Autonomous Vehicle Cost-Prediction-Based Decision-Making Framework For Unavoidable Collisions Using Ethical Foundations

Autonomous Vehicle Cost-Prediction-Based Decision-Making Framework For Unavoidable Collisions Using Ethical Foundations

By Fan Wu, B.Eng

A THESIS SUBMITTED TO
THE DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
AND THE SCHOOL OF GRADUATE STUDIES
OF MCMASTER UNIVERSITY
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
M.A.Sc.

McMaster University

M.A.Sc. (2020)

Hamilton, Ontario (Electrical and Computer Engineering)

TITLE: Autonomous Vehicle Cost-Prediction-Based Decision-Making Framework For Unavoidable Collisions Using Ethical Foundations

AUTHOR: Fan Wu (McMaster University)

SUPERVISOR: Dr. T. Kirubarajan

NUMBER OF PAGES: xii, 91

Abstract

*Autonomous Vehicle*s (*AV*s) hold out the promise of being safer than manually driven cars. However, it is impossible to guarantee the hundred percent avoidance of collisions in a real-life environment with unpredictable objects and events. When accidents become unavoidable, the different reactions of *AV*s and their outcome will have different consequences. Thus, *AV*s should incorporate the so-called 'ethical decision-making algorithm' when facing unavoidable accidents. This paper is introducing a novel cost-prediction-based decision-making framework incorporating two common ethical foundations human drivers use when facing unavoidable dilemma inducing collisions: Ethical Egoism and Utilitarianism. The cost-prediction algorithm consists of *Collision Injury Severity Level Prediction* (*CISLP*) and Cost Evaluation. The *CISLP* model was trained using both *Multinominal Logistic Regression* (*MLR*) and a *Decision Tree Classifier* (*DTC*). Both algorithms consider the combination of relationships among traffic collision explanatory features. Four different Cost Evaluation metrics are purposed and compared to suit different application needs. The Data set used for training and testing the cost prediction algorithm is the 1999-2017 *National Collision Data Base* (*NCDB*) which ensures the realistic and reliability of the algorithm. This paper is a novel paper using Canada's real traffic accident data to propose a cost-prediction-based decision- making framework incorporating different ethical foundations for *AV*s.

Acknowledgements

I would like to express my gratefulness and thanks to all the people that helped me during my graduate study. First and foremost, I would like to present my most sincere thanks to my supervisor Professor T. Kirubarajan, for everything he did to support my study and research throughout my entire graduate study, especially, he gave me the inspiration and ideas of this thesis paper. I really want to say thank you for your patience, time and care you provide me with during my graduate study.

Another person who influenced me and helped a lot is Dr. Ratnasingham Tharmarasa. I want to say thank you for all your patience and help with paper writing as well as helping me out with all the coding and debugging in my projects. All your requirements and advice in your course are really helpful for me when I'm writing this paper.

I would like to thank my family and my best Yafan Liu who supported me a lot with my daily life things so that I can concentrate more on my studies. My families have been going through a very difficult time when I am writing this paper due to the coronavirus pandemic at this very moment and I really hope all my family and friends can be healthy and safe. My dear friend Yafan Liu has always been very supportive of me and helped me a lot with taking care of my little sister to save me time and allow me to dedicate on my study.

Finally, I would like to thank all the administrative staff of the ECE Department. Studying at McMaster ECE for 7 years, all of you are my important friends. Thank Cheryl for always helping me out with different problems and being mentally supportive of my graduate study.

Contents

List of Figures

List of Tables

Acronyms

Acronyms

abs *Absolute*

AV *Autonomous Vehicle*

AVDM *Autonomous Vehicle Decision-Making*

CDMF *Collision Decision-Making Framework*

CISLP *Collision Injury Severity Level Prediction*

DI *dilemma inducing*

DM *decision-making*

DTC *Decision Tree Classifier*

EDMF *Ethical Decision-Making Foundations*

EF *ethical foundation*

EXP *Probability Based*

MLR *Multinominal Logistic Regression*

NCDB *National Collision Data Base*

V2V *vehicle-to-vehicle*

Chapter 1

Introduction

*Autonomous Vehicle*s (*AV*s) have embedded computer systems that assist human drivers by automating vehicles' control. NHTSA (National Highway Traffic Safety Administration) has defined six levels of vehicle automation [1] as shown in Table 1.1 and *AV*s in this paper refers to vehicles in automation Level 4 (High Automation) where the driver can cede full control to the vehicle in some situation and Level 5 (Full self-driving automation) where Vehicle can safely pilot itself for an entire trip, with no expectation for the driver to take control. This means even when facing emergency cases and *dilemma inducing* (*DI*) situations, *AV*s are required to make decisions and execute the controls themselves. I also assume *vehicle-to-vehicle* (*V2V*) communication exists among all vehicles which enable them to wirelessly exchange information about their occupants [2].

Table 1.1: SAE (J3016) Levels of Driving Automation

Automation Level	Society of Automotive Engineers(SAE) Defination	Execution of Control
No Automation (Level 0)	Zero Autonomy; the driver performs all driving tasks.	Human Driver
Driver Assistance (Level 1)	Vehicle is controlled by the driver, but some driving assist features may be included in the vehicle design.	Human Driver
Partial Automation (Level 2)	Vehicle has combined automated function, like acceleration and steering, but the driver must remain engaged with the driving task and monitor the environment at all times.	Human Driver, Several specific functions by Vehicle but only one function at a time in limited circumstances.
Conditional Automation (Level 3)	Driver is a necessity, but is not required to monitor the environment. The driver must be ready to take control of vehicle at all time with notice.	Multiple Functions at the same time by Vehicle, Human Driver attention needed at all times to take over control given notice.
High Automation (Level 4)	The vehicle is capable of performing all driving functions under certain conditions. The driver may have the option to control the vehicle.	Vehicle under several environment conditions
Full Automation (Level 5)	The vehicle is capable of performing all driving functions under all conditions. The driver may have the option to control the vehicle.	Vehicle under all conditions

*AV*s are expected to reduce the number of traffic accidents with all the advanced sensors and technology, however, even a perfectly functioning system cannot avoid every collision. In a scenario when human driver is unable to take control in time, a computer system needs to be responsible for pre-crash behavior even under *dilemma inducing* situations. At present, the research on *AV*s mostly address on-road vehicle automation and collision avoidance[3][4], little research has been published on collision *decision-making* for *AV*s incorporating different *ethical foundation*s like human drivers and use the collision injury severity level prediction to perform a cost evaluation under an emergency situation.

Regarding this problem, this thesis present three major contributions. These contributions address the weaknesses of the previous research related to each of the three aspects (*Autonomous Vehicle Decision-Making* (*AVDM*), *Collision Injury Severity Level Prediction* (*CISLP*), *Ethical Decision-Making Foundations* (*EDMF*)) in order to form a complete *Collision Decision-Making Framework* (*CDMF*) for *AV*s.

1.1 Autonomous Vehicle Decision-Making (*AVDM*)

Theoretical research robotics proved that a crash-free environment is unrealistic. Even with many pieces of literature proposing different techniques for collision avoidance in a dynamic environment[5], none are able to guarantee hundred percent avoidance of collisions in a real-life environment with unpredictable objects and events(e.g. hardware failure or limitation of vehicles due to curvature or icy road). In response, Fraichard and Asama defined "a state for which, no matter what the future trajectory of the system is, a collision with an obstacle occurs" as an inevitable collision state for mobile robots[6]. Many Traffic Literature has raised the concepts of incorporating human *ethical foundations* into *Autonomous Vehicle Decision-Making* [7][8]. A typical scenario represented in the literature by Greig[8]to highlight this emergency case is illustrated in Fig.2. In this scenario, the *AV* is the primary vehicle v_0 containing 1 occupant with a decision to be made among: 1. Steer right and go into the ditch, resulting in the primary vehicle rolling over; 2. Proceed straight, resulting in a rear-end collision with another vehicle V_1; 3. Steer left, resulting in a head-on collision with an oncoming vehicle V_3. Whatever decision made will result in a collision which means this is an unavoidable collision environment. There are several papers proposing different *DM* process for an *AV* under different but specific collision scenarios. In [9], Pickering and James introduced another similar scenario and an ethical Model-to-Decision (M2D) approach to deal with this specific scenario. In recent research done by Liao and Zhang[10], crash severity prediction with Support Vector Machine (SVM) was used for different emergency decisions using utilitarianism as the *ethical foundation* and deal with emergency cases for rear-end collisions only. Relevant research done in the past have many shortcomings: 1. Using a single *ethical foundation* only when making the decision. This neglects the complexity of the human *decision-making* process when facing *DI* situation. Thus, previous research is incapable of comparing the results induced by different decisions based on different *EF*s. 2. Models applicable to a single or a specific type of scenario of collision only, which cause limitation of the framework in general real-life application. 3. Unrealistic simulation data used for training the model and generating the possible outcomes. Thus, only limited inferences can be generalized from research to real-life.

In this paper, I addressed each of the above three shortcomings with the following methods: 1. Use both utilitarianism and ethical egoism as *ethical foundation* options available for the *decision-making* framework and compare the difference in resulting collision consequences. 2. Train the collision severity prediction model with almost all general real-life collision scenarios that enable the applicability of *decision-making* framework in different real-life environments. 3. Use Canada NCDB 1999-2017 data set[11] for model training. With this huge data set and temporal serial correlation considered, this research can be used to infer future real-life collision consequences of AVs using different ethical foundations based *decision-making* framework. The framework is based on *CISLP* and various collision cost evaluation metrics.

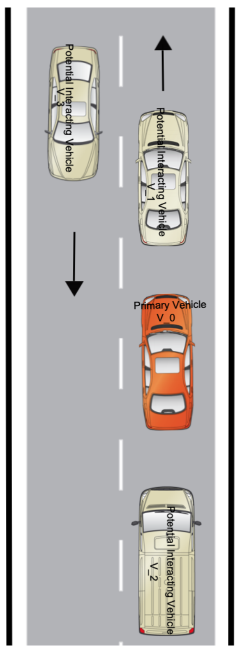

Figure 1.1: A Typical Dilemma Inducing Inevitable Collision Scenario

1.2 Collision Injury Severity Level Prediction (*CISLP*)

CISLP is an important step in the entire decision framework building. At present, studies use passenger characteristics (age, gender), accident characteristics (weather, time), road characteristics (curvature, slope), safety equipment, etc. as the explanatory of collision injury severity. There are in general two classes of models used for prediction: Statistical Models and Machine Learning Models[12]. A popular approach in the statistical model is *Multinominal Logistic Regression* (*MLR*)[12] which is also the most widely used discrete classification model and has a long history of use in crash severity literature. *Decision Tree Classifier* (*DTC*) is a classic Machine Learning model that has been used widely to analyze traffic data and predict collision outcomes[13][14]. There are also many other models used in both major classes. Some studies specifically focus on taking the heterogeneity and intrinsic relevance of the crash data into account and developed more advanced statistical models such as heteroskedastic ordered logit (HOL) model for single and multi-vehicle crash severity analysis by Lee and Li[15]. Shaheed developed a fully Bayesian hierarchical multinominal logit (BHML) model[16] to deal with intrinsic relevance in injury severity to analyze the factors which affect occupant injury severity in winter seasons. As for Machine Learning models, Support Vector Machines (SVM)[10], Recurrent Neural Network (RNN)[17], Artificial Neural Networks (ANN)[18] and Convolutional Neural Networks (CNN)[19] have all been used for *CISLP*. However, there are several common shortcomings or problems need to be strengthened and solved when doing *CISLP*. First, the prediction model needs to output the injury severity level result on a single victim or occupant level instead of the entire accident level which is not addressed in previous papers. Secondly, as the injury severity level is a multiclass and not binary outcome, the accuracy of the prediction over the entire test set is not a good evaluation metric to judge the usefulness of the model. More importantly, almost all the traffic accident data sets either for training or for testing are highly biased which means a high accuracy rate may only indicate classifying all the severity levels as the most common ones which provides no valuable prediction in reality. In real life, people are looking for a higher recall rate for more severe injury levels (Injured or fatal) which are in fact usually the rare cases in the imbalanced data set. Thirdly, many past papers set strict application conditions for their models. For instance: two-vehicle crashes of the urban road environment in the same lane and both vehicles are a standard passenger vehicle. With such a strict condition, I will not be able to embed the model in my *decision-making* framework and even the reported accuracy is not persuasive under more general and realistic environments. Fourthly, some papers used posterior collision factors for severity prediction instead of prior factors only. Even with a high accuracy and recall rate for all severity levels, this type of model is not applicable to realistic severity prediction as the post-collision factors such as: 'time for ambulance' is not available before the collision actually takes place. In my paper, I used a *MLR* model and a *DTC* model for per victim level collision severity prediction and all four concerns mentioned above are dealt with carefully in my model to ensure embedding the *CISLP* model in the *decision-making* framework will not affect framework's applicability in a general and realistic environment.

1.3 *Ethical Decision-Making Foundations* (*EDMF*)

In this paper, I linked *EF*s to *decision-making* framework for an *AV* in a *DI* situation when a collision is unavoidable and only disagreeable alternatives exist.[8] A primary vehicle confronting a *DI* collision adopts courses of action based on the user's predefined selection of one of the two ethical foundations: ethical egoism or utilitarianism. In this way, I can ensure *AV*s make a decision like human drivers who also have different ethical considerations facing collisions.

In conclusion, this paper provides a novel *DM* framework for *AV*s using two different *EF*s: ethical egoism and utilitarianism. The framework includes a *CISLP* model that is trained using both *MLR* and a *DTC*. A human drivers' driving experience and consequences at a collision can be characterized by the traffic environment data and vehicle occupant data. As all these data will remain unchanged for an *AV* facing an emergency case, a realistic *AV* severity injury prediction model is trained using the above data. A simulation will be run to generate new emergency environments and using the trained severity prediction model, different cost evaluation metrics and different *EF*s to make decisions. Results bases on all available options of the model will be compared but no optimal option will be made, instead, all options compared will be available for users to choose from and pre-install to *AV*s bases on their needs.

Chapter 2

Problem Formulation

The flowchart in Figure 2.1 provides an overview of the entire *decision-making* (DM) process for an AV in an inevitable collision environment using the Decision-Making-Framework proposed.

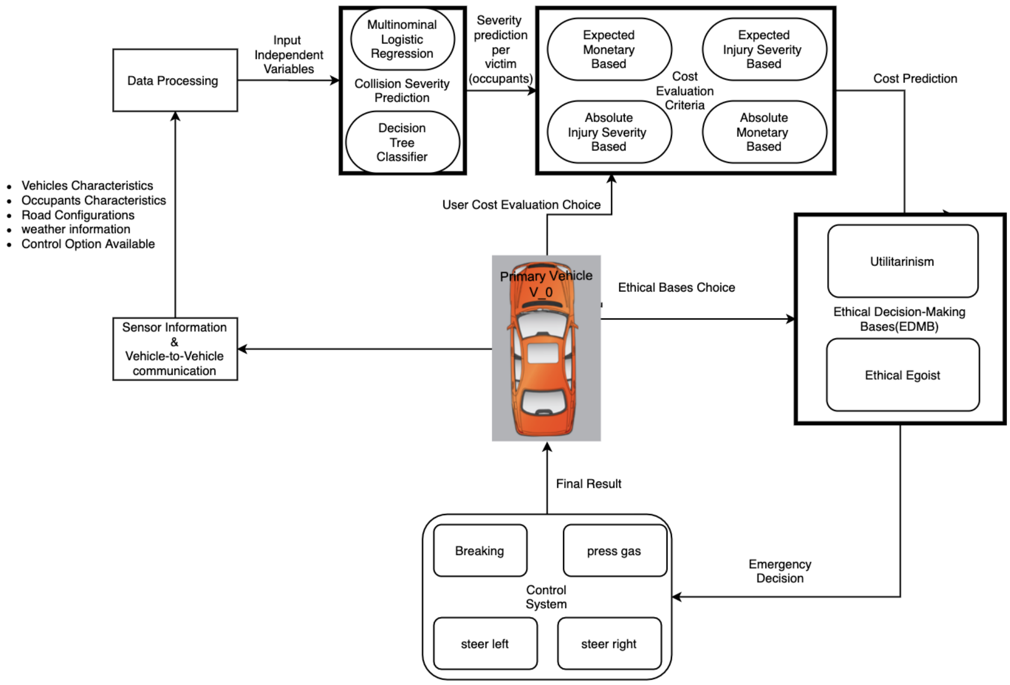

Figure 2.1: Decision Making Framework.

This thesis is base on the assumption that an *AV* in an unavoidable collision environment will always be able to collect sufficient attributes data from its sensor and V2V communication with the surrounding vehicles. And this study is limited to single-vehicle (including single vehicle collision with other obstacles) and two-vehicles collision only. The raw data will be processed by the internal-processor on the primary vehicle and then input into *CISLP* model, which is the collision severity prediction step in the flowchart. The *CISLP* model is a pre-trained model and in this thesis, *National Collision Data Base* (*NCDB*) 1999-2016 is used to train it. All the explanatory attributes in *NCDB* are macro environment attributes like weather, road surface configuration, vehicle year, victims' position. These macro environment attributes do not change between the transitioning from human driving to vehicle self-driving. This consistency allow us to use the past collected collision data from human driver accidents to build *CISLP* model and the entire decision-making framework and apply it to *AV*s. Many statistical and Machine Learning models can be selected as the *CISLP* model. In this thesis, *Multinominal Logistic Regression* (*MLR*) or a *Decision Tree Classifier* (*DTC*) is selected to be trained and perform the prediction function. These two models will lead to different results and these results are compared and analyzed for generalizing criteria for choosing a proper prediction model. The *CISLP* results on a per victim level are feed into the cost evaluation step. Four different cost evaluation metrics (criteria) as shown in the flowchart are used and compared in this paper to show their advantages and shortcomings when used for *DM*. After the cost is predicted, two different human *ethical foundations* (*EF*s): ethical egoism and utilitarianism are used to make the control decision which results in the final collision result.

In this paper, I mainly focus on the three parts highlighted in the flowchart: *CISLP* (Collision Severity Prediction in the flowchart), Cost Evaluation and Ethical Decision-Making. With novel combinations of theses three steps each completed with realistic data, together with a final simulation to present the different results based on different user choices in each step, this paper proposed a complete cost-prediction-based decision-making framework for autonomous vehicles using human ethical foundations.

Chapter 3

Methodology

3.1 Collision Injury Severity Level Prediction

In this paper, two prediction methods, also known as classification methods, were used for CISLP.

3.1.1 *Multinominal Logistic Regression* (*MLR*)

MLR is a classification method that generalizes logistic regression to multiclass problems. This model is used to predict the probabilities of the different possible outcomes of a categorically distributed dependent variable, given a set of independent variables, which may be real-valued, binary-valued, categorical valued.[20][21]

Assumptions

(1) Independent variables are case-specific: each independent variable has a single value for each case.

(2) The dependent variable cannot be perfectly predicted from the independent variables for any case.

(3) No need for the independent variables to be statistically independent of each other

and does not assume normality, linearity, or homoscedasticity.

(4) Collinearity is assumed to be relatively low, as it becomes difficult to differentiate between the impact of several variables if this is not the case. In this regression, Collinearity will be dealt using regularization techniques to add a penalty to model parameters (all except intercepts) so the model generalizes the data instead of overfitting (a side effect of multicollinearity)[22].

MLR **model**

The general problem of *MLR* can be generalized as below:

Data:

$(\mathbf{x_i}, \mathbf{y_i})_{i=1}^{n}$ for n observations

$\mathbf{x_i} \in \mathbb{R}^{n\times m}$ indicating m predictor variables for an single observation

$\mathbf{y_i} \in 1, .., K$ indicating K categories of outcome.

Probability Model:

$$\mathrm{p}_{i,k} = \mathrm{P}\{y_i = k|\mathbf{x_i}\}, k = 1, ..., K \tag{3.1}$$

$\sum_{k=1}^{K} \mathrm{p}_k = 1$, K-1 degrees of freedom

Label Assignment Rule:

$$\hat{y}_i|x_i = \underset{k}{\operatorname{argmax}}\ \hat{\mathrm{p}}_{i,k} \tag{3.2}$$

There are two common ways of interpreting the *MLR* model:

(1) As a set of independent binary regressions

(2) As a log-linear model

These two interpretations will eventually become equivalent after reformatting.

(1) As a set of independent binary regressions

There are in total $k+1$ severity levels $Y = (y_1, ..., y_i, ..., y_{k+1})$

with $y_i = 0$ for all i besides one j with $y_j = 1$ and corresponding probability p_j implying

$$\mathbf{E}\mathbf{Y} = \mathbf{p} \tag{3.3}$$

$$\mathrm{Cov}\mathbf{Y} = \Lambda_p - \mathbf{p}\mathbf{p}^{\mathrm{T}} \tag{3.4}$$

$$\Lambda_p = \begin{bmatrix} p_1 & & 0 \\ 0 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & & p_{k+1} \end{bmatrix} \tag{3.5}$$

The multinomial logit-model is given by

$$\mathrm{p}_i = \frac{\exp(\pi^{(\mathrm{i})^{\mathrm{T}}}\mathbf{x})}{1 + \sum_{j=1}^{k} \exp(\pi^{(j)^{\mathrm{T}}}\mathbf{x})} \quad for \quad i = 1, ..., k \tag{3.6}$$

$$\mathrm{p}_{k+1} = \frac{1}{1 + \sum_{j=1}^{k} \exp(\pi^{(j)^{\mathrm{T}}}\mathbf{x})} \tag{3.7}$$

where $\mathbf{x} = (x_1, ..., x_m)^{\mathrm{T}}$ is the vector of covariates, also called predictor variables or explanatory variables , and $\pi^{(i)}$ is the parameter vector corresponding with the $i-th$ response category, which will be the $i-th$ severity level in this study.

For clarification, corresponding probability of resulting in each collision severity for a single victim are

$$\Pr(y_1 = 1|\mathbf{x}) = \frac{\exp(\pi^{(1)^{\mathrm{T}}}\mathbf{x})}{1 + \exp(\pi^{(1)^{\mathrm{T}}}\mathbf{x}) + \exp(\pi^{(2)^{\mathrm{T}}}\mathbf{x})}$$

$$\Pr(y_2 = 1|\mathbf{x}) = \frac{\exp(\pi^{(2)^{\mathrm{T}}}\mathbf{x})}{1 + \exp(\pi^{(1)^{\mathrm{T}}}\mathbf{x}) + \exp(\pi^{(2)^{\mathrm{T}}}\mathbf{x})}$$

$$\Pr(y_3 = 1|\mathbf{x}) = \frac{1}{1 + \exp(\pi^{(1)^{\mathrm{T}}}\mathbf{x}) + \exp(\pi^{(2)^{\mathrm{T}}}\mathbf{x})}$$

where $\pi^{(i)} = (\pi_1^{(i)}, \pi_2^{(i)}, ..., \pi_m^{(i)})^{\mathrm{T}}$ for $i = 1, 2$

(2) As a log--linear model

I modeled the logarithm of the probability of seeing a given output using the linear predictor as well as an additional normalization factor. The formula given below will correspond to the prediction of severity levels directly.

$$ln\Pr(y_1 = 1|\mathbf{x}) = \pi^{(1)^{\mathrm{T}}}\mathbf{x} - lnZ$$

$$ln\Pr(y_2 = 1|\mathbf{x}) = \pi^{(2)^{\mathrm{T}}}\mathbf{x} - lnZ$$

$$ln\Pr(y_3 = 1|\mathbf{x}) = \pi^{(3)^{\mathrm{T}}}\mathbf{x} - lnZ$$

lnZ ensures that the whole set of probabilities forms a probability distribution, which means they all sum to 1. We need to add a term to ensure normalization, rather than multiply as usual, because we have taken the logarithm of the probabilities. Take the exponential on both sides turns the additive term into a multiplicative factor so that the probability is:

$$\Pr(y_1 = 1|\mathbf{x}) = \frac{1}{Z}\exp(\pi^{(1)^{\mathrm{T}}}\mathbf{x})$$

$$\Pr(y_2 = 1|\mathbf{x}) = \frac{1}{Z}\exp(\pi^{(2)^{\mathrm{T}}}\mathbf{x})$$

$$\Pr(y_3 = 1|\mathbf{x}) = \frac{1}{Z}\exp(\pi^{(3)^{\mathrm{T}}}\mathbf{x})$$

$$\Pr(y_1 = 1|\mathbf{x}) + \Pr(y_2 = 1|\mathbf{x}) + \Pr(y_3 = 1|\mathbf{x}) = 1$$

As a result, the solve for Z is

$$Z = \sum_{j=1}^{3} \exp({\pi^{(j)}}^{\mathrm{T}} \mathbf{x})$$

Then, generally

$$\Pr(y_i = 1|\mathbf{x}) = \frac{\exp({\pi^{(i)}}^{\mathrm{T}} \mathbf{x})}{\exp({\pi^{(1)}}^{\mathrm{T}} \mathbf{x}) + \exp({\pi^{(2)}}^{\mathrm{T}} \mathbf{x}) + \exp({\pi^{(3)}}^{\mathrm{T}} \mathbf{x})}$$

All of the $\pi^{(j)}, j = 1, 2, 3$ vectors of coefficients are uniquely identifiable as probabilities must sum to 1. This results in one of them completely determined once all the rest are known. Thus there are only 2 separately specifiable probabilities, and hence 2 separately identifiable vectors of coefficients. If we add a constant vector C to all of the coefficient vectors, the equations remain the same, left-hand side equals right-hand side of the below equation:

$$\frac{\exp({\pi^{(i)}}^{\mathrm{T}} \mathbf{x})}{\exp({\pi^{(1)}}^{\mathrm{T}} \mathbf{x}) + \exp({\pi^{(2)}}^{\mathrm{T}} \mathbf{x}) + \exp({\pi^{(3)}}^{\mathrm{T}} \mathbf{x})} =$$

$$\frac{\exp(({\pi^{(i)}}^{\mathrm{T}} + \mathbf{C})\mathbf{x})}{\exp(({\pi^{(1)}}^{\mathrm{T}} + \mathbf{C})\mathbf{x}) + \exp(({\pi^{(2)}}^{\mathrm{T}} + \mathbf{C})\mathbf{x}) + \exp(({\pi^{(3)}}^{\mathrm{T}} + \mathbf{C})\mathbf{x})}$$

if we set $\mathbf{C} = -{\pi^{(i)}}^{\mathrm{T}}$ for instance $\mathbf{C} = -{\pi^{(3)}}^{\mathrm{T}}$, then we get:

$${\pi'^{(1)}}^{\mathrm{T}} = {\pi^{(1)}}^{\mathrm{T}} - {\pi^{(3)}}^{\mathrm{T}}$$

$${\pi'^{(2)}}^{\mathrm{T}} = {\pi^{(2)}}^{\mathrm{T}} - {\pi^{(3)}}^{\mathrm{T}}$$

$${\pi'^{(3)}}^{\mathrm{T}} = {\pi^{(3)}}^{\mathrm{T}} - {\pi^{(3)}}^{\mathrm{T}} = 0$$

Now we get the following new equations:

$$\Pr(y_i = 1|\mathbf{x}) = \frac{\exp({\pi'^{(1)}}^{\mathrm{T}} \mathbf{x})}{\exp({\pi'^{(1)}}^{\mathrm{T}} \mathbf{x}) + \exp({\pi'^{(2)}}^{\mathrm{T}} \mathbf{x}) + 1}$$

$$\Pr(y_i = 2|\mathbf{x}) = \frac{\exp({\pi'^{(2)}}^{\mathrm{T}}\mathbf{x})}{\exp({\pi'^{(1)}}^{\mathrm{T}}\mathbf{x}) + \exp({\pi'^{(2)}}^{\mathrm{T}}\mathbf{x}) + 1}$$

$$\Pr(y_i = 3|\mathbf{x}) = \frac{1}{\exp({\pi'^{(1)}}^{\mathrm{T}}\mathbf{x}) + \exp({\pi'^{(2)}}^{\mathrm{T}}\mathbf{x}) + 1}$$

Except for symbols differences on the regression coefficients ${\pi^{(i)}}^{\mathrm{T}}$ and ${\pi'^{(1)}}^{\mathrm{T}}$, this is exactly the same as the form of (1)

Estimating the coefficients

The unknown parameters in each vector $\pi^{(i)}$ can be estimated by Maximum Likelihood Estimation (MLE) and using regularization of the weights to prevent pathological solutions. lasso (least absolute shrinkage and selection operator, 'L1'), Ridge ('L2') are the most commonly used regularization techniques. Here we will use x_i to denote the explanatory factors in the observation i and $y_{i,k}$ to denote the outcome in the observation i being category k.[23][24]

Multinominal Likelihood:

$$Lik \propto \mathrm{p}_{i,1}^0 \times ... \times \mathrm{p}_{i,1}^1 \times ... \times \mathrm{p}_{i,K}^0 = \mathrm{p}_{i,k} \tag{3.8}$$

Log Likelihood:

$$l_i = \log \mathrm{p}_{i,k}, \quad \text{if } y_i = k \tag{3.9}$$

Total Log-Likelihood in a double summation form:

$$\mathrm{p}(\pi) = \sum_{i=1}^{n} l_i = \sum_{i=1}^{n} \{\sum_{k=1}^{K} r_{i,k} \times \log p_{i,k}\} \tag{3.10}$$

with indicator function:

$$r_{i,k} = \begin{cases} 1 & \text{if } y_i = k \\ 0 & \text{otherwise} \end{cases}$$

If the training data are imbalanced for each class k, a sample weight s_k can be applied to balance out the effect of imbalanced data distribution. If no sample weight is applied, s_k by default equals 1. The indicator function will be as follow

$$r_{i,k} = \begin{cases} 1 * s_k & \text{if } y_i = k \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

$$\mathrm{p}_{i,k} = \Pr(y_{i,k} = 1 | x_i; \pi^{(k)^{\mathrm{T}}}) = \frac{\exp(\pi^{(k)^{\mathrm{T}}} x_i)}{\sum_{j=1}^{K} \exp(\pi^{(j)^{\mathrm{T}}} x_i)}$$

Here $\pi^{(1)}, \pi^{(2)}, ...\pi^{(K)} \in \mathbb{R}^m$ are the coefficients of the model. And the term $\frac{1}{\sum_{j=1}^{K} \exp(\pi^{(j)^{\mathrm{T}}} x_i)}$ normalizes the distribution so that it sums to 1. For convenience, I will write π to denote all the coefficients of the model where π is a m-by-K matrix obtained by concatenating $\pi^{(1)}, \pi^{(2)}, ..., \pi^{(K)}$, so that

$$\pi = \begin{bmatrix} | & | & | & | \\ \pi^{(1)} & \pi^{(2)} & \dots & \pi^{(K)} \\ | & | & | & | \end{bmatrix} \tag{3.12}$$

Cost Function used Cross-entropy (log) Loss in *MLR*. Cross-entropy loss measures the performance of classification in terms of the different output categories probability value between 0 and 1. Cross-Entroy increases as the predicted probability diverge from the actual label. Equations need to be solved using optimization solver are as below for n observations:

$$\mathrm{J}(\pi) = -\frac{1}{n}[\sum_{i=1}^{n}\{\sum_{k=1}^{K} r_{i,k} \log \frac{\exp(\pi^{(k)^{\mathrm{T}}} x_i)}{\sum_{j=1}^{K} \exp(\pi^{(j)^{\mathrm{T}}} x_i)}\}] \tag{3.13}$$

we will need to use optimization algorithm to solve for the minimum $J(\pi)$:

$$\min_{\pi} \; -\frac{1}{n}\sum_{i=1}^{n}\{\sum_{k=1}^{K} r_{i,k} \log \frac{\exp(\pi^{(k)^{\mathrm{T}}} x_i)}{\sum_{j=1}^{K} \exp(\pi^{(j)^{\mathrm{T}}} x_i)}\} \tag{3.14}$$

$$r_{i,k} = \begin{cases} s_k & \text{if } y_i = k \\ 0 & \text{otherwise} \end{cases}$$

Regularization of the Cost function (K categories and m explanatory factors): Instead of optimizing the above cost function directly, with regularization, I added a constraint on how big the coefficients can get in order to prevent overfitting. L1 (lasso) and L2 (Ridge) adapt different ways of setting upper bounds of coefficients, which determines that L1 has the ability to do feature selection by making coefficients 0 for less important features and mitigate the issue of multicollinearity, while L2 also penalizes very large coefficients but does not make any to 0. There also exists a parameter that controls the weight of the constraint, λ, so that coefficients won't be punished too hard resulting in underfitting [25]. In below equations, n = number of observations, K = numbers of output categories, m = number of explanatory variables.

1. With L_1 lasso Regularization

$$\min_{\pi} \; -\frac{1}{n}\sum_{i=1}^{n}\{\sum_{k=1}^{K} r_{i,k}\log\frac{\exp(\pi^{(k)^{\mathrm{T}}}x_i)}{\sum_{j=1}^{K}\exp(\pi^{(j)^{\mathrm{T}}}x_i)}\} \tag{3.15}$$

$$r_{i,k} = \begin{cases} s_k & \text{if } y_i = k \\ 0 & \text{otherwise} \end{cases}$$

$$\text{subject to } \|\pi\|_1 \leq C \tag{3.16}$$

The above can be convert to

$$\min_{\pi,\lambda} \; -\frac{1}{n}\sum_{i=1}^{n}\{\sum_{k=1}^{K} r_{i,k}\log\frac{\exp(\pi^{(k)^{\mathrm{T}}}x_i)}{\sum_{j=1}^{K}\exp(\pi^{(j)^{\mathrm{T}}}x_i)}\} + \frac{\lambda}{n}\sum_{j=1}^{m}|\pi_j| \tag{3.17}$$

$$r_{i,k} = \begin{cases} s_k & \text{if } y_i = k \\ 0 & \text{otherwise} \end{cases}$$

2. With L_2 Regularization

$$\min_{\pi} \quad -\frac{1}{n}\sum_{i=1}^{n}\{\sum_{k=1}^{K} r_{i,k}\log\frac{\exp(\pi^{(k)^{\mathrm{T}}}x_i)}{\sum_{j=1}^{K}\exp(\pi^{(j)^{\mathrm{T}}}x_i)}\} \tag{3.18}$$

$$r_{i,k} = \begin{cases} s_k & \text{if } y_i = k \\ 0 & \text{otherwise} \end{cases}$$

$$\text{subject to } \|\pi\|_2^2 \leq C \tag{3.19}$$

The above can be convert to

$$\min_{\pi,\lambda} \quad -\frac{1}{n}\sum_{i=1}^{n}\{\sum_{k=1}^{K} r_{i,k}\log\frac{\exp(\pi^{(k)^{\mathrm{T}}}x_i)}{\sum_{j=1}^{K}\exp(\pi^{(j)^{\mathrm{T}}}x_i)}\} + \frac{\lambda}{2n}\sum_{j=1}^{m}\pi_j^2 \tag{3.20}$$

$$r_{i,k} = \begin{cases} s_k & \text{if } y_i = k \\ 0 & \text{otherwise} \end{cases}$$

Finding Solution of coefficients

The solution is typically found using an iterative procedure. In the scikit-learn library I used to build the classifier model, there are several solvers to choose from. However, SAGA solver[26] is the only one supporting *MLR* with either L1 or L2 Regularization methods. SAGA is an optimization method with good convergence rates and has support for composite objectives (regularization function) where a proximal operator is used on the regulariser. SAGA supports non-strongly convex problems directly and is adaptive to any inherent strongly convexity of the problem. SAGA is an incremental gradient method that avoids the addition of a tunable parameter which is always used by other incremental gradient methods when applied to the non-strongly convex problem.

In particular, we are interested in minimising functions of the form

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$$

where $x \in \mathbb{R}^d$, each f_i is convex and has Lipschitz continuous derivatives with constant L. The 'Composite' (or proximal) case where an regularization function is added:

$$F(x) = f(x) + h(x)$$

where $h : \mathbb{R}^d \to \mathbb{R}^d$ is convex but potentially non-differentiable, and where the proximal operation of h is easy to compute.

Brief walk-through of SAGA algorithm

Start with known initial vector $x^0 \in \mathbb{R}^d$ and known derivatives $f_i^{'}(\phi_i^0) \in \mathbb{R}^d$ with $(\phi_i^0) = x^0$ for each i(iterate). These derivatives are stored in a table data-structure of length n, or a $n \times d$ matrix. SAGA uses a step size of γ and makes the following updates, starting with $k = 0$: Given the value of x^k and of each $f_i^{'}(\phi_i^k)$ at the end of iteration k, the updates for iteration $k + 1$ is as follows:

1. Pick a j uniformly at random.
2. Take $\phi_j^{k+1} = x^k$, and store $f_j^{'}(\phi_j^{k+1})$ in the table. All other entries in the table remain unchanged. The quantity ϕ_j^{k+1} is not explicitly stored.
3. Update x using $f_j^{'}(\phi_j^{k+1})$, $f_j^{'}(\phi_j^k)$ and the table average:

$$w^{k+1} = x^k - \gamma[f_j^{'}(\phi_j^{k+1}) - f_j^{'}(\phi_j^k) + \frac{1}{n}\sum_{i=1}^{n} f_i^{'}(\phi_i^k)], \tag{3.21}$$

$$x^{k+1} = prox_{\gamma}^{h}(w^{k+1}) \tag{3.22}$$

The proximal operator above is defined as:

$$prox_{\gamma}^{h}(y) := \underset{x \in \mathbb{R}^d}{\operatorname{argmin}} \left\{ h(x) + \frac{1}{2\gamma} \|x - y\|^2 \right\} \tag{3.23}$$

The coefficients π in MLR are solved using this SAGA optimization algorithm.

3.1.2 *Decision Tree Classifier* (*DTC*)

DTC is a distribution-free and non-parametric supervised learning method that does not depend on probability distribution assumptions and can create a model to predict the value of a target output by learning simple decision rules learned from the training data. DT is simple to understand and interpret and can be visualized. It is able to handle multi-class output classification problems with little data preparation and able to handle both numerical and categorical data. DTC is a white box model as if a given situation is observable in a model, the explanation for the condition is easily explained by Boolean logic. It is also possible to validate the model using a statistical test, which makes it possible to justify the reliability of the model.

Decision Tree Algorithm

Decision Tree is a flowchart-like structure that uses internal nodes to represent a test on an attribute, branches to represent an outcome of the test and each leaf node (terminal node) represents a class label outcome. Decision Tree learns to partition data set based on attribute values recursively which is called a recursive partition.[27]

Mathematical Formulation [28]:

Given training vectors $x_i \in R^k$, $i = 1, ..., l$ and a label vector $y \in R^l$, a decision tree recursively partitions the space such that the samples with the same labels are grouped together. Let the data at node m be represented by T. For each candidate split $\theta = (a, t_m)$ consisting of a feature a and threshold t_m, partition the data into $T_{\text{left}}(\theta)$ and $T_{\text{right}}(\theta)$ subsets.

$$\begin{aligned} T_{\text{left}}(\theta) &= (x, y) | x_j <= t_m \\ T_{\text{right}}(\theta) &= T \setminus T_{\text{left}}(\theta) \end{aligned} \tag{3.24}$$

The impurity at m is computed using a impurity function (Attribute Selection Measures) H(). When the task is classification, Information Gain and Gini impurity are popular choices of H().

$$\mathrm{G}(T, \theta) = \frac{n_{\text{left}}}{N_m} \mathrm{H}(T_{\text{left}}(\theta)) + \frac{n_{\text{right}}}{N_m} \mathrm{H}(T_{\text{right}}(\theta)) \tag{3.25}$$

Select the parameters that minimises the weighted sum of impurity function $\mathrm{G}(T, \theta)$:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \ \mathrm{G}(T, \theta) \tag{3.26}$$

Recurse for subsets $T_{left}(\theta^*)$ and $T_{\text{right}}(\theta^*)$ until the maximum allowable depth is reached, $N_m < \min_{samples}$ (N_m is the number of observations in the subset and $\min_{samples}$ is a hyperparameter to the tree) or $N_m = 1$.

Attribute Selection Measures (ASM)

ASM is also known as the splitting rule used to study the quality and select the attributes that partition the tuples into distinct classes. It is the way of determining of splitting criterion, which partition data in the best manner. ASM rank each feature by using it to classify (or regression) the given dataset and choose the best split attribute to be the final splitting attribute. The most popular selection measures are Information Gain and Gini impurity.

1. Information Gain (IG)[29]

   This measure used the concept of entropy, which measures the impurity or randomness in a group of examples. Information gain is defined as the difference between entropy before split and average (expected) entropy after split of the data set based on given attribute values. Let T denote a set of training data, each of the form $(\mathbf{x}, y) = (x_1, x_2, ..., x_k, y)$ where $x_a \in vals(a)$ is the value of the a^{th} attribute of example $\mathbf{x}$ and y is the corresponding class label. The information gain for an attribute a is defined in terms of Shannon entropy H() as follows. For a value v of attribute a, define $S_a(v)$ as the set of training inputs of T whose attribute a equals v:

$$S_a(v) = \{\mathbf{x} \in T | x_a = v\} \tag{3.27}$$

The information gain for an attribute a is the difference between the a prior Shannon entropy $\mathrm{H}(T)$ of the training set and the conditional entropy $\mathrm{H}(T|a)$ of T given the value of attribute a.

$$\mathrm{IG}(T, a) = \mathrm{H}(T) - \mathrm{H}(T|a). \tag{3.28}$$

In decision tree, $\mathrm{H}(T)$ is the Entropy of the parent node, $\mathrm{H}(T|a)$ is the expected entropy of the children node which equals to the conditional entropy of T given the value of attribute a. Entropy of the parent node is defined as

$$\mathrm{H}(T) = \mathrm{I}_E\left(\mathrm{p}_1, \mathrm{p}_2, ..., \mathrm{p}_J\right) = -\sum_{i=1}^{J} \mathrm{p}_i \log_2 \mathrm{p}_i \tag{3.29}$$

where $p_1, p_2, ...$ adds up to 1 and represents the percentage of each class presents in the child node that results from a split in the tree.

$$\mathrm{H}(T|a) = \sum_{v \in vals(a)} \mathrm{P}_a(v) \cdot \mathrm{H}\left(S_a(v)\right) \tag{3.30}$$

In particular, the values $v \in vals(a)$ defines a partition of the training set T into mutually exclusive and all-inclusive subsets, inducing a categorical probability distribution $P_a(v)$ on the values $v \in vals(a)$ of attribute a. The distribution is given as

$$\mathrm{P}_a(v) := \frac{|S_a(v)|}{|T|} \tag{3.31}$$

Entropy of children nodes using attribute a for splitting is defined as

$$\mathrm{H}(T|a) = \sum_{v \in vals(a)} \mathrm{P}_a(v) \sum_{i=1}^{J} -[\Pr(i|a) \log_2 \Pr(i|a)] \tag{3.32}$$

The information gain of splitting T with J ouptut categories using attribute a is defined as

$$\mathrm{IG}(T, a) = -\sum_{i=1}^{J} \mathrm{p}_i \log_2 p_i - \sum_{v \in vals(a)} \mathrm{P}_a(v) \sum_{i=1}^{J} -[\Pr(i|a) \log_2 \Pr(i|a)] \tag{3.33}$$

2. Gini impurity

   Gini impurity measures the probability of a randomly chosen element from the set would be incorrectly classified if it was randomly classified according to the distribution of output labels in the subset. The value of Gini impurity varies between 0 and 1, where 0 denotes that all elements belong to a certain class or if there exists only one class, and 1 denotes that the elements are randomly distributed across various classes. The Gini impurity can be computed by summing the probability p_i of an item with label i being chosen times the probability $\sum_{k \neq i} \mathrm{p}_k = 1 - \mathrm{p}_i$ of a mistake in categorizing that item.[30] To compute Gini impurity for a set of items with J output classes, suppose $i \in \{1, 2, ..., J\}$, and let p_i be the fraction of items labeled with class i in the set.

$$\mathrm{Gini} = \sum_{i=1}^{J} \mathrm{p}_i \sum_{k \neq i} \mathrm{p}_k = \sum_{i=1}^{J} \mathrm{p}_i(1 - \mathrm{p}_i) = \sum_{i=1}^{J} \mathrm{p}_i - \sum_{i=1}^{J} {\mathrm{p}_i}^2 = 1 - \sum_{i=1}^{J} {\mathrm{p}_i}^2 \tag{3.34}$$

   i will eventually determine the quality of the split with attribute a by weighting the impurity of each branch by how many elements it has. Then we will choose the split that maximizes the Gini Gain calculated by subtracting the weighted impurities of the branches from the original impurity. While building the decision tree, we would prefer choosing the attribute with the minimum Gini impurity as the root node. Gini impurity is computationally cheaper than Information Gain as it does not have the logarithm function used to calculate entropy in information gain which makes it more preferable over Information gain.

3.2 Cost Evaluation Metrics

3.2.1 Injury Severity Cost

The cost of the collision can be evaluated as the sum of all victims' (utilitarianism) or sum of primary vehicle victims' (ethical egoism) Probability (abs or EXP) of Injury severity of each level (PDO, Injury, Fatal). The difference between *Absolute* (*abs*) and EXP Probability are as shown in the sample prediction case.Table 3.1

Table 3.1: Sample Case of a Prediction

Accident Index (n)	Victim Index (m)	Actual Injury Level	Actual Probability of [PDO, Injury, Fatal]	Predicted (EXP) Probability of [PDO, Injury, Fatal]	Predicted Injury Level	Absolute (abs) Predicted Probability of [PDO, Injury, Fatal]
n	m	Injury	[0, 1, 0]	[0.4, 0.35, 0.25]	Level(max(P_PDO_m, P_Injury_m, P_Fatal_m)) =PDO	[1, 0, 0]

Absolute Probability of Severity level Cost

Absolute Injury Severity level of each victim is as in Table 3.1. *Absolute* Predicted Probability of each injury level is resulted from the Predicted Injury Level of the chosen prediction model. It is written as $Absolute\ Prediction\ Probability\ of\ Level_i$ in Equation 5.5 and used for prediction model performance evaluation. For entire accident, the $abs\_Acc\_total\_Level_i$ is define in Equation 3.35. This can used as cost metric for decision making under utilitarianism as Equation 3.46. Under Ethical Egoism,

$abs\_Prim\_total\_Level_i$ as Equation 3.36 is used as cost in Equation 3.45. Assume m victims in a accident and m_{prim} victims in the primary vehicle.

$$\text{abs\_Acc\_total\_Level}_i = \sum_{i=1}^{m} \text{Absolute Prediction Probability of Level}_i \tag{3.35}$$

$$\text{abs\_Prim\_total\_Level}_i = \sum_{i=1}^{m_{prim}} \text{Absolute Prediction Probability of Level}_i \tag{3.36}$$

EXP Probability of Severity level Cost

EXP Probability of Severity level is as in Table 3.1, the Predicted Probability (EXP) of each injury level. It is a direct output of the chosen prediction model. It is written as $EXP\ Prediction\ Probability\ of\ Level_i$ in Equation 5.6 and used for prediction model performance evaluation. Assume m victims in an accident and m_{prim} victims in the primary vehicle.

For entire accident, the $EXP\_Acc\_total\_Level_i$ is define in Equation 3.37. This can used as cost metric for decision making under utilitarianism as Equation 3.46. Under Ethical Egoism, $EXP\_Prim\_total\_Level_i$ as Equation 3.38is used in Equation 3.45. Assume m victims in a Accident and m_{prim} victims in the primary vehicle.

$$\text{EXP\_Acc\_total\_Level}_\text{i} = \sum_{i=1}^{m} \text{EXP Prediction Probability of Level}_i \tag{3.37}$$

$$\text{EXP\_Prim\_total\_Level}_i = \sum_{i=1}^{m_{prim}} \text{EXP Prediction Probability of Level}_i \tag{3.38}$$

3.2.2 Monetary Cost Evaluation Metric

An alternative approach of quantifying the severity of the Accident is to convert the Injury Severity to monetary Cost. As I used *National Collision Data Base* (*NCDB*) for collision Injury Severity Prediction, I used data From Collision Cost Study for Capital Region in Alberta, Canada[31] to make the Monetary Cost Evaluation Metric realistic and applicable across Canada. This paper was dedicated to the Direct Cost of the collision so I used Direct Cost from the paper to calculate the monetary cost of collisions in NCDB data base[11]. In the original paper[31], there are Four different Injury Levels('Fatal', 'Major Injury', 'Minor Injury','PDO'). However, in NCDB data base there are three injury levels('Fatal', 'Injury','PDO'). So I used average values of 'Major Injury' and 'Minor Injury' as the values for 'Injury' in my study. Each victim in a collision has a individual Direct Cost according to Injury Severity Level Table 3.2 and each Accident has an individual Direct Cost of the entire accident according to its Severity Level Table 3.3.

Table 3.2: Per Victim Direct Cost

Per Victim Injury Severity Level	Fatal($)	Injury($)	Property Damage Only($) (PDO)
Health Service Costs			
Emergency Room Costs	2007	314.5	0
ICU Care Costs	59775	20221.5	0
Acute Care Costs	11517	4305.5	0
Rehabilitation Costs	3946	1206.5	0
Continuing Care Costs	23280	7609.5	0
Legal Costs			
Correctional Services	1294	207.5	
Court Costs	456	73.5	0
Legal Aid and Prosecution	461	74	0
Funeral Costs (Fatal Only)	10109	0	0
Productivity / Disruption Costs			
Short-Term Work-Place (Injury)	18654	4962.5	0
Short-Term Work-Place (Fatal)	4761	0	0
Short-Term Work-Place (PDO)	0	0	59
Per Victim Direct Cost(PVDC)			
Per Victim Direct Cost	136260	38975	59

Table 3.3: Per Accident Direct Cost

Per Accident Severity Level	Fatal($)	Injury($)	Property Damage Only($) (PDO)
Property Damage			
Vehicle Repairs	32,412	17,626	9,130
Auto-Insurance Administration	4774	1751	411
Out-of-Pocket Expenses	1462	946	614
Towing Services	888	735	493
Emergency Response Costs			
Police Costs	6621	741	188
Fire / Rescue Costs	3282	2462	0
Ambulance Costs	992	744	0
Coroners Costs	2165	0	0
Travel Delay Cost			
Delay Costs Caused by Collision	20511	6466	2598
Extra Fuel Consumption	1484	468	188
Environmental / Pollution Costs	3028	954	384
Per Accident Direct Cost(PADC)			
Per Accident Direct Cost	77,619	32,893	14,006

There are two ways for calculating predicted per victim direct cost. One is to use the *abs* predicted probability of each injury level in Table 3.1 and corresponding Per Victim Direct Cost of each level to calculate Predicted *abs* Per Victim Direct Cost $PVDC_{Predicted\_Absolute}$ as Equation 3.39. The other is to use the predicted probability of each injury level in Table 3.1 and corresponding Per Victim Direct Cost of each level to calculate Predicted *Probability Based* (EXP) Per Victim Direct Cost $PVDC_{Predicted\_EXP}$

as Equation 3.42. Here $k \in K = [PDO, Injury, Fatal]$.

Assume m victims in an Accident and m_{prim} victims in the primary vehicle, each victim's direct cost is $PVDC_{Predicted\_EXP\_i}$ or $PVDC_{Predicted\_Absolute\_i}$ and $i \in [1, ..., m]$. $k \in K = [PDO, Injury, Fatal]$. Predicted Per Accident Direct Cost ($PADC_{Predicted}$) takes the value in the severity level of the worst *Predicted Injury Level* as in Table 3.1 of all victims in a accident. The predicted total monetary cost of an accident can be calculated using one of the formulas Equation 3.40 and Equation 3.43. This calculated monetary cost will be used for decision making following Equation 3.46 if using utilitarianism. When making decision using ethical egoism, we will sum up the $PVDC$ of all victims in the primary vehicle only as cost Equation 3.45 for comparison and decision making as in Equation 3.41 and Equation 3.44.

***Absolute* (*abs*) Direct Cost**

$$\text{PVDC}_{\text{Predicted\_Absolute}} = \sum_{k \in K} \text{PVDC}_k \times \text{Absolute Predicted Probability of } k \tag{3.39}$$

$$\text{Predicted\_Absolute\_Total\_DC} = (\sum_{i=1}^{m} \text{PVDC}_{\text{Predicted\_Absolute\_}i}) + \text{PADC}_{\text{Predicted}} \tag{3.40}$$

$$\text{Predicted\_Absolute\_Prim\_DC} = \sum_{i=1}^{m_{prim}} \text{PVDC}_{\text{Predicted\_Absolute\_}i} \tag{3.41}$$

Probability Based(EXP) Direct Cost

$$\text{PVDC}_{\text{Predicted\_EXP}} = \sum_{k \in K} \text{PVDC}_k \times \text{Predicted Probability of } k \tag{3.42}$$

$$\text{Predicted\_EXP\_Total\_DC} = (\sum_{i=1}^{m} \text{PVDC}_{\text{Predicted\_EXP\_}i}) + \text{PADC}_{\text{Predicted}} \tag{3.43}$$

$$\text{Predicted\_EXP\_Prim\_DC} = \left(\sum_{i=1}^{m_{prim}} \text{PVDC}_{\text{Predicted\_EXP\_}i} \right) \tag{3.44}$$

3.3 Ethical Decision-Making Foundations

In this paper, a primary vehicle confronting a *dilemma inducing* (*DI*) collision adopts courses of action based on the user's predefined selection of one of the two *ethical foundation* (*EF*) : ethical egoism or utilitarianism. The cost of each available vehicle control option during the collision $C(i,j)$ will be calculated first using severity prediction and the user selected cost evaluation algorithm and then input to the Decision Making algorithm. $i \in [1,..,n]$ denotes one of the n vehicle control options and $j \in [1,..,k]$ denotes the selected cost evaluation metric in k available metrics.

3.3.1 Ethical Egoism

The goal of Ethical Egoism is that individuals and organizations prioritize their own self-interests. Ethical Egoism does not preclude people from taking actions that manifest in helping others, but it starts from the foundation that people make decisions that maximize benefits (or minimize harm) for/to themselves. Therefore, decisions underpinned by an Ethical Egoism foundation do so from the perspective of the occupant(s) of the primary vehicle. Mathematically we can have the following optimization equation when using Ethical Egoism as the decision-making foundation of the primary vehicles:

$$\begin{aligned}
&\underset{i}{\text{argmin}}\ C_{\text{primary vehicle}}(i,j) \\
&\text{s.t. } i \in [1,...,n] \text{ (control options)} \\
&j = m \text{ (cost evaluation metric predefined by user)}
\end{aligned} \tag{3.45}$$

3.3.2 Utilitarianism

The core premise of utilitarianism is that the right action in any given situation is the one that will produce the best overall outcome, as judged from an impersonal standpoint, which gives equal weight to the interests of everyone. Unlike Ethical Egoism, Utilitarianism can result in sacrificing the benefits (or lead in larger harm) to the occupant(s) of the primary vehicle in return of minimizing the overall cost of all the victims and vehicles involved in the collision. The optimization equation when using Utilitarianism as the decision-making foundation of the primary vehicles is as follows:

$$\begin{aligned} &\underset{i}{\operatorname{argmin}} \sum_{\text{all vehicles}} C_{\text{eachinvolved vehicle}}(i, j) \\ &\text{s.t. } i \in [1, ..., n] \text{ (control options)} \\ &j = m \text{ (cost evaluation metric predefined by user)} \end{aligned} \tag{3.46}$$

Chapter 4

Data Preparation

4.1 Crash Data Description

In this study, the collision data comes from the *National Collision Data Base* (*NCDB*) between 1999-2017 established by the Transport Canada[11], which contains all police-reported motor vehicle collisions on public roads in Canada from 1999 to the most recent available data. The collision data set includes a large number of characteristic variables of collisions on a single victim level such as age, gender, and position, as well as the entire accident level including Roadway configuration, Road surface condition and etc. Each collision has a unique case number 'C_CASE'. The injury severity on a single victim indicated by 'P_ISEV' is divided into 3 types: No Injury (P_ISEV =1), Injury (P_ISEV =2) and Fatality (P_ISEV =2).

4.2 Data Processing

The entire data processing process is shown in Figure 4.1.

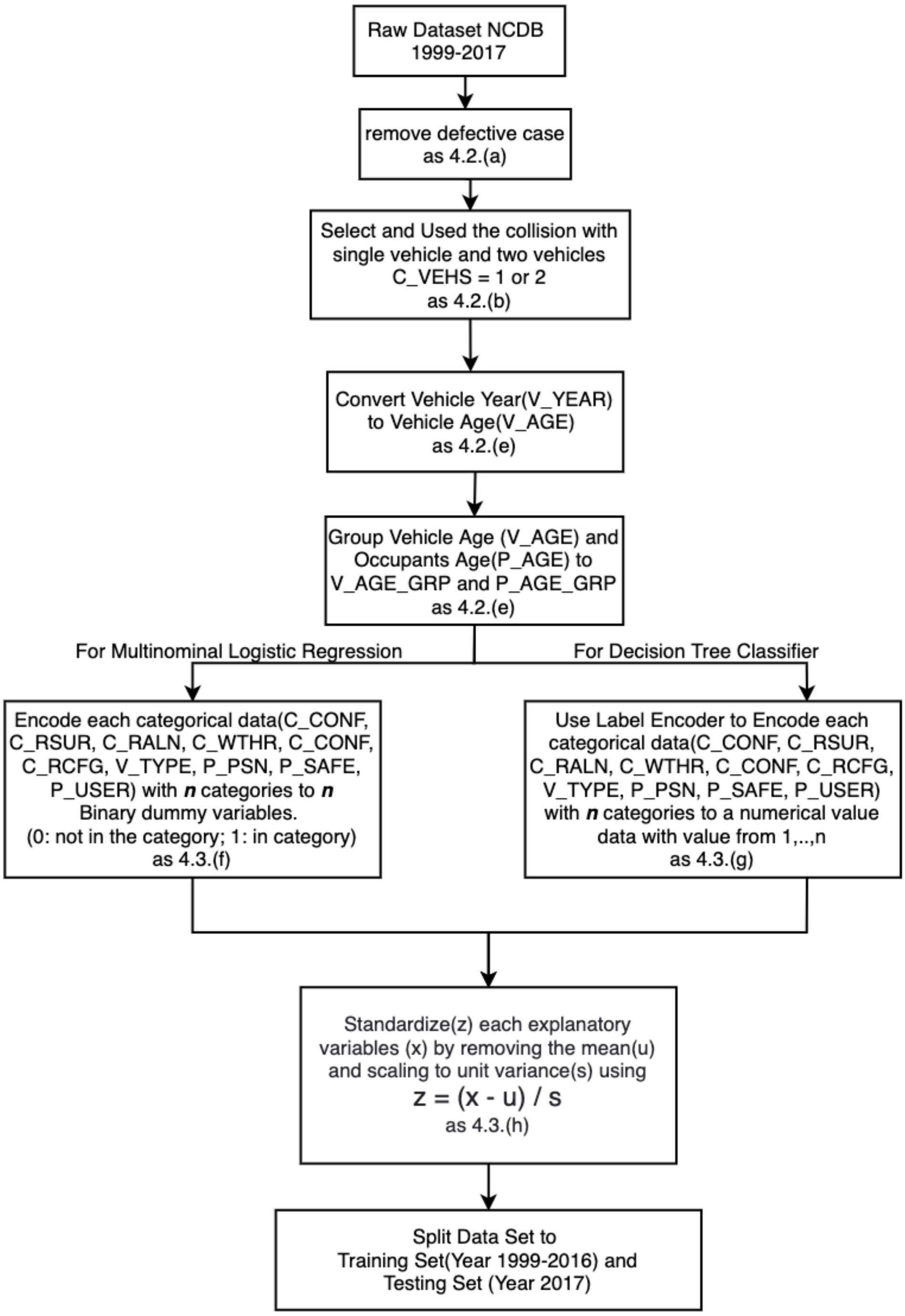

FIGURE 4.1: Data Processing Flowchart.

(a) In order to ensure the accuracy of the model building, crash records with missing or wrong data will be eliminated to obtain complete training samples. The entire traffic accident will be removed if one involved victim has incomplete data.

(b) This paper only studies the collisions on a single-vehicle itself and between two vehicles. Therefore, only cases with variable C_VEHS =1 and C_VEHS =2 are extracted.

(c) Removing some variables in the dataset that are not closely relevant to the collision victim severity like Vehicle sequence number (V_ID)

[(d)] After the above data screening, from 1999-2017 the data set characteristics are as Table 4.1 Data from 1996-2016 is used as the standard model training set and data from 2017 is used for testing.

Table 4.1: 1999-2017 NCDB Collision Severity on per Victim Level

year	No Injury	Injury	Fatality	Total
1999-2016(number)	513,315	790,379	9,867	1,313,561
1999-2016(%)	39.0781%	60.1707%	0.7512%	100%
2017(number)	32,187	51,293	576	84,056
2017(%)	38.2923%	61.0224%	0.6853%	100%
1999-2017(number)	545,502	841,672	10,443	1,397,617
1999-2017(%)	39.0309%	60.2219%	0.7472%	100%

(e) Instead of using the value date of victims' age (P_AGE) and Vehicles' model year (V_YEAR) directly, Vehicles' age are calculated (V_AGE =C_YEAR - V_YEAR) and then grouped as a new variable V_AGE_GRP with fixed interval of 5 years and victims' age are grouped as P_AGE_GRP following the same categories as Canadian Motor Vehicle Traffic Collision Statistics[32] and Use the Group Number as Numerical Values.

(f) For Multinominal Logistic Regression, each categorical variable (with n categories) is separated into n dummy variables with binary characteristic.

(g) For Decision Tree Classifier, each categorical variable(with n categories) is encoded to a single numerical value variable with value from $1, ..., n$ using label encoder.

(h) Standardize (z) each explanatory variable (x) by removing the mean(u) and scaling to unit variance(s) using $z = \frac{(x-u)}{s}$

4.3 Attributes data distribution

The distribution of Attributes in the processed data set is as Table 4.2

Table 4.2: 1999-2017 Attributes(Explanatory Variables) data distribution

No.	**Attribute Name**	**Description**	**Number of model variables**	**Data Type**	**Descriptive Statistics (% of victims)**
1	C_-YEAR	Year of accident occurrence	1	Numeric	1999-2016: 93.9858 % 2017: 6.0142%
2	C_-VEHS	Number of Vehicles collides	1	Numeric [1,2]	1: 22.2192 % 2: 77.7808%
3	C_-CONF	Configuration of the accident type	18	Binary (Encoded)	Single Vehicle in Motion: C_CONF_01: 0.6414 % (Hit a moving object:E.g.a person or an animal) C_CONF_02: 2.8242 % (Hit a stationary object: E.g.tree) C_CONF_03: 3.9504 % (Ran off left shoulder,Including rollover in the left ditch) C_CONF_04: 5.3609 % (Ran off right shoulder,Including rollover in the right ditch) C_CONF_05: 0.3613 % (Rollover on roadway) C_CONF_06: 9.8510 % (Any other single vehicle collision configuration)

3	C_- CONF	Configuration of the accident type	18	Binary (Encoded)	Two Vehicles in Motion - Same Direction of Travel: C_CONF_21: 28.565 % (Rear-end collision) C_CONF_22: 3.7206 % (Side swipe) C_CONF_23: 0.9574 % (One vehicle passing to the left of the other, or left turn conflict) C_CONF_24: 0.6910 % (One vehicle passing to the right of the other, or right turn conflict) C_CONF_25: 0.1011 % (Any other two vehicle - same direction of travel configuration) Two Vehicles in Motion - Different Direction of Travel: C_CONF_31: 3.8085 % (Head-on collision) C_CONF_32: 0.6651 % (Approaching side-swipe) C_CONF_33: 8.0425 % (Left turn across opposing traffic) C_CONF_34: 0.6533 % (Right turn, including turning conflicts) C_CONF_35: 16.9376 % (Right angle collision) C_CONF_36: 12.7865 % (Any other two-vehicle - different direction of travel configuration) Two Vehicles - Hit a Parked Motor Vehicle: C_CONF_41: 0.0822 % (Hit a parked motor vehicle)

4	C_-RCFG	Road type configuration	10	Binary (Encoded)	C_RCFG_01: 40.6786 % (Non-intersection, e.g. 'mid-block') C_RCFG_02: 52.3043 % (At an intersection of at least two public roadways) C_RCFG_03: 5.6151 % (Intersection with parking lot entrance/exit, private driveway or laneway) C_RCFG_04: 0.2465 % (Railroad level crossing) C_RCFG_05: 0.7677 % (Bridge, overpass, viaduct) C_RCFG_06: 0.0942 % (Tunnel or underpass) C_RCFG_07: 0.0097 % (Passing or climbing lane) C_RCFG_08: 0.2277 % (Ramp) C_RCFG_09: 0.0527 % (Traffic Circle) C_RCFG_10: 0.0034 % (Express lane of a freeway system)
5	C_-WTHR	Weather when accident occurence	7	Binary (Encoded)	C_WTHR_1: 70.7189 % (Clear and sunny) C_WTHR_2: 9.4039 % (Overcast, cloudy but no precipitation) C_WTHR_3: 11.1345 % (Raining) C_WTHR_4: 6.3085 % (Snowing, not including drifting snow) C_WTHR_5: 0.5824 % (Freezing rain, sleet, hail) C_WTHR_6: 1.5526 % (Visibility limitation) C_WTHR_7: 0.2992 % (Strong wind)

6	C_- RALN		6	Binary (Encoded)	C_RALN_1: 76.0717 % (Straight and level) C_RALN_2: 10.1622 % (Straight and gradient) C_RALN_3: 7.8927 % (Curved and level) C_RALN_4: 4.4743 % (Curved with gradient) C_RALN_5: 0.8082 % (Top of hill or gradient) C_RALN_6: 0.5909 % (Bottom of hill or gradient)
7	C_- RSUR	Road surface condition	9	Binary (Encoded)	C_RSUR_1: 67.9174 % (Dry, normal) C_RSUR_2: 19.5049 % (Wet) C_RSUR_3: 4.6207 % (Snow (fresh, loose snow)) C_RSUR_4: 1.6623 % (Slush ,wet snow) C_RSUR_5: 5.6241 % (Icy) C_RSUR_6: 0.5491 % (Sand/gravel/dirt) C_RSUR_7: 0.0946 % (Muddy) C_RSUR_8: 0.0212 % (Oil and spilled liquid or road application) C_RSUR_9: 0.0057 % (Flooded)

8	V_-TYPE	Vehicle type	13	Binary (Encoded)	V_TYPE_01: 0.902109% (Light Duty Vehicle) V_TYPE_05: 0.012855% (Panel/cargo van weight $\leq$ 4536 KG GVWR) V_TYPE_06: 0.023031% (Other trucks and vans weight $\leq$ 4536 KG GVWR) V_TYPE_07: 0.013758% (Unit trucks $\geq$ 4536 KG GVWR) V_TYPE_08: 0.012843% (Road tractor) V_TYPE_09: 0.003332% (School bus Standard large type) V_TYPE_10: 0.000183% (Smaller school bus seats $<$ 25 passengers V_TYPE_11: 0.007737% (Urban and Intercity Bus) V_TYPE_14: 0.021960% (Motorcycle and moped) V_TYPE_17: 0.001185% (Bicycle) V_TYPE_18: 0.000418% (Purpose-built motor home) V_TYPE_21: 0.000176% (Fire engine) V_TYPE_23: 0.000412% (Street car)
9	V_-AGE_-GRP	Age of the Vehicle	1	Numerical 1 - 21	1: 32.9296% (-1 ~ 4 years old car) 2: 32.5688 % (5 ~ 9 years old car) 3: 24.1367 % (10 ~ 14 years old car) 4: 7.9195 % (15 ~ 19 years old car) 5: 1.7091 % (20 ~ 24 years old car) 6-21: 0.7363% (every 5 increment the variable value by 1)
10	P_SEX	Passenger Sex	1	Binary	0: 45.918% (Female) 1: 54.082% (Male)

11	P_-AGE_-GRP	Age of the Victim	1	Numerical 1 - 9	1: 2.5024 % (0 ~ 3 years old) 2: 5.8894 % (4 ~ 14 years old car) 3: 10.9823 % (15 ~ 18 years old car) 4: 12.9582 % (19 ~ 23 years old car) 5: 18.0429 % (24 ~ 33 years old car) 6: 15.9653 % (34 ~ 43 years old car) 7: 14.7409 % (44 ~ 53 years old car) 8: 9.8566 % (54 ~ 63 years old car) 9: 9.0619 % (64 years old and above)
12	P_PSN	Position of the victim in the vehicle	12	Binary (Encoded)	P_PSN_11: 69.0774 % (Driver) P_PSN_12: 1.1659 % (Front row, center) P_PSN_13: 17.3661 % (Front row, right outboard, including motorcycle passenger in sidecar) P_PSN_21: 4.3536 % (Second row, left outboard, including motorcycle passenger) P_PSN_22: 1.5863 % (Second row, center) P_PSN_23: 5.3744 % (Second row, right outboard) P_PSN_31: 0.483 % (Third row, left outboard) P_PSN_32: 0.4974 % (Third row, center) P_PSN_33: 0.0619 % (Third row, right outboard) P_PSN_96: 0.4606 % (Position unknown, but the person was definitely an occupant) P_PSN_97: 0.0039 % (Sitting on someonés lap) P_PSN_98: 0.0041 % (Outside passenger compartment E.g:back of a pick-up truck)

13	P_-SAFE	Safety device used by the victim	6	Binary (Encoded)	P_SAFE_01: 2.7068 % (No safety device used or No child restraint used) P_SAFE_02: 93.4806% (Safety device used or child restraint used) P_SAFE_09: 2.0585 % (Only Helmet worn) P_SAFE_10: 0.0008 % (Only Reflective clothing worn) P_SAFE_12: 0.6543 % (Other safety device used) P_SAFE_13: 1.0990 % (No safety device equipped E.g:Bus)
14	P_-USER	Victim Class	4	Binary (Encoded)	P_USER_1 : 67.2078% (Motor Vehicle Driver) P_USER_2 : 30.4777% (Motor Vehicle Passenger) P_USER_4 : 0.1185 % (Bicyclist) P_USER_5 : 2.1960 % (Motorcyclist)

Chapter 5

Result and Simulation

5.1 Cost Prediction Result

I used Scikit-learn open-source machine learning library in Python to develop the *Multinominal Logistic Regression* (*MLR*) model and *Decision Tree Classifier* (*DTC*) for per victim *Collision Injury Severity Level Prediction* (*CISLP*) models. The Training set for both models is *NCDB* between 1999-2016 and Testing Set is *NCDB* in 2017, which was processed as described in Data Preparation. Both models are carefully tuned and the result comparison in predicting severity is presented. Multiple performance evaluation measures are used for both models for comparing the performance as well as for the cost evaluation for decision-making. The accuracy and errors for using each model to perform cost evaluation of collisions with different cost evaluation metrics are presented and compared.

5.1.1 *Collision Injury Severity Level Prediction* Evaluation

Confusion Matrix, Accuracy and Recall Rate

The confusion matrix is a summary of prediction results for classification problems. The number of correct and incorrect predictions are summarized with count values and broken down by each class. The confusion matrix of our prediction model is presented in Table 5.1 with each count value in cells used for further accuracy and recall rate calculation.

Table 5.1: Generalized Confusion Matrix for per victim Collision Injury Severity Prediction

	Predicted Sevrity			
Actual Severity		PDO	Injury	Fatal
	PDO	$\text{Severity}_{\text{PDO\_PDO}}$	$\text{Severity}_{\text{PDO\_Injury}}$	$\text{Severity}_{\text{PDO\_Fatal}}$
	Injury	$\text{Severity}_{\text{Injury\_PDO}}$	$\text{Severity}_{\text{Injury\_Injury}}$	$\text{Severity}_{\text{Injury\_Fatal}}$
	Fatal	$\text{Severity}_{\text{Fatal\_PDO}}$	$\text{Severity}_{\text{Fatal\_Injury}}$	$\text{Severity}_{\text{Fatal\_Fatal}}$

The most common measures of prediction(classification) performance are Accuracy and Recall Rate. They are defined as follows

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}} \tag{5.1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5.2}$$

where TP, TN, FP, FN are abbreviations for true positives, true negatives, false positives, and false negatives. In the case of severity prediction, as there are three categories, the initial formula needs to be modified for each category. $Severity_{Actual\_Predicted}$ stands for the collision result of a victim where the real severity level indicated by $Actual \in [PDO, Injury, Fatal]$ and predicted severity level indicated by $Predicted \in [PDO, Injury, Fatal]$.

$$\text{Accuracy} = \frac{\text{Severity}_{\text{PDO\_PDO}} + \text{Severity}_{\text{Injury\_Injury}} + \text{Severity}_{\text{Fatal\_Fatal}}}{\text{All elements in Confusion Matrix}} \tag{5.3}$$

$$\begin{aligned}
\text{Recall}_{\text{PDO}} &= \frac{\text{Severity}_{\text{PDO_PDO}}}{\text{Severity}_{\text{PDO_PDO}} + \text{Severity}_{\text{PDO_(Injury,Fatal)}}} \\
\text{Recall}_{\text{Injury}} &= \frac{\text{Severity}_{\text{Injury_Injury}}}{\text{Severity}_{\text{Injury_Injury}} + \text{Severity}_{\text{Injury_(PDO,Fatal)}}} \\
\text{Recall}_{\text{Fatal}} &= \frac{\text{Severity}_{\text{Fatal_Fatal}}}{\text{Severity}_{\text{Fatal_Fatal}} + \text{Severity}_{\text{Fatal_(PDO,Injury)}}}
\end{aligned} \tag{5.4}$$

There are several other evaluation metrics used to measure the performance of models.

Probability Error

Root Mean Squared (RMS) Per Victim Absolute Prediction Probability Error Root Mean Squared(RMS) Per Victim Absolute Prediction Probability Error for each Injury Level is defined as

$$\begin{aligned}
&\text{Per Victim Absolute_Prob_Error_Level}_i \\
&= \text{Actual Probability of Level}_i - \text{Absolute Prediction Probability of Level}_i
\end{aligned} \tag{5.5}$$

$$\begin{aligned}
&\text{RMS Per Victim Absolute_Prob_Error_Level}_i \\
&= \sqrt{\frac{\sum_{\text{all victims}}(\text{Per Victim Absolute_Prob_Error_Level}_i)^2}{\text{Total Number of Victims}}}
\end{aligned}$$

Root Mean Squared (RMS) Per Victim EXP Prediction Probability Error Root Mean Squared (RMS) Per Victim EXP Prediction Probability Error for each Injury

Level defined as

$$\begin{aligned}&\text{Per Victim EXP\_Prob\_Error\_Level}_i\\&=\text{Actual Probability of Level}-\text{EXP Prediction Probability of Level}\end{aligned} \tag{5.6}$$

$$\begin{aligned}&\text{RMS Per Victim EXP\_Prob\_Error\_Level}_i\\&=\sqrt{\frac{\sum_{\text{all victims}}(\text{Per Victim EXP\_Prob\_Error\_Level}_i)^2}{\text{Total Number of Victims}}}\end{aligned}$$

Root Mean Squared (RMS) Per Accident Absolute Prediction Probability Error Root Mean Squared Per Accident Absolute Prediction Probability Error for each Injury Level defined as

$$\begin{aligned}&\text{RMS Per Accident Absolute\_Prob\_Error\_Level}_i\\&=\sqrt{\frac{\sum_{\text{all Accidents}}\sum_{\text{all victims in Accident}}(\text{Per Victim Absolute\_Prob\_Error\_Level}_i)^2}{\text{Total Number of Accidents}}}\end{aligned} \tag{5.7}$$

Root Mean Squared (RMS) Per Accident EXP Prediction Probability Error Root Mean Squared Per Accident EXP Prediction Probability Error for each Injury Level defined as

$$\begin{aligned}&\text{RMS Per Accident EXP\_Prob\_Error\_Level}_i\\&=\sqrt{\frac{\sum_{\text{all Accidents}}\sum_{\text{all victims in Accident}}(\text{Per Victim EXP\_Prob\_Error\_Level}_i)^2}{\text{Total Number of Accidents}}}\end{aligned} \tag{5.8}$$

Monetary Cost Error

The Actual per victim direct cost $PVDC_{Actual}$ is defined as

$$\text{PVDC}_{\text{Actual}} = \sum_{k \in K} \text{PVDC}_k \times \text{Actual Probability of } k \tag{5.9}$$

Actual Per Accident Direct Cost ($PADC_{Actual}$) takes the value in the severity level of the worst *Actual Injury Level* as in Table 3.1 of all victims in an collision. The Actual total monetary cost of an collision can be calculated usingEquation 5.10

$$\text{Actual\_Total\_DC} = (\sum_{i=1}^{m} \text{PVDC}_{\text{Actual\_}i}) + \text{PADC}_{\text{Actual}} \tag{5.10}$$

Root Mean Squared Per Accident Absolute Total Direct Cost Error

$$\begin{aligned} &\text{RMS\_Absolute\_Total\_DC\_Error} \\ &= \sqrt{\frac{\sum_{\text{all Accidents}} (\text{Predicted\_Absolute\_Total\_DC} - \text{Actual\_Total\_DC})^2}{\text{Total Number of Accidents}}} \end{aligned} \tag{5.11}$$

Root Mean Squared Per Accident EXP Total Direct Cost Error

$$\begin{aligned} &\text{RMS\_EXP\_Total\_DC\_Error} \\ &= \sqrt{\frac{\sum_{\text{all Accidents}} (\text{Predicted\_EXP\_Total\_DC} - \text{Actual\_Total\_DC})^2}{\text{Total Number of Accidents}}} \end{aligned} \tag{5.12}$$

5.2 Multinominal Logistic Regression(MLR) Prediction Performance Evaluation

sklearn.linear_model.LogisticRegression is the package used for developing MLR model. There are several important parameters Table 5.2 that can be tuned to customize the result of prediction.

Table 5.2: Multinominal Logistic Regression Hyperparameters

Parameter Name	**Parameter Description**	**options used**
penalty	Regularization Method to be used: l1 for Lasso Regularization l2 for Ridge Regularization	l1; l2
C	Parameter C for both Lasso and Ridge Regularization	default C =1.0 and 10 values equally logspaced between 10E-4 and 10E4 [0.0001, 0.00077, 0.00599, 0.046415, 0.35938, 1.0000, 2.782559, 21.5443, 166.810, 1291.55, 10000]
class_weight	For imbalanced training data, class_weight can be set to adjust the value of s_k	'balanced' or adjusted according to the importance of accurately predicting Injury and Fatal cases comparing to PDO [balanced, weight 1,weight 2,weight 3,weight 4]
solver	Optimization Algorithm used to solve loss function for coefficients	saga
multi_class	Indicating the type of logistic Regression	multinominal

As shown in Table 4.1, the number of observations of three injury severity levels are heavily imbalanced and class_weight was applied to solve this problem to set sw_k in section 3.1.1.3 equation(11). When performing the severity prediction, we would value the importance of correctly predicting 'Injury' and 'Fatal' cases over 'PDO' cases. This means assigning different importance to the prediction of three levels of severity. I used r_k to denote this importance factor for $k \in [1, 2, 3]$ for 3 severity levels. Here I assumed the importance of 'Injury' and 'Fatal' are the same, which means $r_2 = r_3 = r$ and assign importance of severity level 'PDO' as the reference $r_1 = 1$. Use n_k to denotes the number of observations in training set at different severity levels and n denotes the total number of observations. Use w_k to denotes the resulting class_weights. Then we have the following equations to solve for w_k:

$$\begin{aligned} n_1 w_1 + n_2 w_2 + n_3 w_3 &= n \\ n_1 w_1 &= \frac{n_2 w_2}{r} \\ n_1 w_1 &= \frac{n_3 w_3}{r} \end{aligned} \tag{5.13}$$

The resulting class_weight is indicated as Table 5.3.

Table 5.3: class_weights

r	class_weights [w1, w2, w3]	name
1	[0.85299, 0.55398, 44.376]	balanced
1.2	[0.75264, 0.58657, 46.986]	weight 1
1.4	[0.67341, 0.61229, 49.047]	weight 2
1.6	[0.60928, 0.63312, 50.715]	weight 3
1.8	[0.55630, 0.65032, 52.093]	weight 4

5.2.1 Prediction Accuracy and Recall Rate

Effects on Accuracy, Recall Rate of modifying the class_weight and set $C = 1.0$ are as Figure 5.1

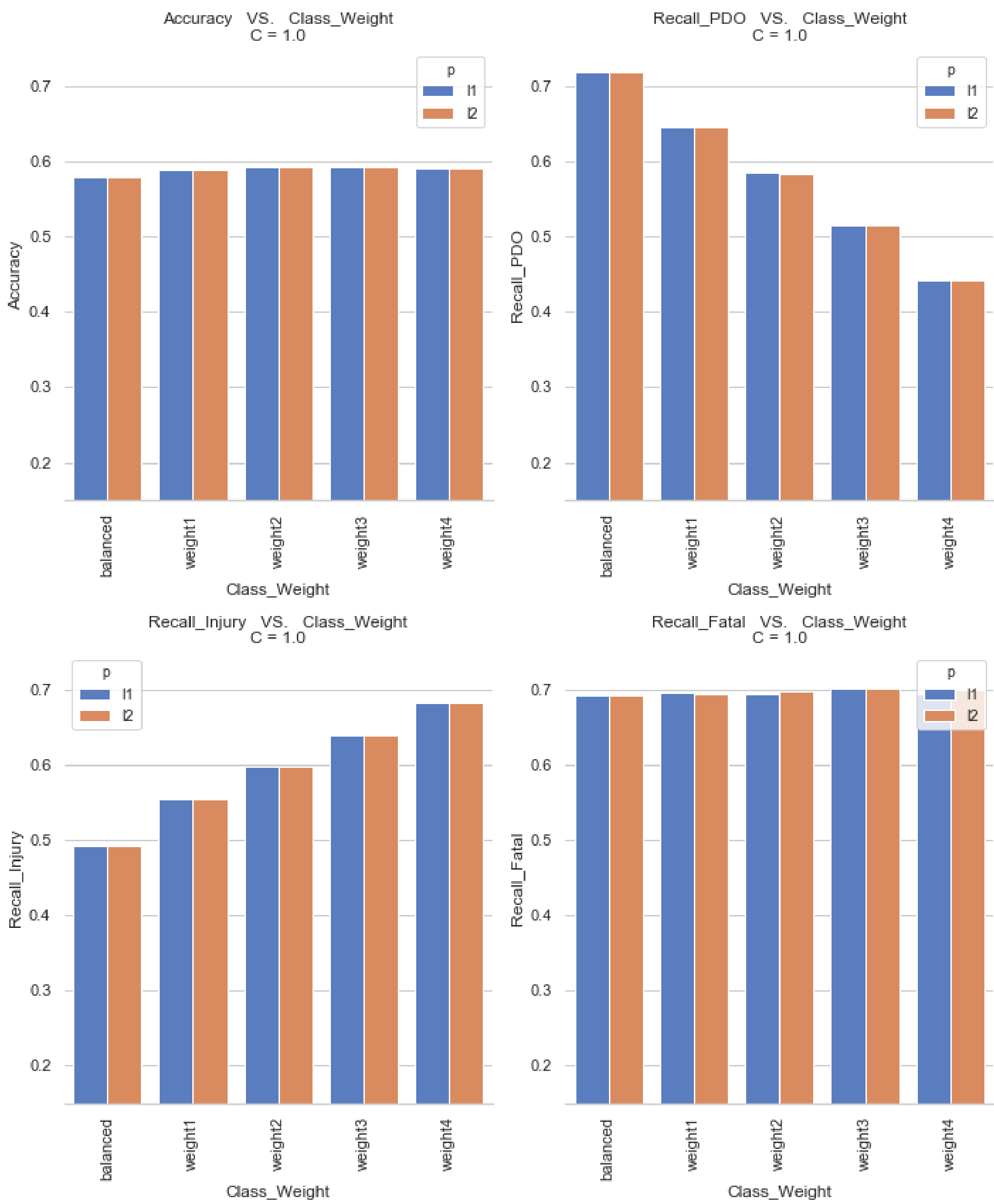

FIGURE 5.1: Accuracy and Recall Rate with C = 1.0 vs. Class_weights.

Effects on Accuracy, Recall Rate of modifying the Regularization settings and set $class\_weight = balanced$ are as Figure 5.2

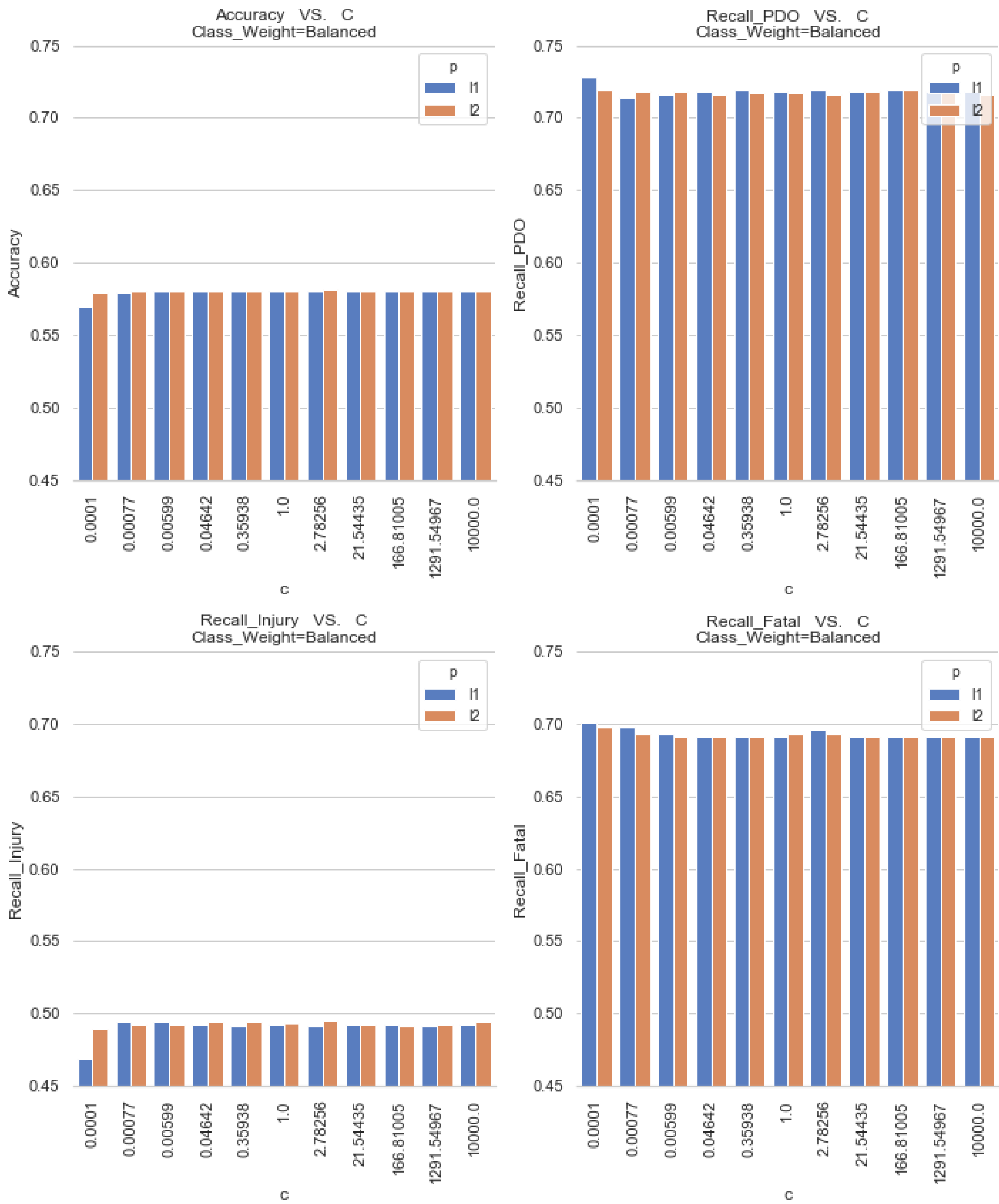

FIGURE 5.2: Accuracy and Recall Rate with class_weight = balanced vs. C.

5.2.2 Probability Error Per Victim Equation 5.5 Equation 5.6

Effects on $RMS\ Per\ Victim\ Absolute\_Prob\_Error\_Level_i$ and $RMS\ Per\ Victim\ EXP\_Prob\_Error\_Level_i$ of modifying the Regularization settings and set $class\_weight = balanced$ are as Figure 5.3

Effects on $RMS\ Per\ Victim\ Absolute\_Prob\_Error\_Level_i$ and $RMS\ Per\ Victim\ EXP\_Prob\_Error\_Level_i$ of modifying the class_weight and set $C = 1.0$ are asFigure 5.4

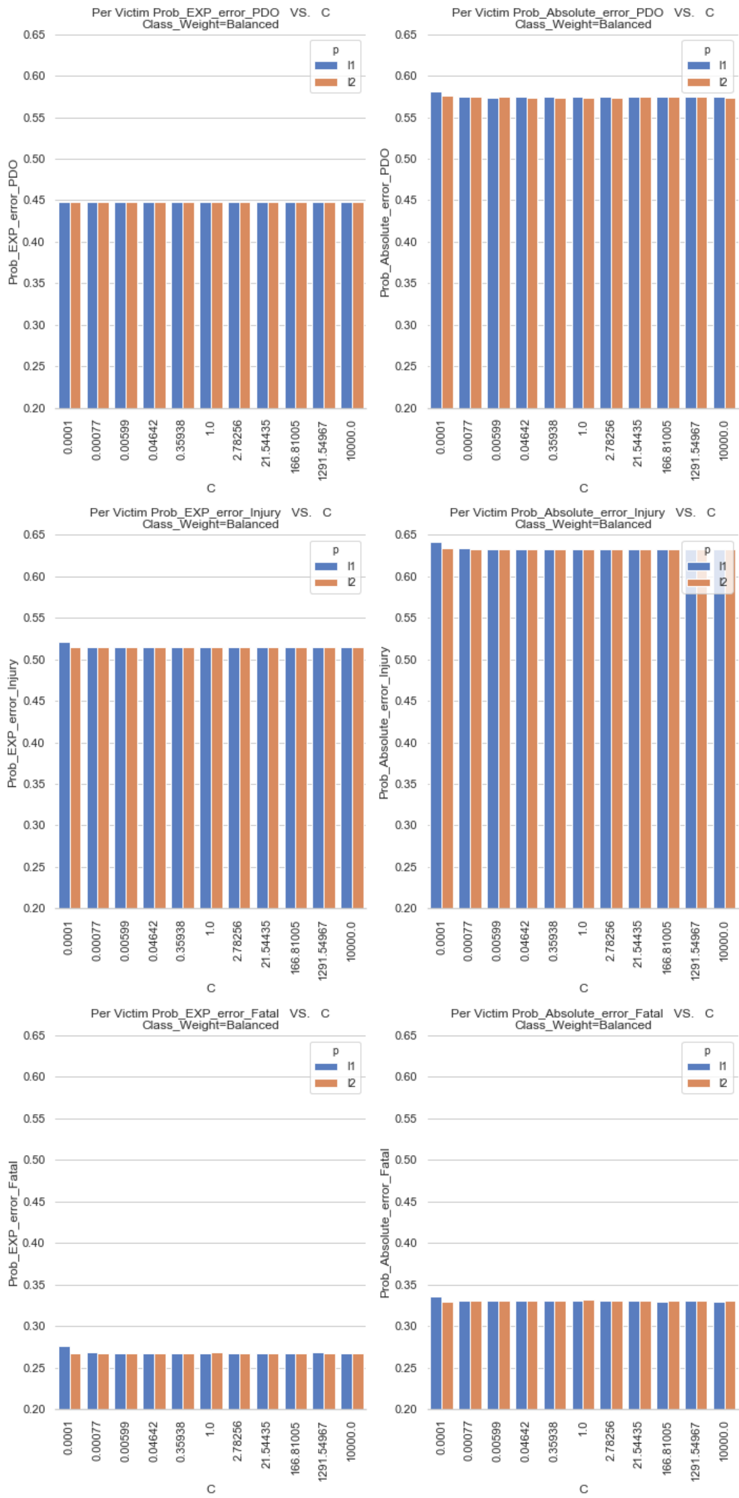

FIGURE 5.3: RMS Per Victim Probability Error with class_weight = balanced vs. C.

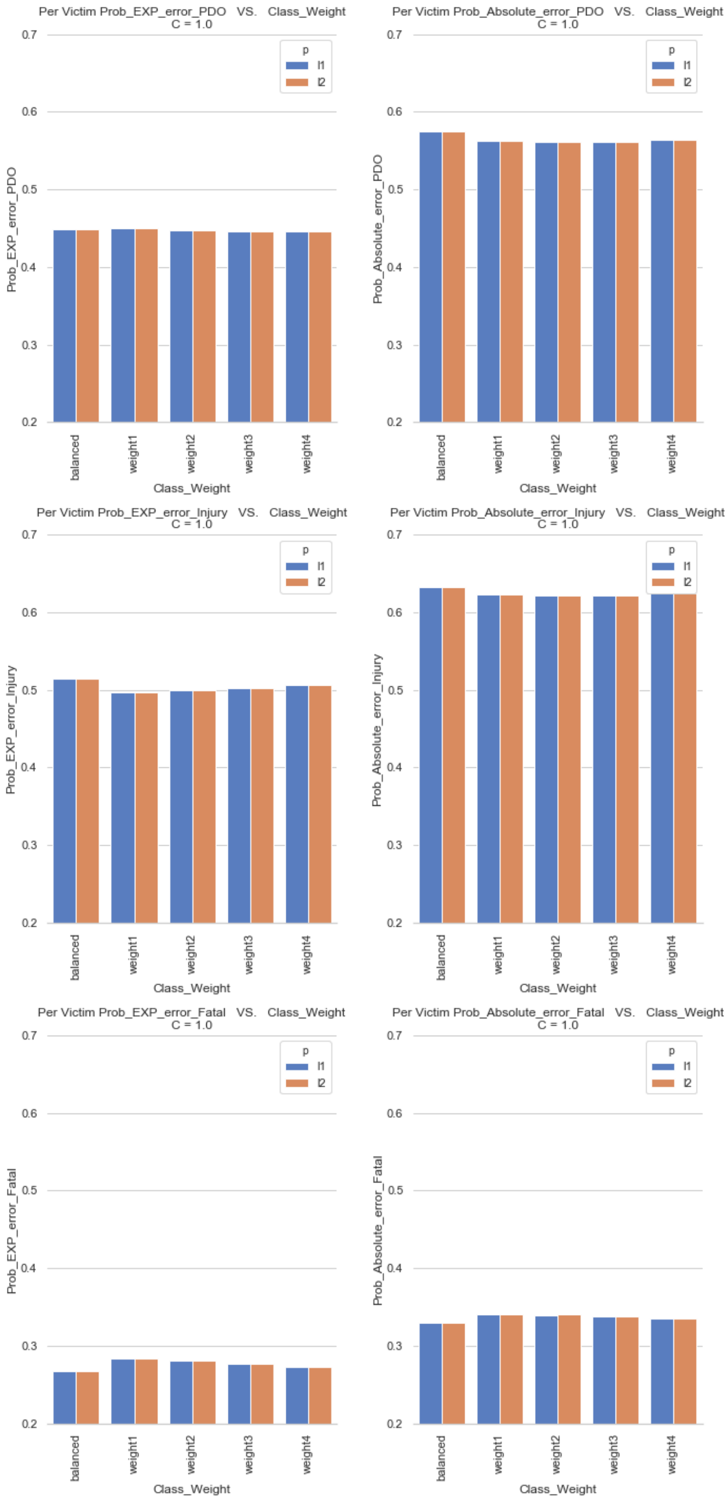

FIGURE 5.4: RMS Per Victim Probability Error with C = 1.0 vs. Class_-weights.

5.2.3 Probability Error per AccidentEquation 5.7 Equation 5.8 and Total Direct Cost Error Per Accident Equation 5.11 Equation 5.12

Effects on RMS *Per Accident* $Absolute\_Prob\_Error\_Level_i$ Equation 5.7 and $RMS\_Absolute\_Total\_DC\_Error$ Equation 5.11 by modifying the Regularization settings and set $class\_weight = balanced$ are as Figure 5.5.

$RMS\_Absolute\_Total\_DC\_Error$ for each level is plotted as "Predict_Total_Error_-$".

RMS *Per Accident* $Absolute\_Prob\_Error\_Level_i$ are plotted as "Acc_Predict_Absolute_Prob_Level_Error".

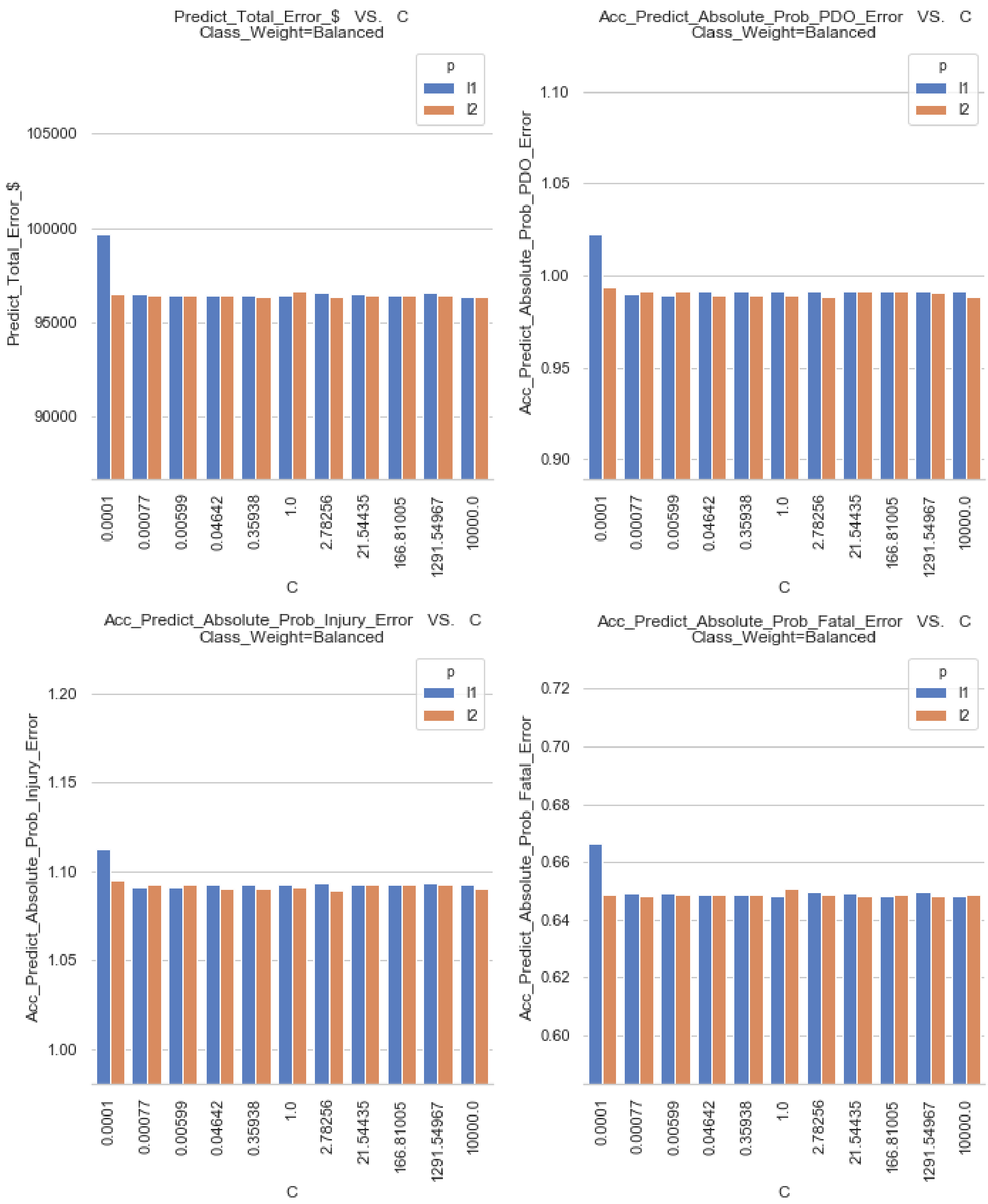

Figure 5.5: RMS Per Accident Absolute_Prob_Error and RMS_Absolute_Total_DC_Error with class_weight = balanced vs. C

Effects on $RMS\ Per\ Accident\ EXP\_Prob\_Error\_Level_i$ Equation 5.8 and $RMS\_EXP\_Total\_DC\_Error$ Equation 5.12 by modifying the Regularization settings and set $class\_weight = balanced$ are as Figure 5.6.

$RMS\_EXP\_Total\_DC\_Error$ for each level is plotted as "Predict_Total_Error_$".

$RMS\ Per\ Accident\ EXP\_Prob\_Error\_Level_i$ are plotted as "Acc_Predict_EXP _Prob_Level_Error".

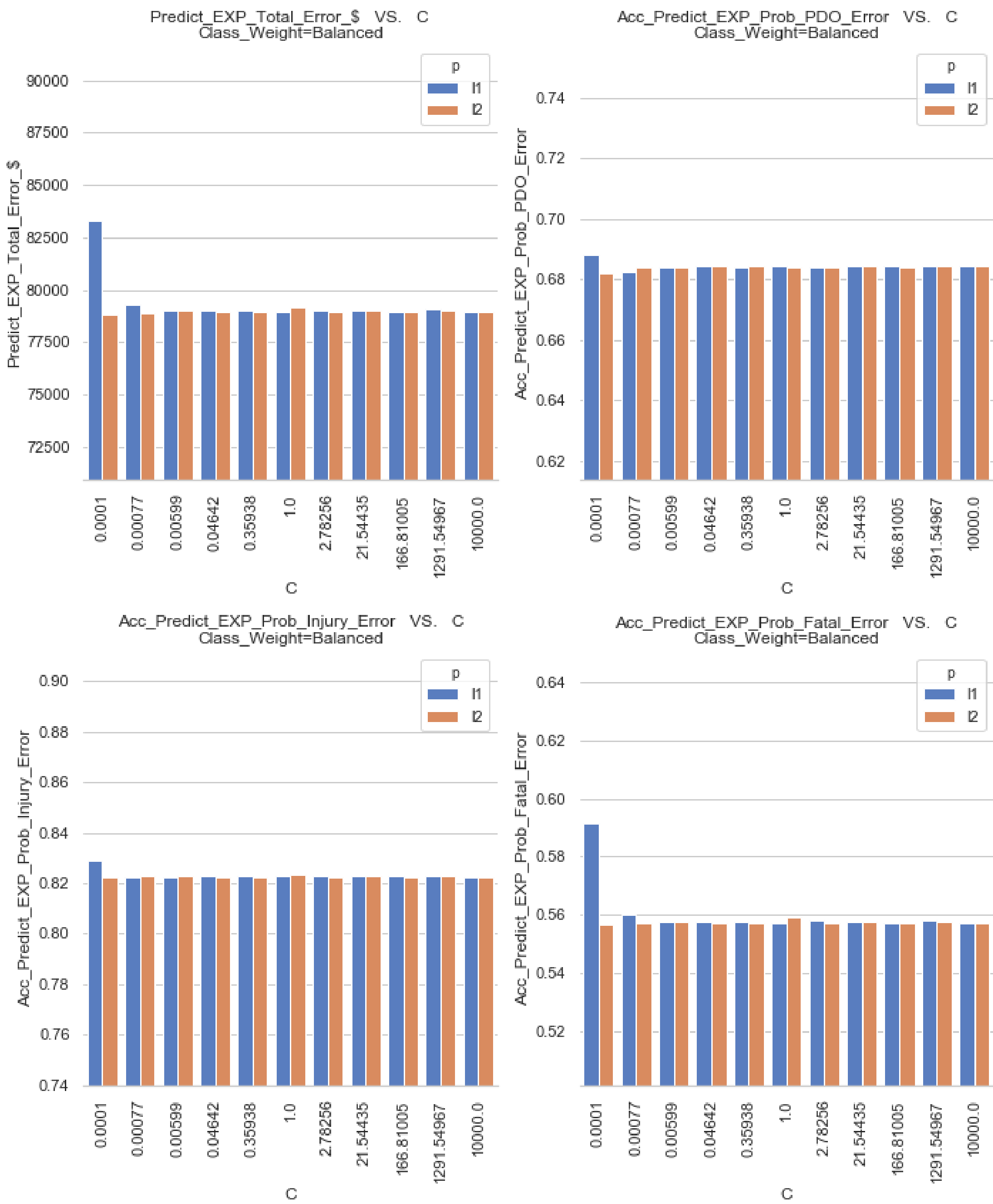

FIGURE 5.6: RMS Per Accident EXP_Prob_Error and RMS_EXP_-Total_DC_Error with class_weight = balanced vs. C

Effects on $RMS\ Per\ Accident\ Absolute\_Prob\_Error\_Level_i$ Equation 5.7 and $RMS\_Absolute\_Total\_DC\_Error$ Equation 5.11 by modifying the class_weight and set $C = 1.0$ are as Figure 5.7.

$RMS\_Absolute\_Total\_DC\_Error$ for each level is plotted as "Predict_Total_Error_-$".

$RMS\ Per\ Accident\ Absolute\_Prob\_Error\_Level_i$ are plotted as "Acc_Predict_Absolute_Prob_Level_Error".

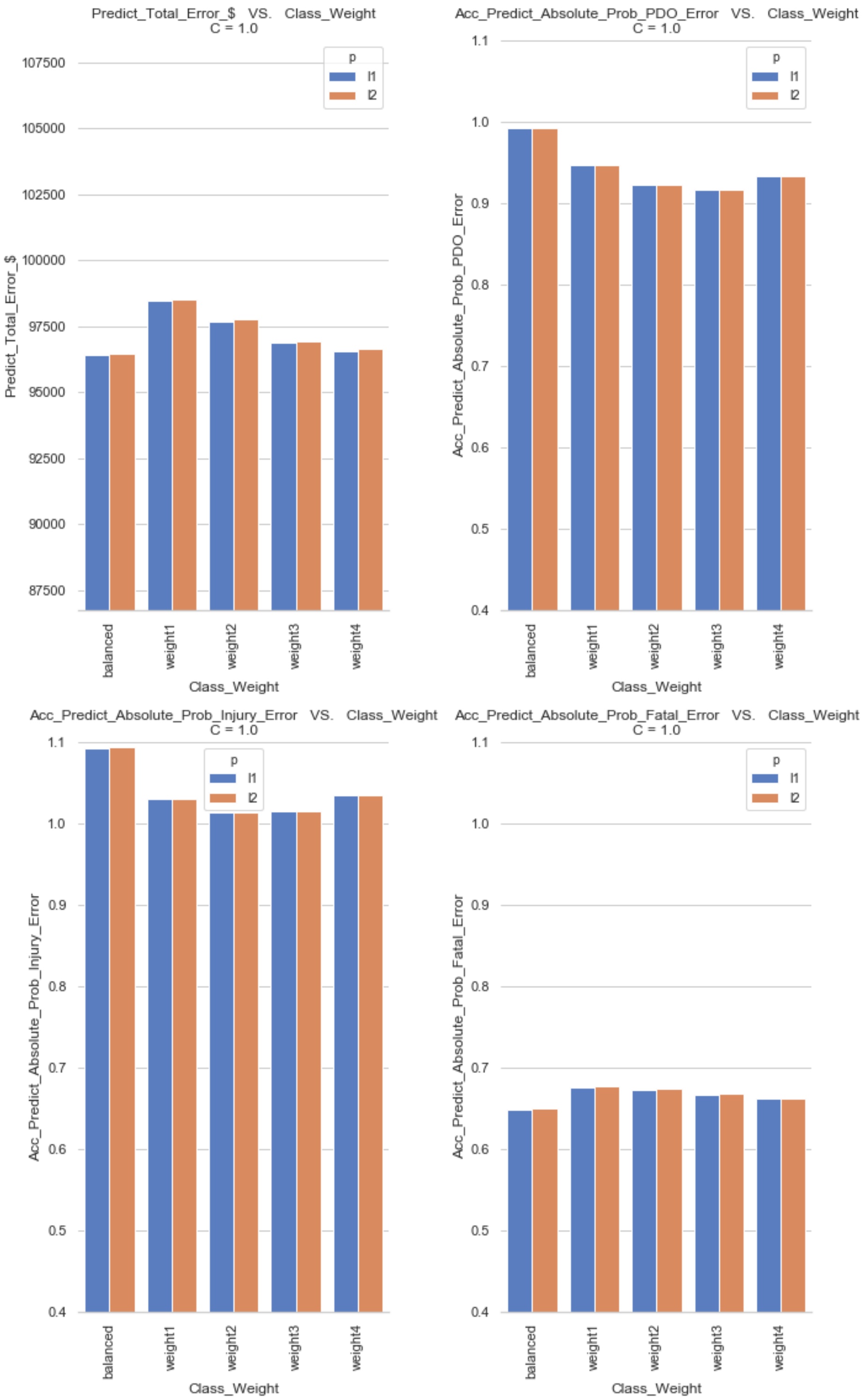

FIGURE 5.7: RMS Per Accident Absolute_Prob_Error and RMS_Absolute_Total_DC_Error with C=1.0 vs. class_weight

Effects on $RMS\ Per\ Accident\ EXP\_Prob\_Error\_Level_i$ Equation 5.8 and $RMS\_EXP\_Total\_DC\_Error$ Equation 5.12 by modifying the class_weight and set $C = 1.0$ are as Figure 5.8.

$RMS\_EXP\_Total\_DC\_Error$ for each level is plotted as "Predict_Total_Error_$".

$RMS\ Per\ Accident\ EXP\_Prob\_Error\_Level_i$ are plotted as "Acc_Predict_EXP _Prob_Level_Error".

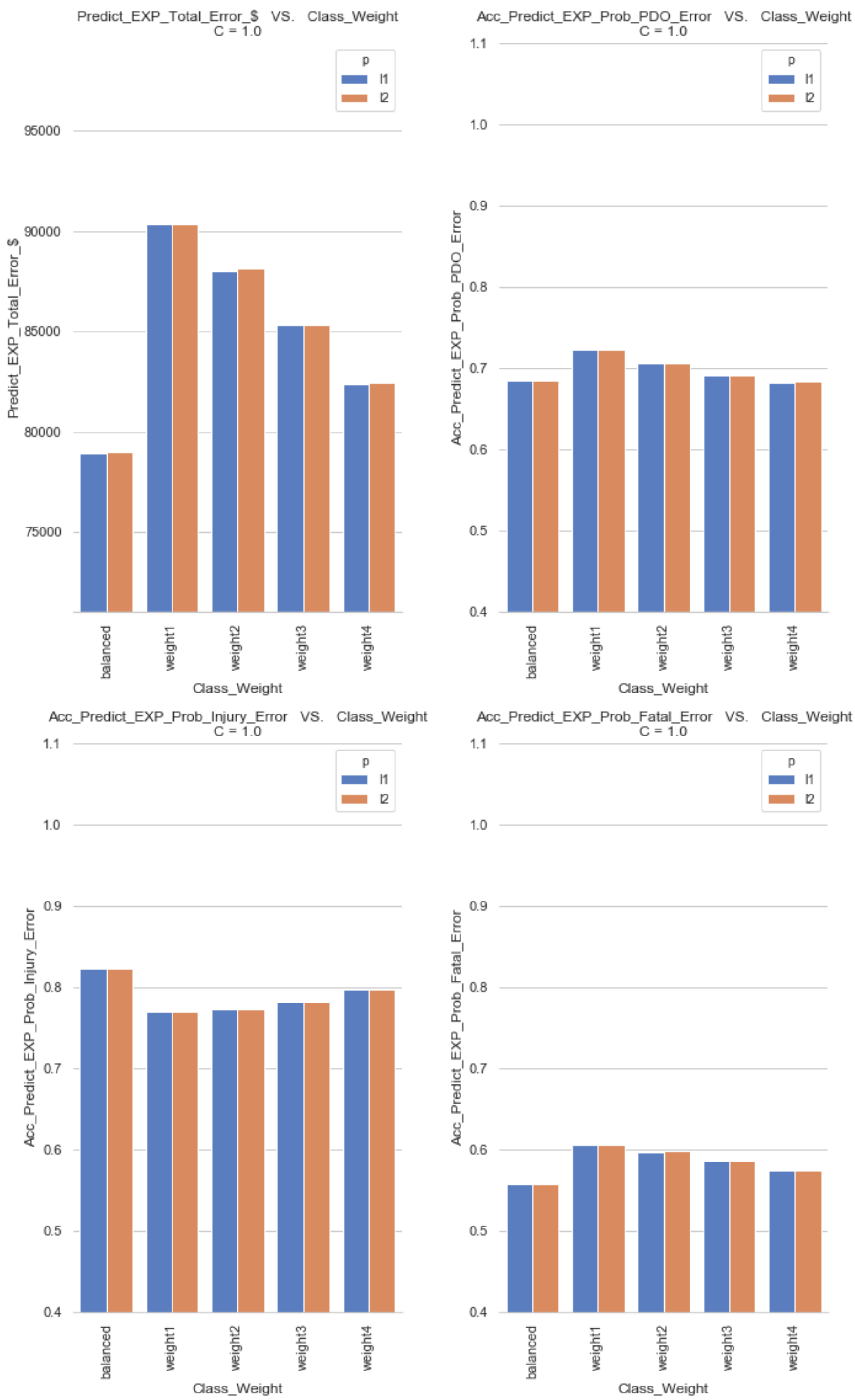

FIGURE 5.8: RMS Per Accident EXP_Prob_Error and RMS_EXP_-Total_DC_Error with C=1.0 vs. class_weight

5.2.4 Number of Coefficients of MLR

As shown in Table 4.2, there are in total 90 explanatory attributes for *MLR*, which result in 91 coefficients (include the intercept) for each Injury Level. This means the maximum numbers of the coefficient of an MLR model will be $3 \times 91 = 273$. Using different Regularization methods (l1: Lasso or l2: Ridge) and assigning different C values, the resulting coefficients number is as Figure 5.9.

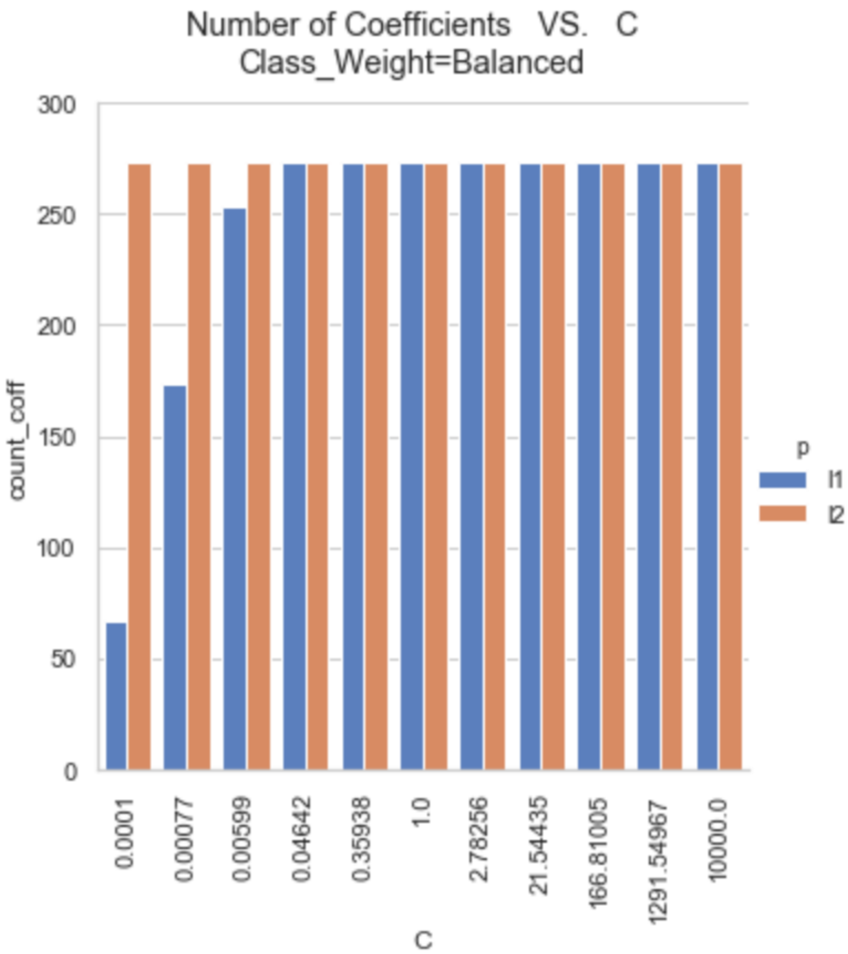

Figure 5.9: Number of Coefficients in the MLR model with class_weight = balanced VS. C

5.3 MLR model Analysis and Selection

5.3.1 Choice of Regularization Method and Parameter C

All the evaluation metrics show the only small difference between using l1: Lasso and l2: Ridge Regularization methods except for when C=0.0001. When C=0.0001, l2 has lower values for Probability Error Per Victim, Probability Error per Accident and Total Direct Cost Error Per Accident; higher accuracy, higher Recall_Injury, and higher Recall_Fatal. This means when C is too small, Lasso Regularization over regularizes many coefficients to 0 which reduces the predict performance of the model. From Figure 5.9 we can see that l1 regularization could reduce the number of coefficients in the model which could lead to a faster computation of prediction and improve the response efficiency of the entire Decision-Making framework when $C \leq 0.35981$. Comparing through all the evaluation metrics, we could see that when using l1: Lasso regularization with C=0.00077426368(second smallest C), the number of coefficients of the MLR model is reduced to the best efficiency without reducing the performance of the prediction.

5.3.2 Choice of class_weight

From Figure 5.1 we can see that as r rises from 1 to 1.8, the accuracy Recall_PDO decrease and Recall_Injury increase which means that the ability of the model correctly predicts Injury Case as 'Injury' improved. At the same time, comparing r =1.6(weight3) with r=1(balanced), the Probability Error Per Victim and Probability Error per Accident for Injury Level 'Injury' both decreases which means a more accurate prediction. However, r=1(balanced) behave the best when I used Total Direct Cost Error Per Accident to present the cost of the collision. From data distribution over 1999-2017, we could see that 'Injury' is the most common result for victims in a collision, thus we would like to choose a model that could predict 'Injury' accurately with low probability error. To meet the above expectation, I chose 'r=1(balanced)' when *Total Direct Cost Per Accident* is used as the cost evaluation metric for making the decision and ‘r=1.6(weight3)’ when ‘Level of Injury Probability’ is used to make decisions.

5.3.3 Choice of Using Predicted Probability of Severity Levels and Absolute Severity Level

From Figure 5.3 Figure 5.4 we can see that using Predicted Probability of Severity Level the resulting EXP_error is always lower than using Absolute Severity Level which gives Absolute_error for all C values and class_weights on the single victim level. Comparing Equation 5.7 with Equation 5.8 and comparing Figure 5.7 with Figure 5.8, we can see that on an collision level, using Predicted Probability also produce lower RMS errors for all C values and class_weights. Thus, on average, using Predicted Probability of Severity Levels to calculate the predicted result(either in the probability of a severity level or in *Total Direct Cost Per Accident*) will be more accurate than using the absolute predicted severity level directly.

5.3.4 Parameters of the selected MLR models and their prediction performance

Table 5.4: Selected MLR model parameters and performance

MLR Model Number	1	2
Model parameters		
panalty	l1	l1
C Value	0.00077426	0.00077426
Solver	SAGA	SAGA
r	1(balanced)	1.6(weight3)
class_weight[w1,w2,w3]	[0.85299, 0.55398, 44.376]	[0.60928, 0.63312, 50.715]
Number of Coefficients (including intercept)	178	161

Application Condition (cost evaluation metric)	Total Direct Cost Per Accident	Level of Injury Probability
Model Performance		
Accuracy(%)	57.96%	59.14%
PDO Recall(%)	71.43%	50.73%
Injury Recall(%)	49.39%	64.30%
Fatal Recall(%)	69.62%	70.31%
RMS Per Victim Absolute Probability Error PDO	0.5742	0.5610
RMS Per Victim EXP Probability Error PDO	0.4475	0.4479
RMS Per Victim Absolute Probability Error Injury	0.6336	0.6220
RMS Per Victim EXP Probability Error Injury	0.5150	0.4998
RMS Per Victim Absolute Probability Error Fatal	0.3308	0.3400
RMS Per Victim EXP Probability Error Fatal	0.2685	0.2820
Average Number of Victims in an Accident(Collision)	2.23	2.23
RMS Per Accident Absolute Probability Error PDO	0.9897	0.9268
RMS Per Accident EXP Probability Error PDO	0.6826	0.7079

RMS Per Accident Absolute Probability Error Injury	1.0913	1.0165
RMS Per Accident EXP Probability Error Injury	0.8220	0.7722
RMS Per Accident Absolute Probability Error Fatal	0.6500	0.6727
RMS Per Accident EXP Probability Error Fatal	0.5607	0.6004
RMS Absolute Total DC Error($)	96560.06	97870.49
RMS EXP Total DC Error($)	79382.78	88748.06

5.4 Decision Tree Classifier(DTC) Prediction Performance Evaluation

$sklearn.tree.DecisionTreeClassifier$ is the package used for developing DTC model. Scikit-learn uses an optimized version of the CART(Classification and Regression Trees) algorithm for model developing and in our application, only the classification is performed in the model training process for prediction. There are several parameters Table 5.5 that can be tuned to customize the result of the prediction.

Table 5.5: Decision Tree Classifier Hyperparameters

Parameter Name	Parameter Description	Option used
class_weight [33]	Small Weight: Less importance, lower impact on node Purity or Entropy Large Weight: More importance, higher impact on node purity or Entropy	same as for MLR Table 5.3 Table 5.2
criterion	Attribute Selection Measures	'gini' or 'entropy'
max_depth	The maximum depth of the tree.	from 1 to 30

5.4.1 Prediction Accuracy and Recall Rate

Effects on Accuracy, Recall Rate of modifying the max_depth, *split criterion* and set $class_weight = balanced$. Both the Accuracy and Recall Rate for training and testing are plotted.

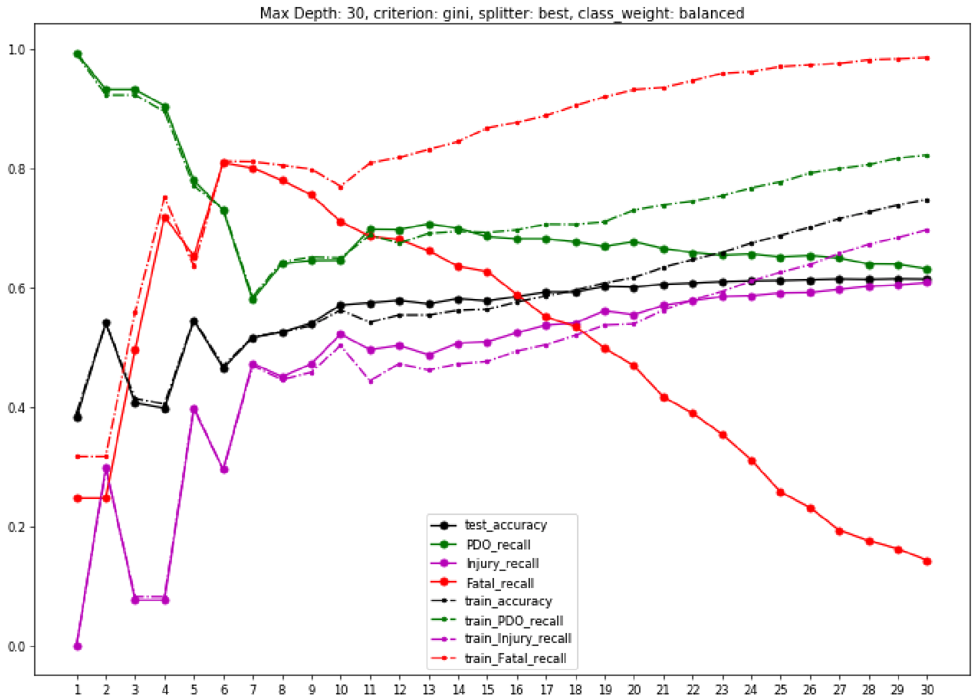

FIGURE 5.10: Accuracy and Recall Rate with class_weight = 'balanced',splitting criterion = 'gini' vs. max_depth

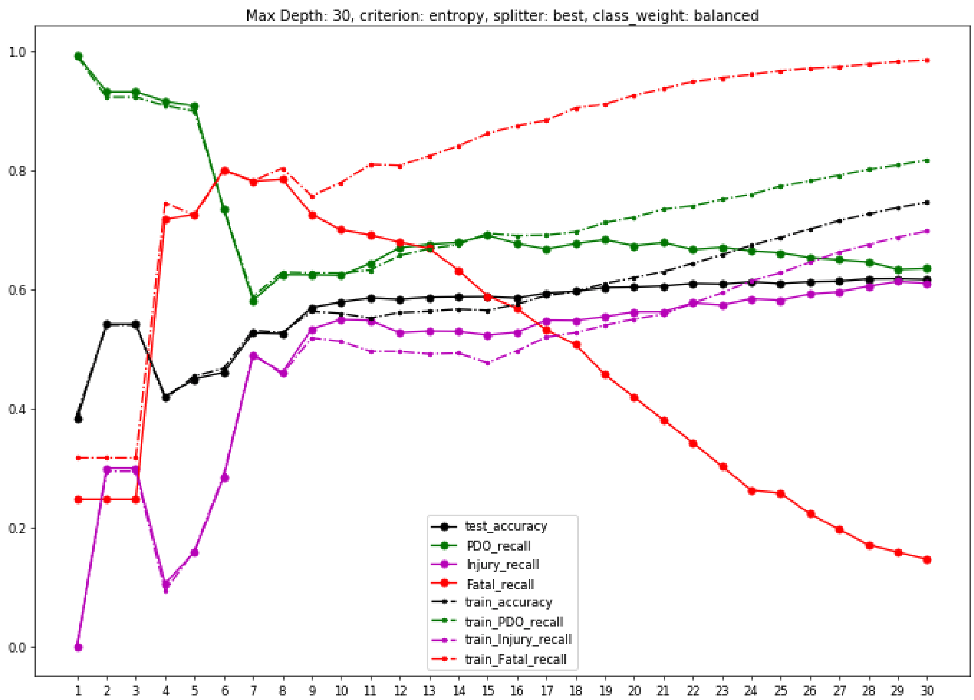

FIGURE 5.11: Accuracy and Recall Rate with class_weight = 'balanced',splitting criterion = 'entropy' vs. max_depth

5.4.2 other evaluation metrics

As we can see from the d est and Training start to split after $max\_depth = 8$. Splitting indicates an over-fitting. The 'entropy' splitting method increases the accuracy and recall rates of the decision tree more smoothly as depth increases compared to the 'gini' method. However, the eventual result of both splitting criterion is close although the 'gini' splitting train the model a lot faster. So 'gini' splitting criterion is always chosen when there is no significant difference in prediction performance.

First I set $max\_depth = 8$, I compared different evaluation metrics for different $class_weight$. After comparison, I found that as r increased ($class\_weight$ from 'balanced' to 'weight4') PDO recall rate decreased and Injury recall rate increased Figure 5.12. PDO and Injury recall rates are what we care more about in severity prediction. So I chose ‘weight4’ as *class_weight* and ‘gini’ as ‘criterion (ASM)’ for DTC.

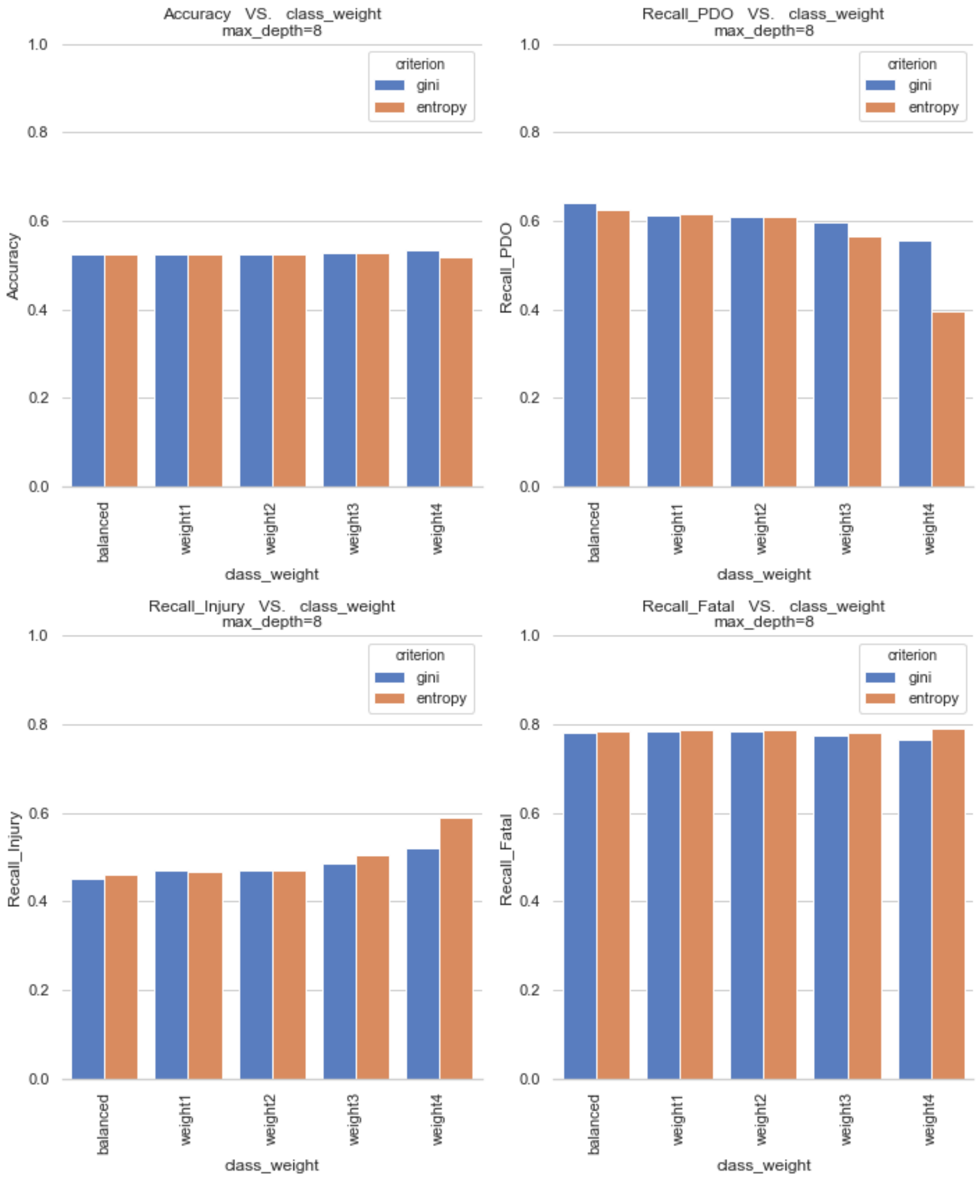

FIGURE 5.12: Accuracy and Recall Rate with max_depth = 8 vs. class_-weight

After selecting ‘weight4’ as *class_weight*, I compared different evaluation metrics for *max_depth* of the tree and found that *max_depth* = 15 gives an acceptable results for both ‘fatal’ and ‘injury’ recall rate Figure 5.13 as well as a relatively low

EXP_Prob_Error for all 3 levels as in Figure 5.14. When $max\_depth = 30$ the monetary cost error for both $RMS\ Per\ Victim\ Absolute\_Prob\_Error\_Level_i$ Figure 5.15 and $RMS\ Per\ Victim$
$EXP\_Prob\_Error\_Level_i$ Figure 5.14 goes to lowest.

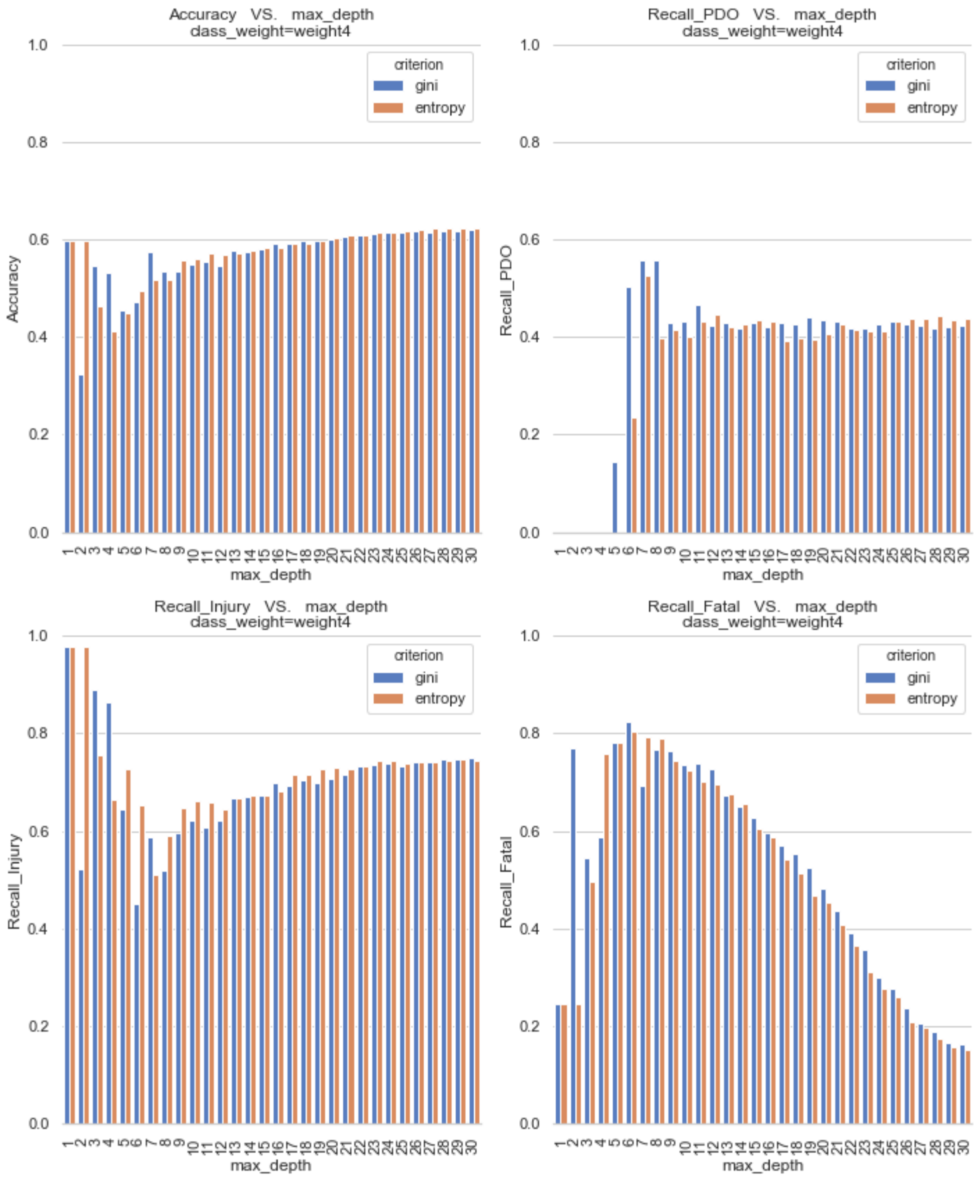

FIGURE 5.13: Accuracy and Recall Rate with class_weight = weight4 vs. max_depth

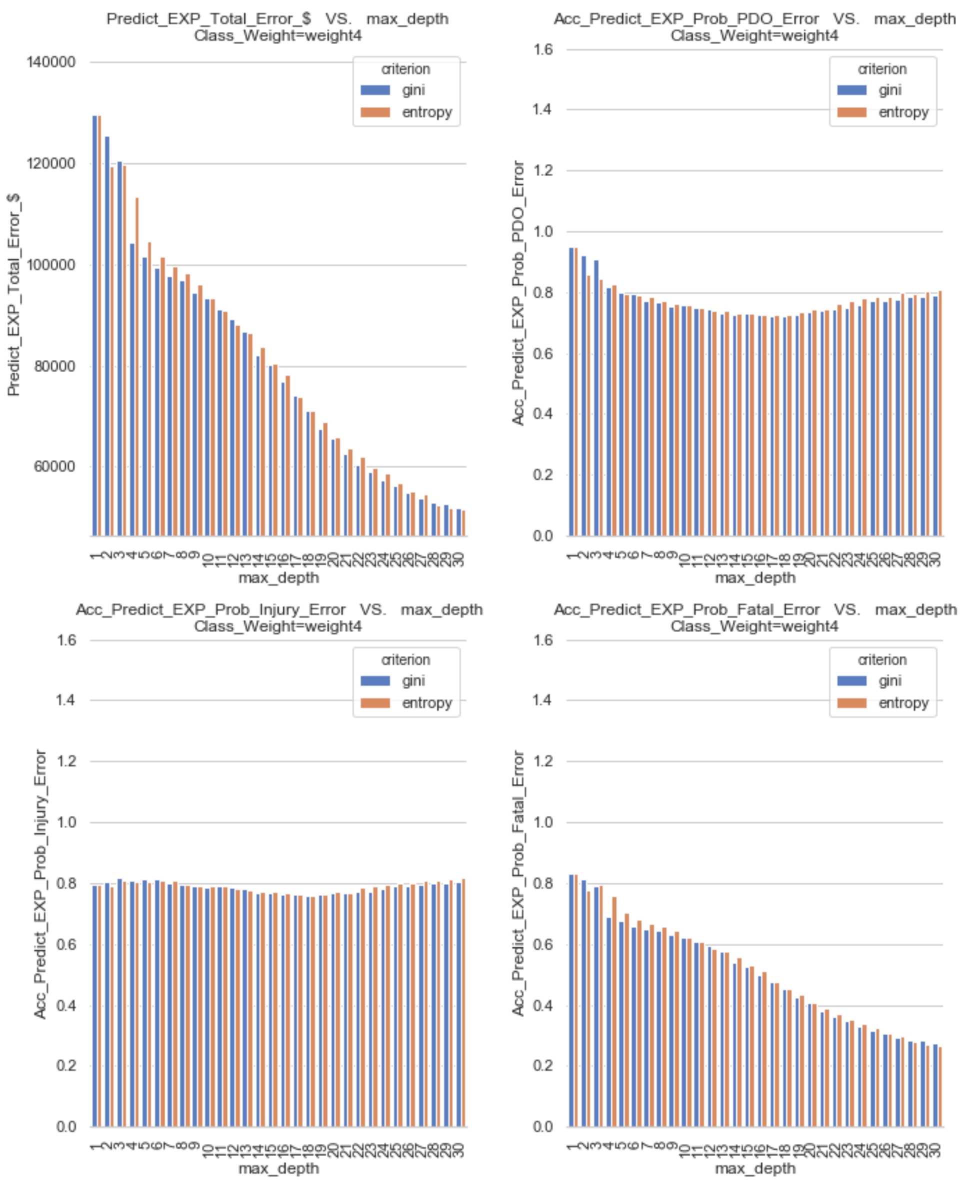

FIGURE 5.14: RMS Per Accident EXP_Prob_Error and RMS_EXP_-Total_DC_Error with class_weight = weight4 vs. max_depth

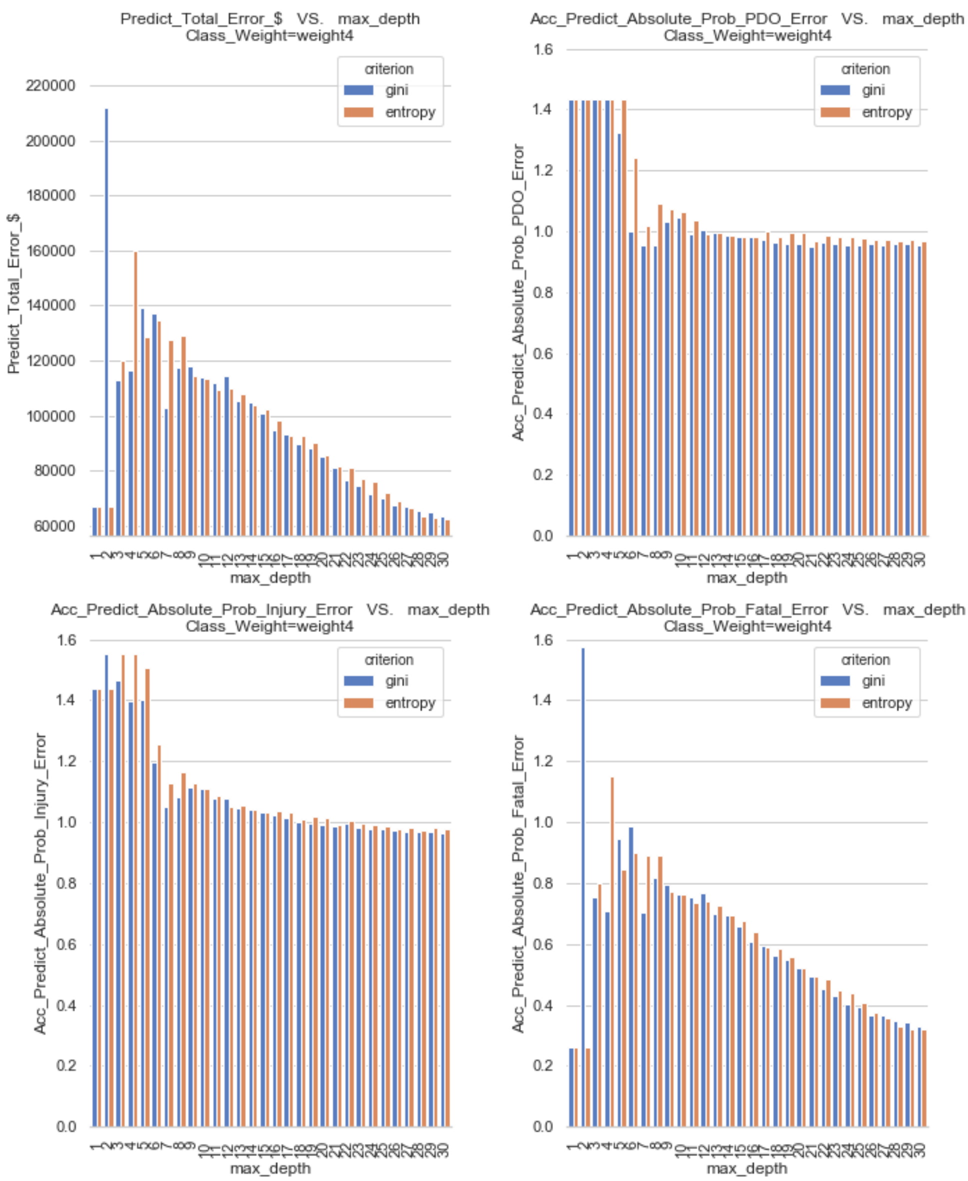

FIGURE 5.15: RMS Per Accident Absolute_Prob_Error and RMS_-Absolute_Total_DC_Error with class_weight = weight4 vs. max_depth

5.4.3 Parameters of the selected DTC models and their prediction performance

Table 5.6: Selected DTC model parameter and performance

DTC Model Number	1	2
Model Parameter		
criterion	gini	gini
max_depth	15	30
splitter	best	best
r	1.8(weight4)	1.8(weight4)
class_weight[w1,w2,w3]	[0.556299, 0.650323, 52.09301]	[0.556299, 0.650323, 52.09301]
Application Condition (cost evaluation metric)	Level of Injury Probability	Total Direct Cost Per Accident
Model Performance		
Accuracy(%)	57.92%	61.98%
PDO Recall(%)	43.04%	42.26%
Injury Recall(%)	67.20%	74.87%
Fatal Recall(%)	62.85%	16.32%
RMS Per Victim Absolute Probability Error PDO	0.5668	0.5610
RMS Per Victim EXP Probability Error PDO	0.4515	0.5079
RMS Per Victim Absolute Probability Error Injury	0.6275	0.5893
RMS Per Victim EXP Probability Error Injury	0.4923	0.5229

RMS Per Victim Absolute Probability Error Fatal	0.3558	0.2070
RMS Per Victim EXP Probability Error Fatal	0.2677	0.1696
Average Number of Victims in an Accident(Collision)	2.23	2.23
RMS Per Accident Absolute Probability Error PDO	0.9840	0.9558
RMS Per Accident EXP Probability Error PDO	0.7288	0.7909
RMS Per Accident Absolute Probability Error Injury	1.0346	0.9634
RMS Per Accident EXP Probability Error Injury	0.7683	0.8034
RMS Per Accident Absolute Probability Error Fatal	0.6579	0.2731
RMS Per Accident EXP Probability Error Fatal	0.5233	0.3292
RMS Absolute Total DC Error($)	100756.46	63345.65
RMS EXP Total DC Error($)	80088.12	51921.27

5.5 Cost Evaluation Metrics for Decision Making

There are four different cost evaluation metrics that can be used for Cs in Equation 3.45 and Equation 3.46 as showed in Figure 2.1 'Cost Evaluation Criteria'.

1 Sum of Absolute Probability of Fatal or Injury (minimize sum of Fatal First, then Sum of Injury, then Sum of PDO) in the primary vehicle (ethical egoism)Equation 3.36 or all involved vehicles (utilitarianism)Equation 3.35.

2 Sum of Predicted Probability of Fatal or Injury (minimize sum of Fatal First, then Sum of Injury, then Sum of PDO) in the primary vehicle (ethical egoism)Equation 3.38 or all involved vehicles (utilitarianism)Equation 3.37.

3 $Predicted\_Absolute\_Total\_DC$ of primary vehicle (ethical egoism)Equation 3.41 or all involved vehicles (utilitarianism)Equation 3.40 based on the ethical foundations used.

4 $Predicted\_EXP\_Total\_DC$ of primary vehicle (ethical egoism)Equation 3.44 or all involved vehicles (utilitarianism) Equation 3.43 based on the ethical foundations used.

In the decision-making process, the same predicted cost might appear for different control options. If the same cost predicted after using the user-selected ethical foundation, we will use the other left ethical foundation for further decision making, if still the same, we will choose the first choice available in the program order. However, if this happens once, we will count it as one $random\_choice$ made. In the decision-making process, our goal is to minimize the $random\_choice$ by using different cost evaluation metrics and severity prediction models(but not by changing different ethical bases as that is determined by user preference).

5.6 Collision and Decision Framework Simulation

The entire simulation process for Collision Environment Formation(generation) are shown in Figure 5.17. Decision making simulation is programmed in Python with logic as the flowchart Figure 5.16.

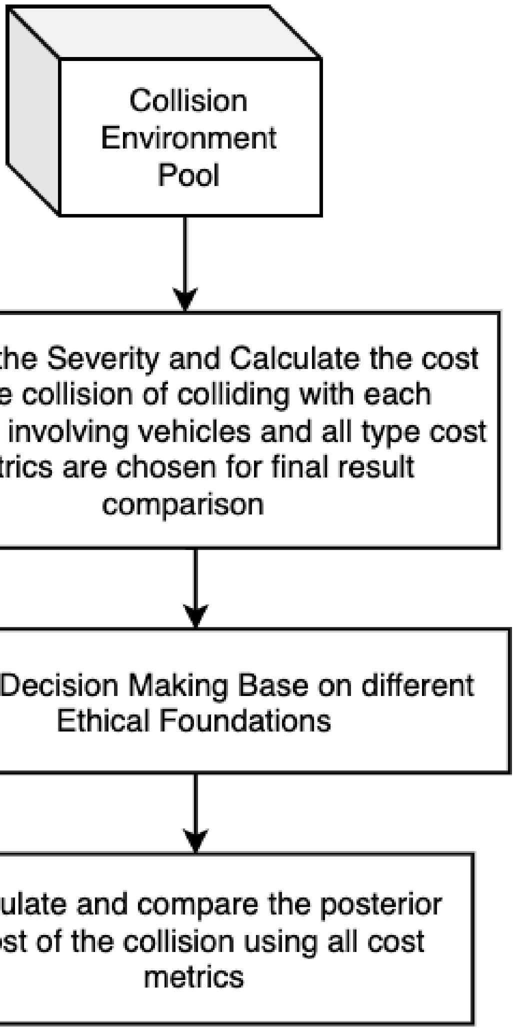

Figure 5.16: Decision Making Simulation

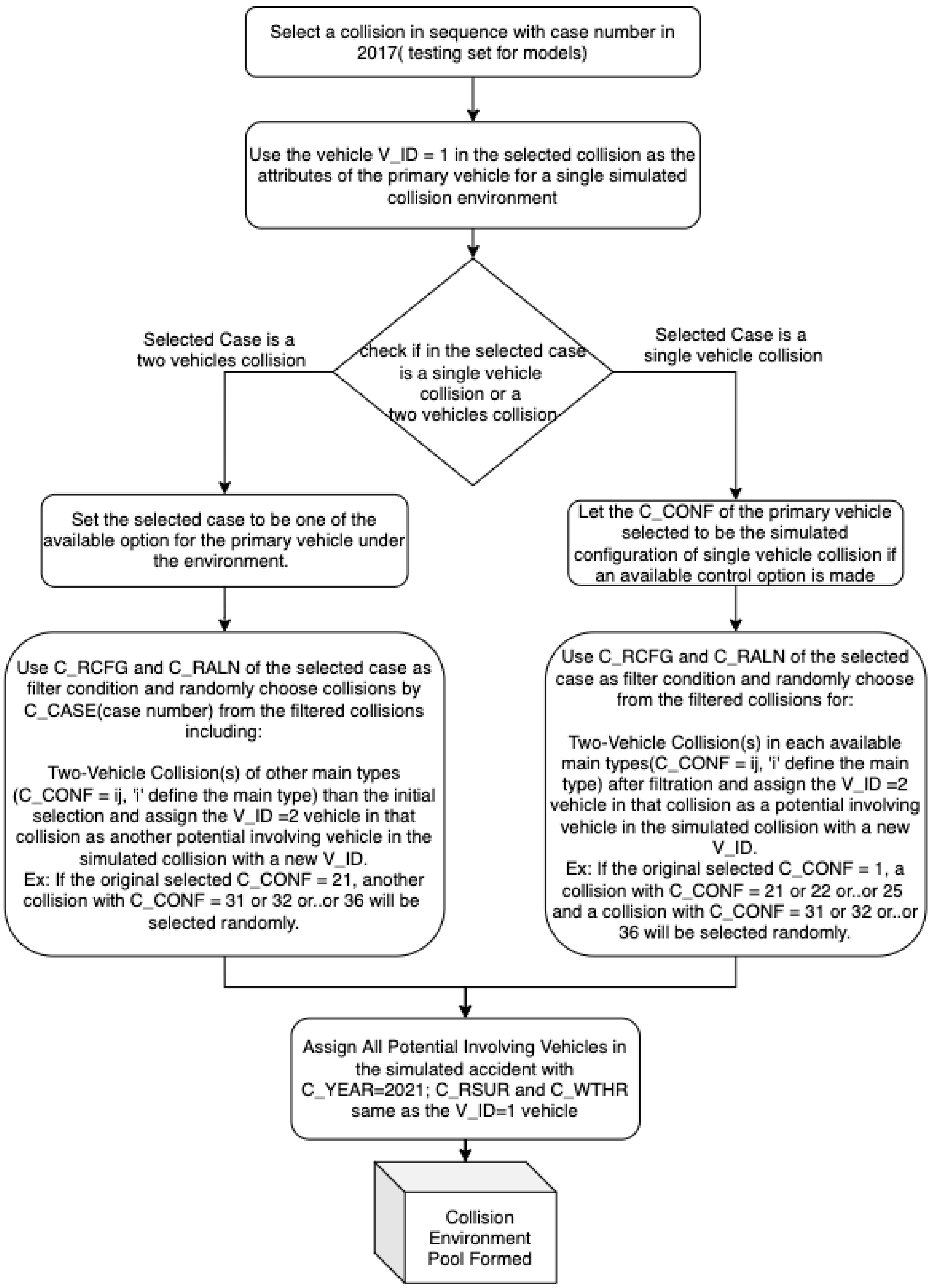

FIGURE 5.17: Collision Environment Generation

5.6.1 Simulation Parameters

Following the procedures in Figure 5.17 and Figure 5.16, the statistics of all Pre-Collision Environments generated from NCDB 2017 are given in Table 5.7.

Table 5.7: All Simulated Pre-Collision Environments Parameters

Number of pre-collision environments simulated (= Number of collisions simulated)	37,662
Number of people involved in all pre-collision environments simulated	226,550
Number of vehicles involved in all pre-collision environments simulated	112,931

5.6.2 Comparison between Decision Making Result using MLR and DTC

From the simulation, the DTC_1 and DTC_2 models in Table 5.6 used a lot of *random_choice* when making the decision. This means both DTC models are not able to differentiate the costs of resulted collisions from different options in pre-collision environments. DTC_1 with less depth has a smaller number of *random_choice* needed compared to DTC_2. This is because the *max_depth* of the DTC model is set to 15(DTC_1) and 30(DTC_2) to reduce the prediction error and increase the accuracy and recall rates. However, when the depth of the tree is higher, the tree is more likely to split until every leaf is pure, which means the probability of sample belongs to a category equals 1. In this case, both 'absolute' and 'EXP (probability)' cost evaluation methods lead to a similar result as 'absolute'. This will be beneficial only if the accuracy of the prediction model is 100%, which means it never goes wrong. When the Severity prediction model exist inaccuracy and errors, we would prefer a model that could deal with this uncertainty and make the decision with the existence of this uncertainty.

The MLR_1 and MLR_2 models in Table 5.2 also have some *random_choice* cases. These *random_choice* arise most when using *Predicted_Absolute_Total_DC* (Equation 3.40) of the entire collision (utilitarianism) or *Predicted_Absolute_Prim_DC* (Equation 3.41) of the primary vehicle (ethical egoism). Using $aEXP\_Acc\_total\_Level_i$ with MLR_1 model and $Predicted\_EXP\_Total\_DC$ (Equation 3.43) with MLR_2 model could eliminate the number of $random_{c}hoice$ happening to 0. The result of model and evaluation metrics comparison and resulting in *random_choice* are shown in Figure 5.18.

Thus, MLR will be used over DTC when performing decision making to avoid *random_choice* from happening and at the same time keeping acceptable accuracy, recall rates and other errors of the Injury Severity prediction.

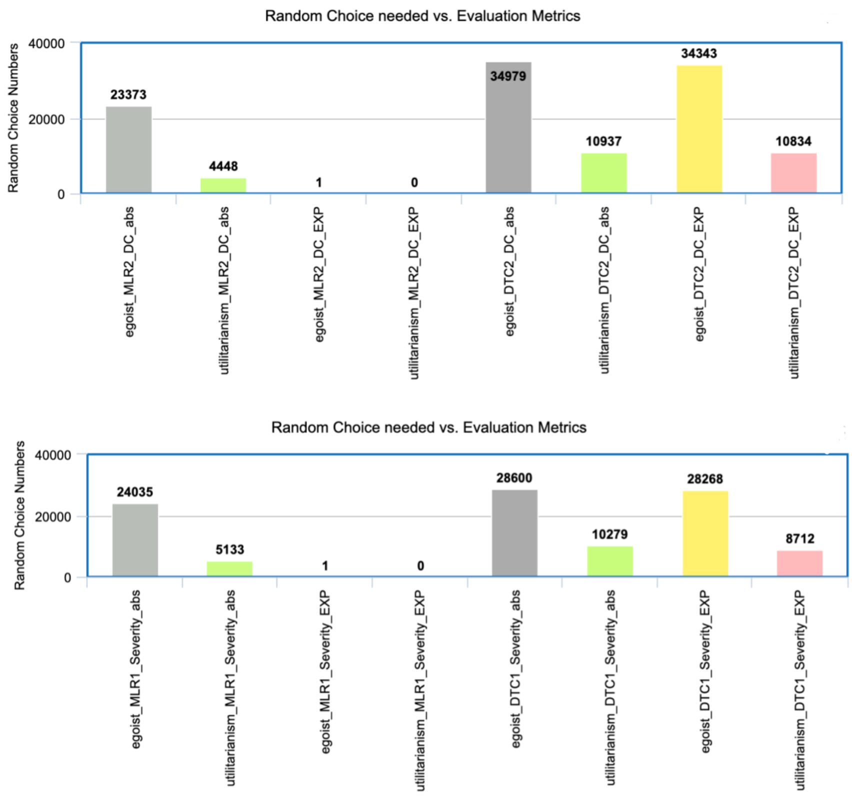

FIGURE 5.18: Random Choice vs. MLR and DTC using DC evaluation

5.6.3 Simulation Result Using Injury Severity Cost

Table 5.8: Simulation Result Using Injury Severity Cost with MLR1

Evaluation Metrics Predicted Result	ethical_egoism _Severity_abs	utilitarianism_ Severity_abs	ethical_egoism _Severity_EXP	utilitarianism _Severity_EXP
Number of people involved in collisions	94837.0	92170.0	105261.0	102472.0
MLR1_abs_Acc_total_PDO	60212.0	**68587.0**	76049.0	73925.0
MLR1_abs_Prim_total_PDO	**35872.0**	33384.0	35767.0	35149.0
MLR1_EXP_Acc_total_PDO	47657.47	49145.92	59789.50	**58224.86**
MLR1_EXP_Prim_total_PDO	25320.90	24624.98	**27976.10**	27514.56
MLR1_abs_Acc_total_Injury	31351.0	**22222.0**	27499.0	27069.0
MLR1_abs_Prim_total_Injury	**12805.0**	15248.0	12910.0	13396.0
MLR1_EXP_Acc_total_Injury	37056.95	34628.74	39474.71	**38634.10**
MLR1_EXP_Prim_total_Injury	19071.14	19613.49	**18738.97**	18941.68
MLR1_abs_Acc_total_Fatal	3274.0	**1361.0**	1713.0	1478.0
MLR1_abs_Prim_total_Fatal	**1246.0**	1291.0	1246.0	1378.0
MLR1_EXP_Acc_total_Fatal	10122.59	8395.33	5996.80	**5613.04**
MLR1_EXP_Prim_total_Fatal	5530.96	5684.53	**3207.93**	3466.77

Based on the simulation results shown in Table 5.8, the highlighted cells are the cost evaluation criteria of the selected cost evaluation metrics (columns' names). The priority is shown as the darkest color to the brightest. As all these outputs come from the prediction of the same model (MLR2) on the same pre-collision environments, the different results in each cell are a consequence of different cost metrics and different ethical foundations for decision making. The minimum number of people involved in collisions is a result of using utilitarianism with absolute Severity Level prediction. However, this evaluation metric leads to a high number of Expected Accident Fatal victims. This means if the prediction recall rate in victims' severity level 'Fatal' is very high (much higher than 69.92% which is current recall rate in MLR1), the absolute severity level cost metrics will be more preferable and accurate than it is right now.

5.6.4 Simulation Result Using Monetary Cost

Table 5.9: Simulation Result Using Monetary Cost with MLR2

Evaluation Metrics Predicted Result	ethical_egoism _DC_abs	utilitarianism _DC_abs	ethical_egoism _DC_EXP	utilitarianism _DC_EXP
Number of people involved in in collisions	92,327	87,658	105,274	93,559
MLR2_abs_DC_Victims	1,870,099,263.0	1,842,602,525.0	2,225,969,976.0	2,050,156,531.0
MLR2_abs_DC_Acc_Type	1,176,678,362.0	1,172,954,271.0	1,207,456,692.0	1,209,890,401.0
MLR2_abs_DC_Total	3,046,777,625.0	**3,015,556,796.0**	3,433,426,668.0	3,260,046,932.0
MLR2_abs_Prim_DC_Victims	**1,092,606,050.0**	1,220,347,505.0	1,097,470,520.0	1,204,915,126.0
MLR2_EXP_DC_Victims	2,921,597,126.9	2,914,904,088.4	2,893,980,979.0	2,691,356,392.0
MLR2_EXP_DC_Acc_Type	1,515,313,341.2	1,538,168,715.7	1,424,965,745.4	1,429,273,982.1
MLR2_EXP_DC_Total	4,436,910,468.1	4,453,072,804.0	4,318,946,724.4	**4,120,630,374.1**
MLR2_EXP_Prim_DC_Victims	1,727,505,215.3	1,867,526,937.3	**1,420,263,598.5**	1,573,554,018.6

Base on the simulation results shown in Table 5.9, the bold text cells are the cost evaluation criteria of the selected cost evaluation metrics(columns' names). Using Monetary cost, compared with Table 5.8, the number of people involved in collisions is mostly reduced for each corresponding cost evaluation metrics(Except for ethical_egoism_DC_EXP where DC and severity have only 13 people difference). This proves that using DC as a cost evaluation metric for making a decision under the same environments will eliminate the number of people involved in the collision.

Chapter 6

Summary

6.1 Discussion

This thesis presents a complete Decision-Making Framework for autonomous vehicles facing inevitable collisions that involves dilemma inducing situation. In the result and simulation section, the collision injury severity prediction performance was compared using both the MLR model and a DTC. We can see that the accuracy and precision rate of using both models can reach a similar value by adjusting the hyperparameters of the models. However, none of the models can reach a perfect prediction of a 100% accuracy. In the simulation, we can see that due to the trade-off between prediction performance and *random_choice* of the DTC model, under a non-perfect prediction situation, the MLR model is more suitable for severity prediction for a cost evaluation for further decision making. Because of the lower number of coefficients, higher recall rate for 'Injury' and 'Fatal' Level prediction, and a lower probability error, MLR_2 in Table 5.2 is a more preferable model for CISLP among all 4 models presented in this thesis.

Compared the cost evaluation criterion, the EXP cost evaluation method is more preferable for non-perfect prediction models than the abs method. However, if the prediction precision can be improved to a very high value(over 95%) especially for Injury and Fatal severity levels, the abs method will become more preferable for cost evaluation in general. Using the monetary cost is actually assigning different weights of importance to three different severity levels when making the decision. The monetary cost is more realistic and easier to quantify the collision cost for decision making comparing to severity level cost. However, there are shortcomings of using monetary cost such as the inappropriate of simply using money value to quantify the injury of victims and loss of human lives. At the same time, this monetary cost evaluation method requires a high accuracy as well as publicly acceptable money values used to quantify the collision cost.

This decision making framework is the first framework designed allowing different ethical bases to be used based on realistic(but not necessarily publicly acceptable) cost predictions of collisions. The simulation result shows obviously that different ethical bases will lead to different decisions by the autonomous vehicle when facing a dilemma inducing situation. Using Utilitarianism generally results in less victims in the collisions and less cost of the entire collision corresponding to different cost evaluation metrics used in the decision-making. However, ethical egoism minimize the cost of the primary vehicle in all collisions accordingly with the price of a higher number of victims involved in collisions and higher total cost of each and all collisions. This proves the urgent need of incorporating human ethical bases to all types of decision making of the autonomous vehicle.

Based on the above discussion, among the four CISLP models and four cost evaluation metrics for decision making, the best combination is in Table 6.1. This combination have an fairly good accuracy and recall rate, minimize the probability error, have the least coefficient numbers and leads to least random choices in simulation.

Table 6.1: optimal model + cost evaluation metric combination

Model type and number	MLR_2
Panalty	l1
C value	0.00077426
Solver	SAGA
r	1.6 (weight3)
class_weight[w1,w2,w3]	[0.60928, 0.63312, 50.715]
Number of Coefficients (including intercept)	161
Cost Evaluation Metric	Predicted_EXP_Total_DC

6.2 Future Work

Future works can be done from three aspects to improve the applicability of this thesis. First, attributes data like relative speed between vehicles should be collected and added to both generating of the prediction models as well as the simulation to improve the accuracy and applicability of the framework. If other attributes other than macro-environment ones are used, the effect of delay in execution of vehicle's control on final decision and consequence should be carefully researched. An algorithm should be built to solve the situation when not all attributes data are available to the CISLP, prediction and further decision-making process should still be able to perform. The effect of missing explanatory attributes and incorrect input attributes on prediction result, evaluated cost and final decision could be discussed more explicitly. Secondly, other machine learning and statistical methods like Bayesian, Neural Networks, Random Forest can be used for prediction model building to improve the severity prediction performance. The computation time of using different model for decision making can be evaluated and compared as respond time is a important aspect of the decision-making framework.

Thirdly, more human ethical bases and foundations like moral machines or virtue ethics or a dynamic ethical foundation system can be used on autonomous vehicles when making decisions to imitate the human decision-making process.

Bibliography

[1] NHTSA, *Automated Vehicles for Safety*, Sep. 2017.

[2] X. Yang, L. Liu, N. H. Vaidya, and F. Zhao, A vehicle-to-vehicle communication protocol for cooperative collision warning, 114–123, 2004.

[3] H. Guo, C. Shen, H. Zhang, H. Chen, and R. Jia, Simultaneous trajectory planning and tracking using an mpc method for cyber-physical systems: A case study of obstacle avoidance for an intelligent vehicle, *IEEE Transactions on Industrial Informatics*, vol. 14(9), 4273–4283, 2018.

[4] J. Funke, M. Brown, S. M. Erlien, and J. C. Gerdes, Collision avoidance and stabilization for autonomous vehicles in emergency scenarios, *IEEE Transactions on Control Systems Technology*, vol. 25(4), 1204–1216, 2016.

[5] C. Laugier and R. Chatila, *Autonomous navigation in dynamic environments.* Springer, 2007, vol. 1.

[6] T. Fraichard and H. Asama, Inevitable collision states—a step towards safer robots?, *Advanced Robotics*, vol. 18(10), 1001–1024, 2004.

[7] N. J. Goodall, Ethical decision making during automated vehicle crashes, *Transportation Research Record*, vol. 2424(1), 58–65, 2014.

[8] G. Mordue, A. Yeung, and F. Wu, The looming challenges of regulating high level autonomous vehicles, *Transportation Research Part A: Policy and Practice*, vol. 132, 174–187, 2020, ISSN: 0965-8564.

[9] J. E. Pickering, M. Podsiadly, and K. J. Burnham, A model-to-decision approach for the autonomous vehicle (av) ethical dilemma: Av collision with a barrier/pedestrian (s), *IFAC-PapersOnLine*, vol. 52(8), 257–264, 2019.

[10] Y. Liao, J. Zhang, S. Wang, S. Li, and J. Han, Study on crash injury severity prediction of autonomous vehicles for different emergency decisions based on support vector machine model, *Electronics*, vol. 7(12), 381, 2018.

[11] T. Canada, *National collisions database online, national collision database online 1.0*, Accessed on 2019-12-20, Nov. 2019.

[12] A. Iranitalab and A. Khattak, Comparison of four statistical and machine learning methods for crash severity prediction, *Accident Analysis & Prevention*, vol. 108, 27–36, 2017.

[13] A. T. Kashani and A. S. Mohaymany, Analysis of the traffic injury severity on two-lane, two-way rural roads based on classification tree models, *Safety Science*, vol. 49(10), 1314–1320, 2011.

[14] J. Abellán, G. López, and J. De OñA, Analysis of traffic accident severity using decision rules via decision trees, *Expert Systems with Applications*, vol. 40(15), 6047–6054, 2013.

[15] C. Lee and X. Li, Analysis of injury severity of drivers involved in single-and two-vehicle crashes on highways in ontario, *Accident Analysis & Prevention*, vol. 71, 286–295, 2014.

[16] M. S. Shaheed, K. Gkritza, A. L. Carriquiry, and S. L. Hallmark, Analysis of occupant injury severity in winter weather crashes: A fully bayesian multivariate approach, *Analytic methods in accident research*, vol. 11, 33–47, 2016.

[17] M. I. Sameen and B. Pradhan, Severity prediction of traffic accidents with recurrent neural networks, *Applied Sciences*, vol. 7(6), 476, 2017.

[18] F. Rezaie Moghaddam, S. Afandizadeh, and M. Ziyadi, Prediction of accident severity using artificial neural networks, *International Journal of Civil Engineering*, vol. 9(1), 41–48, 2011.

[19] M. Zheng, T. Li, R. Zhu, J. Chen, Z. Ma, M. Tang, Z. Cui, and Z. Wang, Traffic accident's severity prediction: A deep-learning approach-based cnn network, *IEEE Access*, vol. 7, 39 897–39 910, 2019.

[20] W. H. Greene and W. H. Greene, 7th. Boston, MA: Pearson, 2012, 803–806.

[21] *Multinomial logistic regression*, Wikipedia Version ID: 936282105, Jan. 2020.

[22] D. N. Schreiber-Gregory, *Regulation techniques for multicollinearity: Lasso, ridge,and elastic nets.*

[23] *Unsupervised feature learning and deep learning tutorial.*

[24] P. Li, *Logistic regression, department of computer science, rutgers university.*

[25] S. Luo, *Optimization: Loss function under the hood (Part II)*, Jun. 2019.

[26] A. Defazio, F. Bach, and S. Lacoste-Julien, Saga: A fast incremental gradient method with support for non-strongly convex composite objectives, in *Advances in neural information processing systems*, 2014, 1646–1654.

[27] R. Aruna, Construction of decision tree : Attribute selection measures, 2013.

[28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research*, vol. 12, 2825–2830, 2011.

[29] J. R. Quinlan, *Induction of decision trees*, 1986.

[30] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*, 2014.

[31] Collision cost study update final report, *Capital Region Intersection Safety Partnership(CRISP)*, Jun. 2019.

[32] T. Canada, *Canadian motor vehicle traffic collision statistics: 2017*, Accessed on 2019-12-20, Feb. 2019.

[33] K. M. Ting, An instance-weighting method to induce cost-sensitive trees, *IEEE Transactions on Knowledge and Data Engineering*, vol. 14(3), 659–665, 2002.