# Recidivism Predictive Model

James Beck, Jia Mody

# 01

# Objective

# Objective

- Bureau of Justice Stat: 68% of criminals commit repeat offences in 3 years
- Comprehensive risk assessment model
- Identify potential repeat offenders
- Public safety
- Support high risk offenders

# 02

## Our Dataset

# Our Dataset

### Overview

Data.gov
Published by U.S. DOJ

### Instances

25,825 instances, each instance represents one prisoner

### Attributes

49 attributes, including gender, dependance, drug test positives, etc

# Attribute Examples

1

**Prior_Revocations_Parole**

Have there been previous violations of parole terms?

2

**Dependents**

Number of people dependent on inmate’s income

3

**Prior_Arrest_Episodes_Felony**

# of arrests due to felonies

# Class Breakdown

1

### 1st Year

7724 committed a crime within 1 year of release (29.9%)

### 2nd Year

4567 committed a crime within 2 years of release (17.7%)

2

3

### 3rd Year

2613 committed a crime within 3 years of release

(10.1%)

### Never

10,931 did not commit a second crime within 3 years. (42.3%)

NO

*right skew

# 03

## Pre-processing

# Pre-Processing

1

**Missing Values**

Filled in missing values

2

**Derived Class**

Created a derived class attribute from 4 potential classes

3

**Normalizing+Changing**

Normalized quantitative variables + attribute alterations

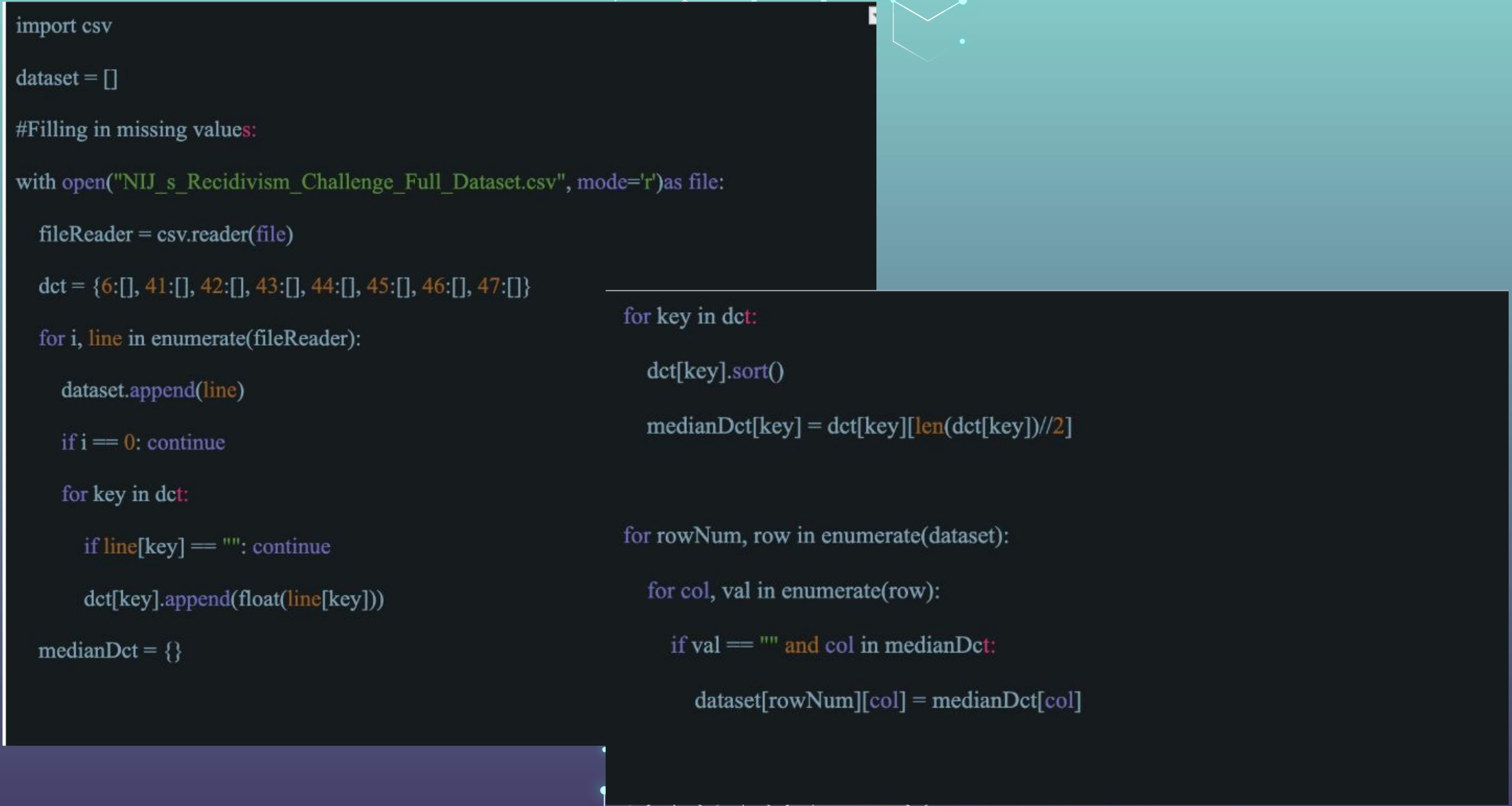
4

**Test/Train**

Created Test/Train split

# Missing Values

- ❏ No attributes were missing >70% values
- ❏ Replaced missing qualitative values with mode
- ❏ Replaced missing quantitative values with median (difficulties in WEKA)
- ❏ Python Script

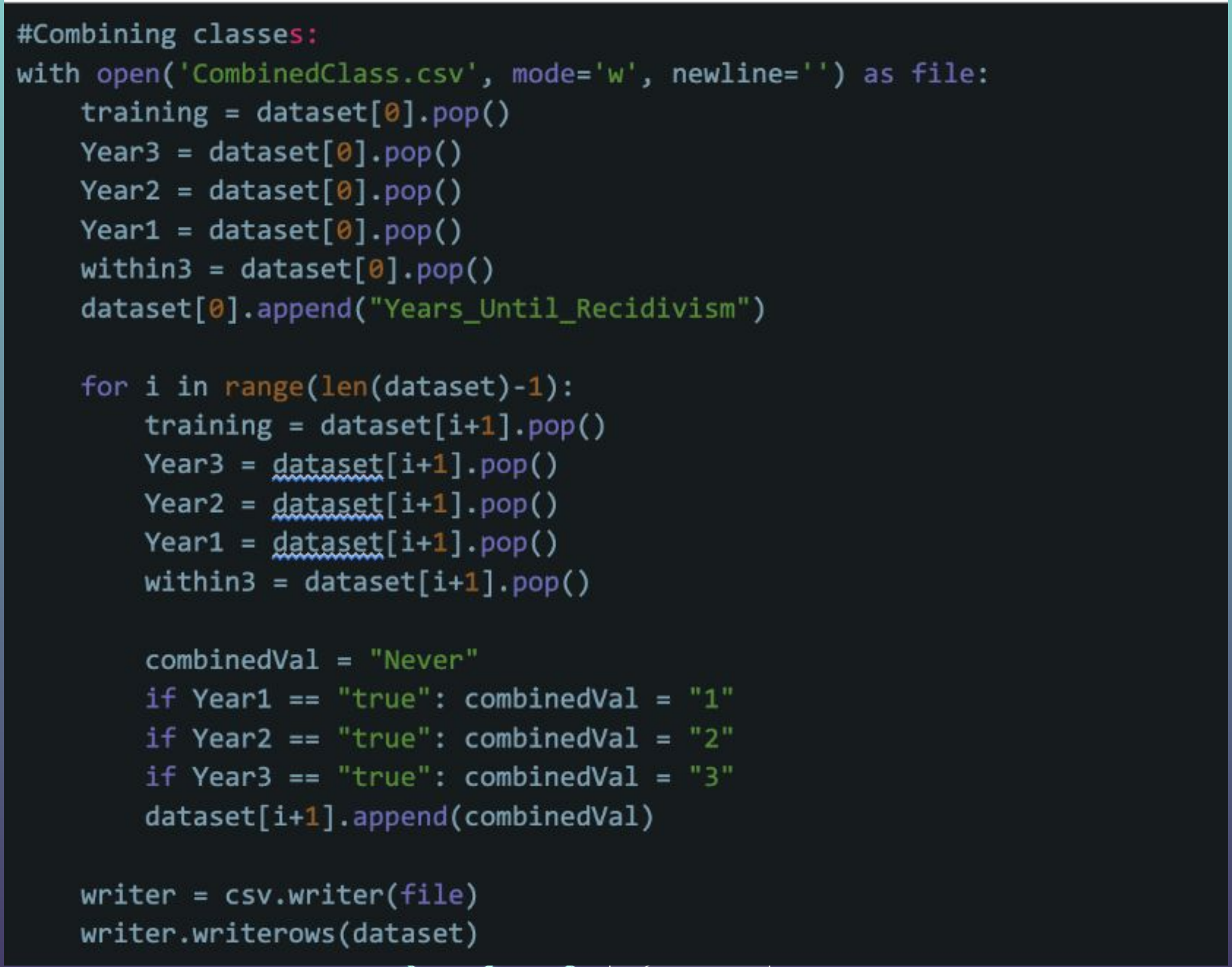


```
import csv

dataset = []

#Filling in missing values:

with open("NIJ_s_Recidivism_Challenge_Full_Dataset.csv", mode='r')as file:

    fileReader = csv.reader(file)

    dct = {6:[], 41:[], 42:[], 43:[], 44:[], 45:[], 46:[], 47:[]}

    for i, line in enumerate(fileReader):

        dataset.append(line)

        if i == 0: continue

        for key in dct:

            if line[key] == "": continue

            dct[key].append(float(line[key]))

    medianDct = {}
```

```
for key in dct:

    dct[key].sort()

    medianDct[key] = dct[key][len(dct[key])//2]


for rowNum, row in enumerate(dataset):

    for col, val in enumerate(row):

        if val == "" and col in medianDct:

            dataset[rowNum][col] = medianDct[col]
```

# Derived Class

- ❑ 4 potential class attributes
- ❑ Combined attributes into single class with following labels:
  - ❑ "1" – Arrested within 1 year of release
  - ❑ "2" – Arrested within 2 years of release
  - ❑ "3" – Arrested within 3 years of release
  - ❑ "Never" – No arrest within 3 years of release



```
#Combining classes:
with open('CombinedClass.csv', mode='w', newline='') as file:
    training = dataset[0].pop()
    Year3 = dataset[0].pop()
    Year2 = dataset[0].pop()
    Year1 = dataset[0].pop()
    within3 = dataset[0].pop()
    dataset[0].append("Years_Until_Recidivism")

    for i in range(len(dataset)-1):
        training = dataset[i+1].pop()
        Year3 = dataset[i+1].pop()
        Year2 = dataset[i+1].pop()
        Year1 = dataset[i+1].pop()
        within3 = dataset[i+1].pop()

        combinedVal = "Never"
        if Year1 == "true": combinedVal = "1"
        if Year2 == "true": combinedVal = "2"
        if Year3 == "true": combinedVal = "3"
        dataset[i+1].append(combinedVal)

    writer = csv.writer(file)
    writer.writerows(dataset)
```

# Normalizing and Changing

### Normalizing

❑ WEKA Normalize Filter

❑ Exception: Residence_PUMA

### Altering

❑ Nominal→Numerical

# Test/Train

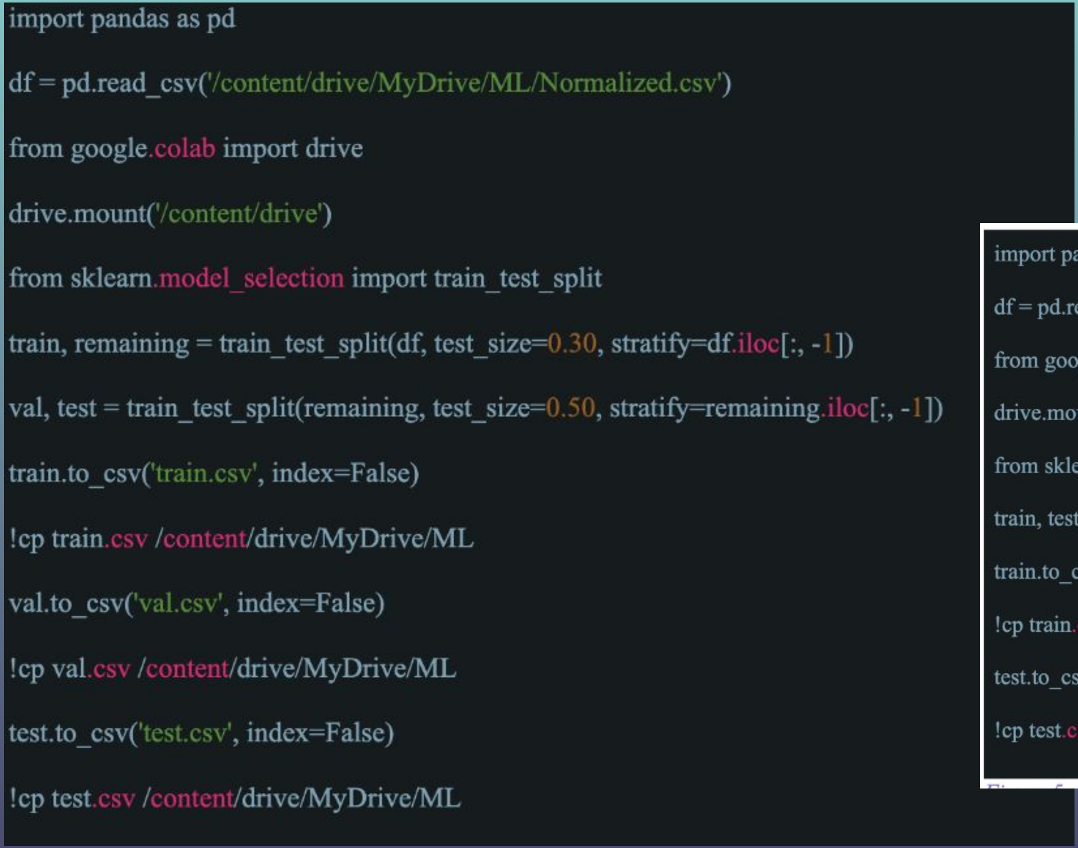
❑ Python Script

❑ Originally 70/15/15, but validation was not working well

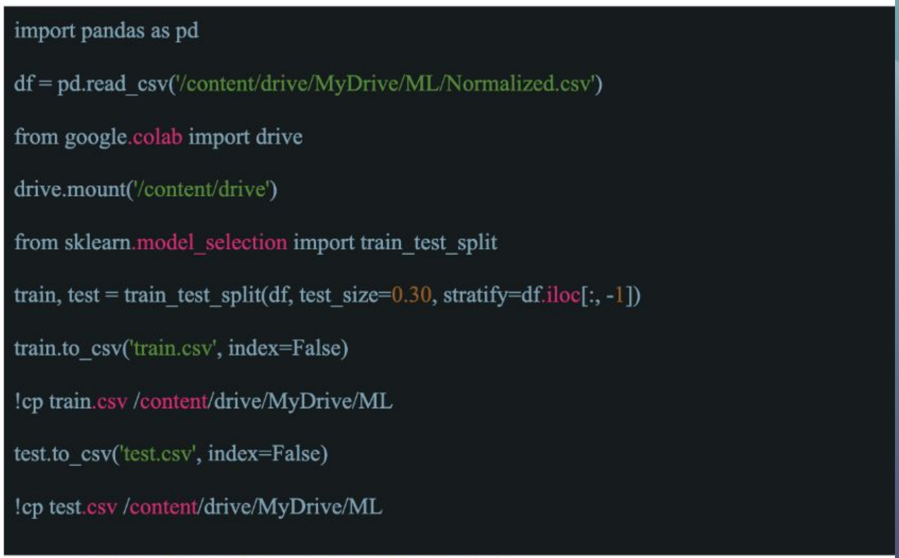
❑ Changed to 70/30 train test

# Test/Train

- Training:
  - 1 yr-5407/18084 (29.9%)
  - 2 yrs-3197/18084 (17.7%)
  - 3 yrs-1829/18084 (10.1%)
  - Never-7651/18084 (42.3%)
- Test:
  - 1 yr-2317/7751 (29.9%)
  - 2 yrs-1370/7751 (17.7%)
  - 3 yrs-784/7751 (10.1%)
  - Never-3280/7751 (42.3%)



```
import pandas as pd

df = pd.read_csv('/content/drive/MyDrive/ML/Normalized.csv')

from google.colab import drive

drive.mount('/content/drive')

from sklearn.model_selection import train_test_split

train, remaining = train_test_split(df, test_size=0.30, stratify=df.iloc[:, -1])

val, test = train_test_split(remaining, test_size=0.50, stratify=remaining.iloc[:, -1])

train.to_csv('train.csv', index=False)

!cp train.csv /content/drive/MyDrive/ML

val.to_csv('val.csv', index=False)

!cp val.csv /content/drive/MyDrive/ML

test.to_csv('test.csv', index=False)

!cp test.csv /content/drive/MyDrive/ML
```



```
import pandas as pd

df = pd.read_csv('/content/drive/MyDrive/ML/Normalized.csv')

from google.colab import drive

drive.mount('/content/drive')

from sklearn.model_selection import train_test_split

train, test = train_test_split(df, test_size=0.30, stratify=df.iloc[:, -1])

train.to_csv('train.csv', index=False)

!cp train.csv /content/drive/MyDrive/ML

test.to_csv('test.csv', index=False)

!cp test.csv /content/drive/MyDrive/ML
```

*Figure 5: Python Script Demonstrating Train Test Split*

# 04

## Attribute Selection

# Attribute Selection

**CorrelationAttributeEval**

Cutoff of 0.1

**InfoGainAttributeEval**

Cutoff of 0.25

**OneRAttributeEval**

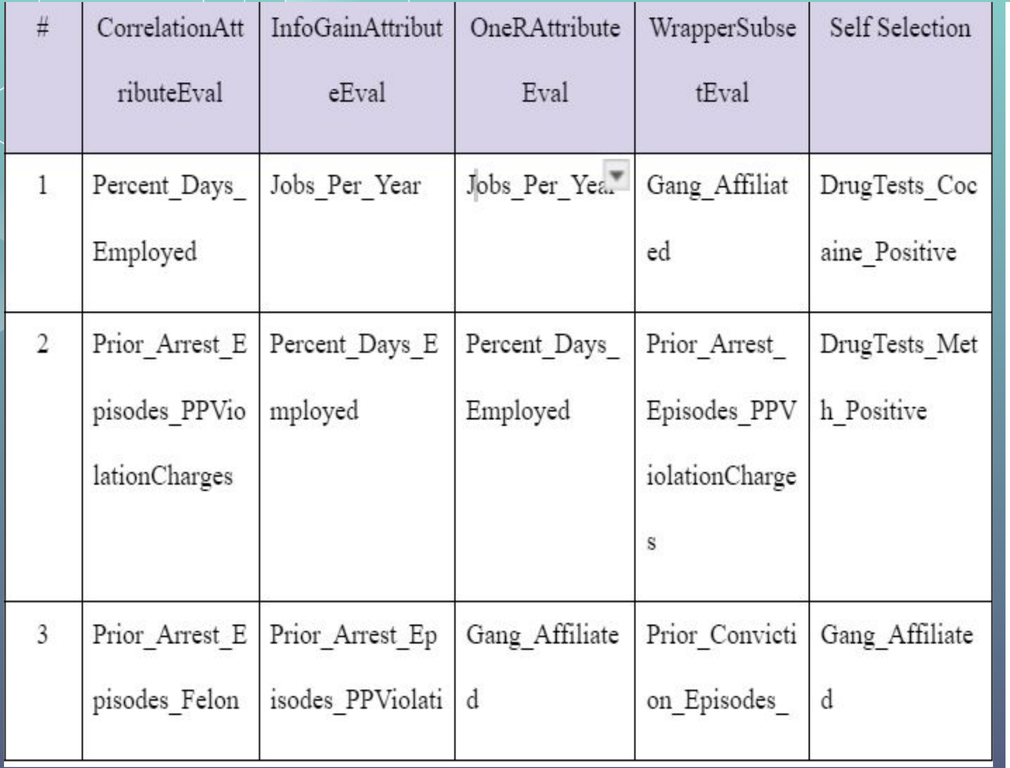
Cutoff of 43.95

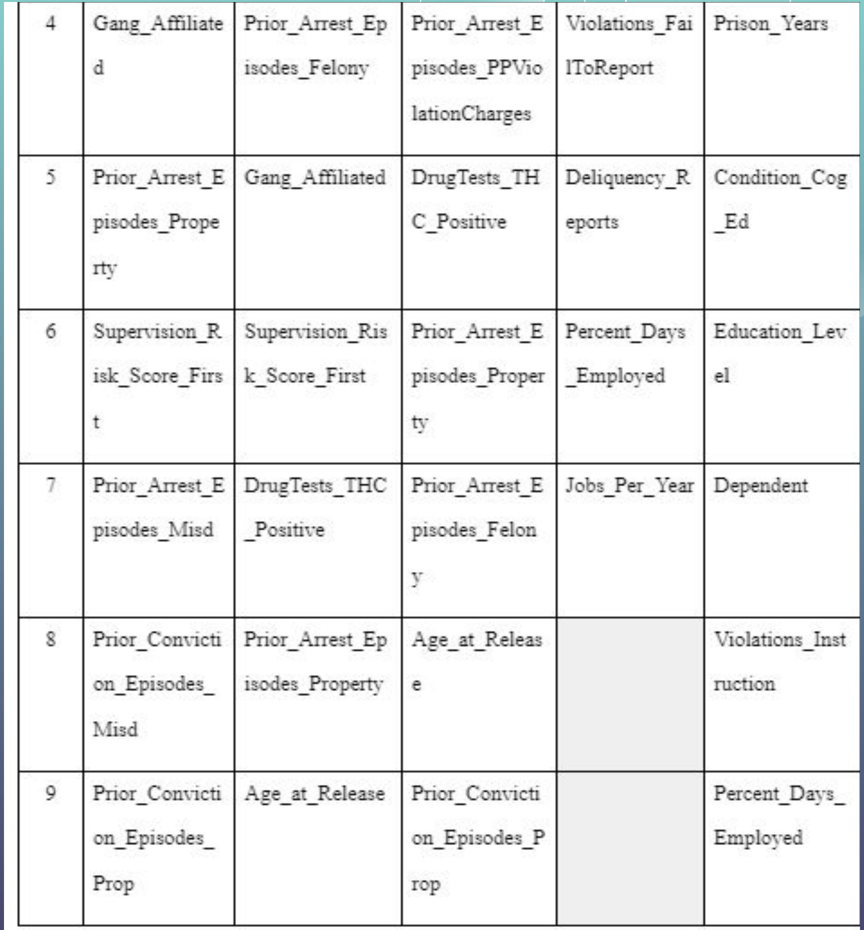
**WrapperSubsetEval**

7 Attributes

**Self Selection**

9 Attributes





| # | CorrelationAttributeEval | InfoGainAttributeEval | OneRAttributeEval | WrapperSubsetEval | Self Selection |
|---|---|---|---|---|---|
| 1 | Percent_Days_Employed | Jobs_Per_Year | Jobs_Per_Year | Gang_Affiliated | DrugTests_Cocaine_Positive |
| 2 | Prior_Arrest_Episodes_PPViolationCharges | Percent_Days_Employed | Percent_Days_Employed | Prior_Arrest_Episodes_PPViolationCharges | DrugTests_Meth_Positive |
| 3 | Prior_Arrest_Episodes_Felon | Prior_Arrest_Episodes_PPViolati | Gang_Affiliated | Prior_Conviction_Episodes_ | Gang_Affiliated |
| 4 | Gang_Affiliated | Prior_Arrest_Episodes_Felony | Prior_Arrest_Episodes_PPViolationCharges | Violations_FailToReport | Prison_Years |
| 5 | Prior_Arrest_Episodes_Property | Gang_Affiliated | DrugTests_THC_Positive | Deliquency_Reports | Condition_Cog_Ed |
| 6 | Supervision_Risk_Score_First | Supervision_Risk_Score_First | Prior_Arrest_Episodes_Property | Percent_Days_Employed | Education_Level |
| 7 | Prior_Arrest_Episodes_Misd | DrugTests_THC_Positive | Prior_Arrest_Episodes_Felony | Jobs_Per_Year | Dependent |
| 8 | Prior_Conviction_Episodes_Misd | Prior_Arrest_Episodes_Property | Age_at_Release | | Violations_Instruction |
| 9 | Prior_Conviction_Episodes_Prop | Age_at_Release | Prior_Conviction_Episodes_Prop | | Percent_Days_Employed |

# Classification Algorithms

1

J48
Classification via decision trees

2

NaiveBayes
Baye’s formula

3

OneR
Rule Based

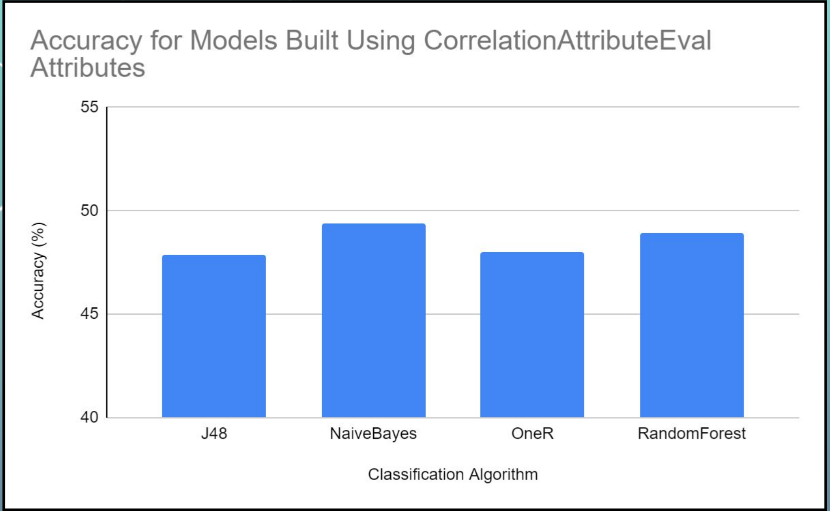
4

RandomForest
Decision trees through random subsets of data

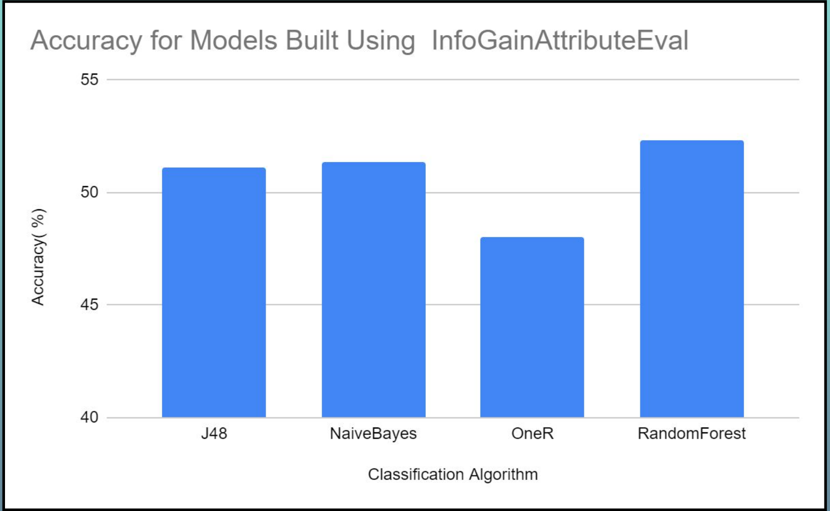
# Best Model

J48 with attributes selected by WrapperSubsetEval

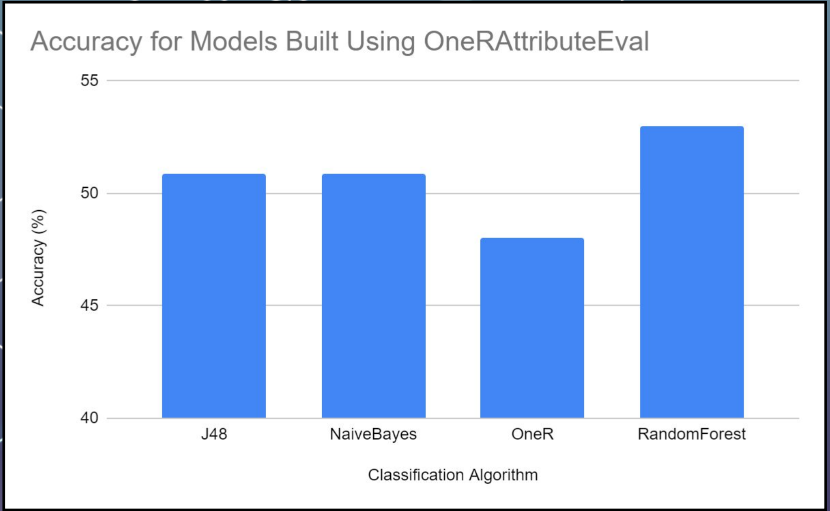
Accuracy: **54.5**
Recall: **0.401**



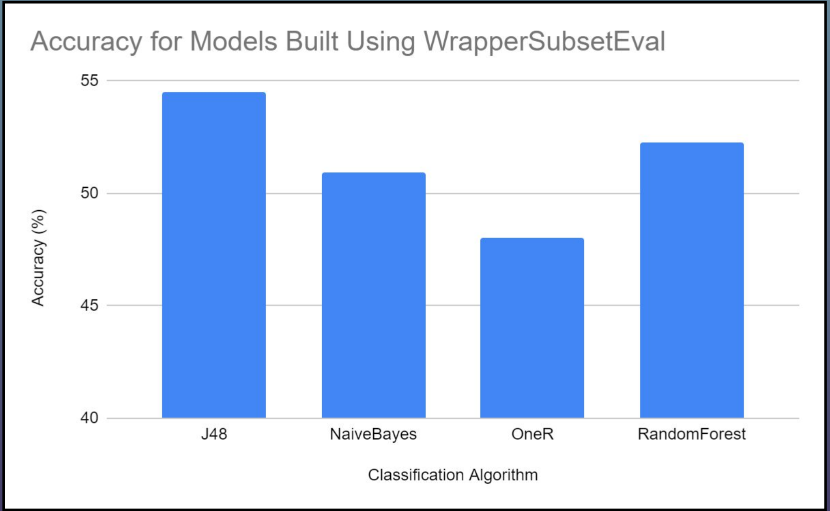
Accuracy for Models Built Using CorrelationAttributeEval Attributes



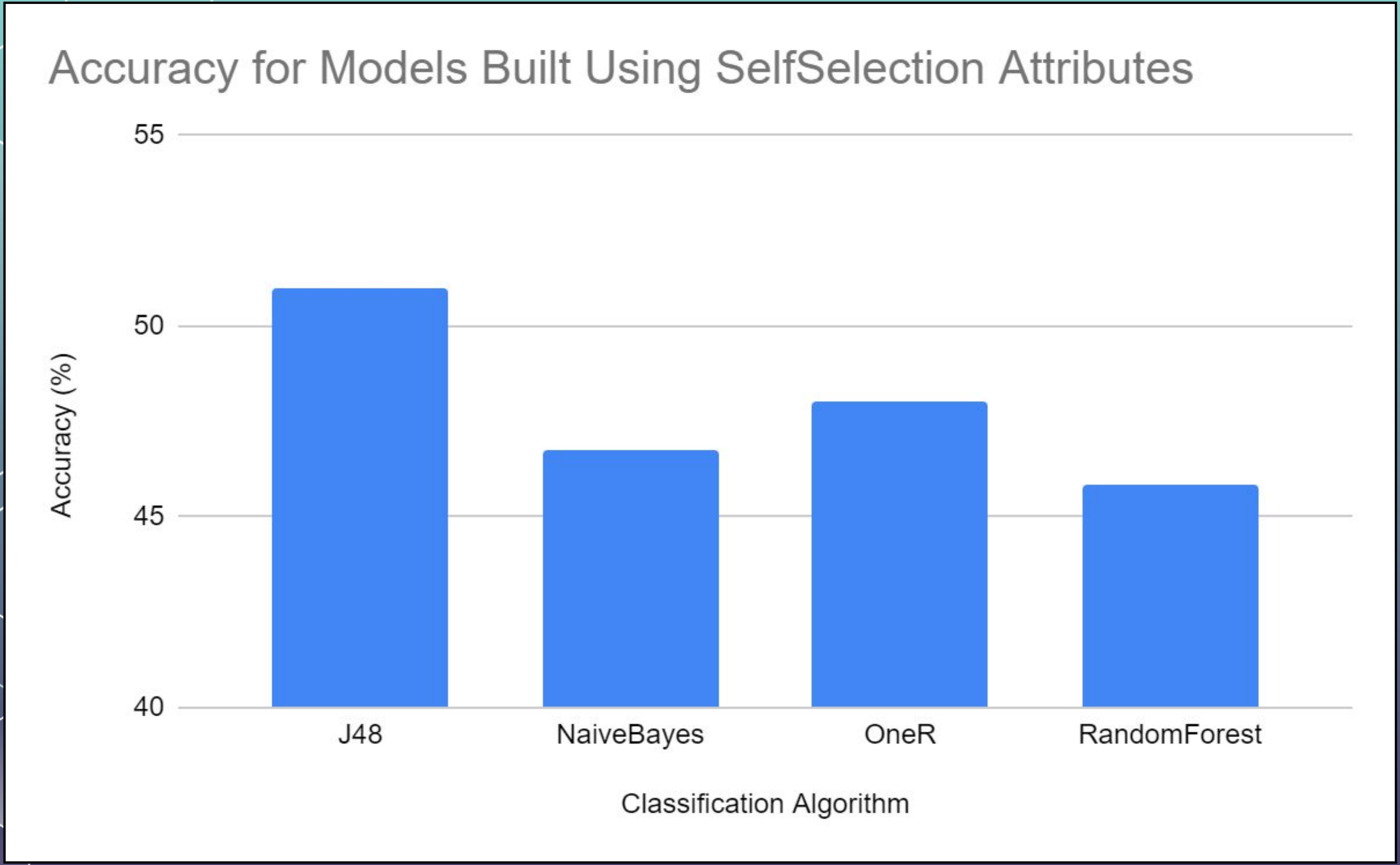
Accuracy for Models Built Using  InfoGainAttributeEval



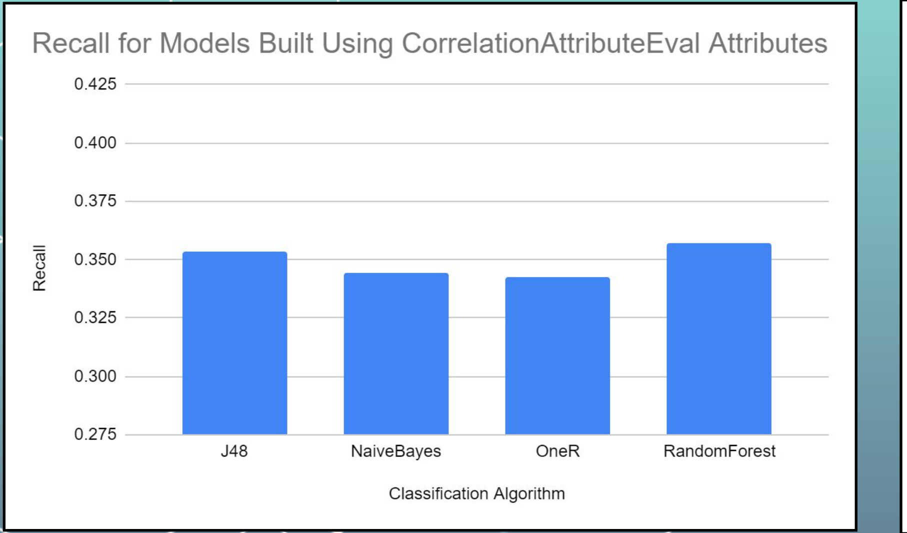
Accuracy for Models Built Using OneRAttributeEval



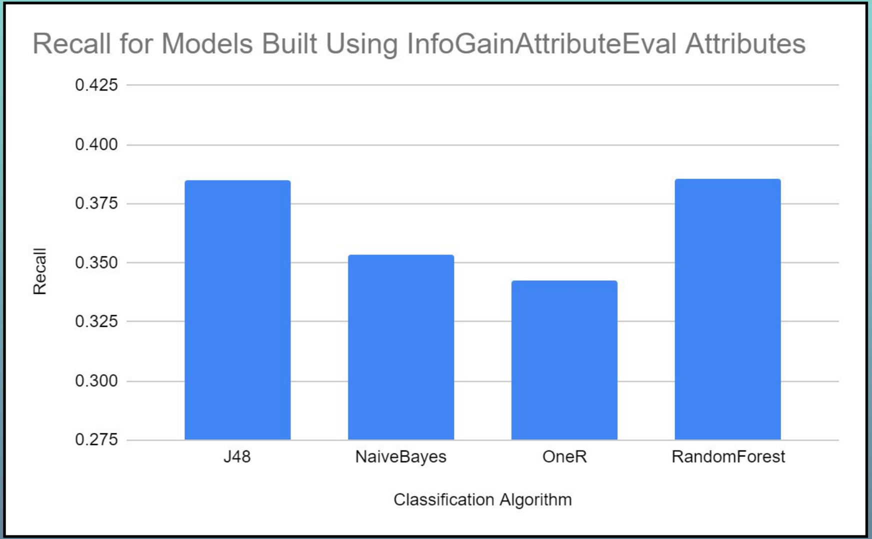
Accuracy for Models Built Using WrapperSubsetEval



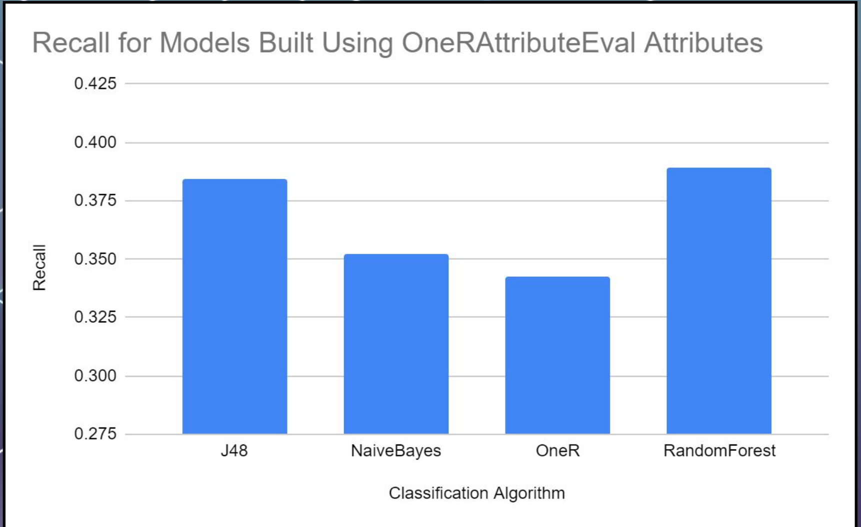
Accuracy for Models Built Using SelfSelection Attributes



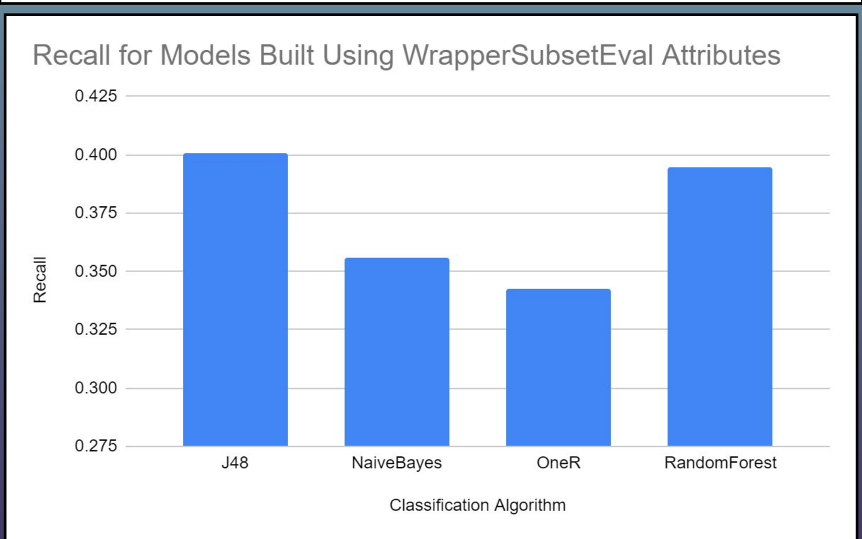
Recall for Models Built Using CorrelationAttributeEval Attributes



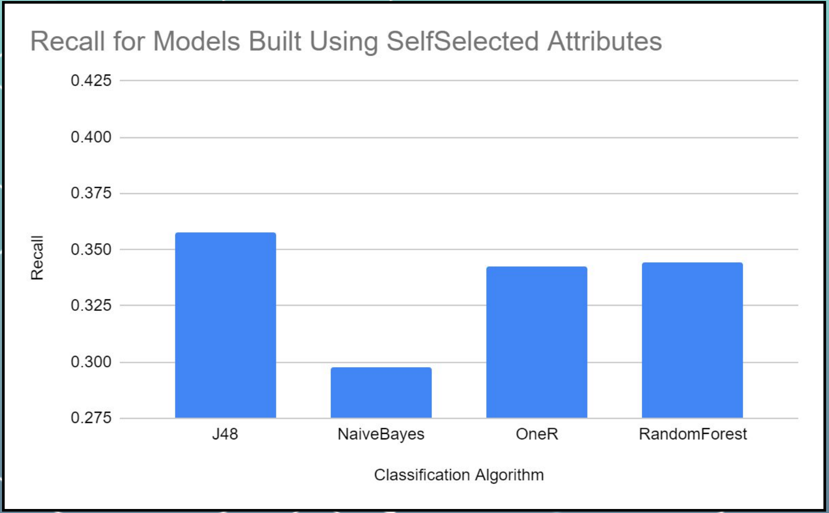
Recall for Models Built Using InfoGainAttributeEval Attributes

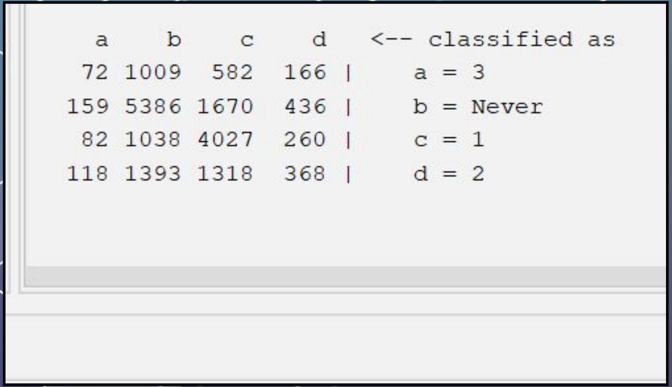


Recall for Models Built Using OneRAttributeEval Attributes



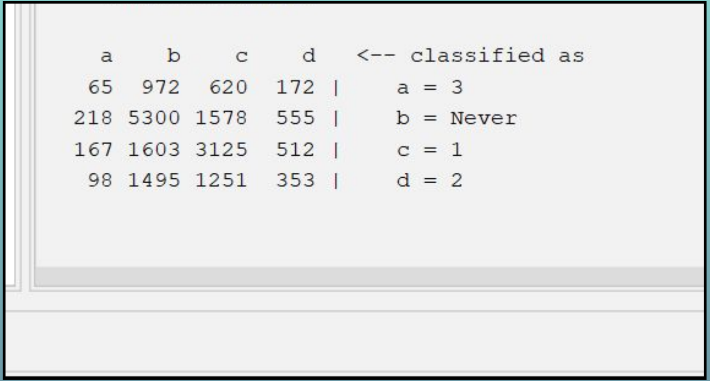
Recall for Models Built Using WrapperSubsetEval Attributes





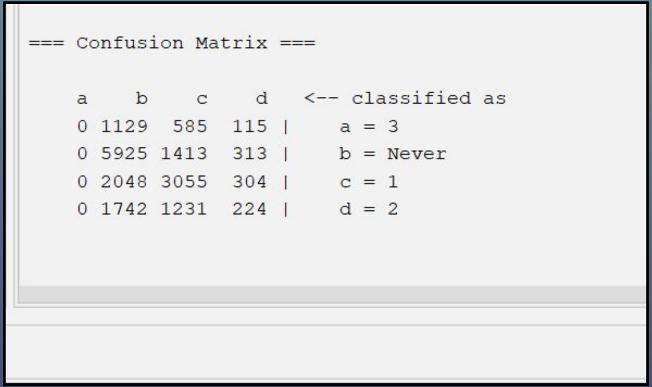
```
   a    b    c    d   <-- classified as
  72 1009  582  166 |    a = 3
 159 5386 1670  436 |    b = Never
  82 1038 4027  260 |    c = 1
 118 1393 1318  368 |    d = 2
```

J48 WrapperSubsetEval



```
   a    b    c    d   <-- classified as
  65  972  620  172 |    a = 3
 218 5300 1578  555 |    b = Never
 167 1603 3125  512 |    c = 1
  98 1495 1251  353 |    d = 2
```

RandomForest CorrelationAttributeEval



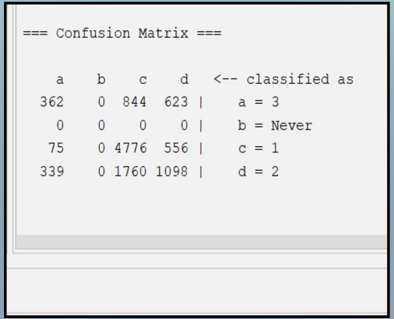
```
=== Confusion Matrix ===

    a    b    c    d   <-- classified as
    0 1129  585  115 |    a = 3
    0 5925 1413  313 |    b = Never
    0 2048 3055  304 |    c = 1
    0 1742 1231  224 |    d = 2
```

NaiveBayes OneRAttributeEval

# 05

# Discussion and Sources of Error

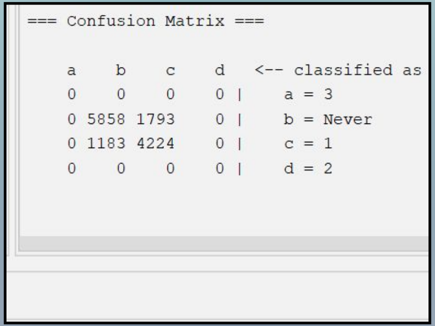
# Increasing Accuracy



```
=== Confusion Matrix ===

    a    b    c    d   <-- classified as
  362    0  844  623 |    a = 3
    0    0    0    0 |    b = Never
   75    0 4776  556 |    c = 1
  339    0 1760 1098 |    d = 2
```

### Removing “Never”

Since “Never” was the most predicted, we decided to remove it to see if it improved accuracy. We got a 59.8% accuracy.



```
=== Confusion Matrix ===

    a    b    c    d   <-- classified as
    0    0    0    0 |    a = 3
    0 5858 1793    0 |    b = Never
    0 1183 4224    0 |    c = 1
    0    0    0    0 |    d = 2
```

### Removing “2” and “3”

These were the least predicted, so we decided to remove them, giving us a 77.2% accuracy.

# Error and Discussion

### Error Sources

❏ Similar characteristics regardless of recidivism year

❏ Not enough data for small categories

### Discussion

❏ Potential to be used in real life, but more analysis is required to build a suitable model

# THANKS

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, infographics & images by Freepik

**Questions?**