Los Angeles Crime Code Predictive Model

Tanush Kallem and Luke Flecker

Project Goal and Dataset Description

Dataset Description

- Dataset reflects incidents of crimes in Los Angeles from 2020
- Has 978,628 instances and 28 attributes
- Attributes include: data, time, location, demographics, weapon used, etc.
- Class is crime codes; the lower they are the worse the crime is

Project Goal

- Build a machine learning model to predict crime codes from LA crime data.
- Overcome data inconsistencies like location errors.
- Support law enforcement with accurate crime predictions.

Data Preprocessing

1. Data Cleaning:
    a. Attribute Removal
    b. Missing Value Correction
    c. Hidden Value Correction
2. Data Reduction:
    a. Remove Redundant Attributes
3. Data Modification

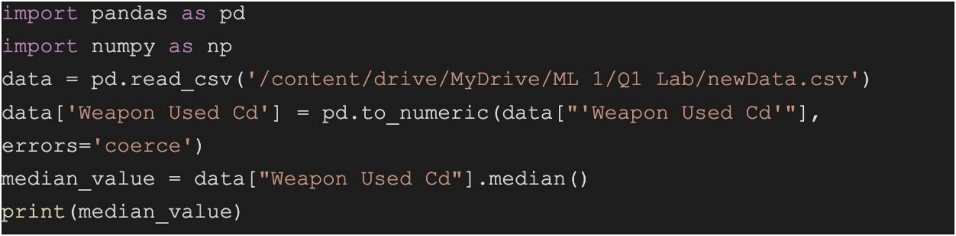
Attribute Removal

Before we started cleaning our data, we removed attributes that had a missing percentage greater than 70%.

- **Crm Cd 2, Crm Cd 3, Crm Cd, 4** were removed
  - 93%, 100%, 100% missing
  - Represent additional crimes to **Crm Cd 1**
- **Cross Street** was removed
  - Had 84% missing.
  - Represents the cross street of the rounded address
- **Part 1-2** was removed
  - Classified crimes in either 1 or 2, with 2 being worse crimes
  - Modified version of class

Missing Value Handling:

Weapon Used Cd: Due to this attribute having arbitrary codes, they were replaced with the median instead of the mean. This was done by first calculating the median through a python



```
import pandas as pd
import numpy as np
data = pd.read_csv('/content/drive/MyDrive/ML 1/Q1 Lab/newData.csv')
data['Weapon Used Cd'] = pd.to_numeric(data["'Weapon Used Cd'"],
errors='coerce')
median_value = data["Weapon Used Cd"].median()
print(median_value)
```

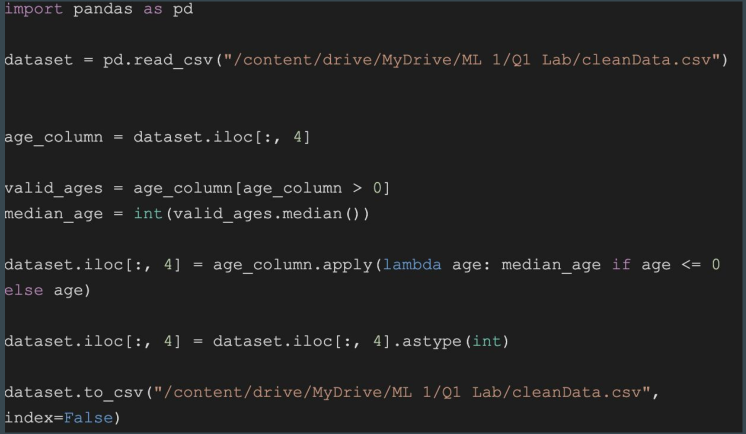
Then, using Weka's feature: ReplaceMissingWithUserConstant, we set the replacement value to the median we calculated.

Mocodes, Vict Sex, Vict Descent, Premis Desc, Weapons Used Cd, Weapon Desc all had missing values that were replaced with ReplaceMissingValues. This used the mean for numerical data and the mode for nominal data.

Hidden Value Handling

- **Vict Age:** Had hidden values that were 0, -1
- **Vict Sex:** Had hidden values that were ‘X’, ‘H’
- **Vict Desc:** Had hidden values that were ‘X’

For each of these attributes, a python script was created that would calculate the median then replace the hidden values with the median. Below is the python script for **Vict Age**



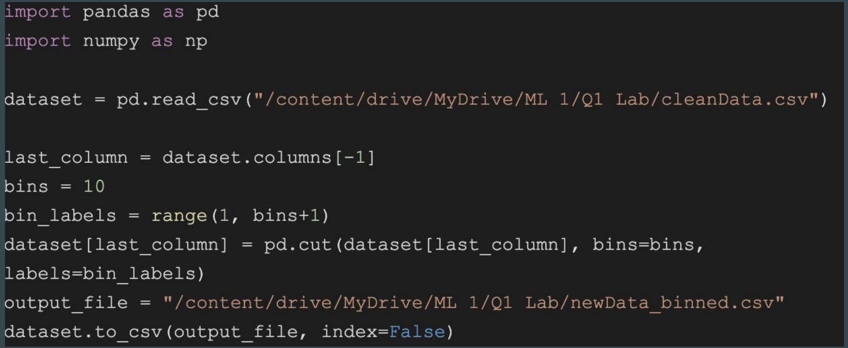
```
import pandas as pd

dataset = pd.read_csv("/content/drive/MyDrive/ML 1/Q1 Lab/cleanData.csv")


age_column = dataset.iloc[:, 4]

valid_ages = age_column[age_column > 0]
median_age = int(valid_ages.median())

dataset.iloc[:, 4] = age_column.apply(lambda age: median_age if age <= 0
else age)

dataset.iloc[:, 4] = dataset.iloc[:, 4].astype(int)

dataset.to_csv("/content/drive/MyDrive/ML 1/Q1 Lab/cleanData.csv",
index=False)
```

Removing Redundant Attributes

- **Crm Cd 1:** Removed due to the description stating it was the same as **Crm Cd**
- **AREA NAME, Crm Cd Desc, Premis Desc, Weapon Desc,Status Desc:** Removed as these are all descriptions of other attributes
- **Location:** Could be derived from **LAT** and **LONG**

Data Modification

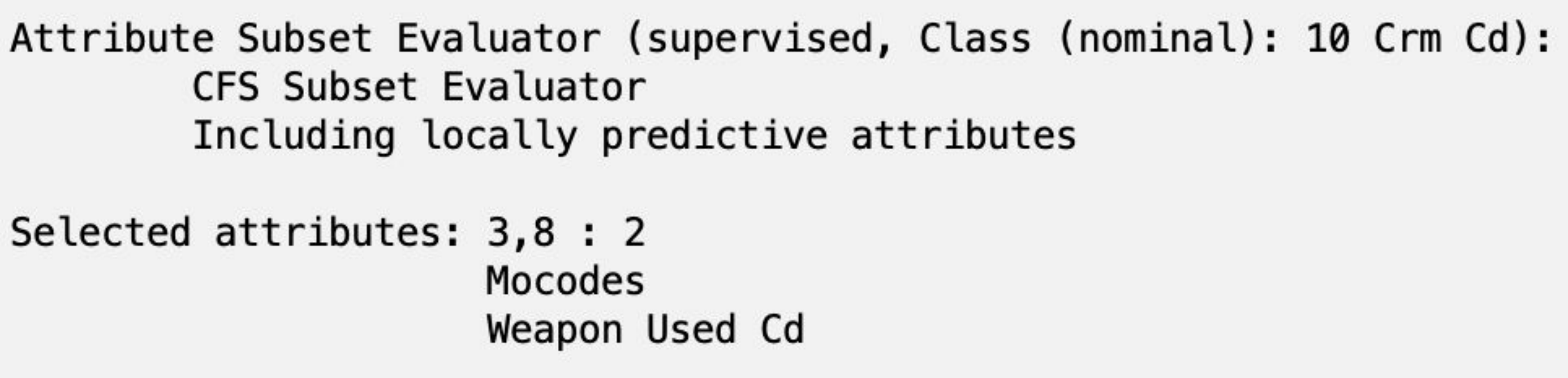
- **Mocodes:** Was given as a string, however, it represents a series of codes describing the crime. For compatibility StringToNominal was used to convert it to a nominal data type to perform further analysis.
- **Crm Cd:** This attribute had 91 unique codes which all varied in frequency, and was numerical. To have an accurate model, we used equal width binning to split the data into 10 bins, causing our class to be a rating of the crime from 1-10, with 1 being the worst. Then, NumerictoNominal was used on the bins.



```
import pandas as pd
import numpy as np

dataset = pd.read_csv("/content/drive/MyDrive/ML 1/Q1 Lab/cleanData.csv")

last_column = dataset.columns[-1]
bins = 10
bin_labels = range(1, bins+1)
dataset[last_column] = pd.cut(dataset[last_column], bins=bins,
labels=bin_labels)
output_file = "/content/drive/MyDrive/ML 1/Q1 Lab/newData_binned.csv"
dataset.to_csv(output_file, index=False)
```

Attribute Selection

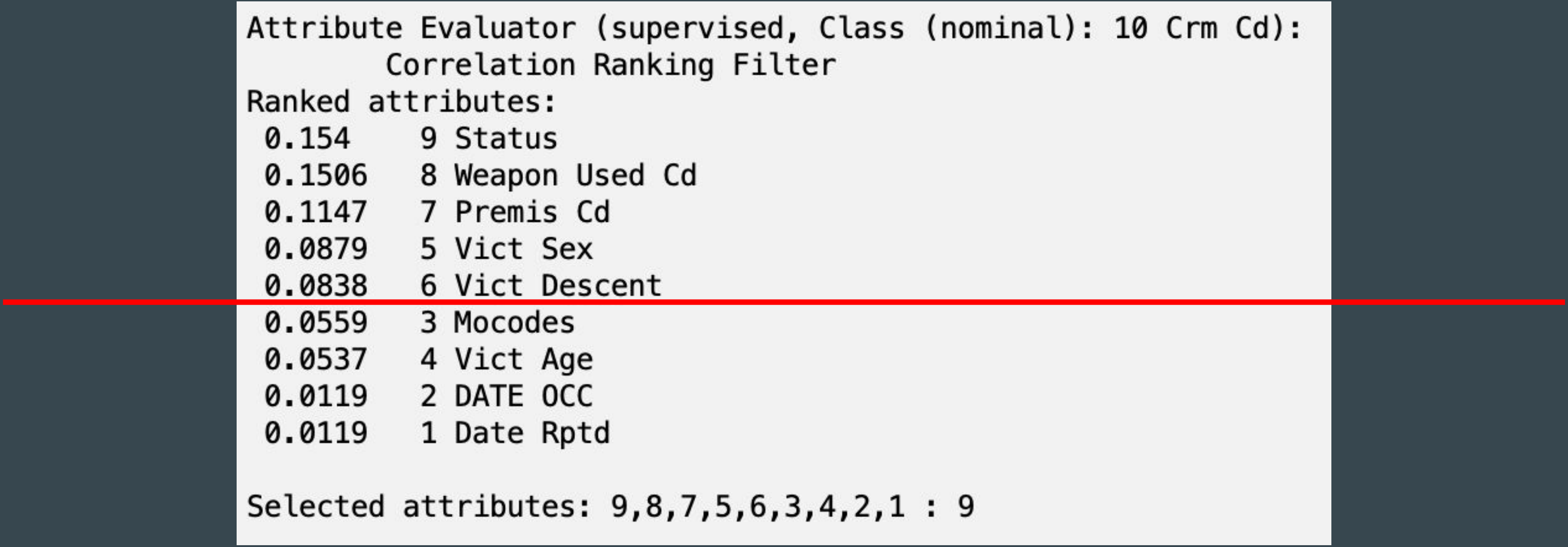
CfsSubsetEval:



```
Attribute Subset Evaluator (supervised, Class (nominal): 10 Crm Cd):
        CFS Subset Evaluator
        Including locally predictive attributes

Selected attributes: 3,8 : 2
                     Mocodes
                     Weapon Used Cd
```

Attribute Selection

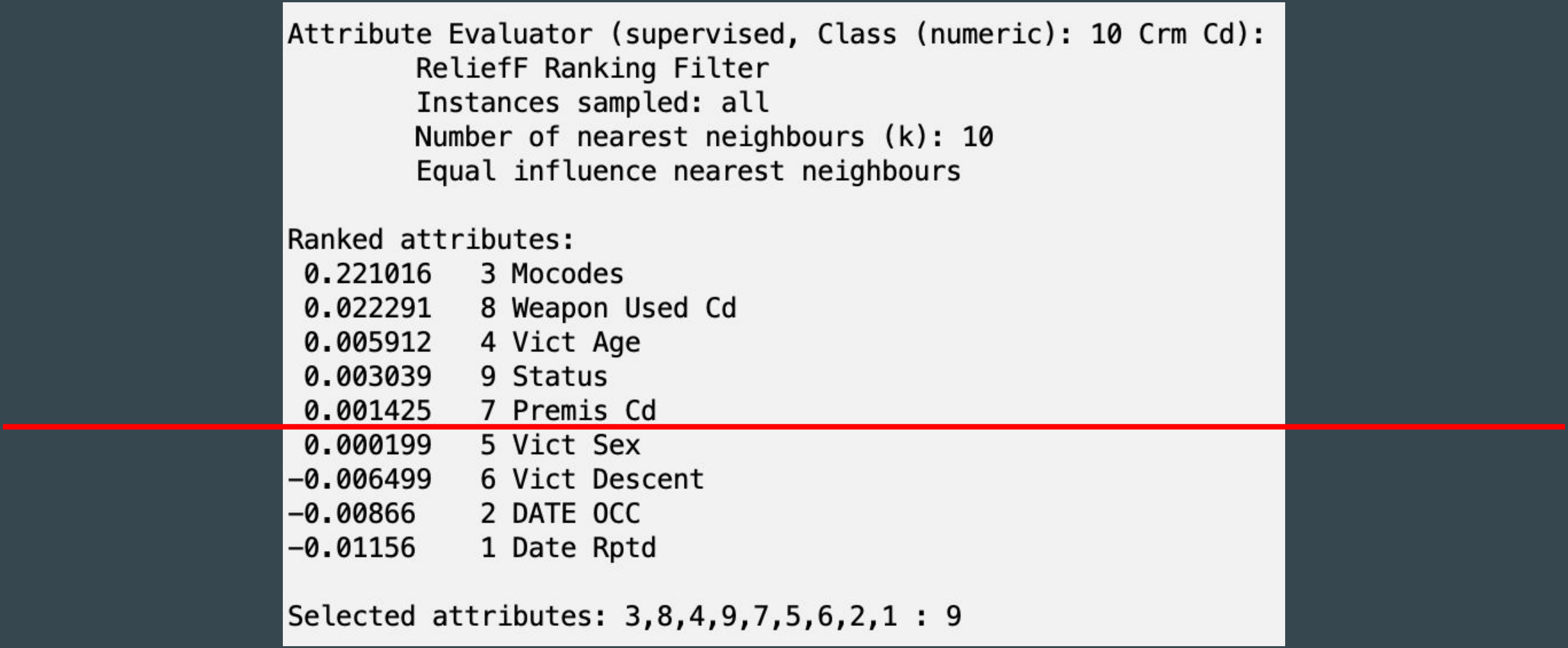
CorrelationAttributeEval: cutoff value of 0.06



```
Attribute Evaluator (supervised, Class (nominal): 10 Crm Cd):
        Correlation Ranking Filter
Ranked attributes:
 0.154     9 Status
 0.1506    8 Weapon Used Cd
 0.1147    7 Premis Cd
 0.0879    5 Vict Sex
 0.0838    6 Vict Descent
 0.0559    3 Mocodes
 0.0537    4 Vict Age
 0.0119    2 DATE OCC
 0.0119    1 Date Rptd

Selected attributes: 9,8,7,5,6,3,4,2,1 : 9
```

Attribute Selection

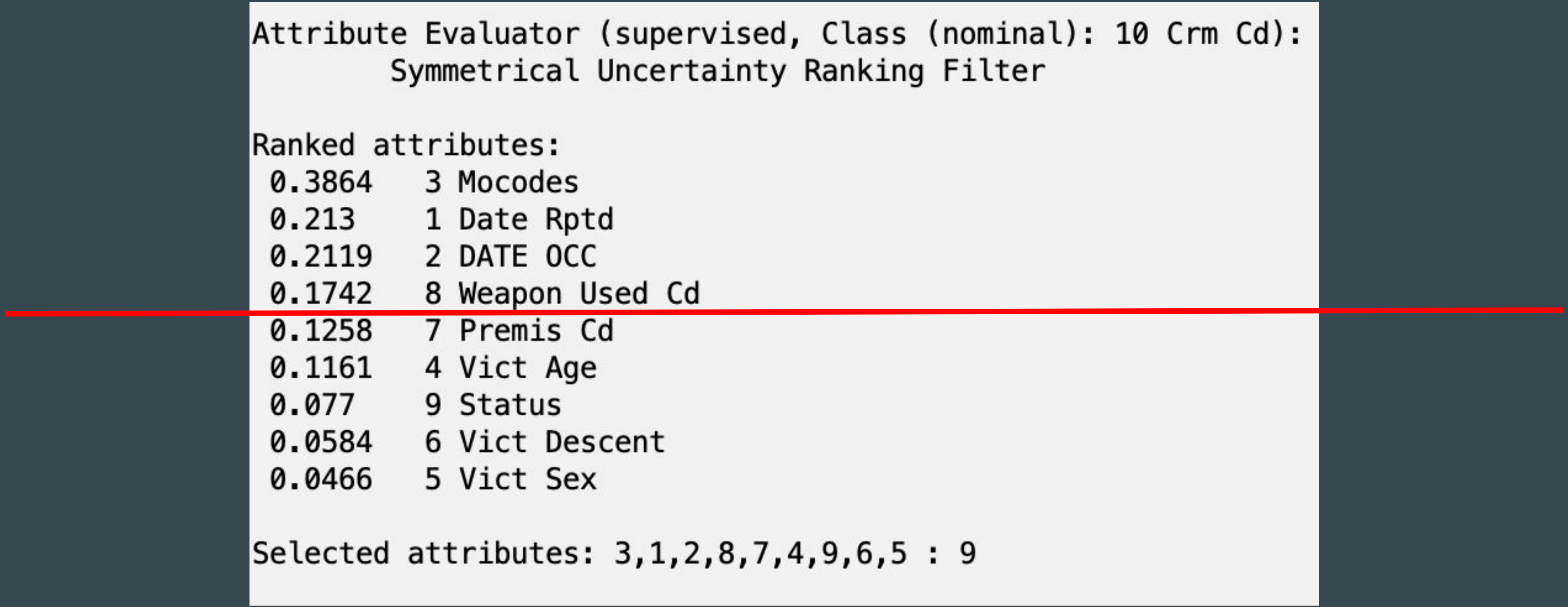
ReliefFAttributeEval: cutoff value of 0.001



```
Attribute Evaluator (supervised, Class (numeric): 10 Crm Cd):
        ReliefF Ranking Filter
        Instances sampled: all
        Number of nearest neighbours (k): 10
        Equal influence nearest neighbours

Ranked attributes:
 0.221016    3 Mocodes
 0.022291    8 Weapon Used Cd
 0.005912    4 Vict Age
 0.003039    9 Status
 0.001425    7 Premis Cd
 0.000199    5 Vict Sex
-0.006499    6 Vict Descent
-0.00866     2 DATE OCC
-0.01156     1 Date Rptd

Selected attributes: 3,8,4,9,7,5,6,2,1 : 9
```

Attribute Selection

SymmetricalUncertAttributeEval: cutoff value of 0.15



```
Attribute Evaluator (supervised, Class (nominal): 10 Crm Cd):
        Symmetrical Uncertainty Ranking Filter

Ranked attributes:
 0.3864    3 Mocodes
 0.213     1 Date Rptd
 0.2119    2 DATE OCC
 0.1742    8 Weapon Used Cd
 0.1258    7 Premis Cd
 0.1161    4 Vict Age
 0.077     9 Status
 0.0584    6 Vict Descent
 0.0466    5 Vict Sex

Selected attributes: 3,1,2,8,7,4,9,6,5 : 9
```

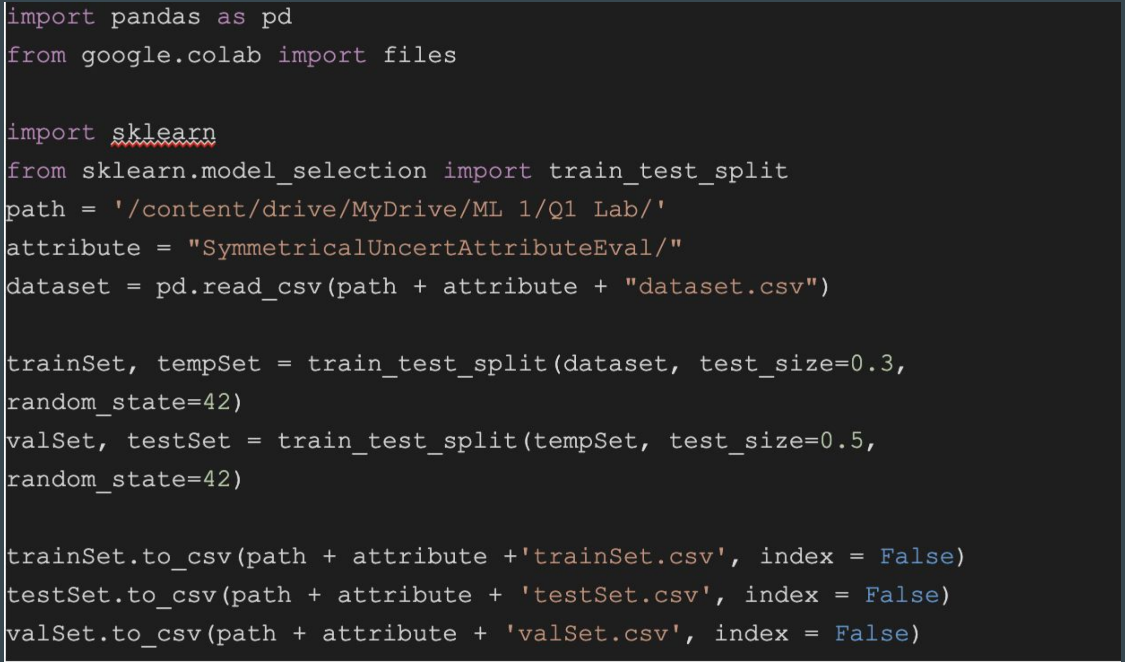
Attribute Selection

Personal Selection: The following attributes were removed

- **Status** : other external factors that could affect the status of the case such as the efficiency of the officers assigned it.
- **Weapon Used Cd** : Many of the weapons used were STRONG-ARM (HANDS, FIST, FEET OR BODILY FORCE,
- **Date Rptd** : **DATE OCC** would give us a more accurate timeline of when the crime happened compared to the date that it was reported.

Train-Test-Validation Split

A 70-15-15 split was performed on all the attribute selections seperately.

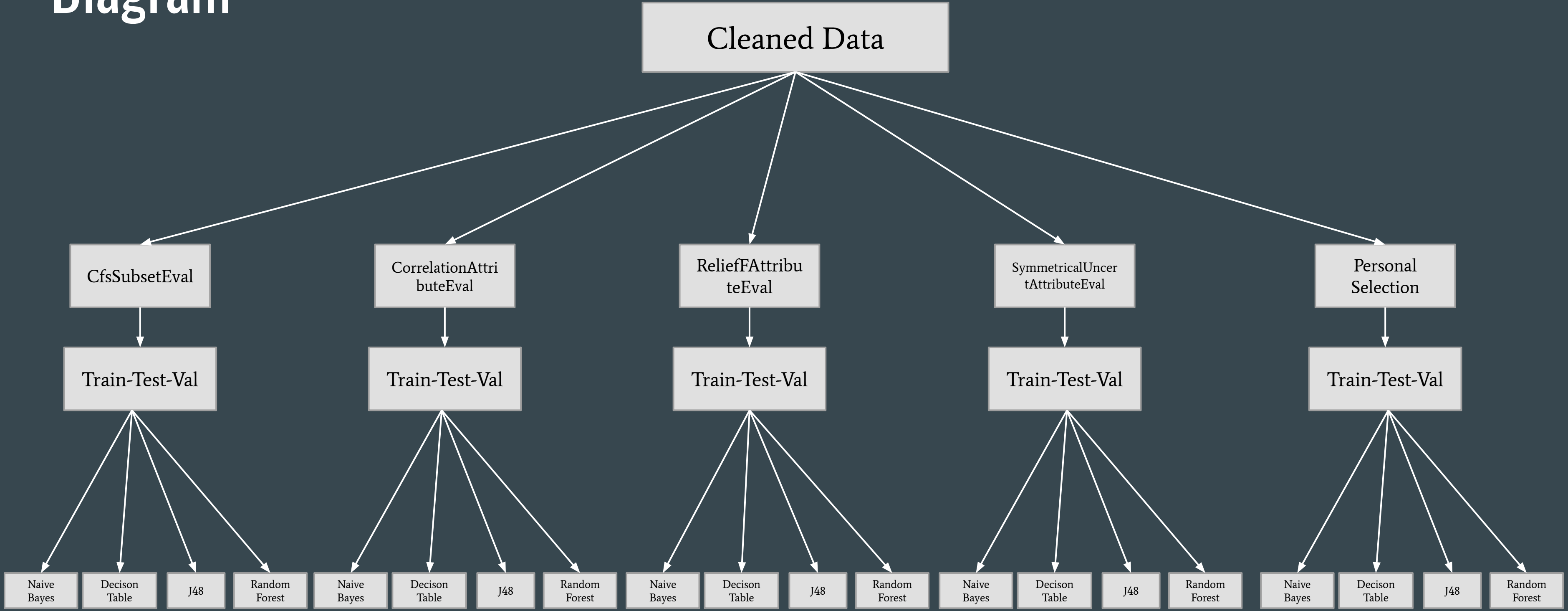


```
import pandas as pd
from google.colab import files

import sklearn
from sklearn.model_selection import train_test_split
path = '/content/drive/MyDrive/ML 1/Q1 Lab/'
attribute = "SymmetricalUncertAttributeEval/"
dataset = pd.read_csv(path + attribute + "dataset.csv")

trainSet, tempSet = train_test_split(dataset, test_size=0.3,
random_state=42)
valSet, testSet = train_test_split(tempSet, test_size=0.5,
random_state=42)

trainSet.to_csv(path + attribute +'trainSet.csv', index = False)
testSet.to_csv(path + attribute + 'testSet.csv', index = False)
valSet.to_csv(path + attribute + 'valSet.csv', index = False)
```

Classifiers

- **NaiveBayes Classifier:** Assumes independence between attributes and uses Bayes theorem to classify.
- **DecisionTable Classifier:** Uses a tabular representation of rules, where each rule is a unique combination of attribute values.
- **J48 Classifier:** Decision tree related to C4.5 that splits data based on the attribute that gives the biggest information gain.
- **Random Forest Classifier:** Creates multiple decision trees and combines them to improve classification.

Diagram

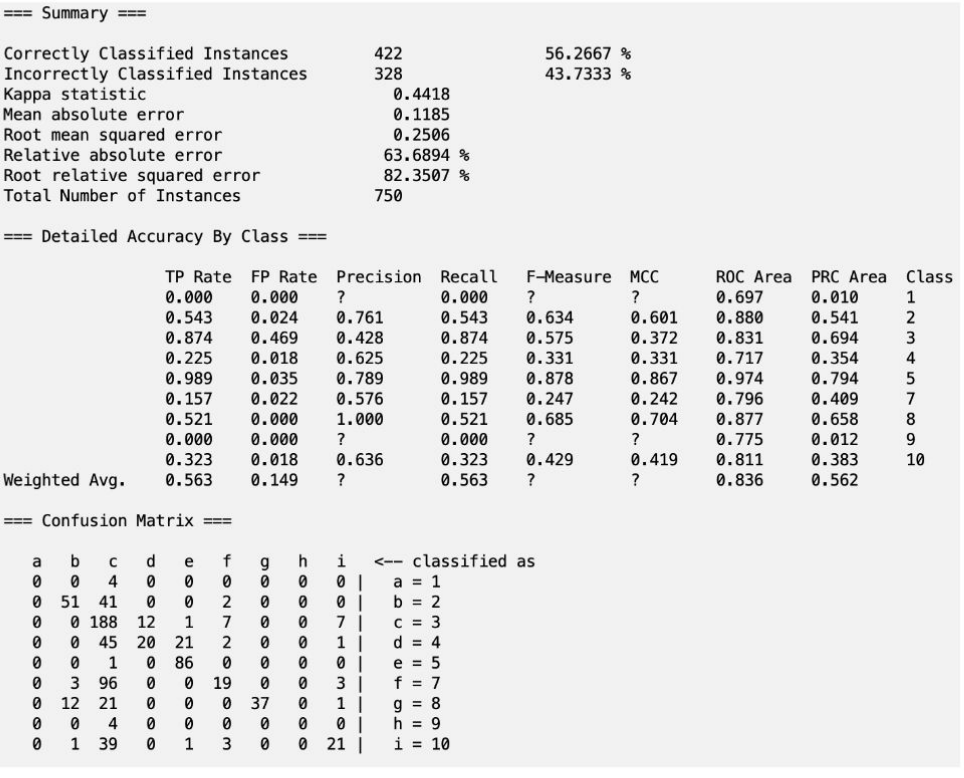


20 models

Top 5 Classifiers

1. ReliefFAttributeEval with J48 --- **56.27%**
2. CfsSubsetEval with J48 --- **56.13%**
3. ReliefFAttributeEval with Random Forest --- **55.20%**
4. CorrelationAttributeEval with J48 --- **54.80%**
5. CorrelationAttributeEval with Random Forest --- **51.87%**

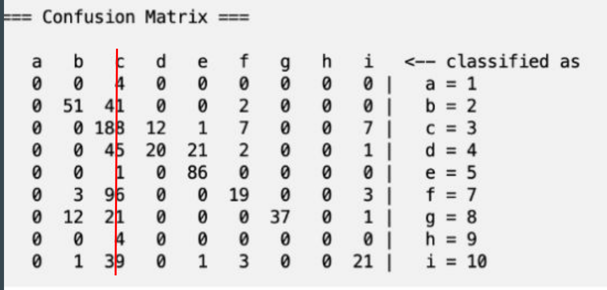
ReliefFAttributeEval with J48:



```
=== Summary ===

Correctly Classified Instances         422               56.2667 %
Incorrectly Classified Instances       328               43.7333 %
Kappa statistic                          0.4418
Mean absolute error                      0.1185
Root mean squared error                  0.2506
Relative absolute error                 63.6894 %
Root relative squared error             82.3507 %
Total Number of Instances              750

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.000    0.000    ?          0.000    ?          ?        0.697     0.010     1
                 0.543    0.024    0.761      0.543    0.634      0.601    0.880     0.541     2
                 0.874    0.469    0.428      0.874    0.575      0.372    0.831     0.694     3
                 0.225    0.018    0.625      0.225    0.331      0.331    0.717     0.354     4
                 0.989    0.035    0.789      0.989    0.878      0.867    0.974     0.794     5
                 0.157    0.022    0.576      0.157    0.247      0.242    0.796     0.409     7
                 0.521    0.000    1.000      0.521    0.685      0.704    0.877     0.658     8
                 0.000    0.000    ?          0.000    ?          ?        0.775     0.012     9
                 0.323    0.018    0.636      0.323    0.429      0.419    0.811     0.383     10
Weighted Avg.    0.563    0.149    ?          0.563    ?          ?        0.836     0.562

=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   <-- classified as
   0   0   4   0   0   0   0   0   0 |   a = 1
   0  51  41   0   0   2   0   0   0 |   b = 2
   0   0 188  12   1   7   0   0   7 |   c = 3
   0   0  45  20  21   2   0   0   1 |   d = 4
   0   0   1   0  86   0   0   0   0 |   e = 5
   0   3  96   0   0  19   0   0   3 |   f = 7
   0  12  21   0   0   0  37   0   1 |   g = 8
   0   0   4   0   0   0   0   0   0 |   h = 9
   0   1  39   0   1   3   0   0  21 |   i = 10
```

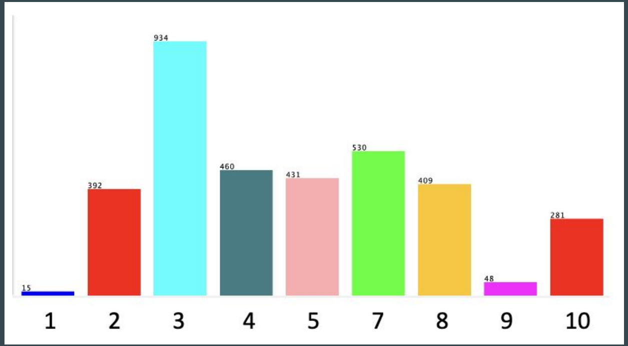
Analysis

The confusion matrix displayed a strong tendency for “3”, as seen below:



```
=== Confusion Matrix ===

  a   b   c   d   e   f   g   h   i   <-- classified as
  0   0   4   0   0   0   0   0   0 |   a = 1
  0  51  41   0   0   2   0   0   0 |   b = 2
  0   0 188  12   1   7   0   0   7 |   c = 3
  0   0  45  20  21   2   0   0   1 |   d = 4
  0   0   1   0  86   0   0   0   0 |   e = 5
  0   3  96   0   0  19   0   0   3 |   f = 7
  0  12  21   0   0   0  37   0   1 |   g = 8
  0   0   4   0   0   0   0   0   0 |   h = 9
  0   1  39   0   1   3   0   0  21 |   i = 10
```

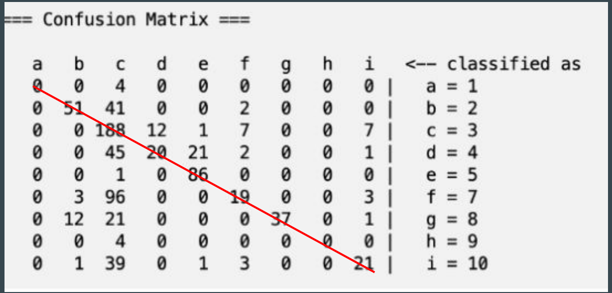
This is explained by the abundance of “3” instances compared to the dataset:



Taking “3” out, increased the accuracy to 97.7%

Analysis

The confusion matrix also shows that the inaccurate results the model predicted were close to the actual value (ignoring ‘3’):



```
=== Confusion Matrix ===

  a   b   c   d   e   f   g   h   i   <-- classified as
  0   0   4   0   0   0   0   0   0 |   a = 1
  0  51  41   0   0   2   0   0   0 |   b = 2
  0   0 188  12   1   7   0   0   7 |   c = 3
  0   0  45  20  21   2   0   0   1 |   d = 4
  0   0   1   0  86   0   0   0   0 |   e = 5
  0   3  96   0   0  19   0   0   3 |   f = 7
  0  12  21   0   0   0  37   0   1 |   g = 8
  0   0   4   0   0   0   0   0   0 |   h = 9
  0   1  39   0   1   3   0   0  21 |   i = 10
```

This is also seen by the classification having a Mean Absolute Error of 0.1185.

Future Improvements

- Discretization: Further discretizing the data to simplify the classification problem could result in a significant improvement in accuracy.
- Resampling Techniques: Addressing the imbalanced nature of the dataset through techniques like Synthetic Minority Oversampling Technique (SMOTE)
- Using PCA: Create new attributes that are more closely correlated to the class
- Regression: Try using regression to predict actual crime codes

References

Data: https://catalog.data.gov/dataset/crime-data-from-2020-to-present

Weka Rules: http://weka.sourceforge.net/doc.dev/weka/classifiers/rules/DecisionTable.html