

CLASSIFICATION OF MISSING YOUTHS CASES USING
SUPPORT VECTOR MACHINES

A Thesis Submitted to the
College of Graduate and Postdoctoral Studies
in Partial Fulfillment of the Requirements
for the degree of Master of Science
in the Department of Computer Science
University of Saskatchewan
Saskatoon

By
Maryam Orafaee Azghan

©Maryam Orafaee Azghan, June/2019. All rights reserved.

PERMISSION TO USE

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan S7N 5C9
Canada

Or

Dean
College of Graduate and Postdoctoral Studies
University of Saskatchewan
116 Thorvaldson Building, 110 Science Place
Saskatoon, Saskatchewan S7N 5C9
Canada

ABSTRACT

A missing person is defined as a person who has been formally reported to the police as someone whose whereabouts are unknown. Many people go missing every year, and the vast majority of them are found or return home within one week. The purpose of this study is to provide the Saskatoon Police Service (SPS) with a set of predictive models for intervention and risk reduction applied to the missing youths (MY) database in a graphical user interface (GUI). The study is conducted on 434 missing persons cases with 91 features for each case. Two classification features are used to build the predictive models presented in this study. The first feature is *missing_again*. This feature allows us to train a predictive model to assess the likelihood of a person to go missing again. The second feature is *gang_involvement*. This feature allows us to train a predictive model to assess the likelihood of a missing person will be involved in gang activity. The machine learning method Support Vector Machines (SVMs) is used to build models based on both classification variables and 91 items as input features for each MY case.

The SVM classifier obtained an accuracy of 89%, sensitivity of 84%, and specificity of 91% for the classification feature *missing_again*. This classifier can be used by officers to consider when deciding whether or how to intervene in cases where youths are at a high risk to go missing again. The SVM classifier obtained an accuracy of 84%, sensitivity of 43%, and specificity of 90% for the classification feature *gang_involvement*. This classifier can be used by officers to determine whether a missing youth is likely to have gang affiliations. This knowledge could change how the case is approached and the safety measures the officers may take. The Ministry of Justice of Saskatchewan and the SPS also desire a graphical user interface for data analysis and reporting amenable for use by end users of the Missing Persons Project (MPP), e.g., police officers and data analysts.

ACKNOWLEDGEMENTS

I wish to thank my supervisor Prof. Raymond J. Spiteri for his fantastic guidance, support, and passion. The door to his office was always open whenever I ran into a trouble spot or had a question about my research or writing. I would like to give my thank you to my parents, and my husband, Navid, for providing me with continuous support and encouragement throughout my years of study, the process of researching and writing this thesis. This accomplishment would not have been possible without them. I would also like to thank the Saskatoon Police Services that provided me and the team I work with ethical access to the data worked with in this thesis.

CONTENTS

Permission to Use	i
Abstract	ii
Acknowledgements	iii
Contents	iv
List of Tables	vi
List of Figures	vii
List of Abbreviations	viii
1 Introduction	1
1.1 Machine Learning	1
1.1.1 Support Vector Machines	2
1.2 Saskatoon Police Service Missing Youths Database	2
1.3 Missing Youths Graphical User Interface	3
1.4 Ethics	3
2 Literature Review	4
2.1 Machine Learning	4
2.1.1 Support Vector Machines	5
2.2 Missing Youths Data	6
2.3 Graphical User Interface	7
3 Machine Learning: Methods and Implementation	8
3.1 Support Vector Machines	8
3.1.1 Linear SVM Classification Algorithm	15
3.2 Model Performance Evaluation	16
3.2.1 Confusion Matrix	17
3.2.2 Accuracy	17
3.2.3 Sensitivity and Specificity	17
3.2.4 Receiver Operating Characteristic Curve	18
3.3 Model Validation	18
3.3.1 Simple Split	18
3.3.2 Multi-fold Cross-validation	18
3.4 Description of the MY Database	19
3.4.1 Classification Features	20
3.5 Graphical User Interface Development	21
3.5.1 Missing Youths Graphical User Interface Development	21
3.5.2 Logical Diagram	22
3.5.3 Physical Diagram	22
4 Results	24
4.1 Basic Statistical Analysis of the Missing Youths Database	24
4.2 Missing Youths Experiment Results	28
4.2.1 Results for the missing_again classification feature	29
4.2.1.1 Confusion Matrix for missing_again Classification Feature	29

4.2.1.2	ROC Curve for the missing_again Classification Feature	30
4.2.2	Results for the gang_involvement Classification Feature	31
4.2.2.1	Confusion Matrix for the gang_involvement Classification Feature	32
4.2.2.2	ROC Curve for the gang_involvement Classification Feature	33
4.2.3	Discussion	34
5	Conclusion and Future Work	36
5.1	Conclusion	36
5.2	Future Work	36
	Bibliography	38
	Appendix A Missing Youths Graphical User Interface	44
A.1	“Login” Page	44
A.1.1	“Invalid Username/Password” Page	45
A.2	“Home” Page	45
A.3	“Missing Persons” Page	46
A.3.1	“Date Range” Page	46
A.3.2	“List” Page	47
A.3.3	“Heat Map” Page	48
A.3.4	“Reports” Page	49
A.4	“Settings” Page	52
A.4.1	“Modules” Page	52
A.5	“Users” Page	53
A.5.0.1	“List of Users” Page	53
A.5.1	“Add User” Page	54
A.6	“Email” Page	55
A.6.1	“Inbox” Page	55
A.6.2	“Compose” Page	56
A.7	“Data sets” Page	57
A.7.1	“List of data sets” Page	57
A.7.2	“Add data set” Page	58
A.8	Data Analysis	59
A.8.1	Decision Trees	59
A.8.2	Selective Decision Trees	60
A.8.3	Support Vector Machines	61

LIST OF TABLES

3.1	Comparison of different kernel functions for the MY database	15
3.2	A confusion matrix	17
4.1	Basic information about the MY cases	24
4.2	Basic statistics for age (years)	24
4.3	Basic statistics for age grouped by gender (years)	25
4.4	Basic statistics for time missing (days)	25
4.5	Basic statistics for time missing grouped by gender (days)	26
4.6	Basic statistics for repeat occurrences (number of times)	26
4.7	Basic statistics for repeat occurrences grouped by gender (number of times)	27
4.8	Basic statistics for time to the next event (days)	28
4.9	Basic statistics for time to the next event grouped by gender (days)	28
4.10	SVMs results for the <i>missing_again</i> classification feature (%) with 10-fold cross-validation . .	29
4.11	SVM results for the <i>gang_involvement</i> classification feature (%) with 10-fold Cross-validation . .	32
4.12	SVM results in percent (%)	34

LIST OF FIGURES

3.1	How SVMs linearly classify data in a two-dimensional space	10
3.2	Misclassified data using the ξ_i variables	12
3.3	Visualization of mapping data in a higher-dimensional space using a kernel method	12
3.4	Visualization of SVM linear kernel	13
3.5	Visualization of SVM polynomial kernel	14
3.6	Visualization of SVM radial basis function kernel	14
3.7	Visualization of SVM sigmoid kernel	15
3.8	Logical diagram for the MY GUI	22
3.9	Physical diagram of the MY GUI	23
4.1	Histogram of age	25
4.2	Histogram of time missing	26
4.3	Histogram of repeat occurrences	27
4.4	Histogram of time to next event	28
4.5	Confusion matrix for the <i>missing_again</i> classification feature	30
4.6	Receiver Operating Characteristic curve for the <i>missing_again</i> classification feature	31
4.7	Confusion matrix for the <i>gang_involvement</i> classification feature	33
4.8	Receiver Operating Characteristic curve for the <i>gang_involvement</i> classification feature	33
A.1	Screenshot for “Login” page	44
A.2	Screenshot for “Forgot Password?” page	44
A.3	Screenshot for “Invalid Username/Password” page	45
A.4	Screenshot for “Home” page	46
A.5	Screenshot for specifying a date range using a calendar interface	46
A.6	Screenshot for “Date Range” page	47
A.7	Screenshot for “List” page	47
A.8	Screenshot for “Heat Map” page	48
A.9	Screenshot for Table Report	49
A.10	Screenshot for Chart Reports	50
A.11	Screenshot for PDF Report	51
A.12	Screenshot for “Modules” page	52
A.13	Screenshot for “List of Users” page	53
A.14	Screenshot for “Add User” page	54
A.15	Screenshot for “Inbox” page	55
A.16	Screenshot for “Compose” page	56
A.17	Screenshot for “List of data sets” page	57
A.18	Screenshot for “Add data set” page	58
A.19	Screenshot for “Decision Trees” page	59
A.20	Screenshot for “Selective Decision Trees” page	60
A.21	Screenshot for “Support Vector Machines” page	61

LIST OF ABBREVIATIONS

MP	missing persons
MPP	Missing Persons Project
SPS	Saskatoon Police Service
GUI	graphical user interface
MVC	Model View Controller
SVM	Support Vector Machine
CPS	child protective services
MY	missing youths
CV	cross-validation
RBF	Radial Basis Function
DFB	distance-from-boundary
CART	Classification and Regression Tree Analysis
NB	Naive Bayes
KNN	K-Nearest Neighbor

CHAPTER 1

INTRODUCTION

In January 2016, the Saskatoon Police Service (SPS), the University of Saskatchewan, and the Government of Saskatchewan announced a partnership to implement the Saskatoon Police Predictive Analytics Lab (SPPAL) (SPS, 2016). The purpose of this research partnership was to focus on missing persons (MP), especially youths at risk of running away from home, in order to improve responses to missing youths (MY) cases. The SPPAL partner agencies have been working towards a better understanding of the risk factors surrounding young people and families that are particularly vulnerable to go missing. To this end, various machine learning techniques are used to classify, treat, and aid the management of MY cases. The research in this thesis, produced within the SPPAL, explores a classification method based on the machine learning method of Support Vector Machines (SVMs) to predict the risk that a youth who has gone missing before to go missing for a second time and the risk that a missing youth who has gone missing more than once has an association to a criminal street gang.

1.1 Machine Learning

From a practical and theoretical standpoint, a task of major importance to computer science researchers has been to build systems that can learn from past experiences. Carrying out any task using a computer requires a sequence of instructions or “algorithms”. However, these algorithms are not able to solve many problems without a prior mathematical model. In the 1990s, scientists began creating algorithms for computers that analyze large amounts of data and learn from the results. This field of study that enables computers to learn from data and even improve themselves, without being explicitly programmed is called machine learning (Alpaydin, 2009).

Machine learning algorithms are typically divided into several broad categories, such as supervised, semi-supervised, active, unsupervised, and reinforcement learning, depending on whether the data are labeled or not and other factors. For example, supervised learning is the process of an algorithm learning from a training data set where the user provides the desired inputs and outputs. In contrast, unsupervised learning is the process of algorithms discovering and presenting a structure for the training data set. Supervised learning problems can be further categorized as *classification* and *regression* problems. In classification problems, the input is divided into one or more classes, and the outputs are to be assigned to one or more of these

classes (Murphy, 2012). The classifiers used in a classification problem can be categorized as *linear* and *nonlinear*. Linear classifiers can be particularly useful because they support an efficient training process for classifying documents (Yu et al., 2012). Linear classifiers can be employed to classify large data sets with a large number of examples as well as a large number of features within a sparse data matrix (Fan et al., 2008).

1.1.1 Support Vector Machines

This thesis applies an SVM classifier for analysis of the MY database, predicting whether or not a missing youth is likely to go missing again and whether they are likely to be involved with gang activities. SVMs were introduced by Vapnik and Cortes (Vapnik and Cortes, 1995) for solving classification problems. SVM is a supervised classifier and has been successfully shown to apply to a variety of classification problems (Henderson et al., 2000). The goal of using the SVM classifier is to produce a model that predicts the target values of the test data given only the test data attributes. It achieves this goal by finding the “best” way to separate the training data, thus giving the best chance of new data being classified correctly. SVMs are designed to simultaneously minimize classification error and maximize separation between classes. In more technical terms, SVMs map the input vectors into a higher-dimensional feature space and maximize the *margin* (the distance between the separator (hyperplane) and the nearest data points of each class in the space (Xu et al., 2009)).

There are many strategies used to separate data. In this thesis, *linear* SVM strategies are used, that map a data point to a vector space, then apply a linear classifier within this space. A more detailed theoretical overview of SVMs is provided in section 3.1. Additionally, the reader is invited to consult section 3.2 for detailed performance measurements of the SVM model, including the accuracy, sensitivity, specificity, false negative rate, false positive rate, and the area under the receiver operating characteristic curve.

1.2 Saskatoon Police Service Missing Youths Database

Conducting machine learning research on MY data requires an understanding of the physical and logical structure of the data set and how information is accessed. Thanks to our partnership with the Saskatoon Police Service, we have received access to detailed information on the Saskatoon missing youths data, including standard definitions of data elements, their meanings, and allowable values. All information is located within the MY database, which contains 434 MY cases with 91 features for each case. The logical and physical relationship diagram for the MY GUI is provided in chapter 3. Section 3.3 provides information that is particularly pertinent to the police force in that it provides access to the analytic models needed to investigate the MY cases.

The purpose of this thesis was to create and develop a user interface for the SPS with predictive model capabilities that is applied to the MY database. Section 3.4 gives a description of the MY database, the features that may be important, and the tables with which they are associated. We note that throughout

this thesis, the term “feature” is used to describe the categories of information collected about each MY case.

1.3 Missing Youths Graphical User Interface

The MY graphical user interface (GUI) gives police officers access to statistical tools that can analyze different fields of the MY database and provide graphical results for their questions. The GUI also allows officers to create reports easily and quickly from the database in a user-friendly manner. This system offers various machine learning algorithms that can be applied to the SPS data set to gain insight into patterns and perform predictive analysis.

The MY GUI architecture is designed using Model View Controller (MVC) structures. The MVC is a way of splitting up an application so it is easier to change pieces of the implementation according to various needs quickly and efficiently. In the Missing Persons Project (MPP), the GUI is developed by a type of MVC structure named “migrate” ([Deacon, 2013](#)) and implemented by using a Python template engine called “Jinja2” ([Ronacher, 2008](#)). This template engine brings together the modules and packages that allow rapid building of an application without involving low-level details such as database connections. The system has been programmed in the Python programming language ([Lutz, 2013](#)) using the CherryPy web framework ([Hellegouarch, 2007](#)). A SQLServer connection code is used to provide a connection to the MY database ([Larson et al., 2016](#)). Section 3.5 provides the process of the design and development of GUI for the MY database.

1.4 Ethics

Because the database contains the personal information of a large number of individuals who belong to vulnerable populations, the data in this thesis are highly sensitive, and great care was taken to uphold a high ethical standard of anonymity and privacy. The results given in this thesis do not divulge aspects of the data that could be seen as sensitive or proprietary. The University of Saskatchewan ethics file that contains the necessary permissions to work with the data is BEH# 16-166.

CHAPTER 2

LITERATURE REVIEW

This chapter provides the background information associated with the main concepts of this thesis. Section 2.1 is a description of SVM machine learning techniques to produce a classification model as well as some applications of these models. Section 2.1.1 is a discussion of the SVM classifier as a classification model. Section 2.2 explains how machine learning techniques have helped officers to solve missing youths cases. Section 2.3 is background information about how a user interface can provide an easy and secure way for data analysis and reporting amenable for use by the police services.

2.1 Machine Learning

Machine learning can be defined as the study of algorithms that a computer system can use to learn and adapt to new data without human involvement (*Pang et al., 2002; Nguyen and Armitage, 2008; Carbonell et al., 1983*). Machine learning can be used for both predictive and descriptive purposes. For example, machine learning has had important applications in the field of medicine (*Marr, 2016*). Since the 1990s, certain types of diagnostic issues have been solved through the use of SVM linear classifiers (*Kim and Na, 2018*). *Kim and Na* demonstrated that machine learning classification approaches can be applied to risk prediction of disease through the analysis of brain MRI structure (neuroimaging). The authors showed that individual-level classifications can be provided using the machine learning-based brain MRI approach. Before the development of such techniques, there was no direct way to translate the findings of brain MRI structure analysis to clinical practice. The brain MRI application using SVMs was able to obtain between 67.6% and 90.3% diagnostic accuracy.

Another machine learning study was done by *Furey et al.* This study introduced a model based on SVM methods to identify sets of yeast genes with a similar function from expression data. *Furey et al.* examined various SVM models with different similarity metrics. The models were implemented to classify genes through gene expression. The data set used for this study had 2,467 gene records with 79 different DNA microarrays for each of them (*Furey et al., 2000*). The SVM kernel function provided the best predictions aimed to identify the function of unannotated yeast genes, better than four other machine learning methods, namely Parzen windows (*Bishop et al., 1995*), Fisher's linear discriminant (*Duda and Hart, 1973*), and two decision tree learners (*Furey et al., 2000*). This thesis focuses on how SVM methods and predictive analytics might

inform, assist, and improve decision making for youth protection.

2.1.1 Support Vector Machines

Support Vector Machines have been proposed as a learning algorithm in different areas for classification. SVM methods have been applied to audio classification and retrieval research in 2003. Based on this research, the problem of audio classification was tackled by building an SVM with a binary tree recognition strategy (*Guo and Li, 2003*). Audio retrieval was introduced with a new metric, the distance-from-boundary (DFB). In the binary tree recognition strategy, first, the system finds an inside boundary for the audio data. Then, the system calculates the DFB for each instance of audio data, and finally, SVMs can be trained for all DFBs (*Guo and Li, 2003*). The results of this research show that audio classification can be effectively learned through the use of SVMs and can achieve a low error rate. However, the computational cost of the DFB is high in the case of having a large number of support vectors.

In another study by *Bazi and Melgani*, SVM classifiers were examined for hyper-spectral remote sensing images. In this study, a genetic optimization framework was built to automatically determine the best SVM classifier parameters. The hyper-spectral imagery classification proposed was based on SVM methods. The objective of this system was to optimize the SVM classifier and calculate the accuracy. This meant that the system was required to detect a subset of the best discriminative features and solve the selection issue for the SVM classifier. To reach this objective, an optimization framework based on a genetic algorithm (GA) was used. GA was an effective optimization method to retrain a large number of solutions for each iteration, and it was optimized directly into the objective function (*Bazi and Melgani, 2006*). In this experiment, two classifiers, namely SVM-GA-SV and SVM-GA-R2W2, were used to provide a controlled environment. The SVM-GA-SV classifier was compared with the SVM-GA-R2W2 classifier in terms of the probability of detecting a set of discriminative features among the noisy ones and leading to classification accuracy. According to *Bazi and Melgani*, the SVM-GA-SV classifier was better able to detect noise and led to higher classification accuracies than the SVM-GA-R2W2 classifier. First, the SVM-GA-SV experiment detected 43 features for each class using the SVM classifier and was applied directly to the hyperdimensional space. Then, the SVM classifier parameters split the training data set into k subsets and were separately modeled on the two classifiers. The overall accuracy result for the SVM-GA-SV experiment was 87.66% (*Bazi and Melgani, 2006*). The SVM-GA-R2W2 experiment detected 133 features for each class using the SVM classifier and obtained significant accuracy changes compared to the other classifier. The overall accuracy result for the SVM-GA-R2W2 experiment was 91.05% (*Bazi and Melgani, 2006*). In the next chapter, the SVM linear classifier model is applied to the MY database and explained in more detail.

2.2 Missing Youths Data

For the purpose of this thesis, a missing person (particularly youths) refers to an individual who has been reported as missing within a few hours or up to months to a policing agency. In 2005, *Pfeifer* employed the Saskatchewan police policies and practices to analyze the Saskatchewan MP database. Based on the analysis of MP files, Saskatchewan agencies filed 4496 MY reports in 2005. However, the data set used had 2956 records for each MY case. This means that some youths were reported more than once that year, indicating that the youth was a possible runaway. In addition, the data in this record illustrate a clear age distribution trend, with a significant number of cases between the ages 9 and 18 years.

Fyfe et al. report that 80% of MY cases are resolved within 24 hours. Police agencies have performed two different actions that have reduced the number of runaways by 10%. The first action is to schedule a monthly discussion addressing at-risk youths. The second is to provide ongoing support and service to the Provincial Task Force of Missing Persons (*SPS, 2015*). According to the MP Partnership Committee of Saskatchewan, youths are the highest risk group of MY in Saskatchewan (*STO, 2007*). In this partnership, the main issue related to individuals (particularly youths) at risk is to understand the roles and responsibilities that families, communities, and police can play in preventing or responding to MY situations. Knowing this, the MP Partnership Committee has provided health education programs, community councils, and publications to build awareness of MY target groups. For example, the health education program has suggested on-going education throughout the high school years to provide information about the risks associated with MY cases. Additionally, the “Child Find Alert Magazine” has provided regular information on the motivating factors and measures available in preventing youths from running away from home (*STO, 2007*). Based on the 2005 *STO* Saskatchewan missing youths annual report, 64% of MY cases were reported, and the majority of them appear as runaways cases. The number of MY cases increased by 16% between 1968 and 2005 (*STO, 2007*). *Bonny et al.* examined the relationship between a risk level assigned to the MY and understanding risk level and risk factors of their behavioral act. According to this study, the risk level assigned for each MY case was significantly associated with age and mental health (*Bonny et al., 2016*).

Russell and Macgill show that development of predictive analysis approaches might improve the potential of decision making about youth protection. Research by *Sledjeski et al.* in 2008 introduced a Classification and Regression Tree Analysis (CART) model based on the child protective services (CPS) system. The CPS system provides appropriate services for child and families to reduce the likelihood of child maltreatment (*Brookes and Webster, 1999*). The results of this study showed that neglect and multi-type abuse (the initial type of maltreatment) were the two most common forms of child abuse in families at 48% and 21%, respectively. Psychological and medical abuse were together ranked as the third most common form at 14%. Physical and sexual abuse were identified as the least common forms of abuse at 11% and 6%, respectively (*Sledjeski et al., 2008*).

2.3 Graphical User Interface

Recent research (*Redmond and Baveja, 2002*) has argued that one of the critical points for police departments to deal with crime issues is the sharing of data informally throughout the police investigative process. Based on this research, police departments must create reasonable strategies and initiatives to improve their methods of dealing with violence. For many years, police departments have used software and analytical tools for their problem-solving tasks. An application carries out a particular task for police systems using a GUI. A GUI can develop visual indicators for decision-making and generate graphical icons to share information locally or globally. For example, police departments have employed emerging technologies for collecting and using information through graphical icons and visual indicators to develop new problem-solving software (*Redmond and Baveja, 2002*). Data mapping tools (e.g., Google My Maps) have been used to map and visualize a variety of police data in a GUI. Data visualization can help to improve police planning and problem solving. The use of GUI for the tracking of arrests, crimes, crime types, criminal history, and missing persons are some examples of how the police use software for problem-solving tasks as well as automated fingerprint identification (*Van Duyn, 1991*) and computer-aided dispatch systems (*Sparrow, 1993*). This software can provide officers with more data to respond to a problem and generate a faster response (*Redmond and Baveja, 2002*).

Information and communication technologies can be developed faster than ever before (*Pramanik et al., 2017*). Many different security organizations provide preventive measures software in order to predict future crimes and criminals based on data collected during criminal investigations. Various data mining techniques, such as machine learning, neural networks, and intelligent agents for classification and prediction, have been designed since the 1980s (*Burt, 1980*). Research by *Pramanik et al.* has shown that the strategies most effective in preventing criminal re-offense are the extraction of hidden network structures among criminals and the inference of their respective roles from criminal information. In recent years, many possibilities for criminal investigation and forensic science have been offered by machine learning algorithms. These machine learning algorithms can be quickly applied on a database by the use of a GUI without prior knowledge of data analysis. This thesis develops the MY GUI for the basis of an analysis and reporting system intended for use by the Saskatoon Police Service.

CHAPTER 3

MACHINE LEARNING: METHODS AND IMPLEMENTATION

Machine learning can be defined as enabling computers to build models for prediction, classification, or simulation using past experiences. There are many classification techniques, such as Naive Bayes (NB), K-Nearest Neighbor (KNN), and SVMs, that can be applied to classify data into distinct groups. In this thesis, the SVM classifier is employed to analyze the MY database. This chapter details how the SVM classifier can build predictive models using the MY database. In this thesis, the SVM classifier built two predictive models using the variables *missing_again* and *gang_involvement* from the MY database as the classification variables. The variable *missing_again* has been used to build a model to predict youths who are likely to go missing again. The variable *gang_involvement* has been used to build a model to predict youths who are likely to have gang involvement.

Section 3.1 describes SVM kernels and classification models, and introduces the model designed for two classification variables. Section 3.2 calculates the confusion matrices of the SVM classification models for the MY database. Section 3.3 describes the model validation that employed in the construction of each predictive model in this thesis. Section 3.4 outlines the MY database and the process of data querying employing the classification features. Section 3.5 provides information on the importance and the process of development for the GUI. Screenshots of the MY GUI are provided in Appendix A.

3.1 Support Vector Machines

An SVM is a classifier based on hyperplane separators. The SVM model is based on the idea of mapping the feature vectors onto a high (possibly infinite) dimensional space and then utilizing linear models in this new space. It is a supervised learning method, meaning that the model must be built by sample data that have already been classified. SVMs have been highly successful in a variety of applications, such as handwritten digit recognition (*Scholkopf and Smola, 2002; Decoste and Scholkopf, 2002*), categorization of Web pages (*Dong et al., 2005*), and face detection (*Joachims, 1998*). SVMs are particularly effective when dealing with continuous data or data sets that are linearly separable (*Scholkopf and Smola, 2002*). For large data sets with a large number of features, the cost of training and prediction is high. In the case that the data are linearly separable, the linear classifier can build a model to make a prediction. In the process of modelling, because both training time and performance are concerns, a simple linear classifier may be sufficient because

it is convenient to use (it usually does not involve the tuning of many parameters) ([Huang and Lin, 2016](#)).

SVMs are a technique that build a learning model based on computing posterior probabilities ([Awad and Khanna, 2015](#)). SVM method is known for its good generalization ability and robustness, making it one of the most popular machine learning methods ([Awad and Khanna, 2015](#)). The method was introduced in 1992 and soon became known as one of the best classifiers and a powerful prediction tool. In general, SVMs have shown better results on popular benchmark problems than neural networks and other statistical models ([Kecman, 2005](#)).

The mathematical foundations of SVMs make it suitable for classifying data of relatively high dimension. The oldest and simplest machine learning model is the *linear* model ([Awad and Khanna, 2015](#)). The linear model takes multiple input features and creates weights that are used to provide predictions. The SVM linear classification model can be found by applying the linear kernel method to the input data during the training process until a sufficiently small error is found. For this purpose, a small error is required because it ensures the algorithm is able to generalize the classifier to test data without over-fitting. In this thesis, for example, the small size of the MY database for the *gang_involvement* classification feature may lead to overfitting, but using a simple linear kernel and the cross-validation (CV) method may reduce the risk of overfitting. Then, an optimization algorithm is applied to find the minimum difference between the estimated and true values of the model ([Corrigan, 2018](#)). Next, the model learns from the training set how to classify new data (the testing set) according to the same criteria.

In this thesis, the most important notion behind SVM models is the idea that among the infinitely many hyperplanes that may separate the data, the chosen linear classifier is the one that maximizes the separation of the data, i.e., the one whose distance from it to the nearest data point on each side is maximized ([Steinwart and Christmann, 2008](#)). The use of linear classifiers in the transformed space depends heavily on the computational methods used to find a classifier that performs well on the training data. Another important feature to distinguish itself from features of data is its good accuracy on the training set.

The problem of finding the optimal hyperplane is an optimization problem. Figure 3.1 illustrates an SVM classifier. In this figure, there are many possible hyperplanes, but the solid black line is the one that maximizes the margin (the distance between the hyperplane and the nearest data point of each class). The dashed lines identify the optimal separating hyperplanes with the largest margin. The selected data points nearest to the possible separating hyperplanes for a given training set are termed support vectors (SVs) ([Steinwart and Christmann, 2008](#)). The SVs are points of a data set that, if removed, would alter the position of the optimal separating hyperplane.

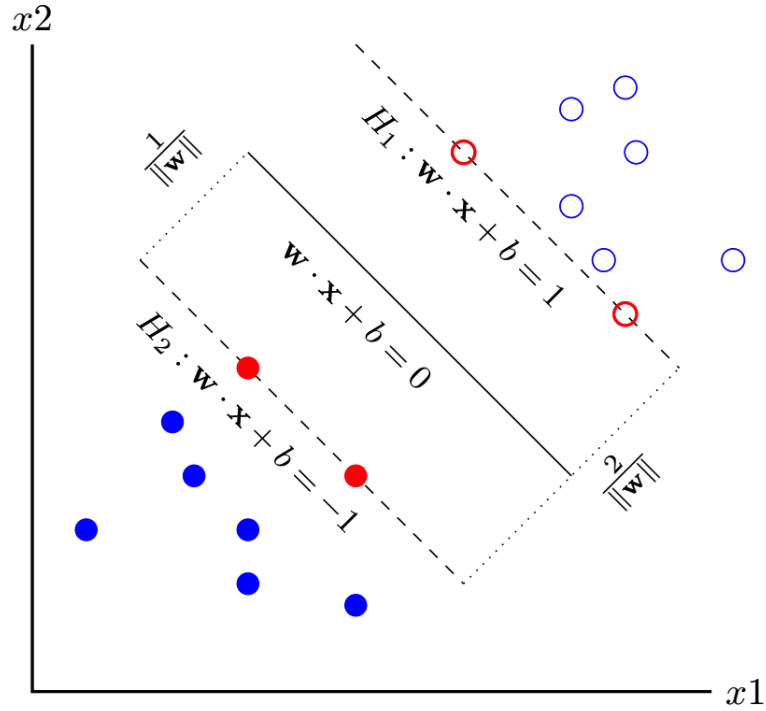


Figure 3.1: How SVMs linearly classify data in a two-dimensional space

In technical terms, an SVM classifier for training examples labeled as belonging to two classes maps the input vectors $\mathbf{x}_i \in \mathbb{R}^n$, $i = 1, 2, \dots, m$, into a higher-dimensional feature space and maximizes the margin (Xu *et al.*, 2009). The given data points take the form

$$\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), (\mathbf{x}_3, y_3), \dots, (\mathbf{x}_m, y_m)\},$$

where $y_i = \pm 1$ is the classification variable. For example, the class label $+1$ may correspond to a person who has gone missing on more than one occasion (*missing_again*) and the class label -1 may correspond to those who have not.

The margin is defined as the distance of the hyperplane to the nearest of the positive and negative data points. The optimal hyperplane formula takes the form

$$\mathbf{w} \cdot \mathbf{x} + b = 0,$$

where the weight vector \mathbf{w} is the normal vector to the hyperplane. The optimal hyperplane can scale the length of \mathbf{w} to force the closest point in the positive area to have inner product 1 and the closest point in the negative area to have inner product -1 . So, the supporting hyperplanes can be written as:

$$\mathbf{w} \cdot \mathbf{x} + b \geq +1 \quad \text{or} \quad \mathbf{w} \cdot \mathbf{x} + b \leq -1 \quad \text{for} \quad y_i = \pm 1, \quad \text{respectively.} \quad (3.1)$$

The two inequalities in (3.1) can be combined into one equation as:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1. \quad (3.2)$$

Equation 3.2 is called the functional margin and defines whether a training sample is properly classified.

Now consider in Figure 3.1 the SVs that lie on the hyperplanes, H_1 and H_2 , which are parallel to the optimal hyperplane.

Geometrically, the distance between these two hyperplanes is $\frac{2}{\|\mathbf{w}\|}$. So, to define an optimal hyperplane we need to maximize the width of the margin. The margin m can be defined as $m = \frac{1}{\|\mathbf{w}\|}$. In other words, to maximize the margin m , we can minimize $\|\mathbf{w}\|$. So, we can formulate the optimization problem as:

$$\min \frac{1}{2} \|\mathbf{w}\|^2. \quad (3.3)$$

The general method to solve this problem is called *quadratic programming* optimization ([Chang and Lin, 2011](#)).

In the case where the data do not offer linearly separable problems in the input space, they can become linearly separable problems via a nonlinear mapping into a higher-dimensional space. Nonlinear SVMs can be realized by the kernel mapping method to simulate a nonlinear projection of data into a higher-dimensional space where the classes are linearly separable ([Mercier and Lennon, 2003](#)). When data are linearly separable, the linear classifier is used to find a perfect classifier. To find a perfect classifier, every data point (\mathbf{x}_i, y_i) satisfies the functional margin (3.2). If the data are not linearly separable, SVM can use “slack” variables that allow constraint violations. The algorithm then attempts to minimize the misclassified data using the slack variables as follows:

$$\min_{\mathbf{w}, b, \xi_i} \left(\frac{1}{2} \|\mathbf{w}\|^2 + 1 + C \sum_{i=1}^m \xi_i \right), \quad \text{subject to} \quad y_i(\mathbf{w} \cdot \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i \quad \xi_i \geq 0. \quad (3.4)$$

In Equation (3.4), a slack variable is used to penalize all classification mistakes made in the training. The minimization in Equation (3.4) tries to balance the best of both worlds: maximizing the separation of classes while minimizing the amount of misclassification. Figure 3.2 shows an example of misclassified data using the ξ_i variables.

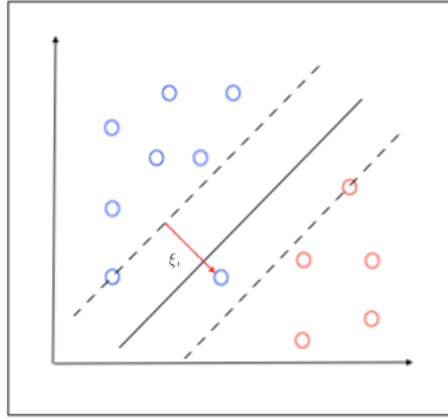


Figure 3.2: Misclassified data using the ξ_i variables

The simplest way to separate two groups of data is with a hyperplane. However, there are situations where a nonlinear surface can more efficiently separate the groups. Kernel methods are the way to provide a maximized margin and support nonlinear separation ([Yu et al., 2003](#)). SVMs are regarded as an active field of machine learning and have been the main application of kernel methods. Kernel methods map input data points into a higher-dimensional space. Figure 3.3 shows an example of mapping data into a higher-dimensional space using a kernel method.

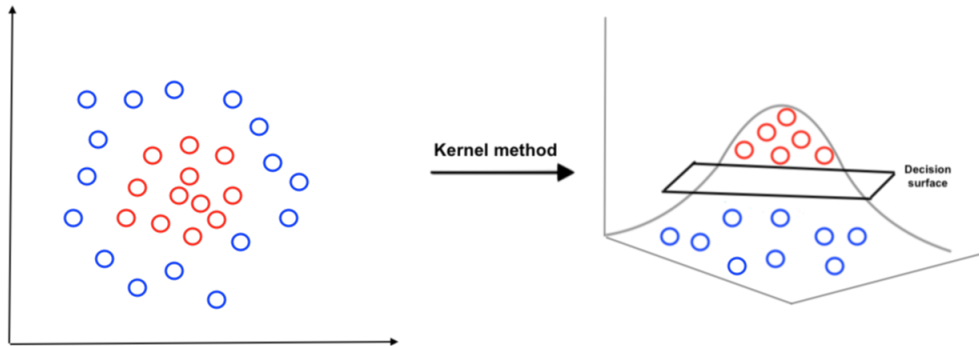


Figure 3.3: Visualization of mapping data in a higher-dimensional space using a kernel method

Kernel methods attempt to allow us to perform linear classification on nonlinear features of the data without explicitly generating the features. Choosing a kernel method depends strongly on the data specifications ([Awad and Khanna, 2015](#)). A great number of kernels exist, and they are categorized into two types: *local kernels* and *global kernels*. With local kernels, kernel values can be affected only by the nearest

data points. However, with global kernels, kernel values can be affected by the faraway data points too (*Smits and Jordaan, 2002*). In the nonlinear SVM, the n -dimensional input \mathbf{x} are implicitly transformed into a (higher) \tilde{n} -dimensional feature space using a transformation function (ϕ), a procedure that is called the kernel trick (*Schölkopf, 2001*).

The kernel trick can be applied to nonlinear data to make them linearly separable. The ultimate goal of the model is to find a transformation function that creates $(\phi(\mathbf{x}_i), y_i)$ in a new feature space with a separable hyperplane. In the new feature space, we have:

$$\mathbf{w} \cdot \phi(\mathbf{x}_i) + b \geq 1.$$

then we can find an optimal separating hyperplane by solving the optimization problem in the new feature space.

There are four most common SVM kernel methods. The first kernel is the *linear kernel*, which is regarded as a simple and useful method for classification and regression in large group of support vectors, defined as $k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)$. Figure 3.4 illustrates some contours of the linear kernel and shows that it can separate data linearly with a single line in a feature space (*Schölkopf, 2001*).

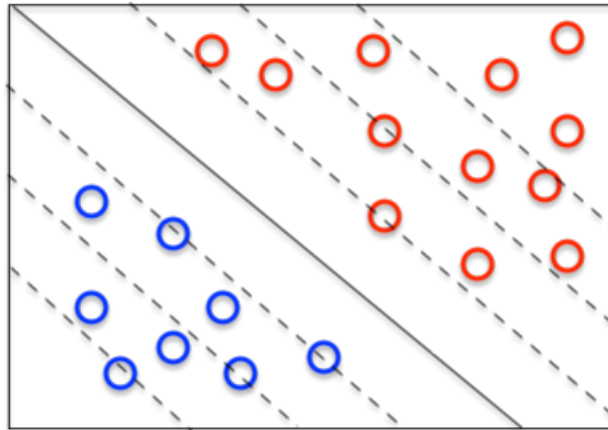


Figure 3.4: Visualization of SVM linear kernel

The second kernel is the *polynomial kernel*, widely used in image processing and defined as $k(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \mathbf{x}_i \cdot \mathbf{x}_j + C)^n$, $\gamma > 0$, where γ and C are the regularization parameters that are chosen via validation, and n is a dimensional parameter. Figure 3.5 illustrates some contours of the polynomial kernel (*Schölkopf, 2001*).

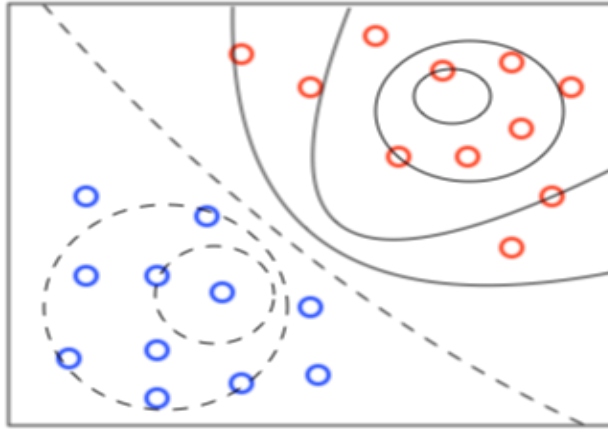


Figure 3.5: Visualization of SVM polynomial kernel

The third kernel is the *radial basis function* (RBF) kernel, defined as $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2)$, $\gamma > 0$. The γ parameter defines how far the influence of a single training example reaches. The value of γ is chosen by the LIBSVM package as default value. Figure 3.6 illustrates the RBF kernel ([Schölkopf, 2001](#)).

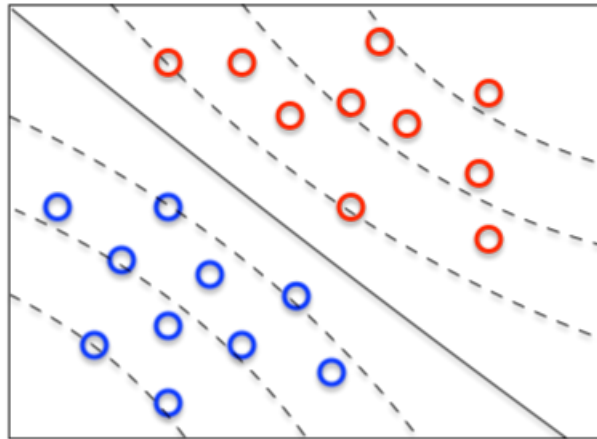


Figure 3.6: Visualization of SVM radial basis function kernel

The fourth kernel is the *sigmoid kernel*, mostly used for neural networks. The sigmoid kernel is the least popular kernel because of its poor performance ([Ren et al., 2016](#)). The sigmoid kernel is defined as $k(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\gamma\mathbf{x}_i \cdot \mathbf{x}_j + c)$, where γ and c are regularization parameters that are chosen via validation. Figure 3.7 illustrates contours of the sigmoid kernel ([Schölkopf, 2001](#)).

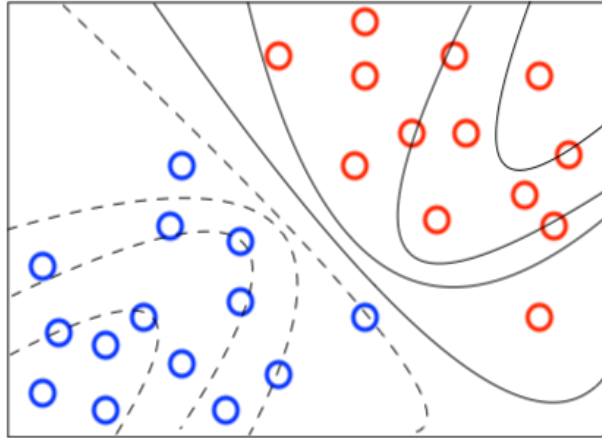


Figure 3.7: Visualization of SVM sigmoid kernel

One way to choose an appropriate kernel and kernel parameters is through cross-validation. In the SVM kernel methods, the parameter C is a regularization term, which provides a way to control overfitting. The reverse defines how far the influence of a single training example reaches. The values of C and γ are chosen by the LIBSVM package as default values. Table 3.1 provides a comparison of different kernel methods for the MY Database. The results show the linear kernel method performs better than the other kernel methods for the MY database for the specific default values of the C and γ parameters.

Type of kernel method	Linear	Polynomial	RBF	Sigmoid
optimal parameters	$C=1; \gamma=0.01$	$C=1; \gamma=0.01$	$C=1; \gamma=0.01$	$C=1; \gamma=0.01$
AUC	0.92	0.89	0.61	0.38

Table 3.1: Comparison of different kernel functions for the MY database

In the linear SVM classification model, sometimes over-fitting problems occur when the sample data size is too small. The CV method is used to deal with over-fitting problems. The linear SVM works best when the data contain a small number of features. The 10-fold CV function is performed as a method to find the best model using LIBSVM. Section 3.3.2 provides a description of how CV was used in this study.

3.1.1 Linear SVM Classification Algorithm

The SVM classification model is implemented here with the classification method called Support Vector Classification (SVC) using the Python programming library (*Bergstra et al., 2015*). LIBSVM is an open-source machine learning library that is used for the SVC method implementation. LIBSVM is applied in two steps to build the SVC model. The first step is to train a model on a data set. The second step is to use the model to make predictions on a testing data set. LIBSVM is implemented using the Scikit-Learn

python package. The Scikit-Learn package is designed as a machine learning library for the numerical and scientific Python programming libraries (*Pedregosa et al., 2011*). The Scikit-Learn package contains built-in classes for different SVM classification models. A basic structure of how to build and use an SVM is given in Algorithm 1 (*Hsu et al., 2003; Chang and Lin, 2011*).

Algorithm 1 Linear SVM classification model for specific case of two classes

- 1: **procedure** LINEAR SVM(data)
 - 2: Structure the data into features and class labels $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$.
 - 3: Divide the data into training and testing sets.
 - 4: Define an optimal hyperplane using SVC: maximize margin.
 - 5: (If the hyperplane cannot separate data linearly) Extend the linear classification definition for non-linearly separable problems: have a penalty term for misclassifications.
 - 6: (If the hyperplane cannot separate data linearly) Map data to higher-dimensional space where it may be easier to classify with linear classifier: reformulate problem so that data are mapped to this space.
 - 7: Validate the model by computing the percentage of correct classifications, etc., on the testing set.
-

Algorithm 1 describes the process used to build and use an SVM predictive model. The first step (line 2) is to structure the data into features and class labels. For example, all the columns of the data are stored in the \mathbf{x} variable except for *missing_again* and *gang_involvement*, which are used as class labels and stored in the y variable for the two different analyses performed. The second step (line 3) is to divide the data into two data sets, the training data set used for building the model and the testing data set used for validating the model. The third step (line 4) is to define an optimal hyperplane using SVC from the training data set. In practice, this is performed by finding the maximum margin hyperplane. If the hyperplane cannot separate data linearly, the fourth step (line 5) is to have a penalty term for the misclassified data points. If the hyperplane cannot separate data linearly, the fifth step (line 6) is to map nonlinearly separable data in a higher-dimensional feature space to classify with a linear classifier. The sixth step (line 7) is to validate the predictive capability of the model by, for example, computing the percentage of correct classifications (true positives) or accuracy (*Azadeh et al., 2013*). For a more detailed step-by-step breakdown of the process of building SVMs, see (*Hsu et al., 2003*).

3.2 Model Performance Evaluation

After training and validation of the model, the next step is to find out how effective the model is in terms of its performance on a test set. Different metrics are used to evaluate the performance of a model (*Muller and Guido, 2017*). The following sections focus on the metrics used in this study to estimate the performance of the constructed models.

3.2.1 Confusion Matrix

The performance of a classifier can be visualized by a matrix known as the confusion matrix (*Kononenko and Kukar, 2007*). The rows and columns of the confusion matrix present observed and predicted class labels, respectively. The confusion matrix for a class feature with two unique values of 1 and 0 is

		Predicted Value	
Observed Value	1	True Positive (TP) = 11	False Negative (FN) = 10
	0	False Positive (FP) = 01	True Negative (TN) = 00

Table 3.2: A confusion matrix

where 1 \equiv Positive, 0 \equiv Negative, true positive (TP) is the number of testing instances observed in class 1 that are correctly predicted to be in class 1, false negative (FN) is the number of testing instances observed in class 1 that are incorrectly predicted to be in class 0, true negative (TN) is the number of testing instances observed in class 0 that are correctly predicted to be in class 0, and false positive (FP) is the number of testing instances observed in class 0 that are incorrectly predicted to be in class 1. Using the confusion matrix, one can easily obtain the accuracy and some other performance measures such as sensitivity and specificity in terms of the entries of the confusion matrix as follows.

3.2.2 Accuracy

The accuracy of a model tells us that what portion of the testing data is correctly classified. In terms of confusion matrix elements, accuracy is defined by the following formula

$$\mathcal{A} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (3.5)$$

where the numerator is the total number of correctly predicted outcomes, and the denominator is the total number of instances.

3.2.3 Sensitivity and Specificity

The sensitivity or true positive rate (TPR) and the specificity or true negative rate (TNR) are two other measures that show to what extent a classifier is correct in classifying testing data as positive and to what extent all positives are classified correctly (*Costa et al., 2007*). The sensitivity and specificity are defined by

$$TPR = \frac{TP}{TP+FN}, \quad TNR = \frac{TN}{TN+FP}. \quad (3.6)$$

Having sensitivity and specificity, false positive rate (FPR) and false negative rate (FNR) become

$$FPR = 1 - \text{specificity}, \quad FNR = 1 - \text{sensitivity}. \quad (3.7)$$

The false positive rate (FPR) is the number of cases incorrectly predicted as positive. The false negative rate (FNR) is the number of cases incorrectly predicted as negative.

3.2.4 Receiver Operating Characteristic Curve

The Receiver Operating Characteristic (ROC) curve is a tool that commonly used to illustrate the performance of a binary classifier (*Carter et al., 2016*). In World War II, the British Royal Air Force developed a ROC curve method for the radar signal detection and to find the different signals of interest (*Carter et al., 2016*). The ROC curve is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. It is created by plotting the (TPR) (sensitivity) against the (FPR) ($1 - \text{specificity}$) at various threshold settings. The area under the ROC curve (AUC) can be used to compare the performance of multiple classifiers. A classifier with a higher AUC has better performance. The recommended tool for the accuracy in statistical analysis is the AUC (*DeLong et al., 1988*).

Using the data from the MY database, a statistical model was developed to predict from whether a missing youth is likely to go missing again or whether a missing youth is likely to have gang affiliations.

3.3 Model Validation

In general, a predictive model can be validated in various ways. In the following sections, the two methods of simple split and cross-validation employed in the construction of each predictive model in this study as validation methods are briefly reviewed.

3.3.1 Simple Split

In the simple split method, the original data are randomized and split into two groups, 80% for the training set and 20% for the testing set. In this method, the samples are selected with uniform distribution, meaning that each sample has the same probability for being selected (*Qiang and Zhongli, 2011*).

3.3.2 Multi-fold Cross-validation

Cross-validation is the method that results in the best selection of the hyper-parameters compared to the other split methods, but for a large database, it may have a high execution time. One way to reduce risk of over-fitting is to use a multi-fold CV process, so a model is not simply over-fit to a training subset. Particularly given the large number of features considered, there is a higher likelihood of over-fitting, and it is not acceptable to simply use one designated subset for training and another for testing. In this thesis, a CV is used as a re-sampling method. This method refits a model of interest as samples formed from the training set in order to obtain additional information about the fitted model. For example, it can provide estimates of test-set prediction error, and the standard deviation and bias of the parameter estimates.

CV is a statistical method that aims at minimizing the probability of over-fitting and creating a more unbiased model (*Kononenko and Kukar, 2007; Refaeilzadeh et al., 2009*). Another goal of CV is to select the best model among different training algorithms (*Reitermanov, 2010*). Also, CV can be used to find the optimal parameters of models with different levels of complexity (*Reed and Marks, 1998*), such as Random Forest Classifications with different depths or number of Decision Tree Classifications in the forest. CV has many methods, but k -fold CV is the most popular one (*Murphy, 2012*). The reason for the popularity of k -fold CV is that all observations are used for both training and validation. In addition, each observation is used exactly once for validation. So, it is helpful to reduce overfitting.

In the k -fold CV method, the original data are divided into k subsets with nearly equal sizes. For example, the MY database contains 434 observations divided into 10 folds. So, there are 6 folds that contain 43 observations and 4 folds that contain 44 observations. In k -fold CV, $k - 1$ subsets form the training set, and the remaining subset forms the validation set. The model is trained based on the training set for a total of k times, and the prediction error is estimated using the validation set. This process is repeated for each subset, and the average of the prediction error values is determined using

$$\mathcal{E}(i) = \frac{\sum_{j=1}^k \mathcal{E}_j}{k}.$$

The main disadvantage of k -fold CV is that depending on the value of k , this method can be computationally expensive (*Muller and Guido, 2017*). The choice of the number of folds depends on the computation time and the number of samples in the data. With a larger value of k , the error estimation tends to be more accurate, but the process can be computationally expensive. Usually, researchers choose $k = 10$, but for a large number of samples it is better to choose a smaller value for k in order to decrease the computation time (*Reitermanov, 2010*). In this thesis, a combination of both methods of simple split and 10-fold CV can be used by the following steps (*Ozkan, 2017; Hastie et al., 2008*):

Step 1: Split the randomized original data into training and testing sets.

Step 2: Use k -fold CV on the training set to build the model. In this step, by considering the least average prediction error on the test set, the best parameters for the model can be chosen.

Step 3: Evaluate the model performance using the testing set.

3.4 Description of the MY Database

The purpose of this section is to describe a set of theoretical and empirical risk factors surrounding different aspects of MY investigations, a description of each feature found within the tables in the MY database, and a set of descriptive statistics for the features within the database. The MY database was initially received from SPS in the form of 11 linked tables. There are the main data tables, the translation tables, and the tables containing extra details about the data table to which they are linked. A classification feature is a set of features on which a prediction model is trained. In this thesis, the classification features along with a

variety of other features associated with missing youth and their cases were pulled (using an SQL query) from all available tables to make up the missing youth data set. All other available data from the MY database were used to ensure no bias was introduced by preselecting features by hand. The code used to query the MY data is written in MySQL (*Greenspan and Bulger, 2001*). The final step to build a predictive model was to encode the output data to a binary value. This does not change what the data represent but rather only changes the format in which they are presented and allows for easier manipulation within the algorithm.

The final data set contained 434 missing youths cases reported between September 27, 1971, and February 18, 2016, with 91 distinct features. The feature *missing_again* contains 171 instances of youth who went missing more than once (and accordingly, 263 instances of youth who only went missing once). The feature *gang_involvement* contains 71 instances of youth with suspected gang involvement (and accordingly 363 instances of no suspected gang involvement). The full extent of the features cannot be disclosed due to confidentiality.

3.4.1 Classification Features

In this thesis, a missing person is a person who has been formally reported to the police as someone who has gone missing. In 2016, *Bonny et al.* in their research indicate that the personal behaviours of a youth and family violence are the main reasons for youths to go missing or run away. The actual number of missing youths cases is lower than the number of missing youths reporting files. This indicates that some cases are for youths who go missing more than once in a year (*Pfeifer, 2006*).

Youth gang involvement is also a serious problem for many law enforcement agencies in the United States. More than one million youths are involved in gang in the United States every year (*Walters, 2019*). Youths at risk of running away are more likely to engage in gang affiliations.

Based on the studies (*Bonny et al., 2016; Pfeifer, 2006; Walters, 2019*), the classification features *missing_again* and *gang_involvement* are respectively chosen to build predictive models to indicate when a youth is likely to go missing again and is likely to be involved in gang activity. It is also important to identify the features that could be most important to answering those two questions. The predictive models are trained by the classification features. In this section, the two classification features that are used to build the predictive models for the MY database are presented.

The first classification feature is *missing_again*. If a youth has gone missing more than once, this feature has a value of 1, otherwise it has a value of 0. Using *missing_again* as a classification feature allows us to train a predictive model to predict whether or not a youth is likely to go missing again. The second classification feature is *gang_involvement*. If a youth is believed to have *gang_involvement*, this feature has a value of 1, otherwise it has a value of 0. Using *gang_involvement* as a classification feature allows us to train a predictive model to predict whether or not a missing youth is likely to be involved in gang activity. These classification features along with 91 features associated with the MY and their case are merged into one table that is used to train the predictive model.

3.5 Graphical User Interface Development

This section provides the process of the design and development of the MY GUI. The main concept of the MY GUI is to provide a visual grouping of data with minimal effort required regarding user actions.

3.5.1 Missing Youths Graphical User Interface Development

A GUI provides a user-friendly environment featuring elements, such as menus, buttons, text boxes, and lists, for easy access for the end users. However, a GUI can be complex and hard to develop, debug, and modify for the programmer. A programmer can attract users by creating a strong interaction between front-end and back-end design. In order to have a desirable visualization, a GUI needs to be beautiful and functional. The front-end design is a visualization for the programmer to divide the design and development worlds. GUI development requires that the programmer deal with created graphics and multiple ways of giving the same commands to the input devices. The important point of GUI development is to pay extensive attention to the needs and limitations of the end users.

GUI development can be divided into five high-level steps. The first step is to analyze the user needs. The second step is to create a development plan and find a possible path of development. This step involves a schematic image of the screens, linked together through a visual service. The third step is to find an appropriate platform. During the GUI development, different platforms lead to different tools. For instance, for a web-based application, design tools such as HTML, JavaScript, and .css are common. The fourth step is to evaluate the GUI. The main goal of this step is to verify that the GUI meets the needs of the users. The fifth step is to determine if the GUI is complete. The process of GUI development is stopped when the evaluation process is no longer generating any new requirements, or a small number of relatively unimportant requirements (*Bischofberger and Pomberger, 2012*).

There has been significant progress in software tools to help with creating a GUI. The MY GUI structure is programmed in the Python programming language using a Python template engine called Jinja2 (*Ronacher, 2008*). Python is a high-level programming language that focuses on code readability for web-based development. Python has a large number of web libraries and frameworks that make development of code short and simple (*Lokhande et al., 2015*).

The MY GUI is built by the CherryPy object-oriented web framework (*Hellegouarch, 2007*). CherryPy is a Python library providing a GUI to the HTTP protocol. CherryPy was chosen because of its simplicity, self-containment, and non-intrusiveness. The program is easily able to narrow and expand the coverage of the libraries, and it is an independent module that can be designed by developers (*Shinde and Patel, 2014*). The front-end of the MY GUI is programmed in HTML, JavaScript, and CSS (*Ferguson and Heilmann, 2013*).

The main concept of the MY GUI is to provide a visual analysis and grouping of data with minimal effort required regarding user actions. Appendix A contains the screenshots of the selection of the most important aspects of the GUI, representing a static demo, along with textual descriptions of their purpose and contents.

In this appendix, a mock database is used for demonstration purposes because the MY database is a secure data set.

3.5.2 Logical Diagram

The logical diagram for the MY GUI represents information gathered from the SPS requirements. Developing a logical diagram for the MY GUI affords a clear understanding of how the GUI operates. The MY GUI can be a lightweight web application, where all functionality is grouped visually and logically into thematic units. The benefit of the logical diagram for the MY GUI is that it is more easily understandable to non-technical people. Figure 3.8 represents the logical diagram for the MY GUI.

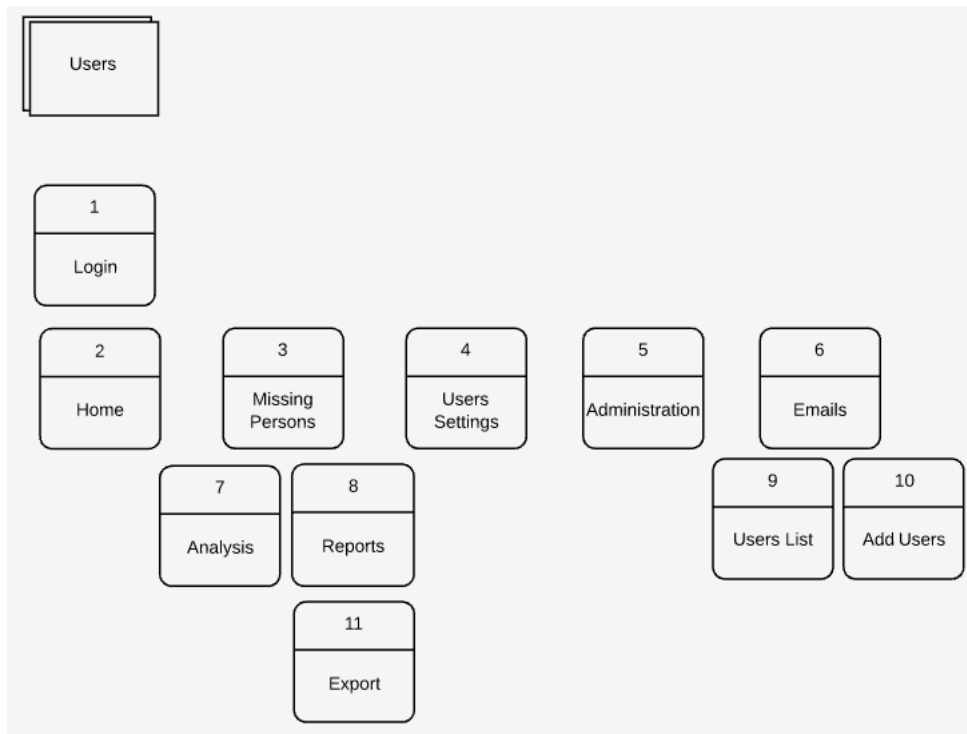


Figure 3.8: Logical diagram for the MY GUI

3.5.3 Physical Diagram

The logical development of a GUI is used to create a physical diagram of the system. The physical diagram shows the structure of possible navigation paths and connections and the system functionality through the GUI. In the MY GUI development, the logical and physical diagrams both contain entities and relationships, but they differ in the purposes for which they are created and the audiences they are meant to target. The physical diagram of the MY GUI can be seen in Figure 3.9.

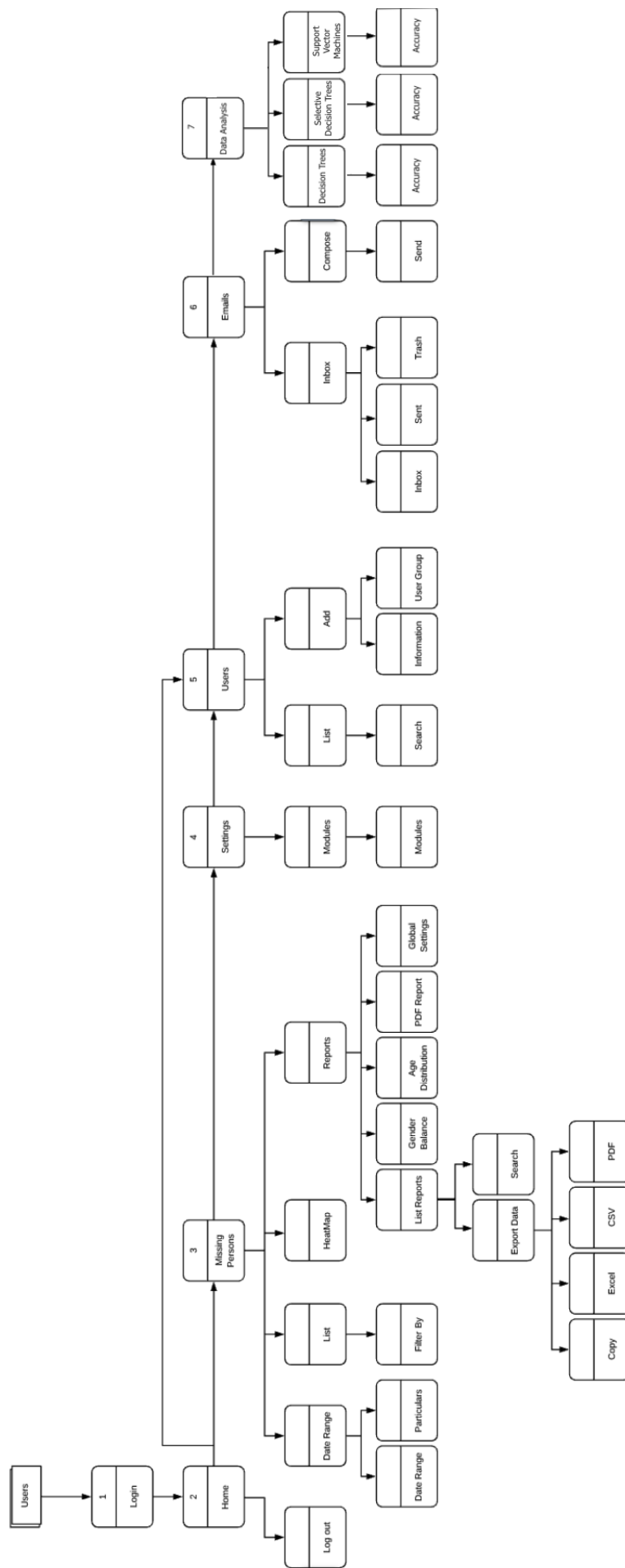


Figure 3.9: Physical diagram of the MY GUI

CHAPTER 4

RESULTS

The results of some data analysis including the SVM classifier is discussed in this chapter. The MY database contains 434 MY cases records with 95 features. It is valuable to use some features such as surname, g1 (first name), g2 (middle name), and dob (date of birth) from the MY database to determine unmatched MY cases. For example, a shared family name can give indications as to gang affiliation beyond for a single case. Table 4.1 contains some descriptive statistics about the MY cases.

Total	Male	Female	Missing Again	Gang Involvement
434	155	279	171	71

Table 4.1: Basic information about the MY cases

4.1 Basic Statistical Analysis of the Missing Youths Database

Repeat missing youths are an important issue for the Ministry of Justice of Saskatchewan and the SPS. In this section, some basic descriptive statistical analysis, e.g., means, standard deviations, modes, and medians, are provided for the samples of the MY database.

The focus on MY data is to find out the MY cases age groups, missing period, occurrences, and the time to the next event for each gender. These statistical analyses can provide a significant amount of information to contribute to addressing MY issues. The age of each MY is calculated based on the difference between their date of birth from the *dob* column and the date from the *date_missing* column in the MY database. The results are shown in Table 4.2. The mean age of the MY is 13.8 years with a standard deviation of 3.2 years, and the mode and median age of the MY are both 15 years.

Mean	Standard Deviation	Mode	Median
13.8	3.2	15	15

Table 4.2: Basic statistics for age (years)

Figure 4.1 gives a histogram of the ages for the people at the centre of the MY cases, from which we observe a high number of cases involving teenagers.

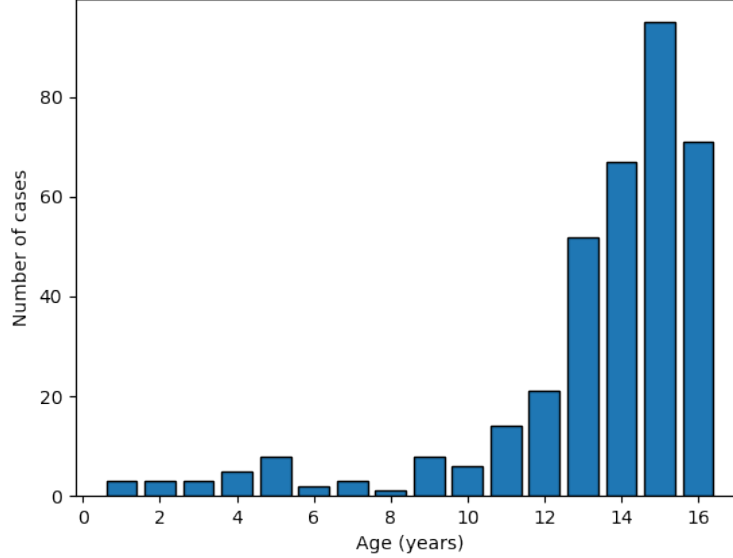


Figure 4.1: Histogram of age

Table 4.3 shows the mean, standard deviation, mode, and median results for male and female missing youth in different age groups. The most common (mode) age of missing youths for males is 13 years and for females it is 15 years.

Gender	Mean	Standard Deviation	Mode	Median
Male	13	3.9	13	14
Female	14.2	2.6	15	15

Table 4.3: Basic statistics for age grouped by gender (years)

The time missing for each MY is calculated based on the difference between the date from the *date_located* column and the date from the *date_missing* column in the MY database and excluded the cases where there was no *date_located*. The results are shown in Table 4.4. The mean of the time missing is 3.1 days with a standard deviation of 7.7 days, and the mode and median age of MY are 0 (less than 24 hours) and 1 day, respectively. The MY database has around 30% of missing cases that do not have a *date_located* entry.

Mean	Standard Deviation	Mode	Median
3.1	7.7	0	1

Table 4.4: Basic statistics for time missing (days)

Figure 4.2 gives a semi-log histogram of the time missing for the number of the MY cases, from which we observe a high number of cases are resolved within one day (24 hours).

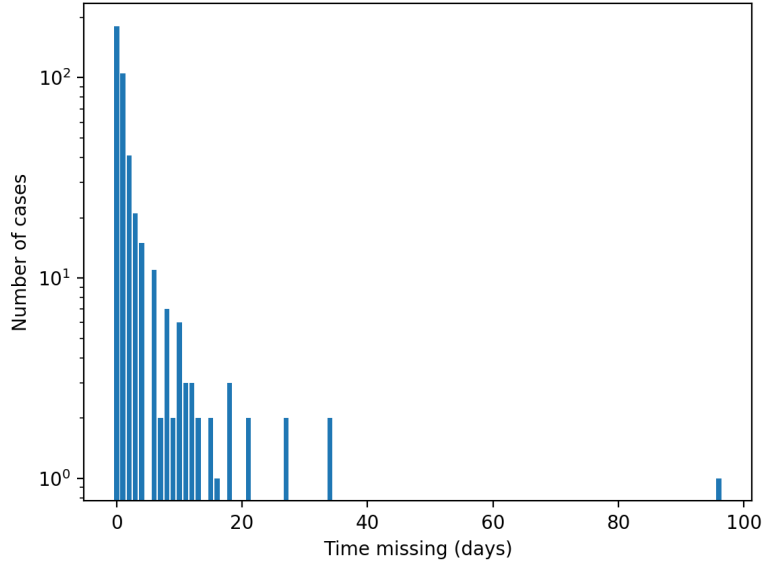


Figure 4.2: Histogram of time missing

Table 4.5 shows the mean, standard deviation, mode, and median results for time missing grouped by gender. The most common time missing for both males and females are less than a day, indicating that most missing youths are found within 24 hours.

Gender	Mean	Standard Deviation	Mode	Median
Male	2.8	6.6	0	1
Female	3.3	8.2	0	1

Table 4.5: Basic statistics for time missing grouped by gender (days)

The average number of repeat events of each incident of a missing youth is calculated based on the *missing_again* column in the MY database. The results are shown in Table 4.6. The mean of the repeat events is 4 times with a standard deviation of 3.8 times, and the mode and median age of MY are 2 and 4 times, respectively.

Mean	Standard Deviation	Mode	Median
4	3.8	2	4

Table 4.6: Basic statistics for repeat occurrences (number of times)

Figure 4.3 gives a histogram of the repeat events for the number of the MY cases, from which we observe a high number of youths who are reported missing twice.

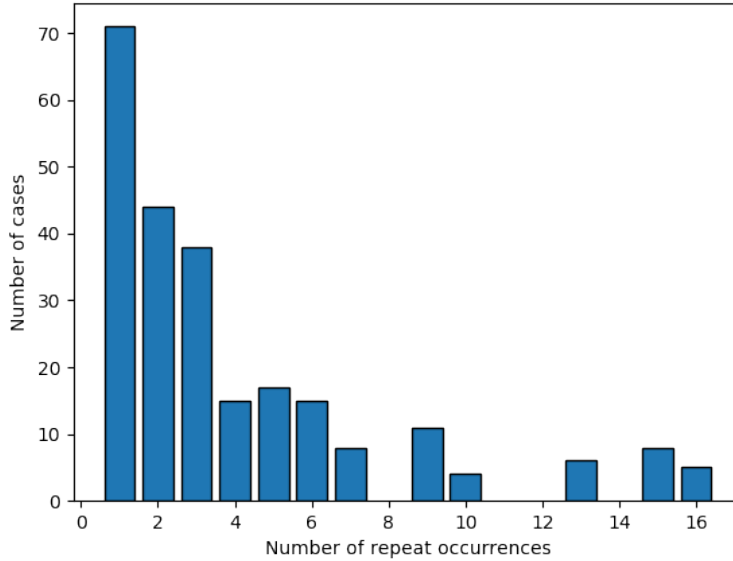


Figure 4.3: Histogram of repeat occurrences

Table 4.7 shows the mean, standard deviation, mode, and median results for repeat occurrences grouped by gender. The most common number of the repeat events for males is 1 time and for females it is 2 times.

Gender	Mean	Standard Deviation	Mode	Median
Male	3.3	1.5	1	3
Female	5.7	4.3	2	4

Table 4.7: Basic statistics for repeat occurrences grouped by gender (number of times)

The time to the next event of each MY is calculated based on the *missing_again*, *surname*, *g1* (first name), *g2* (nickname), and *date_located* columns in the MY database. In this analysis, *g1*, *g2*, and *surname* of each case are required to make sure that different nicknames are not used in different occurrences. The results are shown in Table 4.8. The mean of the time to the next event is 584 days (1.6 years) with a standard deviation of 511 days (1.4 years), and the mode and median age of MY are 73 days (0.2 years) and 401 days (1.1 years), respectively. In this analysis, there are two different situations for missing youth cases that do not have any next event. The first situation is that a youth ages out of the youth age category. In this case, there is no more information about the case. The second situation is that a missing youth case has not gone missing or run away again. To handle these situations, each case which does not contain *date_located* has been removed from the MY database.

Figure 4.4 gives a histogram of the time to next event for the number of the MY cases, from which we see a high frequency of time to the next event of within three months.

Mean	Standard Deviation	Mode	Median
584	511	73	401

Table 4.8: Basic statistics for time to the next event (days)

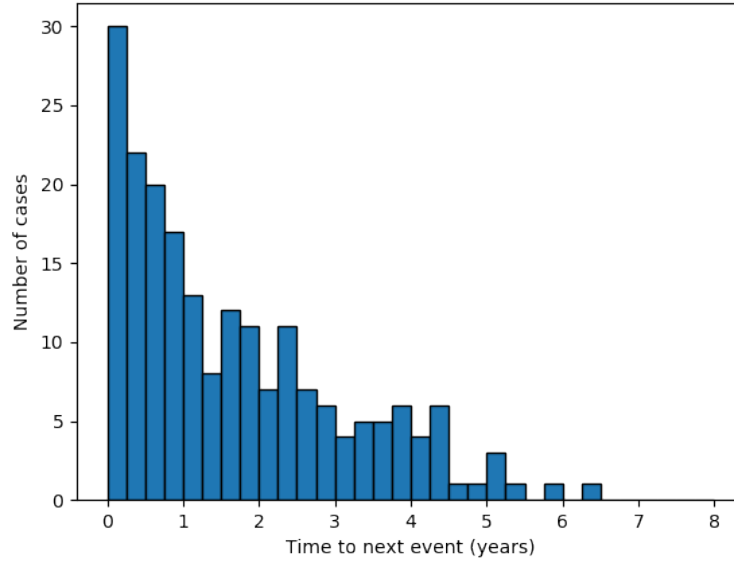


Figure 4.4: Histogram of time to next event

Table 4.9 shows the mean, standard deviation, mode, and median results for time to the next event grouped by gender. The most common time to the next event for males is 58 days and for females it is 98 days. Based on the repeat occurrences result, female missing cases are repeated two times more than those for males.

Gender	Mean	Standard Deviation	Mode	Median
Male	511	547	58	368
Female	620	1460	98	730

Table 4.9: Basic statistics for time to the next event grouped by gender (days)

4.2 Missing Youths Experiment Results

In subsections 4.2.1 and 4.2.2, the results of the SVM classifier accepting a combination of 91 input features for each MY case from the MY database as input and *missing_again* and *gang_involvement* as two classification features are discussed. The results for the SVMs have been measured according to accuracy, sensitivity, specificity, FNR, and FPR. The SVM classifier was able to obtain an 89% accuracy for the *missing_again*

classification feature and an 84% accuracy for the *gang_involvement* classification feature.

4.2.1 Results for the missing_again classification feature

Table 4.10 shows the results for the SVMs applied to the MY database for the *missing_again* classification feature with the simple split and the 10-fold CV method. The sample size of youths who go missing more than once is 171 out of the total 434 cases, and accordingly there are 263 instances of youth who go missing only once.

In the 10-fold CV method, the training set is broken down randomly into 10 folds. For each validation, the first fold is treated as a testing set and the model is fit on the remaining 9 folds. Based on the results, the accuracies are in the range of 82% for fold 9 as testing set to 94% for fold 6 as testing set. Also, the FNR are in the range of 8% for fold 6 as testing set and 30% for fold 9 as testing set. In this case, fold 6 has been selected as leading to the best model because it has the highest accuracy and low FNR in the training set. Looking at the results produced for the *missing_again* classification feature in Table 4.10, the SVM model was able to obtain high accuracies and high sensitivities using the 10-fold CV. The higher numerical value of sensitivity indicates the low likelihood of diagnostic false-positive results. So, we conclude that the linear SVM classifier model is a model with relatively few misclassified data points for the MY database.

CV	Accuracy	Sensitivity	Specificity	False Negative Rate	False Positive Rate
fold 1	91.9	86.4	96.0	13.5	4.0
fold 2	91.9	83.8	96.4	16.1	3.5
fold 3	89.6	88.8	90.1	11.1	9.8
fold 4	83.9	87.0	82.1	12.9	17.8
fold 5	88.5	88.5	88.4	11.4	11.5
fold 6	94.2	91.8	96.0	8.1	4.0
fold 7	88.5	83.3	90.4	16.6	9.5
fold 8	86.2	71.4	96.1	28.5	38.4
fold 9	81.6	70.2	90.0	29.7	9.9
fold 10	91.9	90.6	92.7	9.3	7.2

Table 4.10: SVMs results for the *missing_again* classification feature (%) with 10-fold cross-validation

4.2.1.1 Confusion Matrix for missing_again Classification Feature

A confusion matrix is used to evaluate empirically the results for the linear classification and estimate the test-set prediction error. This method provides probabilities using the extracted prediction errors. In

Figure 4.5, the classification performance of the *missing_again* classifier is summarized using a confusion matrix. According to the normalized confusion matrix of the *missing_again* classifier, the TPR is 93, which means 93% of youth in this class are correctly predicted as who have gone missing. The FNR is 7, which means 7% of youth in this class are incorrectly predicted as who have gone missing. The TNR is 91, which means 91% of youth are correctly predicted as who have not gone missing. The FPR is 9, which means 9% of youth are incorrectly predicted as who have not gone missing.

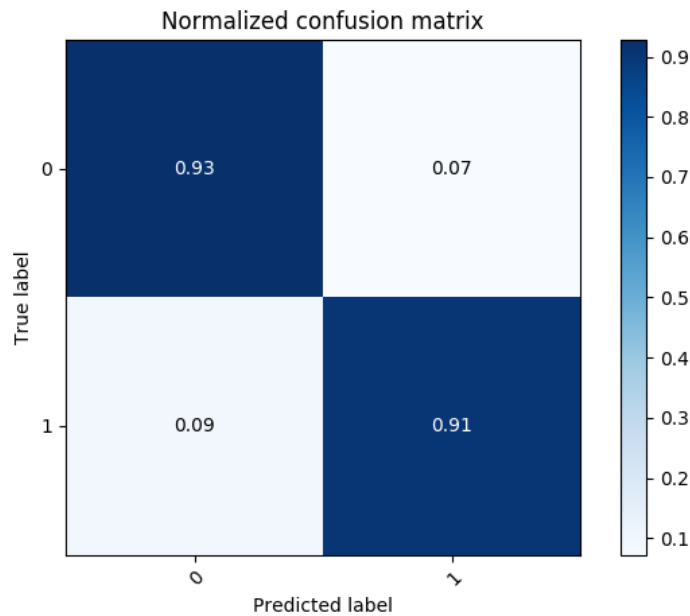


Figure 4.5: Confusion matrix for the *missing_again* classification feature

4.2.1.2 ROC Curve for the *missing_again* Classification Feature

Figure 4.6 refers to the diagnostic performance of the accuracy of a test to discriminate where youths are at a high risk to go missing again. In this figure, the area under curve is equal 92%; we consider this to be good at separating youth who have gone missing more than once from youth who have not. The correct positive results represent the values plotted above of the equality line (denoted by the dotted green line).

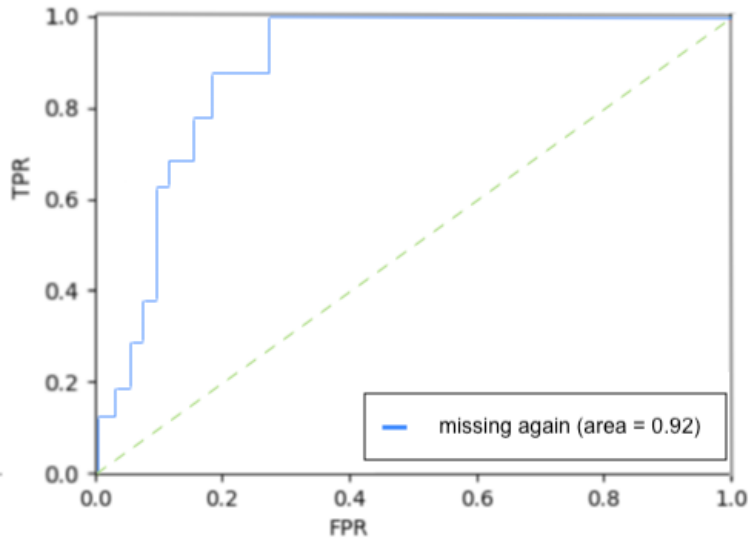


Figure 4.6: Receiver Operating Characteristic curve for the *missing_again* classification feature

4.2.2 Results for the *gang_involvement* Classification Feature

Table 4.11 shows the results for the SVMs applied to the MY database for the *gang_involvement* classification feature with the simple split and the 10-fold CV. There are the sample size of youths who were suspected of gang involvement is 71 out of the total 434 cases, and accordingly there are 363 instances of youth who were not suspected of gang involvement.

In the 10-fold CV method, the training set is broken down randomly into 10 folds. For each validation, the first fold is treated as a testing set and the model is fit on the remaining 9 folds. When 9 of those folds are used as the training data set, relatively few samples remain for testing. This would lead to greater variability in the estimates. Based on the results, the accuracies are in the range of 79% for folds 1 as testing set and 3 to 87% for folds 4 and 10 as testing set. Also, the FNR are in the range of 36% for fold 4 as testing set and 71% for fold 1 as testing set. So, fold 4 has been selected as leading to the best model because it has the highest accuracy and a relatively low FNR in the training set. Looking at the results produced for the *gang_involvement* classification feature in Table 4.11, the SVM model was able to obtain high accuracies but low sensitivities using the 10-fold CV. The lower numerical value of sensitivity indicates the high likelihood of diagnostic false-positive results.

SVMs	Accuracy	Sensitivity	Specificity	False Negative Rate	False Positive Rate
fold 1	79.3	28.5	89.0	71.4	10.9
fold 2	86.2	56.2	92.9	43.7	7.0
fold 3	79.3	45.4	84.2	54.5	15.7
fold 4	87.3	64.2	91.7	35.7	8.2
fold 5	80.4	47.0	88.5	52.9	11.4
fold 6	82.7	33.3	90.6	66.6	9.3
fold 7	80.4	42.8	87.6	57.1	12.3
fold 8	86.2	52.9	94.2	47.0	5.7
fold 9	83.9	50.0	89.3	50.0	10.6
fold 10	87.3	53.8	93.2	46.1	6.7

Table 4.11: SVM results for the *gang_involvement* classification feature (%) with 10-fold Cross-validation

4.2.2.1 Confusion Matrix for the *gang_involvement* Classification Feature

A confusion matrix is used to evaluate empirically the results for the linear classification and estimate the test-set prediction error. This method provides probabilities using the extracted prediction errors. In Figure 4.7, the classification performance of *gang_involvement* classifier is summarized using a confusion matrix. According to the normalized confusion matrix of *gang_involvement* classifier, the TPR is 93%, which means 93% of youth in this class are correctly predicted as having gang involvement. The FNR is 7%, which means 7% of youth in this class are incorrectly predicted as having gang involvement. The TNR is 54%, which means 54% of youth are correctly predicted as not having gang involvement. The FPR is 46%, which means 46% of youth are incorrectly predicted as not having gang involvement.

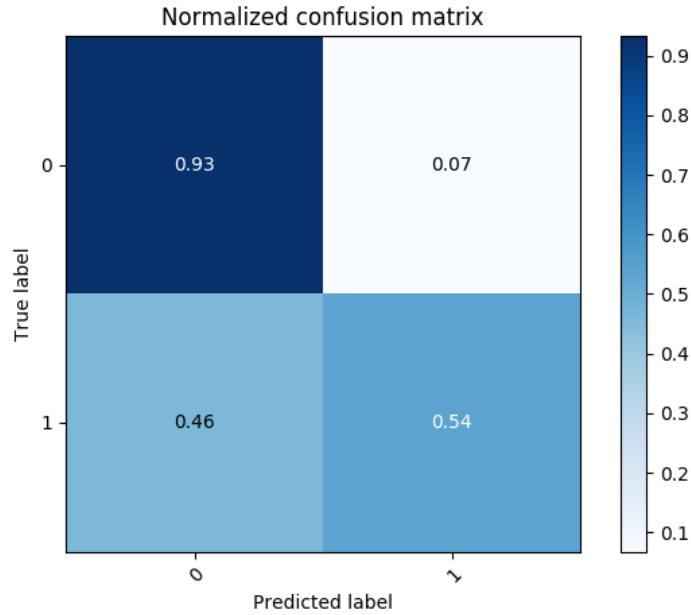


Figure 4.7: Confusion matrix for the *gang_involvement* classification feature

4.2.2.2 ROC Curve for the *gang_involvement* Classification Feature

Figure 4.8 refers to the diagnostic performance of the accuracy of a test to discriminate whether a missing youth is likely to have gang affiliations. In this figure, the AUC is 74%; we consider this to be good at separating youth who have gang involvement from youth who have not.

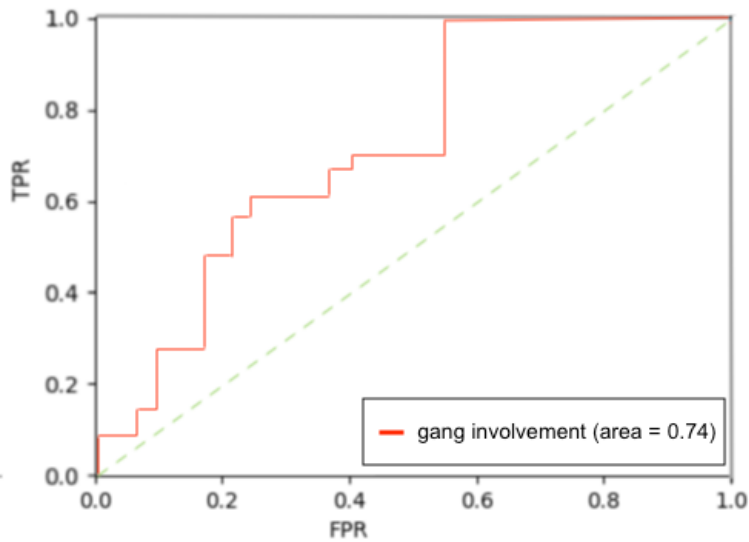


Figure 4.8: Receiver Operating Characteristic curve for the *gang_involvement* classification feature

4.2.3 Discussion

Table 4.12 shows the results of the best CV using SVMs validated on the testing set of the MY database. The SVM classification model for the *missing_again* classification feature obtained an accuracy of 89%. An SVM feature selection library (scikit-learn) for the *missing_again* classification feature is used for the best CV model and then applied to the testing set resulted in providing some predictive features such as *missing_period*, *cause_of_absence*, *missing_per_year*, *age_last*, *hair_color*, *citizenship*, *occ_date*, *date_located*, *age_group*, *disability* (Weston et al., 2001). An SVM was able to correctly identify youth who are likely to go missing again with 84% sensitivity and 92% specificity. SVMs are able to identify youth who are likely to go missing again with a 16% FNR. The FNR measures how often youth who are identified as not at a high risk to go missing actually go missing again. The FNR shows that 16% of MY who have not gone missing more than once have actually gone missing more than once. This information may be useful for officers to consider when deciding whether or how to intervene in cases where a youth is at a high risk to go missing again.

The SVM classification model for the *gang_involvement* classification feature obtained an accuracy of 84%. An SVM feature selection library (scikit-learn) for the *gang_involvement* classification feature is used for the best CV model and then applied to the testing set to provide some predictive features such as *place_name*, *vehicle_type*, *location*, *place_last_seen*, *cause_of_absence*, *missing_period*, *employer*, *missing_per_year*, *sex*, *height*, *hair_style*, *occupation*, *family_violence*, *gang_type*, and *week_day* (Weston et al., 2001). It was able to correctly identify youth who are likely to be involved with gangs with 43% sensitivity and 90% specificity. Looking at the results obtained, the false negative rate was high. SVMs are able to identify youth who are likely to be involved with gangs with 57% FNR. The FNR measures how often youths who are identified as not likely to be involved in gangs are actually involved in gangs. The FNR shows that 57% of MY who are not identified to be involved in gangs are actually involved in gangs. The sample size of *gang_involvement* is 71 out of 434 cases. When 80% of those samples are used in the training data set, relatively few samples remain for testing. In this case, another machine learning method such as Decision Trees might be more suitable (Quinlan, 1987).

Support Vector Machine Results	Accuracy	Sensitivity	Specificity	PPV	NPV	FPR	FNR
Missing Youths <i>missing_again</i>	88.8	84.1	91.8	77.7	80.9	8.2	15.9
Missing Youths <i>gang_involvement</i>	83.8	42.9	90.1	44	89.9	9.9	57.1

Table 4.12: SVM results in percent (%)

Table 4.12 shows that the SVM classification model for the *missing_again* classification feature has a smaller FNR than that for the *gang_involvement* classification feature. Also, the experimental results indicate that the SVM classifier is working effectively for the MY database especially for the *missing_again* classification feature in terms of accuracy. Generally, it is beneficial to use both the simple split and 10-fold CV in one model to provide a solution to the problem of over-fitting. Table 4.12 shows the results for the best 10-fold CV model tested on the 20% of testing set. The effectiveness of the SVM classifier is to a large extent dependent on finding an appropriate separating hyperplane. The statistical results in section 4.1 shows that teenage girls are particularly associated with the risk of missing or running away.

Comparing the results obtained by both classification features provides useful insight into possible effects of the sample size in the performance of the model. The initial observation is that the SVM model for the *missing_again* classification feature using 171 samples performed better than SVM model for the *gang_involvement* classification feature using 71 samples in term of accuracy. A small sample size is prone to “overfitting”. In this situation, adding more data points into the model should lead to better results.

The right censoring problem is another problem for the MY database. In basic statistical analysis, the situation when the value of a feature is only partially known is called censoring. In this thesis, the study is conducted to predict the number of repeat events of each incident of the MY. Based on the MY database, the age group for the MY cases is under 18 years. So, censoring applies to the MY cases of youth who are more than 18 years old. Also, censoring occurs when the MY case is no longer reported as missing because of death or illness or simply omitted in the database. In the other case, the study is conducted to predict the time to the next event of MY cases. In the MY database, the *date_located* value is Null in some cases that means all the MY cases have not reported as solved. In this situation, censoring occurs when there is no *date_located* for the missing report. In this thesis, the above situations have been ignored.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

The main goal of this thesis was to introduce the SVM classifier for the MY database and to implement a graphical user interface for it. A series of experiments was designed and conducted to develop the MY data set provided by the SPS and to evaluate the performance of the SVM classifier. In the first part of this study, basic information is provided about the structure of the MY database and the process of the SVM classification model application for the database.

Machine learning is a tool that provides computer systems the ability to learn and improve their performance of learning (*Michalski et al., 2013*). The machine learning method used in this thesis was SVMs. It is important to understand how the SVM classification model is built. An SVM classifier is built for the MY database. This model is used with two different classification features, namely *missing_again* and *gang_involvement*. An SVM classification model is applied and its validity tested using the MY database. The SVM model was applied using the Python programming language. The SVM classification model is able to produce high accuracy for both classification features based on a split and 10-fold cross-validation of the data. The SVM classifier was able to model the *missing_again* classification feature better than the *gang_involvement* classification feature.

In the second part of this thesis, the process of development of a graphical user interface for the MY database was carried out. The analysis section was implemented in the MY GUI based on SVM classification model and two other machine learning methods (see Appendix A). The design and development of the MY GUI offer the possibility to apply various machine learning algorithms to the MY database to gain insight into patterns and perform predictive analysis. Furthermore, a reporting system for the MY database was developed to provide quick and easy access to the data for officers and data analysts.

5.2 Future Work

In this thesis, to apply a machine learning model to character data, the data must first be converted to numerical values. The time taken to clean and prepare the data could have been reduced by more rigorous data entry protocols, accentuating why proper data entry is critical. To help with data entry in the future,

a few suggestions to improve the database are given.

First, the MY database has many variables that are not filled in properly. This creates problems when building machine learning models that rely on features being filled in consistently. In the future, some of these gaps will be filled in by integrating the Saskatchewan Ministry of Social Services. The Social Services data contain the samples of the youths who have gone missing from care. Having good data collection practices is key in data analytics and when using machine learning techniques to aid in predictive policing.

Second, if the tables were more clearly documented or queries of the data were pre-determined and stored in the database, it would save coding and computation time. Implementing a simpler database structure and adhering to its structures will create cleaner data with which to work.

Third, a standardized method for filling in feature fields that are character based would be optimal. For example, in the feature field describing the location from which an individual went missing, there are a variety of ways that is filled in, e.g., with an address, “friend’s house”, or “group home”. Eliminating this inconsistency will allow for future machine learning techniques to more effectively use these features.

In the future, the addition of data from the Saskatchewan Ministry of Social Services to the MY data may lead to improvements in the predictive models. Features that have missing values in the current data set may be able to be filled in by the addition of new data that include family life and drug/alcohol use. As future directions, the MY GUI will incorporate the Social Services data structure for use in predictive analyses and user interface development and identify the linking methodology and overlap between police and social services data sources.

BIBLIOGRAPHY

- (2007), Final report of the provincial partnership committee on missing persons, available at <http://publications.gov.sk.ca/documents/9/30559-missing-persons-final.pdf>.
- (2015), Saskatoon police service 2015-2019 business plan, available at https://saskatoonpolice.ca/pdf/general/Business_Plan_2015-19.pdf.
- (2016), Saskatoon police service 2016 annual report, available at [https://saskatoonpolice.ca/pdf/annual_reports/SPS_Annual_Report_2016_\(web\).pdf](https://saskatoonpolice.ca/pdf/annual_reports/SPS_Annual_Report_2016_(web).pdf).
- Alpaydin, E. (2009), *Introduction to machine learning*, MIT Press.
- Awad, M., and R. Khanna (2015), *Efficient learning machines: theories, concepts, and applications for engineers and system designers*, Apress.
- Azadeh, A., M. Saberi, A. Kazem, V. Ebrahimipour, A. Nourmohammadzadeh, and Z. Saberi (2013), A flexible algorithm for fault diagnosis in a centrifugal pump with corrupted data and noise based on ANN and support vector machine with hyper-parameters optimization, *Applied Soft Computing*, 13(3), 1478–1485.
- Bazi, Y., and F. Melgani (2006), Toward an optimal SVM classification system for hyperspectral remote sensing images, *IEEE Transactions on geoscience and remote sensing*, 44(11), 3374–3385.
- Bergstra, J., B. Komer, C. Eliasmith, D. Yamins, and D. D. Cox (2015), Hyperopt: a python library for model selection and hyperparameter optimization, *Computational Science & Discovery*, 8(1), 014,008.
- Bischofberger, W. R., and G. Pomberger (2012), *Prototyping-oriented software development: Concepts and tools*, Springer Science & Business Media.
- Bishop, C. M., et al. (1995), *Neural networks for pattern recognition*, Oxford University Press.
- Bonny, E., L. Almond, and P. Woolnough (2016), Adult missing persons: Can an investigative framework be generated using behavioural themes?, *Journal of Investigative Psychology and Offender Profiling*, 13(3), 296–312.
- Brookes, D., and D. Webster (1999), Child welfare in the united states: Policy, practice and innovations in service delivery, *International Journal of Social Welfare*, 8(4), 297–307.

- Burt, R. S. (1980), Models of network structure, *Annual review of sociology*, 6(1), 79–141.
- Carbonell, J. G., R. S. Michalski, and T. M. Mitchell (1983), An overview of machine learning, in *Machine Learning, Volume I*, pp. 3–23, Elsevier.
- Carter, J. V., J. Pan, S. N. Rai, and S. Galandiuk (2016), ROC-ing along: Evaluation and interpretation of receiver operating characteristic curves, *Surgery*, 159(6), 1638–1645.
- Chang, C.-C., and C.-J. Lin (2011), LIBSVM: a library for support vector machines, *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3), 27.
- Corrigan, O. (2018), An investigation into machine learning solutions involving time series across different problem domains, Ph.D. thesis, Dublin City University, School of Computing.
- Costa, E., A. Lorena, A. Carvalho, and A. Freitas (2007), A review of performance evaluation measures for hierarchical classifiers, in *Evaluation methods for machine learning II: Papers from the AAAI-2007 workshop*, pp. 1–6.
- Deacon, J. (2013), Model-view-controller (MVC) architecture, <http://www.jdl.co.uk/briefings/MVC.pdf>, 4, 1–6, (Online).
- Decoste, D., and B. Scholkopf (2002), Training invariant support vector machines, *Machine learning*, 46(1-3), 161–190.
- DeLong, E. R., D. M. DeLong, and D. L. Clarke-Pearson (1988), Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach., *Biometrics*, 44(3), 837–845.
- Derrough, J. (2013), *Instant Interactive Map Designs with Leaflet JavaScript Library How-to*, Packt Publishing Ltd.
- Dong, J.-x., A. Krzyzak, and C. Y. Suen (2005), Fast SVM training algorithm with decomposition on very large data sets, *IEEE transactions on pattern analysis and machine intelligence*, 27(4), 603–618.
- Duda, R. O., and P. E. Hart (1973), Pattern classification and scene analysis, *A Wiley-Interscience Publication, New York: Wiley, 1973*.
- Fan, R.-E., K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin (2008), Liblinear: A library for large linear classification, *Journal of machine learning research*, 9(Aug), 1871–1874.
- Ferguson, R., and C. Heilmann (2013), HTML and JavaScript, in *Beginning JavaScript with DOM Scripting and Ajax*, pp. 69–100, Springer.
- Furey, T. S., N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler (2000), Support vector machine classification and validation of cancer tissue samples using microarray expression data, *Bioinformatics*, 16(10), 906–914.

- Fyfe, N. R., O. Stevenson, and P. Woolnough (2015), Missing persons: the processes and challenges of police investigation, *Policing and Society*, 25(4), 409–425.
- Greenspan, J., and B. Bulger (2001), *MySQL/PHP database applications*, John Wiley & Sons, Inc.
- Guo, G., and S. Z. Li (2003), Content-based audio classification and retrieval by support vector machines, *IEEE transactions on Neural Networks*, 14(1), 209–215.
- Hastie, T., R. Tibshirani, and J. Friedman (2008), *The Elements of Statistical Learning*, Springer.
- Hellegouarch, S. (2007), *CherryPy essentials: Rapid Python web application development*, Packt Publishing Ltd.
- Henderson, M., C. Kiernan, P. Henderson, et al. (2000), Missing persons: incidence, issues and impacts, *Trends and Issues in Crime and Criminal Justice/Australian Institute of Criminology*, (144), 1.
- Hsu, C.-W., C.-C. Chang, C.-J. Lin, et al. (2003), A practical guide to support vector classification, *National Taiwan University, Department of Computer Science*, available at <https://www.csie.ntu.edu.tw/~cjlin>.
- Huang, H.-Y., and C.-J. Lin (2016), Linear and kernel classification: When to use which?, in *Proceedings of the 2016 SIAM International Conference on Data Mining*, pp. 216–224, SIAM.
- Joachims, T. (1998), Text categorization with support vector machines: Learning with many relevant features, in *European conference on machine learning*, pp. 137–142, Springer.
- Kecman, V. (2005), Support vector machines—an introduction, in *Support vector machines: theory and applications*, pp. 1–47, Springer.
- Kim, Y.-K., and K.-S. Na (2018), Application of machine learning classification for structural brain mri in mood disorders: Critical review from a clinical perspective, *Progress in Neuro-Psychopharmacology and Biological Psychiatry*, 80, 71–80.
- Kononenko, I., and M. Kukar (2007), *Machine Learning and Data Mining: Introduction to Principles and Algorithms*, Horwood Publishing.
- Larson, B., D. English, and P. Purington (2016), *Microsoft SQL Server 2016 Reporting Services*, McGraw-Hill Education.
- Lokhande, P., F. Aslam, N. Hawa, J. Munir, and M. Gulamgaus (2015), Efficient way of web development using python and flask, *International Journal of Advanced Research in Computer Science*.
- Lutz, M. (2013), *Learning Python: Powerful Object-Oriented Programming*, O’Reilly Media, Inc.

- Marr, B. (2016), A short history of machine learning - every manager should read, *Forbes*.
 URL: <https://www.forbes.com/sites/bernardmarr/2016/02/19/a-shorthistory-of-machine-learning-every-managershould-read>.
- Mercier, G., and M. Lennon (2003), Support vector machines for hyperspectral image classification with spectral-based kernels, in *Geoscience and Remote Sensing Symposium, 2003. IGARSS'03. Proceedings. 2003 IEEE International*, vol. 1, pp. 288–290, IEEE.
- Michalski, R. S., J. G. Carbonell, and T. M. Mitchell (2013), *Machine learning: An artificial intelligence approach*, Springer Science & Business Media.
- Muller, A. C., and S. Guido (2017), *Introduction to Machine Learning with Python*, O'Reilly Media, Inc.
- Murphy, K. P. (2012), *Machine Learning: A Probabilistic Perspective. Adaptive Computation and Machine Learning*, MIT Press.
- Nguyen, T. T., and G. Armitage (2008), A survey of techniques for internet traffic classification using machine learning, *IEEE Communications Surveys & Tutorials*, 10(4), 56–76.
- Ozkan, T. (2017), Predicting recidivism through machine learning, Ph.D. thesis, Department of Computer Science, University of Texas at Dallas.
- Pang, B., L. Lee, and S. Vaithyanathan (2002), Thumbs up?: sentiment classification using machine learning techniques, in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pp. 79–86, Association for Computational Linguistics.
- Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011), Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research*, 12, 2825–2830.
- Pfeifer, J. (2006), *Missing persons in Saskatchewan: Police policy and practice*, Law Foundation of Saskatchewan Chair in Police Studies, University of Regina, Department of Forensic and Social Psychology.
- Pramanik, M. I., R. Y. Lau, W. T. Yue, Y. Ye, and C. Li (2017), Big data analytics for security and criminal investigations, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 7(4), e1208.
- Qiang, W., and Z. Zhongli (2011), Reinforcement learning model, algorithms and its application, in *2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC)*, IEEE.
- Quinlan, J. R. (1987), Decision trees as probabilistic classifiers, in *Proceedings of the Fourth International Workshop on Machine Learning*, pp. 31–37, Elsevier.
- Redmond, M., and A. Baveja (2002), A data-driven software tool for enabling cooperative information sharing among police departments, *European Journal of Operational Research*, 141(3), 660–678.

- Reed, R. D., and R. J. Marks (1998), *Neural Smoothing: Supervised Learning in Feedforward Artificial Neural Networks*, Massachusetts Institute of Technology, The MIT Press, Cambridge, Massachusetts, London, England.
- Refaeilzadeh, P., L. Tang, and H. Liu (2009), Cross-validation, *Encyclopedia of Database Systems*, pp. 532–538, doi:10.1007/978-0-387-39940-9.
- Reitermanov, Z. (2010), Data splitting, in *WDS'10 Proceedings of Contributed Papers*, pp. 31–36.
- Ren, Y., F. Hu, and H. Miao (2016), The optimization of kernel function and its parameters for svm in well-logging, in *2016 13th International Conference on Service Systems and Service Management (ICSSSM)*, pp. 1–5, IEEE.
- Ronacher, A. (2008), Jinja2 documentation release 2.10, <http://www.jinja.pocoo.org/docs/2.10/>.
- Russell, J., and S. Macgill (2015), Demographics, policy, and foster care rates; a predictive analytics approach, *Children and Youth Services Review*, 58, 118–126.
- Schölkopf, B. (2001), The kernel trick for distances, in *Advances in neural information processing systems*, pp. 301–307.
- Scholkopf, B., and A. J. Smola (2002), *Learning with kernels: support vector machines, regularization, optimization, and beyond*, MIT Press.
- Shinde, S., and M. K. S. Patel (2014), CherryPy: Common programmer issues with solutions, *International Journal of Research*, 1(4), 874–879.
- Sledjeski, E. M., L. C. Dierker, R. Brigham, and E. Breslin (2008), The use of risk assessment to predict recurrent maltreatment: A classification and regression tree analysis (CART), *Prevention science*, 9(1), 28–37.
- Smits, G. F., and E. M. Jordaan (2002), Improved svm regression using mixtures of kernels, in *Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on*, vol. 3, pp. 2785–2790, IEEE.
- Sparrow, M. K. (1993), *Information Systems and the Development of Policing*, 16, US Department of Justice, Office of Justice Programs, National Institute of Justice.
- Steinwart, I., and A. Christmann (2008), *Support vector machines*, Springer Science & Business Media.
- Van Duyn, J. (1991), *Automated crime information systems*, TAB Professional and Reference Books.
- Vapnik, V., and C. Cortes (1995), Support-vector networks, *Machine learning*, 20(3), 273–297.

- Walters, G. D. (2019), Gang influence: Mediating the gang–delinquency relationship with proactive criminal thinking, *Criminal Justice and Behavior*, p. 0093854819831741.
- Weston, J., S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik (2001), Feature selection for svms, in *Advances in neural information processing systems*, pp. 668–674.
- Xu, X., C. Zhou, and Z. Wang (2009), Credit scoring algorithm based on link analysis ranking with support vector machine, *Expert Systems with Applications*, 36(2), 2625–2632.
- Yu, H., J. Yang, and J. Han (2003), Classifying large data sets using svms with hierarchical clusters, in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 306–315, ACM.
- Yu, H.-F., C.-J. Hsieh, K.-W. Chang, and C.-J. Lin (2012), Large linear classification when data cannot fit in memory, *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(4), 23.

APPENDIX A

MISSING YOUTHS GRAPHICAL USER INTERFACE

A.1 “Login” Page

The “Login” page is the main page for entering into the MY GUI. Figure A.1 represents a screenshot of the “Login” page. The MY GUI is implemented as a multiple-user system, and a person who wants to use the system must use his/her specific username and password. The “Forgot Password?” link provides an option to change a password for a user who does not remember his/her password. A screenshot of the “Forgot Password?” page is given in Figure A.2. A successful login lands the user on the MY homepage.



Figure A.1: Screenshot for “Login” page

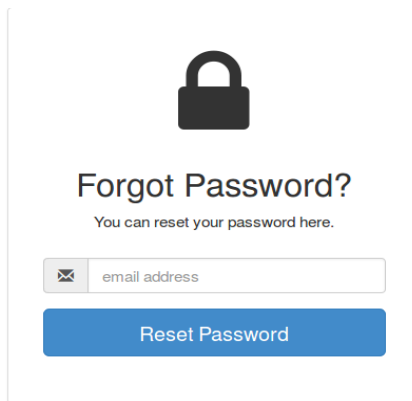


Figure A.2: Screenshot for “Forgot Password?” page

A.1.1 “Invalid Username/Password” Page

The “Invalid Username/Password” message is displayed to the user when the login permission is denied. A screenshot for the “Invalid Username/Password” page is given in Figure A.3.

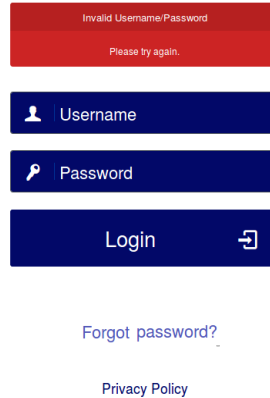


Figure A.3: Screenshot for “Invalid Username/Password” page

A.2 “Home” Page

The “Home” page is displayed as a start page of the MY GUI when a user logs into the system. At the top of the page, proceeding from left to right, you can see the main menu under the logo in the top left corner, a drop-down menu associated with the user name of the person logged in (in this case, “Admin”), the “data sets” menu, and finally a “Log Out” button in the top right corner. The main menu includes “Home”, “Missing Persons”, “Settings”, “Users”, “Email”, “data sets”, and “Analysis”. The main menu directs to various sub-parts, and each of them directly connects to new pages. The Saskatoon police logo on top of the menu links a user to the SPS webpage. A drop-down menu contains a sub menu called “Inbox” and enables an access to the email. There is a “Logout” option placed on the top right side of the MY GUI. A full screenshot is given in Figure A.4.

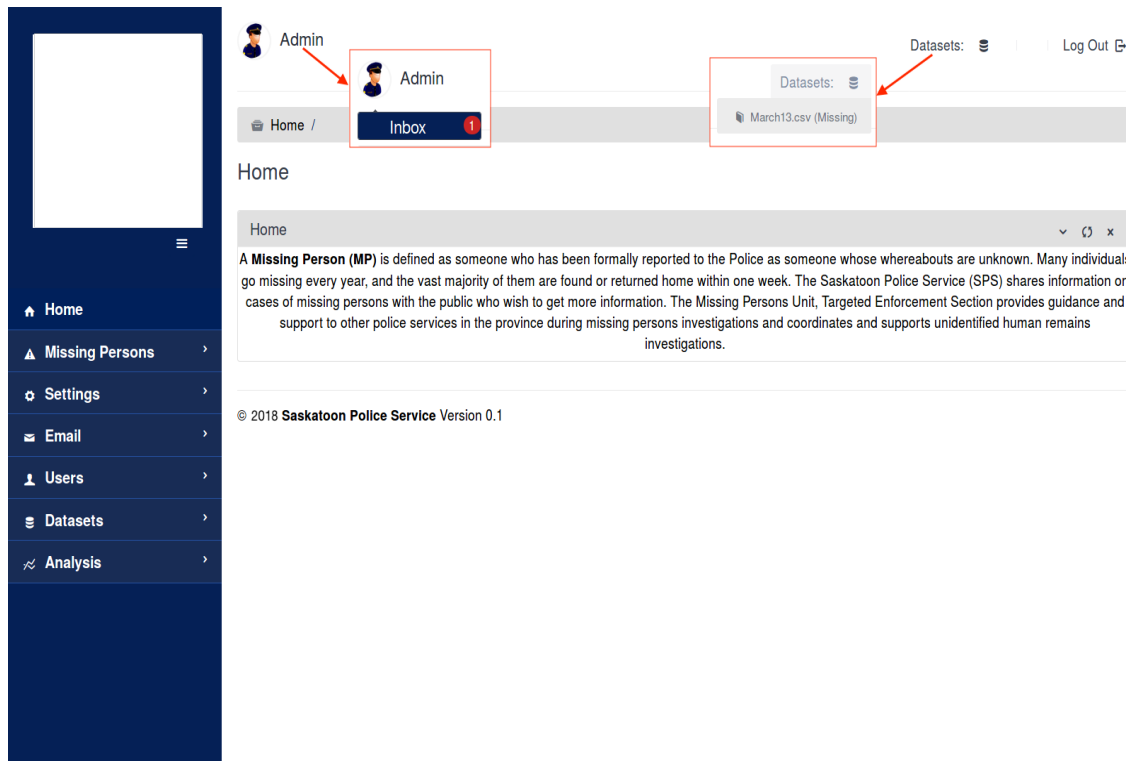


Figure A.4: Screenshot for “Home” page

A.3 “Missing Persons” Page

The “Missing Persons” page features a quick and reliable system for the police reporting system.

A.3.1 “Date Range” Page

The “Date Range” page provides a report based on a specific date range and with a particular name. Figure A.5 gives a screenshot for specifying a date range using a calendar interface. The calendar interface programmed using a Javascript calendar library. Figure A.6 gives a screenshot of the “Date Range” page in the MPY GUI.

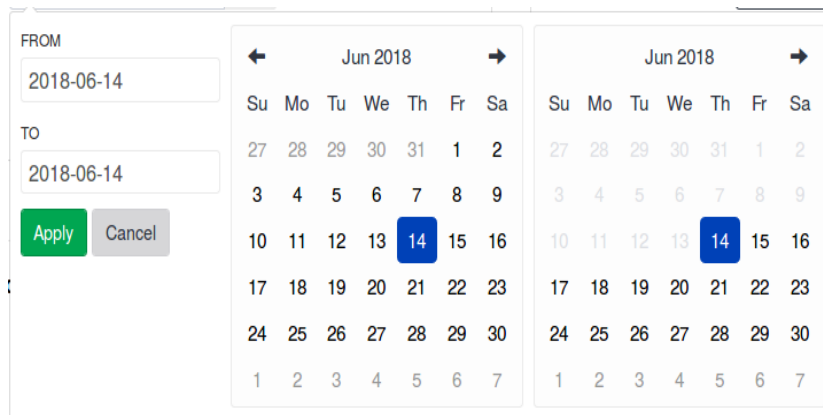


Figure A.5: Screenshot for specifying a date range using a calendar interface

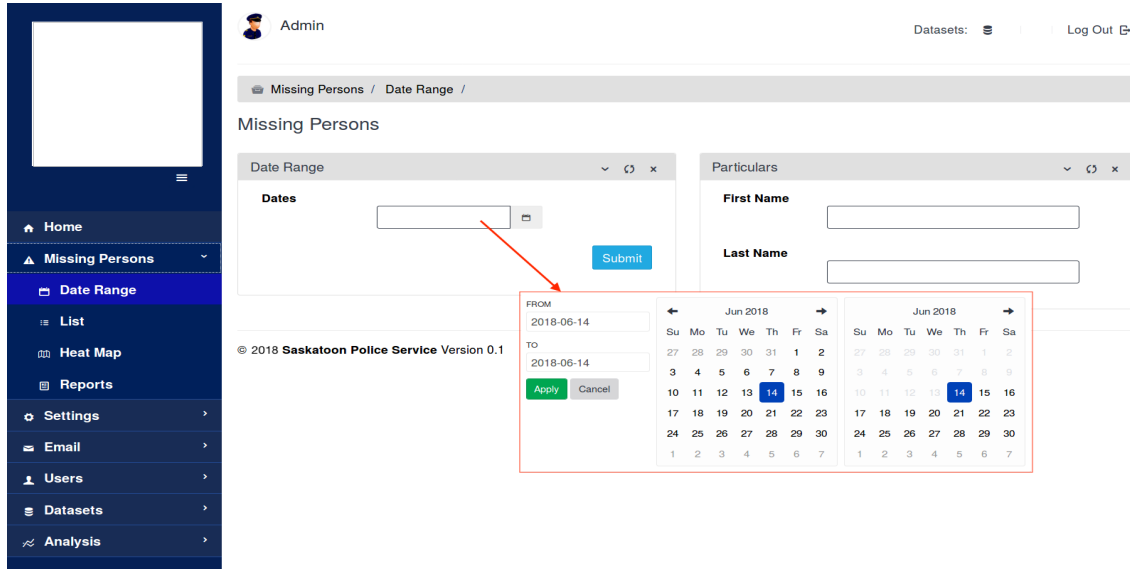


Figure A.6: Screenshot for “Date Range” page

A.3.2 “List” Page

The “List” page creates a report file that can be filtered by attributes such as gender, age, role, or location. An example screenshot of the “List” page is given in Figure A.7. In this page, the filtering section is categorized by the type, and a “Location” filtering feature creates a report file using the GPS coordinates of Saskatoon neighbourhoods.

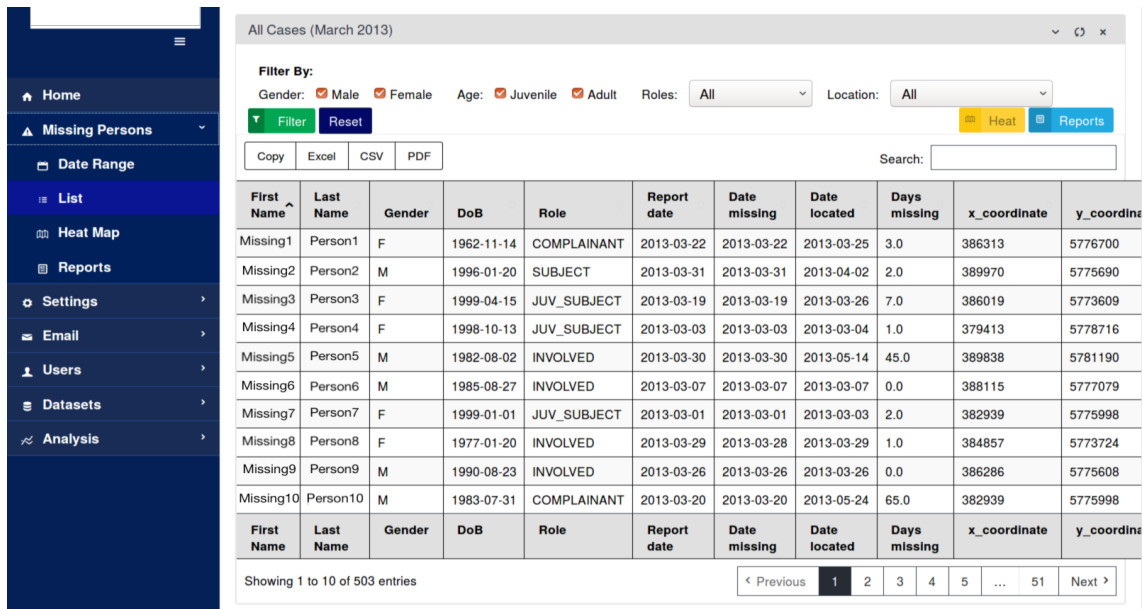


Figure A.7: Screenshot for “List” page

A.3.3 “Heat Map” Page

“Heat Map” can display data on a map scale by the x and y coordinates from the go_data table in the MY database. For example, the hot spots on the heat map could represent group homes or locations frequented by gang members or drug dealers. A screenshot of the “Heat Map” page is given in Figure A.8. For security reasons, the “Heat Map” section is implemented as an offline map in this version by using the Leaflet Heatmap Layer Plugin. The Leaflet is an open-source JavaScript library for interactive web maps ([Derrough, 2013](#)).

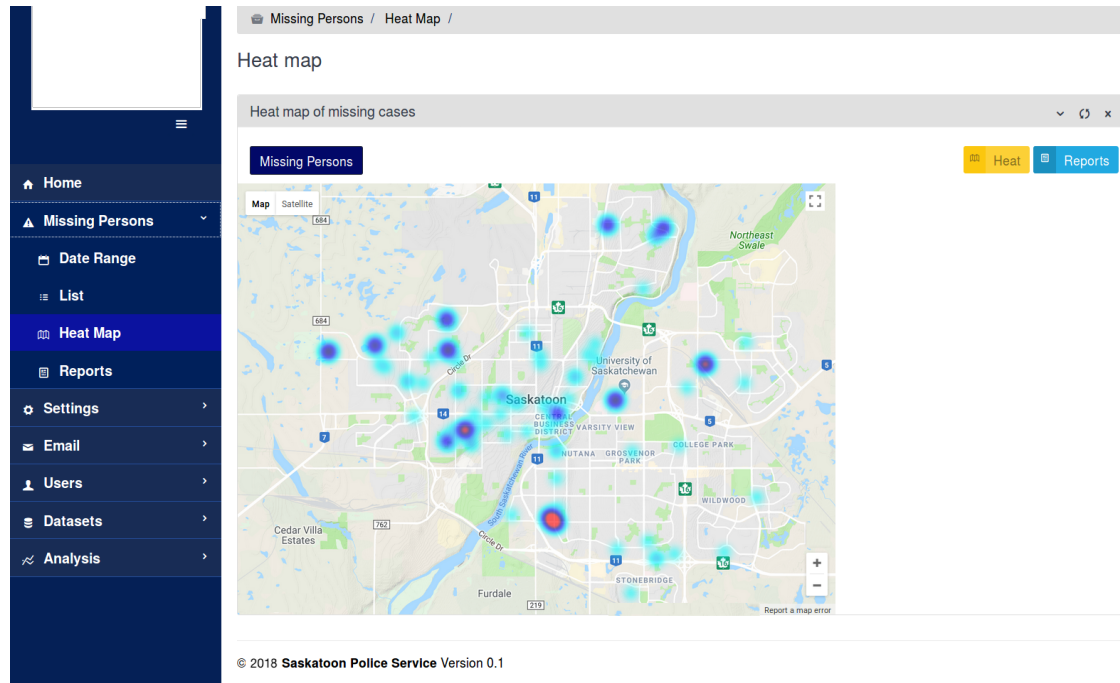


Figure A.8: Screenshot for “Heat Map” page

A.3.4 “Reports” Page

The “Reports” page can provide a report file in various formats from the MY database in an easy and a secure fashion. The first report format for the MY data is the Table Report for each particular. The “Reports” page creates a Table Report for each particular by typing a particular name in a search box. A screen shot of the Table report is given in Figure A.9.

Reports of Missing Persons

Reports (March 2013)

Missing Persons Heat Reports

Copy Excel CSV PDF Search:

Day	Name	YM	YF	AM	AF	OCC
1	Missing2 Person2		1			1
1	Missing8 Person8		2			1
1	Missing14 Person14		3			1
1	Missing32 Person32		4			1
1	Missing21 Person21		5			1
1	Missing10 Person10		6			1
2	Missing1 Person1		7			1
2	Missing5 Person5		8			1
2	Missing24 Person24		9			1
2	Missing46 Person46	1				1

Showing 1 to 10 of 192 entries

< Previous 1 2 3 4 5 ... 20 Next >

Figure A.9: Screenshot for Table Report

The second report format is the Chart Reports. A screenshot of the Chart Reports is given in Figure A.10. In this example, the graphs represent gender (percentage female vs. male), age (number of cases for a given age), and day distributions (number cases per day) for the MY database. The x and y axes in the bar graph are labelled, and the history report for each bar is illustrated on the Table Report on top of the page.

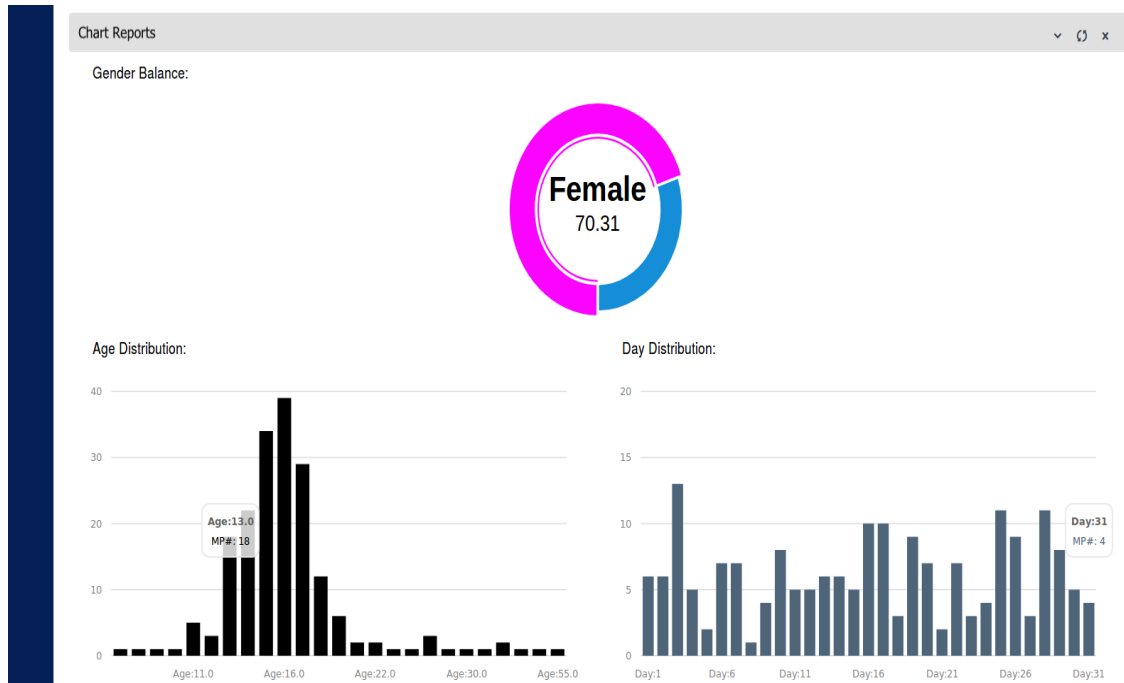


Figure A.10: Screenshot for Chart Reports

The third report format is the PDF file. This format generates a brief report suitable for bi-weekly or monthly reporting. A sample screenshot is given in Figure A.11.

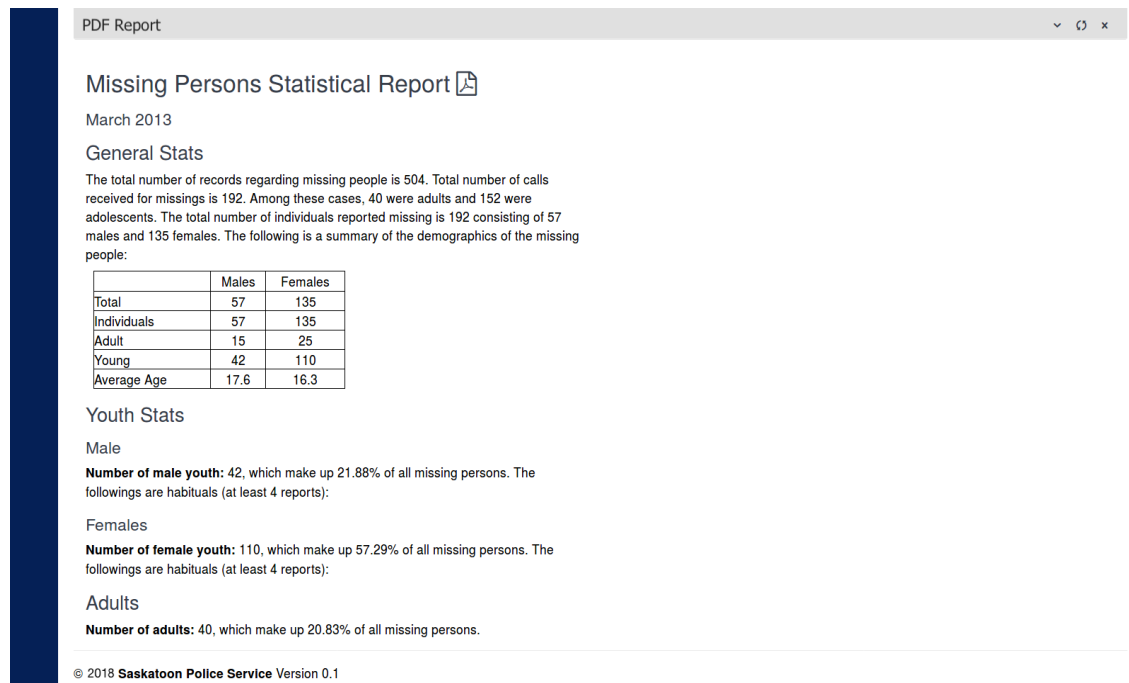


Figure A.11: Screenshot for PDF Report

A.4 “Settings” Page

The “Settings” correspond to different modules related to user access in the MY GUI.

A.4.1 “Modules” Page

The “Modules” page generates a list of settings according to the level of user access. A high level of security of the MY data can be maintained through appropriate management of the user permissions. A screenshot is given in Figure A.12.

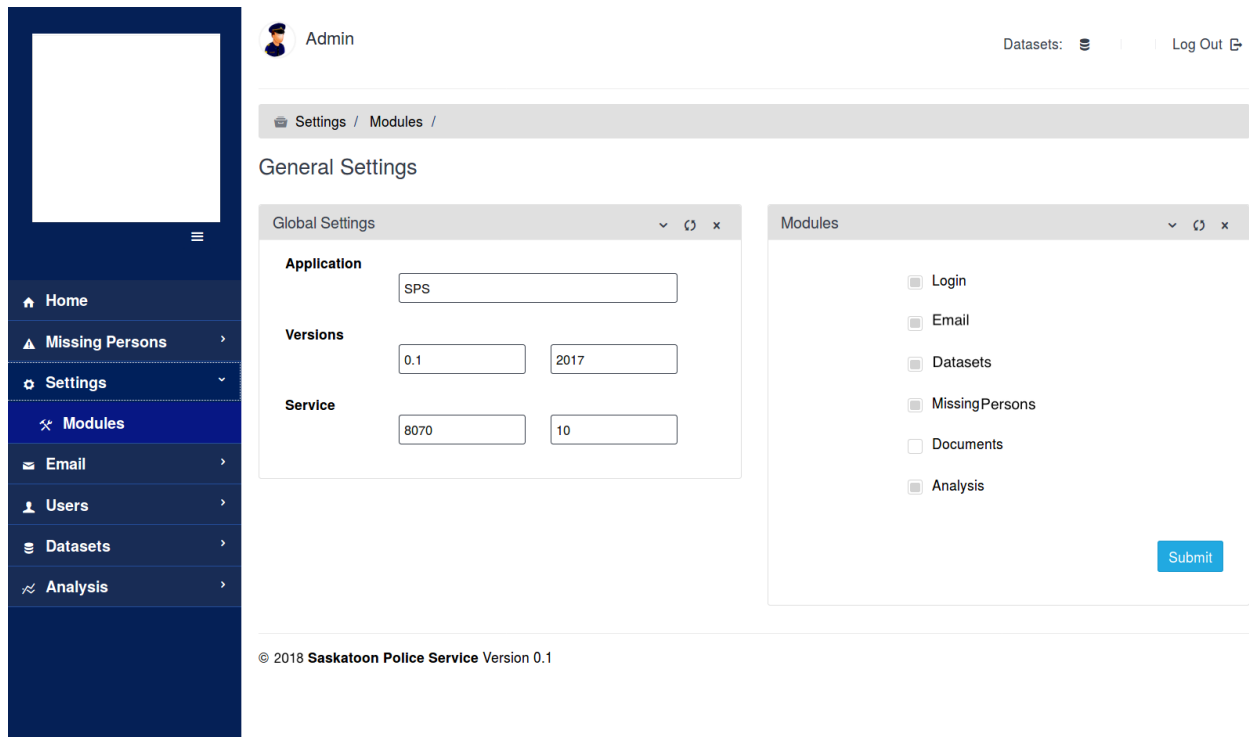


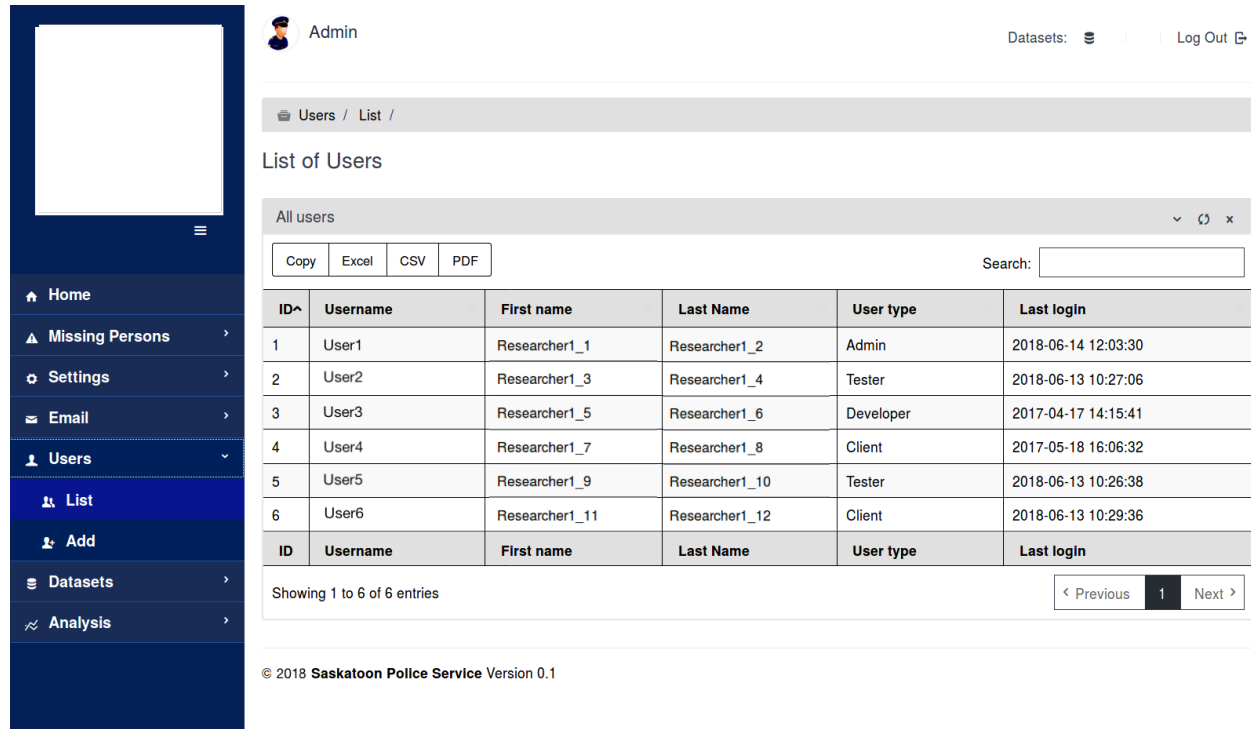
Figure A.12: Screenshot for “Modules” page

A.5 “Users” Page

The “Users” page creates new user accounts and assigns permissions in the MY GUI.

A.5.0.1 “List of Users” Page

The “List of Users” page describes available user data in the MY GUI. A screenshot of the “List of Users” page is given in Figure A.13.



The screenshot displays the "List of Users" page. At the top, the user "Admin" is logged in, with a "Log Out" button. The breadcrumb trail is "Users / List /". The page title is "List of Users". Below the title, there are options to "All users" and a search bar. There are also buttons for "Copy", "Excel", "CSV", and "PDF". The main content is a table with 6 rows of user data. The table has columns for ID, Username, First name, Last Name, User type, and Last login. Below the table, it says "Showing 1 to 6 of 6 entries" and has "Previous" and "Next" navigation buttons. The footer shows "© 2018 Saskatoon Police Service Version 0.1".

ID	Username	First name	Last Name	User type	Last login
1	User1	Researcher1_1	Researcher1_2	Admin	2018-06-14 12:03:30
2	User2	Researcher1_3	Researcher1_4	Tester	2018-06-13 10:27:06
3	User3	Researcher1_5	Researcher1_6	Developer	2017-04-17 14:15:41
4	User4	Researcher1_7	Researcher1_8	Client	2017-05-18 16:06:32
5	User5	Researcher1_9	Researcher1_10	Tester	2018-06-13 10:26:38
6	User6	Researcher1_11	Researcher1_12	Client	2018-06-13 10:29:36

Figure A.13: Screenshot for “List of Users” page

A.5.1 “Add User” Page

A system administrator can create a valid username and password for a new user through the “Add User” page. A screenshot is given in Figure A.14.

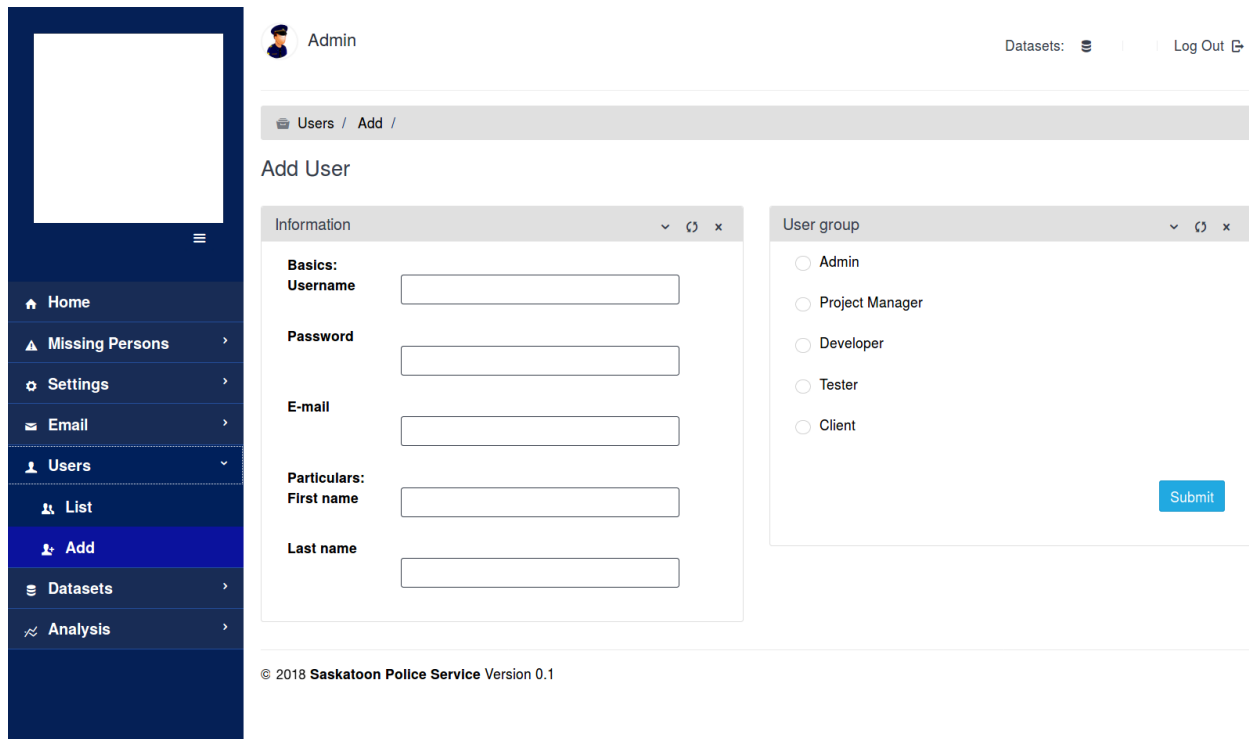


Figure A.14: Screenshot for “Add User” page

A.6 “Email” Page

The “Email” page represents as the user message profile and is divided into “Inbox” and “Compose” pages. This page is not a general email page. The main purpose of the “Email” page is to provide a secure environment to communicate and share data and information between police and other organizations.

A.6.1 “Inbox” Page

The “Inbox” page represents a folder to store received messages in the MY system. Figure A.15 gives a screenshot of the “Inbox” page.

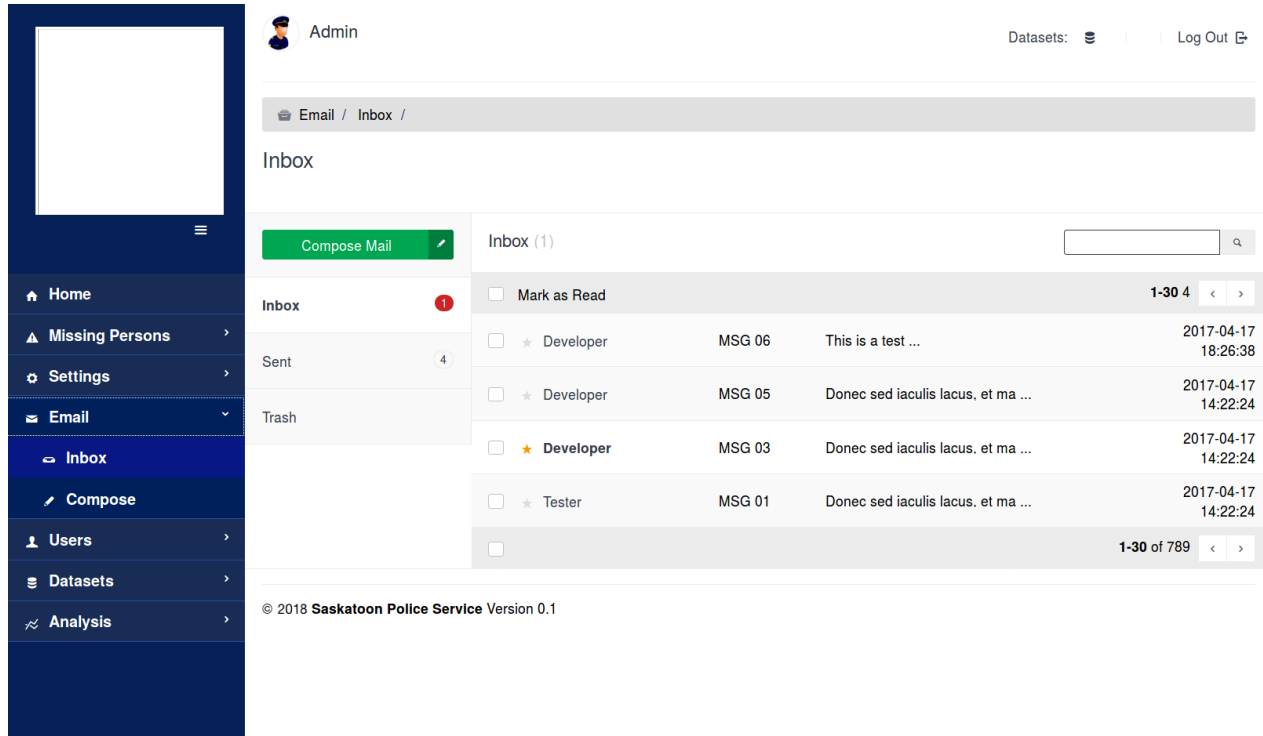


Figure A.15: Screenshot for “Inbox” page

A.6.2 “Compose” Page

The “Compose” page provides an option for users to create a new message for others within the MY GUI. A screenshot is given in Figure A.16.

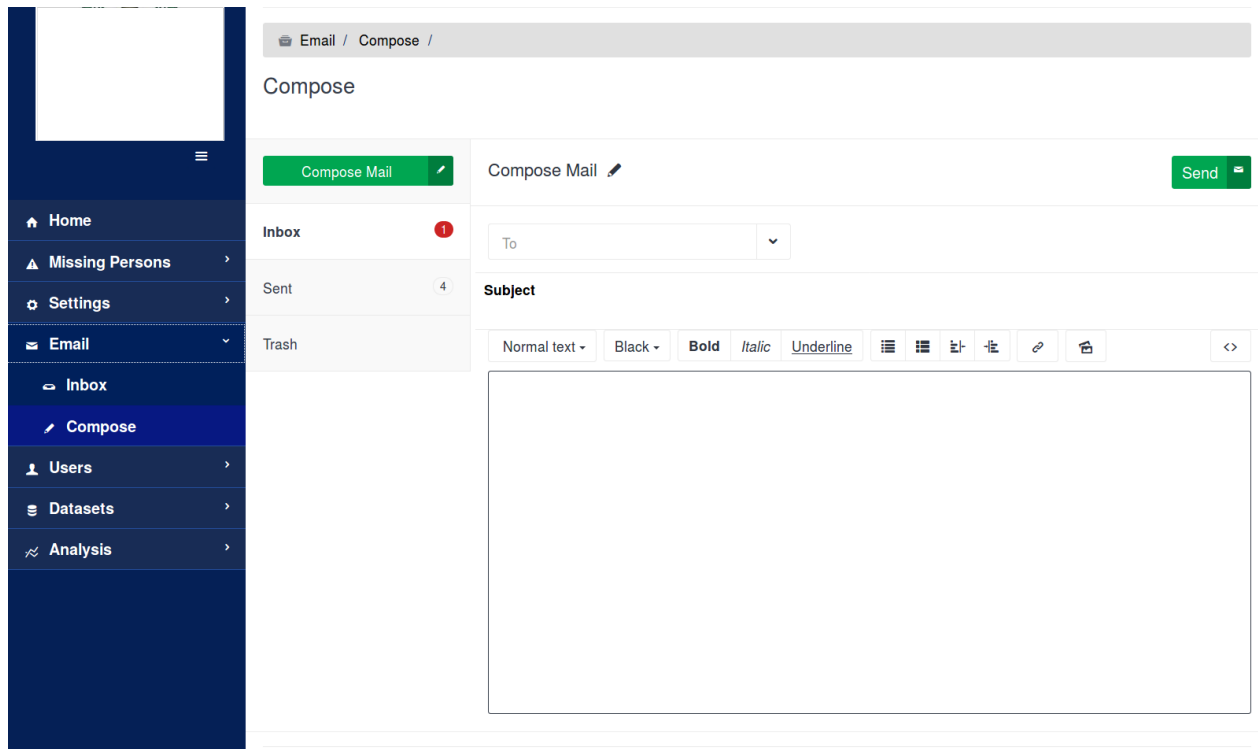


Figure A.16: Screenshot for “Compose” page

A.7 “Data sets” Page

The “data sets” page provides the facility to save and re-use data sets (or *snapshots*) of the MY database using the GUI to facilitate quick and easy access to saved data and report settings from previous sessions. Multiple snapshots may be open simultaneously. The snapshots allow the user to save only relevant data from one session to the next. Data are not updated, however, when new data become available.

The snapshots are local to the user. No other users have access to the snapshots; however, they can be shared using the secure messaging system in the GUI. The MY database from which the snapshots are created is not modifiable by any user regardless of the permissions associated with their account. Issues surrounding modifying the MY database are addressed elsewhere.

A.7.1 “List of data sets” Page

The “List of data sets” page describes available data sets (or *snapshots*) in the MY GUI. A screenshot of the “List of data sets” page is given in Figure A.17. The “List of data sets” page is implemented to locally save customized versions of data sets (snapshots) that can be worked from later.

The screenshot displays the 'List of Datasets' page. At the top, the user is identified as 'Admin' and there are options for 'Datasets' and 'Log Out'. The breadcrumb trail shows 'Datasets / List /'. The main content area is titled 'List of Datasets' and contains a table of dataset buffers. The table has columns for 'Initiation', 'Name', 'Default', 'Module', 'Description', and 'Action'. A single entry is shown: '2017-06-20 23:59:40.167188' for 'March13.csv' in the 'Missing' module, with a description of 'March 2013'. The 'Action' column for this entry includes 'Edit', 'Default', 'Delete', and 'Select' buttons. Below the table, it indicates 'Showing 1 to 1 of 1 entries' and provides navigation for 'Previous' and 'Next'. The footer shows '© 2018 Saskatoon Police Service Version 0.1'.

Initiation	Name	Default	Module	Description	Action
2017-06-20 23:59:40.167188	March13.csv		Missing	March 2013	Edit Default Delete Select

Figure A.17: Screenshot for “List of data sets” page

A.7.2 “Add data set” Page

A user in the MY GUI can add a saved data set (snapshot) from previous work to the system and re-use previously stored snapshots through the “Add data set” page. A screenshot is given in Figure A.18.

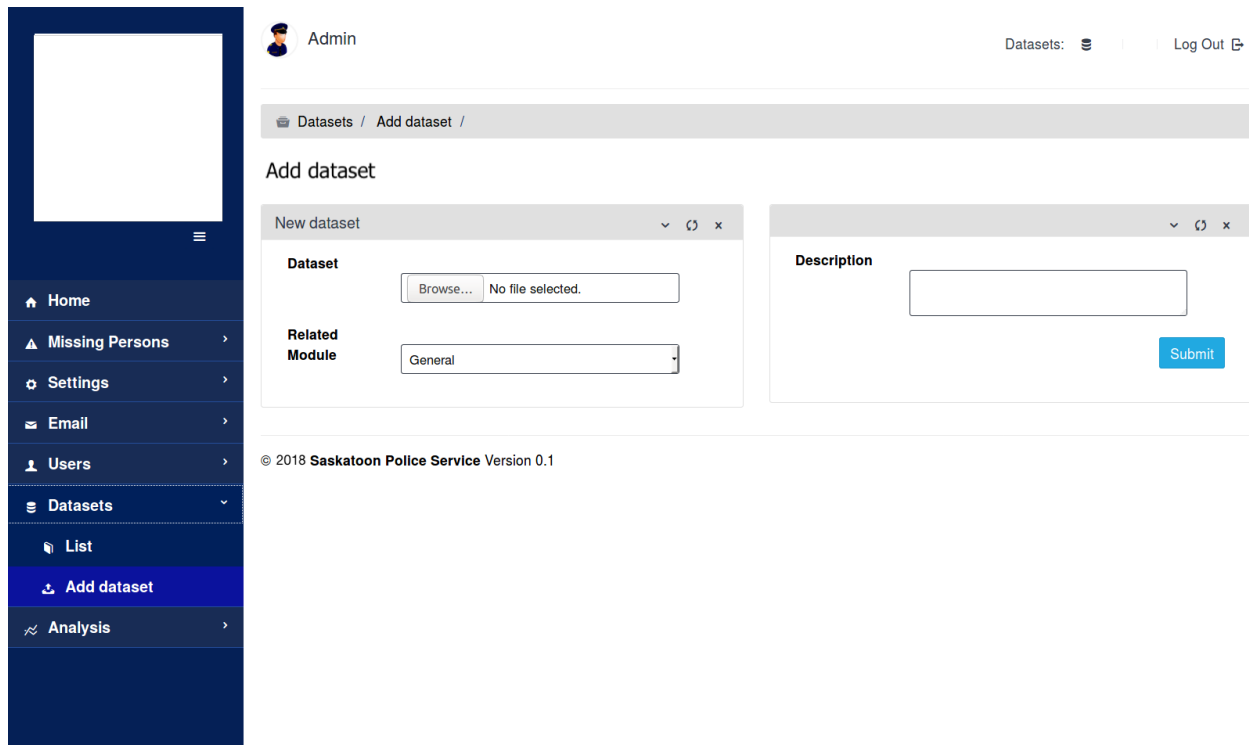


Figure A.18: Screenshot for “Add data set” page

A.8 Data Analysis

The “Data Analysis” page provides an easy and reliable data analysis system for the end users of the MPP. For both classification features *missing_again* and *gang_involvement*, the machine learning methods Decision Trees, Selective Decision Trees, and SVMs are implemented in a separate tab of the MY GUI.

A.8.1 Decision Trees

The “Decision Trees” page provides a data analysis report based on the Decision Trees model that is implemented for the MY database. From this page, the end users can analyze the accuracy results for both classification features for the MY database. Figure A.19 gives a screenshot of the “Decision Trees” page.

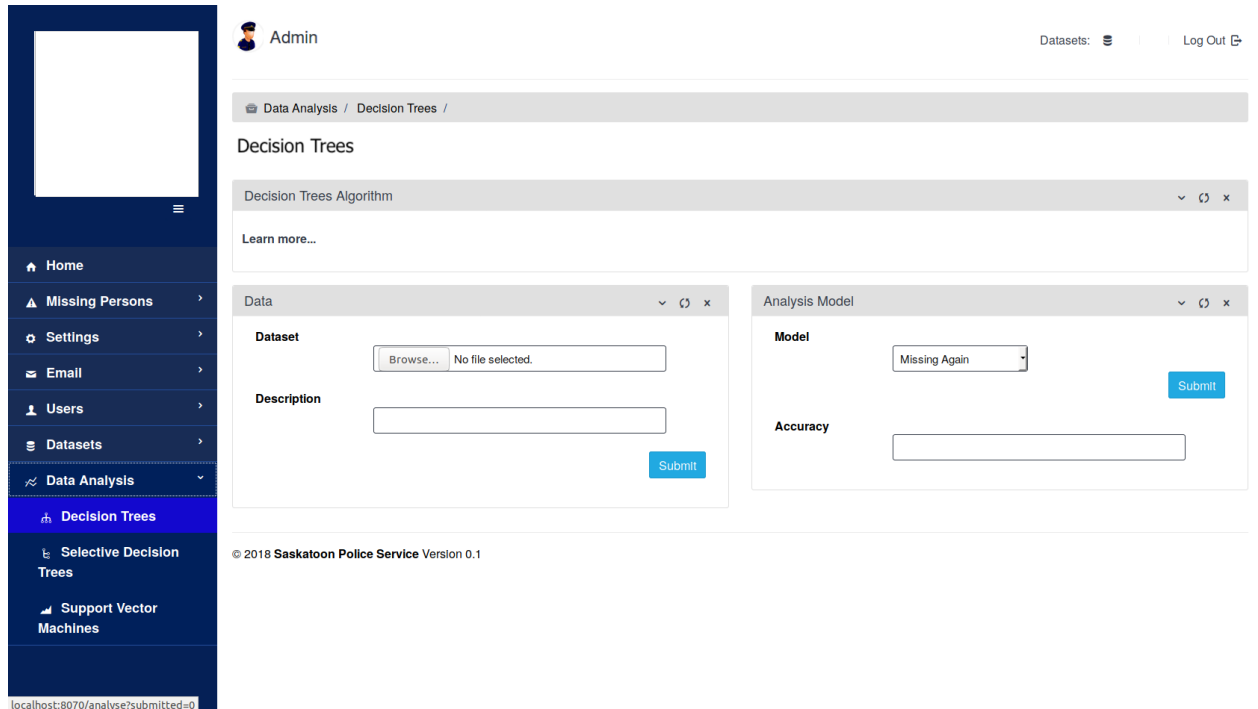


Figure A.19: Screenshot for “Decision Trees” page

A.8.2 Selective Decision Trees

The “Selective Decision Trees” page represents the feature selection method based on Decision Trees. Based on each classification feature, the accuracy results can provide for the MY database. Figure A.20 gives a screenshot of the “Selective Decision Trees” page.

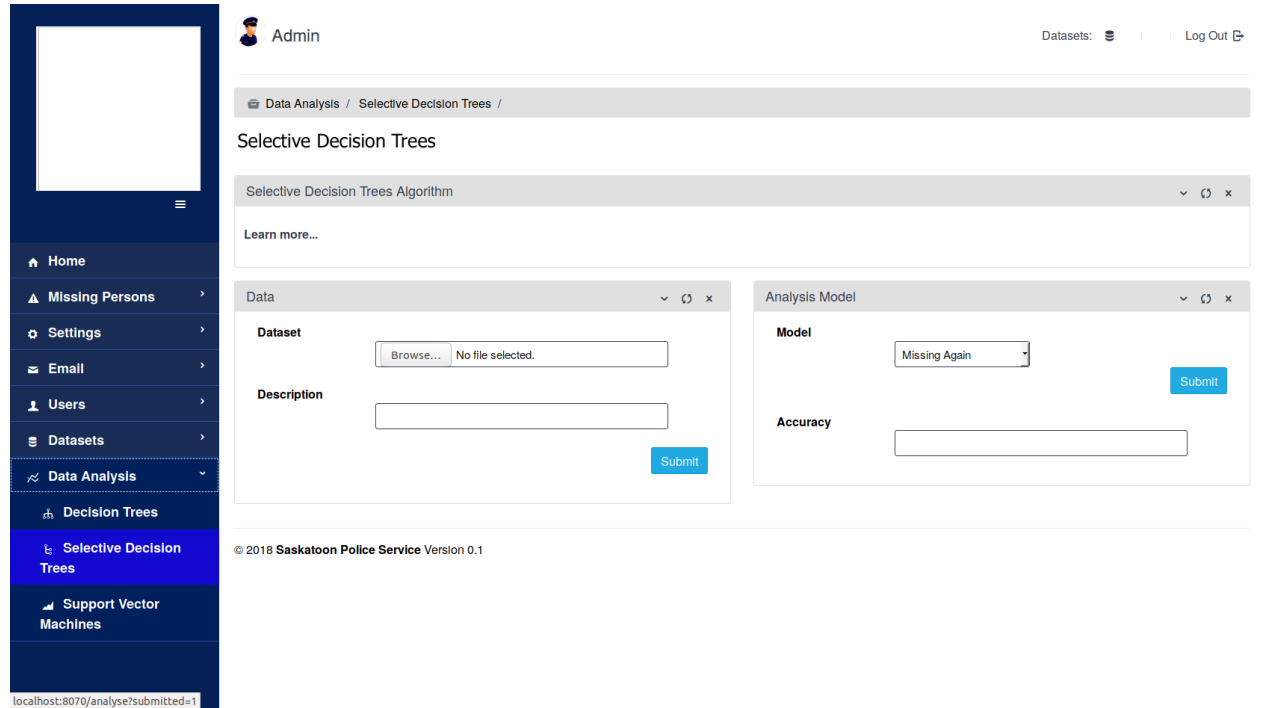


Figure A.20: Screenshot for “Selective Decision Trees” page

A.8.3 Support Vector Machines

The “Support Vector Machines” page provides the accuracy results for both classification features for the MY database. Figure A.21 gives a screenshot of the “Support Vector Machines” page.

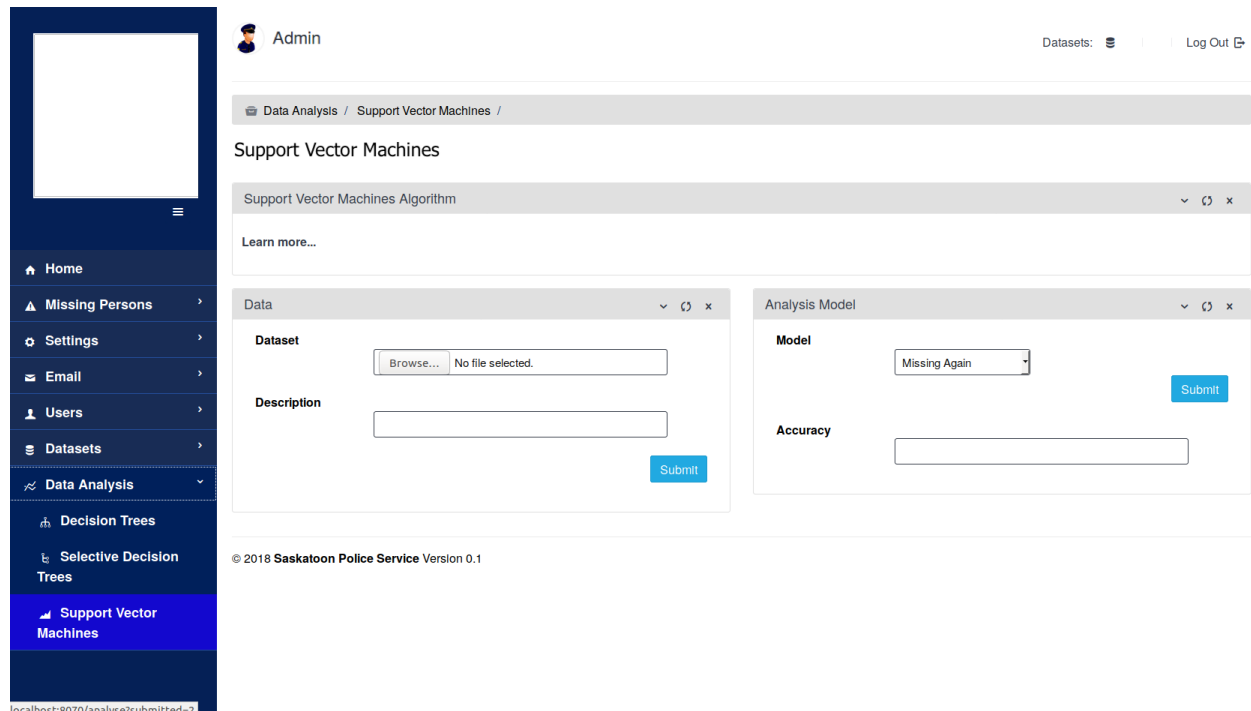


Figure A.21: Screenshot for “Support Vector Machines” page