

**SCTR's Pune Institute of Computer Technology  
Dhankawadi, Pune**

**AN INTERNSHIP REPORT ON**

**ActOne ML for Pattern Recognition**

**SUBMITTED BY**

**Name: Shubham Rajendra Chemate**

**Class: TE-1**

**Roll no: 31118**

**Under the guidance of**

**Prof. M. S. Wakode**



**DEPARTMENT OF COMPUTER ENGINEERING  
ACADEMIC YEAR 2021-22**



## DEPARTMENT OF COMPUTER ENGINEERING

SCTR's Pune Institute of Computer Technology  
Dhankawadi, Pune  
Maharashtra 411043

### CERTIFICATE

This is to certify that the SPPU Curriculum-based internship report entitled  
**“ActOne ML for Pattern Recognition”**

Submitted by  
*Shubham Rajendra Chemate*  
31118

has satisfactorily completed the curriculum-based internship under the guidance of *Prof. M. S. Wakode* towards the partial fulfillment of third year Computer Engineering Semester VI,  
Academic Year 2021-22 of Savitribai Phule Pune University.

*Prof. M. S. Wakode*  
Internship Guide  
PICT, Pune

Dr. G. V. Kale  
Head  
Department of Computer Engineering  
PICT, Pune

Place:Pune  
Date: 28/04/2022

## Acknowledgement

It gives me great pleasure in presenting the internship report on "ActOne ML for Pattern Recognition".

First of all I would like to take this opportunity to thank my internship guide Prof. M. S. Wakode for giving me all the help and guidance needed. I am really grateful for her kind support and valuable suggestions that proved to be beneficial in the overall completion of this internship.

I am thankful to our Head of Computer Engineering Department, Dr. G. V. Kale, for her indispensable support and suggestions throughout the internship work.

I would also genuinely like to express my gratitude to the Department Internship Coordinator, Prof. P. P. Joshi, for her constant guidance and support and for the timely resolution of the doubts related to the internship process.

Finally, I would like to thank my mentor, Prof. T. S. Pinjan for his constant help and support during the overall internship process.

## Contents

|          |                                      |           |
|----------|--------------------------------------|-----------|
| <b>1</b> | <b>Title</b>                         | <b>3</b>  |
| <b>2</b> | <b>Introduction</b>                  | <b>3</b>  |
| <b>3</b> | <b>Problem Statement</b>             | <b>4</b>  |
| <b>4</b> | <b>Objectives and Scope</b>          | <b>4</b>  |
| <b>5</b> | <b>Methodological Details</b>        | <b>5</b>  |
| 5.1      | Models . . . . .                     | 5         |
| 5.1.1    | Linear Regression Model . . . . .    | 5         |
| 5.1.2    | Naive Bayes Classifier . . . . .     | 6         |
| 5.2      | Architecture Diagram . . . . .       | 6         |
| <b>6</b> | <b>Modern Engineering Tools Used</b> | <b>7</b>  |
| <b>7</b> | <b>Results of Internship Work</b>    | <b>8</b>  |
| <b>8</b> | <b>Achievement</b>                   | <b>11</b> |

## List of Figures

|   |   |    |
|---|---|----|
| 1 | Dataset for Linear Regression . . . . . | 6  |
| 2 | Architecture Diagram . . . . .          | 6  |
| 3 | Data Preprocessing . . . . .            | 8  |
| 4 | Relation between Alerts . . . . .       | 8  |
| 5 | Data Visualization . . . . .            | 9  |
| 6 | Linear Regression Model . . . . .       | 9  |
| 7 | Naive Bayes Classifier . . . . .        | 10 |
| 8 | Predictor . . . . .                     | 10 |

# 1 Title

ActOne ML for Pattern Recognition

## 2 Introduction

In today's rapidly growing technological era, the rate of financial crimes has significantly increased over the past few years. The new types of crimes are becoming more smarter than those which were present already, and because of this crime investigation is also becoming very complex.

Financial crimes analysts spend lots of time on manual processing and repetitive work. According to the data, 56% of organizations spend 30% time on manual processing which is actually considerably high. This issue must be addressed with the help of modern tools and technology.

To address this issue we are developing a solution (Machine Learning based) which will predict the next best steps of investigation for a transaction fraud. The model is based on Linear Regression and Naive Bayes Classifier with a accuracy of around 93%.

### 3 Problem Statement

We are provided a dataset which contains number of alert so using an efficient algorithm of ML we have to predict some pattern and when further such alerts trigger we have to give our predicted results i.e next best step, the aim is to reduce the manual investigation time.

### 4 Objectives and Scope

The motive behind this project is to reduce the manual process of analysing fraud and determining investigation steps and automate it using the technology to reduce the time and increase the efficiency. The main challenge is how to design the model which will take details about financial crime and predict investigation steps by analysing it according to the requirements. We can use the recent technologies like machine learning and design the predictor that takes the input as a transaction fraud details(which is in form of alerts) and then it will get processed and analysed using ML algorithm and give result. With this output we can reduce the steps of investigation.

As we are building a model which will predict the workflow of new alert generated depending on the previous existing data we can reduce the time of investigation up to 70% and also increase the efficiency of investigation process. Also we are developing a generalized plugin or extension so that any organization can used it for detecting any fraud and to predict the next best step of investigation.

## 5 Methodological Details

This project consists of Natural Language Processing and Machine Learning. Natural language processing refers to the branch of computer science and more specifically, the branch of artificial intelligence or AI—concerned with giving computers the ability to understand text and spoken words in much the same way human beings can.

Different tasks performed in the project include:

- Collect and clean datasets for business & non-business classification, sentiment analysis and entity recognition.
- Train the custom classification and entity recognition models using Python.
- Apply Linear Regression and Naive Bayes Classifier on the preprocess the data.
- Create plugin for predicting next best step of investigation from previously collected data.
  - Train new models for custom document classification and custom entity recognition.
  - Check the status of their models.
  - Extract trade insights from the text data submitted.

### 5.1 Models

- a) Linear Regression Model
- b) Naive Bayes Classifier

#### 5.1.1 Linear Regression Model

Linear regression is a basic and commonly used type of predictive analysis. The overall idea of regression is to examine two things:

1. Does a set of predictor variables do a good job in predicting an outcome (dependent) variable?
2. Which variables in particular are significant predictors of the outcome variable, and in what way do they—indicated by the magnitude and sign of the beta estimates—impact the outcome variable?

These regression estimates are used to explain the relationship between one dependent variable and one or more independent variables. The simplest form of the regression equation with one dependent and one independent variable is defined by the formula

$$y = c + b * x$$

where

- y = estimated dependent variable score,
- c = constant,
- b = regression coefficient,



$x$  = score on the independent variable.

| CUSTOM_MEDIUM_STRING_13 | CUSTOM_MEDIUM_STRING_14 | RULE_FP                               | EXECUTED_PROCESS_ID | ISSUE_TIMESTAMP | ISSUE_KEY                             | ALERT_ID | ALERT_DATE |
|-------------------------|-------------------------|---------------------------------------|---------------------|-----------------|---------------------------------------|----------|------------|
| NULL                    |                         | 3 ca6115dcf471ac5b4efd4a8ad6c1311e    | NULL                | 54:43.2         | 0dd88923-d922-451c-b54c-6810fe194a0e  | SAM1-1   | NULL       |
| NULL                    | NULL                    | 6ff0c7d52cf06eca7bfe6e21191d064       | NULL                | 59:18.0         | 2bdc6ab0-da89-4e25-9fcb-2f34daecb8fe  | SAM1-1   | NULL       |
| NULL                    |                         | 906 e28e7892e9dac725dd107f1c3804ff07  | NULL                | 54:43.3         | 2fb2a504-c371-4f70-ad4e-a79c3eeab3dc  | SAM1-1   | NULL       |
| NULL                    |                         | 1681 5aec577cd4f26eea74c469f2d7bc81e4 | NULL                | 54:45.8         | 3a749837-79cb-4e2f-a16a-5796943136eb  | SAM1-1   | NULL       |
| NULL                    | NULL                    | fe549f31c6997a444fba9cc4e947dfb       | NULL                | 59:18.0         | 4b8576a9-556c-4c3e-8f9b-22f7c1fdbc6a2 | SAM1-1   | NULL       |
| NULL                    |                         | 908 37509c4ab152338bad5ad79310ad3e4e  | NULL                | 54:46.1         | 531dfcccd-bf20-48a8-9102-32214d8ba1eb | SAM1-1   | NULL       |
| NULL                    |                         | 1357 c42e062ab1f83fc4a893e1e175a88759 | NULL                | 54:45.6         | 59ceea3f-3fca-462c-8710-a3230ad2bc1f  | SAM1-1   | NULL       |
| NULL                    |                         | 2189 3ac399c678f7fc9f820ffbb36d027df4 | NULL                | 54:44.9         | 5c953e09-9345-4563-9cef-9e336ed08a35  | SAM1-1   | NULL       |
| NULL                    |                         | 1316 5e9ebe12451242454065684337289a22 | NULL                | 54:43.9         | 6aa6c7e9-e575-4fd4-b0ef-815ded66edee  | SAM1-1   | NULL       |
| NULL                    |                         | 4 c4e5d29ef6036ad6e6bf8f460e36cd      | NULL                | 54:44.1         | 6c9cd8d6-a19e-4e0d-bdd6-519aa43e2f0b  | SAM1-1   | NULL       |
| NULL                    |                         | 1667 5aec577cd4f26eea74c469f2d7bc81e4 | NULL                | 54:46.1         | 711577e2-3da9-4124-adab-ce70e1889abd  | SAM1-1   | NULL       |
| NULL                    | NULL                    | 82fb4266c412ecf8bd6db1b59c069ec8      | NULL                | 59:18.0         | 78418d8e-71e5-43f3-9ad7-d83eb562c5bd  | SAM1-1   | NULL       |
| NULL                    |                         | 1682 5aec577cd4f26eea74c469f2d7bc81e4 | NULL                | 54:45.8         | 8179af2f-e5cb-41ee-8b92-c44deb447d71  | SAM1-1   | NULL       |
| NULL                    |                         | 2191 09bc94cd8ede84ae2f6dcf46e6445dab | NULL                | 54:45.3         | a96a0c20-031f-4227-adc6-043c1c22ff44  | SAM1-1   | NULL       |
| NULL                    |                         | 2190 b73a1cd95b7b1c670125219173f895bb | NULL                | 54:45.1         | a9ab47e5-dfb7-496d-8703-f0486bf2dbdd  | SAM1-1   | NULL       |
| NULL                    |                         | 1668 c4450399b0578727c5c86469277934eb | NULL                | 54:45.6         | b02349ce-090e-4526-9271-85f0936a925f  | SAM1-1   | NULL       |
| NULL                    |                         | 1963 09bc94cd8ede84ae2f6dcf46e6445dab | NULL                | 54:45.3         | ca27eb79-8b7e-4ffa-944d-8b191574f311  | SAM1-1   | NULL       |
| NULL                    |                         | 1680 5aec577cd4f26eea74c469f2d7bc81e4 | NULL                | 54:45.7         | d5d0fb32-9399-437f-9e17-dae1e6a0f0c4  | SAM1-1   | NULL       |
| NULL                    |                         | 1683 5aec577cd4f26eea74c469f2d7bc81e4 | NULL                | 54:46.0         | ed4bb583-85de-89fe-842f-02190c3381b4  | SAM1-1   | NULL       |
| NULL                    |                         | 5 363875940d37431986974e32df300abd    | NULL                | 54:42.4         | f9902c0a-6f5f-49f7-bda3-64a0185528d1  | SAM1-1   | NULL       |
| NULL                    |                         | 1964 b73a1cd95b7b1c670125219173f895bb | NULL                | 54:45.2         | fb81fa56-86b5-425a-aded-7110e001437c  | SAM1-1   | NULL       |
| NULL                    |                         | 1666 5aec577cd4f26eea74c469f2d7bc81e4 | NULL                | 55:07.5         | 00332a86-9ab9-4d69-82d1-64f2fadc2541  | SAM1-10  | NULL       |
| NULL                    |                         | 1374 5aec577cd4f26eea74c469f2d7bc81e4 | NULL                | 55:07.4         | 02d759d3-e8f1-49a7-ac53-a17d1fcf59a4  | SAM1-10  | NULL       |

Figure 1: Dataset for Linear Regression

### 5.1.2 Naive Bayes Classifier

Naive Bayes is a classification algorithm for binary (two-class) and multiclass classification problems. It is called Naive Bayes or idiot Bayes because the calculations of the probabilities for each class are simplified to make their calculations tractable”.

## 5.2 Architecture Diagram

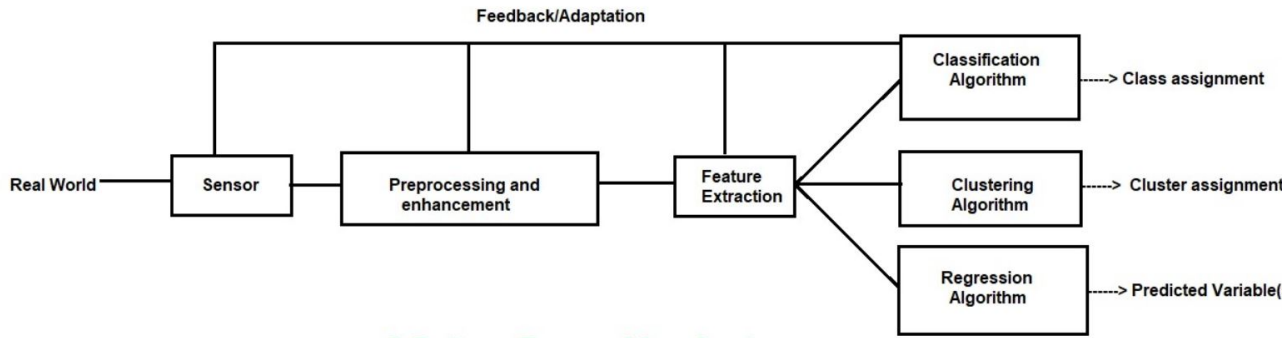


Figure 2: Architecture Diagram

## 6 Modern Engineering Tools Used

Technologies which we have used in this project are Amazon Comprehend & Amazon Simple Storage Service (Amazon S3) for Natural Language Processing, Machine Learning to train and test the model and AI algorithm for prediction using python language.

1. Amazon Comprehend:

It is a natural-language processing (NLP) service offered by Amazon Web Services that uses machine learning to uncover valuable insights and connections in text.

2. Amazon S3:

Amazon Simple Storage Service is also a service offered by Amazon Web Services that provides object storage through a web service interface.

3. Jupyter Notebook:

The Jupyter Notebook is an open-source web application that allows data scientists to create and share documents that integrate live code, equations, computational output, visualizations, and other multimedia resources, along with explanatory text in a single document

4. Python:

Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a general-purpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems.

5. Excel:

Analyze Data in Excel empowers you to understand your data through natural language queries that allow you to ask questions about your data without having to write complicated formulas. In addition, Analyze Data provides high-level visual summaries, trends, and patterns.

## 7 Results of Internship Work

(Screenshots of work done)

```

In [1]: #Importing Required Libraries
import pandas as pd
import numpy as np
from sklearn import linear_model
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
from word2number import w2n
import scipy.stats as stats
import pylab

In [27]: df=pd.read_csv('Sample1.csv')

In [3]: l=df.shape
rows=l[0]
col=l[1]

In [4]: df.dtypes
Out[4]:
RULE_ID      object
Avg Value    int64
Threshold    int64
ALERT_ID     object
Score        int64
dtype: object

In [5]:
df1=df[['Threshold']]
df2=df[['Avg Value']]
d=0
j=60
k=30
l=[]

```

Figure 3: Data Preprocessing

```

In [8]: df['Score'].mean()
Out[8]: 67.3076923076923

In [9]: df['RULE_ID'][0]
Out[9]: 'AML-EBB-IFT-ALL-A-D05-EOP'

In [10]: df1=df[df['ALERT_ID']=='SAM1-1']['RULE_ID']
df2=df[df['ALERT_ID']=='SAM1-2']['RULE_ID']

In [11]: df1
Out[11]:
0    AML-EBB-IFT-ALL-A-D05-EOP
1    AML-EFC-NEN-ACT-A-001-CVA
2    AML-EFC-NEN-ACT-A-001-CVA
3    AML-TSD-EFT-ALL-A-S01-FSI
4    AML-EFC-RSK-LVL-A-039-CVA
5    AML-ULN-LNP-INI-A-LNB-DEL
6    AML-SUS-SUR-FRE-A-M12-FRA
7    AML-SRD-SIN-ALL-A-S01-CTY
8    AML-OLN-LNP-INI-A-S01-DET
9    AML-SIM-OSI-000-A-D30-MLA
10   AML-TSD-EFT-ALL-A-S01-FSI
11   AML-EFC-RSK-CTY-A-008-CLA
12   AML-TSD-EFT-ALL-A-S01-FSI
Name: RULE_ID, dtype: object

In [12]: df2
Out[12]:
13   AML-EBB-IFT-ALL-A-D05-EOP
14   AML-EFC-NEN-ACT-A-001-CVA
15   AML-EFC-NEN-ACT-A-001-CVA
16   AML-TSD-EFT-ALL-A-S01-FSI
17   AML-EFC-RSK-LVL-A-039-CVA

```

Figure 4: Relation between Alerts

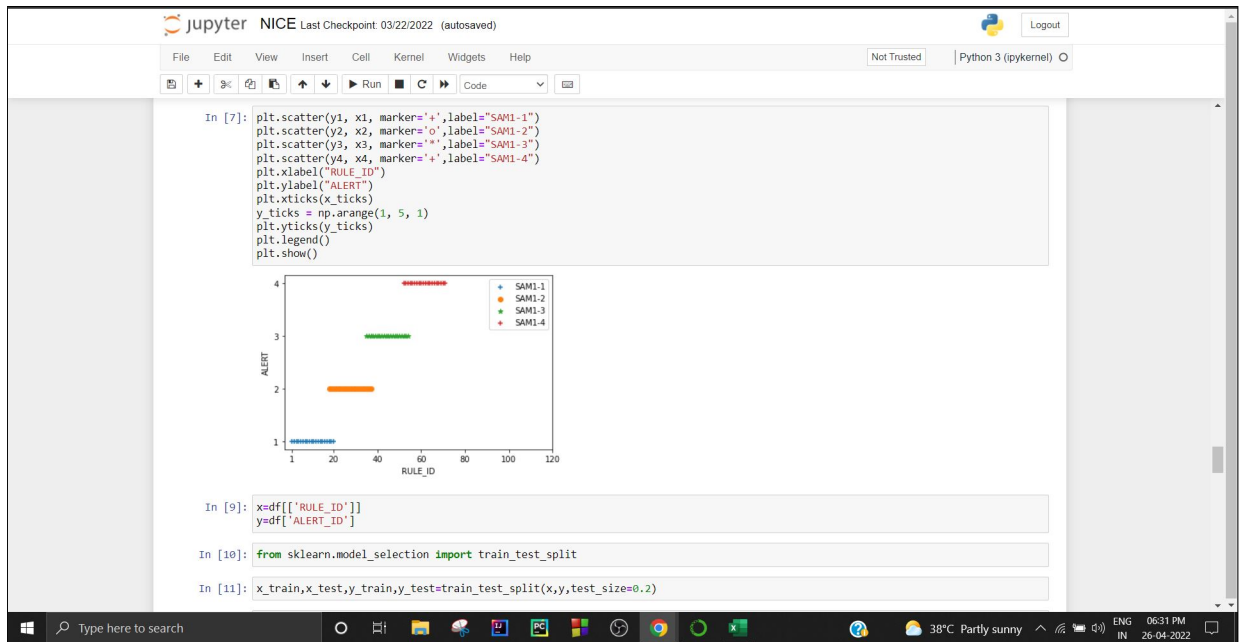


Figure 5: Data Visualization

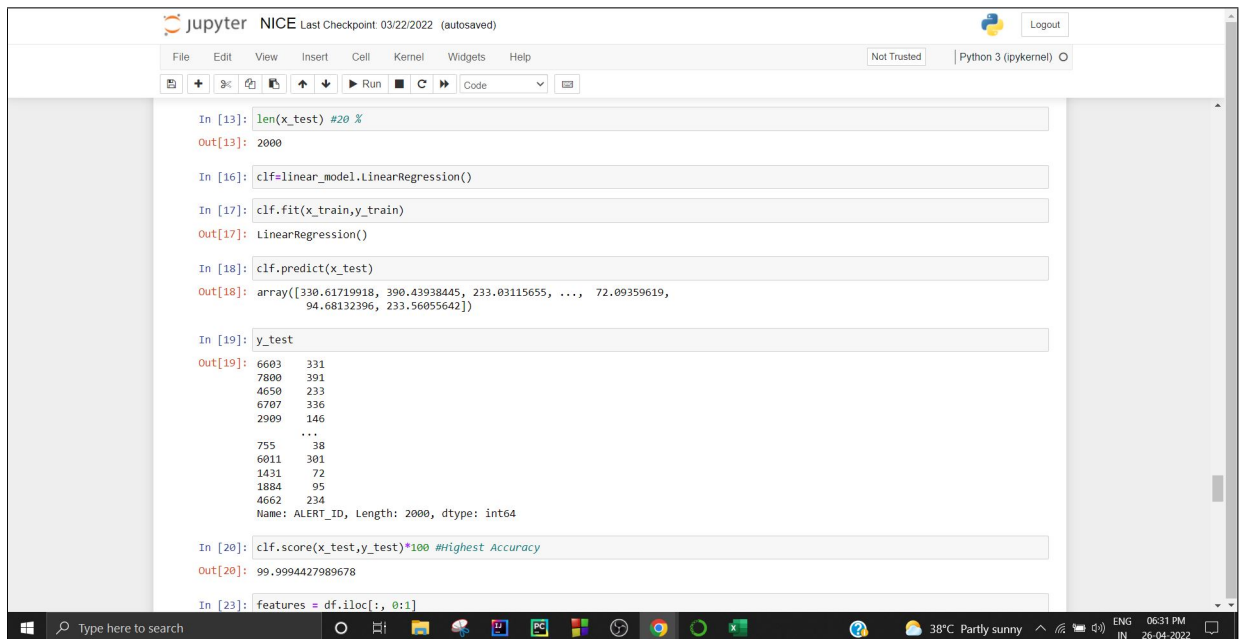


Figure 6: Linear Regression Model

The image shows a Jupyter Notebook window titled "NICE" with a last checkpoint of "03/22/2022 (autosaved)". The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and code execution. The notebook is running on a Python 3 (ipykernel) environment.

The code in the notebook is as follows:

```

3 SAM-1
4 SAM-1
Name: ALERT, dtype: object

In [27]: print('The Initial DataFrame Contained %d Rows And %d Columns'%(df.shape))
         print('The Features Matrix Contains %d Rows And %d Columns'%(features.shape))
         print('The Target Vector Contains %d Rows And %d Columns'%(np.array(target).reshape(-1, 1).shape))

The Initial DataFrame Contained 10000 Rows And 6 Columns
The Features Matrix Contains 10000 Rows And 1 Columns
The Target Vector Contains 10000 Rows And 1 Columns

In [28]: from sklearn.naive_bayes import GaussianNB
         algorithm = GaussianNB(priors=None, var_smoothing=1e-9)

In [29]: algorithm.fit(features, target)
Out[29]: GaussianNB()

In [30]: print(algorithm.classes_)

['SAM-1' 'SAM-10' 'SAM-100' 'SAM-101' 'SAM-102' 'SAM-103' 'SAM-104'
 'SAM-105' 'SAM-106' 'SAM-107' 'SAM-108' 'SAM-109' 'SAM-11' 'SAM-110'
 'SAM-111' 'SAM-112' 'SAM-113' 'SAM-114' 'SAM-115' 'SAM-116' 'SAM-117'
 'SAM-118' 'SAM-119' 'SAM-12' 'SAM-120' 'SAM-121' 'SAM-122' 'SAM-123'
 'SAM-124' 'SAM-125' 'SAM-126' 'SAM-127' 'SAM-128' 'SAM-129' 'SAM-13'
 'SAM-130' 'SAM-131' 'SAM-132' 'SAM-133' 'SAM-134' 'SAM-135' 'SAM-136'
 'SAM-137' 'SAM-138' 'SAM-139' 'SAM-14' 'SAM-140' 'SAM-141' 'SAM-142'
 'SAM-143' 'SAM-144' 'SAM-145' 'SAM-146' 'SAM-147' 'SAM-148' 'SAM-149'
 'SAM-15' 'SAM-150' 'SAM-151' 'SAM-152' 'SAM-153' 'SAM-154' 'SAM-155'
 'SAM-156' 'SAM-157' 'SAM-158' 'SAM-159' 'SAM-16' 'SAM-160' 'SAM-161'
 'SAM-162' 'SAM-163' 'SAM-164' 'SAM-165' 'SAM-166' 'SAM-167' 'SAM-168'
 'SAM-169' 'SAM-17' 'SAM-170' 'SAM-171' 'SAM-172' 'SAM-173' 'SAM-174'
 'SAM-175' 'SAM-176' 'SAM-177' 'SAM-178' 'SAM-179' 'SAM-18' 'SAM-180'
 'SAM-181' 'SAM-182' 'SAM-183' 'SAM-184' 'SAM-185' 'SAM-186' 'SAM-187'

```

Figure 7: Naive Bayes Classifier

The image shows a Jupyter Notebook window with the following code and output:

```

In [31]: print('The Gaussian Model Has Achieved %.2f Percent Accuracy'%(algorithm.score(features, target)))
The Gaussian Model Has Achieved 0.85 Percent Accuracy

In [32]: observation = [[84]]

In [35]: predictions=algorithm.predict(observation)

In [36]: predictions
Out[36]: array(['SAM-5'], dtype='<U7')

In [ ]:

```

Figure 8: Predictor

## 8 Achievement

- Guided Investigation: Consistent approach to investigation using guided ML steps
- Cost Saving: Time-spent on similar event is reduced
- Integrated the Model with Actimize Case Manager so that it can be used on larger scale