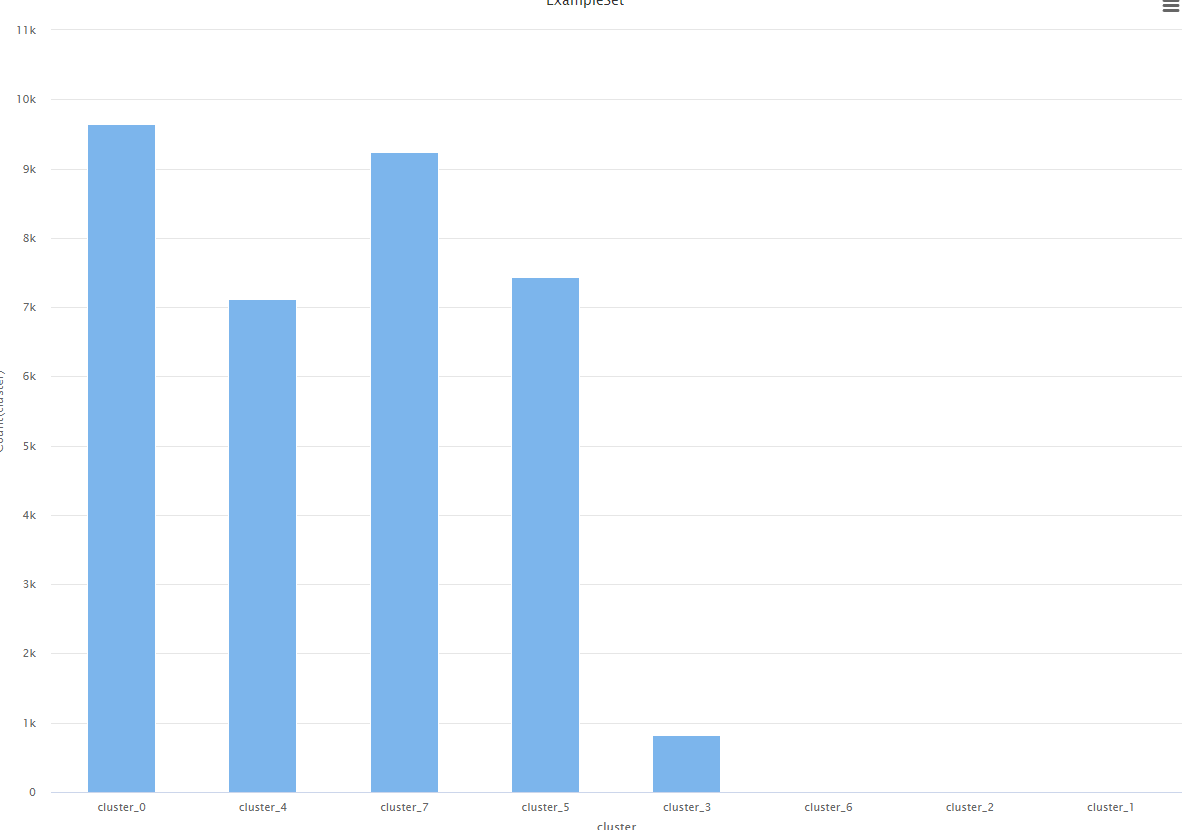
Cluster\_1: 2

Cluster\_2: 3

Cluster\_3: 9864

Cluster\_4: 12

Cluster\_5: 2025

k=8:

Cluster\_0: 9647

Cluster\_1: 2

Cluster\_2: 3

Cluster\_3: 817

Cluster\_4: 7120

Cluster\_5: 7438

Cluster\_6: 12

Cluster\_7: 9240

A recurring pattern may be observed here. For starters, clusters 1 and 2 are always at 2 and 3/4 instances, with one cluster having 12 instances at all times (3, 4, and 6). Cluster\_0 seems to always have the highest number of instances, and as the k increases, the number of instances seem to increase in the middle and last clusters. Therefore it’s likely the best clustering is from k=8.

Looking closer, it seems recovery is spread evenly across the clusters.

Cluster\_0: 106 Recovered / 9647 Total → 1.1%

Cluster\_3: 8 Recovered / 817 Total → 1%

Cluster\_4: 71 Recovered / 7120 Total → 1%

Cluster\_5: 84 Recovered / 7438 Total → 1.1%

Cluster\_7: 96 Recovered / 9240 Total → 1%

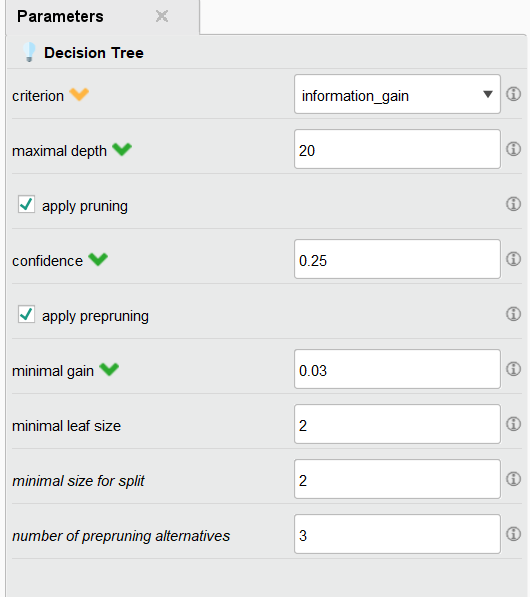
Based on the clusters formed, there seems to be no noticeable grouping of values. Each cluster seems to have a fairly balanced load between an attribute and the total instances.

That being said, there is a notable

## Classification(Decision Tree)

The process involves training a decision tree classifier on a dataset that has been split into a training set (70%) and a test set (30%).70% is used for training the model, and the remaining 30% for testing its performance.

**Decision Tree Configuration:**



* Maximal Depth: The tree is allowed to grow until it reaches a depth of 20
* Pruning: Both pre-pruning and post-pruning are applied to avoid overfitting:
  + Post-pruning with a confidence level set at 0.25 is used to trim the tree after it has been fully grown.

Interpretation of Results:

| BikeMakes = GI: STOLEN {STOLEN=1396, RECOVERED=12, UNKNOWN=1}  BikeMakes = NO: STOLEN {STOLEN=805, RECOVERED=4, UNKNOWN=0}  BikeMakes = OT: STOLEN {STOLEN=4640, RECOVERED=32, UNKNOWN=9}  BikeMakes = Others  | PREMISES\_TYPE = Apartment: STOLEN {STOLEN=2705, RECOVERED=31, UNKNOWN=39}  | PREMISES\_TYPE = Commercial: STOLEN {STOLEN=1315, RECOVERED=18, UNKNOWN=27}  | PREMISES\_TYPE = Educational: STOLEN {STOLEN=606, RECOVERED=5, UNKNOWN=9}  | PREMISES\_TYPE = House: STOLEN {STOLEN=2315, RECOVERED=28, UNKNOWN=81}  | PREMISES\_TYPE = Other: STOLEN {STOLEN=1563, RECOVERED=18, UNKNOWN=63}  | PREMISES\_TYPE = Outside  | | NewDivision = 1: STOLEN {STOLEN=528, RECOVERED=10, UNKNOWN=39}  | | NewDivision = 2  | | | BikeTypes = EL: STOLEN {STOLEN=6, RECOVERED=2, UNKNOWN=1}  | | | BikeTypes = MT  | | | | BikeColor = BLK: STOLEN {STOLEN=3, RECOVERED=0, UNKNOWN=3}  | | | | BikeColor = BLU: STOLEN {STOLEN=2, RECOVERED=0, UNKNOWN=0}  | | | | BikeColor = GRN: UNKNOWN {STOLEN=0, RECOVERED=0, UNKNOWN=3}  | | | | BikeColor = GRY: STOLEN {STOLEN=2, RECOVERED=0, UNKNOWN=0}  | | | | BikeColor = Other: STOLEN {STOLEN=10, RECOVERED=0, UNKNOWN=1}  | | | | BikeColor = RED: STOLEN {STOLEN=2, RECOVERED=0, UNKNOWN=1}  | | | BikeTypes = OT: STOLEN {STOLEN=2, RECOVERED=0, UNKNOWN=0}  | | | BikeTypes = Others: STOLEN {STOLEN=2, RECOVERED=0, UNKNOWN=1}  | | | BikeTypes = RC: STOLEN {STOLEN=3, RECOVERED=0, UNKNOWN=0}  | | | BikeTypes = RG: STOLEN {STOLEN=13, RECOVERED=0, UNKNOWN=0}  | | NewDivision = 3: STOLEN {STOLEN=270, RECOVERED=3, UNKNOWN=26}  | | NewDivision = 4: STOLEN {STOLEN=389, RECOVERED=4, UNKNOWN=16}  | | NewDivision = 5: STOLEN {STOLEN=2880, RECOVERED=74, UNKNOWN=71}  | PREMISES\_TYPE = Transit: STOLEN {STOLEN=252, RECOVERED=2, UNKNOWN=4}  BikeMakes = TR: STOLEN {STOLEN=1285, RECOVERED=7, UNKNOWN=3}  BikeMakes = UK: STOLEN {STOLEN=2348, RECOVERED=7, UNKNOWN=7} |
| --- |

The tree identifies **'BikeMakes'** as the most significant predictor

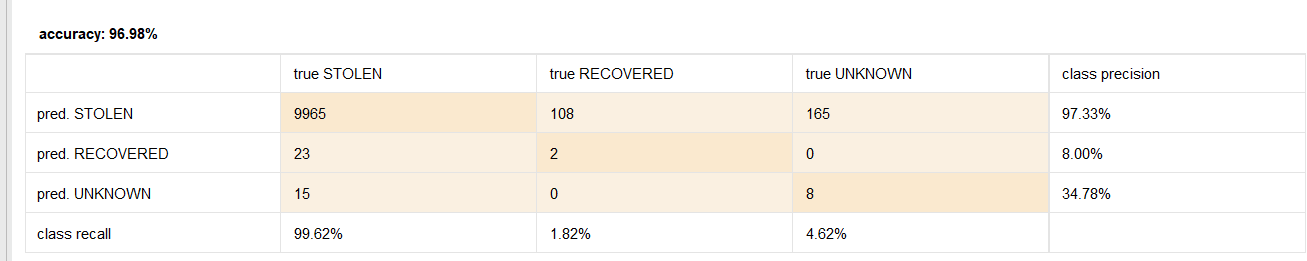
* Certain makes (OT, GI) are stolen more frequently than others.

**'Premise Type' and 'Division Area'** are the next most important factors

* 'NewDivision = 5' shows a higher number of thefts, indicating certain areas are more prone to bike theft.
* Within those higher-risk areas, certain types and colors of bikes may be more likely to be stolen.For instance, make 'OT' parked in apartments have a notably high number of thefts

**'BikeTypes' and 'BikeColor'** also contribute to risk, though to a lesser extent

**Model Evaluation:**



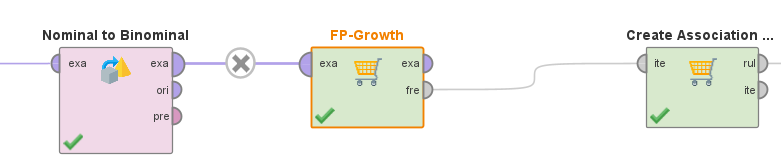
The confusion matrix shows the true vs. predicted classifications, with the accuracy of the model being 96.98%.

* However, the class recall and precision for categories other than 'STOLEN' (such as 'RECOVERED') are relatively low, that the model is much better at identifying 'STOLEN' bikes than at classifying 'RECOVERED' or 'UNKNOWN' statuses.

### 

## Association Rule Mining

**Modeling Process:**



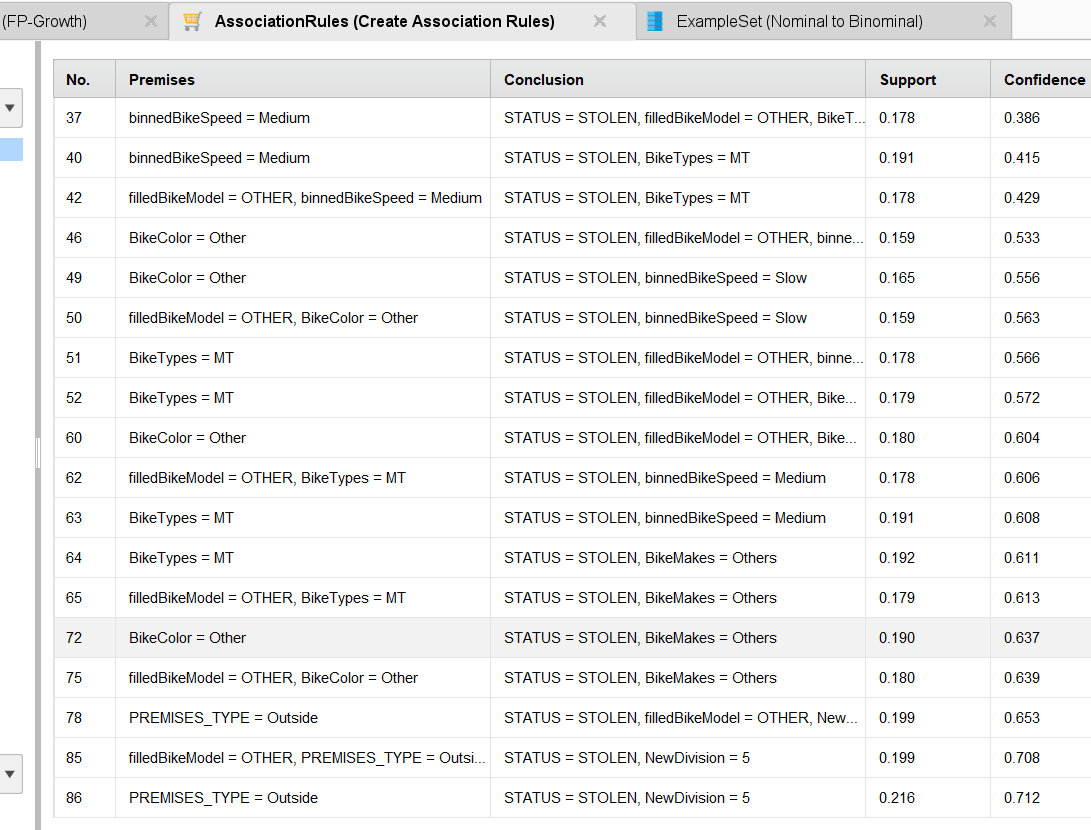
Data Transformation: Convert nominal attributes to binomial, possibly using the Nominal to Binomial operator to prepare the data for the FP-Growth operator.

FP-Growth Application: Apply the FP-Growth operator to find frequent itemsets, setting the minimum support to 0.2 to filter out less common patterns.

Rule Generation: Use the Create Association Rules operator with a life threshold set to 1 to generate rules from the frequent itemsets.

**Rule Interpretation:**

Focus on rules with lift greater than 1.1 and support greater than 0.15 as the most relevant to the status=”stolen“.



BikeSpeed (especially when binned as 'Medium' or 'Slow') is often part of the premises leading to a bike's status as 'STOLEN'.

NewDivision = 5 seems to be a significant factor, implying a higher risk of theft in this particular division.

BikeMakes, BikeModel, BikeColor, BikeType, and PREMISES\_TYPE are also part of the premises that contribute to the theft risk.

Seasonal Influence: The data suggest that the season has little to do with the likelihood of theft, whereas NewDivision = 5 and PREMISES\_TYPE = Outside are bigger factors.

**Associate Rules for Bike Stolen**

| premise | conclusion | support | confidence | lift |
| --- | --- | --- | --- | --- |
| NewDivision = 5, BikeMakes = Others, binnedBikeCost = Medium, PREMISES\_TYPE = Outside, binnedReportHourGap = fast | filledBikeModel = OTHER, STATUS = RECOVERED | 0.001545640128 | 0.1045364892 | 10.51189505 |
| BikeMakes = Others, binnedBikeSpeed = Medium, binnedBikeCost = Medium, PREMISES\_TYPE = Outside, binnedReportHourGap = fast | filledBikeModel = OTHER, STATUS = RECOVERED | 0.001079031788 | 0.1088235294 | 10.94298775 |
| filledBikeModel = OTHER, NewDivision = 5, BikeMakes = Others, binnedBikeCost = Medium, PREMISES\_TYPE = Outside, binnedReportHourGap = fast | STATUS = RECOVERED | 0.001545640128 | 0.111814346 | 10.4471769 |
| NewDivision = 5, BikeMakes = Others, binnedBikeCost = Medium, PREMISES\_TYPE = Outside, binnedReportHourGap = fast | STATUS = RECOVERED | 0.001662292213 | 0.1124260355 | 10.50432904 |
| BikeMakes = Others, binnedBikeSpeed = Medium, binnedBikeCost = Medium, PREMISES\_TYPE = Outside, binnedReportHourGap = fast | STATUS = RECOVERED | 0.001166520852 | 0.1176470588 | 10.99214618 |
| filledBikeModel = OTHER, BikeMakes = Others, binnedBikeSpeed = Medium, binnedBikeCost = Medium, PREMISES\_TYPE = Outside, binnedReportHourGap = fast | STATUS = RECOVERED | 0.001079031788 | 0.1217105263 | 11.37180912 |

**Associate Rules for Bike Recovered**

| premise | conclusion | support | confidence | lift |
| --- | --- | --- | --- | --- |
| filledBikeModel = OTHER, binnedBikeSpeed = Slow, BikeColor = Other | STATUS = STOLEN, BikeMakes = Others | 0.1078156897 | 0.6571276218 | 1.238820504 |
| binnedBikeSpeed = Slow, BikeColor = Other | STATUS = STOLEN, BikeMakes = Others | 0.112394284 | 0.6568944946 | 1.238381012 |
| filledBikeModel = OTHER, BikeColor = Other | STATUS = STOLEN, BikeMakes = Others | 0.1798775153 | 0.6391709845 | 1.204968556 |
| BikeColor = Other | STATUS = STOLEN, BikeMakes = Others | 0.1895304754 | 0.6367822849 | 1.200465366 |
| binnedBikeSpeed = Slow, BikeColor = Other | STATUS = STOLEN, filledBikeModel = OTHER, BikeMakes = Others | 0.1078156897 | 0.6301346514 | 1.262699696 |
| NewDivision = 5, BikeTypes = MT | STATUS = STOLEN, binnedBikeSpeed = Medium | 0.1094488189 | 0.6228011948 | 1.39225849 |
| filledBikeModel = OTHER, NewDivision = 5, BikeTypes = MT | STATUS = STOLEN, binnedBikeSpeed = Medium | 0.1014581511 | 0.6201426025 | 1.386315264 |
| BikeTypes = MT | STATUS = STOLEN, binnedBikeSpeed = Medium | 0.1907844853 | 0.6078795763 | 1.358901537 |
| filledBikeModel = OTHER, BikeTypes = MT | STATUS = STOLEN, binnedBikeSpeed = Medium | 0.1776027997 | 0.6064528978 | 1.355712228 |
| BikeColor = Other | STATUS = STOLEN, filledBikeModel = OTHER, BikeMakes = Others | 0.1798775153 | 0.6043503821 | 1.211031709 |
| BikeMakes = Others, BikeTypes = MT | STATUS = STOLEN, binnedBikeSpeed = Medium | 0.1184601925 | 0.5845445388 | 1.306736569 |
| filledBikeModel = OTHER, BikeMakes = Others, BikeTypes = MT | STATUS = STOLEN, binnedBikeSpeed = Medium | 0.1108194809 | 0.5839864761 | 1.305489032 |

* The rules with the highest lift scores are particularly informative. For instance, the rule with NewDivision = 5 and BikeMakes = Others, with specific conditions on PREMISES\_TYPE and binnedReportHourGap, leading to the bike being RECOVERED, indicates a strong relationship worth investigating.
* New Division: NewDivision = 5 appears in several rules with high lift values, indicating that this particular division may have effective recovery strategies or factors conducive to recovery.
* Bike Speed and Cost: binnedBikeSpeed = Medium and binnedBikeCost = Medium suggest that bikes of medium speed and cost are recovered with a relatively high frequency.
* Premises Type: The PREMISES\_TYPE = Outside condition is common across rules, which may indicate that bikes stolen from outdoor locations are more likely to be recovered, perhaps due to the visibility or public nature of these thefts.
* Report Hour Gap: binnedReportHourGap = fast implies that bikes reported stolen quickly ("fast" hour gap) are associated with recovery, highlighting the importance of timely theft reporting.

# Discussion of Results

kNN

The kNN results were fairly consistent. Though the class imbalance made it difficult to accurately predict a satisfying amount of bikes recovered, the accuracy overall wasn't the worst (refer to confusion matrix in Modeling).

Outlier Detection

Outlier detection, though not very practical for predicting whether a bike gets recovered or not, still allowed us to note the values within the dataset that off balance the predictive machine learning of other methods. Though outliers can be random and happen anywhere, they are important to understand in order to consider inaccuracies.

Clustering

The clusters were not so much of a success. Though It presented interesting trends with scaling k value, the clusters themselves were not very unique or interesting, having evenly spread attributes within each cluster. But a few things could be observed. For instance, the number of bikes stolen outside were significantly higher than other values within the total pool of instances within each cluster.

Decision Tree Conclusion & Associate rules mining:

Influence on Theft and Recovery:

* 'BikeMakes' (especially 'OT' and 'GI') are the top predictors for theft.
* 'NewDivision = 5' and outdoor 'Premise Type' locations experience more thefts.
* 'BikeTypes' and 'BikeColor' also affect theft incidence but to a lesser degree.
* Model accuracy is high at 96.98% for predicting 'STOLEN' bikes, but lower for 'RECOVERED' status.

Associative Rules for Theft and Recovery:

* Bikes with 'Medium' or 'Slow' speed and those in 'NewDivision = 5' are more prone to theft.
* Recovery is more likely for bikes in 'NewDivision = 5', especially when theft is promptly reported ('fast' hour gap).

Recommendations:

* Prioritize security for vulnerable bike makes and high-risk areas, especially 'NewDivision = 5'.
* Encourage rapid theft reporting and enhance security in outdoor premises to improve recovery rates.

To conclude, the likelihood of your bike getting recovered is proportionally very low compared to the vast amount of stolen bikes. More often than not, there is a lack of information, urgency, or convenience revolving around a stolen bike. But even with the inconsistency of data within datasets, there are still methods to study and predict the likelihood of recovery.