Can Machine Learning beat Physics at Modeling Car Crashes?

Gavin Byrne
Technological University Dublin, Ireland

Can Machine Learning beat Physics at Modeling Car Crashes?

Gavin Byrne

C06828931

A dissertation submitted in partial fulfillment of the requirements of

Dublin Institute of Technology for the degree of

M.Sc. in Computing (Data Analytics) 2018

Date

Declaration

I certify that this dissertation which I now submit for examination for the award of MSc in Computing (Stream), is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

This dissertation was prepared according to the regulations for postgraduate study of the Dublin Institute of Technology and has not been submitted in whole or part for an award in any other Institute or University.

The work reported on in this dissertation conforms to the principles and requirements of the institutes guidelines for ethics in research.

Signed:

Date:

Acknowledgments

I would like to thank my supervisor Sean O'Leary for his guidance and sharing of expertise throughout the this research project.

I would also like to thank my wife for all of her support during it and for giving me the inspiration to better myself.

Contents

List of Figures

List of Tables

List of Acronyms

AIC	**A**ikaike **I**nformation **C**riterion
AUC	**A**rea **U**nder **C**urve
BC	**B**ack **C**entre
CFC	**C**hannel **F**requency **C**lass
CRISP-DM	**CR**oss **I**ndustry **S**tandard **P**rocess -for- **D**ata **M**ining
FC	**F**ront **C**entre
HMM	**H**idden **M**arkov **M**odel
NHTSA	**N**ational **H**ighway **T**raffic **S**afety **A**dministration
NDI	**N**ational **D**amage **I**dex
PDOF	**P**rinciple **D**irection **of** **F**orce
RBF	**R**adial **B**ased **F**unction
SL	**S**ide **L**eft
SR	**S**ide **R**ight
SVM	**S**upport **V**ector **M**achine
VECG	**VE**hicle **C**entre -of- **G**ravity

Chapter 1

Introduction

1.1 Abstract

This study aimed to look at a traditional method used for measuring the severity and principle direction of force of a car crash and see if it could be improved on using machine learning models. The data used was publicly available from the NHTSA database and included descriptions of the vehicle, test and sensors as well as the accelerometer data over the period of the crashes. The models built were SVM classifiers and multinomial regression models. Although the SVM and Regression models were build successfully and gave higher levels of accuracy than the momentum models in terms of the severity, the traditional momentum model's severity results were not statistically significant and it was therefore impossible to say the SVM classifier was an improvement using the same data. The principle direction of force was improved on using both a multi-level SVM classifier and a multinomial regression and the results were statistically significant.

1.2 Background

The inspiration for this study came from a privately commissioned paper that was created for a company called Xtract by the University of Limerick. The company had pitched a Telematic business proposal where they would supply a computer system that would enable insurance claims managers to assess crashes in real time using sensor data contained in a black box in a car. Many insurance companies have rolled out similar Telematics policies where they install a black box that can read the accelerometer data in a car and then inform the claims manager in real time about a claim when it occurs enabling them to make sound decisions.[1].

The NHTSA has a massive database of over 164k sensors in over 5k tests ranging back to 1978. The crashes were uploaded to the test site by third parties that were interested in sharing their data which dates back to the late 1970s. These tests included vehicle structural tests, passenger impact assessments and vehicle to vehicle impact assessments. The sensors available in these tests read various measures including accelerometer data in various parts of the vehicle or crash test dummy.

There is a wealth of knowledge in Physics and Engineering about how to measure vehicle dynamics and the impact of forces on rigid bodies. This knowledge can be applied to various real world examples to create models that give insight to the surrounding world. In engineering, the traditional method for modelling car crashes is to use inductive models where there is no requirement to train the model. Therefore, accurate results can be output even if there is only data on one crash.

Machine Learning is a growing field where data is used to create an algorithm that can generalize to a larger population. Currently, a person interacts with machine learning algorithms regularly throughout their daily lives. For example, there are algorithms that suggest product advertising through Google searches, algorithms that recognise what a person says and others that predict what the weather will be tomorrow. The industry will only expand into ever more complex avenues of everyday life to the point that people will not have to make as many daily decisions, whether that is buying milk before it goes off or driving their cars. This study aimed to assess whether machine learning could be an improvement over the more traditional methods used in engineering when applied to car crash test data. This was driven by a real business need in the insurance industry and could hopefully act as a resource to this end.

[1] This current industry phenomenon is getting lots of coverage in today's media http://www.ft.com/content/894c3f5e-786c-11e7-a3e8-60495fe6ca71

1.3 Informal Description of the Problem

The database supplied c.5k examples where the vehicle centre of gravity sensor was used. All of these sensors had an x, y and z coordinate that showed the amount of g-force applied by millisecond. Currently, there are no similar studies using the same data which may be down to the difficulty of downloading the files. There was a need to identify the impact time of the crash using the max sum of the g-force in all directions. Once calculated, this data was added to the database and was applicable to all models. The momentum model aimed to show the impact angle (continuous variable) and severity (categorical) of crash using the impact time and the accuracy was assessed by checking the number of correct results against the database. Following this, an SVM was used to classify the crash severity. Regression was then used to assess the impact angle and it was compared to the physical models to see if there was any improvement in accuracy from one model to the next.

1.4 Research Project/problem

This research aimed to classify the severity of a crash using a inductive momentum model and compare the results to a deductive Support Vector Machine (SVM) classifier. These models were judged on the accuracy, specificity and sensitivity of the results. Secondly, the research aimed to calculate the principle direction of force (PDOF) felt on a vehicle during a crash using the same inductive model and then compare it to a deductive SVM classifier and a deductive multinomial regression model.

1.4.1 Research Question

"Can ML models match the accuracy of a momentum model for car crashes using force in x,y,z planes as the independent variables and the angle of impact or severity of crash as the dependent variables?"

1.5 Research Objectives

The primary goal of this research was to determine if machine learning could be applied in this area of engineering and see if it was possible for it to improve on the traditional techniques applied.

- Perform a comprehensive evaluation and analysis of the existing research relating to car crash mechanics and machine learning.
- Select the existing data set and investigate the scope and limitations of this data set.
- Build momentum model used in the privately commissioned paper.
- Assess the accuracy of the momentum model on the dataset.
- Select and build the machine learning models.
- Assess the accuracy of the machine learning models on the dataset.
- Analyse the results.
- Evaluate the statistical significance of the results.

This research was quantitative and empirical in nature. It relied on an existing dataset and as such, was a form of secondary research. Further, the goal of this research is to inform primary research. The experiment was designed to enable acceptance or rejection of the null hypothesis derived from the research question. Model performance was judged on the accuracy of the classifier against the ground truth contained in the database. The overall methodology could be considered deductive in nature as a number of different models were used to compare results even though the models used were both inductive and deductive. A literature review was done to ensure that any methods used had been in a similar capacity and also to ensure that the model machine learning model selection was suitable.

1.6 Scope and Limitations

The data selected for the experiment was the best available public datasource that contained the inputs for the models. The ground truth was based on an engineering classification called the National Damage Index (NDI) that was a subjective opinion so the decision boundary for any classifier could not be accurately defined. The types of tests that could be used in the momentum model significantly reduced the number of cases that could be used from the database which meant the accuracy from the severity was not statistically significant. The reduced bias in the PDOF however ensured conclusions could be drawn on the second hypothesis.

The fact that the data for each crash only showed a short time window and no other activity meant that a classifier could be considered a good choice. For exmpale, it would not classify a car slowing down as a low severity crash. This was the main advantage of the format of the data. The SVM classifier was a useful model that was implemented but if there were no time constraints, a Hidden Markov Model (HMM) would have been implemented to see if an unsupervised model could have classified the crash area automatically. Implementing a more complex inductive model and its improvement over the momentum model could have added more value. A spring model or momentum model based on the crush area may have yielded better results but the same data constraints would have applied so this type of study would need a better datasource.

1.7 Document Outline

The remaining chapters in this document are structured as follows.

- Chapter 2 begins describing a crash pulse and then explains the difference between a deductive and inductive model. It then examines the database and describes the concepts behind the momentum model, the SVM and Regression.
- Chapter 3 describes the design and methodology of the experiment. It follows the CRISP-DM structure beginning with an evaluation of the research problem

and formulation of the hypothesis before describing the data understanding and preparation phases. This chapter discusses how the models were built, evaluated and compared with one another. It concludes with a discussion on the strengths and limitations of the approach.

- Chapter 4 describes the implementation of all of the 50 experiments and aims to show much of the results and give a step by step account of how the experiments were conducted. It aims to identify some of the problems that occurred along the way and outline the different models that were run to counter some of these problems.

- Chapter 5 takes all of the results that were discovered in chapter 4 and brings them altogether. It takes the best performing models that were created and compares them to the momentum model in order for conclusions to be drawn and the research questions answered.

- Chapter 6 this chapter provides an overall evaluation of the research and experiment and contains suggestions for future work.

Chapter 2

Literature Review and Related Work

2.1 Introduction

This chapter gives some context to the study and aims to use the literature to reinforce the techniques used while ensuring that any previous knowledge can be learned from or added to. It opens with a background of the study itself which shows the inspiration for the thesis. Next, a description of the main elements to consider in a crash pulse are outlined followed by a comparison of the different types of models that are seen in the study. A section on the data follows this, with reference to some similar studies that also used it.

As the study had two different types of modelling techniques, the next section describes the deductive model selected, why it was selected and any other studies that use similar techniques. The section then finishes with a description of the model and publications that implemented similar models. The next section shows the reasons for selecting the deductive models, their description mathematically and some studies that used them. The chapter then finishes with a summary of the findings and what were done.

2.2 Background

According to (Kevin Brosnan, 2017) fraudulent insurance claims cost the Irish insurance industry over two hundred million dollars a year. That private report was tasked with assessing whether the angle of impact and the crash severity could be determined using the time series data that was available from accelerometer sensors fitted in cars. This information could then to be used as a prototype for a business pitch to insurance companies that helped assess insurance claims in real-time. As outlined in the paper, there was a need for more data that could be used to train machine learning models. This acted as the inspiration for this study.

Lots of research mentions that a key determinant of the severity of a crash is the PDOF. For example, in (Ryb, Dischinger, Kufera, & Burch, 2007) it was found that the higher the change in velocity, the higher the chances are of mortality. This was also coupled with a finding that the PDOF has a synergistic effect with the change in velocity on the mortality rate. This reinforced the opinion in this study that the momentum model selected should be accurate as the severity and PDOF are determined by the same model. In 2000, (M. Richter, Otte, Pohlemann, Krettek, & Blauth, 2000) mentioned that the severity of the impact was not an indicator of the length of time a person will complain of an injury but they did find that the PDOF (i.e. head on or rear end) collision did result in higher levels of whiplash. This again reinforced that the PDOF, if available, may be a useful factor in calculating the severity. Angled collision or the impact of a vehicle into the side door of a vehicle has also been shown to increase the chances of death or injury in a vehicle collision (Abdel-Aty & Abdelwahab, 2004). The two points in this paragraph may indicate why the NHTSA had such a high number of tests that are from the side or from the front or rear. This point was the reason why the data that was fed into the momentum model was so heavily biased.

In (Gabler, Hampton, & Roston, 2003) an argument was made that when using a traditional method of calculating crash severity, it can be inconsistent and especially with complex crash pulses. This added to the suggestion in (Kevin Brosnan, 2017) that, given the right amount of data a machine learning algorithm would be a good option when trying to model these effects.

When trying to understand how severe a car crash was and what direction a vehicle was hit from the first area of understanding was classical mechanics. Kinematics is the branch of classical mechanics focused on the motion of bodies. There may be many different forces acting on a body when it is in motion like friction but a simplified model would need to be used in this study as any other type of model would require significant research. (M. Huang, 2002) looked into a number of more complex models that could be used for future research but they could have experience the same difficulties that were found for this study (biased data). (Locey et al., 2012) (Lenard, Hurley, & Thomas, 1998) (Linder, Avery, Krafft, & Kullgren, 2003) all mentioned the deceleration of the vehicle during the crash or Delta-V(ΔV) as the a method for measuring crash severity. These papers showed that the momentum model or similar (ΔV) models have use in the industry for simple studies.

2.3 The Crash Pulse

When modelling a car crash, a key element considered was the crash pulse. (M. Huang, 2002) section 1.1 described a crash pulse as the deceleration time history at a point in the vehicle during impact while (Locey et al., 2012) described it as the “Characteristics of vehicle motion during an impact”. (Varat & Husher, 2003) did an excellent job of modelling a crash pulse mathematically and went, in length, into the detail of what influences the shape of a crash pulse from vehicle weight, age and size to the speed of the crash and the length of the time the impact occurred for. What these papers all agreed on is there are a number of characteristics that make up a crash pulse

- Maximum deceleration
- Time of maximum deceleration (ms))
- Pulse duration as defined below(ms)

- ΔV maximum change in velocity (i.e. integral of the deceleration-time curve over the pulse duration) (km/h)

All of these characteristics were available from the NHTSA data. The maximum deceleration was the largest force felt on the absolute sum of the x,y and z-direction of the accelerometer data. The time of the occurrence was the timestamp contained in the database. The pulse duration was determined by finding the maximum deceleration and then determining the end point of the crash pulse. Lastly, the maximum change in velocity was the vector of the change in velocity over the pulse duration.

2.4 Mathematical Modeling

The experiments conducted in this study were all based on mathematical modelling which is essentially a way of describing a system through mathematical language. When describing a system, it is impossible to take every variable into account therefore a simplified version of the system must be used. In engineering, people create models of bridges to try to measure their structural integrity, when these engineers create a model they use a small version with the same materials and do not recreate the entire bridge brick for brick. A mathematical model is similar in that it is impossible to recreate everything in the system and it is not computationally viable. A scaled down version is created which can generalise to the wider system or population. The aim was to take a model that was based on engineering theory and try to improve on it by using a machine learning model or multiple machine learning models.

2.4.1 Deductive Modelling versus Inductive Modelling

When a model is a logical structure based on theory or knowledge it is called a deductive model. For example, if it is **known** that there are 3 coins in a bag, the question could be: how many are left in the bag after 1 is removed? A deductive model can be created for this which would look like this: $3 - 1$. This can be applied to any scenario where there are 3 coins to start with and one is removed. There can be issues with this model even though it looks sound and it should work in every scenario. For example, there may be cases where there is a hole in the bag and one falls out. If this occurred, it would lead to errors in the results received.

Inductive modelling is where empirical results are taken and this is generalised to the system or population. Similar to the previous example there are 3 coins in a bag, every time someone removes a coin the bag is checked and the results are recorded. If, in every test case someone removed one coin from the bag this will create a presumption for a simple model that the next time someone takes a coin they will remove only one. As can be seen, this data is flawed as there are no rules governing that someone can only take one coin. It is said that this data is biased towards one and represents a similar issue in this study where the data was biased towards crashes with low severity and specific angles of impact. In the example, it will be generalised that only one coin will be removed from the bag and this presumption is passed onto every future case. As can be expected this may not be the case all of the time and therefore this type of model can cause errors in the future as the model is not representative of all cases (the population).

This study aimed to take a deductive model that was based on physics (the momentum model) and improve on the accuracy it got when compared to an inductive model (machine learning models).

2.5 Data - NHTSA Crash Test Database

In (Kevin Brosnan, 2017) the report stated they intended on creating a machine learning model but the dataset was limited in that they only had six crash tests. This led to the study focusing on data that was available publicly on-line. There were not many sites that supply data, for example, the AGU[1] was a good site but it was difficult to download data as it was in German. It was also difficult to tell if the data was available en-masse and if the sensor readouts were available. CTS.com [2] was another website

[1] http://www.agu.ch/1.0/en/crashtest-datenbank/

[2] https://www.crashtest-service.com/en/database/registration/

that appeared to have really good data but there was a fee to join. It was therefore decided to use the NHTSA database[3] as this appeared to have the data required to test the momentum model as well as having enough test cases to be able to train and test a machine learning algorithm. The data has also been widely used in similar studies whether they were to do with physical crashes (Shelby, 2011) or other highway statistics (Chong, Abraham, & Paprzycki, 2005) using the National Automotive Sampling System (NASS). The NHTSA was also the creator of the CRASH3 system which measured crash severity by determining what the ΔV was through measuring the impact area. The database was therefore reputable and had what appeared to be enough cases.

The data available from the NHTSA came in two formats, there was the descriptive data for the tests, vehicles and sensors in one down-loadable file but the sensor output was only available to download one test at a time. This represented significant work in creating a web-scraping script and then combining all of the files into a SQL server database. Before data cleaning commenced, the data was summarised and it was decided that there was enough data to be able to create a machine learning algorithm as well as test the selected momentum model.

2.6 Momentum Model (deductive model)

2.6.1 Pulse Duration

The pulse duration can be thought of as a length of time that the crash pulse occurred for. As cars have developed over the years they have been designed to crush during the pulse in order for the car to absorb the impact. When a car crash occurs it is never perfectly rigid which would mean there is no crush and the vehicle instantly bounces off the target with no compression occurring to the structure. Instead, it impacts with the object, experiences a short crush time where the car deforms and

[3]https://www-nrd.nhtsa.dot.gov/database/VSR/veh/QueryTest.aspx

then it experiences a short time where there is a spring effect felt. It was important to get this duration correct because if measurements were taken while the car was still deforming, it could give a different result to an experiment that took the measurement accurately. As can be seen in 2.2, the angles that the vehicles were facing are different at the point of impact to the end of the crash pulse.

Figure 2.1: Start of Pulse T0

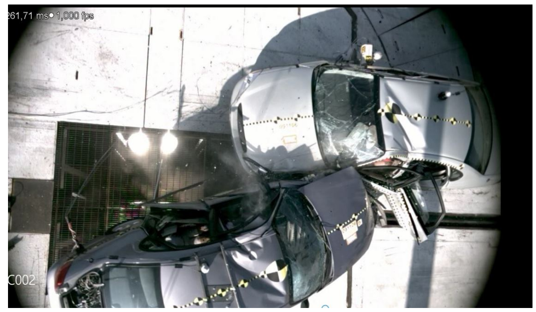

Figure 2.2: End of Pulse T1

A key issue that arose when trying to calculate the pulse duration was that it could be difficult to work out when the pulse oscillation stopped because of other oscillations felt in the crash. The sensors that were reading the forces in the car crash were mounted on the car and they experienced oscillations themselves which could make any reading convoluted (Linder et al., 2003). To counter this issue, the data was filtered in order to find the underlying crash oscillation. (Linder et al., 2003) suggested using the Channel Frequency Class 60 (CFC 60) when filtering before integrating and trying to calculate the time T_p which is the end of the crash pulse. At the end of the study, it was suggested using CFC 36 as a better filter so this was also examined. They suggested that once ΔV has been calculated for the whole data set, the time T_p can be calculated by finding 90% of the total ΔV and then checking the next time it goes negative as this will occur when the pulse has ended.

(M. Huang, 2002) dedicated the first chapter of the book to this issue and laid out in a table the different test measurements and CFC that needed to be selected. This seemed to agree with (Linder et al., 2003) by saying that CFC 60 needed to be selected when a collision simulation or total vehicle comparison was used but it also mentioned a CFC 180 when integrating for velocity. If there were any inaccuracies in

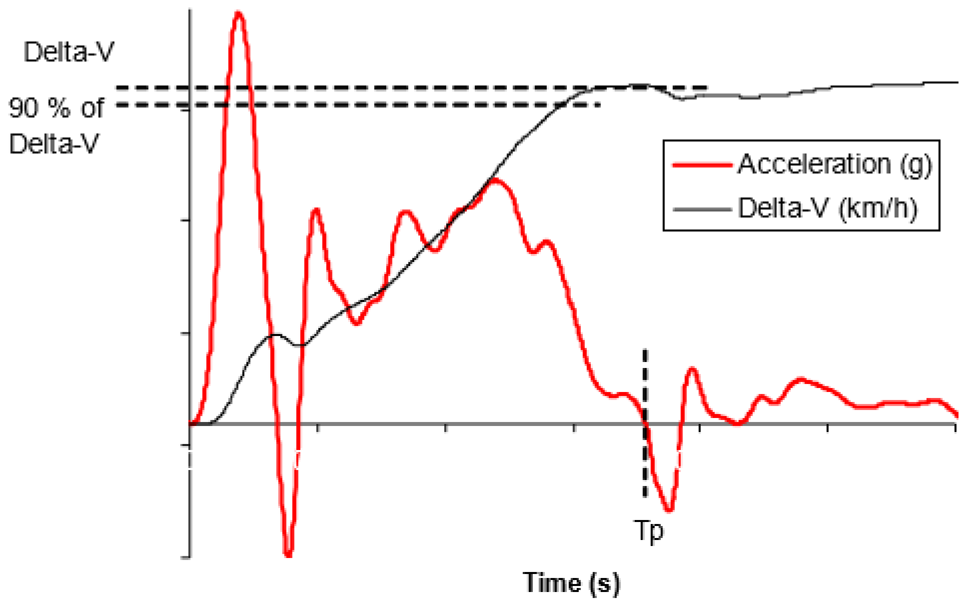

Figure 2.3: End of Pulse

the test results, CFC 180 could also be examined. (M. Huang, 2002) also went on to give different suggestions on algorithms that could be used to help with the filtering but that was beyond the scope of the study.

(Locey et al., 2012) tackled the filtering problem by building on the (Linder et al., 2003) approach and first filtering to CFC 18 to smooth extreme oscillations and the filtered acceleration was traced back to 25% of the maximum acceleration. It was then traced forward to where the filtered acceleration was equal to 5% of the total magnitude and this was considered the end of the pulse time. The approach decided on in this study was to set a constant length of half a second for the crash pulse. This was combined with no filtering and was the technique used by (Kevin Brosnan, 2017), a simple method that may lead to lower accuracy. The SVM model used a low pass filter in order for it to be down-sampled correctly. The oscillations were tested in a number of different ways and the filtered datasets gave no different readings to the non-filtered data.

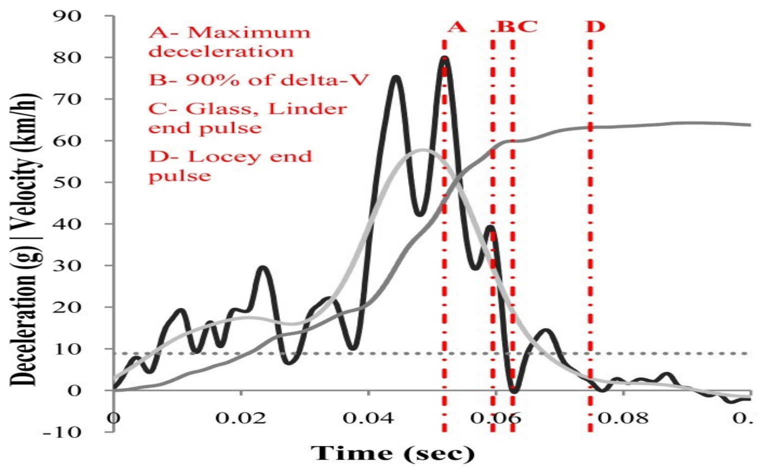

Figure 2.4: Pulse Filter Locey et, al(2012)

2.6.2 Delta V(ΔV)

It is commonly agreed that ΔV is the way of measuring a crashes severity (Locey et al., 2012) (Linder et al., 2003) (Gabauer & Gabler, 2006) (Gabauer & Gabler, 2008) (Lenard et al., 1998) (M. Huang, 2002). (Kullgren, Krafft, Nygren, & Tingvall, 2000) did some excellent studies that talked about the influence of ΔV and the potential for serious injury as a result of it. They also went on to say that even specific changes in velocity nearer the end of the crash pulse can actually influence the outcome even more.

(Lenard et al., 1998)was a study where they examined the CRASH3 software which uses the pulse deformation on the car body to estimate the ΔV. They described ΔV as the difference between the initial velocity of a crash and the end velocity in a crash. This measure was a vector and because of this, the resultant difference between the two vectors was a vector itself and therefore had magnitude and direction

The technique that CRASH3 used to generate ΔV was to measure the crush area of a barrier and a vehicle and then calculate the energy that dissipated between the two points. The database gave the NHTSA dimensions of the crush area and this could be an area of study where the machine learning technique was compared with the crush area technique to see which was more accurate. From (Lenard et al., 1998) it became obvious that this was a good way of measuring ΔV but it could be improved

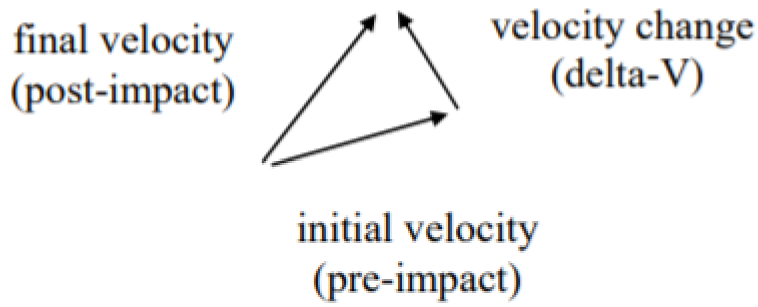

Figure 2.5: DELTA-V

upon. The method used in (Vangi, 2009) was also very similar in that the crush area could be simplified and approximated into triangular, trapezoidal or rectangular form and the force estimated from the derived shape of the crush area.

As mentioned in (Bundorf, 1996) (Robinette, Fay, & Paulsen, 1994) and (Zeidler, Schreier, & Stadelmann, 1985) the most common way of measuring ΔV is through the integrating of accelerometer data and although it states that where there is a high level of yaw in the crash the results may be inaccurate, this was the method used in this study as it was relatively simple compared to the other techniques found and was also the technique used in (Kevin Brosnan, 2017). The first step in determining ΔV using this method was to calculate the time of impact. This was done by measuring the largest force experienced on the dataset (the maximum deceleration). The next thing that needed to be done was to calculate the end of the crash pulse in order to have the pulse duration (see 2.6.1). Once the length of the crash pulse was understood, it was enough to calculate ΔV because it was the instantaneous velocity at the start of the crash pulse added to the instantaneous velocity throughout the the crash pulse. This velocity was calculated from the accelerometer data by getting the integral of each the g-force axis using the trapezoid rule. As the study was aimed at improving at the technique used in (Kevin Brosnan, 2017), these findings were reassuring in that there were examples of similar publications using the model or very similar. The model used in this study was essentially the same as calculating ΔV as a measure of the severity of the crash. Instead, the mass of each individual car was added so the change in momentum was taken instead of the change in the velocity. The research around the subject showed that people have used similar models in the past but there was a lot of software available that can model car crashes like PC CRASH and CRASH3 so future work may be done on comparing to these systems. In order to calculate the change in momentum, the pulse duration must first be determined in order to work out the crash over a period of time.

The method used, created a momentum vector which measured the instantaneous momentum at each point in time during the crash pulse. The magnitude of this vector was then taken to show how severe the crash was. Mathematically this can be expressed like this -

$$\Delta\hat{p} = \int_{timp-\epsilon}^{timp+\epsilon} m\hat{a}dt$$

The key issue with using this method was determining the window of time (ϵ) that the crash needed to be. Once this was determined, the model needed to measure the acceleration at each point in time and multiply it by the mass of the vehicle and the change in time (which was constant in the study as there were no instruments that output at different rates of time over a crash).

The angle of impact was determined directly from the same output as the the momentum model which was a vector that had magnitude and direction. This direction could then be rotated by one hundred and eighty degrees to determine where the force was coming from i.e. the PDOF.

2.6.3 Strengths and Weaknesses

When using this model for a car crash, a couple of assumptions had to be made. Firstly, it was assumed the car that was in the crash was perfectly rigid meaning there was no crush during the crash pulse. This was unrealistic in a real-world example, as cars are purposely made to absorb a large part of the impact which ensures the resulting forces acting on the passenger are reduced. The model used in this study was similar to what could be used when measuring two snooker balls impacting rather than a complex spring model which aims to measure the compression damping and rebound effects felt in the crash.

The next assumption that was made was that there were no frictional forces acting on the vehicle during the crash. The model used did not aim to measure the forces that the vehicle felt in the opposite direction to the forces being applied by the engine and the crash. Friction can be felt everywhere on the vehicle from the brakes, tyres and air.

Lastly, a large assumption made was that the crash pulse could be shortened to the same length of time for every crash. This would not be the case in a real-world scenario as the crush time was different for every vehicle depending on materials and speed etc. This led to the expectation that there would be a level of error in the model that could be reduced. This also ensured that the data would not be directly comparable to some machine learning models due to the number of features input per test case would vary.

2.6.4 Principle Direction of Force (PDOF)

Principle direction of the force is the measure of the angle of the combined vector that is the result of a collision. For example, in a head-on collision where a car hits a wall, the principal direction of force will be directly ahead of the vehicle acting towards the rear. When measuring the PDOF the vehicle is thought to exist on a plane with 0 degrees accounting for a PDOF in a head-on collision. The degrees then rotate clockwise around the vehicle with a lateral impact from the right equaling a PDOF of 90 degrees, a rear impact equaling 180 degrees and so on. (Neades & Smith, 2011) looked to determine the PDOF using a complex model which first determined the ΔVİt then used the coefficient of restitution to help create a model that could determine the PDOF even with complex crashes where the car rotates significantly. This complex model may have been more accurate than the current model chosen but the added complexity was in a field outside of the scope of the study. The method chosen in this study was based on (Kevin Brosnan, 2017) and was also the technique used in (Tolkiehn, Atallah, Lo, & Yang, 2011) where momentum was already a vector with magnitude and direction. It was then just a case of rotating this vector 180 degrees to determine where the principal direction of force came from. In (Tolkiehn et al., 2011) it was found to have a high accuracy of predicting falls of 81% when people were wearing accelerometers. As the method was simple it was expected that lateral crashes may be more inaccurate as the rotation on the vehicle was not accounted for and the length of the crash pulse could also have a bearing on the PDOF.

2.7 Machine Learning (Inductive Modelling)

2.7.1 What is Machine Learning?

According to (Alpaydin, 2014, pg. 1-4) machine learning is broken into 4 sections - Data Mining, Artificial Intelligence, Pattern Recognition, Predictive Analytics and Descriptive Analytics. Artificial Intelligence can be thought of computers mimicking human behaviour and the most important human behaviour is learning. The lines can get blurred aas most behaviours are learned but a computer can be hard-coded to move a finger on an artificial hand and this would still be artificial intelligence even though the computer did not learn the behaviour. Data Mining is the application of machine learning to databases which enables pattern recognition, predictive analytics and descriptive analytics to be done on the data.

Machine learning can also then be broken into the two different classes; supervised and unsupervised learning. Supervised learning is where the data has an output variable of Y and an input variable of X. The algorithm that explains the relationship between X and Y is a function of X (Alpaydin, 2014, pg. 4-5). There are two separate ways of applying supervised learning and this depends on the output variable type. If Y is a categorical variable then the application will need to be classification and if it is a continuous variable it will need to be regression. When selecting which type of model was going to be used, the database was summarised to see if there were any variables that could act as labels. The database contained two fields that could act as these labels -

- The 'PDOF' field or principal direction of force
- The 'NDI' field or National Damage Index

As can be seen, the PDOF in the database could act as a training label for the machine learning model and this could be compared directly back to the PDOF in the momentum model. When initially looking at the data, the PDOF could be categorised as a continuous variable as it was a number between 0 and 360. The PDOF part of the study could, therefore, be thought of as a regression problem.

The NDI was not exactly the same as the damage classification in the momentum model in that this damage was on a scale of 1 - 9 and was determined subjectively by an engineer who viewed the wreck after the test. It was therefore assumed that there would be some data that would give dubious results. In the Vehicle Damage Guide for Traffic Investigators (TEXAS, 2008) it specifically mentioned that a damage scale of 4 or above is moderate to high levels of damage. The study, therefore, used this guideline to define levels 1 to 3 as low severity and 4 to 9 as high. The crash severity problem could, therefore, be thought of as a classification problem.

2.7.2 Crash Severity Classification Model Selection

Although there were no direct comparisons with a publication that used the same data and tried to get similar results, there were publications that have been done on classifying movements using accelerometer data. (Ravi, Dandekar, Mysore, & Littman, 2005) was a publication where the aim was to classify human activity by using an accelerometer that was attached to a person's waist. A window of activity was taken from the overall sensor outputs and it was these numbers that were classified. As can be seen in 2.6 the data that was gathered throughout the day had multiple different categories of activity with a wavelength that was very different. Whereas in this study (see 3.2) each test case had one activity per test and the classification was not on the differences contained in each test but rather the severity of each test was classified. There was, however, a window taken of the number of sensor outputs to use per test case but the window started at the point of impact in the crash and finished a half a second later.

(Ravi et al., 2005) went on to try multiple classifiers and it acted as a good reference point to choose a model. The results showed that a boosted SVM had the highest accuracy at classifying the activity where the training cases were different to the testing cases. This was similar to the data that was used in this study as each car crash was an individual and could not be repeated.

Another similar publication was (Cheng & Jhan, 2013) where a person's activity

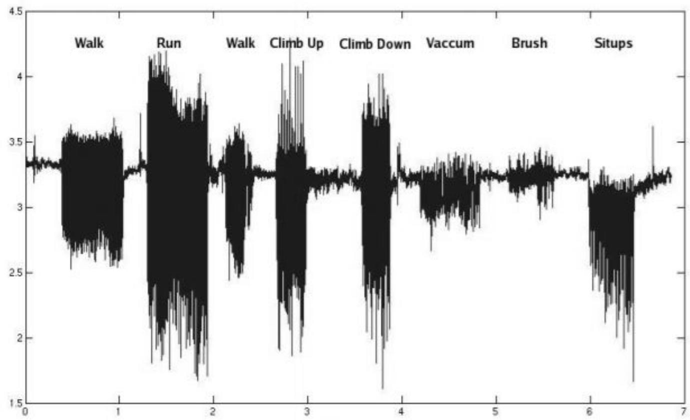

Figure 2.6: Activity Classification Accelerometer Data Reprinted from Ravi et al.2005

was classified using accelerometers positioned on different parts of the body. The publication aimed to get more accuracy from the model by selecting an ADA boosted SVM model which is a parallel classifier that consists of multiple weaker classifiers that focus on one part of the problem. Due to the nature of this study, each individual crash event happened in isolation as the data was only gathered in 5-second segments that just contained the crash. The comparison with the other publications, where there are multiple events occurring in each case is something that would need to be explored if the data became available.

(Zhao, Pawlus, Karimi, & Robbersmyr, 2014)was a publication that tried to reproduce the vehicle kinematics of a crash using an adaptive neuro-fuzzy inference system. Although the data that was used in the publication was similar to this study, the quantities and output were very different. They tried to reproduce all the interactions that occurred in the time-series rather than the output of the crash as a category.

Another publication that used the same tri-axial accelerometer data that was used in this study was (Khan, Lee, Lee, & Kim, 2010). The difference between this publication and the previous two cited above was this one aimed to used time series analysis to see if the upcoming sensor data could be predicted. This would be useful for the likes of an airbag test where the potential magnitude of the crash could be predicted in real time as an impact is occurring.

In (Torrão, Coelho, & Rouphail, 2013) the same result was sought using a classifier but the data used for the study to train the classifier did not use accelerometer data but did use vehicle characteristics that are available in the NHTSA (except age and sex of driver as it is a test centre). They used a logistic regression model and a decision tree classifier and it experienced some biased data that should be expected in this study. It also added to the idea that potentially the vehicle characteristics could add to the severity of the crash or similarly influence the PDOF. In (Selmanaj, Corno, & Savaresi, 2014) there was a very similar requirement as in this study in that they were looking to create an airbag on a motorcycle that was activated using a neural network classifier.

On weighing up the research it was decided to use an SVM classifier as this technique was in the most cited papers and the ones that could be transferred best to this study. Although the activities that were classified by these studies in general related to human activity this study did not need to differentiate between different activities but instead just needed to differentiate between the severity of the crash. The added bonus of having data that described the vehicle as well as the closing speed meant that these features could also be used in the model to try and differentiate between crashes.

2.7.3 Principle Direction of Force Regression/Classification Model Selection

While searching for a model that determined an angle using accelerometer data, a publication was found (Mayagoitia, Nene, & Veltink, 2002) that calculated joint rotation using accelerometer data from sensors placed at multiple points on the leg. This was a deductive model similar to the momentum model in this study and the results were compared back to visual software to determine if the angles the sensors moved relative to each other on the joints were correct. Although not directly comparable back to the machine learning model in this study, it does show that deductive modelling is widely used when it comes to sensor data problems. This point was also backed up in (Tolkiehn et al., 2011) who used a similar deductive model to this study to determine the direction a person was falling through use of accelerometer sensors attached to various parts of their body.

(Swartz et al., 2000) was a study that used accelerometers attached to people to try to determine the calories burned during certain activities. The activities ranged from gardening to jogging and the results were mapped to a calorie counter also attached to the subject. The results showed promise but the data was a bad fit of 0.56 which accounts for the high variation in the activities. Although not directly comparable to this study, the fact that a continuous variable was regressed using accelerometer data was promising and the fact that the issues that arose were related to activity variation should not be a problem in this case. (L. Huang & Chen, 2001) used a multi-linear regression technique that measuring accelerometer data connected to a sensor on a lathe to try to measure the surface roughness of a machine worked piece. Although the problem set was different it showed the applicability of regression using accelerometer data.

(Stansfield, Hillman, Hazlewood, & Robb, 2006) was a publication were camera data was used to analyse children's gaits and then the kinematic data was used as independent variables when predicting a person's walking speed. The publication did not include accelerometer data but it did use some descriptive features of the child on top of the kinematic features. In this study, the kinematic features were not as relevant to a car unless the study was to conduct a full finite element model where all of the structural elements are measured.

(C. Richter, King, Falvey, & Franklyn-Miller, 2018) was a publication where machine learning techniques were used to try to classify change in direction movements. The data used was based on accelerometer data that was gathered from a force platform under the test subjects and this was combined with a series of infrared cameras that picked up the motion. The input variables to the machine learning algorithms picked up the full range of kinematic variables and the activities were labelled.

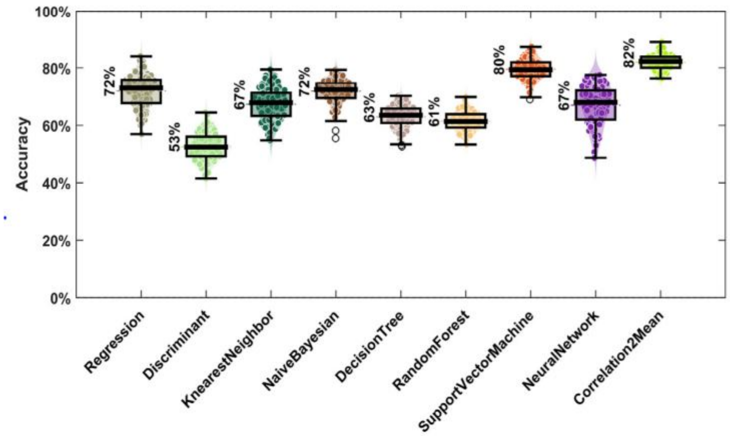

Figure 2.7: Direction Change Classifier Results Reprinted from C. Richter et al. 2018

As can be seen in 2.7(C. Richter et al., 2018) the various classifier results were compared to each other. The study mentioned that Correlation2Mean had the best performance but the SVM model was also a very high performer. Although this study did not use a continuous variable as the input, it did use a multinomial regression model which also showed high results.

Although there were not any direct comparison studies available that aimed to use machine learning to either regress or classify the principal direction of force, there were a number of studies that used similar input variables. The majority of studies that used these variables and are using a machine learning solution are in the classification application of machine learning. It was therefore decided to start off using a regression model with the posit that the angle of impact was a continuous variable. For completeness, the experiment further examined whether assigning the PDOF as a categorical variable and using a multinomial regression model could improve on those results and also if a multi-level SVM classifier could outperform them.

2.8 Support Vector Machines (SVM)

Conceptually, a basic linear classifier works by plotting all of the possible instances contained in the data or otherwise into the input space. A decision boundary is then determined between the points and this is then used as a plane that will determine future classifications. When this is mapped out 2 dimensionally it can be easy to understand but not always linearly separable. In SVMs the data can be projected into the feature space which has the same number of dimensions as features and although this can be difficult to conceptualise (Flach, 2012, p. 23) it can be much easier to define a linear boundary in this higher dimensional space. This linear boundary can then be projected back to a non-linear boundary in the input space. For example, see 2.8 wherein the original input space does not have a clear decision boundary which is linearly separable as it is a triangle of green points within a circle of red points. It is, however, linearly separable in 2.9 where the three-dimensional rotation of the feature space on the left shows the red circles are actually above the green ones and a simple linear plane can be drawn between them (represented by the black plane).

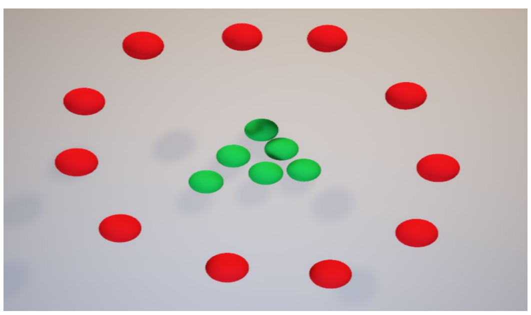

Figure 2.8: Data Points From Above

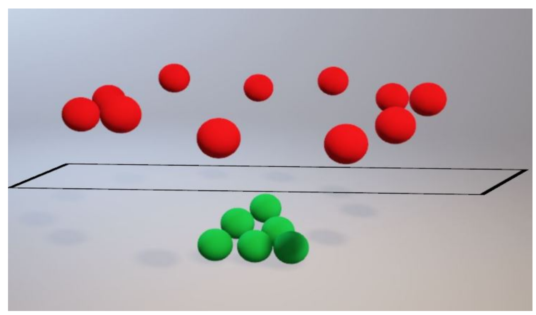

Figure 2.9: Data Points Rotated in 3D

A support vector machine selects the best possible decision boundary by selecting the line that maximises the margin between the two closest points aka the support vectors. For better generalisation, the distance between the closest points must be maximised as this will stop points being mis-classified that are very close to the margin. This can be described formally (Hsu, Chang, Lin, et al., 2003) as follows: Given a training set of instance-label pairs $(x_i, y_i, i = 1, ..., l \ where \ x_i \epsilon R^n)$ and $y \epsilon 1, -1^l$, the support vector machines requires the solution of the following optimisation problem -

$$\frac{MIN}{w, b, \xi} \ \frac{1}{2}||w||^2 + C\sum_{i=1}^{l}\xi_i \quad subject \ to \quad y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i, \xi_i \geq 0t$$

This optimisation problem can also be looked at graphically as in 2.10 which was reprinted from (Raschka, 2015).

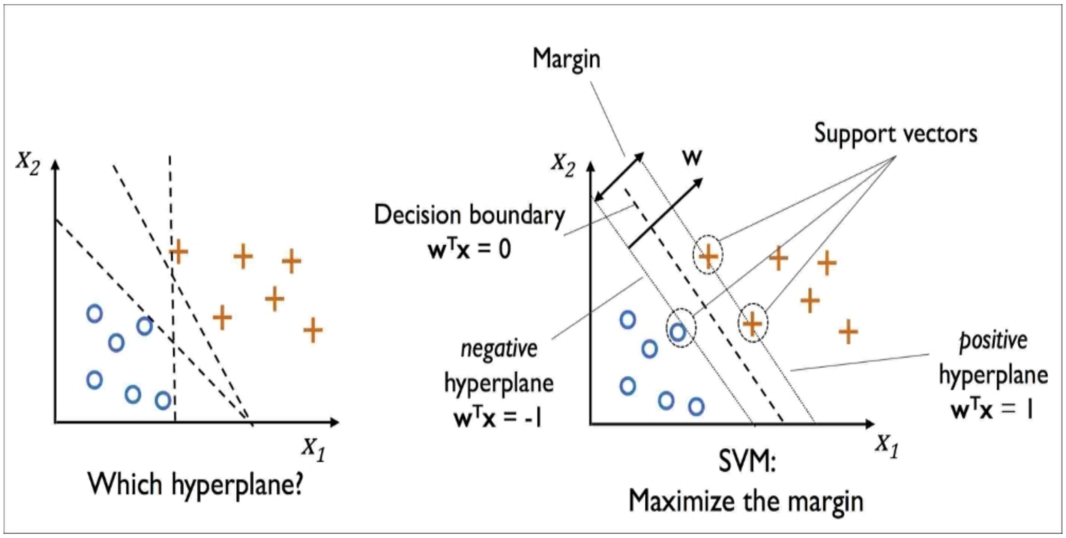

Figure 2.10: SVM Margin Reprinted From Python Machine Learning by Sebastian Raschka

As can be seen in the left diagram in 2.10 the data is divided into two data types using a line between their closest points. An infinite number of lines can be drawn so the issue becomes how to select which one of infinite lines for the best decision boundary. The answer is to maximise the area between the closest two points in each of the separate groups and by having two separate hyperplanes, one at positive 1 and one at negative 1, the best decision boundary can be set at 0. This can all be done in infinitely higher dimensional space for complex problems using kernels. The experiment followed the following steps (Hsu et al., 2003) -

- Transform the data to the format of the SVM package
- Conduct simple scaling on the data
- Consider the RBF kernel
- Use Cross Validation to find the best value of C and gamma
- Use the best values of C and gamma and train the whole training set
- Test

Transforming the data for the SVM in R was not a problem as the e1071 package automatically did this. This was where it took the categorical variables that were in the training set and it converted them to continuous variables. Next scaling the data did not represent a problem either as the sensor data input to the model and the speed were converted to the same scale automatically using the e1071 package.

2.8.1 Model Selection

This represented a key element of the experiment as the number of models that could be created were infinite and it was important to get it right. The first step of model selection was to select the kernel method to be used. In (Hsu et al., 2003) the suggestion was to use the Radial kernel as it could be more accurate and efficient to use but it also suggested that the linear kernel may be better when there are a larger number of features. It was therefore decided to use both in this experiment and compare the results. The k-fold cross validation method discussed in the paper was also available in the e1071 package out of the box, the shortcomings mentioned where a large dataset can be slow to return results was not an issue.

2.8.2 Strengths and Limitations

The SVM is a good model that performs really well in complicated domains that have a clear separation but do not perform well when there is a very large dataset with lots of noise in the data. The SVM was presumed to perform well in this study as each case only contained one crash event recorded and no other events occurred. This coupled with the fact that the initial point of impact was always selected correctly at the highest combined force and then the next half second of data was added to this to get the crash pulse. Therefore, there was a relatively small amount of noise in the data. The fact that the NDI was subjective did blur the lines of separation and this was assumed to be the biggest barrier to high performance.

The SVM may also be used to determine the angle of impact as the sensor data is a dependent variable in time series. A number of studies including (Steinwart, Hush, & Scovel, 2009) mentioned that SVMs can take dependent variables in the learning environment as they will not weight these variables more heavily.

2.9 Multivariate, Logistic and Multinomial Regression

2.9.1 Multivariate Regression

Regression is another type of geometric inductive model where the data is plotted to an imaginary space. This input space can be thought of as an actual model of space like when latitude is plotted against longitude or a non-intrinsically geometric space like when weight is plotted against height. As in SVM, these model types use geometric concepts like lines or planes to impose structure on the data.
A linear regression model aims to take the data and plot it in the input space. It then chooses a line that best fits the data and the slope of this line (the coefficient notated by β) is the relationship between the two variables. When the relationship is just between two variables i.e. the response Y and the dependent variable X this is called a simple linear regression, see 2.11

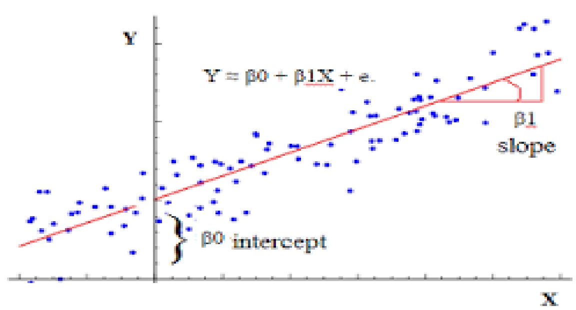

Figure 2.11: Simple Linear Regression

As can be seen from 2.11 the points do not all sit on the line as the difference between these points and the line is called the residual and can be explained formally as $\epsilon_i = f(x_i) - \hat{f}(x_i)$. In order to determine what is the line that best fits from the infinite number of lines, the least squares method is used. This is where the sum of all of the residuals is squared and then the line with the lowest value is the best fit. Multilinear regression is similar to simple linear regression in that terms are just added to the linear formula to make the problem space multidimensional. Instead of a line, the data can be described with a plane and the same process follows to fit a plane as a line with the least squares method. A multilinear regression model will be in the format -

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i$$

Where y_i is the response variable or dependent variable. β_0 is the intercept of the line and the way of anchoring it. β_1 and β_2 are the relationships between y and x_1 and x_2 and ϵ_i is the error in the model. When adding in a factor to a regression model, it changes the format of the model slightly as dummy variables need to be applied to the factors (similar to the SVM model). When interpreting a model with categorical variables in the dependent variable category, the intercept is replaced with the first level of the factor and this reference variable is then compared to the other levels in the factor across the model -

$$y_i = \beta_0 + \beta_1 \delta_{i1} + \beta_2 \delta_{i2} + \beta_3 \delta_{i3} + \epsilon_i$$

The above model is a regression model for one continuous response variable with zero dependent categorical variable. In R, the tool selected the β_0 represents the first level of the factor and is compared to the next three levels of the factor with the next three β's representing the relationship and the δ being the dummy variable that can either be one or zero.

2.9.2 Logistic Regression

When using a response variable that is a two-level categorical variable, a linear model does not work as the data cannot be described as a line. First, the response variable must be assigned a number. For example in 2.12 you can see the y variable has been assigned a value of zero or one. The x value, in this case, is between zero and 100% so as the percentages increase you can see the Y value is either zero or one and not any number in between. This example is in relation to motorcross racing, the x-axis

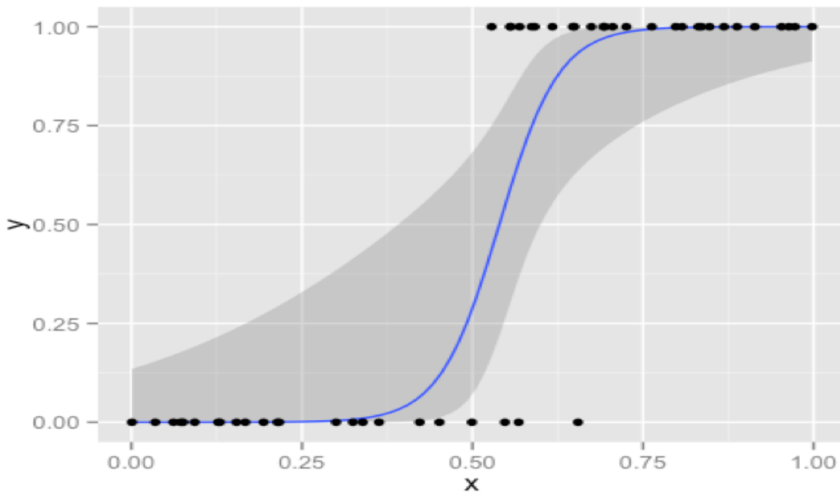

Figure 2.12: Logistic Regression Shape

is the percentage of jumps 'cleared' in a race. If the rider lands badly or goes around a jump it reduces this percentage. The y-axis is whether a rider finishes the race or not (one or zero). This relationship could be interpreted as: most riders that cleared more than 50% of the jumps, finished the race. As can be seen, there are three riders who got over 50% but still did not finish the race (zero in y response but over 50% in x). This model is fitted differently to the multivariate regression as it represents a different type of model - a Generalised Linear Model. A logistic regression model is fit using a maximum likelihood method -

$$L(\beta_0, \beta_1) = \prod_{i=1}^{n} \left(\frac{e^{\beta_0 + \beta_1 (x_i)}}{1 + e^{\beta_0 + \beta_1 (x_i)}} \right)^{y_i} \left(\frac{1}{1 + e^{\beta_0 + \beta_1 (x_i)}} \right)^{1-y_i}$$

The likelihood is the joint probability of the data and assuming conditional independence the values of β_0 and β_1 are the ones that maximise the above expression. The logistic regression model can, therefore, be written as -

$$\log\left(\frac{p_i}{1-p_i}\right) = n_i = \beta_0 + \beta_1 x_i$$

Where the linear predictor is the logit of p_i.

2.9.3 Multinomial Regression

Multinomial regression models follow directly from logistic models but they can be applied to variables that have more than two categorical levels in the response variable. According to (Bridle, 1990) the following formula applies if all classes can be treated uniformly -

$$y_i = \hat{P}(C_i|X) = \frac{\exp\left[\log\left(\frac{p(X|C_i)}{p(X|C_k)}\right)\right]}{\sum_j^K = 1 \exp\left[\log\left(\frac{p(X|C_i)}{p(X|C_k)}\right)\right]}$$

2.9.4 Strengths and Weaknesses

Firstly, when using regression it was important to know what the relationship looked like because using a linear model on data that was not linear can give incorrect results. Fortunately, there are generalised models that can ensure this was accounted for. Using the multivariate regression for the PDOF was the initial thought when creating the experiment but this variable may actually turn out to be more of a categorical variable in which case the other methods need to be checked (including the multi-level SVM). The problem was; if a category was the response variable, when using the logistic or multinomial methods, it was assumed that there was a linear discrimination between the different categories. This was assumed to be the case in this study but the fact that the NDI was a subjective opinion means this probably was not the case and the SVM may perform better. Depending on the angle of impact, the multinomial regression may work if the multivariate does not, as there should be a clear linear discrimination between the different angles for each of the tests.

A common problem with the logistic and multinomial regression model, however, is that each observation needs to be independent of the next. This was a problem in the PDOF model as there were a number of observations for each crash that were related in time series. This point was considered when creating the model.

2.10 Summary of Literature Review

There was a business need for insurance companies to be able to assess claims in real time as this allows for better budgeting and can help avoid fraud. The private report acted as a good starting point which enabled clear direction in what way the research should go and what type of experiments might work. There were reports that measured crash severity and angle of impact using similar methods implemented in this study but there were no papers found where a comparison was made between this technique and any machine learning techniques. The data has been used in a similar fashion but it was only ever on small numbers of cases (3-5 for example) and there were no large-scale tests. This may be down to the fact that the data was not as suitable for large-scale tests. What did become obvious from the literature review was that in simple test cases the model choice was ok but from an engineering point of view where inductive models were used there are more complex options that may increase accuracy which included a Spring Model or a finite element method.

There were a few similar studies to this one in terms of machine learning but none used the same data with a comparison back to a deductive model. There were many successful classification studies that used similar data. The research into SVMs showed that was a suitable model for both sides of the problem while the use of regression threw up a number of concerns that were heeded in the study.

The research had the potential to both improve on a simple model that was suggested to a business and also help guide future research in what models could be used when classifying using the NHTSA dataset. This led to the research question that guided the study -

Research Question

Can ML models match the accuracy of a momentum model for car crashes using *force in x,y,z planes as the independent variables and the angle of impact or severity of crash as the dependent variables?*

Chapter 3

Design and Methodology

3.1 Introduction

This chapter outlines the Cross Industry Standard Process for Data Mining (CRISP-DM) methodology used for creating the experiment and the steps outlined in (Wirth & Hipp, 2000). The experiment was divided into two distinct parts: **1. Momentum method** and **2. Machine Learning method**. There are significant differences between how both of these methods were created but an initial test was done on each using the same final dataset and then different slices of the data were used for exploratory findings. Due to the complexity of the study, a methodology was used to provide a conceptual framework within which the experiment was conducted. This framework was then used as a guide for the experiment and also structured the chapter ahead. As can be seen in 3.1 the process was an iterative one that generally followed these steps consecutively. The framework needed to have a certain level of versatility because the nature of the problem meant new information was constantly being discovered. This new knowledge then needed to be fed through the framework so issues that were discovered in later stages could result in a re-assessment of knowledge discovered at earlier stages.

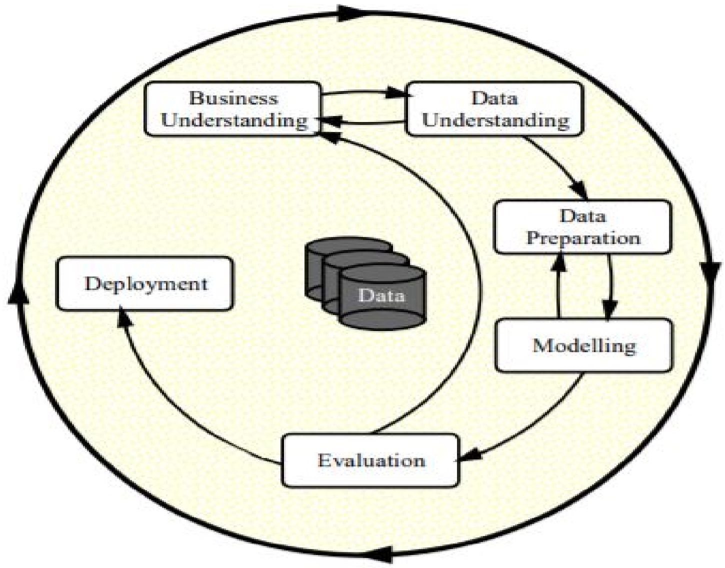

Figure 3.1: CRISP-DM (Wirth Hipp 2000)

3.2 Business Understanding

3.2.1 Determine Business Objectives

Insurance Claims Managers (CM) have a need to assess the severity of a car crash and understand the angle of impact as this has can influence the cost of a claim. The quicker and more accurate these assessments are, the better-prepared insurance companies are for pay-outs. There are currently numerous ways of calculating the severity of a crash using numerical methods and other engineering techniques but these can be computationally intensive with a high requirement of data at the point of impact and there is a certain degree of mathematical expertise needed to create these models. Creating a model that could be used in real-time as the car crash happens may not be best served with models that have high computational requirements and may be better served with an algorithm that is trained on the data and can run quicker in real time.

The objective from the business point of view was to develop a method that models a car crashs severity and PDOF in order to give a CM real-time information about

a car crash. This new model was compared to a simple momentum model that has been used in studies to determine these variables and conclusions were drawn on whether the machine learning model was an improvement.
This leads us to the research question:

Can Machine Learning beat Physics at Modelling Car Crashes?

3.2.2 Assess situation

The main resource used for this study was the NHTSA database, this database stores crash tests that have been conducted back to 1978 and is available publicly for download. A full outline of the data available in this site will be outlined in 3.3.

3.2.3 Initial Data Collection and Data Description Report

The data that was used for the experiment was divided into two types -

- Test, Vehicle and Sensor Level descriptive data
- Sensor Output Data

The descriptive data that was measured in the tables is available as one download from the NHTSA database[1]. This data describes the characteristics of each of the entities involved in the test at varying hierarchical levels.
Test Level (tst table)- This table consisted of 8,291 cases and showed a single line for every test that was identified through the test identification number (TSTNO field). It described important information about the test including the test type (TSTCFN field), the speed of the vehicle before impact (CLSSPD field) and the impact angle of the vehicle (IMPANG field). The number of tests used for the study was needed to be reduced to the number that had a test where there was a vehicle used. There were a number of tests that used impactors, barriers and trolleys which could not be

[1]https://www-nrd.nhtsa.dot.gov/database/veh/veh.htm

Table 3.1: Test Data Extract

MAKE	YEAR	BODY	ENGINE	VEHTWT	VEHCG	VEHSPD	PDOF
20	1979	4S	4CIF	1530	1298	56.3	0
16	1979	2S	4CIF	1202	1214	56.3	0
15	1980	3H	4CIF	1090	1024	55.8	0
26	1980	4S	4CIF	1177	1087	56.3	0
6	1980	4S	V6IF	1730	1349	57	0
28	1980	4S	4CIF	1685	1379	56.3	0

used. The impactors did not have any mass associated with them in the database so the test was not relevant or testable using the momentum model. The tests that used the trolleys could have the vehicle moving in a direction but the car was pointed in a different direction. This gave misleading results so they were also excluded. There were also some tests that had vehicle into vehicle tests where the complexity of the model was not sufficient to be able to determine how bad the crash was due to severe forces felt on the vehicles. It was decided to ensure the model gave accurate results to only use vehicles that were accelerating at the start of the test as the model required the variables available in these tests.

Vehicle Level (veh table)- This table (see 3.1 for snapshot) consisted of 10,790 cases and showed a single line for every vehicle in a test that was identified through the test identification number (TSTNO field) and the vehicle number (VEHNO field). This table had lots of descriptive data about the vehicle ranging from the weight, model, year of car to the speed it was travelling and the crush dimensions of the crash. Key fields for this study were the vehicle damage index (VDI field), which described the collision deformation in an engineering classification code (see 3.5) and the make of the vehicle (MAKE field) which identified any barriers used in the test by the NHTSA description. The number of vehicles used for the study was needed to be reduced to the moving vehicle (aka bullet vehicle) in the test and also to any cases that contained a VDI.

Sensor Level (instr table)- This table consisted of 422,380 cases and showed a single line for every sensor in every vehicle in a test and the axis it measured, this was identified through the test identification number (TSTNO field), the vehicle number (VEHNO field) and the sensor identification number (CURNO field). The sensors in this table could refer to sensors that are in relation to the structure of the car, barriers, crash test dummies and impactors and could vary in their outputs. Key fields for this study were the sensor type (SENATT field), which described the sensor (e.g. VECG is vehicle centre of gravity) and the axis the sensor was measuring (AXIS field). Depending on the sensor type, they could give out any combination of x,y or z-axis but for this study, the majority of the vehicle centre of gravity sensors gave all 3. There was also a field that showed whether the sensor was primary or redundant as there were some sensors with multiple data channels. If multiple data channels were mixed in with each other it would return incorrect outputs. The number of vehicles used for the study needed to be reduced to any VECG sensors in the tests that are selected from the other tables.

The output data that was measured in the tables could only be viewed on-line, one test at a time and represented a significant obstacle to extract the tests required for the study.

Sensor Output Level (Sensor_Output table)- This table was not originally available as one downloadable file and its creation is described in section 3.3.1. The finished table consisted of 30,674,133 cases and showed a single line for every output from a sensor during a test and was identified through the test identification number (TSTNO field), the vehicle number (VEHNO field) and the sensor identification number (CURNO field) and the time (Time field) that the output was at. Generally, each test occurred for 3 to 5 seconds and had units of 0.0008 seconds. Each record was identified by combining the identification numbers in the three descriptive tables above (TSTNO, CURNO and VEHNO). 3.2 shows test 3845 and plots the forces that were experienced in the vehicle centre of gravity sensors, in the X-axis (forwards and backwards), throughout the time of the test in the top left chart as they were in the database. As can be seen, a large spike was experienced around the 0.05-second mark

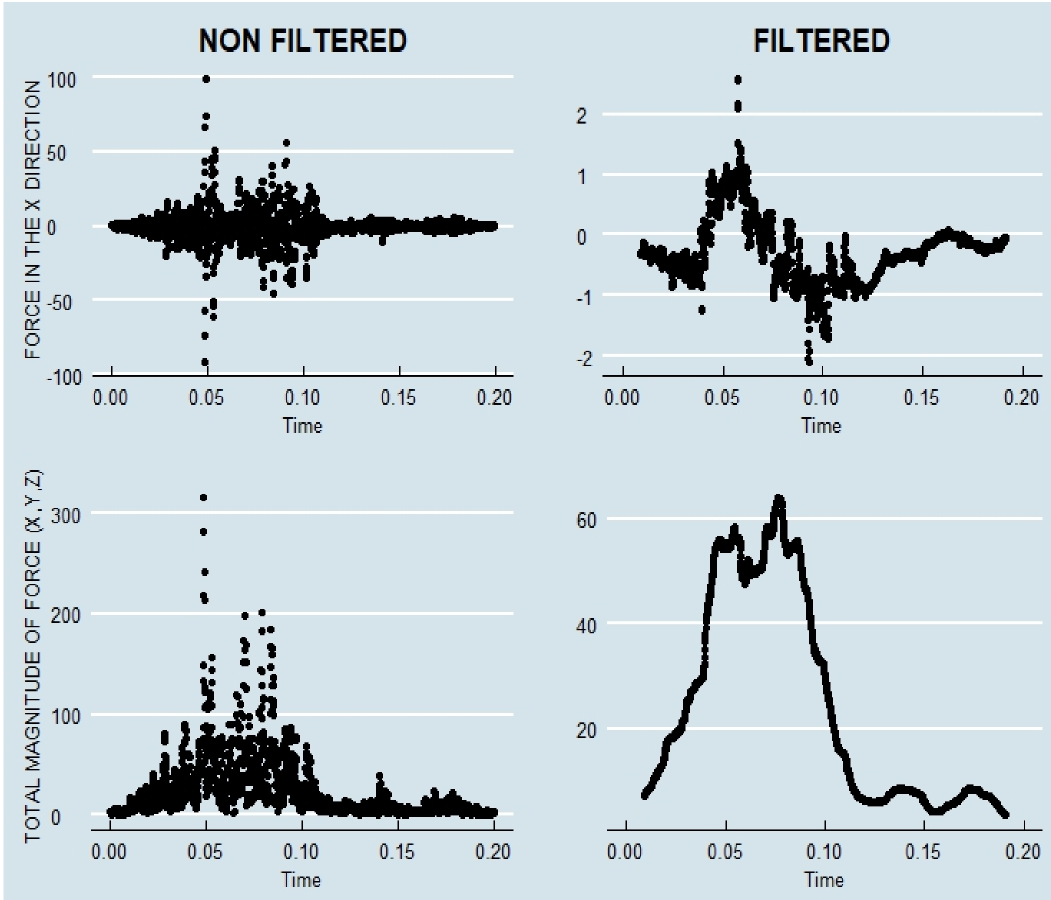

Figure 3.2: Test 3845 Sensor Output Total Force v Time

and this was the start of the crash. The crash pulse then finished around the 0.10-second mark.

The bottom row of graphs represented the total magnitude of the forces experienced on the sensor over the period of the crash with the left one showing the data as was gathered from the database and the right one filtering to 180 hertz (M. Huang, 2002).In the literature, there was mention of filtering the data to ensure that any vibrations felt on the sensor mountings were accounted for. As can be seen in 3.2, the low pass filtering did change the shape of the wave slightly (from left to right) but there was minimal impact on the length of the crash. In all four of these charts, it was seen that the crash begins in around the 0.05-second mark and the crash then ends around the 0.10 mark.

3.2.4 Key Steps

The key steps for the experiment will be

- Selecting the relevant data for the momentum model
- Combining the data from multiple datasets
- Prepare data for Momentum model
- Create the momentum model
- Determine the accuracy of the momentum model using the ground truth in the database
- Create the severity model using same features as momentum model
- Create the PDOF model using same features as momentum model
- Determine the accuracy of the model using the ground truth in the database.
- Create the severity model using other features in the database
- Create the PDOF model using other features in the database
- Compare the models

3.3 Data Understanding

The first major assumption made when starting the study was in relation to the ground truth. The first ground truth used was the crash severity and was determined by the vehicle damage index (VDI) which was an engineering classification in the vehicle table. The NDI was broken into three parts - the first two numbers were the direction the impact occurred from, the next three to four letters were the location of the crush on the vehicle and the last one to two numbers were the crash severity. The crash severity part of the classification is the only part that was used in this study. This classification of the crash severity was purely based on the size of the indentation and was subjective in that it was assessed by an engineer by eye. As can be seen in 3.5 the two numbers in the engineering classification were used to code the indent. This scale was measured from one to ten with one representing a slight indentation and ten representing an indentation that may result in the car being half its length. The momentum model used the rate of change in momentum to determine how bad the crash was so there were expected differences between an accurate readout from the momentum model to the database based on the subjective opinion of the engineer who classified the crash.

As both models aimed to classify the crash severity into high or low, it was important to determine if there was any way of classifying the data into two levels from the ten levels in the NDI. By taking the mean speed of all bullet vehicles which had a rating and plotting this mean against each level a graphical representation of the data was created with 95% confidence intervals (see 3.3). There appears to be a slight linear relationship between the speed and the severity which would be expected. As can be

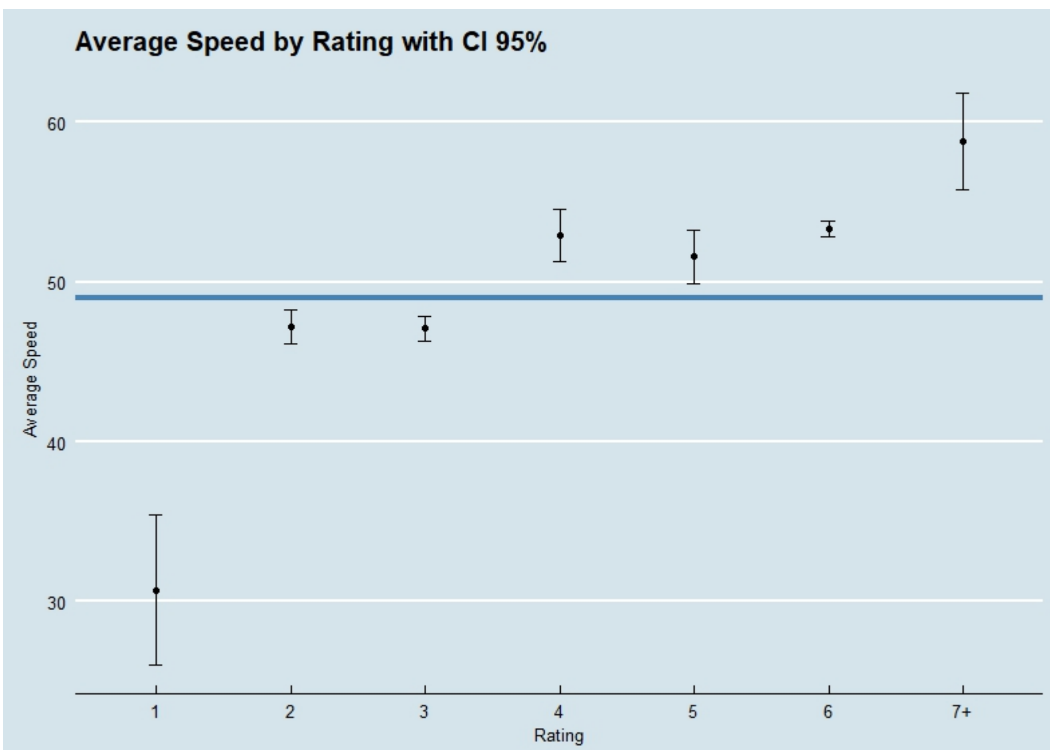

Figure 3.3: Average Speed by Rating

seen in the chart, there was an overlap of the ratings from four to six and again in two to three so it was determined to split the data from one to three as low and four to seven plus as high. This point was reinforced by the Crash Investigators paper (TEXAS, 2008) were the rating one to three corresponds to a car with light damage. If the levels needed to be split into four levels, level one would represent its own level as it is clearly different to the others and seven plus would represent its own category. Another issue related to this field was that the test engineer may input the data incorrectly, in the wrong format or may have left this field blank. As this was key to the study, all data in this field that was in the wrong format needed to be removed and it would then have to be assumed that any field that was filled out correctly was accurate as there was no way of proving otherwise.

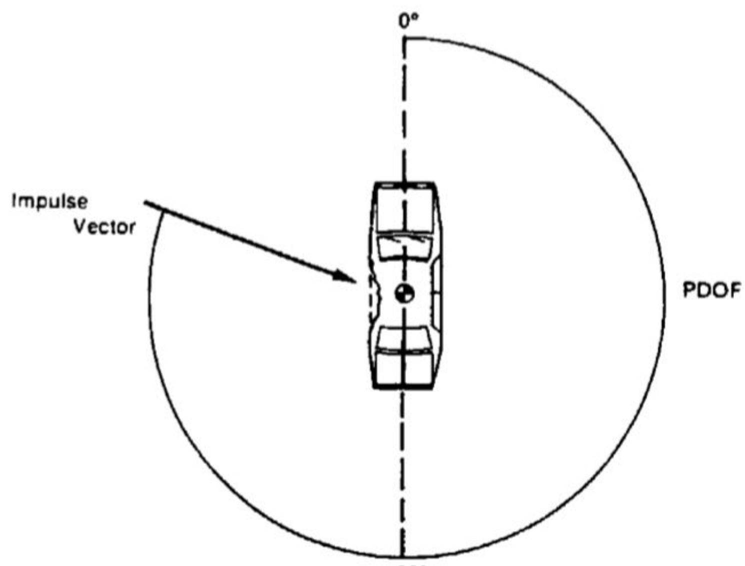

Figure 3.4: Principle Direction of Force (PDOF)

The other ground truth that was being used was the principal direction of force (PDOF) which was contained as a number in the Test database. The PDOF was the angle at which the test vehicle felt the impact coming from relative to its direction of travel. As all tests selected in the study were moving forward, if the car test was a head-on collision this was a zero degree impact angle. The angle then increased clockwise depending on where the impact came from. Problems were expected to arise in that this could be inaccurate in the database or there may be missing data which would result in a smaller sample space to work with.

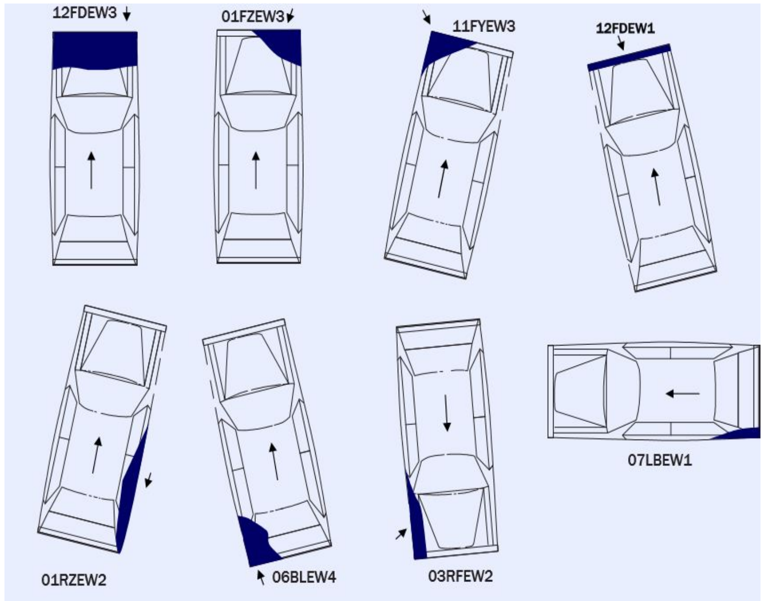

Figure 3.5: Engineering Rating

3.3.1 Momentum Model Input Data

A key consideration when looking at the car crash was that the model being built needed to be used to determine the crash severity and the PDOF for the bullet car in the impact. The bullet car in testing was the car that was travelling at speed into a fixed object. There were a number of types of crash tests that the NHTSA track and the data needed to be reduced to only have tests that a bullet car was involved in and this was to be selected as the test case. Another point that followed from this was that all barrier, impact sledge and stationary cars needed to be removed from the sample space.

The momentum model that was used was based on rigid body mechanics and a simplified momentum model that did not take into account friction or spring forces. It was therefore very important that the sensor that was used in the experiment was always the vehicle centre of gravity (VECG) or a sensor that was very close to the VECG. This ensured that any measurements that were taken referred to the mass of the vehicle as a whole and not to any specific part of the car. For example, if the sensor was on the bumper of the car it would experience rotational forces that may not be felt at the centre of the car. The Sensor table contained the description of the type of sensor and it was therefore needed to be reduced to only the cases that had a VECG sensor.

The other data that mattered to the momentum model were the vehicle weight, the initial speed of the vehicle, and then the forces felt in the x,y and z directions. The vehicle weight was contained in the vehicle table and was measured in kilograms while the initial speed was in the test table and was measured in kilometres per hour. The initial speed was the measurement taken of the vehicle at the point of impact so there were slight differences between this speed and the actual vehicle speed but the assumption had to be made that it was correct as there was no other way of measuring it.

The forces in the x, y and z directions represented a difficult problem because they were not available as one mass download from the website and each sensor was divided into the 3 different files which represented each of the x, y and z directions. This meant that each of these files had to be matched to the relevant test and then each record in each of the three files had to be matched to the corresponding time. Table 3.2 shows a sample of one of the tests data in the x-direction. In order for there to be a complete readout of this sample there needed to be 2 more files that were named test 6 and also they must have had the same times in the Time column in order for the data to be joined together.

The length of time that the crash occurs for must be selected in order to calculate the ΔV of the crash. This represented a problem when the comparison was done against the machine learning model. In order for a direct comparison to be done including all sensor outputs, a standard length of time was needed to be selected for the length of the crash. This ensured that the same number of features were fed into the SVM. If, however, a variant number of sensor outputs were input into the momentum model it would not be possible to compare the two models with the same input features.

In order for the model to work correctly any rows that did not have a time or x,y,z

Table 3.2: Example Accelerometer Data Long Format in X Direction

TIME	FORCE
0	0.00241
0.000075	0.03241
0.00015	-0.03515
0.000225	0.00215
0.0003	-0.41709
0.000375	-0.02421

value in each of their columns needed to be removed.

3.3.2 Machine Learning Model Input Data

The machine learning model aimed to improve on the speed and accuracy of the momentum model by using different input variables and training the model before use. In order for the model to be usable in an insurance company environment, feature selection needed to be based on the data that would be available to the company before and after the crash. Data such as the crush distance is difficult to measure at the time of the incident and there were no sensors in the database that measured this. The problem was divided into two separate models with one classifying the crash severity and the other either classifying or regressing the PDOF.

A number of separate tests had to be done on the PDOF to determine which type of model needed to be used. The total number of cases in the vehicle table was 10,790 and when a “count distinct” of all of the levels was done on the angle it only gave 71 different outcomes. This would seem to indicate that the angle was more of a classification problem than a direct regression problem. A key determinant of which model to use was to find out how many distinct cases the angle had reduced to after all cases had been cleaned.

In order to determine which was more accurate, the momentum or the machine learning models, the first point of comparison was to compare them on the same input features.

Depending on how many cases the dataset was reduced to, this was predicted to give results that would be difficult to draw conclusions from. The next step was to add in more features from the dataset to see if the accuracy could be improved on but the key here was to ensure that any features that were added to the model could only be features that were available to the insurance company at the time of the crash.

The most easily available data for the feature selection were descriptive features about the vehicle and these should be available prior to any car crash. The sensor output data was therefore excluded from the severity section of the model and the following features used instead -

MAKE - This was a categorical variable that described the make of the car and ranged from various car makes such as Toyota to Ford. This feature was thought to have an impact on the severity of the crash as it could influence the size and weight etc. It was given a number to categorise the car instead of using the name. This field was used instead of MAKED which was the text name of the car which could be miss-spelled or written differently as well as not input to an SVM model.

MODEL - This was a categorical variable and was similar to the MAKE but gave the actual model of the car which was categorised by number. Again, this was more accurate over MODELD, which was a text field, due to the potential of miss-spelled or use of different names.

YEAR - This was a interval variable and was the year the car was manufactured and was stored in the vehicle database. This could have an impact on the severity of the crash due for any number of reasons ranging from stronger, heavier materials, restrictions, engine size and safety features.

BODY - This was a categorical variable that described the body type of the vehicle which could be anything from van to 2 door saloon. The data may not have had any benefit to the model as this information was contained in the make and model data.

ENGINE - This was a categorical variable that described the type of engine in the vehicle. This may not have been contained in the other variables above as there could be cars with the same make and model but different engine types. This code was again determined to be more accurate than the text field that described it and can be an input to an SVM model with dummy variables.

ENGDSP - This was a categorical variable that described the displacement of the car. This was the total volume of the pistons and was measured in cubic centimetres. This was thought to contain information that was not contained in any of the other variables.

VEHTWT - This was a ratio variable that gave the measured test weight of the vehicle including all payload (crash test dummies etc). This gave a relatively accurate measure of how an average vehicle would weigh in a real-world scenario and therefore was assumed to generalise well.

VEHWID - This was a ratio variable which gave the vehicle width in centimetres.

VEHLEN - This was a ratio variable which gave the vehicle length in centimetres.

VEHSPD - This was ratio variable and was the actual closing speed of the vehicle as it approached impact. If there were two vehicles in the crash then this was the velocity of the two centres of gravity before impact.

PDOF - This was an interval variable and represented principle direction of force felt on the vehicle during the crash. It rotates around the vehicle centre of gravity clockwise with 0 representing a head-on collision and 180 representing a rear end collision. This was used as the response variable for second part of the model. VDI - This was a categorical variable and had seven categories of information contained in it. The only part that was needed for the study was the damage index which was the last two numbers at the end of the string. These were numbered 1 to 10 and represented increasingly larger indentations on the specified area of the car.

All of the data that was described above can be found in the NHTSA website Test reference guide[2].

[2]https://one.nhtsa.gov/Research/Databases-and-Software/NHTSA-Test-Reference-Guides

3.4 Data Exploration/Quality Report

3.4.1 Selection of Usable Data for Modelling

Due to the large level of data that was involved in the study an SQL server database was needed to be created to store the data in. This was then queried using R for analysis and modelling. The descriptive tables that were available as one downloadable file could then be uploaded to this database using the relevant fields as keys. This data could then be queried to get the Sensor Id number (CURNO field) of each file that had a VECG sensor in the test. The Sensor output files that were downloadable from the site were all named in the following format vXXXXX.CC with the X representing the test number and the C representing the sensor number. Using the sensor table in the database a script could be written to loop through all of the relevant sensors, combining them into one file per test and then adding columns with the test number, vehicle number and the axis to give a final Sensor Output table in the format set out in 3.3

Table 3.3: Sensor Output in wide format

Signal	Time	Force	TSTNO	CURNO	SENATT	AXIS	VEHNO
0	-0.125	-0.21913	1105	11	VECG	X	1
0	-0.125	-0.46495	1105	12	VECG	Y	1
0	-0.125	-3.0174	1105	16	VECG	Z	1

Unless each Signal number had a representation in the X,Y or Z directions they were removed from the dataset. Initial analysis showed to have times that were negative (See 3.3) but this was discovered to be consistent with the data description document. A minus time could occur when the first data point or time zero does not occur at the start of the reading. It was more important for the sensor to be reading out data at the same time increment to each other. if this did not happen data would not match up and had to be excluded.

The analysis was conducted on the sensor output table after all other cleaning steps were done a number of tests showed up with strange results. As can be seen in 3.6, Test 709 had what looks like a sensor malfunction at the beginning of the test and gave both the maximum sensor output in either direction for a sustained period. This resulted in an error in the test as this was determined as the point of impact where the actual point of impact appeared to be midway in the test. In order to ensure that the test was accurately measured the sensor data was only taken from time 0 of the test as it was discovered that these maximum force readouts never occurred during the crash. No crashes occurred prior to this time so it was taken to be the start point.

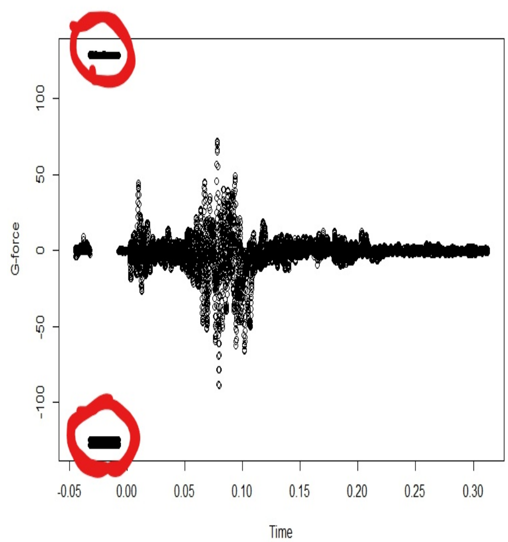

Figure 3.6: Test Example 709

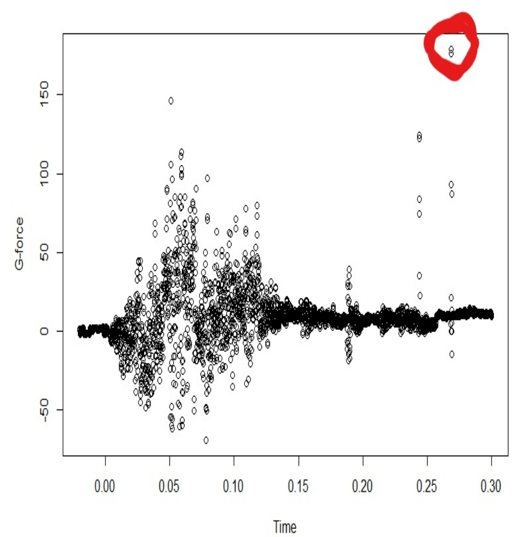

Figure 3.7: Test Example 5470

Test 5470 (see(3.7) shows what appears to be a sensor malfunction where the sensor read out a very large output at one point near the end of the test. Again this would cause an error when applying the model because it would determine the point of impact to be at the end of the dataset where it appears to have occurred a quarter of the way into the test. This readout was discovered as the length of the crash period was longer than the crash. Any tests that gave this error were examined further and removed from the final dataset.

A major issue with the selection of data for the Momentum model was there were multiple variables that needed to go into the model and because the model only worked in specific scenarios the data set used was reduced dramatically. Starting at the lowest level, the sensor output, only sensors with x, y and z data could be included which then reduced the dataset. Next, at the sensor level, only sensors with VECG as the type could be used as well as only sensors that were primary. The vehicle level data also had to be reduced as only vehicles that had a VDI record in the correct format could be used as well as excluding all vehicles that were made by NHTSA (this referred to all barriers and impact objects) and lastly the tests had to be reduced to all crashes were the tests involves a moving vehicle. The combination of all of these reductions resulted in a small data sample. Luckily this was a deductive model and was still able to give results but the problem was the number of crashes that were split between high and low was now biased.

As machine learning needs a large number of inputs to create a usable model the feature selection for the crash severity and impact angle were not as restrictive. The problem could be broken into two and the training done separately but in order for it to be an accurate comparison, it must first have been done with the same input variables. If the data was reduced to a size were the training of an accurate model could not be done, then splitting the data out and using the wider database to train the machine learning model was an option. This would give much more data to get good accuracy readings. Due to the specific selection of suitable data that was input into the momentum model a certain level of bias was that dataset and it was difficult to get accurate results on. It was therefore better to randomly select the training data set from a combination of the momentum dataset and everything else.

Another issue with data quality was in the sensor output downloadable files from the NHTSA. As the Python Selenium script looped through all of the test files and it downloaded each file and saved to the desktop. There was a total of 164,074 sensors in the database and 130,421 files were downloaded cleanly. There were 603 test files that equated to the 33,364 sensor files that returned a 'file not found' or partial download of the data. These files were excluded from the database from here on. All of these sensors were downloaded for completeness of the database and most were not needed for the models as the momentum model only used VECG.

Once all tables were in the database a major issue in data quality was the NDI. There were initially 10,790 observations in the vehicle table but the engineering classification was found to be entered incorrectly in a number of ways. There were initially 344 different ways the NDI was entered but on closer inspection, these included inputs in the incorrect format, null values and values that did not make any sense. It was decided to use regular expressions to reduce the data set down to only data points with the format of 1 to 2 numbers followed by 3 to 4 letters followed by 1 to 2 numbers.

3.5 Data Preparation

In order to ensure that accurate data was input correctly into each of the models, each of the tables were reduced to the cases that could be input to the model. This could be rolled back on in the final tests if the data was reduced to levels that were not suitable to train a model.

The tests selected from the test table were all crashes that involved a bullet vehicle. This was determined by selecting only the tests that had a TSTCFN (test classification) of - VTB (vehicle to barrier), VTI (vehicle to impactor), VTP (vehicle to pole). There were a number of cases that were VTV (vehicle to vehicle) but in order to calculate the ΔV of these crashes a more complex momentum model would have to be used. With the reduction of these cases, there were 3,559 tests left.

As mentioned the NDI needed to be in the correct format in the vehicle table. Using regular expressions to get the correct format reduced the table to 4,790 observations. Unfortunately, there was no way to determine that what was entered in the right format by the engineer is an accurate reflection of the crash. It was also important to ensure that no barriers or impactors were the subjects of the test and that they were applied to vehicles only. All cases that had a make of 'NHTSA' were therefore removed which reduced the table to 4,735.

The instrument table then needed to be reduced to ensure that the type of data that was being read into the models was in the right format. This meant that all sensors that did not have a SENATT (sensor attachment) of 'VECG' (vehicle centre of gravity) must be removed as the momentum model could work very differently on a sensor in the wrong location. Next the right type of sensor needed to be selected so only sensors with a SENTYPD (sensor type description) of 'Accelerometer' could be included. Lastly there could be a couple of different channels in the one sensor so it was important to only select cases that have a CHSTATD (Channel status description) of primary. This reduced the sensor table to 8,205 usable sensors.

3.6 Data Cleaning

On inspecting the sensor output dataset there were a number of cases that had a low number of observations or they had sensor output data that looked like a malfunction - 2663,6928,5408,6979, 5405,5408,6286,1804,5470, 6867,6220,6508,2301,3899,6867. It was therefore decided to remove them completely from the final datasets as they would give error messages if they are input into the models.

3.7 Data Construction

When creating the datasets that needed to be input into the models it was important to create new data that would act as the start point of the crash. The method used was to find the largest forces felt on the vehicle in the x,y and z-direction. This was done by adding the square of each of the forces in the 3 directions and then getting the square root of these. This ensured that no matter what direction the force was felt in, the magnitude of all of the sensor data was summed for all directions.

3.8 Modelling

3.8.1 Momentum Model

As the momentum model was a deductive model there was less work done in terms of model creation. The model used was the same as the one in a number of publications and the commissioned report. All that needed to be done was to feed the data into the model. The model used a table with a list of all the tests that were to be fed into it and all of the re-shaped sensor data was needed to loop through. For each test, the initial speed, mass and x,y,z data was feed into the model. At the point where the highest combination of forces was felt on the sensors, this was referenced as the start point of the crash. The length of the crash was then determined using a constant measure of close to half a second. The integral of the forces was then taken by multiplying the mass by the vector of the crash. Depending on the magnitude of this vector a severity was determined and lastly, the rotation of 180 degrees of the direction of the vector was taken and this was the PDOF.

3.8.2 Machine Learning Models

The machine learning section of the study was divided into two separate models. One model aimed to measure the crash severity using a classification technique called Support Vector Machine (SVM). The other section aimed to measure the angle of impact using a regression model.

SVM crash severity - an excellent technique for classification are SVMs. They are a supervised learning technique that is a non-probabilistic linear classifier. SVMs classify data points into separate groups by dividing them out using a hyperplane. The best choice hyperplane is chosen by finding the best fit line that maximises the distance between the two closest points. As the study was aimed at improving on the momentum model the starting point was to use the same input features that were selected for the momentum model. SVMs are useful in that they can be created using a relatively small number of data points yet still generalise well.

The key assumptions there were made about the SVM was that the engineering classification was an accurate measure of crash severity and that therefore when it was trained it was able to generalise to other crashes.

A major foreseeable issue that arose from this model was that the narrowed down data set applied to the momentum model was biased in that it was very specific crashes that could be used i.e. crashes that had a bullet car and an impact point. It may not, therefore, generalise to other more complex crashes and may not be very good at generalising outside the specific dataset that was chosen for the momentum model.

Regression Model - the second section of the machine learning section was to input the data points into a regression model to see if there was a relationship between the angle of impact and the same input variables that went into the momentum model.

The assumptions for this model were similar to the assumptions in the SVM model in terms of the data but there was also the added assumption that the angle of impact was a continuous variable. The angle of impact was the dependent variable and therefore had a major bearing on what type of regression model to choose. If the angle was more of a continuous variable then a simple linear regression model may have been chosen but if the angle turned out to be more of a factor than a non-linear model like logistic regression or ordinal logistic regression could be chosen.

3.9 Evaluation

All of the models that were used gave an output that was checked versus the ground truth contained in the database. The results could, therefore, be checked using a confusion matrix with the positive value being high severity in the crash and the negative value being the low severity of a crash. This gave an output of the data in the following format -

- Accuracy - Overall how often was the classifier right. This is measured as a percentage and is mathematically expressed as $Accuracy = (TP + TN)/(TP +$

$FP + TN + FN)$

- No Information Rate - Ensures that if the majority class made up too much of the dataset the accuracy of the model could be discounted. For example, if a dataset had the majority class making up 90% of the data then if the model predicts the majority class 100% of the time it would have a 90% accuracy. In order for a model to have any credibility, it must have a no information rate lower than its accuracy

- P-Value - A measure to see if the accuracy of the model was better than the no information rate (or just selecting the output by chance).

- Sensitivity - Also known as the Recall rate, was the rate at which the positive class (high severity in the crash) was predicted correctly. It can be expressed mathematically as $Sensitivity = TP/(TP + FN)$

- Specificity - Was the proportion of the negative class (low severity in the crash) that was predicted correctly and is expressed mathematically as - $Specificity = TN/(TN + FP)$

3.10 Strengths and Limitations of Approach

3.10.1 Strengths

The ability to compare all of the models outputs using the confusion matrix was a key strength meaning the overall accuracy could be compared directly against each other using a few key statistics. The ability of the machine learning models to be split out and be trained on different datasets meant there was less restrictive data that could be used to train the models if the models could not be trained on the reduced dataset for the momentum model.

The full data dictionary supplied with the dataset enabled the dataset to be understood completely in relatively short period of time and also allowed the removal of a number of irrelevant variables from the models. This coupled with the wide selection of data in the tables allowed for versatility when choosing the approach.

3.10.2 Limitations

A number of weaknesses became apparent as the study was conducted. Majority of these weaknesses came from the data itself and it is therefore inherent to this particular study.

Ground Truth - The ground truth in the database was divided into two with the crash severity being determined by the NDI and the principal direction of force being determined by PDOF in the dataset. The issue with the crash severity it is a subjective opinion from an engineer on site that is determined by visual inspection. This means that what one engineer may classify as a relatively small impact another may classify as larger or smaller.

The principal direction of force was a better measure in the database in that it was determined by where the vehicle was impacted and was a clearer definition with less ambiguity. The problem with this field was that the data set may have been heavily biased in that certain tests were more important for the NHTSA. Therefore there were many more tests of a certain angle of crash and this coupled with the reduction to a specific crash type where a bullet vehicle was involved led to more bias in the final clean dataset that was used for the models.

Data Bias - Through the literature review and also the data understanding section of the study it became apparent that the database was heavily biased towards a number of types of crashes. The NDI for the cleansed crashes was heavily biased to crashes that had a low severity and also with head-on crashes and crashes from the left. This makes sense in that the data that was selected had to have a car moving so it is unlikely that there would be a high number of rear-end crashes on these. This is backed up in the literature that says head-on collisions and lateral collisions cause severe injuries. Rear end collisions cause higher levels of injuries but these tests would not have a moving vehicle as the test case and therefore could not be used in the test.

Data Quality - On initial inspection of the dataset the number of test crashes that were available for the study where over 3000+. This was one of the reasons for selecting this study as there was enough data to have a versatile study that may give genuine insight into car crash modelling. As the dataset was cleaned however this significantly reduced the available cases that could be fed into either model which restricted the results.

Model Complexity/Selection - On the momentum model side a simple model was selected due to the literature and the complexity added into a model such as a Spring model would have required significantly more research outside of the scope of the study. The momentum model can increase in complexity adding in more features such as friction or elasticity, again subjects that are outside of the scope of the study. It was therefore decided to select a simple model that can give results that are easily compared with the outputs from the machine learning models.

When selecting the SVM, a current successful classification model was chosen that has shown in the research to be accurate in similar problem types. The issue with using the model as when the study was conducted as it was shown that a large time series was used as an input to the momentum model. Therefore when the exact same input features were selected model for model the SVM may not be the most accurate model of choice and may be better served with a time series model such as a hidden Markov model.

Similarly when selecting the model choice for the angle of impact the independent variable at the start was the angle of impact which was assumed to be a continuous variable. This in fact turned out to be more of a biased factor and may be better served with a classifier or ordinal regression model.

Chapter 4

Implementation and Results

4.1 Background

This chapter details how the experiment was performed and outlays the results of the experiment. As this experiment is a comparison of two methods the experiment followed two distinct paths, the momentum model and the machine learning models.

4.2 Business Understanding

As discussed in the previous chapter the goals that were set using the CRISP-DM methodology are as follows -

- Selecting the relevant data for the momentum model
- Combining the data from multiple datasets
- Prepare data for Momentum model
- Load data into momentum model
- Determine the accuracy of the momentum model using the ground truth in the database

- Create the severity model using same features as momentum model
- Create the PDOF model using same features as momentum model
- Determine the accuracy of the model using the ground truth in the database.
- Create the severity model using other features in the database
- Create the PDOF model using other features in the database
- Compare the models

These goals will be referenced throughout this chapter to ensure that the targets are being met throughout the experiment. Although the goals were followed it was difficult to follow them sequentially as the process was developed with the information gained from each section.

4.3 Data Understanding

This represented a major part of the study as the amount of data available in the database meant it was difficult to know which variables would be needed. The avenue chosen was to first focus on the data that was required for the momentum model and then to come back to the original data and partition it into datasets for different tests.

4.3.1 Selecting the Data

Starting with the Test table the important thing was to understand what type of information was contained in the table. The table started with 8,291 observations but this needed to be reduced substantially to be applicable to the models in the experiment. Firstly the NHTSA uses a number of different types of crashes that can involve vehicles, barriers, impactors and poles. The type of test in the table is contained in the ' TSTCFN' field in the table.

Table 4.1: Types of tests in NHTSA database

TEST CONFIGURATION	NO. TESTS
IMPACTOR INTO BARRIER	20
IMPACTOR INTO IMPACTOR	10
IMPACTOR INTO VEHICLE	2167
LANE DEPARTURE WARNING PERFORMANCE TEST	33
LOW RISK DEPLOYMENT	671
NO VALUE	15
OTHER	90
ROLLOVER	56
SLED WITH VEHICLE BODY	314
SLED WITHOUT VEHICLE BODY	521
STATIC AIR BAG TEST SIDE	474
VEHICLE INTO BARRIER	3198
VEHICLE INTO IMPACTOR	14
VEHICLE INTO POLE	347
VEHICLE INTO VEHICLE	311

As can be seen in 4.1 there are a number of types of crashes that will not be relevant to the study. As this was based on an insurance company problem and needs to be applied to vehicles on the road the data set was reduced to only cases that have a vehicle involved in a crash. Therefore all of the crashes that had a TSTCFN that began with a vehicle where included in the study. After running through the entire experiment it was decided to come back to the various types of tests and add them in to see if the dataset could be improved and the bias reduced. All impactor tests were added along with "ROLLOVER", "OTHER" and "VEHICLE INTO VEHICLE" tests. The problems that arose from this were the impactors were not added correctly to the model as they had no mass in the database. When the results were plotted against the National Damage Index (NDI) it gave a flatter relationship than was experienced in the 4.4 so it was decided to proceed using the original dataset.

The total number of tests that could be used from the test table was 3,559 as all "VEHICLE INTO VEHICLE" crashes were excluded due to the added complexity needed in the model when two vehicles are involved. In total, this represented a 57% reduction in the number of usable tests in the dataset from the original number.

Next, the vehicle table was assessed for which variables were important and which ones to reduce the dataset by. The variables in the vehicle table were all descriptive of the vehicle both before the crash and after it. Only cases that had a valid NDI were kept in the dataset and this reduced the table from 10,790 cases to 4,790 cases - a reduction of 56% in usable cases. This combined with a further reduction in the 55 cases that had were listed as an NHTSA make in the database represented a significant reduction in potential cases.

The last major reduction in the number of cases that were to be used was in the sensor table. This table needed to only have VECG sensors that were primary and were measuring accelerometer data. This represented a reduction from 422,380 cases to 8,205, a 98% reduction in the data from the original dataset. This was the most significant reduction in the sensors that could be used but also represented a different avenue to test because if this reduction could be excluded and the machine learning algorithm trained on some of this missing data it could represent a better way of modelling the crash.

4.3.2 Combining the Data

Once the data was reduced down to cases that could be applied to the momentum model they needed to be joined together to get a final dataset. The final datasets that were used in the momentum model were divided into two type. The first (4.2) had all of the data that occurred once in the test including the max force. This table acted as the test list and was looped through to select the sensor output from the main sensor output table (4.3).

The final momentum table had 190 tests to apply to the model and this was used to select 190 sets of sensor data from the sensor readout table. Each test had a vehicle

Table 4.2: Momentum Model Data Sample

Time	TSTNO	VEHNO	Force.X	Force.Y	Force.Z	absum	mag
0.0051	7	1	-44.206	23.719	-88.945	156.87	102.1175
-0.0518	518	1	44.516	153.11	144.77	342.396	215.3664
-0.0315	709	1	-125.18	129.15	-127.98	382.31	220.7456
0.0575	14	1	-5.9945	-24.484	57.872	88.3505	63.12344
-0.03138	709	1	-125.18	129.15	-127.98	382.31	220.7456
-0.029	662	1	0.83733	128.72	-128.43	257.9873	181.8345
-0.0458	476	1	44.729	150.73	142.26	337.719	212.0333
-0.02763	803	1	-138.7	139.17	-143.4	421.27	243.2479

Table 4.3: Accelerometer Sensor Output in Wide Format

Time	TSTNO	VEHNO	Force.X	Force.Y	Force.Z
0	7	1	-0.12103	0.070007	0.078013
0.000075	7	1	-0.12103	0.070007	0.078013
0.00015	7	1	-0.12103	0.070007	0.078013
0.000225	7	1	0.32882	0.5627	0.078013
0.0003	7	1	-0.12103	0.070007	0.078013
0.000375	7	1	0.32882	0.5627	0.078013

travelling at speed and one VECG sensor that output the correct data. This represented a smaller data set than was expected due to the initial size of the database but the advantage of using the physical model over the machine learning models was it did not need to be trained. It did, however, need to have enough results to show statistical significance in the results and the initial presumption was this would be difficult to prove with this dataset.

The sensor table was initially in long format as the table listed each of the sensor readouts in three separate lines so for every millisecond there were three readings with

three forces all categorised by the AXIS column. This was re-shaped to a wide format in order for each time step to have one occurrence with x,y,z force readout (see 4.3). These were combined with the data that occurred once per test i.e. the test number, vehicle number, the speed of the vehicle at impact and the vehicle weight.
The combined dataset was then used as the reference dataset as it contained all of the details that occurred once per car test and each test number could be looped through while all of the sensor data was contained in the sensor output table.

4.4 Data Preparation

Once the combined dataset was created it was loaded into the momentum model. There were a couple of things that needed to be done with the data before this loading could occur. First thing was to create a variable that could be used to determine the start point of the crash (see column "mag" in 4.2). This was determined by the largest force felt on the vehicle in the combined x,y,z planes and was determined by the following formula -

$$magnitude = \sqrt{x^2 + y^2 + z^2}$$

This new variable was added to the combined dataset so every record had a magnitude assigned to it. It was therefore easy to determine which record per crash was the impact point as it was just a case of finding the max magnitude per test number. This point acted as point 0 of the crash and the last point in the crash was determined by taking the point half a second later in the crash. All of the points in between these points were decided on as the crash data points. The half-second time length was the length of time used in the private study. This led to inaccurate results for the momentum model as each crash length can be different. Each of the sensors in the tests read the accelerometer data that was felt on the sensor so inherently the sensor also picked up the vibrations felt by each sensor on the mount in the vehicle. In order to filter out this noise low pass filtering was used in the machine learning models but it was not included in the original momentum model as created in the private study.

Lastly, the data was summarised to see which variables had different numbers of occurrences to each other and any "N/As". As the data was fed into the model it was important no "N/As" in the data were imported as it made the model crash. The initial combined data frame was actually 198 observations but by running it through the momentum model a number of anomalies were noticed as the model crashed. The following test cases were therefore removed as they did not have enough data points in the crash data or the sensor gave out strange readings such as maximum output for a sustained period before the crash (3.6) - 2663,6928,5408,6979,5405,5408,6286,1804,5470, 6867,6220,6508,2301,3899,6867.

4.5 Momentum Model

4.5.1 Crash Severity

The momentum model selected was one that was used widely in engineering to determine the ΔV of a moving body. The mass was multiplied by this to get the change in momentum. It worked by first determining the impact point of the vehicle, then taking the length of time the crash lasted for and then calculating the change in velocity over this time. Once the change in velocity was calculated it was then classified into two separate levels - high and low. The angle of impact was calculated by using the vectors calculated for the change in velocity and then rotating the vector 180 degrees to determine where the crash occurred from.

The initial impact point was determined in the data preparation section 4.4 and was -ϵ in the following model -

$$\Delta\hat{p} = \int_{timp-\epsilon}^{timp+\epsilon} m\hat{a}dt$$

The endpoint of the crash ($+\epsilon$) was taken from the private study which took around half a second of the crash as the length of time. This was determined by getting the rate of change of time and if it was larger it increased the number of steps taken in the crash. In this study, the time step was generally 0.0008 seconds so the following was done to determine the number of steps to choose as the car crash length -

$$T = \frac{0.04}{\Delta t}$$

By having the fixed number 0.04 as the numerator any changes in Δt resulted in a change in the number of points that were taken from the test to ensure the total time taken for the car crash was constant at around 0.5 seconds. This ensured the number of input variables remained constant and enabled there to be a direct comparison with the SVM model.

Once the length of the crash was determined the next thing to do was to multiply the x,y,z forces by gravity for each data point to convert the acceleration from g-force to metres per second. After this, the velocity of the vehicle was determined by getting the integral of this acceleration data using the trapezoid rule which determined the total change in velocity over the period by calculating the total area under the curve. This area was determined by breaking each point into a trapezoid and getting the area of this and adding them together. The area of a trapezoid was calculated like this -

$$\frac{1}{2}(ai + ai_{+1})\Delta t$$

In this study ai was the metres per second at any point, ai_{+1} was acceleration at the point that follows and Δt was the rate of change of time between the two points. Once the data total magnitude of the crash was calculated using the trapezoid rule the magnitude was multiplied by mass and then needed to be classified into high or low. In the literature (Shelby, 2011) there have been studies that determined that a change in velocity of 2.5 or over was considered severe or serious and below this is considered non-serious. This level was chosen to categorise the crash into high or low and is also reflected in the private study (Kevin Brosnan, 2017). The data was fed into the model and the results were taken for every crash and were added as a column into table 4.2.

This brought the experiment to the fourth goal as determined by the CRISP-DM methodology - determining the accuracy of the momentum model. The ground truth contained in the database needed to be added to the table 4.2 first. This was contained in the NDI so the last two numbers were removed as these classified the depth of the crash distance by an engineer. Looking at this data it was important to try

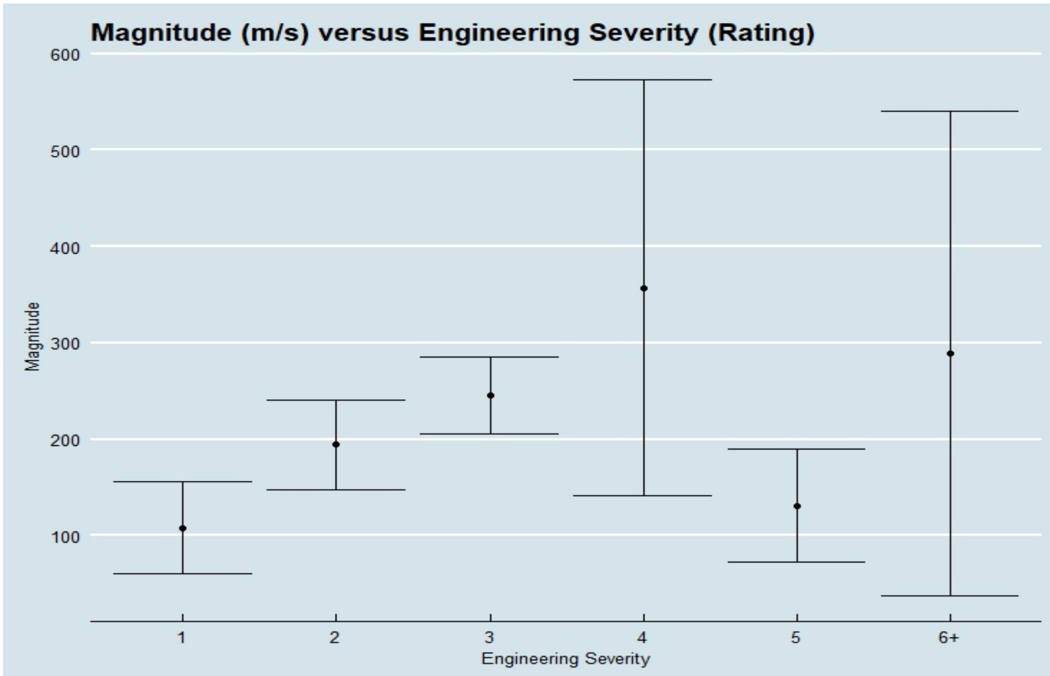

Figure 4.1: Mean Magnitude versus Engineering Classification

and establish a visual relationship in the data to ensure that the results would be accurate. A slight linear relationship was detected between the max magnitude and the engineering classification. The data was graphed with 95% confidence intervals in 4.4. As can be seen, the data seemed to be following a linear trajectory from classification levels 1 to 3 but the levels above this show very wide confidence intervals and some reductions in the magnitude of the crashes at the higher levels. This was due to the fact the data had to be reduced to the very small number of cases that could satisfy the momentum model. As can be seen in 4.4 there are a relatively high number of cases at the lower end of the classification scale but once it gets up to level 4 there is no longer a meaningful relationship as the sample space is reduced significantly. This was down to the low number of tests that were in the final dataset that had a high severity of the crash. This was unfortunate in that it also affected the no information rate in the output of the results by giving biased results to the low severity cases as there were not enough high results to compare to.

Table 4.4: Engineering Classifications Cases

ENG CLASS	# CASES
1	9
2	60
3	93
4	7
5	10
7+	8

It was decided to use the levels 1 to 3 as low severity and levels 4 and above as high severity in the study. This was to reflect the guidelines in the Vehicle Damage Scale for Traffic investigators (TEXAS, 2008). The scale clearly states that the damage in the set of photographs from scale 1 to 3 are considered minor, the damage in everything above this is considered moderate to high.

With this scale in mind the data was put through the model and then the results were compared to the ground truth using a confusion matrix 4.5. As can be seen, the data returned some disappointing results.

	Reference	
Prediction	High	Low
High	6	4
Low	19	158

Table 4.5: Confusion Matrix - Momentum Model no up-sampling

Measures	Statistics
Accuracy	88%
95% CI	77% - 88%
Sensitivity	24%
Specificity	98%
No information Rate	87%

Table 4.6: Confusion Statistics- Momentum Model no up-sampling

Although the initial look at the table suggested there was a high level of accuracy from the experiment it soon becomes obvious that this was because of the bias in the dataset towards low severity crashes. The no information rate of 87% in the dataset coupled with an accuracy of 88% suggests that this model was only accurate because the majority of the cases have a low severity. In order for the model to show any meaningful results, the data-set needed to have the no information rate reduced. This could either be done by taking a new set of results which was impossible at the time or up-sampling the data to reflect what the sample would look like when each level had the same number of tests.

With up-sampling there is no additional information added to the sample so the levels 5 and 7+ so the results of this experiment were therefore only really representative of the original sample information.

Although this data was less biased the data was also less accurate as would be expected because of the relatively linear set of data that was occurring between 1 and 4 when graphed in 4.1. The new up-sampled data had 93 tests per level.

	Reference	
Prediction	High	Low
High	67	15
Low	212	263

Table 4.7: Confusion Matrix - Momentum Model - Up-Sampled

Measures	Statistics
Accuracy	59%
95% CI	55% - 63%
Sensitivity	24%
Specificity	94%
No information Rate	50%

Table 4.8: Confusion Statistics - Momentum Model - Up-Sampled

This showed a statistically significant result of a p-value < 0.01 so the null could be rejected that the results were due to chance. The problem was the up-sampled data cannot be representative of the population because there were not enough samples of higher severity crashes.

As the data had been shown to be unreliable in determining results from the momentum model it was decided to change some of the parameters in the model to try and get more even results. The first thing done was to change the engineering classification in order to split the data more equally between high and low. This resulted in 118 cases that were high and 69 cases that were low. The problem now shifted from the being highly biased with a no information rate of 87 and an accuracy of 85 to a less biased data set of 63 and an accuracy of 42. There was less need to up-sample this dataset but it was tested anyway but it had little effect on the accuracy with an increase of 5 to 47.

Lastly, the classification rate in the momentum model was amended to try to get better results. The number used to classify a crash as high was originally set to 2.5 and this was reduced to 0.89 by trying to split the dataset with half of them high and half of them low. This gave an accuracy 55 but with a no information rate of 63 this was shown to be no improvement over the up-sampled method in 4.8.

4.5.2 PDOF

As the PDOF was determined from the same dataset and was a simple rotation of the momentum vector the data did not have to be cleaned or re-shaped for this section. The data that was being used to determine the PDOF was the PDOF field in the vehicle table. With the final sample that was fed into the momentum model a summary of the number of cases for the angle of impact was taken 4.9

Table 4.9: PDOF Distribution

Angle	0	1	20	90	270	285	297	310	330	354	356
Count	98	1	1	3	18	59	1	1	2	2	1

It was clear from this point that the data was a classification problem as there are not enough tests that occurred from continuous angles. As the tests for angles like 1 degree or 20 could represent data quality issues it was decided to bin the angles into similar categories. If the crash had an impact angle between 330 degrees and 30 this was classed as a front centre impact (FC). If it occurred between 30 and 150 this was classed as a side right impact (SR), if it was between 150 and 210 this was classed as a rear impact (BC) and if the impact occurred from 210 to 330 it was considered a left impact (SL). These classifications were run through the model and the results were compared to the PDOF field in the database.

	FC	SL	SR	BC
FC	76	23	0	0
SL	13	43	0	0
SR	10	7	3	0
BC	4	8	0	0

Table 4.10: Confusion Matrix - PDOF Momentum Model

Measures	Statistics
Accuracy	65%
95% CI	58% - 72%
Sensitivity	74% 53% 100% NA
Specificity	73% 88% 91% 93%
No information Rate	55%

Table 4.11: Confusion Statistics- PDOF Momentum Model

As can be seen, the accuracy was similar to the up-sampled engineering classification at 65% and the sensitivity and specificity are high for both in that they are over 70%. The high level of frontal impacts was due to the data having to be reduced to cars that were moving and these were coupled with the high level of side left crashes. This was reflected in the literature where head-on crashes and lateral crashes are mentioned as causing lots of damage to cars. The NHTSA database was biased towards these tests as these were the most important for car manufacturers.

Although the data could be classified into a rear shunt (BC) the data did not provide any samples which was due to the restrictive nature of the momentum model's need to have a vehicle travelling at speed i.e. there were no tests that a vehicle was travelling at speed that was crashed into.

4.6 Machine Learning - Severity Classification

Following the CRISP-DM methodology the next step was to take the data that was used in the momentum model and try to see if the results could be improved on using the machine learning models. The fact that there was a relatively small amount of data for training and validation means a Support Vector Machine was a good choice for classification. The SVM is a linear model and these are preferable if there is limited data and a need to avoid over-fitting to the data. The model was built in R using the 'e1071' package.

4.6.1 Momentum Data Model

As the study aimed to prove that the momentum model could be improved on by using machine learning the first step was to use only the data that was used in the momentum model. As there were around 50 points with an x,y, z-direction this led to a large number of features that fed into the model. Once these features were all fed in the same way it did not impact the model as they were all fed in for each crash consecutively. A problem that arose, however, was the curse of dimensionality which is the more features that are added to the model the more likely they will fit the data-source and not generalise well due to over-fitting.

Data Preparation

The first step was to take all of the points in each crash and get the x,y,z data that outputs for each point. This dataset was created and re-shaped so every line for each crash has all of the x points running from 1 to 51 followed by all of the y and z points. The data points were then low-pass filtered before downsizing to ensure there was a reasonable number of features input to the model. These points were then joined to the initial speed, mass and test number for each point giving each crash a total of 156 features and the damage level.

To start with only the data that was available to the momentum model was used, so the training and validation data sets needed to be chosen from the 190 observations. This was done randomly with fifty percent used for training and fifty percent used for validation while there was no test data set used as there were not enough observations. From the literature the following steps needed to be taken when the SVM was created -

- Transform the data to the format of the SVM package
- Conduct simple scaling on the data
- Consider the RBF kernel
- Use Cross Validation to find the best value of C and gamma
- Use the best values of C and gamma and train the whole training set
- Test

The e1071 package ensured that the data that was in categorical format was automatically converted into dummy variables and applied to the model. Similarly, the option of 'scale' was selected to ensure that the continuous data were all brought to the same scale. For every experiment done it was decided to first do it with a linear kernel and then test it with a radial kernel to ensure all results could be compared. It was also important that any model that could have the PDOF as a dependent variable was tested with the PDOF in it and without it. This ensured that if the model were to be deployed with an insurance company they could use either depending on what data was available.

Cross Validation for C and gamma

From the literature (Kohavi et al., 1995) it was decided that in order to choose the correct type of model 10 fold cross validation would be used to calculate the best estimate of accuracy for the model. C is the cost error for a soft margin when applied to a multidimensional model. The soft margin is slightly different to the margin discussed in the literature review in that the soft margin allows some points to fall on the wrong side of the margin which can lead to the model better generalising. The model used a cross-validation technique where the data was partitioned to allow training on 90% of the data and then validation on 10%. This was then done randomly on 10 different folds of the data and the average value of c, the cost error for the soft margin was taken. A large value for C is expected to give a low bias and high variance.

Gamma is the parameter of a non-linear kernel. As the data may not be linearly separable in a particular dimension it may need to be projected to a higher dimension where it is linearly separable. Gamma controls the shape of the peaks of the data in a higher dimension which can lead to points that are far from each other in lower dimensions being classified as similar or vice versa. A low gamma is expected to give a low bias and high variance. This was again tested for using the k-fold cross validation similar to the C value.

The data was run through the model with the maximum set of features that matched the momentum model. The k-fold cross validation gave a model that had a C of 2 and a sigma of 0.04548272. The output of results did not show an improvement on the momentum model as it gave an accuracy rate of 88% with the same no information rate of 88%. Interestingly the correlation matrix showed high levels of correlation (over 50%) for each data point and its following 5 occurrences. It was therefore decided to reduce the data-points to measure every 5 occurrences by doing a low pass filter on the data and downsizing it. This reduction in the number of data points that were used in the model did not give a higher rate of accuracy as the no information rate was still too high due to bias. The number of data points per crash was then varied and each output of the models but the outputs were all the same. Next, the radial model was applied to the same data using the same parameters but this gave an improvement of 1% in accuracy with the same "no information rate". This "no information rate" came back to the same problem where the data had been reduced to fit the momentum model which in turn made the data biased.

It was therefore impossible to say which model was more accurate as there was not enough data. To counter this issue the data set was up-sampled similar to the momentum model. Although no new information had been added to the dataset this up-sampling did help to see what the results would have been if there had been more tests per engineering classification.

The low level of accuracy may have been down to the high level of features in the data set so the x,y,z data were fed through a correlation matrix. It was therefore decided to reduce the data-points to measure every 5 occurrences through low pass filtering and then downsizing the data using the decimate function in R. There were two types of filters to be chosen from so 8 experiments were done with multiple downsize factors, the results of these experiments are in table 4.12. In order to counter to any shift in time that was expected to be introduced due to the decimate function a sample of 60 was taken both before and after the crash.

Table 4.12: SVM Output Using Momentum Features

Downsize Factor	0	5	10	50
Accuracy IIR filter	62%	61%	61%	61%
Accuracy FIR Filter	68%	68%	68%	68%

As can be seen the lower the number of x,y,z rate of accuracy is higher using the FIR filter and there are no differences across the various downsizing experiments. This coupled with the high correlation of every sensor readout with its next five readouts led to the decision to use the downsizing factor of 5 or 10 for future experiments. This filtering was applied to the model and it was tested both in Linear format and radial on the up-sampled data. The linear model gave promising results with a 70% accuracy, a sensitivity of 88% and specificity of 45%. The radial model gave the best results with an accuracy of 98% which was statistically significant with a p-value $<$ 0.01, shown in 4.13.

	Reference	
Prediction	High	Low
High	186	5
Low	0	134

Table 4.13: Confusion Matrix Up-sampled Radial SVM

Measures	Statistics
Accuracy	98%
95% CI	96% - 99%
Sensitivity	100%
Specificity	96%
No information Rate	57%

Table 4.14: Confusion Statistics Up-sampled Radial SVM

4.6.2 Descriptive Features Models

Model Type 1 - All features - Momentum Model Data

The next experiment aimed to maximise the potential of the NHTSA database by looking at other features in the database and building models based on these features. These were features that were descriptive of the vehicle and also gave some information about the crash. These features ranged from year, weight and length of the vehicle to the speed it was travelling at impact. In order to maximise the number of cases that could be used all of the sensor data was excluded in the first model see 4.15. Again, all of the models were tested with a linear and radial kernel and with the PDOF and without it.

	Reference	
Prediction	High	Low
High	179	27
Low	7	112

Table 4.15: Confusion Matrix Upsampled Linear SVM All Features

Measures	Statistics
Accuracy	89%
95% CI	85% - 93%
Sensitivity	96%
Specificity	80%
No information Rate	57%

Table 4.16: Confusion Statistics Upsampled Linear SVM All Features

As can be seen in 4.16, the linear model had a high level of accuracy and low level of no information rate. The problem was the up-sampling of the data may not be indicative of other car crashes. This model was then amended to a radial model and ran on the same data which gave improved results to 99% accuracy with 100% sensitivity and 97% specificity. This data gave the highest levels of accuracy throughout the study so far but the problem was the data had been up-sampled and the other issue was the PDOF was one of the features input to the model. As this may not be available to the insurance company at the crash time it was decided to test the same two models, on the up-sampled data but without the PDOF. The accuracy of the radial model deceased but the accuracy of the linear model increased by 2%. All of the models in this section were then ran through the same processes with the engineering classification rate set to 1 to 2 as low and 3 to 10 as high but none of the models showed any improvements over the previous method.

Model Type 2 - No Sensor Data

In order to have a good understanding of the potential for the data to be used in a business solution, a model was created just based on the descriptive features and the speed at the time of impact. The data available for this test was larger than the previous model as there were no crashes that had to be removed because they did not have a VECG sensor. The data was partitioned based on the new set of features and without having to use the vehicle centre of gravity sensors. This gave a dataset of 2,023 cases that could be used for the model split 1,012 for training and 1,011 for testing. This was run through the model and the results are shown in 4.17. The same technique was used to select the best C and Gamma as was done in the first model. In order to see which type of model was the best to be selected a number of tests were done with the output of the tests in **??**. The first model had all of the features selected from the vehicle table in the database. The second model used a recursive feature selection process which used a random forest method to slice the data into different size partitions and then pruned back the features that were contained in the model by their order of importance to the model. This was done multiple times (as opposed to a decision tree which is just once) and the accumulated results were given. As can be seen from 4.2 the cross-validated accuracy increased to 85% with 2 variables then decreased with 3 variables and then increased to its highest rate of accuracy with 5 variables. Again, each model was tested with a linear and radial kernel and in the first iteration, the PDOF was included as a predictive feature as it was contained in the database but a second pass was done on all of these tests that excluded the PDOF as this may not be necessarily available in a real-world example. Each model was also tested using a grid to test for c and gamma.

As can be seen, the first linear model gave a reasonable output with a 75% accuracy with a low no information rate of 58%. The fact the sensitivity and specificity are similar shows the data was not biased either.

	Reference	
Prediction	High	Low
High	321	134
Low	99	457

Table 4.17: Confusion Matrix SVM No Sensor - Linear

Measures	Statistics
Accuracy	77%
95% CI	74% - 80%
Sensitivity	76%
Specificity	77%
No information Rate	58%

Table 4.18: Confusion Statistics SVM No Sensor - Linear

Next a radial model was selected using the same data and parameters. This gave a slightly lower accuracy result of but an improved sensitivity of 84% and a reduction in specificity to 60% meaning the model classified low crashes as high on more occasions but it got the high classifications correct on more occasions.

These variables were the ones that were selected for the second test and were - YEAR, VEHSPD, PDOF, ENGINE and BODY. These were really good results as it was expected partially from reading the literature which mentions vehicle safety and ability compress on impact has increased throughout the years due to test results. The material used in today's cars are much more compressible compared to the rigid bodies of cars in the past. The speed at impact was a key feature without the sensor data to run through and contributed to the model as it was a main description of the event rather than the car. PDOF was another event description which was available in the database. Although it may not have been calculated in this test it could be something that an insurance company would have access to. The principal direction of force could be calculated at the point of impact and this could then feed the model. Lastly, the "ENGINE" and "BODY" were two descriptive features that also contributed to the

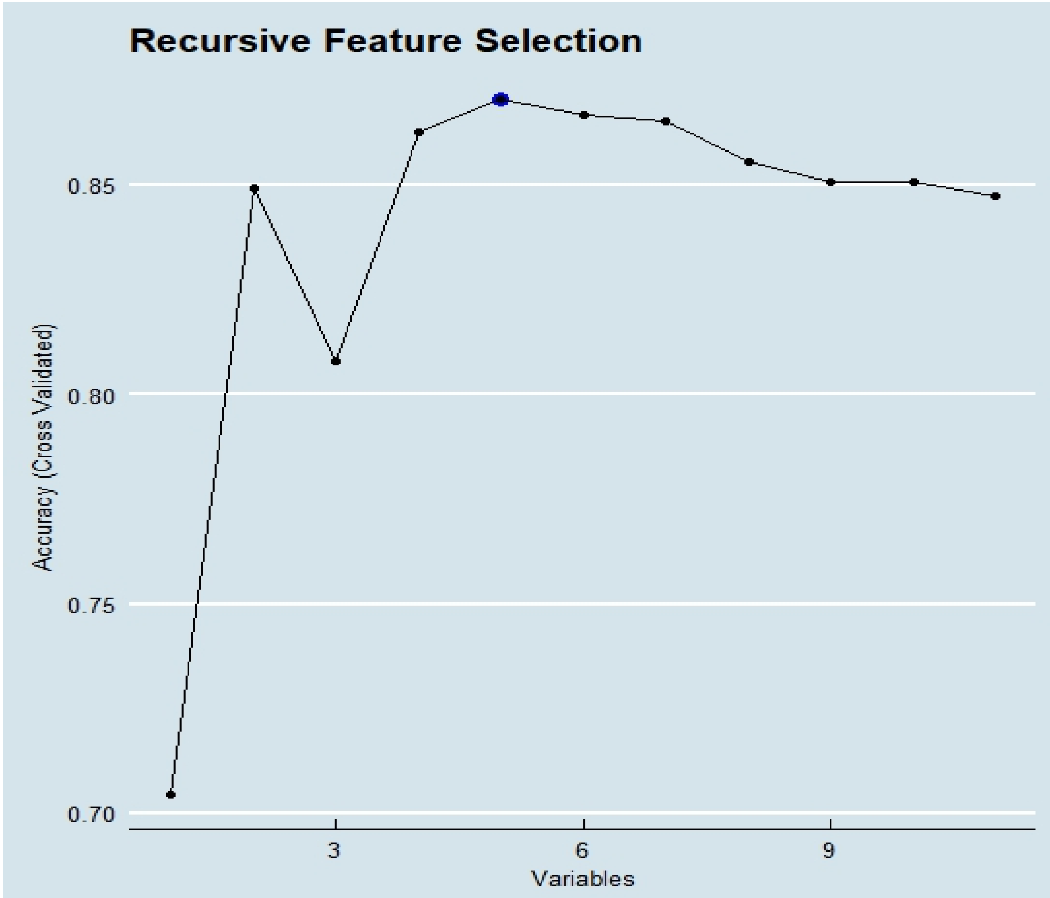

Figure 4.2: Recursive Feature Selection SVM

crash as the size of the engine and where it was located in the car could significantly increase the damage in a crash and the "BODY" differentiated between small and large cars which did also influence the model. The reduced number of features where then run through a cross-validated model to determine C and Gamma. These first three models where all input to a linear model. After this, a radial model was tested and with the total features and with the recursive features selected. The output of these tests is in 4.19.

The results showed that the Radial model that used the reduced features was the most accurate by 3% and it also had the best sensitivity/specificity split in there was only a 2% difference between the two. This coupled with how high both rates were, made this the best model for selection from this group. This was reinforced by the highest AUC (area under the ROC curve) which according to (J. Huang, Lu, & Ling,

Table 4.19: SVM Model Accuracy Comparison with PDOF

Model	Accuracy	No info rate	Sensitivity	Specificity	AUC
All features linear	73%	58%	79%	64%	0.71
Reduced features linear	70%	58%	78%	60%	0.69
Tuned all features linear	73%	58%	80%	63%	0.71
All features Radial	72%	58%	87%	52%	0.70
Tuned All features Radial	72%	58%	86%	52%	0.69
Tuned reduced features Radial	79%	58%	82%	76%	0.79

2003) is the best way of comparing models.
Once these comparisons where done the PDOF was then removed from the data and the table 4.20 were created on the same models. The removal of the PDOF meant the new reduced feature model did not use as many features as the original. Instead of using four predictive features it was now only using two - the speed at impact and the year of the car.

This is reflective of the findings in (Khattak, 2001) which found vehicle age can be a contributing factor to crash severity as the materials were stiffer in older models and do not compress as easily, therefore, absorbing some of the force. As can be seen from the two tables the tuned radial model with reduced features was the most accurate with the model containing the PDOF slightly better as the accuracy is the same but the AUC is higher at 0.79 compared to the model without the PDOF AUC of 0.77.

Table 4.20: SVM Model Accuracy Comparison no PDOF

Model	Accuracy	No information rate	Sensitivity	Specificity	AUC
All features linear	76%	58%	77%	76%	0.75
Reduced features linear	76%	58%	77%	75%	0.72
Tuned all features linear	76%	58%	78%	76%	0.75
All features non linear	74%	58%	60%	83%	0.72
Tuned All features non linear	74%	58%	62%	82%	0.72
Tuned reduced features non linear	79%	58%	80%	78%	0.77

4.7 Machine Learning - Principle Direction of Force

4.7.1 Regression

Following the CRISP-DM methodology the last section of the testing for the overall study was the PDOF. In the momentum model, the data that was used was narrowed down due to the need for the engineering classification coupled with the sensors that were a VECG sensor. As the aim was to improve on the momentum model the initial test was with the same data set. Next, the crashes that did not have an NDI were added back into the data set to improve the training and testing of the machine learning models. The initial thought when looking at the data was that the PDOF represented a continuous variable and could be solved using a continuous variable model like regression. On further inspection of the data, it appeared to be a category so it was thought a classifier would be a better model.

With the momentum data the angles were distributed unevenly and it looked like there may have been data quality issues. As can be seen in 4.9 there are a number of angle categories that showed one test per angle. This may have been in error as the vast majority of tests appear to be from specific angles. This could probably be expected considering it is test data and the literature mentions that specific types of crashes cause injuries.

The PDOF was therefore categorised into 4 groups similar to the momentum model so there could be an easy comparison - FC, SL, SR and BC. The first step was to build a multiple linear regression with the PDOF as a continuous dependent variable and all the descriptive features in the vehicle table used for the severity model were included as independent variables. The only sensor data used in the initial model was the force in the x,y and z directions for the point of impact as this could be input to the model easy enough. The model then used the AIC stepwise selection process which penalises a model for having too many features but tries to reduce the model to the least number of predictive features. The best model selected with the lowest AIC used the force in the x-direction and force in the y-direction as well as initial speed, vehicle weight and vehicle width. This was intuitive as a model but the problem was it had an adjusted R squared of 0.06107 which meant that there was a very low fit of the model to the data. The data had been partitioned into training and testing data and when the model was run over the testing data it gave a low accuracy result of 5% which was expected as the model type was not correct. It also gave some misleading results such as greater than 360 degrees or less than 0 degrees.

The next step was to try the same model with the sensor data added to the model. Then check if the model had high levels of multicollinearity as the sensor output would be related over a certain time-step. A correlation test had been done on the data previously and this led to the belief that the data over 5 steps is not as correlated. The sensor data was therefore downsampled using the same filtering technique in the severity model. The final model was still a bad fit to the data with an adjusted R-squared of 0.3426.

4.7.2 Multinomial Regression

The model used for the multinomial regression needed to include the sensor data for the crash pulse of every test that was input to the momentum model first and then a wider model was created using only all crashes that had vehicle centre of gravity data. A multinomial regression model is not supposed to use features that are related as inputs to a model. So in this case the time series of the x,y,z data cannot be used completely and instead just the output at the point of impact was used. The first model that was created used step-wise backward and forward selection and used the model with the lowest AIC.

	Reference		
Prediction	SL	FC	SR
SL	41	5	1
FC	2	41	0
SR	0	0	1

Table 4.21: Multinomial Regression Momentum Data

Measures	Statistics
Accuracy	91%
95% CI	83% - 96%
Sensitivity	95% 89% 50%
Specificity	88% 96% 100%
No information Rate	50%

Table 4.22: Confusion Statistics

The results in 4.21 were quite promising on first inspection but these results were taken in test cases that were very similar crashes and the thought would be that this model would not generalise well. This coupled with the fact that there were no rear-end crashes meant that the model would not be applicable in a real-world scenario. It was therefore decided to take the wider data in the database that was not used in the momentum model but could be used in a multinomial regression because it did not need to have an accurate NDI.

When the data was reduced down to just data that had a vehicle centre of gravity the number of cases that were available for the new model was 374 tests which enabled the splitting of the data into a training and testing dataset. The continuous model was skipped as it was felt there was no reason to create a model as it was not suitable.

Table 4.23: Multi-Level SVM Momentum Data - Accuracy Table

Model	Accuracy	Sensitivity	Specificity
Multi level SVM Linear	85%	91% 91% 78%	87% 96% 100%
Multi SVM Radial linear	79%	74% 92% 100%	92% 74% 100%
Multi SVM Linear Tuned linear	69%	96% 100% 58%	59% 100% 96%
Multi SVM Radial Tuned linear	80%	98% 100% 58%	59% 100% 98%

A multinomial regression model was created using the backward and forward stepwise selection but the final model used the vehicle weight and initial speed as predictors even though the sensor data was included in the model. Even though this model has accurate results with 78% accuracy when tested on other cases this model would not generalise well to a wider population as it is highly unlikely that initial speed and the vehicle weight are a determinant of the principal direction of force. It was therefore decided to move onto a multilevel SVM to see if this could classify better and use inputs that would be indicative of an angle of force.

4.7.3 Multi-level SVM

A multi-level SVM can be created with the same code as a binary SVM in R but the model works differently in the background. The e1071 package uses the "one against one" method of multilevel classification which is computationally intensive but performs better than the one against many. "One against one" essentially creates a machine for each pair of classes and then uses voting to see what class the point belongs to. The first step was to apply a multilevel SVM on the momentum model data to see how it would perform. This was done with both a linear and radial model as in the other sections and then both were tuned to see which could be the most accurate models.

Table 4.24: Multi-Level SVM VECG Data - Accuracy Table

Model	Accuracy	Sensitivity	Specificity
Multi level SVM Linear	74%	69% 76% NA%	80% 74% 96%
Multi SVM Radial linear	75%	73% 80% 66%	86% 76% 98%
Multi SVM Linear Tuned linear	74%	85% 0% 63%	86% 76% 98%
Multi SVM Radial Tuned linear	70%	91% 43% 40%	41% 100% 91%

As can be seen in the best performer was the linear model but due to the low number of cases that were in these models and particularly the SL impact any of these models could outperform each other with a larger dataset. It was therefore decided to use the larger dataset for the multinomial regression to try to find a model that would generalise better.

As can be seen in 4.24 the accuracy is not as good as some of the models that were used with the momentum data but it was felt these models should generalise better as there are more cases and they also take the sensor output data into account which should be a clear determinant in which direction a car was hit from.
In summary this chapter aimed to outline all the experiments carried out in this study. In all there were 50 separate experiments done - 9 in relation to the momentum model, 28 in relation to the severity classification and 13 related to the PDOF classification/regression. The table of all results are mapped out in the appendix. The next chapter aims to determine which models are the most useful and discuss any issues related to them.

Chapter 5

Analysis Evaluation and Discussion

5.1 Outline

This chapters outlines the results in their totality and aims to relate them back to the literature. It compares all models with each other and aims to draw a conclusion from the data.

5.2 Momentum Model

The aim of the experiment was to see if there was a more accurate model that could be used to measure a car crash's severity and PDOF which could be used in the insurance industry. (Kevin Brosnan, 2017) gave a good start point as it was a commissioned report and the model used in this was the one had already been used as part of a pilot project. The advantage of using this model was that it incorporated both of the variables that were required as an output. As the momentum model output results for both the PDOF and the severity, the data needed to be reduced and this led to heavy bias in the remaining data. Due to the heavy bias in the NDI, the model used in (Kevin Brosnan, 2017) did not give conclusive results that were statistically significant and the null hypothesis could not be rejected.

For experimental purposes, the data was viewed in a number of different ways to see if it was possible to draw some conclusions from the data that was available. These included up-sampling the data, changing the momentum model to classify more cases as high severity and including and excluding the PDOF in the severity model. The most accurate of these momentum models was the up-sampled version of the model suggested in (Kevin Brosnan, 2017) this was also similar to the models suggested in (Ryb et al., 2007),(Locey et al., 2012),(Lenard et al., 1998) and (Linder et al., 2003). This gave an accuracy of 65% with a confidence interval of 95% between 58% and 72% with a p-value that was statistically significant. For experimental purposes this measure was used to see if a machine learning model could improve on it but there was a caveat on the findings that the up-sampled data did not have any added information to the original dataset and as the original dataset's accuracy could not be stated due to the bias in it, this dataset could not be representative of the population. Another issue was the original model had a low sensitivity of 24% which meant that very little high severity crashes were categorised correctly while the specificity was high at 98% meaning the low categories were categorised as correct in most situations. This was reflective of the dataset being biased to the low categories but if the model had categorised everything as low it would have gotten an 87% accuracy (the no in information rate). The specificity and sensitivity of the up-sampled model only changed slightly which is indicative of the up-sampled data not containing any added information.

5.3 SVM Severity

In order to create a model that could be directly compared back to the momentum model a number of variations had to be created. The models were created based on these factors -

- Dataset - Momentum data versus wider database
- Kernel - Linear versus Radial
- Features - Sensor data included or not
- PDOF - include or not

- Model - tuned or not

These iterations led to 28 models with the top 5 models graphed versus the momentum model. Mod 1 was the up-sampled momentum model and as can be seen in the graph 5.1, this was outperformed by all of the best performing SVMs that were created. Mod 2 was an SVM that was created using all of the same features as the momentum model and up-sampled at the same rate. Mod 3 and 4 were the up-sampled momentum data but without using the sensor data and just using the initial speed at impact and the descriptive features of the car. These were fed into a Radial SVM and were the best performing in terms of accuracy. The only difference between these was Mod 3 included the PDOF while Mod 4 excluded it. The PDOF was available in the database but it may not be available for an insurance company at the time of the crash. The difference between accuracy performance of both models was minimal.
The main problem with 2,3 and 4 was the data was restricted down to cases that would fit the momentum model and because of this the training and testing data source were very similar to each other and therefore heavily biased to these type of tests. These models would certainly not generalise well to a complex situation that an everyday car crash could be and therefore not recommended for use with an insurance company.

Mod 5 and 6 used a wider range of test cases for training and testing because they did not use the sensor data for the models. This ensured a much larger dataset could be used and as can be seen the accuracy decreased significantly. This accuracy was still better than the momentum model which did not depend on a training and testing data source. Both SVMs used radial kernels and both used a random forest recursive feature selection. Mod 5 kept the PDOF in the feature selection and the model gave back 5 features which included YEAR, VEHSPD, PDOF, ENGINE and BODY. When PDOF was removed and the same feature selection was done, only VEHSPD and YEAR were kept in the model. Both models make sense as the vehicle speed and year have been found to be determinants in crash severity in (Khattak, 2001). These two models would be the ones that would generalise best to a real-world scenario as they have used the widest set of cases but they are still restricted in that the cases are all still crash tests and not actual accidents on the road. If these models

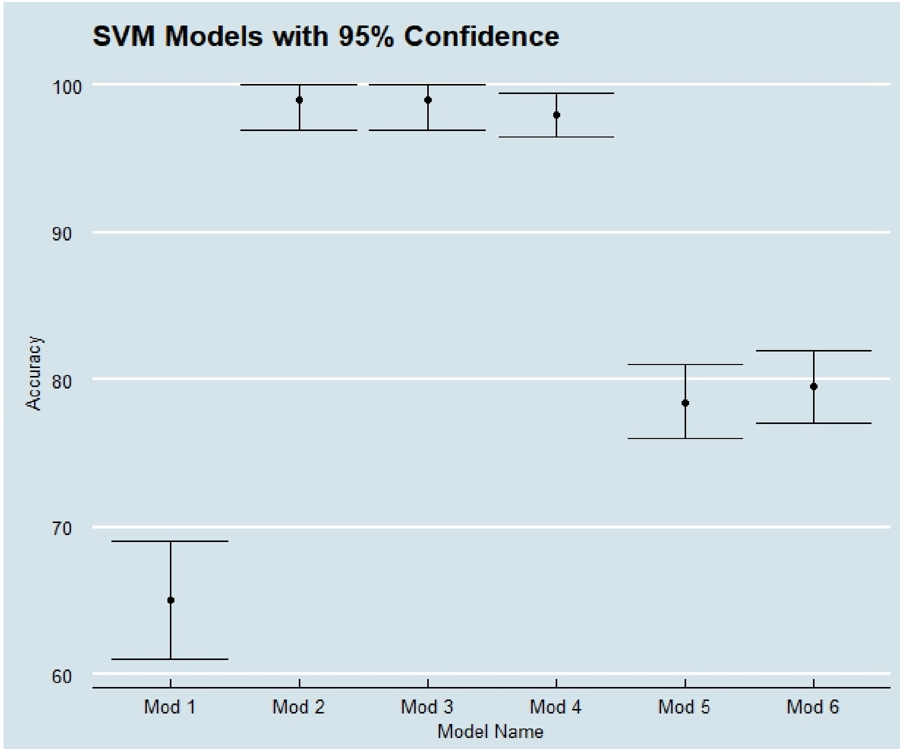

Figure 5.1: SVM Accuracy

were to be tested on real-world examples that an insurance company may have, the expectation would be for models 5 and 6 to be the best performers.

5.3.1 Hypothesis 1

Can a ML model be more accurate than a momentum model when determining the severity of an accident using g-force in the x,y,z directions as a dependent variable at time of impact?

H_0: Null Hypothesis

No statistically significant difference between the accuracy of the momentum model and all of the ML models

H_1: Alternative Hypothesis

The accuracy of the momentum model is statistically significantly lower than any of the ML models.

From the experiments conducted it is impossible to reject the null hypothesis with the data that was used. The NDI was too subjective as a measure of the crash severity and the data was too biased to be able to draw an accurate result. If the up-sampled data could be relied on the SVM far outperformed the momentum model but to reject the null based on this would be a mistake as the data may not be representative of the population.

5.4 PDOF

The PDOF experiments represented a problem in that the initial thought was that the PDOF could be considered a continuous variable. On running the variables through a Regression model it became clear that the variable was in fact a categorical one and would be better serviced with a multinomial regression model. The biggest issue with the angle of impact was the data was biased towards a PDOF of 0 degrees and 270 degrees. This is reflected in the literature where a head-on collision or lateral collision can cause severe injuries. The NHTSA therefore probably geared most of the testing towards these types of tests to measure impacts on crash test dummies. There were a number of tests from other angles but there was not a significant number so in order to counter some of the bias in the database the angles were grouped together into 4 categories.

The most accurate models that were created were graphed in 5.2 to give a visual comparison. Mod 1 was the momentum model run over the data that satisfied the momentum model. As the data that was input to this model was slightly biased but not as much as the NDI, a conclusion could be drawn from these results. The accuracy was 65% (CI 0.58 - 0.72) with a no information rate of 55% so the results were statistically significant with p-value of 0.003. This did not occur due to chance and could be compared with other models.

Mod 2 was a multinomial regression model created using the same data as the mo-

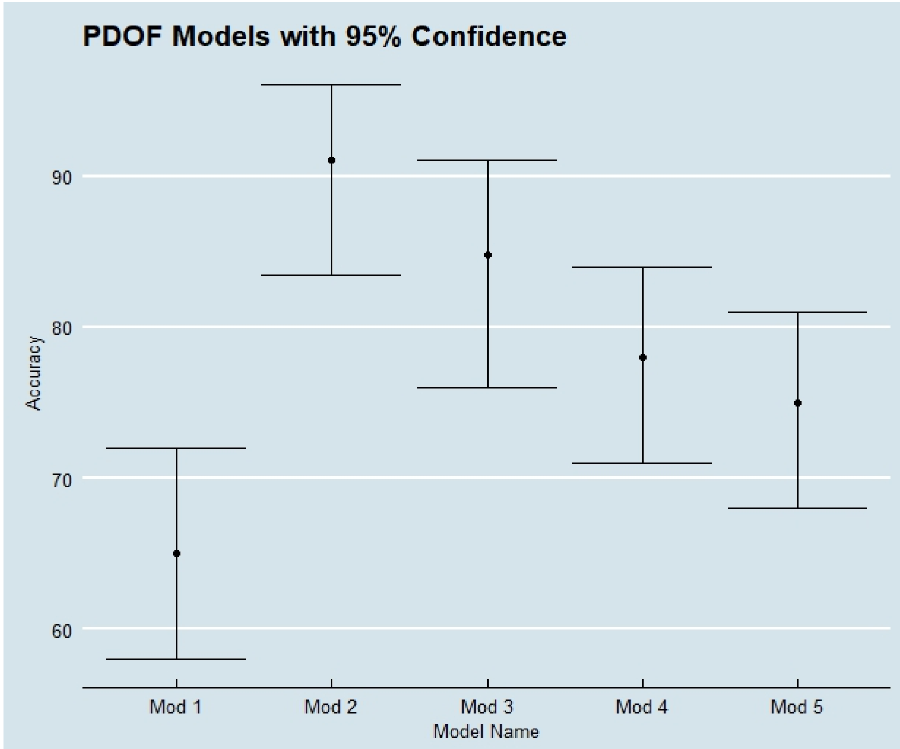

Figure 5.2: PDOF Model Accuracy

mentum model. As a multinomial regression model is built on the assumption that all data in the model is case specific and each variable is one occurrence, only one sensor output was added to the model per x, y or z-direction. The best model was selected using a forwards and back-wards stepwise algorithm but the independent variables selected for the model were initial speed and the magnitude of the crash impact. This makes sense when thought of in terms of the data that was input as each car is travelling at speed so a high magnitude impact probably indicates a head-on collision. The fact that there were only 2 examples of a side right impact and it correctly classified one may be due to chance. It got a high rate of accuracy on the side left as it got 41 of the 43 correct but this is more down to the bias in the training and testing sets and would not generalise well to the population.

Mod 3 was a multi-class Linear SVM which again outperformed the momentum model using the momentum data but this is subject to the same problems experienced by the multinomial regression. The SVM did however also include the sensor data as the

SVM can process related sensor outputs.

Mod 4 was a multinomial regression model in the same format as mod 2 but the data source used to create the model was expanded to include all cases that had a vehicle centre of gravity sensor. Even with this expansion the model was still heavily biased and the would not generalise well to a real-world scenario. Mod 5 was a radial SVM on the same data source as Mod 4 so also experienced the same data bias problems.

5.4.1 Hypothesis 2

Can a SVR or regression model be more accurate than a momentum model when determining the angle of impact of a car crash using g-forces in the x,y and z directions at time of impact?

H_0: Null Hypothesis

No statistically significant difference between the accuracy of the momentum model and all of the ML models

H_1: Alternative Hypothesis

The accuracy of the momentum model is statistically significantly lower than any of regression or SVR models.

It is safe to say that the models created from the same data-source and the wider data source in available outperformed the momentum model in terms of accuracy. These were also statistically significant so the null hypothesis which was: there is no difference between the accuracy of the momentum model and the ML models, can be rejected. These models did outperform the momentum model in these experiments but in order for one of these models to be implemented for an insurance company, more data would be needed.

5.5 Discussion

The NDI was a clear weakness in the study as it was a subjective opinion of an expert on site. The model created, could act as a replacement for this classification but would have to be amended slightly and be trained on the crush dimension rather than the subjective NDI. This could save time and money in the industry as engineers are no longer required to make an assessment of the damage to the vehicle.

The large bias in the database made the study difficult to prove. The need for very specific types of data for the momentum model was a clear limitation in that it can be difficult to get data that will satisfy these needs from the database. If the data feed was available however the model may have performed well. This was also a clear strength of the SVM model in that it could be trained on more test cases. This was also an advantage of separating the problem into two separate issues and approaching them individually. An amendment needs to be made to the momentum model to be able to handle rear shunt crashes as there are 4,391 cases in the database where the vehicle has a speed of zero so this would significantly increase the sample space and therefore should help with the bias in the momentum dataset.

One of the strengths of the momentum model was it being an inductive model and because of this it does not require a dataset to train and therefore can be tested on a low number of cases. This also doubles as a weakness, however, as it will never improve as more data is fed in while a machine learning model can improve as more data becomes available. The fact that cars travelling at zero mph during the test could not be tested, meant that a number of cases had to be excluded that could have been compared to an SVM. If the momentum model could be amended to satisfy these cases, a less biased dataset could be obtained.

The momentum model that was used also used a half second as the length of time a crash occurred for and this restriction would definitely lead to lower accuracy as a crash pulse can vary greatly.

Using the descriptive features SVM models with reduced features shows the importance of the following features in predicting a crash: YEAR, VEHSPD, PDOF, ENGINE and BODY. As these features have not been used in any way in the momentum model it shows there is significant weakness in the model. The suggestion from this study is to try to build a model that incorporates some or all of these features, as this may significantly increase accuracy.

Chapter 6

Conclusion

6.1 Research Overview

The research took the inductive momentum model and ran it using the data that was available in the NHTSA database. It researched the subject and the concepts around the model and when the severity did not give significant results the data was manipulated to create a hypothetical scenario. This meant the data was not as biased and there was an equal number of tests per NDI. 50 separate experiments were run to try to assess every way a model could be created from the data and to try to ensure that the most accurate model was also the one that could generalise to the population best.

Unfortunately, the data that was used was very specific to the types of crashes that may be useful for testing seat belt integrity or injury prevention. As the machine learning models are deductive, this narrowed the scope that they could be applied to. Data quality issues ranging from incorrect inputs and inaccurate sensor data to missing files where identified and every challenge addressed.

6.2 Problem Definition

The research done into crash tests showed that most studies use inductive models to measure crash outcomes. There were very little studies that use these models to identify the crash severity and even less that used machine learning to measure them. This may have been down to the lack of available data that uses accelerometers. No studies had been conducted yet to try to compare an inductive model against a deductive model in terms of car crashes but there are many examples of SVMs using the same data to classify other activities related to people. A clear advantage of the data was that the sensor output only ever recorded the crash event and therefore other activities could not be confused with a crash such as hard braking. This afforded the selection of the SVM as there was no danger of misclassification to another type of event. With the PDOF there were fewer studies where this was compared against. The initial thought of using a continuous regression model proved incorrect as the angle was a classification problem. Again, the major gap was there were no studies found where an accuracy of an angle from an inductive model was compared to the accuracy of an angle for a deduction model. However, similar studies were found where classification was used. The use of the momentum model compared with the two separate classifiers led to the research question: ”Can ML models match the accuracy of a momentum model for car crashes using force in x,y,z planes as the independent variables and the angle of impact or severity of crash as the dependent variables?”

6.3 Design/Experimentation, Evaluation and Results

The experiment was dictated by the data and due to this over 50 models had to be created to ensure whatever results were received could be projected back to the insurance company scenario. The first problem was the reduction of the data to the satisfy the momentum model and with this the difficulty in drawing any conclusions from the outputs. This was countered by up-sampling the data and amending the inputs to the model. The SVMs were then trained and tested as if this data was the only data available but no conclusion could be draw from it. The use of the wider data source showed the advantages of the machine learning model on the type of data it could use to train and the accuracy of this was only slightly improved if the PDOF was included in as a severity feature.

6.4 Contribution and Impacts

Although the first null hypothesis related to the crash severity could not be rejected due to the biased data, the SVM created in the hypothetical scenario was shown to outperform the momentum model.

The second part of the research gave more conclusive results as the PDOF was less biased and the results from the momentum model were statistically significant. Overall, a radial based SVM that was tuned seemed to be the most accurate model and would generalise best for the crash severity but this would not be applicable to an insurance company. It does, however, give ample evidence for a pitch to be made to the insurance company to share their data.

The PDOF was best serviced by the radial based multilevel SVM in this study and a conclusion could be drawn with the results because the results were statistically significant with p value < 0.01. The SVM did outperform the momentum model in terms of accuracy and the results were statistically significant with p value < 0.01 so the 2nd null hypothesis could be rejected.

6.5 Future Work and Recommendations

The model that was created to measure crash severity could be amended slightly to be used by to NHTSA to classify the crash automatically. This could save time and money as an engineer is not needed to supply his opinion. The model could be amended to use the crush distance to ensure it is not biased and is objective.

Suggestions for the Xtract company would be for them to amend their model to satisfy rear shunt crashes, where vehicles are not travelling at speed, this would increase the dataset that can be used to test on by potentially 4,391 cases (the number of vehicles travelling at 0 speed in the database).

The overall experiment has shown that a deductive machine learning model can be implemented using the same data or with less features than an inductive model. Given the right data source and less biased tests, there is potential for a deductive model to far outperform an inductive model. Further work with insurance company black box data would be needed to ensure this could be investigated further. Due to the nature of the data in the NHTSA database, an SVM was a suitable model of selection for this study. If multiple activities are recorded, as would be the case with insurance company data, a more suitable model could be a Hidden Markov Model or other time series model where the variations in the accelerometer data could be far greater and far more activities recorded.

The dataset is not the most suitable for training a machine learning model as the data is biased in a couple of directions but it could be used by a company to validate the accuracy of a momentum model or spring model. If this was to be done it is important the model is built to satisfy rear shunts.

The title of the study "Can Machine Learning beat Physics at modelling car crashes?" has been show to be partially possible within this thesis but in order to get more conclusive results a more complex physics model needs to be chosen in future and this needs to be complemented with a machine learning model with sufficiently unbiased data.

References

Abdel-Aty, M., & Abdelwahab, H. (2004). Analysis and prediction of traffic fatalities resulting from angle collisions including the effect of vehicles configuration and compatibility. *Accident Analysis & Prevention*, *36*(3), 457–469.

Alpaydin, E. (2014). Introduction to machine learning. MIT press.

Bennett, K. P., & Bredensteiner, E. J. (2000). Duality and geometry in svm classifiers. In *Icml* (pp. 57–64).

Bridle, J. S. (1990). Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neurocomputing* (pp. 227–236). Springer.

Bundorf, R. T. (1996). *Analysis and calculation of delta-v from crash test data* (Tech. Rep.). SAE Technical Paper.

Cheng, W.-C., & Jhan, D.-M. (2013). Triaxial accelerometer-based fall detection method using a self-constructing cascade-adaboost-svm classifier. *IEEE journal of biomedical and health informatics*, *17*(2), 411–419.

Chong, M., Abraham, A., & Paprzycki, M. (2005). Traffic accident analysis using machine learning paradigms. *Informatica*, *29*(1).

Evans, L. (1994). Driver injury and fatality risk in two-car crashes versus mass ratio inferred using newtonian mechanics. *Accident Analysis & Prevention*, *26*(5), 609–616.

Evans, L. (1996). Safety-belt effectiveness: the influence of crash severity and selective recruitment. *Accident Analysis & Prevention*, *28*(4), 423–433.

Flach, P. (2012). Machine learning, the art and science of algoritms that make sense of data. In *Icml.* Cambridge.

Gabauer, D. J., & Gabler, H. C. (2006). Comparison of delta-v and occupant impact velocity crash severity metrics using event data recorders. In *Annual proceedings/association for the advancement of automotive medicine* (Vol. 50, p. 57).

Gabauer, D. J., & Gabler, H. C. (2008). Comparison of roadside crash injury metrics using event data recorders. *Accident Analysis & Prevention*, *40*(2), 548–558.

Gabler, H. C., Hampton, C., & Roston, T. (2003). Estimating crash severity: can event data recorders replace crash reconstruction? In *Proceedings of 18th international technical conference on the enhanced safety of vehicles, held nagoya, japan, 19-22 may 2003.*

Handel, P., Skog, I., Wahlstrom, J., Bonawiede, F., Welch, R., Ohlsson, J., & Ohlsson, M. (2014). Insurance telematics: Opportunities and challenges with the smartphone solution. *IEEE Intelligent Transportation Systems Magazine*, *6*(4), 57–70.

Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., & Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their applications*, *13*(4), 18–28.

Hsu, C.-W., Chang, C.-C., Lin, C.-J., et al. (2003). A practical guide to support vector classification.

Huang, C.-L., & Wang, C.-J. (2006). A ga-based feature selection and parameters optimizationfor support vector machines. *Expert Systems with applications*, *31*(2), 231–240.

Huang, J., Lu, J., & Ling, C. X. (2003). Comparing naive bayes, decision trees, and svm with auc and accuracy. In *Data mining, 2003. icdm 2003. third ieee international conference on* (pp. 553–556).

Huang, L., & Chen, J. C. (2001). A multiple regression model to predict in-process surface roughness in turning operation via accelerometer. *Journal of Industrial Technology*, *17*(2), 1–8.

Huang, M. (2002). *Vehicle crash mechanics.* CRC press.

Joachims, T. (n.d.). *Introduction to support vector machines.*

Kamijo, S., Matsushita, Y., Ikeuchi, K., & Sakauchi, M. (2000). Traffic monitoring and accident detection at intersections. *IEEE transactions on Intelligent transportation systems*, *1*(2), 108–118.

Karantonis, D. M., Narayanan, M. R., Mathie, M., Lovell, N. H., & Celler, B. G. (2006). Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *IEEE transactions on information technology in biomedicine*, *10*(1), 156–167.

Kevin Brosnan, R. G. J. G. M. G. C. H. A. H. S. M. K. D. Y. K. M. L. D. M. C. N. S. O. . S. O., Michael Coughlan. (2017). Estimating vehicle collision damage using motion data. *Mathematics Applications Consortium for Science and Industry*.

Khan, A. M., Lee, Y.-K., Lee, S. Y., & Kim, T.-S. (2010). A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer. *IEEE transactions on information technology in biomedicine*, *14*(5), 1166–1172.

Khattak, A. (2001). Injury severity in multivehicle rear-end crashes. *Transportation Research Record: Journal of the Transportation Research Board*(1746), 59–68.

Kohavi, R., et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai* (Vol. 14, pp. 1137–1145).

Kullgren, A., Krafft, M., Nygren, Å., & Tingvall, C. (2000). Neck injuries in frontal impacts: influence of crash pulse characteristics on injury risk. *Accident Analysis & Prevention*, *32*(2), 197–205.

Lenard, J., Hurley, B., & Thomas, P. (1998). The accuracy of crash3 for calculating collision severity in modern european cars.

Li, X., Lord, D., Zhang, Y., & Xie, Y. (2008). Predicting motor vehicle crashes using support vector machine models. *Accident Analysis & Prevention*, *40*(4), 1611–1618.

Linder, A., Avery, M., Krafft, M., & Kullgren, A. (2003). Change of velocity and pulse characteristics in rear impacts: real world and vehicle tests data..

Locey, C. M., Garcia-Espana, J. F., Toh, A., Belwadi, A., Arbogast, K. B., & Maltese, M. R. (2012). Homogenization of vehicle fleet frontal crash pulses from 2000–2010. In *Annals of advances in automotive medicine/annual scientific conference* (Vol. 56, p. 299).

Mayagoitia, R. E., Nene, A. V., & Veltink, P. H. (2002). Accelerometer and rate gyroscope measurement of kinematics: an inexpensive alternative to optical motion analysis systems. *Journal of biomechanics*, *35*(4), 537–542.

McConnell, W. E., Howard, R. P., Guzman, H. M., Bomar, J. B., Raddin, J. H., Benedict, J. V., . . . Hatsell, C. P. (1993). *Analysis of human test subject kinematic* *responses to low velocity rear end impacts* (Tech. Rep.). SAE Technical Paper.

Neades, J., & Smith, R. (2011). The determination of vehicle speeds from delta-v in two vehicle planar collisions. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, *225*(1), 43–53.

Pawlus, W., Karimi, H. R., & Robbersmyr, K. G. (2011a). Development of lumped-parameter mathematical models for a vehicle localized impact. *Journal of mechanical science and technology*, *25*(7), 1737–1747.

Pawlus, W., Karimi, H. R., & Robbersmyr, K. G. (2011b). Mathematical modeling of a vehicle crash test based on elasto-plastic unloading scenarios of spring-mass models. *The International Journal of Advanced Manufacturing Technology*, *55*(1), 369–378.

Preece, S. J., Goulermas, J. Y., Kenney, L. P., Howard, D., Meijer, K., & Crompton, R. (2009). Activity identification using body-mounted sensorsa review of classification techniques. *Physiological measurement*, *30*(4), R1.

Raschka, S. (2015). *Python machine learning.* Packt Publishing Ltd.

Ravi, N., Dandekar, N., Mysore, P., & Littman, M. L. (2005). Activity recognition from accelerometer data. In *Aaai* (Vol. 5, pp. 1541–1546).

Richter, C., King, E., Falvey, E., & Franklyn-Miller, A. (2018). Supervised learning techniques and their ability to classify a change of direction task strategy using kinematic and kinetic features. *Journal of biomechanics*, *66*, 1–9.

Richter, M., Otte, D., Pohlemann, T., Krettek, C., & Blauth, M. (2000). Whiplash-type neck distortion in restrained car drivers: frequency, causes and long-term results. *European Spine Journal*, *9*(2), 109–117.

Robinette, R. D., Fay, R. J., & Paulsen, R. E. (1994). *Delta-v: basic concepts, computational methods, and misunderstandings* (Tech. Rep.). SAE Technical Paper.

Ryb, G. E., Dischinger, P. C., Kufera, J. A., & Burch, C. A. (2007). Delta v, principal direction of force, and restraint use contributions to motor vehicle crash mortality. *Journal of Trauma and Acute Care Surgery*, *63*(5), 1000–1005.

Selmanaj, D., Corno, M., & Savaresi, S. M. (2014). Accelerometer-based data-driven hazard detection and classification for motorcycles. In *Control conference (ecc), 2014 european* (pp. 1687–1692).

Shelby, S. G. (2011). Delta-v as a measure of traffic conflict severity..

Stansfield, B., Hillman, S., Hazlewood, M., & Robb, J. (2006). Regression analysis of gait parameters with speed in normal children walking at self-selected speeds. *Gait & posture*, *23*(3), 288–294.

Steinwart, I., Hush, D., & Scovel, C. (2009). Learning from dependent observations. *Journal of Multivariate Analysis*, *100*(1), 175–194.

Swartz, A. M., Strath, S. J., Bassett, D. R., Obrien, W. L., King, G. A., & Ainsworth, B. E. (2000). Estimation of energy expenditure using csa accelerometers at hip and wrist sites. *Medicine & Science in Sports & Exercise*, *32*(9), S450–S456.

TEXAS, S. O. (2008). *Vehicle damage scale for traffic accident investigators.* National Safety Council.

Tolkiehn, M., Atallah, L., Lo, B., & Yang, G.-Z. (2011). Direction sensitive fall detection using a triaxial accelerometer and a barometric pressure sensor. In *Engineering in medicine and biology society, embc, 2011 annual international conference of the ieee* (pp. 369–372).

Torrão, G., Coelho, M., & Rouphail, N. (2013). Binary classification and logistic regression models application to crash severity. In *World conference on transportation research (wctr). rio de janeiro, brazil.*

Vangi, D. (2009). Simplified method for evaluating energy loss in vehicle collisions. *Accident Analysis & Prevention*, *41*(3), 633–641.

Varat, M., & Husher, S. E. (2003). Crash pulse modelling for vehicle safety research. In *Proceedings: International technical conference on the enhanced safety of vehicles* (Vol. 2003, pp. 9–p).

Wirth, R., & Hipp, J. (2000). Crisp-dm: Towards a standard process model for data mining. In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining* (pp. 29–39).

Yu, R., & Abdel-Aty, M. (2013). Utilizing support vector machine in real-time crash risk evaluation. *Accident Analysis & Prevention*, *51*, 252–259.

Zeidler, F., Schreier, H.-H., & Stadelmann, R. (1985). *Accident research and accident reconstruction by the ees-accident reconstruction method* (Tech. Rep.). SAE Technical Paper.

Zhao, L., Pawlus, W., Karimi, H. R., & Robbersmyr, K. G. (2014). Data-based modeling of vehicle crash using adaptive neural-fuzzy inference system. *IEEE/ASME Transactions on mechatronics*, *19*(2), 684–696.

Table 6.1: Total Accuracy Results

Model	Eng Class	Type	Data	Acc	Sens	Spec	No info
Mom Sev	1-3 L	2.5	Mom	88	24	98	87
Mom Sev	1-3 L	2.5	Mom Up	65	37	92	50
Mom Ang	1-3 L	2.5	Mom	65	73	74	55
Mom Sev	1-2 L	2.5	Mom	42	13	93	63
Mom Sev	1-2 L	2.5	Mom Up	50	26	98	67
Mom Ang	1-2 L	2.5	Mom	65	54	55	63
Mom Sev	1-2 L	0.89	Mom	55	54	55	63
Mom Sev	1-2 L	0.89	Mom Up	61	62	49	67
Mom Ang	1-2 L	0.89	Mom	65	73	74	55
SVM Sev	1-3 L	Linear	Mom	88	0	100	88
SVM Sev	1-3 L	Radial	Mom	89	0	100	88
SVM Sev	1-3 L	Linear	Mom Up	69	83	50	57
SVM Sev	1-3 L	Radial	Mom Up	99	100	96	57
SVM Sev w PDOF	1-3 L	Linear	Mom no sen	89	100	75	57
SVM Sev w PDOF	1-3 L	Radial	Mom no sen	99	100	97	57
SVM Sev n PDOF	1-3 L	Linear	Mom no sen	91	100	79	57
SVM Sev n PDOF	1-3 L	Radial	Mom no sen	98	100	95	57

SVM Sev time	1-3 L	Linear	Mom 30	87	94	79	57
SVM Sev	1-2 L	Linear	Mom	64	94	11	63
SVM Sev	1-2 L	Radial	Mom	65	93	18	63
SVM Sev w PDOF	1-2 L	Linear	Mom no sen	87	96	66	71
SVM Sev w PDOF	1-2 L	Radial	Mom no sen	99	100	97	57
SVM Sev n PDOF	1-2 L	Linear	Mom no sen	84	93	63	71
SVM Sev n PDOF	1-2 L	Radial	Mom no sen	88	99	59	71
SVM Sev time	1-2 L	Linear	Mom 30	86	82	94	71
SVM Sev	1-3 L	Linear	Des	75	77	73	58
SVM Sev	1-3 L	Linear	Des Red	73	76	68	58
SVM Sev	1-3 L	Linear	Des Red Tun	76	77	74	58
SVM Sev	1-3 L	Radial	Des	74	84	60	58
SVM Sev	1-3 L	Radial	Des Red	73	82	61	58
SVM Sev	1-3 L	Radial	Des Red Tun	78	81	74	58
SVM Sev n PDOF	1-3 L	Linear	Des	73	79	64	58
SVM Sev n PDOF	1-3 L	Linear	Des Red	70	78	60	58
SVM Sev n PDOF	1-3 L	Linear	Des Red Tun	73	80	63	58
SVM Sev n PDOF	1-3 L	Radial	Des	72	87	52	58
SVM Sev n PDOF	1-3 L	Radial	Des Red	72	86	52	58
SVM Sev n PDOF	1-3 L	Radial	Des Red Tun	79	82	76	58

Regression	1-3 L	Continuous	Mom	3	NA	NA	60
Regression	1-3 L	Continuous	Reg	9	NA	NA	60
Regression	1-3 L	Multinom	Mom	91	95 85 50	87 96 100	52
SVM Ang Cat	1-3 L	Linear	Mom	85	91 78 NA	90 91 99	52
SVM Ang Cat	1-3 L	Radial	Mom	79	74 92 100	92 74 100	52
SVM Ang Cat	1-3 L	Linear Tun	Mom	69	96 100 58	59 100 96	52
SVM Ang Cat	1-3 L	Radial Tun	Mom	80	98 100 58	59 100 98	52
Regression	1-3 L	Multi nom cat	Reg	78	NA	NA	60
Regression	1-3 L	Muli nom all	Des Red Tun	69	88	57	58
SVM Angl	1-3 L	Linear	Des	74	69 76 NA	80 74 96	58
SVM Ang Cat	1-3 L	Radial	Des Red	75	73 80 66	86 76 98	68
SVM Ang Cat	1-3 L	Linear Tun	Des Red Tun	74	73 80 66	86 76 98	60
SVM Ang Cat	1-3 L	Radial Tun	Des Red Tun	70	91 43 40	41 100 91	60

All code for these experiments is available at this github address -
https://github.com/gabtab/Thesis-Crash-Machine-Learning