# A Decision Support System to Predict Highway Accident Alerts in Sri Lanka

K. A. D. S. H. Rodrigo
179476A

Faculty of Information Technology
University of Moratuwa
2020

# A Decision Support System to Predict Highway Accident Alerts in Sri Lanka

K. A. D. S. H. Rodrigo
179476A

Dissertation submitted to the Faculty of Information Technology, University of Moratuwa, Sri Lanka for the fulfillment of the requirements of Degree of Master of Science in Information Technology.
2020

## Declaration

I declare that this is my own work and has not been submitted in any form for another degree or diploma at any university or other institution of tertiary education. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

Name of Student

K. A. D. S. H. Rodrigo

Signature of Student

……....

Date: ……...

Supervised By

Name of Supervisor

S. C. Premarathne

Signature of Supervisor

…….

Date: ……

# Acknowledgement

Firstly, I would like to express my sincere gratitude to my supervisor Mr. Saminda Premarathne for the continuous support of my dissertation study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this dissertation. I could not have imagined having a better advisor and mentor for my research study.

Besides my supervisor, I would like to thank Dr. Mohamed Firdhous who taught Research Methodology and Literature Review and Thesis Writing modules which helped me to widen my research from various perspectives.

I would not forget to remember the Sri Lanka Police for their timely support by providing all the accident records that have been required for the success of this research.

In addition, I would like to thank all the staff of Paragon Software Lanka for their insightful comments and encouragement which helped me lot to enhance my knowledge from various perspectives.

Furthermore, I would also like to thank all the batch mates of the M.Sc. in IT degree program who gave their valuable feedbacks to improve the results of the research, my family for the support they provided me throughout my entire life and in particular.

# Abstract

Road traffic and rapidly increasing road accidents become a vast problem not only for Sri Lankans but for all human beings who are living in this planet. According to the Sri Lanka police highway reports, there have been a number of deaths recorded due to fatal accidents in highways and also affected significant amount of people for non-fatal accidents as well. Several key factors vastly contributed directly to cause an accident, such as environmental factors, human factors and road condition factors., etc. In this research, machine learning techniques and methods have been applied to the southern highway accident records retrieved by Police Highway Control Center for the period of 2015 to June 2019 in order to establish a model which enables to forecast reason for the accidents that will be occurring in the future. The python scikit-learn package has been used on top of the anaconda framework to discover hidden patterns of data with the help of decision tree, support vector machine and logistic regression powered by a one-vs-rest classifier. The two well-known ensembles; random forest and gradient boost classifiers have also performed in this dataset in order to enhance the accuracy. The performance of each classifier has compared critically based on their results. The results obtained by performing various experiments show that the ensemble's work well when compare to other classifiers.

# Table of Content

# List of Tables

# List of Figures

# List of Abbreviations

SLP - Sri Lanka Police

USA - United States of America

CART - Classification and Regression Tree

ID3 - Iterative Dichotomiser

UAE - United Arab Emirates

ANN - Artificial Neural Network

SVM - Support Vector Machine

LR - Logistic Regression

OVR - One-Vs-Rest

OVO - One-Vs-One

OVA - One-Vs-All

GPS - Global Positioning System

SMS - Short Message Service

IOS - iPhone Operating System

LHS - Left Hand Side

RHS - Right Hand Side

KNN - K-Nearest Neighbor

PCA - Principal Component Analysis

DT - Decision Tree

ROS - Random Over Sampler

GS - Grid Search

RS - Randomize Search

RF - Random Forest

GB - Gradient Boost

API - Application Program Interface

AWS - Amazon Web Services

UI - User Interface

UX - User Experience

# List of Appendices

# CHAPTER 1

# Introduction

## 1.1 Prolegomena

Road traffic and rapidly increasing road accidents become a vast problem nowadays in Sri Lanka. It is recorded that the road accident occurs every ten minutes in Sri Lanka due to a high number of vehicles and thus creating the road traffics [1]. Recent research has been found that 18,980 road accidents occurred from January to June in 2017 and in which 1,473 were fatal, causing 1,547 deaths of them [1]. There are growing concerns regarding the social and economic implications of the constantly increasing number of traffic accidents and fatalities in particular. One of the major goals in all countries is to develop a safe and sustainable highway system within the country to considering economic growth but, again there is a number of accidents were recorded in expressway as well. However, 956 numbers of accidents were recorded from 2011 to 2013 and nine accidents were identified as fatal [2].

The usage of expressways going to be increased day by day due to the heavy traffic in normal routes and busy schedules of people needs to save their time not as in the past, always find an easy way to fulfill their tasks. Because of that number of users in expressway gradually increasing, accident rate also increasing rapidly. This can be further proven by the past statistics released by the Expressway Police Control Stations located in Galanigama, Kurudugaha Hathakma and Pinnaduwa in Sri Lanka. Therefore, all the relevant parties have to take safety precautions and measures to prevent highway accidents, thus can be minimized damage as well. This is going to be a need nowadays.

The above numbers and statistics clearly show that how accidents became a major threat not only for the death of human beings, but resulting in enormous losses with respect to economic and social levels. This global challenge requires significant attention to mitigate the rate of accidents to a reasonable rate. The historical data related to previous accidents lead researchers to discover important factors which directly affected in such accidents.

## 1.2 Problem Statement

The usage of expressways going to be increased day by day due to the heavy traffic in normal routes and busy schedules of people needs to save their time not as in the past, always find an easy way to fulfill their tasks. Because of that number of users in expressway gradually increasing, with that accident rate also gradually increasing. This can be further proven by the past statistics released by the Expressway Police Control Stations located in Galanigama, Kurudugaha Hathakma and Pinnaduwa in Sri Lanka. Therefore, all the relevant parties have to take safety precautions and measures to prevent highway accidents, thus can be minimized damage as well. This is going to be a need nowadays.

## 1.3 Aim and Objectives

This research aims to successfully forecast the reason which responsible to occur an accident in southern expressway based on statistics derived from Sri Lanka Police (SLP) and proposed a real-time solution in order to mitigate them.

Based on the research aim, this research identified following as objectives.

1. Gather the accident records on the southern expressway in Sri Lanka from the expressway police control stations located in Galanigama, Kurudugaha Hathakma and Pinnaduwa for the period of January 2015 to June 2019.
2. Analyze the data which has been collected in order to identify accident-prone locations, possible causes of accidents such as date, time, type of vehicle, driver age, driver sex, driving side, weather, road surface condition and available vision.
3. Application of data mining and machine learning techniques to forecast the reasons for the cause of accidents.
4. Introducing a mobile application which alerts the users about accident-prone locations and reasons beforehand while driving on the southern expressway in Sri Lanka.

## 1.4 Scope of Research

This research aims to mitigate the future accident occurrence in the southern highway in Sri Lanka with the help of data mining techniques and mobile technologies. In addition, the study also expects to address the limitation found especially identifying most related accident factors, modeling classifiers and complete the voice alert while providing a driver to enough time to react the danger as well.

## 1.5 Proposed Solution

This research expects to proposed real-time mobile based application in order to predict the cause of accidents that have been taken place in the southern highway in Sri Lanka. The previous accident records nearly five years have been learning thoroughly with the help of big data and machine learning techniques in order to build the model properly. The learned model is responsible to predict the result based on the features sent by the client and the result will be delivered to the mobile application in real-time.

## 1.6 Structure of the Thesis

The complete structure of this thesis as follows. The first chapter provides an initial introduction to this study by explaining research aim and objectives, scope of the research and finally the problem statement which this research tries to overcome. The related works that have been done by the chosen fields has critically analyzed and summarized in the chapter two. The third chapter describes how the environment has been set up and also the tools and packages in order to function the various operations coming up in chapter four. The fourth chapter demonstrates the complete research methods, including machine learning techniques and methods which use to experiment with the various studies on top of the derived dataset. In addition, the same chapter provides information on mobile application development which will be started as soon as the machine learning exercises completes. The fifth chapter critically analyses the result obtained in the fourth chapter by comparing the various machine learning schemes that have been functioned in order to learn the model. The final chapter will draw conclusions based on the results in the above chapter and listed the areas which has to be concerned in the future.

# CHAPTER 2

# Literature Review

## 2.1 Introduction

This chapter critically demonstrates what sort of data mining and machine learning experiments that have already been done in order to forecast highway accidents. In addition, this chapter also reviews how the mobile applications have been used to alert the driver in real-time as well. Moreover, the gaps and limitations within the existing experiments have identified based on the research problem and discussed in detail.

## 2.2 Application of Data Mining and Mobile Technologies on Accident Data

Chinthanie [2] critically analyzed the recorded accidents data in between Kottawa to Pinnaduwa interchange for the period of November 2011 to October 2013 to identify the accident-prone locations and the reasons for the cause of accidents. The results generated were based only on road environment factors such as road condition, road geometry, surface contains water, etc. The study discovered nine accident-prone locations from Kottawa to Pinnaduwa interchange due to rainy weather.

Li, Shrestha and Hu [3] investigated traffic data carefully to find out the significant features which directly affected for the cause of fatal accidents happened in highways. The dataset used in this study contains all fatal accidents recorded through the year of 2007 in National Highway in California, USA. Several features such as collision manner, weather, light condition, drunk driver and surface condition were critically analyzed in order to discover the occurrence of fatal accident rate. An associate rule mining was introduced by applying the apriori algorithm on the derived dataset in order to find similar patterns. In addition, classifiers have been developed with the help of a Naive Bayes classifier and K-means clustering algorithm was used to form the clusters as well. The outcome of research clearly indicates that the human factors were strongly affected for the cause of a fatal accident rate. However, the findings also suggested that the influence of environmental factors for causing fatal accidents were fewer when compared to other.

Several decision trees (CART, J48 and ID3) and Naive Bayes classifiers have been

employed by Bahiru, Singh and Tessfaw [4] in order to forecast the severity of an accident. The application of classifiers has been done on selected accident factors, such as the time of the accident, victim type, vehicle type, road surface condition, weather condition, lighting condition, victim age and sex, seat belt on or off, speed limit and accident area as well. The findings indicated that the J48 decision tree performed effectively on derived dataset than the other classifiers that have been used in this research.

The models established with the help of several data mining techniques by Taamneh, Alkheder and Taamneh [5] to predicting injury severity of future accidents in Abu Dhabi, UAE. Several selected features such as driver-related factors, road-related factors and accident-related factors have been taken into account when modeling classifiers. The classifiers have been built with the help of several data mining algorithms, including Decision Tree (J48), PART, Multilayer Perceptron and Naive Bayes. The results demonstrated that the J48, PART and Multilayer Perceptron performed similarly while Naive Bayes produced lower accuracy. However, the research identified that the factors like age, gender, nationality, casualty status, accident year and collision type have a significant relation when predicting fatal severity of an accident.

Edirisinghe and Edirisinghe [6] conducted research recently in order to analyze accident frequency that has been occurred in between Kottawa and Godagama in southern expressway in Sri Lanka. The outcome of the study showed that the accidents were higher during holidays. The study also identified that the accident occurrence, relatively higher in right side when travelling from Welipanna to Kurudugaha and Pinnaduwa to Imaduwa as well. However, the accident frequency was higher again when traveling from Baddegama to Pinnaduwa in left-hand side as well. When considering both sides, the results recorded that the accident rate was extremely high in between Welipenna to Kurudugaha.

The study that has been done by [7] discovered that the factors influenced by the cause of accidents at toll road in Indonesia. The two popular data mining techniques such as Artificial Neural Networks (ANN) and Support Vector Machine (SVM) have been adopted to build models on the derived dataset. The performance found to be superior

when identifying factors that cause accidents when compared to classical techniques like Logistic Regression (LR).

The recent research that has been conducted by [8] discovered that the One-Vs-Rest (OVR) multiclass classifier functioned well for various imbalanced multi-class problems. In addition, the author also suggested with the help of several citations that One-Vs-One (OVO) technique successfully solves multi-class classification problem rather than the available versions.

Jahan, Hossen and Patwary [9] proposed solution to track the bus location using two available techniques; Global Positioning System (GPS) based location and Short Message Service (SMS) service. The proposed mobile application has to be installed on client's smartphone in order to get the tracking service. The implementation based on a client-server architecture where server responsible to extract the exact location of the bus and send it to the user via SMS. The proposed system performed relatively well when compared to existing tracking systems.

Mane et al. [10] conducted research recently to alert the road users when reaching danger zone. Data mining techniques such as clustering, classification and association rule were employed to forecast the accident-prone areas on highways in India. In addition, an Android mobile application was proposed to provide the notification and voice alert to the users who were driving on highways. Firebase, which is Backend-as-a-Service (BaaS) and NoSQL database was used in this study in order to store the outcome of classifiers. The alert produced based on time and location has also been fetched by the help of this application as well.

An android mobile application was introduced by [11] to identify the accident-prone areas on the southern highway in Sri Lanka based on accident causing factors like speeding, abnormal driving, bad weather, and poor light conditions. The proposed application identified that the accident-prone areas with respect to geographical locations (latitude and longitude). The device sounds an alarm when a user reaching danger zone and it will be faded away as the user passed the danger zone. In addition, a text message will be displayed at the bottom of the application, including the warning level and accident probability.

## 2.3 Gaps and Limitations

However, the studies conducted recently have its own limitations as well. The feature selection directly affected the performance of each and every classifier. Even though the type of day is vital for predicting accidents on highways, most of the studies omit that factor when modeling classifiers. But [6] found that the most accidents that have occurred in the holidays on the southern highway in Sri Lanka. In addition, the features which directly affected for causing accidents may differ from country to country as well. Almost all studies [3, 4, 5] in literature forecasting an injury severity, but not predicting the reason for occurring an accident. [2] and [6] analyzed recorded accidents on the southern highway in Sri Lanka but again, those predictions have been done without applying data mining techniques. Therefore, the prediction accuracy of those researches may not be at a certain level.

Based on the literature, the researchers always keen to adopt an Android platform by omitting iPhone Operating System (IOS) in order to alert the drivers when reaching danger zone. However, the functionality has to be available in IOS platforms as well due to the rapid increase of iPhone usage in Sri Lanka. In addition, the alert services always have to be depended on the reaction time on human beings when driving on highways. The average reaction time of a driver is 2.3 seconds [12]. The reaction time of the driver always depends on travelling speed of the vehicle and the accuracy to function the break [13]. As seen in [11], the alert was produced when a driver enters the radius of 100 meters of the danger zone. If vehicle maintaining a speed of 100/80KMPH when entering the danger zone, the driver is able to reach the exact danger zone within 3.6 and 4.5 seconds respectively. Therefore, in between the above time frame, the mobile application has to complete the voice alert by providing the driver enough time to react. Assuming that the length of voice alert is about 3 to 4 seconds, then the driver does not have enough time to react the alert and which may again cause an accident due to the sudden awareness of danger.

## 2.4 Summary

The purpose of reviewing the literature is to identify the recent trends and techniques that have been used in data mining in order to forecast accident occurrence in highways. In addition, it also identifies certain technologies that can be used when implementing

a mobile application in order to track the current GPS location of the driver. The knowledge gain by conducting a literature survey also identified that the classification algorithms have been used frequently when predicting accidents on highways. Apart from that, association rule and clustering mechanisms have been rapidly adopted to identify key features involved in causing accidents as well. The mobile applications with the help of GPS and SMS services implemented regularly in order to find the exact geographical location and which in return facilitate researchers to provide warning messages when a user reaches to the particular location. The next chapter demonstrates the overall environment setup which will be used throughout this study.

# CHAPTER 3

# Environment Setup

## 3.1 Introduction

This chapter reveals how the experimental environment has been established in order to apply various machine learning techniques and methods identified in last chapter on top of the accident data.

## 3.2 Anaconda Framework

This research makes use of Anaconda Software Distribution as experimental framework which consist over 150 data science packages [14]. These packages used to perform various machine learning tasks which have been listed in the next chapter. Figure 3.1 illustrates an overall setup guide.

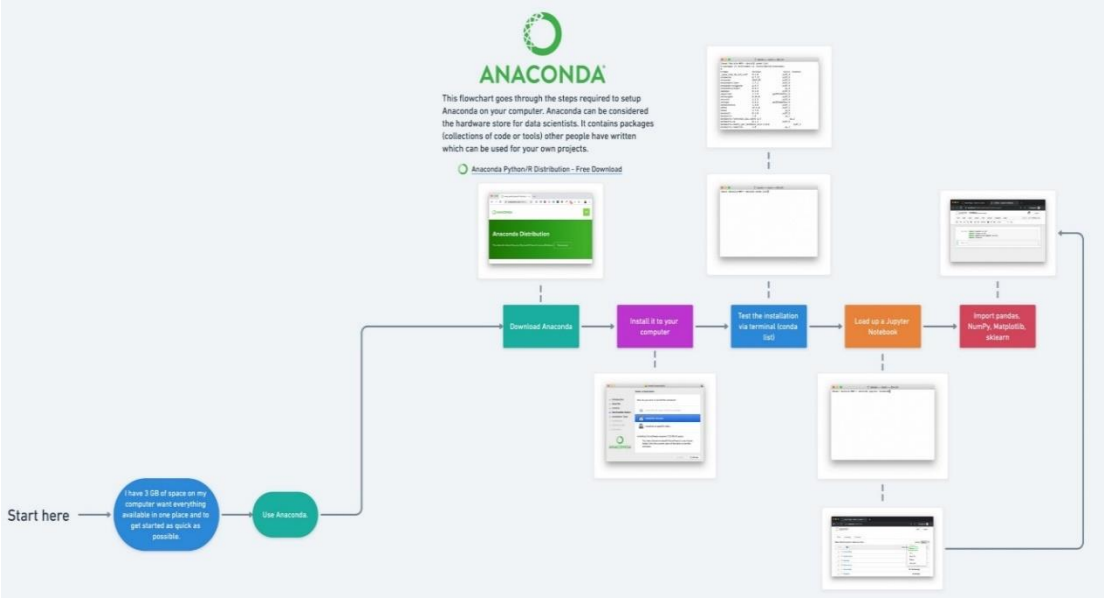



Figure 3.1: Anaconda Software Distribution Setup Guide

## 3.3 Packages and Tools

Several packages that have been used throughout this research in order to deal with several machine learning tasks. Jupyter Notebook was the prime selection to write the python code which enables to running various experiments. In addition, pandas and NumPy packages have imported to manipulating data while exploring and to perform

calculation on numeric data respectively. Furthermore, Matplotlib package used to visualize the findings. Finally, the scikit-learn package has imported as well to learn and analyze various machine learning models and techniques. Apart from that, the MongoDB NoSql database has used in order to store the holidays from 2015 to June 2019. These holiday records were used as required in data preprocessing.

### 3.4 Summary

This chapter explained the reasons to choose the Anaconda Framework to deal with data mining and machine learning techniques. In addition, the packages and tools that have been used on top of the experimental framework have also discussed in detail. The next chapter will address the limitation found in the previous chapter especially identifying most related accident factors and modeling classifiers accordingly in order to predict highway accident accurately in Sri Lanka.

# CHAPTER 4

# A Novel Approach to Forecast Highway Accident in Sri Lanka

## 4.1 Introduction

Chapter four demonstrates how the data mining and machine learning methods and techniques have actually adopted and experimented in order to address the research problem. In addition, each and every step within the data mining and machine learning application domain explained in detail. Further, the development architecture of mobile application has also been introduced later in this chapter.

## 4.2 Data Mining and Machine Learning Application Domain

Data mining techniques and methods are widely used as an approach which has been derived the concepts of both artificial intelligence and statistics. Therefore, it definitely is an advanced tool which can be able to discover complex and hidden patterns exist in huge datasets. The techniques come along with data mining has a huge benefit over other traditional statistical methods with respect to the complicated dataset such as the current dataset on forecasting accident in southern highway, Sri Lanka. The construction of proper model always vital in order to forecasting the reason behind the accidents occurred in southern expressway. The combination of decision-making methods and ensemble techniques are significant to improve the accuracy of the trained model and validated the same with the help of test data.

Based on the objectives of every algorithm, data mining methods can be divided either as predictive or descriptive [15]. Predictive methods such as classification, regression, etc., commonly used to forecast future values which are unknown yet depending on the existing variables. Likewise, the descriptive methods discover the significant patterns and relationships exist in the available data (e.g., association, clustering). The datasets that have been derived for this particular experiment falls into a classification problem because the overall target is assigned to class labels for each and every sample exists in the dataset. In addition, the order of the samples is not making an impact to the outcome due to the fact that the classes are discrete. Furthermore, every sample of this dataset has been assigned to a class and which makes researchers to investigate the performance

of the build model. Therefore, the data can be used either to train or test the model.

### 4.2.1 Preparation of Data Set

Accurate and clean accident data records are vital in order to retrieve better knowledge by applying various and best-fit machine learning algorithms [16]. However, it is quite challenging to prepare the 100% accurate accident data set to deal with machine learning methods. Therefore, this research proposed following instructions which executes carefully to retrieve accurate data set and build machine learning models on top of that. Figure 4.1 represents the overall working process of the proposed model.

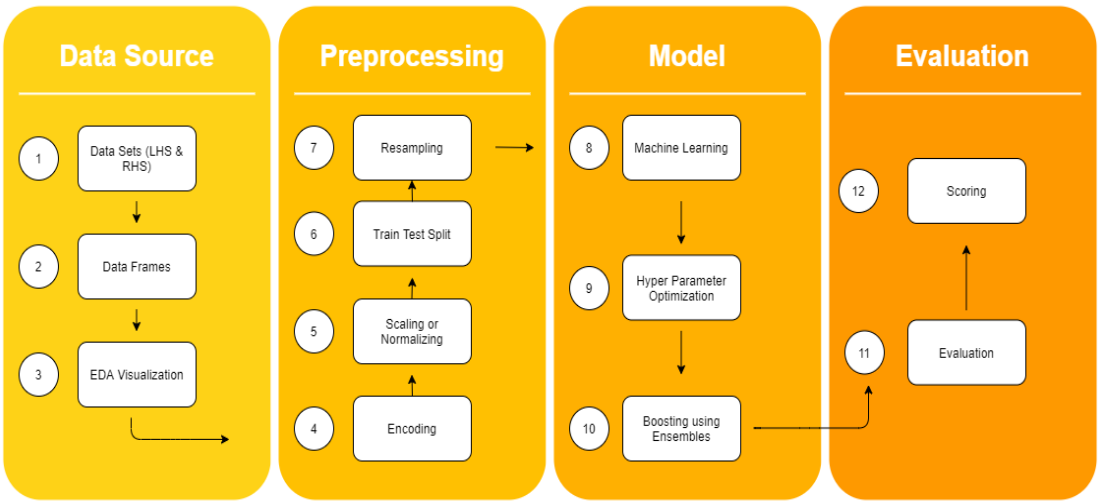



Figure 4.1: The Working Mechanism of Proposed Model

### 4.2.2 Data Source and Description

Accident records on the southern expressway in Sri Lanka have been derived from manual ledgers for the period from January 2015 to June 2019. These ledgers have been maintaining by Expressway Police Control Stations located in three different locations; Galanigama, Kurudugaha Hathakma and Pinnaduwa.The dataset is categorized into two sets; Accidents that happened when driving from Kottawa to Godagama (LHS) and vice-versa (RHS). LHS dataset contains 928 instances (i.e., accident data) and 8 predictors, 6 of them being nominal and the other two is being linear.

On the other hand, the RHS dataset exists with 762 instances of accident records and 8 predictors in which 6 of them have been identified as nominal and other two is being

linear. Table 4.1 demonstrates the detail description of each attribute of both LHS and RHS data set.

Table 4.1: Attribute Description for Both LHS and RHS Dataset

| No. | Features | Data Type | Description |
|---|---|---|---|
| 1 | Accident DateTime | Nominal | The time in which the accident happened. |
| 2 | Vehicle Type | Nominal | The type of a vehicle which is involved in the accidents |
| 3 | Driver Age | Numeric | Age of the driver |
| 4 | Driver Sex | Nominal | Gender of the driver |
| 5 | KM Post | Numeric | The approximate KM Post where the accident happened |
| 6 | Weather Condition | Nominal | The weather condition at the time of accident |
| 7 | Road Surface Condition | Nominal | The road surface condition at the time of accident |
| 8 | Accident Severity | Nominal | Severity of the accident categorized as property damage, fatal, and injury |
| 9 | Accident Cause | Nominal | The reason for the accident and it is the class attribute |

### 4.2.3 Mining Scheme

When choosing relevant Machine Learning algorithms, it is always important to aware how the dataset that is going to use in this study is structured and the purpose that the machine learning task will fulfil. First of all, the purpose of forecasting accident causes makes it a multi-class classification problem due to the data set contains more than two classes (i.e., 6 accident causes), which in this case dismisses all association-rule and clustering techniques. In addition, it is supervised, because the data set consisted with a class value, which assigned to each instance of the data set and machine learning algorithms can be considered this during training the model. The regression functions also are out of the equation due to the fact that the output is discrete. Furthermore, application of neural network probably takes too much time to complete the task which in fact causes certain disadvantages in this study because the model will be used in real-time. The remaining classification techniques can be categorized into four types; Decision trees, Bayes' algorithms, Lazy algorithms, and Rule-based algorithms.

In the Bayes network, there are two types of algorithms; Bayesian Networks and Naive Bayes. Naive Bayes is a kind of low-variance and high-bias classification algorithm which can be built better model even with the small dataset and was used in accident datasets recently [17]. It always requires the features to be independent from one another in order to perform better. This requirement can be fulfilled by looking at the dataset that has been going to use in this study. Bayesian Networks, on the other hand demonstrate the phenomenon for events based on the theory of Bayesian probability rather than logically and also used in real-time accident prediction as well [18]. Both these algorithms function mostly on probabilities rather than boundaries, which will definitely be benefited because the dataset consists with a high mixture of the data.

Due to this mixture and since all the predictors in this study are relevant for determining the accident cause, have to reject the rule-base algorithm as it considers only few features can predict the outcome, whereas several factors associated for the occurrence of an accident. Therefore, these algorithms are out of the equation with respect to this dataset.

In addition, the lazy algorithms such as K-Nearest Neighbor (KNN) has also been eliminated from this study because unlike other classifiers, it waits until the creation of query from the system in order to generalize the training data [19]. This atmosphere does not suit well for this type of research due to the fact that the real-time driver alert system always depends on the time which requires to build the model. Decision trees and ensemble methods, such as Random Forest, Gradient Boost, etc., work better with multi-class problems and also used in the field of accident prediction. These algorithms will be used frequently in this study in order to build the learning models.

Moreover, One-vs-All (OVA) and OVR techniques used to treat the multi-class problem as a binary classification problem expect to adopt along with selected binary classifiers such as SVM, LR for this study because the recent research discovered that OVA solves multi-class problem accurately [20].

### 4.2.4 Preprocessing

Data preprocessing is one of the vital steps in machine learning because the quality and useful information of the derived data matters significantly to the ability of the model to learn. In machine learning point of view, data preprocessing is a state-of-art to get data transformed, or in other word encoded in order to bring whole data, such a way that the machine can easily interpret with the help of various machine learning algorithms.

#### Handling Missing Values and Drop Unwanted Features

The initial exploration of the both data sets (i.e., LHS and RHS) identify few missing values exists in whole data set, and almost all cell values can be found under one attribute. Thus, application of special pre-processing methods to handle missing values may not necessary for this problem. In addition, there are three attributes; Driver Sex, Accident Type, and Driving Side have been removed from the data set because these attributes may not relevant to this study. Out of whole instances for both LHS and RHS datasets, only 14 and 8 accidents recorded against female drivers respectively. Hence, the model is unable to classify the accident cause accurately when the driver is a female. Therefore, driver sex attribute has removed from the dataset along with the female tuples. In addition, the accident occurred due to alcohol consumption has also been removed from both datasets due to the low number of occurrences (10 in RHS and 6 in LHS). Accident type attribute will be considered in the future research by taking it as a class value and removed from the data set. The driving side is not necessary after the dataset categorization into LHS and RHS and eliminated from the data set.

#### Data Exploration

Data exploration has been conducted as an initial step in order to see feature correlation. This research used seaborn package in order to generate the following heatmap with the correlation value to each feature and also to the class.

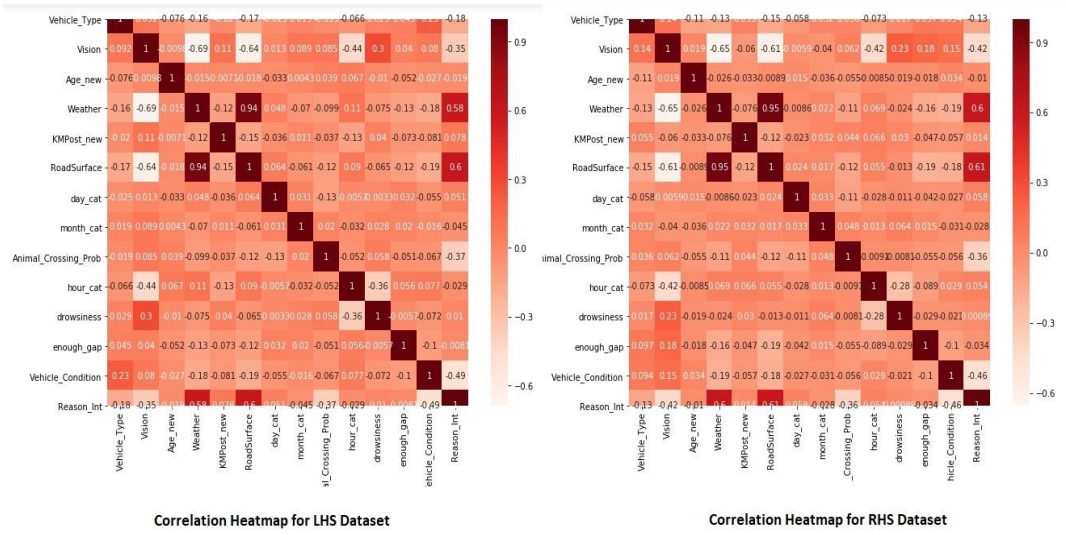



Figure 4.2: Correlation Heatmaps for Both Datasets

According to the heatmap illustrated in Figure 4.2, there are several features have negative correlation to the class and also a positive relation towards few features and also to the class as well. The only concern is that the weather and road surface features have been highly correlated with each other. This study discarded the road surface feature from the dataset while keeping the weather feature in order to mitigate curse of dimensionality and also to simplify the model. In addition, the feature selection does not do well due to the lower number of features and decided to keep all of them as predictors to forecast the reason for occurring an accident.

**Feature Engineering**

Feature engineering has been performed on datetime series with the help of pandas date time series package in order to reproduce new attributes such as month, day and hour of the accident. In addition, Day_Cat attribute has also been generated to determine whether the accident was occurred on a weekday, weekend or public holiday based on the Sri Lankan Calendar. The holidays have been stored in MongoDB Atlas retrieved in order to reproduce the above feature. Further, light condition feature has also produced with the help of hour feature and the weather feature.

Besides that, there are two more features, namely; vehicle condition and enough gap have introduced for this research because the vehicle condition and the gap between vehicles were vital for occurring an accident. This study takes a probability such a way that the accident may occur due to the vehicle condition is solely depends on the bad

vehicle condition. The probability of vehicle condition is always accepting true if the vehicle completed the destination from where the vehicle entered to the highway without facing any accidents due to vehicle condition such as tire blast, tire punch, break bind, etc. If not, the vehicle condition of the particular vehicle categorized as bad. On the other hand, enough gap between two vehicles definitely vital when traveling on a highway in which driver can drive 100 KM per hour. The accidents took place due to tailgating has been categorized as false while true if it was not. Moreover, the critical exploration with the help of aggregating function in pandas discovered that there is a high probability for animals to cross through the road mostly on non-rainy weekdays. Therefore, another new feature has been engineered with the name of animal crossing probability.

Further, drowsiness feature has also been reproduced with the help of aggregating accident reasons by comparing the time of the accident have taken place. The Circadian Rhythm has direct connection to make driver drowsy when driving on highways. This is kind of biologically proven process which has been associated with light [21]. [22] described that the drivers have fallen into a sleep called microsleep during 2PM to 4PM. The melatonin, is a hormone which secreted at night with the help of Circadian Rhythm and it is most likely to occur nearly two hours before the one's regular bedtime [23]. In addition, the food that human being consumes also influences the production of melatonin in the brain [24]. As a result of this hormone, post-meal sleepiness may occur. By considering the above facts, the study raised a flag as true to drowsiness feature when the accidents happened during 8AM to 10AM, 2PM to 4PM and 09PM to 05AM and make it false for other occurrences.

**Feature Encoding**

As described before, the whole purpose behind the data preprocessing is to transform the data, such a way that machine learning algorithm can understand. The reason to perform feature encoding is to encode data such that the machine learning algorithm accepted it as input while retaining its native meaning. This study required to adopt scikit-learn label encoding and pandas dummies (one-hot-encoding) packages in order to transform categorical features to numeric.

### Label Encoding

The approach of label encode is quite simple and it involves transforming each categorical value in a column to a number. Table 4.2 shows the results after application of the label encoder for Bridge_Type column.

Table 4.2: Label Encoded Bridge_Type Column

| BRIDGE-TYPE (TEXT) | BRIDGE-TYPE (NUMERICAL) |
|---|---|
| Arch | 0 |
| Beam | 1 |
| Truss | 2 |
| Cantilever | 3 |
| Tied Arch | 4 |
| Suspension | 5 |
| Cable | 6 |

### One-Hot Encoding

Even though the label encode is easy and straightforward to accomplish, it still suffers with a significant disadvantage. This means that the numeric values may misinterpreted by the certain algorithm due to the sequence of the converted values. This issue has been addressed with the help of one-hot encoder where each category has been converted new column and then assigned a 0 or 1. Table 4.3 shows the example how is the one-hot encoded getting done.

Table 4.3: One-Hot Encoded Safety-Level Column

| SAFETY-LEVEL (TEXT) | SAFETY-LEVEL (None) | SAFETY-LEVEL (Low) | SAFETY-LEVEL (Medium) | SAFETY-LEVEL (High) | SAFETY-LEVEL (Very High) |
|---|---|---|---|---|---|
| None | 1 | 0 | 0 | 0 | 0 |
| Low | 0 | 1 | 0 | 0 | 0 |
| Medium | 0 | 0 | 1 | 0 | 0 |
| High | 0 | 0 | 0 | 1 | 0 |
| Very-High | 0 | 0 | 0 | 0 | 1 |

### Data Discretization

Data Discretization also known as binning is used in this study as well in order to handle the noise of the data. Table 4.4 represents all the features that have been binned.

Table 4.4: Data Discretization or Binning

| Feature Name | Discretization Range | Name Given |
|---|---|---|
| KMPost | (0, 25)<br>(26, 50)<br>(51, 75)<br>(76, 100)<br>(101, 126) | KM1<br>KM2<br>KM3<br>KM4<br>KM5 |
| Age | (17, 29)<br>(30, 49)<br>(50, 76) | Young<br>Mid<br>Older |
| Month | (3, 4, 5, 6, 7, 8, 9)<br>(10, 11, 12, 1, 2) | Peak<br>Off Peak |
| Hour | (2100, 0500)<br>(0600, 0800 \| 1600, 2000)<br>(0900, 1500) | Free of charge<br>Rush<br>Normal |
| Vision | (1900, 0529)<br>(0530, 0659 \| 17:30, 1859)<br>(0700, 1729)<br>(0700, 1729) If rain falls | Poor<br>Glare<br>Normal<br>Blurred |

This research used pandas qcut function in order to achieve the data discretization.

**Standardization**

This is also a vital step in data preprocessing and used to transform the value such a way that the mean of the particular value is 0 whereas the standard deviation is set to 1. There are certain algorithms such as KNN, SVM, LR, etc., required to scale the data for accurate learning. The scaling has been done with the help of the function named StandardScaler which consisted in scikit-learn package.

**Dimensionality Reduction**

Most datasets exist in the field of machine learning consist with large number of features. The more features, in other words huge dimensions make the data analysis task more complicated. As the name implies, the dimension reduction expects to reduce the number of features into smaller dimensions as required for the study.

**Curse of Dimensionality**

The curse of dimensionality refers the phenomena that the task of performing data analysis become complicated as the dimensionality of the features increase. The reason is that the number of planes occupied is increased particularly when dimensionality

increases and resulting huge sparsity of the data which is again complicated to learn and visualize.

By considering the above facts, the dataset that has been derived for this study is undergone the dimension reduction with the help of Principal Component Analysis (PCA) available in scikit-learn package even though the dataset does not contain a large number of features (9). This has been done in order to simplify the data which in fact work well when visualizing the data and learning the model as well. Figure 4.3 illustrated the class distribution of each dataset after reducing the dimensionalities.

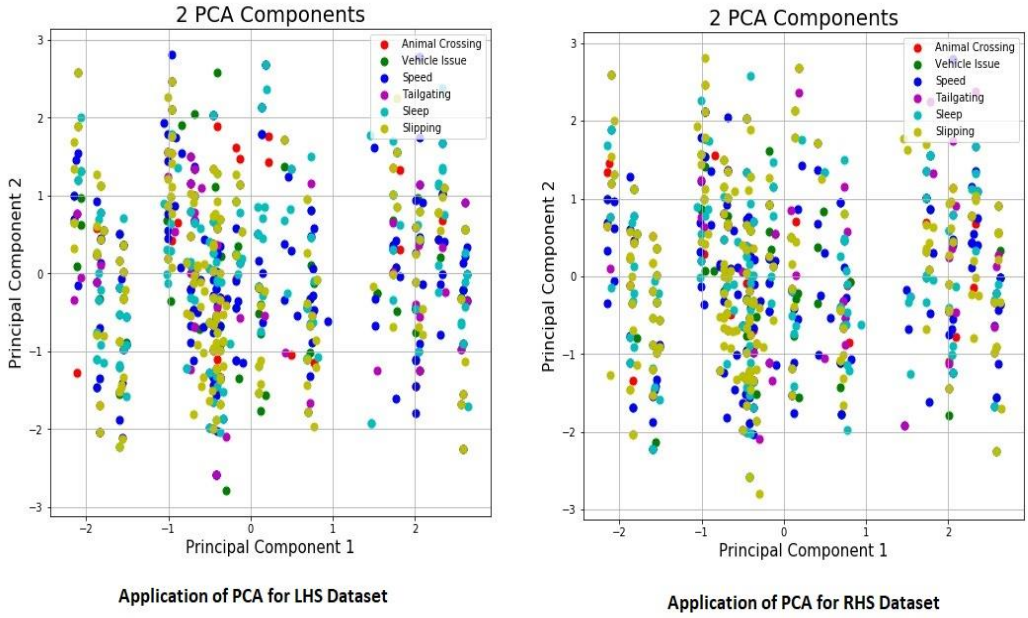



Figure 4.3: PCA for Both Datasets

The above visualization clearly showed that the distribution of feature variance towards classes was not good at all because high variance of feature set has been wrapped as one principal component by omitting few features and use second principal component for the class distribution. However, the all features in this study have provided some sort of pattern which definitely helpful for classifiers to learn well due to the dataset size. By taking all the points into an account, this research discarded to use special dimension reduction or in other words, reducing features with the help of Principal Component Analysis (PCA) and continue with the original features that comes with initial dataset. Finally, the following preprocessed dataset that has been shown in Table 4.5 used to learn the model.

Table 4.5: Preprocessed Dataset

| Vehicle_Type | Vision | Age_new | Weather | KMPost_new | day_cat | month_cat | Animal_Prob | hour_cat | drowsiness | enough_gap | Vehicle_Condition | Reason_Int |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 3 | 2 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 2 | 2 | 0 | 0 | 1 | 1 | 0 | 2 | 0 | 0 | 0 | 5 |
| 0 | 3 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 2 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 5 |

### Train / Validation / Test Split

The dataset has to be divided into train, validation and test sets before applying any machine learning algorithm. Figure 4.4 demonstrates the graphical view of train validation and test splits.


Figure 4.4: Mechanism for Train Validation Test Splits

### Training Data

This is the set of data that the machine learning algorithm actually used to train in order to build the model.

### Validation Data

The part of the dataset used to validate the model fits. In simple, validation dataset is used to enhance the power of model's hyperparameters.

### Test Data

The test data is the part that has been involved to test the hypothesis of the model.

The dataset which used in this study has been split with the split ratio of 0.30. The reason is to use this ratio due to the small no of instances (914) in the experimented dataset. The task is easily done by the train test split function available in scikit-learn package.

### Resampling

Both data sets are highly imbalanced. In LHS data set, 256, 244 and 195 instances

belonging to the class2 (Speed), class4 (Sleep) and class5 (Slipping) respectively. The remaining 219 instances are disproportionately distributed between residual 3 classes. On the other hand, the exploration on RHS dataset also revealed that 213, 204 and 160 instances out of all 754 falls into the class2 (Speed), class5 (Slipping) and class4 (Sleep) respectively. The rest, which is 177 instances unevenly distributed between the other remaining 3 classes. Hence, resampling methods have to be applied especially for the training set before applying classification algorithms. If not, the learning algorithm may predict the outcome mostly based on majority two classes.

### 4.2.5 Parameter Tuning

The performance of almost every classification algorithm has been highly dependent on the decision to choose the relevant hyperparameters. Therefore, the selection of hyperparameters is going to be a hectic task but yet to be functioned in order to gain better results. There are two important searches namely; Grid Search, Random Search available to accomplish this task.

#### Grid Search (GS)

This method is going to test almost all possible combinations of hyperparameters with respect to the algorithm that has been used to learn the model.

#### Randomize Search (RS)

The random search always tries to discover permutation combinations of hyperparameters randomly. This approach is quite time consuming, but manages to provide the best combination of hyperparameters which enhance the performance. The drawback of this method is that the parameter combinations that have been generated randomly and due to that the researchers may not have a chance to narrow down the search and also to identify if there are better parameter combinations exist.

The both searches have been functioned on top of the best performed algorithms and evaluate the results of each critically and carefully.

### 4.2.6 Performance Metric

When selecting an appropriate algorithm to deal with classification problem, it is necessary to keep in mind that there is a useful underlying comparison mechanism exists in order to evaluate the outcome. In general, it is better to consider the overall classification accuracy along with false positive and false negative rates. However, the metrics which measure the performance of classification algorithm has to be selected based on the dataset meaning and the accomplishment of the machine learning task.

In this dataset, the classification accuracy may not be the best metric to evaluate the performance of an algorithm because the classes are highly imbalanced. In this case, even though the learning model predict all the samples as the frequent class, it is quite certain to get a remarkable classification accuracy rate. This does not make sense at all due to the model learnt nothing, but it predicts all the samples based on the majority class. Therefore, it is most important to predict accident cause in this study. Hence, the main values for measuring an algorithm's performance are; Precision, Recall and F1-Score.

Precision is used to identify how often that the model predicts positive samples correctly.

**Precision= True_Positive/ (True_Positive+ False_Positive)**

The other vital metric to consider is the value of recall. This measures how the learning model correctly predicted the fraction of samples belong to a class.

**Recall= True_Positive/ (True_Positive+ False_Negative)**

F1 Score is significant for the problems where both the precision and recall have to be considered like in this study.

**F1-score= 2*Precision*Recall/(Precision+Recall)**

The confusion matrix also considered as significant metrics in this study due to the multi-class behavior and will make use of it when required.

### 4.2.7 Boosting Techniques

The ensemble methods such as Random Forest (RF), XGBoost, Voting, etc., will be used later in this research in order to discover whether there are any improvements after applying them.

### 4.2.8 Experimental Setups

This module represents the two-way experimental plans of the various experiments that have been carried out with the selected LHS and RHS datasets. Further information about the obtained results have been compared with each experiment in detail in the next chapter. The confusion metrics and classification reports can be found in appendix.

The preprocessed dataset illustrated in Table 4.5 has been taken into the account while performing following experiments. Therefore, the preprocessed combination was not performed because the dataset itself has generated by executing the preprocessing techniques except the categorical encoding and resample. The several experiments have performed with the help of all possible combinations and result was increased reasonably. The combinations which worked best out of the lot have been listed in this research and other combinations which did not provide good results were left out.

The initial experiment has been done in detail for the selected classifier first and discovered random over sample along with one-hot encoding worked well for the dataset. Therefore, this study continues to perform experiments with other classifiers as well based on initial experiment.

The selected algorithms initially run on top of the preprocessed datasets including the two well-known encoders namely; label and one-hot. Then, the algorithm which performed decent has been selected and applied resample methods. There were two resample methods, namely; random over sampler for minority classes and class_weight function in scikit-learn for unbalanced dataset.

The class_weight function set to balanced and used Loss Function (LF) keyword to illustrate it because the class_weight replicates the minor class until it has as many samples like in majority classes and which reduces the loss function. Based on the

results, several combinations have been experimented with the help of hyperparameters, bagging and boosting techniques. The algorithm which produced best results has highlighted in yellow.

### Experimental Setup for LHS Dataset

The following were the experiments that have been performed on LHS dataset.

### Decision Tree Classifier

Table 4.6 demonstrates the results generated by decision tree classifier for all combinations.

Table 4.6: Performance of Decision Tree Classifier for LHS Dataset

| See Appendix A | Enc | Res | param | Boost | Acc | F1-Score | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | L | - | - | - | .71 | .64 | 1 | .55 | 1 | .58 | .81 |
| 2 | OH | - | - | - | .73 | .62 | 1 | .60 | 1 | .63 | .81 |
| 3 | OH | ROS | - | - | .71 | .67 | 1 | .55 | 1 | .60 | .80 |
| 4 | OH | LF | - | - | .70 | .69 | 1 | .51 | 1 | .59 | .81 |
| **5** | **OH** | **ROS** | **GS** | **-** | **.74** | **.56** | **1** | **.57** | **1** | **.66** | **.85** |
| **6** | **OH** | **ROS** | **RS** | **-** | **.74** | **.62** | **1** | **.57** | **1** | **.67** | **.85** |
| 7 | OH | ROS | GS | Bag | .73 | .64 | 1 | .56 | 1 | .63 | .85 |
| 8 | OH | ROS | GS | AB | .73 | .67 | 1 | .61 | .94 | .60 | .83 |

### SVM with One-vs-Rest Classifier

As represented in Table 4.7, the SVM with OVR classifier performed as follows.

Table 4.7: Performance of SVM Classifier for LHS Dataset

| See Appendix B | Enc | Res | Param | Acc | F1-Score | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | 0 | 1 | 2 | 3 | 4 | 5 |
| **1** | **OH** | **ROS** | **-** | **.74** | **.75** | **1** | **.56** | **1** | **.66** | **.85** |
| **2** | **OH** | **ROS** | **GS** | **.74** | **.75** | **1** | **.56** | **1** | **.66** | **.85** |
| **3** | **OH** | **ROS** | **RS** | **.74** | **.75** | **1** | **.56** | **1** | **.66** | **.85** |

### Logistic Regression with One-vs-Rest Classifier

The Table 4.8 showed that the results obtained by performing logistic regression classifier with OVR approach.

Table 4.8: Performance of Logistic Regression Classifier for LHS Dataset

| See Appendix C | Enc | Res | param | Acc | F1-Score | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | OH | ROS | - | .75 | .72 | 1 | .58 | 1 | .68 | .85 |
| **2** | **OH** | **ROS** | **GS** | **.75** | **.69** | **1** | **.57** | **1** | **.68** | **.85** |
| **3** | **OH** | **ROS** | **RS** | **.75** | **.69** | **1** | **.57** | **1** | **.68** | **.85** |

**Random Forest Classifier**

The results generated by random forest classifier can be represented with the help of Table 4.9.

Table 4.9: Performance of Random Forest Classifier for LHS Dataset

| See Appendix D | Enc | Res | param | Acc | F1-Score | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | OH | ROS | - | .74 | .72 | 1 | .61 | 1 | .62 | .84 |
| **2** | **OH** | **ROS** | **GS** | **.76** | **.69** | **1** | **.62** | **1** | **.68** | **.85** |
| 3 | OH | ROS | RS | .75 | .75 | 1 | .59 | .97 | .67 | .85 |

**Gradient Boosting Classifier**

The gradient boost classifier returned results (Table 4.10) for all experiment combinations.

Table 4.10: Performance of Gradient Boost Classifier for LHS Dataset

| See Appendix E | Enc | Res | param | Acc | F1-Score | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | OH | ROS | - | .74 | .72 | 1 | .58 | 1 | .67 | .85 |
| **2** | **OH** | **ROS** | **GS** | **.76** | **.75** | **1** | **.61** | **1** | **.67** | **.85** |
| 3 | OH | ROS | RS | .75 | .72 | 1 | .58 | 1 | .67 | .85 |

**Experimental Setup for RHS Dataset**

The following were the experiments that have been performed in RHS dataset.

**Decision Tree Classifier**

Table 4.11 illustrated the all the results that has been generated by decision tree classifier for RHS dataset.

Table 4.11: Performance of Decision Tree Classifier for RHS Dataset

| See Appendix F | Enc | Res | param | Boost | Acc | F1-Score | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | L | - | - | - | .71 | .70 | 1 | .58 | 1 | .46 | .82 |
| 2 | OH | - | - | - | .71 | .67 | 1 | .58 | 1 | .43 | .83 |
| 3 | L | ROS | - | - | .67 | .74 | 1 | .50 | 1 | .42 | .81 |
| 4 | L | LF | - | - | .68 | .78 | 1 | .47 | 1 | .44 | .85 |
| 5 | L | LF | GS | - | .71 | .88 | 1 | .58 | 1 | .31 | .91 |
| 6 | L | LF | RS | - | .71 | .88 | 1 | .58 | 1 | .31 | .91 |
| **7** | **L** | **LF** | **GS** | **Bag** | **.75** | **.88** | **1** | **.64** | **1** | **.44** | **.91** |
| 8 | L | LF | GS | AB | .67 | .75 | 1 | .49 | 1 | .40 | .84 |

### SVM with One-vs-Rest Classifier

The application of SVM with OVR approach return results that has been shown with the help of Table 4.12.

Table 4.12: Performance of SVM Classifier for RHS Dataset

| See Appendix G | Enc | Res | Param | Acc | F1-Score | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | L | LF | - | .73 | .88 | 1 | .57 | 1 | .46 | .91 |
| 2 | L | LF | GS | .71 | .88 | 1 | .49 | 1 | .44 | .91 |
| **3** | **L** | **LF** | **RS** | **.74** | **.82** | **1** | **.57** | **1** | **.49** | **.92** |

### Logistic Regression with One-vs-Rest Classifier

The results produced by logistic regression with the help of OVR classifier can be shown in Table 4.13.

Table 4.13: Performance of Logistic Regression Classifier for RHS Dataset

| See Appendix H | Enc | Res | param | Acc | F1-Score | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | L | LF | - | .70 | .78 | 1 | .53 | 1 | .38 | .91 |
| **2** | **L** | **LF** | **GS** | **.71** | **.88** | **1** | **.53** | **1** | **.39** | **.91** |
| **3** | **L** | **LF** | **RS** | **.71** | **.88** | **1** | **.53** | **1** | **.39** | **.91** |

### Random Forest Classifier

Table 4.14 represented the overall results generated by random forest classifier for all experiment combinations.

Table 4.14: Performance of Random Forest Classifier for RHS Dataset

| See Appendix I | Enc | Res | param | Acc | F1-Score | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | 0 | 1 | 2 | 3 | 4 | 5 |
| **1** | **L** | **LF** | **-** | **.73** | **.88** | **1** | **.59** | **1** | **.46** | **.89** |
| 2 | L | LF | GS | .71 | .78 | 1 | .50 | 1 | .46 | .91 |
| 3 | L | LF | RS | .70 | .78 | 1 | .47 | .97 | .45 | .90 |

**Gradient Boosting Classifier**

The final results obtained by performing gradient boost classifier can be shown with the help of Table 4.15.

Table 4.15: Performance of Gradient Boost Classifier for RHS Dataset

| See Appendix J | Enc | Res | param | Acc | F1-Score | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | L | ROS | - | .70 | .70 | 1 | .54 | 1 | .42 | .89 |
| 2 | L | ROS | GS | .72 | .74 | 1 | .58 | 1 | .43 | .90 |
| **3** | **L** | **ROS** | **RS** | **.73** | **.78** | **1** | **.58** | **1** | **.54** | **.85** |

## 4.3 Mobile Application Development

The IOS Mobile Application will be introduced in order to retrieve the type of vehicle and other relevant records which are important to predict the accident cause. In addition, the exact time and location will be retrieved with the help of the mobile application as well. Furthermore, the Dark Sky API has been used in order to retrieve the weather information based on the current location. When a user reaches the danger zone in expressway, the audio file which stored in the AWS (Amazon Web Services) cloud makes a warning noise. The warning alert will be given to users in advanced based on the vehicle type, driver age, type of day, etc., and the current location of the user who travels on the southern expressway. This also depends on the results generating from the best-tuned classifier.

### 4.3.1 Mobile Application Development Architecture

This research proposed the following mobile application architecture to receive the accident alerts accurately for the drivers who are going to use this service. Figure 4.4 represents the all the relevant technologies that will be going to use for make this service available.

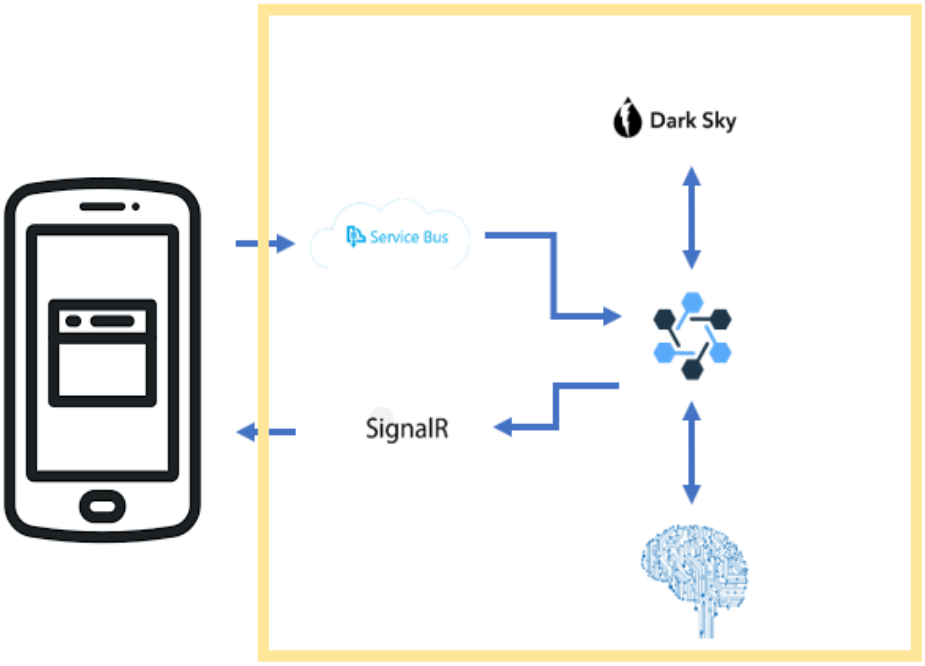



Figure 4.5: Proposed Architecture of Mobile Application

### 4.3.2 Mobile Application Design (UI/UX)

UI (User Interface) and UX (User Experience) design will be performed in order to provide user-friendly interface for drivers by enhancing interaction between user and computer.

### 4.3.3 Implement Database API (Application Program Interface)

Amazon Database Architecture will be implemented and facilitates mobile application to retrieve the required data by calling appropriate APIs.

### 4.3.4 Functional Testing

Functional testing will be performed in order to validate each and every function of the mobile application in order to verify that each function works as expected.

## 4.4 Summary

The 4th chapter represents the complete flow of the data mining and machine learning application domain and discussed each stage in detail. Besides that, it also elaborates the application of selected machine learning algorithms on top of the preprocessed dataset and the result of each have also been listed. The next chapter critically evaluates and compare the results generated by each machine learning algorithms with various modifications.

# CHAPTER 5

# Experimental Result and Discussion

## 5.1 Introduction

This chapter interprets and compare the results which have been generated by each algorithm.

## 5.2 Experimental Analysis

The dataset which used in this study consists of real-time data and was quite complicated to identify the related pattern exists among each due to the dataset size. In addition, this dataset does not use any sort of research till today and experiments on top of it was extremely challengeable. However, the results proved that this research welcomed the existence challenge quite well and dealt with accurately.

There are 1690 highway accident records from Expressway Police Control Centers in Sri Lanka for the period from 2015 to June 2019 were taken into consideration for this study. The whole dataset has been divided into two sets; LHS and RHS with the instance count of 908 and 744 respectively. A total of 12 features used as predictors with the class of accident reason in order to build a classification model to forecast the reason for accident occurrences. The class label on the other hand consisted with 6 discrete classes; animal crossing, vehicle issue, speed, tailgating, sleep and slipping. The Jupyter Notebook powered by Anaconda Framework has been used for this study as a scripter. Python was the language that the study preferred and used scikit-learn package along with pandas data frames and numpy arrays for performing machine learning tasks. The visualization has been functioned with the help of matplotlib along with seaborn package. There were five different classifiers; decision tree, random forest, gradient boost, support vector machine and logistic regression were applied for both datasets and results were examined critically. The support vector machine and logistic regression classifiers have applied with OVR techniques because the problem itself was classified as multi-class. In addition, decision tree has been used as tree-based classifier which was naturally worked well with multi-class problems. Further, two ensembles, random forest and gradient boost were derived for enhancing the performances.

### 5.2.1 Experimental Analysis for LHS Dataset

The detailed experimental analysis for LHS dataset as follows.

**Decision Tree Classifier**

The experimentation has begun by applying the decision tree classifier which, by its nature handled multi-class problems accurately. There were several combinations used on top of the decision tree classifier and the outcome of each evaluate critically. The experiment began by applying two well-known categorical encoding mechanisms; one-hot and label and discovered that the one-hot encoder performed quite well when compared to label encoder. This can be proven with the help of classification reports generated for both instances. As per the confusion matrixes, almost all classes have classified correctly except the speed and sleep class. The correctly classified count has been increased slightly for speed and sleep (46 and 44) when applied one-hot over label encoder (42 and 41). The resample methods have then been applied in order to solve a class imbalance problem. The two well-known approaches; random over sampler and class_weight of scikit-learn were used on top of one-hot encoded dataset and evaluate the outcome. The evaluation showed that the random over sampler just outperformed class_weight function by recording 0.71 as accuracy. This can also be proven by investigating the classification reports along with confusion matrixes. The accuracy decreased slightly when compared to encoding results, but can be accepted because the results were generated on balanced dataset. The experimentation then tunes the hyperparameters with the help of grid search and randomized cross validation approaches and performed almost equally by recording accuracy as 0.74. However, the analysis over classification report shows that the precision and recall were quite good for the randomized grid search approach and selected as best performer so far. Finally, bagging and boosting classifiers have also been applied after tuning the hyperparameters and scores for precision and recall were slightly lower. By considering all the facts, decision tree performed better with the combination of one-hot, random over sampling and randomized search cv.

**SVM with OVR Classifier**

The SVM, one of the best classifiers for binary classification has also been adopted for the multi-class atmosphere with the help of OVR approach. The all combinations listed

in particular Table 4.7 have been experimented on top of one-hot encoded dataset and surprisingly, the results for all 3 combinations were identical. The recorded accuracy was 0.74.

**Logistic Regression with OVR Classifier**

Another binary classifier was also applied on top of the one-hot encoded dataset along with OVR approach. Initially, the classifier run with the default parameters and score 0.75 as accuracy. The scores for precision and recall were quite balanced, even with the default parameters. However, there were 23 samples which belonged to speed class have incorrectly classified as sleep and 21 samples which belong to sleep have also been misclassified as speed. The tuning of hyperparameters does not accurate the model, but shown perfect precision and recall scores for the animal crossing class when compared to the model without tuning the parameters.

**Random Forest Classifier**

Same combination was functioned with the help of the random forest classifier and performed significantly well with the combination of one-hot, random over sampler and grid search as a hyperparameter tuning mechanism. The overall accuracy for this particular combination was 0.76 which was the highest so far. The precision and recall values have almost distributed equally. In addition, the misclassified samples for speed and sleep slightly lower than previous experimental classifiers (19 and 21 respectively).

**Gradient Boost Classifier**

The combination that has performed well in random forest classifier also performed well in this category as well by scoring 0.76 as accuracy. However, the careful evaluation of classification report and confusion matrix suggested that the scores of precision and recall was distributed badly when compared to random forest. In addition, 22 samples which belonged to sleep was misclassified as speed and 21 samples of speed were incorrectly classified as sleep.

### 5.2.2 Experimental Analysis for RHS Dataset

The detailed experimental analysis for LHS dataset as follows.

### Decision Tree Classifier

Like in LHS dataset analysis, the experimental analysis fired up by executing two encoders and analyze their overall performance. The analyzation discovered that both encoders have worked similar with respect to the accuracies but label encoder gained more balanced scores for precision and recall than one-hot encoder. Then, the resample techniques were used and found that the class_weight function in scikit-learn worked well for this dataset rather than of random over sampler. The tuning of hyperparameters using grid and randomized search performed equally for this dataset. The accuracy raised to 0.75 when applying bagging classifier as a boosting technique along with the grid search. However, the scores of precision and recall were unevenly distributed. The AdaBoost classifier performed badly for this dataset with the accuracy of 0.67.

### SVM with OVR Classifier

The application of joint SVM and one-vs-rest classifier showed that tuning of hyperparameters using randomized search outperformed all other combinations that have experimented under this category by scoring 0.74 as accuracy. However, the precision and recall values for sleep class found to be bad just like in decision tree classifier with the values of 0.45 and 0.52 respectively.

### Logistic Regression with OVR Classifier

Another approach of one-vs-rest has been experimented with logistic regression classifier and the results have monitored carefully. The results showed that this approach was also unable to perform well for sleep class with the help of all 3 combinations. The results showed that the values of precision and recall for sleep class still under 0.50 which was the worst so far.

### Random Forest Classifier

The random forest classifier worked well with the default parameter settings and scored 0.73 as accuracy. The distribution of precision and recall scores slightly good when compared to other classifiers, but does not do well for sleep class. The scores of precision and recall for sleep class was 0.47 and 0.48 respectively.

### Gradient Boost Classifier

The gradient boost classifier outperformed all other classifiers in the RHS dataset by classifying most instances correctly of each class. The results were obtained by tuning hyperparameters with the help of randomized search. The recorded accuracy was 0.73 while performing well in all six classes. The scores for precision and recall were well above 0.50 and which was quite an achievement when compared to other classifiers.

## 5.3 Experimental Summary

The experimental results showed that every sample belongs to vehicle issues and tailgating classes have been classified perfectly by scoring 1.00 for both precision and recall. This is quite obvious because all the accident recorded under vehicle issues class has been taken place due to the bad vehicle condition. Likewise, minimum gap was key for the all accidents took place due to tailgating. The samples of slipping class almost performed relatively well with all classifiers by scoring over 0.80 for all precision, recall, f1-score and accuracy as well. However, there was a clear issue in sleep and speed class with respect to the classification. Most of the classifiers experienced that the samples of one of classes incorrectly classified as other class and vice-versa. This may due to the vehicle speed attribute at the time the accident occurred which, unfortunately was not found in both datasets.

## 5.4 Comparison of Experiment Schemes

The ensembles; Random Forest and Gradient Boost classifiers have outperformed from the lot, but it is still in debate to identify an overall winner for both datasets. Almost all classifiers have able to achieve the range of accuracy from 0.70 – 0.76 which in fact is a great achievement by considering this dataset. However, random forest classifier discovered the hidden patterns of data slightly higher than the other classifiers in the LHS dataset while gradient boost performed best for RHS dataset by considering all six classes; animal crossing, vehicle issue, speed, tailgating, sleep and slipping.

## 5.5 Comparison with Previous Work

There are no researches have been carried out specially to forecast the reason behind the occurrences of highways in Sri Lanka and open for the experimentations from now

onwards.

### 5.6 Summary

This chapter critically evaluated the results obtained by applying selected machine learning algorithm in last chapter. In addition, the comparison has been performed between the two experimental schemes and discussed the outcome in detail. Moreover, the results have also been compared with previous experiments as well. The next chapter will conclude the overall result of this study and list the works that need to be concerned in near future.

# CHAPTER 6

# Conclusion and Future Work

## 6.1 Introduction

This dissertation attempted to analyze the accident data on the southern highway in Sri Lanka in order to discover the pattern exist which directly related the cause of an accident with the help of machine learning techniques and methods. The results that have obtained by conducting various experimentations with the help of five selected classifiers; decision tree, support vector machine, logistic regression, random forest and gradient boost suggested that the vehicles in bad conditions will be met with an accident when driving on highway. In addition, the vehicle is in definite danger if it does not follow the minimum gap rule while driving on highways. The lack of speed attribute has lowered the values for both sleep and speed class but can be acceptable. After critical evaluation of classification reports and confusion matrixes, the random forest classifier produced best prediction accuracy (0.76) for LHS dataset while gradient boost performed significantly well for RHS classes with the highest recorded accuracy of 0.73.

## 6.2 Future Developments

The data augmentation and deep learning will expect to use for same datasets near future and evaluate the outcome in detail. The development of mobile application which is used to alert the driver is still in the process and the prototype will be available soon.

## 6.3 Summary

This chapter summarize the thesis by classifying accident reasons for the occurrence of an accident into six separate classes; animal crossing, vehicle issue, speed, tailgating, sleep and slipping. The task was quite challenging but was able to deal with it accurately.

# References

[1] "A road accident every 10 minutes!", Srilankamirror.com, 2019. [Online]. Available: https://srilankamirror.com/news/6874-a-road-accident-every-10-minutes.

[2] R. Chinthanie, "Accident analysis of Southern expressway", Dl.lib.mrt.ac.lk, 2015. [Online]. Available: http://dl.lib.mrt.ac.lk/handle/123/12187.

[3] L. Li, S. Shrestha and G. Hu, "Analysis of road traffic fatal accidents using data mining techniques", in 2017 IEEE 15th International Conference on Software Engineering Research, Management and Applications (SERA), London, UK, 2017.

[4] T. Bahiru, D. Singh and E. Tessfaw, "Comparative Study on Data Mining Classification Algorithms for Predicting Road Traffic Accident Severity", in 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, India, 2018.

[5] M. Taamneh, S. Alkheder and S. Taamneh, "Data-mining techniques for traffic accident modeling and prediction in the United Arab Emirates", Journal of Transportation Safety & Security, vol. 9, no. 2, pp. 146-166, 2016. Available: 10.1080/19439962.2016.1152338.

[6] E. Edirisinghe and A. Edirisinghe, "Analysis of Accidents on the Southern Expressway", in R4TLI Conference Proceedings, 2017.

[7] A. Irfan, R. Al Rasyid and S. Handayani, "Data mining applied for accident prediction model in Indonesia toll road", in AIP Conference Proceedings, 2018.

[8] G. B. S.e., A. Singhai, and R. R. Parida, “Realtime Email Delivery Failure Prediction Using the One-vs-All Classifier,” 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 2018.

[9] N. Jahan, K. Hossen and M. Patwary, "Implementation of a vehicle tracking system using smartphone and SMS service", in 2017 4th International Conference on Advances in Electrical Engineering (ICAEE), Dhaka, Bangladesh, 2017.

[10] S. Mane, M. Patil, S. Chauhan and O. Raut, "Road Accident Alert System using Data Mining", International Journal of Engineering Science and Computing, vol. 8, no. 4, 2018.

[11] K. Sudheera and G. Sandamali, "Warning system for Southern Expressway, Sri Lanka", International Journal of Conceptions on Computing and Information Technology, vol. 2, no. 4, 2014.

[12] D. Sawicki, "Braking Factors", Copradar.com. [Online]. Available: https://copradar.com/redlight/factors/.

[13] M. Zhuk, V. Kovalyshyn, Y. Royko and K. Barvinska, "Research on drivers' reaction time in different conditions", Eastern-European Journal of Enterprise Technologies, vol. 2, no. 386, pp. 24-31, 2017. Available: 10.15587/1729-4061.2017.98103.

[14] E. Bourke, "Get your computer ready for machine learning: How, what and why you should use Anaconda, Miniconda…," 2019 [online]. Available at: https://towardsdatascience.com/get-your-computer-ready-for-machine-learning-how-what-and-why-you-should-use-anaconda-miniconda-d213444f36d6 [Accessed 15 Feb. 2020].

[15] F. Gorunescu. Data mining concepts, models and techniques. Springer, Berlin, 2011.

[16] M. F. Labib, A. S. Rifat, M. M. Hossain, A. K. Das and F. Nawrine, "Road Accident Analysis and Prediction of Accident Severity by Using Machine Learning in Bangladesh," 2019 7th International Conference on Smart Computing & Communications (ICSCC), Sarawak, Malaysia, Malaysia, 2019, pp. 1-5.

[17] L. G. Cuenca, E. Puertas, N. Aliane and J. F. Andres, "Traffic Accidents Classification and Injury Severity Prediction," 2018 3rd IEEE International Conference on Intelligent Transportation Engineering (ICITE), Singapore, 2018, pp. 52-57.

[18] M. Wu, D. Shan, Z. Wang, X. Sun, J. Liu and M. Sun, "A Bayesian Network Model for Real-time Crash Prediction Based on Selected Variables by Random Forest," 2019 5th International Conference on Transportation Information and Safety (ICTIS), Liverpool, United Kingdom, 2019, pp. 670-677.

[19] L. Wahab and H. Jiang, "A comparative study on machine learning based algorithms for prediction of motorcycle crash severity," 2019, PLOS ONE, 14(4), p.e0214966.

[20] G. B. S.E., A. Singhai and R. R. Parida, "Realtime Email Delivery Failure Prediction Using the One-vs-All Classifier," 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, 2018, pp. 961-964.

[21] Chipman, Mary, and Yue Lena Jin. "Drowsy Drivers: The Effect of Light and Circadian Rhythm on Crash Occurrence." Safety Science, vol. 47, no. 10, 2009, pp. 1364–1370., doi:10.1016/j.ssci.2009.03.005.

[22] Varagur, Krithika. "Why Drivers Should Take The Afternoon Slump Seriously." HuffPost, HuffPost, 9 June 2016, www.huffpost.com/entry/afternoon-slump-drowsy-driving_n_57586b13e4b0ced23ca6c179.

[23] Khullar, Atul. "The Role of Melatonin in the Circadian Rhythm Sleep-Wake Cycle." Psychiatric Times, 10 July 2012, www.psychiatrictimes.com/sleep-disorders/role-melatonin-circadian-rhythm-sleep-wake-cycle.

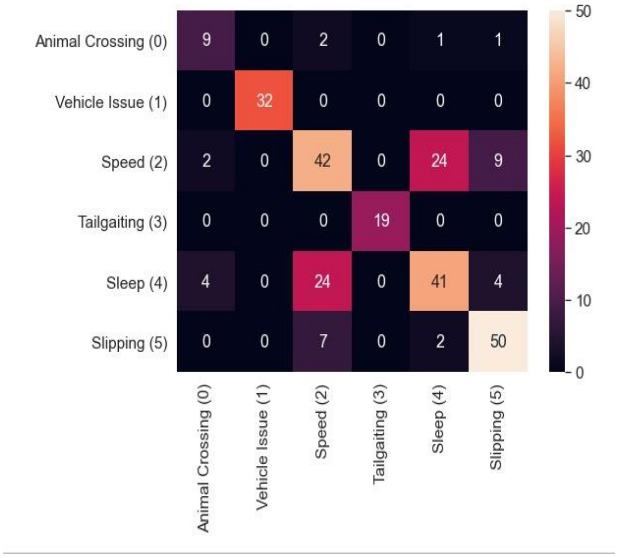
[24] Jakarta Post, “Why do we feel sleepy after eating a meal?,” The Jakarta Post. [Online]. Available: https://www.thejakartapost.com/life/2016/10/04/why-do-we-feel-sleepy-after-eating-a-meal.html. [Accessed: 11-May-2020].

# Appendix - A

## Decision Tree Classifier LHS

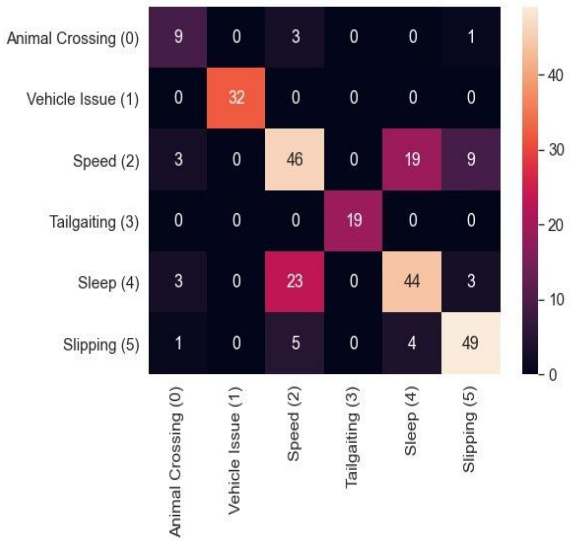
Appendix A: 1 Classification Report for Decision Tree Classifier 1

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.60 | 0.69 | 0.64 | 13 |
| 1 | 1.00 | 1.00 | 1.00 | 32 |
| 2 | 0.56 | 0.55 | 0.55 | 77 |
| 3 | 1.00 | 1.00 | 1.00 | 19 |
| 4 | 0.60 | 0.56 | 0.58 | 73 |
| 5 | 0.78 | 0.85 | 0.81 | 59 |
| accuracy | | | 0.71 | 273 |
| macro avg | 0.76 | 0.77 | 0.77 | 273 |
| weighted avg | 0.70 | 0.71 | 0.70 | 273 |



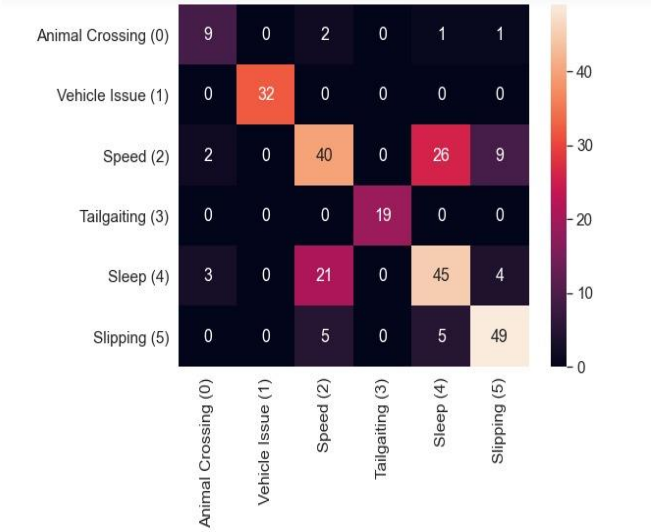

Appendix A: 2 Classification Report for Decision Tree Classifier 2

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.56 | 0.69 | 0.62 | 13 |
| 1 | 1.00 | 1.00 | 1.00 | 32 |
| 2 | 0.60 | 0.60 | 0.60 | 77 |
| 3 | 1.00 | 1.00 | 1.00 | 19 |
| 4 | 0.66 | 0.60 | 0.63 | 73 |
| 5 | 0.79 | 0.83 | 0.81 | 59 |
| accuracy | | | 0.73 | 273 |
| macro avg | 0.77 | 0.79 | 0.78 | 273 |
| weighted avg | 0.73 | 0.73 | 0.73 | 273 |



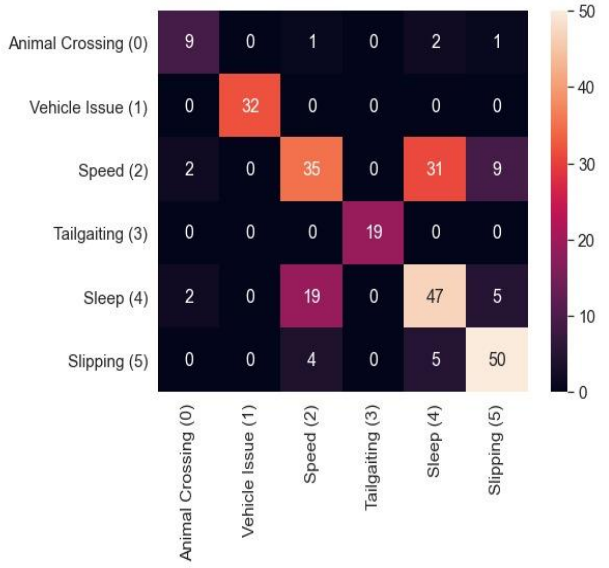
Appendix A: 3 Classification Report for Decision Tree Classifier 3

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.64 | 0.69 | 0.67 | 13 |
| 1 | 1.00 | 1.00 | 1.00 | 32 |
| 2 | 0.59 | 0.52 | 0.55 | 77 |
| 3 | 1.00 | 1.00 | 1.00 | 19 |
| 4 | 0.58 | 0.62 | 0.60 | 73 |
| 5 | 0.78 | 0.83 | 0.80 | 59 |
| accuracy | | | 0.71 | 273 |
| macro avg | 0.77 | 0.78 | 0.77 | 273 |
| weighted avg | 0.71 | 0.71 | 0.71 | 273 |



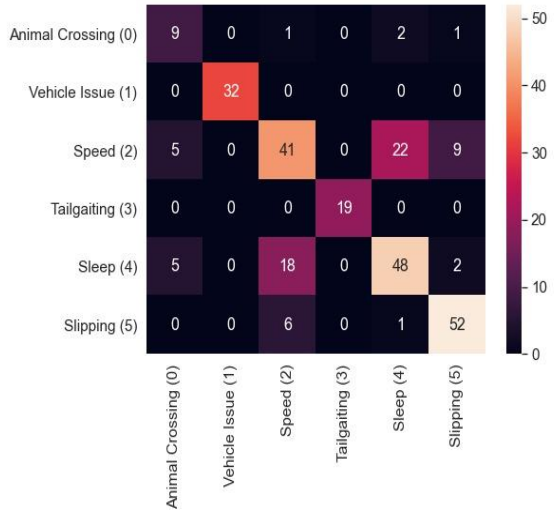

Appendix A: 4 Classification Report for Decision Tree Classifier 4

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.69 | 0.69 | 0.69 | 13 |
| 1 | 1.00 | 1.00 | 1.00 | 32 |
| 2 | 0.59 | 0.45 | 0.51 | 77 |
| 3 | 1.00 | 1.00 | 1.00 | 19 |
| 4 | 0.55 | 0.64 | 0.59 | 73 |
| 5 | 0.77 | 0.85 | 0.81 | 59 |
| accuracy | | | 0.70 | 273 |
| macro avg | 0.77 | 0.77 | 0.77 | 273 |
| weighted avg | 0.70 | 0.70 | 0.70 | 273 |



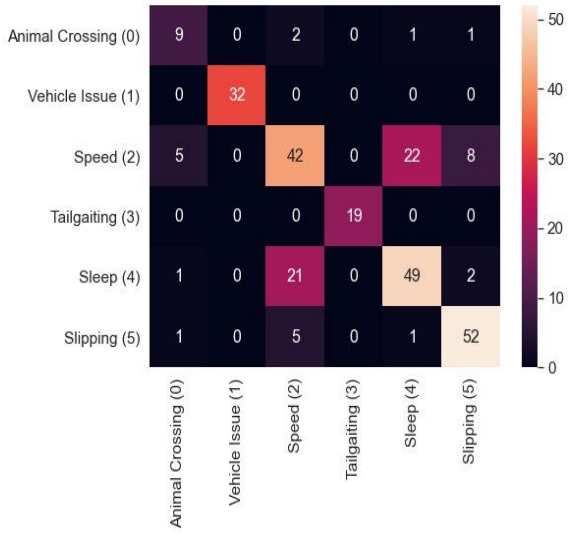
Appendix A: 5 Classification Report for Decision Tree Classifier 5

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.47 | 0.69 | 0.56 | 13 |
| 1 | 1.00 | 1.00 | 1.00 | 32 |
| 2 | 0.62 | 0.53 | 0.57 | 77 |
| 3 | 1.00 | 1.00 | 1.00 | 19 |
| 4 | 0.66 | 0.66 | 0.66 | 73 |
| 5 | 0.81 | 0.88 | 0.85 | 59 |
| accuracy | | | 0.74 | 273 |
| macro avg | 0.76 | 0.79 | 0.77 | 273 |
| weighted avg | 0.74 | 0.74 | 0.73 | 273 |



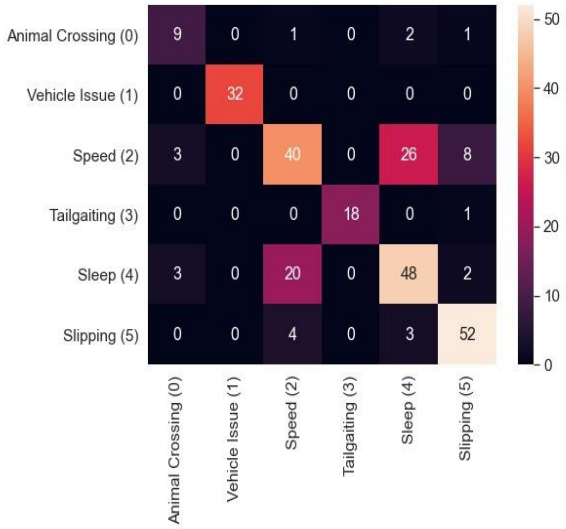

Appendix A: 6 Classification Report for Decision Tree Classifier 6

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.56 | 0.69 | 0.62 | 13 |
| 1 | 1.00 | 1.00 | 1.00 | 32 |
| 2 | 0.60 | 0.55 | 0.57 | 77 |
| 3 | 1.00 | 1.00 | 1.00 | 19 |
| 4 | 0.67 | 0.67 | 0.67 | 73 |
| 5 | 0.83 | 0.88 | 0.85 | 59 |
| accuracy | | | 0.74 | 273 |
| macro avg | 0.78 | 0.80 | 0.79 | 273 |
| weighted avg | 0.74 | 0.74 | 0.74 | 273 |



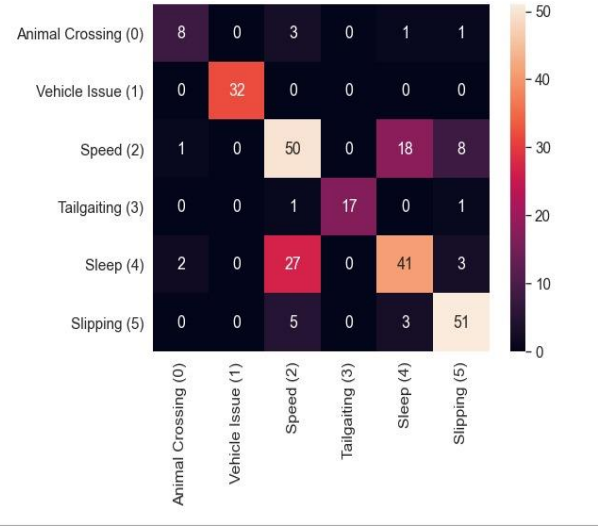
Appendix A: 7 Classification Report for Decision Tree Classifier 7

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.60 | 0.69 | 0.64 | 13 |
| 1 | 1.00 | 1.00 | 1.00 | 32 |
| 2 | 0.62 | 0.52 | 0.56 | 77 |
| 3 | 1.00 | 0.95 | 0.97 | 19 |
| 4 | 0.61 | 0.66 | 0.63 | 73 |
| 5 | 0.81 | 0.88 | 0.85 | 59 |
| accuracy | | | 0.73 | 273 |
| macro avg | 0.77 | 0.78 | 0.78 | 273 |
| weighted avg | 0.73 | 0.73 | 0.73 | 273 |




Appendix A: 8 Classification Report for Decision Tree Classifier 8

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.73 | 0.62 | 0.67 | 13 |
| 1 | 1.00 | 1.00 | 1.00 | 32 |
| 2 | 0.58 | 0.65 | 0.61 | 77 |
| 3 | 1.00 | 0.89 | 0.94 | 19 |
| 4 | 0.65 | 0.56 | 0.60 | 73 |
| 5 | 0.80 | 0.86 | 0.83 | 59 |
| accuracy | | | 0.73 | 273 |
| macro avg | 0.79 | 0.76 | 0.78 | 273 |
| weighted avg | 0.73 | 0.73 | 0.73 | 273 |

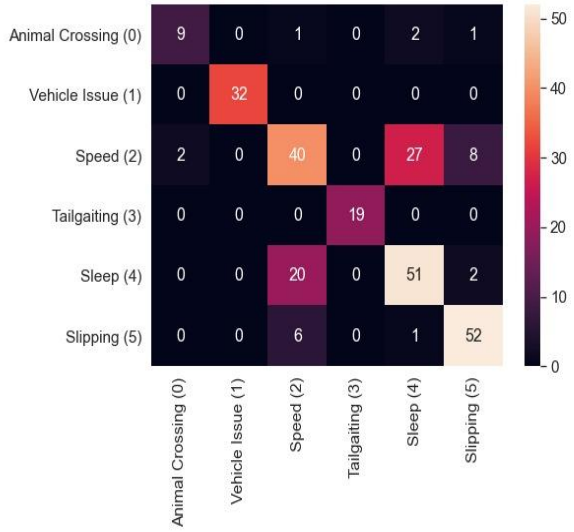


# Appendix - B

**SVM with OVR Classifier LHS**

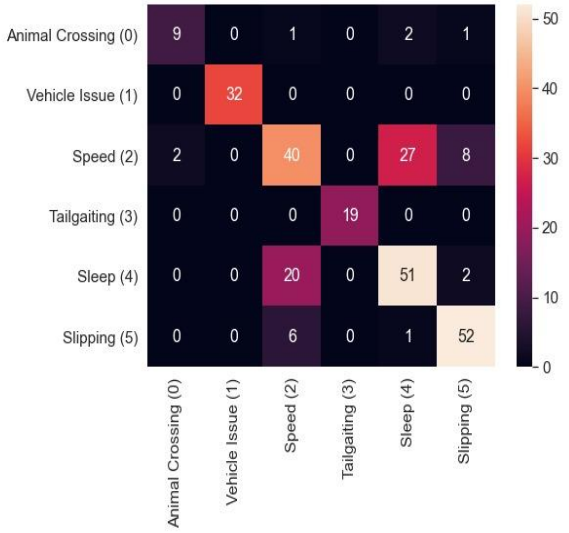
Appendix B: 1 Classification Report for SVM Classifier 1

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.82 | 0.69 | 0.75 | 13 |
| 1 | 1.00 | 1.00 | 1.00 | 32 |
| 2 | 0.60 | 0.52 | 0.56 | 77 |
| 3 | 1.00 | 1.00 | 1.00 | 19 |
| 4 | 0.63 | 0.70 | 0.66 | 73 |
| 5 | 0.83 | 0.88 | 0.85 | 59 |
| accuracy | | | 0.74 | 273 |
| macro avg | 0.81 | 0.80 | 0.80 | 273 |
| weighted avg | 0.74 | 0.74 | 0.74 | 273 |



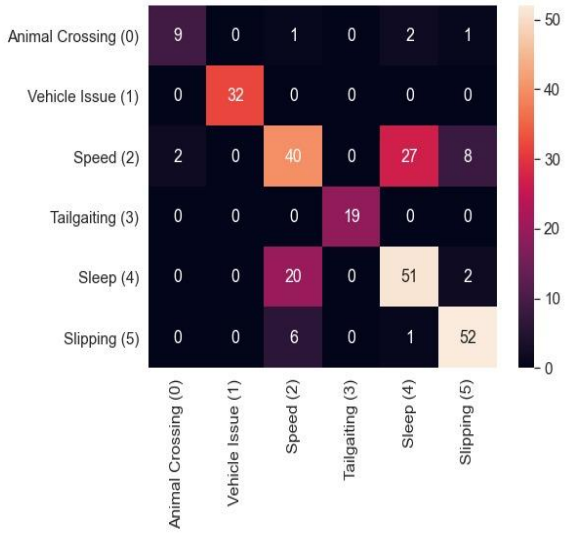

Appendix B: 2 Classification Report for SVM Classifier 2

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.82 | 0.69 | 0.75 | 13 |
| 1 | 1.00 | 1.00 | 1.00 | 32 |
| 2 | 0.60 | 0.52 | 0.56 | 77 |
| 3 | 1.00 | 1.00 | 1.00 | 19 |
| 4 | 0.63 | 0.70 | 0.66 | 73 |
| 5 | 0.83 | 0.88 | 0.85 | 59 |
| accuracy | | | 0.74 | 273 |
| macro avg | 0.81 | 0.80 | 0.80 | 273 |
| weighted avg | 0.74 | 0.74 | 0.74 | 273 |



Appendix B: 3 Classification Report for SVM Classifier 3

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.82 | 0.69 | 0.75 | 13 |
| 1 | 1.00 | 1.00 | 1.00 | 32 |
| 2 | 0.60 | 0.52 | 0.56 | 77 |
| 3 | 1.00 | 1.00 | 1.00 | 19 |
| 4 | 0.63 | 0.70 | 0.66 | 73 |
| 5 | 0.83 | 0.88 | 0.85 | 59 |
| accuracy | | | 0.74 | 273 |
| macro avg | 0.81 | 0.80 | 0.80 | 273 |
| weighted avg | 0.74 | 0.74 | 0.74 | 273 |

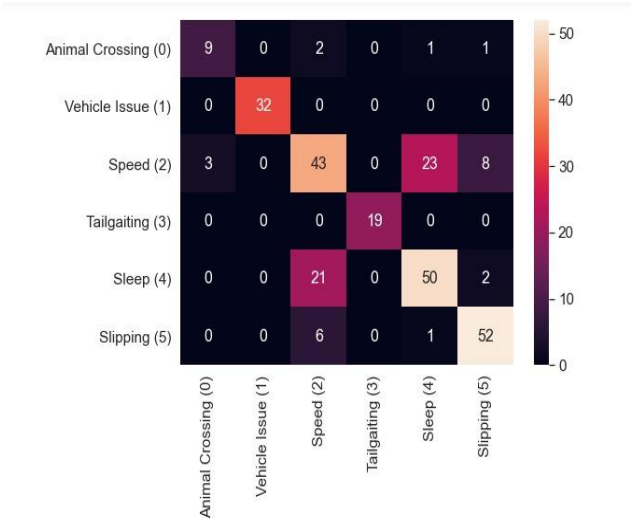


# Appendix - C

## LR with OVR Classifier LHS

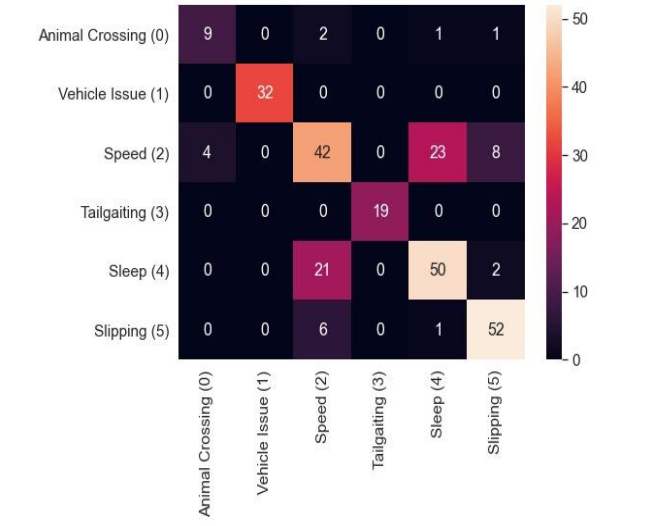
Appendix C: 1 Classification Report for LR Classifier 1

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.75 | 0.69 | 0.72 | 13 |
| 1 | 1.00 | 1.00 | 1.00 | 32 |
| 2 | 0.60 | 0.56 | 0.58 | 77 |
| 3 | 1.00 | 1.00 | 1.00 | 19 |
| 4 | 0.67 | 0.68 | 0.68 | 73 |
| 5 | 0.83 | 0.88 | 0.85 | 59 |
| accuracy | | | 0.75 | 273 |
| macro avg | 0.81 | 0.80 | 0.80 | 273 |
| weighted avg | 0.75 | 0.75 | 0.75 | 273 |



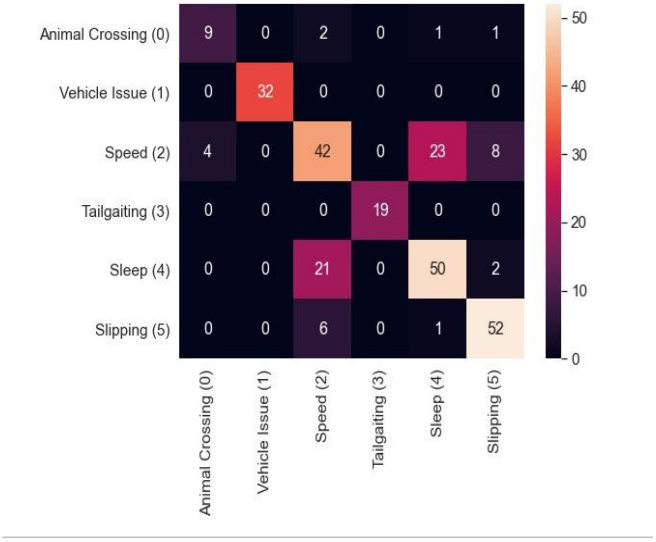

Appendix C: 2 Classification Report for LR Classifier 2

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.69 | 0.69 | 0.69 | 13 |
| 1 | 1.00 | 1.00 | 1.00 | 32 |
| 2 | 0.59 | 0.55 | 0.57 | 77 |
| 3 | 1.00 | 1.00 | 1.00 | 19 |
| 4 | 0.67 | 0.68 | 0.68 | 73 |
| 5 | 0.83 | 0.88 | 0.85 | 59 |
| accuracy | | | 0.75 | 273 |
| macro avg | 0.80 | 0.80 | 0.80 | 273 |
| weighted avg | 0.74 | 0.75 | 0.74 | 273 |



Appendix C: 3 Classification Report for LR Classifier 3

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.69 | 0.69 | 0.69 | 13 |
| 1 | 1.00 | 1.00 | 1.00 | 32 |
| 2 | 0.59 | 0.55 | 0.57 | 77 |
| 3 | 1.00 | 1.00 | 1.00 | 19 |
| 4 | 0.67 | 0.68 | 0.68 | 73 |
| 5 | 0.83 | 0.88 | 0.85 | 59 |
| accuracy | | | 0.75 | 273 |
| macro avg | 0.80 | 0.80 | 0.80 | 273 |
| weighted avg | 0.74 | 0.75 | 0.74 | 273 |

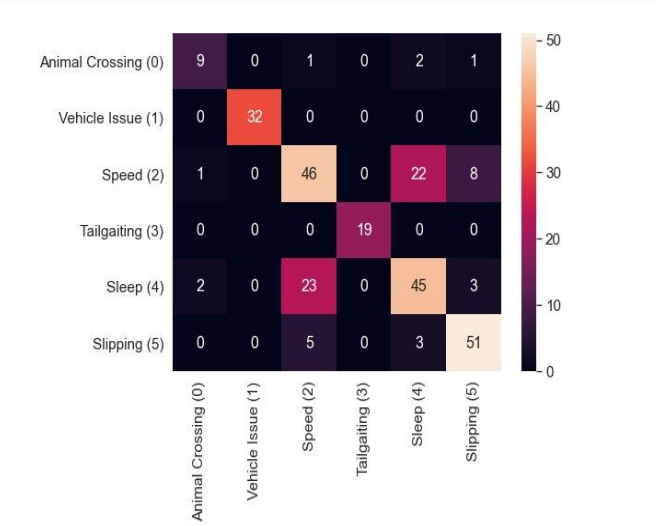


# Appendix - D

## Random Forest Classifier LHS

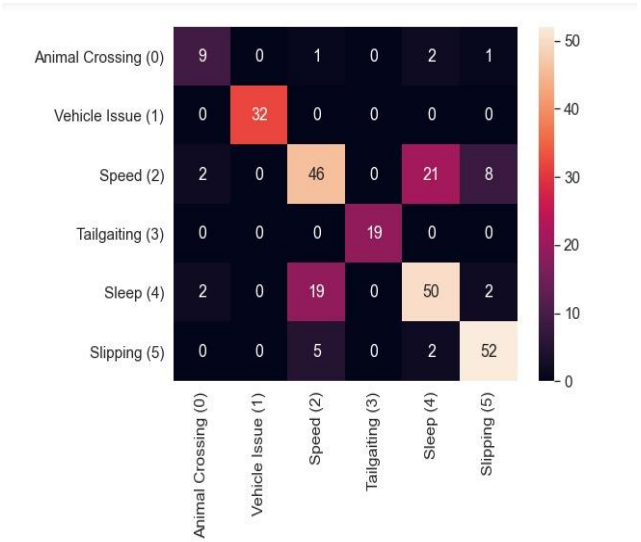
Appendix D: 1 Classification Report for Random Forest Classifier 1

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.75 | 0.69 | 0.72 | 13 |
| 1 | 1.00 | 1.00 | 1.00 | 32 |
| 2 | 0.61 | 0.60 | 0.61 | 77 |
| 3 | 1.00 | 1.00 | 1.00 | 19 |
| 4 | 0.62 | 0.62 | 0.62 | 73 |
| 5 | 0.81 | 0.86 | 0.84 | 59 |
| accuracy | | | 0.74 | 273 |
| macro avg | 0.80 | 0.80 | 0.80 | 273 |
| weighted avg | 0.74 | 0.74 | 0.74 | 273 |



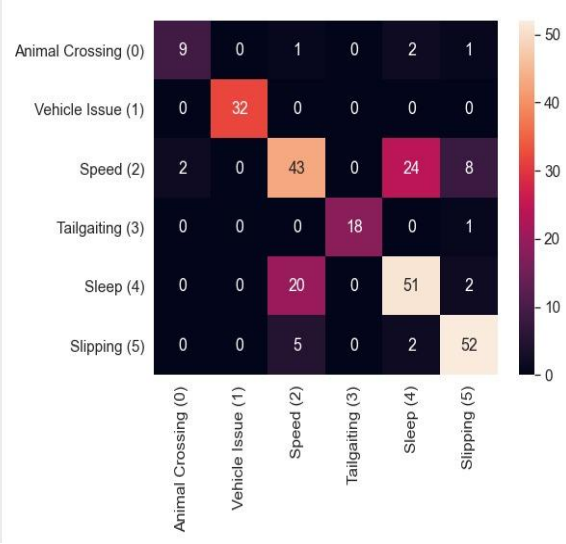

Appendix D: 2 Classification Report for Random Forest Classifier 2

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.69 | 0.69 | 0.69 | 13 |
| 1 | 1.00 | 1.00 | 1.00 | 32 |
| 2 | 0.65 | 0.60 | 0.62 | 77 |
| 3 | 1.00 | 1.00 | 1.00 | 19 |
| 4 | 0.67 | 0.68 | 0.68 | 73 |
| 5 | 0.83 | 0.88 | 0.85 | 59 |
| accuracy | | | 0.76 | 273 |
| macro avg | 0.81 | 0.81 | 0.81 | 273 |
| weighted avg | 0.76 | 0.76 | 0.76 | 273 |



Appendix D: 3 Classification Report for Random Forest Classifier 3

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.82 | 0.69 | 0.75 | 13 |
| 1 | 1.00 | 1.00 | 1.00 | 32 |
| 2 | 0.62 | 0.56 | 0.59 | 77 |
| 3 | 1.00 | 0.95 | 0.97 | 19 |
| 4 | 0.65 | 0.70 | 0.67 | 73 |
| 5 | 0.81 | 0.88 | 0.85 | 59 |
| accuracy | | | 0.75 | 273 |
| macro avg | 0.82 | 0.80 | 0.80 | 273 |
| weighted avg | 0.75 | 0.75 | 0.75 | 273 |

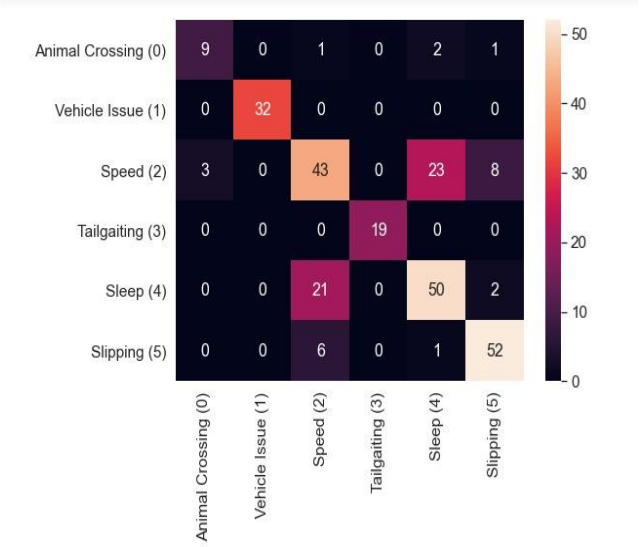


# Appendix - E

## Gradient Boosting Classifier LHS

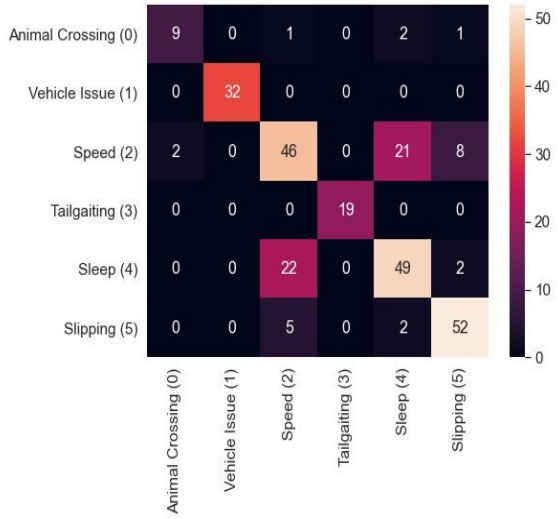
Appendix E: 1 Classification Report for Gradient Boost Classifier 1

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.75 | 0.69 | 0.72 | 13 |
| 1 | 1.00 | 1.00 | 1.00 | 32 |
| 2 | 0.61 | 0.56 | 0.58 | 77 |
| 3 | 1.00 | 1.00 | 1.00 | 19 |
| 4 | 0.66 | 0.68 | 0.67 | 73 |
| 5 | 0.83 | 0.88 | 0.85 | 59 |
| accuracy | | | 0.75 | 273 |
| macro avg | 0.81 | 0.80 | 0.80 | 273 |
| weighted avg | 0.75 | 0.75 | 0.75 | 273 |



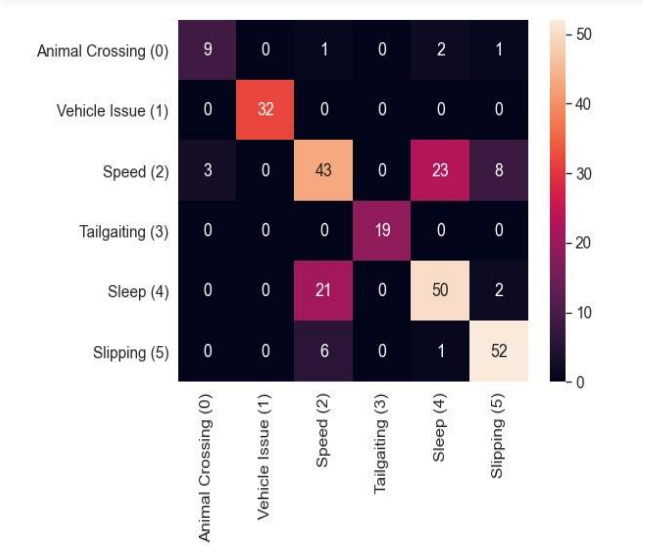

Appendix E: 2 Classification Report for Gradient Boost Classifier 2

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.82 | 0.69 | 0.75 | 13 |
| 1 | 1.00 | 1.00 | 1.00 | 32 |
| 2 | 0.62 | 0.60 | 0.61 | 77 |
| 3 | 1.00 | 1.00 | 1.00 | 19 |
| 4 | 0.66 | 0.67 | 0.67 | 73 |
| 5 | 0.83 | 0.88 | 0.85 | 59 |
| accuracy | | | 0.76 | 273 |
| macro avg | 0.82 | 0.81 | 0.81 | 273 |
| weighted avg | 0.76 | 0.76 | 0.76 | 273 |



Appendix E: 3 Classification Report for Gradient Boost Classifier 3

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.75 | 0.69 | 0.72 | 13 |
| 1 | 1.00 | 1.00 | 1.00 | 32 |
| 2 | 0.61 | 0.56 | 0.58 | 77 |
| 3 | 1.00 | 1.00 | 1.00 | 19 |
| 4 | 0.66 | 0.68 | 0.67 | 73 |
| 5 | 0.83 | 0.88 | 0.85 | 59 |
| accuracy | | | 0.75 | 273 |
| macro avg | 0.81 | 0.80 | 0.80 | 273 |
| weighted avg | 0.75 | 0.75 | 0.75 | 273 |

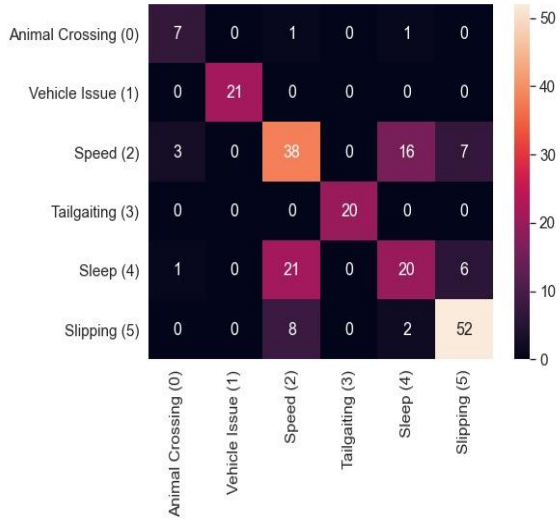


# Appendix - F

## Decision Tree Classifier RHS

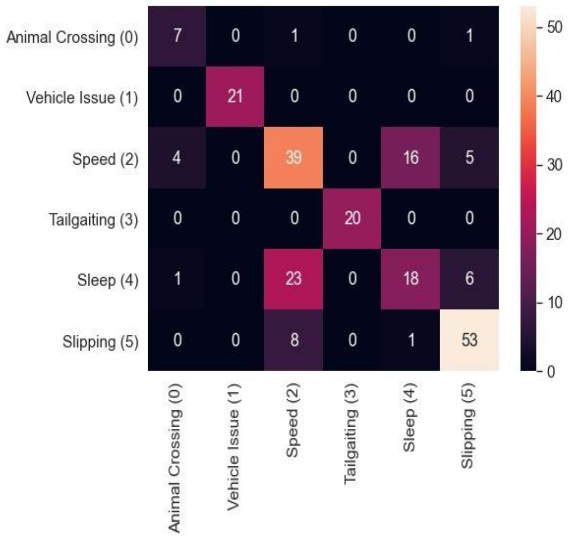
Appendix F: 1 Classification Report for Decision Tree Classifier 1

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.64 | 0.78 | 0.70 | 9 |
| 1 | 1.00 | 1.00 | 1.00 | 21 |
| 2 | 0.56 | 0.59 | 0.58 | 64 |
| 3 | 1.00 | 1.00 | 1.00 | 20 |
| 4 | 0.51 | 0.42 | 0.46 | 48 |
| 5 | 0.80 | 0.84 | 0.82 | 62 |
| accuracy | | | 0.71 | 224 |
| macro avg | 0.75 | 0.77 | 0.76 | 224 |
| weighted avg | 0.70 | 0.71 | 0.70 | 224 |



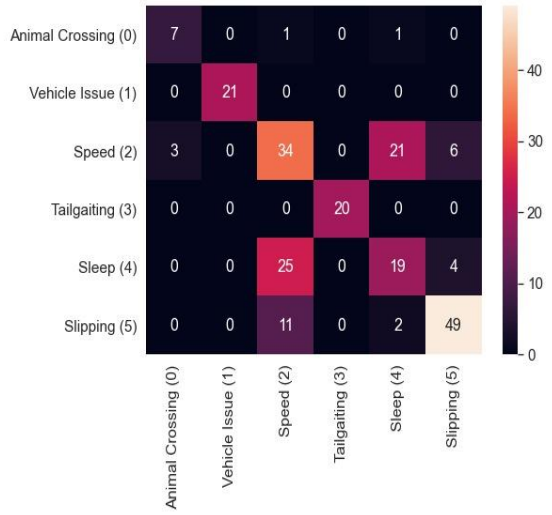

Appendix F: 2 Classification Report for Decision Tree Classifier 2

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.58 | 0.78 | 0.67 | 9 |
| 1 | 1.00 | 1.00 | 1.00 | 21 |
| 2 | 0.55 | 0.61 | 0.58 | 64 |
| 3 | 1.00 | 1.00 | 1.00 | 20 |
| 4 | 0.51 | 0.38 | 0.43 | 48 |
| 5 | 0.82 | 0.85 | 0.83 | 62 |
| accuracy | | | 0.71 | 224 |
| macro avg | 0.74 | 0.77 | 0.75 | 224 |
| weighted avg | 0.70 | 0.71 | 0.70 | 224 |



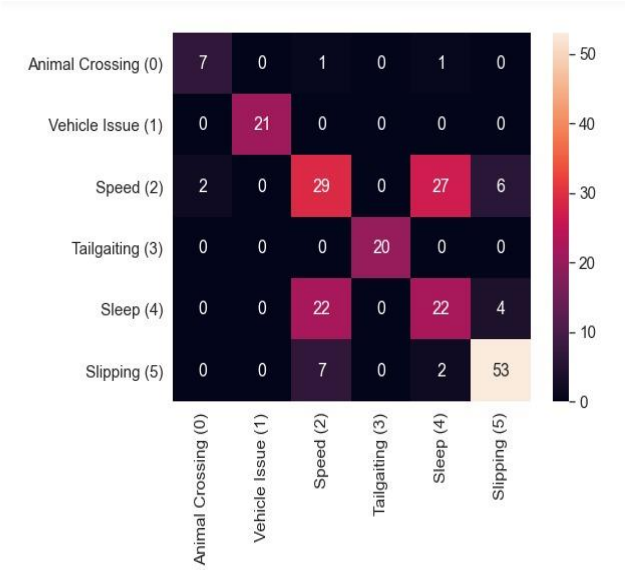
Appendix F: 3 Classification Report for Decision Tree Classifier 3

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.70 | 0.78 | 0.74 | 9 |
| 1 | 1.00 | 1.00 | 1.00 | 21 |
| 2 | 0.48 | 0.53 | 0.50 | 64 |
| 3 | 1.00 | 1.00 | 1.00 | 20 |
| 4 | 0.44 | 0.40 | 0.42 | 48 |
| 5 | 0.83 | 0.79 | 0.81 | 62 |
| accuracy | | | 0.67 | 224 |
| macro avg | 0.74 | 0.75 | 0.74 | 224 |
| weighted avg | 0.67 | 0.67 | 0.67 | 224 |



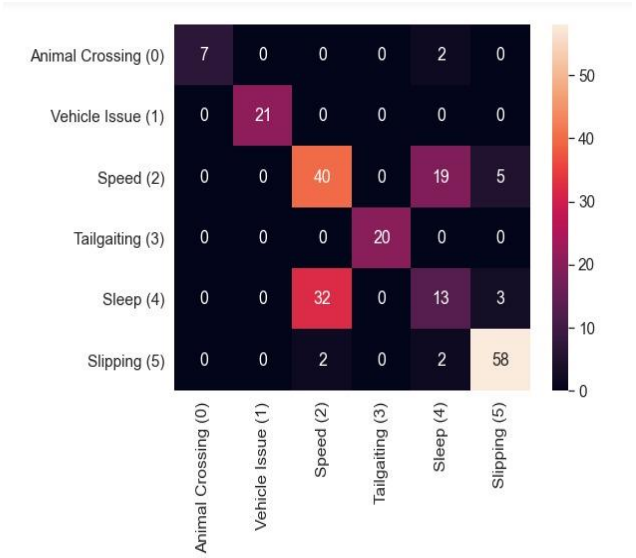

Appendix F: 4 Classification Report for Decision Tree Classifier 4

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.78 | 0.78 | 0.78 | 9 |
| 1 | 1.00 | 1.00 | 1.00 | 21 |
| 2 | 0.49 | 0.45 | 0.47 | 64 |
| 3 | 1.00 | 1.00 | 1.00 | 20 |
| 4 | 0.42 | 0.46 | 0.44 | 48 |
| 5 | 0.84 | 0.85 | 0.85 | 62 |
| accuracy | | | 0.68 | 224 |
| macro avg | 0.76 | 0.76 | 0.76 | 224 |
| weighted avg | 0.68 | 0.68 | 0.68 | 224 |



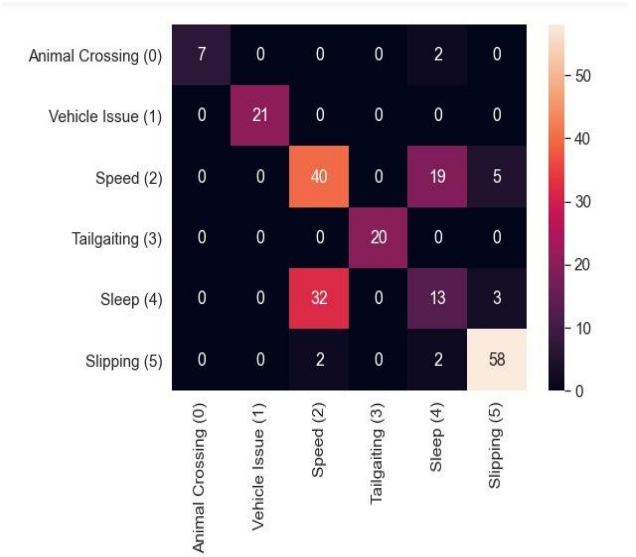
Appendix F: 5 Classification Report for Decision Tree Classifier 5

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 1.00 | 0.78 | 0.88 | 9 |
| 1 | 1.00 | 1.00 | 1.00 | 21 |
| 2 | 0.54 | 0.62 | 0.58 | 64 |
| 3 | 1.00 | 1.00 | 1.00 | 20 |
| 4 | 0.36 | 0.27 | 0.31 | 48 |
| 5 | 0.88 | 0.94 | 0.91 | 62 |
| accuracy | | | 0.71 | 224 |
| macro avg | 0.80 | 0.77 | 0.78 | 224 |
| weighted avg | 0.70 | 0.71 | 0.70 | 224 |



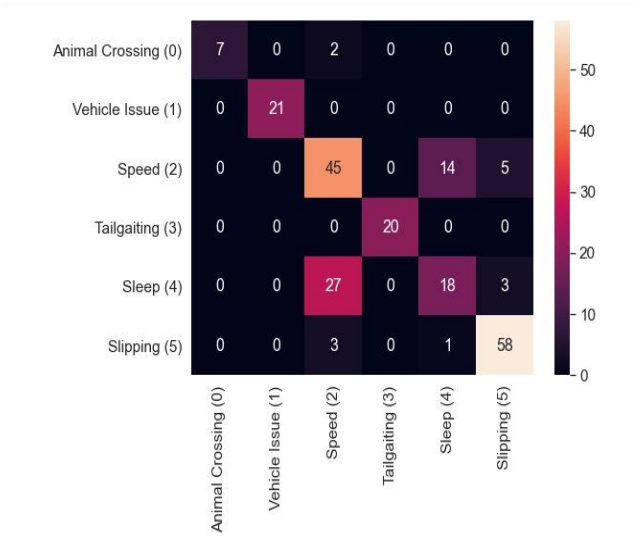

Appendix F: 6 Classification Report for Decision Tree Classifier 6

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 1.00 | 0.78 | 0.88 | 9 |
| 1 | 1.00 | 1.00 | 1.00 | 21 |
| 2 | 0.54 | 0.62 | 0.58 | 64 |
| 3 | 1.00 | 1.00 | 1.00 | 20 |
| 4 | 0.36 | 0.27 | 0.31 | 48 |
| 5 | 0.88 | 0.94 | 0.91 | 62 |
| accuracy | | | 0.71 | 224 |
| macro avg | 0.80 | 0.77 | 0.78 | 224 |
| weighted avg | 0.70 | 0.71 | 0.70 | 224 |



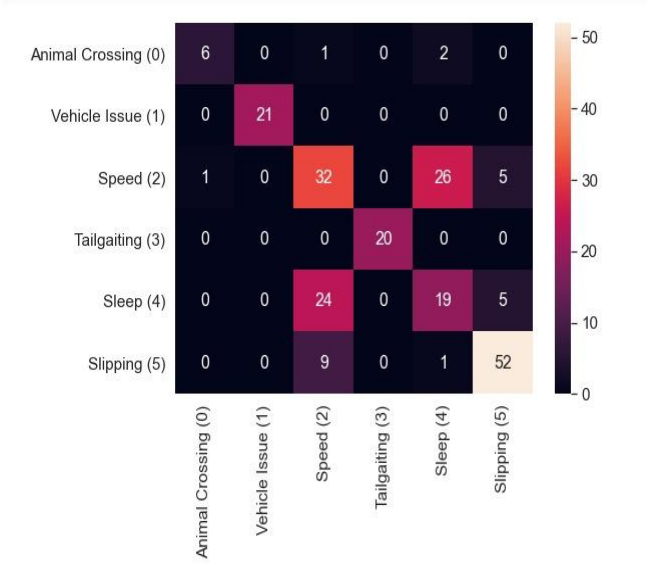
Appendix F: 7 Classification Report for Decision Tree Classifier 7

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 1.00 | 0.78 | 0.88 | 9 |
| 1 | 1.00 | 1.00 | 1.00 | 21 |
| 2 | 0.58 | 0.70 | 0.64 | 64 |
| 3 | 1.00 | 1.00 | 1.00 | 20 |
| 4 | 0.55 | 0.38 | 0.44 | 48 |
| 5 | 0.88 | 0.94 | 0.91 | 62 |
| accuracy | | | 0.75 | 224 |
| macro avg | 0.83 | 0.80 | 0.81 | 224 |
| weighted avg | 0.75 | 0.75 | 0.75 | 224 |




Appendix F: 8 Classification Report for Decision Tree Classifier 8

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.86 | 0.67 | 0.75 | 9 |
| 1 | 1.00 | 1.00 | 1.00 | 21 |
| 2 | 0.48 | 0.50 | 0.49 | 64 |
| 3 | 1.00 | 1.00 | 1.00 | 20 |
| 4 | 0.40 | 0.40 | 0.40 | 48 |
| 5 | 0.84 | 0.84 | 0.84 | 62 |
| accuracy | | | 0.67 | 224 |
| macro avg | 0.76 | 0.73 | 0.75 | 224 |
| weighted avg | 0.67 | 0.67 | 0.67 | 224 |

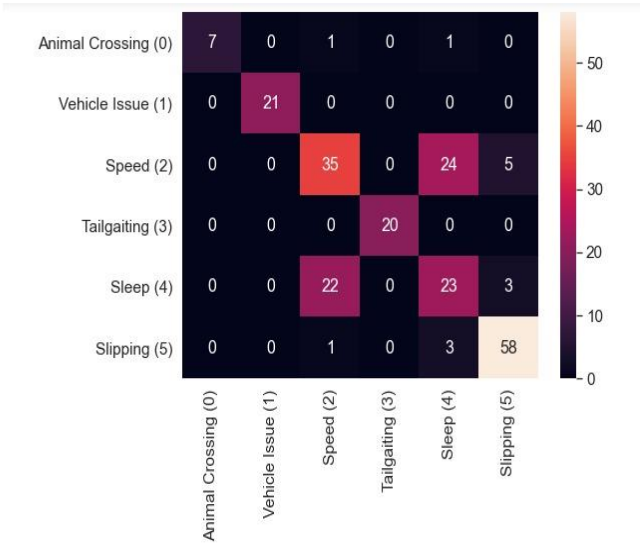


# Appendix - G

**SVM with OVR Classifier RHS**

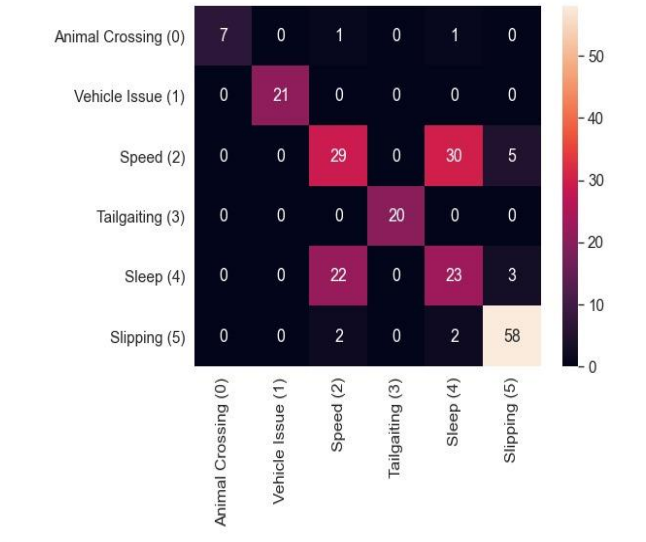
Appendix G: 1 Classification Report for SVM Classifier 1

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 1.00 | 0.78 | 0.88 | 9 |
| 1 | 1.00 | 1.00 | 1.00 | 21 |
| 2 | 0.59 | 0.55 | 0.57 | 64 |
| 3 | 1.00 | 1.00 | 1.00 | 20 |
| 4 | 0.45 | 0.48 | 0.46 | 48 |
| 5 | 0.88 | 0.94 | 0.91 | 62 |
| accuracy | | | 0.73 | 224 |
| macro avg | 0.82 | 0.79 | 0.80 | 224 |
| weighted avg | 0.73 | 0.73 | 0.73 | 224 |



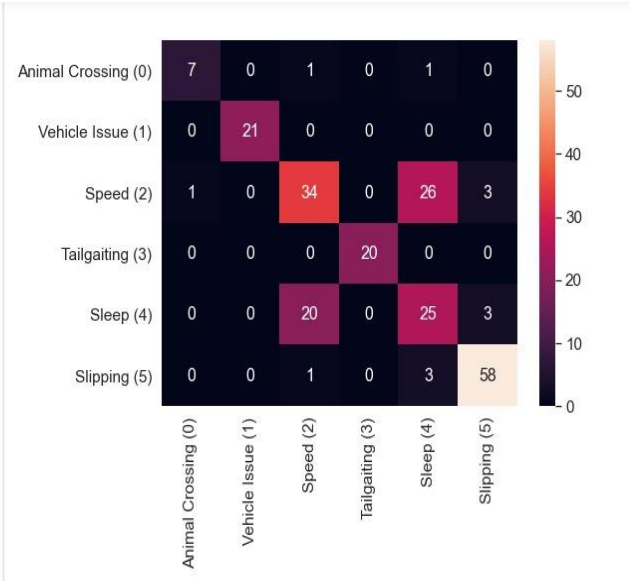

Appendix G: 2 Classification Report for SVM Classifier 2

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 1.00 | 0.78 | 0.88 | 9 |
| 1 | 1.00 | 1.00 | 1.00 | 21 |
| 2 | 0.54 | 0.45 | 0.49 | 64 |
| 3 | 1.00 | 1.00 | 1.00 | 20 |
| 4 | 0.41 | 0.48 | 0.44 | 48 |
| 5 | 0.88 | 0.94 | 0.91 | 62 |
| accuracy | | | 0.71 | 224 |
| macro avg | 0.80 | 0.77 | 0.79 | 224 |
| weighted avg | 0.71 | 0.71 | 0.70 | 224 |



Appendix G: 3 Classification Report for SVM Classifier 3

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.88 | 0.78 | 0.82 | 9 |
| 1 | 1.00 | 1.00 | 1.00 | 21 |
| 2 | 0.61 | 0.53 | 0.57 | 64 |
| 3 | 1.00 | 1.00 | 1.00 | 20 |
| 4 | 0.45 | 0.52 | 0.49 | 48 |
| 5 | 0.91 | 0.94 | 0.92 | 62 |
| accuracy | | | 0.74 | 224 |
| macro avg | 0.81 | 0.79 | 0.80 | 224 |
| weighted avg | 0.74 | 0.74 | 0.74 | 224 |

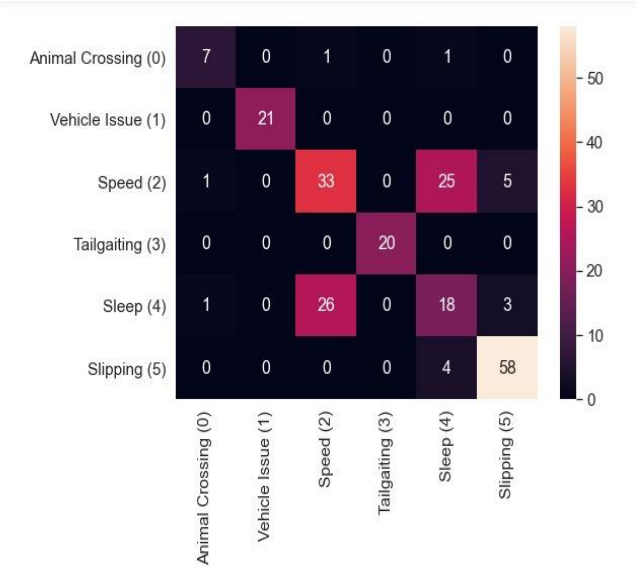


# Appendix - H

## LR with OVR Classifier RHS

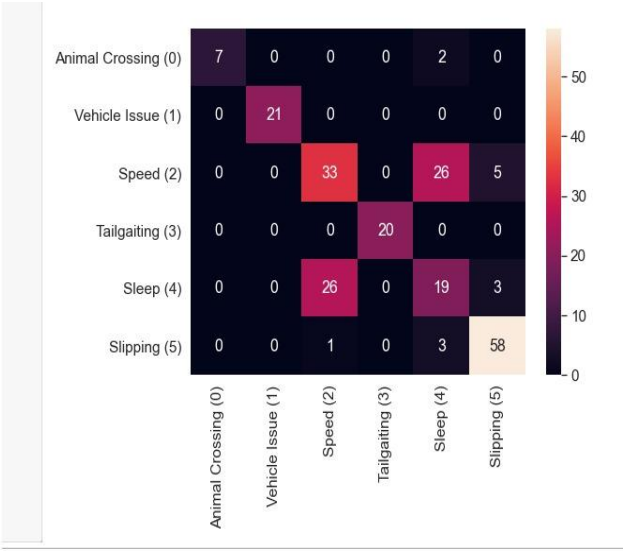
Appendix H: 1 Classification Report for LR Classifier 1

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.78 | 0.78 | 0.78 | 9 |
| 1 | 1.00 | 1.00 | 1.00 | 21 |
| 2 | 0.55 | 0.52 | 0.53 | 64 |
| 3 | 1.00 | 1.00 | 1.00 | 20 |
| 4 | 0.38 | 0.38 | 0.38 | 48 |
| 5 | 0.88 | 0.94 | 0.91 | 62 |
| accuracy | | | 0.70 | 224 |
| macro avg | 0.76 | 0.77 | 0.77 | 224 |
| weighted avg | 0.70 | 0.70 | 0.70 | 224 |



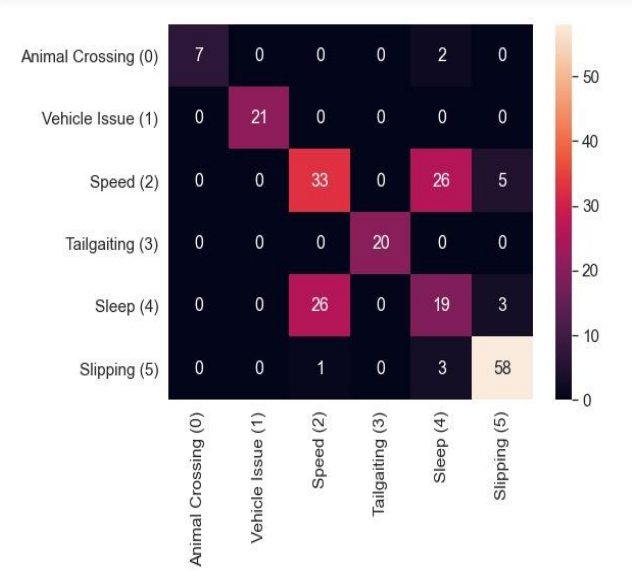

Appendix H: 2 Classification Report for LR Classifier 2

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 1.00 | 0.78 | 0.88 | 9 |
| 1 | 1.00 | 1.00 | 1.00 | 21 |
| 2 | 0.55 | 0.52 | 0.53 | 64 |
| 3 | 1.00 | 1.00 | 1.00 | 20 |
| 4 | 0.38 | 0.40 | 0.39 | 48 |
| 5 | 0.88 | 0.94 | 0.91 | 62 |
| accuracy | | | 0.71 | 224 |
| macro avg | 0.80 | 0.77 | 0.78 | 224 |
| weighted avg | 0.71 | 0.71 | 0.70 | 224 |



Appendix H: 3 Classification Report for SVM Classifier 3

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 1.00 | 0.78 | 0.88 | 9 |
| 1 | 1.00 | 1.00 | 1.00 | 21 |
| 2 | 0.55 | 0.52 | 0.53 | 64 |
| 3 | 1.00 | 1.00 | 1.00 | 20 |
| 4 | 0.38 | 0.40 | 0.39 | 48 |
| 5 | 0.88 | 0.94 | 0.91 | 62 |
| accuracy | | | 0.71 | 224 |
| macro avg | 0.80 | 0.77 | 0.78 | 224 |
| weighted avg | 0.71 | 0.71 | 0.70 | 224 |

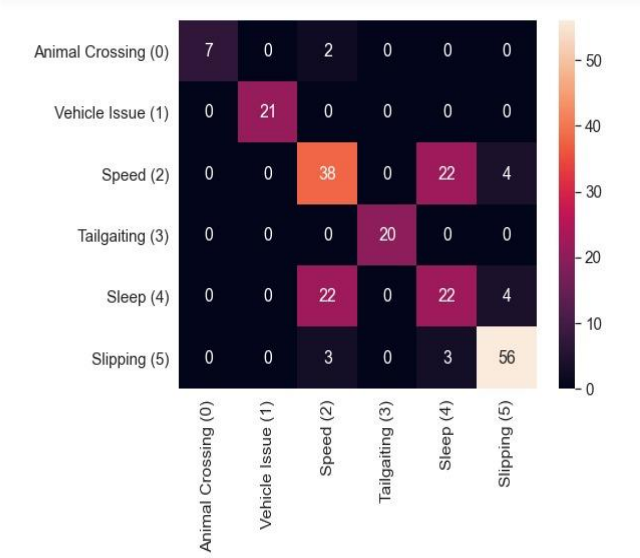


# Appendix - I

## Random Forest Classifier RHS

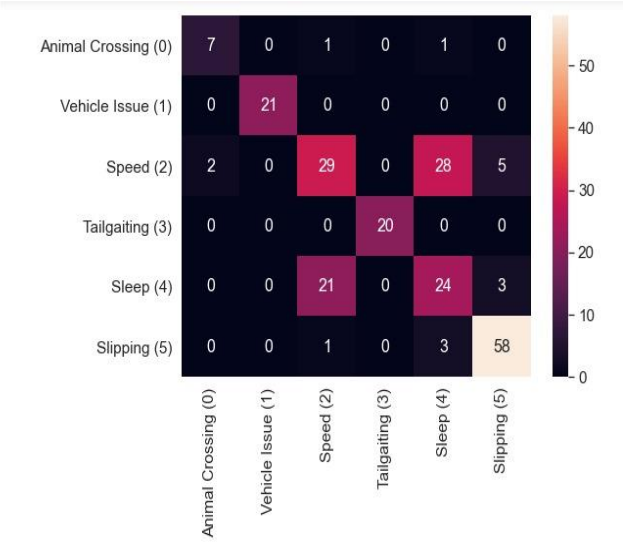
Appendix I: 1 Classification Report for Random Forest Classifier 1

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 1.00 | 0.78 | 0.88 | 9 |
| 1 | 1.00 | 1.00 | 1.00 | 21 |
| 2 | 0.58 | 0.59 | 0.59 | 64 |
| 3 | 1.00 | 1.00 | 1.00 | 20 |
| 4 | 0.47 | 0.46 | 0.46 | 48 |
| 5 | 0.88 | 0.90 | 0.89 | 62 |
| accuracy | | | 0.73 | 224 |
| macro avg | 0.82 | 0.79 | 0.80 | 224 |
| weighted avg | 0.73 | 0.73 | 0.73 | 224 |



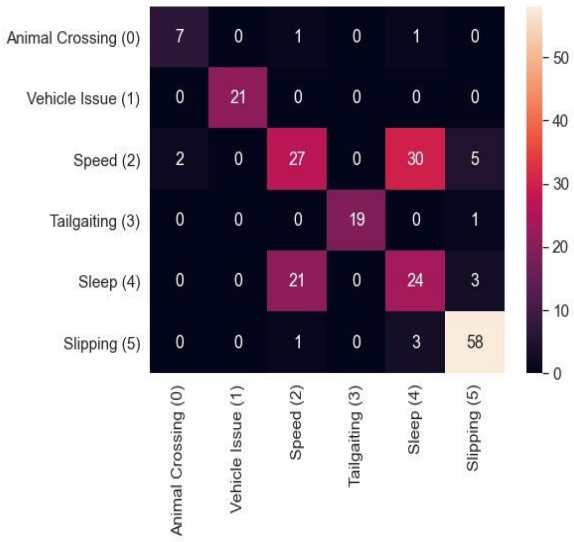

Appendix I: 2 Classification Report for Random Forest Classifier 2

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.78 | 0.78 | 0.78 | 9 |
| 1 | 1.00 | 1.00 | 1.00 | 21 |
| 2 | 0.56 | 0.45 | 0.50 | 64 |
| 3 | 1.00 | 1.00 | 1.00 | 20 |
| 4 | 0.43 | 0.50 | 0.46 | 48 |
| 5 | 0.88 | 0.94 | 0.91 | 62 |
| accuracy | | | 0.71 | 224 |
| macro avg | 0.77 | 0.78 | 0.77 | 224 |
| weighted avg | 0.71 | 0.71 | 0.71 | 224 |



Appendix I: 3 Classification Report for Random Forest Classifier 3

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.78 | 0.78 | 0.78 | 9 |
| 1 | 1.00 | 1.00 | 1.00 | 21 |
| 2 | 0.54 | 0.42 | 0.47 | 64 |
| 3 | 1.00 | 0.95 | 0.97 | 20 |
| 4 | 0.41 | 0.50 | 0.45 | 48 |
| 5 | 0.87 | 0.94 | 0.90 | 62 |
| accuracy | | | 0.70 | 224 |
| macro avg | 0.77 | 0.76 | 0.76 | 224 |
| weighted avg | 0.70 | 0.70 | 0.69 | 224 |

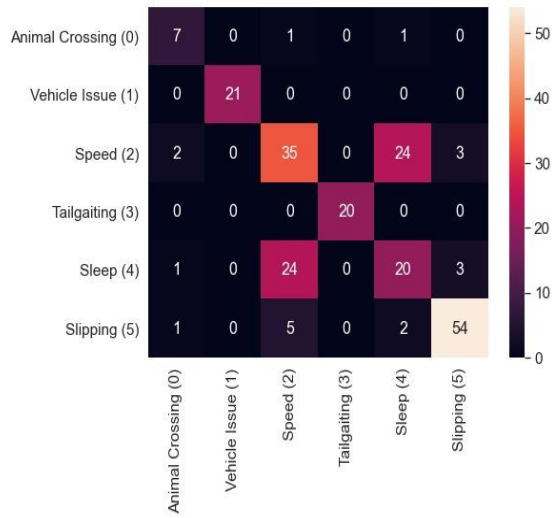


# Appendix - J

## Gradient Boosting Classifier RHS

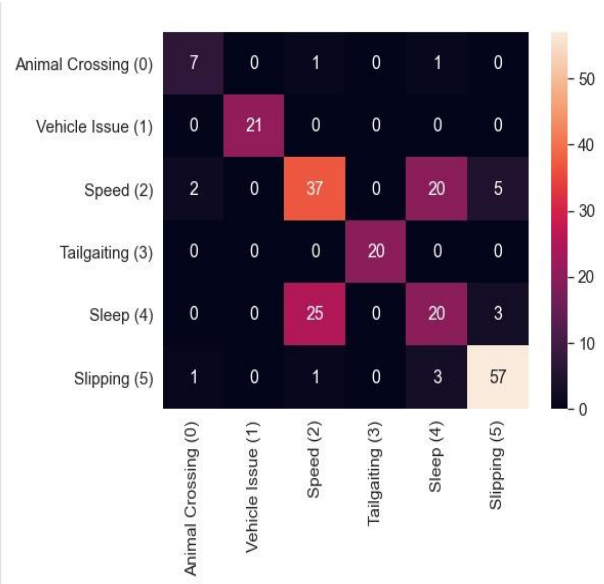
Appendix J: 1 Classification Report for Gradient Boost Classifier 1

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.64 | 0.78 | 0.70 | 9 |
| 1 | 1.00 | 1.00 | 1.00 | 21 |
| 2 | 0.54 | 0.55 | 0.54 | 64 |
| 3 | 1.00 | 1.00 | 1.00 | 20 |
| 4 | 0.43 | 0.42 | 0.42 | 48 |
| 5 | 0.90 | 0.87 | 0.89 | 62 |
| accuracy | | | 0.70 | 224 |
| macro avg | 0.75 | 0.77 | 0.76 | 224 |
| weighted avg | 0.70 | 0.70 | 0.70 | 224 |




Appendix J: 2 Classification Report for Gradient Boost Classifier 2

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.70 | 0.78 | 0.74 | 9 |
| 1 | 1.00 | 1.00 | 1.00 | 21 |
| 2 | 0.58 | 0.58 | 0.58 | 64 |
| 3 | 1.00 | 1.00 | 1.00 | 20 |
| 4 | 0.45 | 0.42 | 0.43 | 48 |
| 5 | 0.88 | 0.92 | 0.90 | 62 |
| accuracy | | | 0.72 | 224 |
| macro avg | 0.77 | 0.78 | 0.77 | 224 |
| weighted avg | 0.72 | 0.72 | 0.72 | 224 |



Appendix J: 3 Classification Report for Gradient Boost Classifier 3

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.78 | 0.78 | 0.78 | 9 |
| 1 | 1.00 | 1.00 | 1.00 | 21 |
| 2 | 0.59 | 0.56 | 0.58 | 64 |
| 3 | 1.00 | 1.00 | 1.00 | 20 |
| 4 | 0.54 | 0.54 | 0.54 | 48 |
| 5 | 0.83 | 0.87 | 0.85 | 62 |
| accuracy | | | 0.73 | 224 |
| macro avg | 0.79 | 0.79 | 0.79 | 224 |
| weighted avg | 0.73 | 0.73 | 0.73 | 224 |