Title
Severity Prediction and Time-Series Analysis of Vehicle Accidents Using Statistical Models

Author
Kaunitz, Lisa

Publication Date
2022

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Severity Prediction and Time-Series Analysis of Vehicle Accidents Using Statistical Models

A thesis submitted in partial satisfaction

of the requirements for the degree

Master of Science in Applied Statistics

by

Lisa Kaunitz

2022

ABSTRACT OF THE THESIS

Severity Prediction and Time-Series Analysis of Vehicle Accidents Using Statistical Models

by

Lisa Kaunitz
Master of Science in Applied Statistics
University of California, Los Angeles, 2022
Professor Frederic Paik Schoenberg, Chair

This study explores factors that effect vehicle accidents, predicts the severity of accidents through logistic regression, and forecasts the number of future accidents to occur using time-series analysis. From insights gathered during exploration, a final dataset is prepared for the use of a logistic regression model. The final model predicts whether or not an accident will be severe with an accuracy of 82%, and reveals the three main features that statistically contribute to the odds of an accident having a severe impact on traffic. Finally, a time-series analysis is run in order to model the number of accidents that can occur on a given day using historical data. This paper evaluates the dataset in ways that have yet to be explored, and provides a great baseline understanding of what is possible for the future of transportation.

The thesis of Lisa Kaunitz is approved.

Vivian Lew

Yingnian Wu

Frederic Paik Schoenberg, Committee Chair

University of California, Los Angeles

2022

To my parents . . .

who have been incredible role models

and supported me through every stage of life

TABLE OF CONTENTS

1 Introduction **1**

2 Goals and Areas of Interest **3**

3 The Dataset **5**

3.1 Obtaining the Data 6

3.2 Data Engineering 6

3.3 Final Dataset 8

4 Exploratory Data Analysis **12**

4.1 Number of Accidents by State 12

4.2 Number of Accidents by Accident Signals in CA 14

4.3 Number of Accidents per Year in CA 15

4.4 Number of Accidents per Hour in CA 16

4.5 Time-Series Visualization 17

5 The Models **19**

5.1 Logistic Regression for Severity Prediction 19

5.1.1 Results 24

5.2 ARMA Model for Time-Series 26

5.2.1 Results 30

6 Conclusion **32**

7 Limitations and Future Work **34**

References **36**

LIST OF FIGURES

3.1 Data Creation Process Architecture per the Countrywide Traffic Accident Dataset 5

4.1 Distribution of the Number of Accidents by State in the Full Dataset. 13

4.2 Distribution of the Number of Accidents by Accident Signals in the State of CA. 14

4.3 Distribution of the Aggregate Number of Accidents per Hour in the State of CA. 16

4.4 Time-Series Visualization of Accident Trends in the State of CA by Month and by Days of the Week. 18

5.1 Receiver Operating Characteristic (ROC) Curve for the Final Logistic Regression Model. 25

5.2 Time-Series Representation of Accidents in CA from 2016 to 2021. 27

5.3 Auto Correlation Function (ACF) Plot of full Time-Series data in CA. . . . 28

5.4 Residuals Diagnostic Plots: Time-Series, ACF, and Distribution. 29

5.5 Time-Series Forecast (window from 2019 to 2023) 31

LIST OF TABLES

4.1 Count of the Number of Accidents per Year in the State of CA. 15

5.1 Count of the Number of Accidents per Severity Level in the State of CA. . . 19

5.2 Final Logistic Model Output for Binary Severity Prediction 22

5.3 Odds Ratio Estimates and Confidence Intervals of Logistic Regression Model. 23

5.4 Confusion Matrix of Actual vs. Predicted Severity. 24

ACKNOWLEDGMENTS

(Acknowledgments omitted for brevity.)

CHAPTER 1

Introduction

If you have been driving for the past 18 years, you will likely have experienced the ramifications of being in an automobile crash multiple times [1]. Every time a driver gets on the road, they have a 1 in 103 chance of dying in a car crash, making it one of the riskiest decisions we make daily. While these numbers are staggering, the automobile industry has slowly added more safety features to cars: first starting with the introduction of backup cameras, then self-parking capabilities, and now a level of autonomous driving. These features have been leading up to full autonomy on the road.

In addition to the risk of safety that driving has on individuals, there is also an economic cost of traffic accidents. The annual financial cost of motor vehicle crashes in California alone is 19.998 billion dollars [2]. It nearly costs the United States 1 trillion dollars a year [3]. It is estimated that three-quarters of the total costs come from insurance and medical expenses, higher taxes, and adverse economic effects of congestion. More acutely, the severity of accidents - for this data, meaning the impact on traffic - leaves vehicles idle on freeways, resulting in more gas emissions and increased frequency of filling up gas.

Another impact that should not be overlooked is the effect on the environment. Vehicle accidents can often result in gas and fluid leaks that emit harmful chemicals into the atmosphere. On a graver scale, accidents that result in totaled cars leave parts that will eventually end up in landfills. While the issue of gas emissions is actively being worked on with the innovation seen in the electric vehicle market, there is still an environmental cost associated until we see a majority of electric vehicles.

With the three effects of vehicle accidents highlighted being human, economic, and environmental, we can see that it is worth finding a way to best predict the severity of accidents.

If we can find the variables that have a statistical significance in the number of traffic accidents, we can use that information to try to minimize the number of accidents seen across the country. Additionally, by running a time-series analysis on historical daily accident data to predict future accident data, we can have more information that will allow us to make better-informed decisions.

In this paper, I developed two areas of research concentrating on a time-series analysis forecasting vehicle accidents, specifically in the state of California, between the years 2016 to 2021. First, is building a model for predicting the severity of accidents, and examining the key factors that affect the severity of an accident, which can give insight into how to prevent accidents from occurring. Finally, an ARMA model will be created to forecast the daily number of accidents using historical data.

CHAPTER 2

Goals and Areas of Interest

Traffic congestion affects most of the working population worldwide. Aside from the three effects stated above - human, economic, and environmental - congestion becomes part of our everyday lives. The most significant growth areas in the automotive and transportation industries have been geared towards relieving these pains two-fold: with the implementation of electric vehicles (EVs) and autonomous vehicles (AVs).

Electric vehicles have an electric motor instead of an internal combustion engine and a battery instead of a gasoline tank. The first fully electric car was introduced in 1890 by William Morrison, a chemist from Des Moines, Iowa [4]. The history of EVs is extensive; however, they made a mass reappearance at the start of the 21st century. The Prius became the world's first mass-produced hybrid electric vehicle, released in Japan in 1997. The second most significant turning point was in 2006 when Tesla Motors announced it would start producing fully electric luxury vehicles for the masses. Since then, almost all automakers have accelerated their work on creating EVs for the newly adjusted market. EV production by automakers is only part of the equation; there are many other moving parts, like consumer adaptation, regulations, battery production, and refueling infrastructure. With many industry forecasts, the consensus is that about half of the cars on the road will be electric by 2050 [5], which would drive the goal in the US of being carbon neutral, alleviating much of the environmental impact of congestion.

Autonomous vehicles, also known as self-driving cars, are capable of operating and transporting passengers without any input from a human driver. The first autonomous capability to hit the US market was in 1945 with the introduction of cruise control [6]. We can bucket autonomous technology into six levels: Level 0 has no driving automation and is manually

controlled. The data used for this study will be coming from vehicles with level 0 capabilities. Level 1 is categorized as driver assistance, including adaptive cruise control and breaking. Level 2 is partial driving automation, where the vehicle can control both steering and accelerating/ decelerating. The most popular vehicle on the road with level two automation include the Tesla Autopilot. Level 3 is categorized as conditional driving automation, which differs from the previous level by having the ability to make informed decisions with environmental data. Level 4 is high driving automation, which introduces the ability for the vehicle to intervene if things go wrong and would not require human interaction in most circumstances; however, a human still has the option to override manually. Companies like Waymo are introducing level four self-driving taxis to the market. The last fully autonomous level is Level 5, where the vehicle can communicate with other vehicles, and a steering wheel and pedals are not even included for a human driver [7]. The idea behind fully-autonomous vehicles stems from a potential solution for decreasing traffic congestion and improving the safety of cars. The main concerns raised with full AVs revolve around legislative, legal, and technological fears of system failures. Studies have been conducted to investigate the effects of the automated vehicle on driver's behavior and traffic performance. In a conducted research paper, Aria found that average travel time improved by 9 percent in an AV scenario [8].

This study is motivated by the culmination of everyday congestion problems and the modern-day solutions being worked on. While the dataset is not based on information coming from EVs or AVs specifically, modeling key factors that lead to traffic congestion and vehicle accidents will allow for better input when implementing these two features as a means of everyday transportation. Studies like this can shed light on how much the human factor plays a role in vehicle accidents based on exterior features and give a more substantial level of confidence for the societal adaptation to EVs and AVs once more regulations become implemented. For example, predicting the severity of traffic accidents will give an insight into how much congestion is created per day and look at the key factors contributing to accidents. Building a time-series model to forecast the number of accidents provides a new understanding and framework of the data which has yet to be explored.

CHAPTER 3

The Dataset

The original dataset, formally titled "A Countrywide Traffic Accident Dataset," was collected by Sobhan Moosavi from The Ohio State University. The data is open source and available via Kaggle, among other research paper databases. What makes this dataset unique is the large-scale information and APIs used to collect several environmental and contextual variables. As stated above, multiple APIs provide streaming traffic, environmental, and event data. The primary forms of collecting the streaming data were from two real-time providers, "MapQuest Traffic" [9] and "Microsoft Bing Map Traffic" [10]. The raw data is collected over 49 states of the US, starting from February 2016 to the most updated version, December 2021. There are about 2.8 million observations recorded in the raw dataset with 47 features.

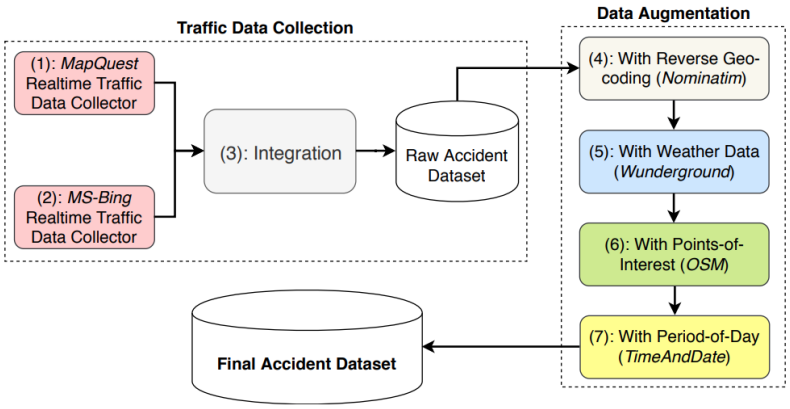

Figure 3.1: Data Creation Process Architecture per the Countrywide Traffic Accident Dataset

Figure 3.1 above was adapted from the original collectors of the data and included for background and visual understanding of the data ingestion process [11].

3.1 Obtaining the Data

The data were obtained via a comma-separated values download from the Kaggle datasets website [12] and imported into R. Kaggle has many open-source datasets and competitions with the purpose of solving data science challenges, being used as a tool for teaching beginners, and working with other data scientists to share code and kernels. Aside from the collaborative nature that Kaggle provides, the datasets are generally pretty clean once uploaded, which makes it an attractive source of data.

The original dataset contained 47 features that can be generally bucketed into four categories of variables: time, geographic, environmental factors, and point of interest road features (POIs). Due to the number of geographic variables, I began by removing redundant variables that did not pertain to my research questions. Generally, the number of variables makes this dataset very usable for a wide range of research topics and many applications. There are more than enough possibilities to explore topical or geospatial effects, given the granularity of the data. For example, it is feasible to look at accidents within a three-mile radius of your neighborhood, given many observations and variables. However, since this paper will be looking at the key features that contribute to vehicle accidents, severity prediction, and time-series forecasting, my first step was to eliminate redundant variables.

3.2 Data Engineering

Since the data was downloaded via Kaggle and was created for research purposes, it was already pretty clean in its raw format. However, to get the data in a usable form that makes sense for this paper, I focused on these five areas of data cleaning: removing, creating, transforming, dealing with NAs, and grouping variables.

Each of the focuses of this paper would require a different data frame. For example, it is worth keeping as much original information as possible to understand the whole dataset when doing general exploratory analysis. Then, when looking at the prediction of severity, the variables of interest change depending on the model used. Finally, to do a time-series

ARMA model, the data must be pivoted such that it is indexed by the date and has only one column showing the number of accidents on each day. The ARMA model will not utilize any of the explanatory variables, so the first two research questions are essential to understanding the full scope of the dataset.

Removing: The first variables removed were ID, Country, Number, Airport Code, and the environmental time variables such as Sunrise Sunset, Civil Twilight, Nautical Twilight, and Astronomical Twilight. The ID variable was used as an index for the data and was unnecessary for the analysis. I decided to remove the Country variable because all the data came from the United States, so it did not provide any information. The Number variable showed the street number of an address field, which I got rid of because that level of granularity for location data is not necessary. The Airport Code variable denotes an airport-based weather station which is the closest one to the location of the accident, and I opted to remove this because the State variable could capture the information. After removing these initial variables, I decided to add variables that could condense existing information.

Creating: I added the Impact Duration variable, calculated by subtracting the original Start Date variable from the End Date in minutes. The new variable shows the length of time the accident impacted traffic. This variable was created only for exploratory measures because this variable will not be helpful for severity prediction since it can only be recorded after the accident occurs.

Transforming: The Weather Condition variable needed the most transformation by consolidating the 128 unique conditions into five categories: rain, snow, low visibility, clear, and cloudy. This was necessary because the levels in the original data were collected by various APIs with different explanations for the same weather condition. An example of this is binning rows of "light drizzle", "rain showers", "heavy drizzle", and "showers" into the "Rain" category. This was done for all five bins and will be the most useful for the exploratory data analysis.

Missing Variables: To prevent loss of information by removing observations with NA's, I imputed the NA values for continuous variables with the mean of the corresponding variable.

For the NA categorical variables, there were only three that had NA counts, Street (1 missing), City (131 missing), and Wind Direction (17,505 missing). Unlike the imputing method for the numeric variables, I removed all observations with the missing categorical variables because they only made up a fraction of the total data.

Grouping: Once the entire dataset has been cleaned by removing, creating, and transforming variables, the next step is to group the data to narrow the scope. While data is provided for 49 states within the US, it would be non-trivial to make any causation claims by means of controlling for variables by state. For this reason, the research will be conducted in the state of California specifically. In order to do this, I filtered for all observations with CA in the State column using the dplyr library from R. This reduced the final data set to 770,977 observations.

3.3 Final Dataset

The final cleaned dataset contained 770,977 observations and 48 variables, with no missing data. As mentioned, these are all observations filtered explicitly to the state of California to narrow the study's scope. In the next chapter, there will be a graphical representation of the number of accidents in each state within the dataset, which will show the states that make up a majority of the data. Additionally, outside of this specific dataset, California has been a state at the forefront of the automotive industry, fostering start-ups and research and development teams for companies like Tesla, Waymo, Lucid, and more. Below is a data codebook with all the final variables, with the variable structure, and a description for reference.

Variable	Structure	Description
Severity	num	Shows the severity of the accident, a number between 1 and 4, where 1 indicates the least impact on traffic (i.e., short delay as a result of the accident) and 4 indicates a significant impact on traffic (i.e., long delay).
Date	Date	The exact date the accident occurred (YYYY-MM-DD).
Time	num	The time the accident was recorded via API.
End_Time	POSIXct	Shows end time of the accident in local time zone. End time here refers to when the impact of accident on traffic flow.
Start_Time	num	Shows start time of the accident in local time zone.
Start_Lat	num	Shows latitude in GPS coordinate of the start point.
End_Lat	num	Shows latitude in GPS coordinate of the end point.
Start_Lng	num	Shows longitude in GPS coordinate of the start point.
End_Lng	num	Shows longitude in GPS coordinate of the end point.
Distance	num	The length of the road extent affected by the accident (in miles).
Description	chr	Shows natural language description of the accident.
Street	chr	Shows the street name in address field.
Side	chr	Shows the relative side of the street (Right/Left) in address field
City	chr	Shows the city in address field.
County	chr	Shows the county in address field.
State	chr	Shows the state in address field.
Zipcode	chr	Shows the zipcode in address field.
Timezone	chr	Shows the timezone in address field.
Weather_Timestamp	POSIXct	Shows the time when the weather data was collected.

Variable	Structure	Description
Temperature	num	Shows the temperature (in Fahrenheit).
Wind_Chill	num	Shows the wind chill (in Fahrenheit).
Humidity	num	Shows the humidity (in percentage).
Pressure	num	Shows the air pressure (in inches).
Visibility	num	Shows visibility (in miles).
Wind_Direction	chr	Shows wind direction.
Wind_Speed	num	Shows wind speed (in miles per hour).
Precipitation	num	Shows precipitation amount in inches, if there is any.
Weather_Condition	chr	Shows the weather condition (rain, snow, thunderstorm, fog, etc.)
Amenity	logi	A POI annotation which indicates presence of amenity in a nearby location.
Bump	logi	A POI annotation which indicates presence of speed bump or hump in a nearby location.
Crossing	logi	A POI annotation which indicates presence of crossing in a nearby location.
Give_Way	logi	A POI annotation which indicates presence of give way in a nearby location.
Junction	logi	A POI annotation which indicates presence of junction in a nearby location.
No_Exit	logi	A POI annotation which indicates presence of no exit in a nearby location.
Railway	logi	A POI annotation which indicates presence of railway in a nearby location.

Variable	Structure	Description
Roundabout	logi	A POI annotation which indicates presence of roundabout in a nearby location.
Station	logi	A POI annotation which indicates presence of station in a nearby location.
Stop	logi	A POI annotation which indicates presence of stop in a nearby location.
Traffic_Calming	logi	A POI annotation which indicates presence of traffic calming in a nearby location.
Traffic_Signal	logi	A POI annotation which indicates presence of traffic signal in a nearby location.
Turning_Loop	logi	A POI annotation which indicates presence of turning loop in a nearby location.
Impact_Duration	num	Created variable by subtracting the End Time from the Start Time to see how long the impact of the accident had on traffic (in minutes).
Year	num	Split Date variable, indicating which Year the accident took place.
Month	num	Split Date variable, indicating which Month the accident took place.
Day	num	Split Date variable, indicating on which Day the accident took place.
Wday	num	Split Date variable, indicating which Day of the Week the accident took place.
Hour	num	Split Time variable, indicating which Hour of the day the accident took place.
Severity_Binary	num	Binary indicator of whether or not an accident was severe (0 = Not Severe, and 1 = Severe).

CHAPTER 4

Exploratory Data Analysis

Due to the many directions this dataset can be explored, it is essential to keep a narrow roadmap of critical areas of discovery for the purposes of my two research questions. Since the data is cleaned and filtered for California, a majority of the exploratory analysis included will be specific to the state of California, as opposed to the whole dataset. Doing this will give more distinct insights and intuition for modeling within the scope of research. First, I decided to take a preliminary look at the distribution of accidents by each state. Then, I began to look at which attributes would give interesting findings that supported the motivation behind the research by exploring where most accidents were taking place. The next sections focused on the time aspect of the data - seeing the distribution of accidents per year and at which times most accidents occur. Finally, I decided a time-series visualization of when most accidents occur would create more areas of investigation before modeling.

4.1 Number of Accidents by State

First, I looked at the distribution of accidents grouped by state using the whole dataset. The distribution gave me an insight into where the majority of data was coming from and justified the decision to focus on the state of California, which contains about a third of the full data (770,977 observations).

Aside from California being the state with the most accidents, Figure 4.1 also reveals that the top 5 states with the most accidents come from California, Florida, Texas, Oregon, and New York. These five states account for about 56 percent of the full dataset, which is not surprising when considering that California, Texas, Florida, and New York are the top four

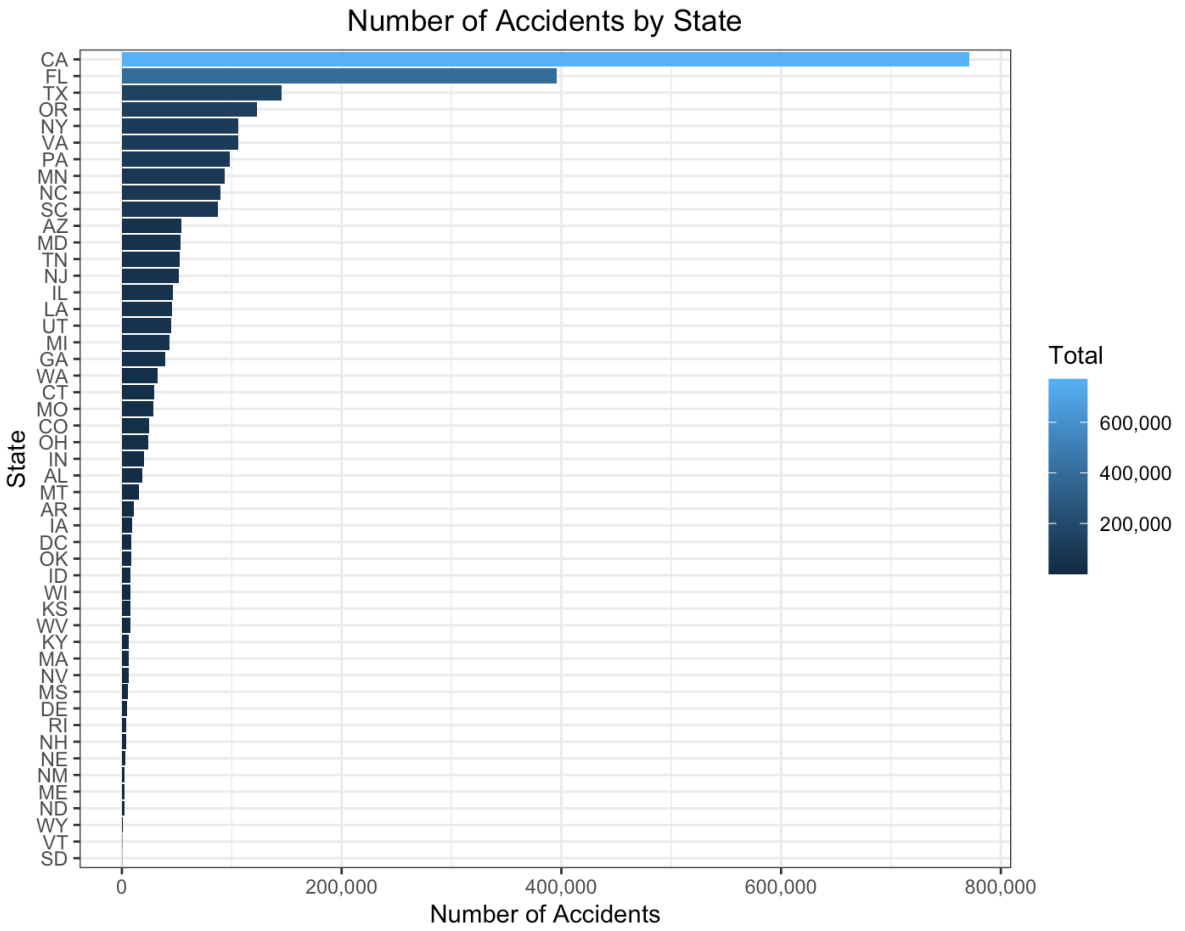

Figure 4.1: Distribution of the Number of Accidents by State in the Full Dataset.

states that make up the entire US population. The state with the least amount of accidents in this dataset turned out to be South Dakota, which currently has a total population that makes up about 0.27 percent of the total US population and ranks 46th out of 50, according to the 2020 census [13]. This information will be used in the context of only exploring and modeling the data specific to California.

4.2 Number of Accidents by Accident Signals in CA

The next area I wanted to explore was where the specific accidents occurred on the road regarding the points of interest. There are twelve accident signals in the cleaned dataset: roundabouts, bumps, traffic calming, no exit, give way, amenity, railway, station, stop signs, crossing, traffic signal, and junctions.

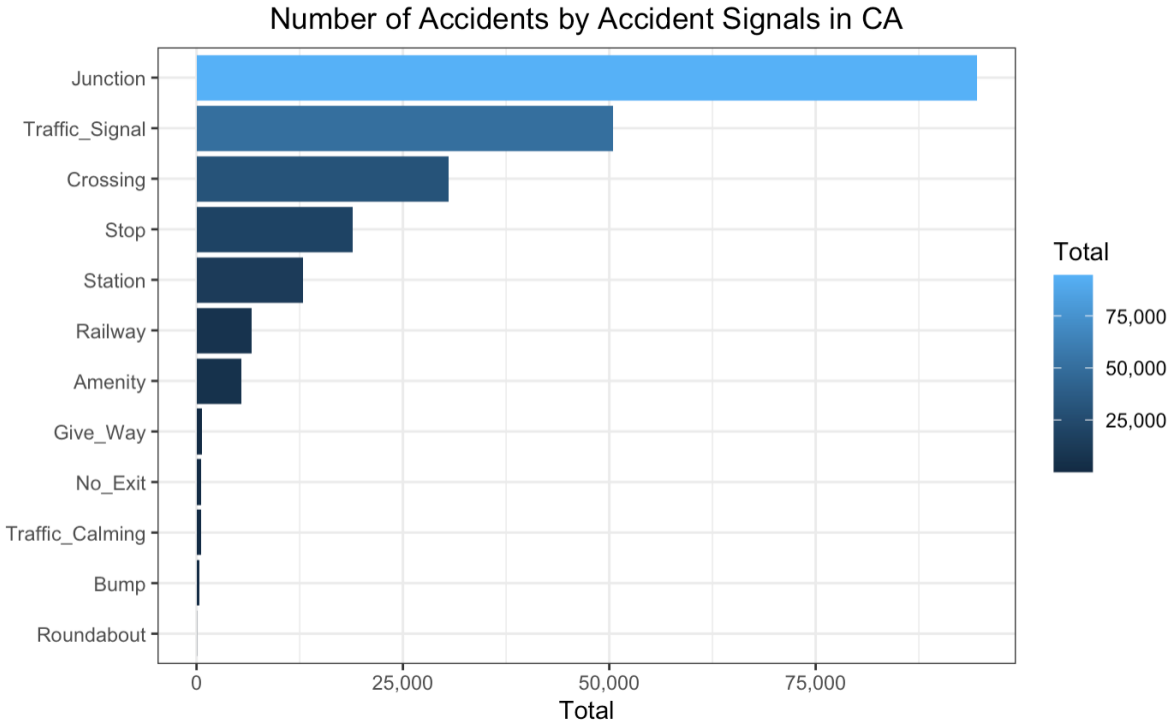

Figure 4.2: Distribution of the Number of Accidents by Accident Signals in the State of CA.

The results from Figure 4.2 show that the overwhelming majority of accidents occurred at road Junctions in California. This differs from what was observed in the whole dataset, as Traffic Signals showed up as having the most accidents. While this observation can not be said to be a direct effect of traffic accidents, it is important to note specific signals that produce the most accidents. Junctions are defined as locations where two or more roads meet. Most commonly, this would be an interchange or an intersection. This information can be dissected further when looking to apply safety features in autonomous vehicles. It is important to note which factors may contribute to making road junctions so dangerous; for example, the physical factors that may obstruct the driver's attention, inattention, poorly

timing a merge, or poor judgment on the speed at which another car is approaching. The following two areas that contribute most to accidents are Traffic Signals and Crossings, which always include a second party or pedestrian. These observations can give better insight into image recognition algorithms used in AVs that need to distinguish objects on the road, i.e., the difference between a biker, a bus, and a poster of a biker on the side of a bus.

4.3 Number of Accidents per Year in CA

Looking at the number of accidents recorded each year over the lifetime of the dataset reveals a note about the data quality. Table 4.1 shows a striking increase in the number of accidents after 2018, which continues to stay constant until another spike in 2021.

Year	Number of Accidents
2016	33,127
2017	34,619
2018	33,919
2019	103,928
2020	188,601
2021	376,783

Table 4.1: Count of the Number of Accidents per Year in the State of CA.

The table indicates there must have been a change in how the data was collected after 2018. When further researching how the data was collected, there were no conclusive results on why this increase came to be. What may be even more compelling about this information is that there seemed to be a large spike in the number of accidents during and after the peak times that COVID-19 affected the state of CA. Intuition would lead me to hypothesize an observed decrease in the number of accidents during these final two years due to the decreasing number of cars on the road, commuters, and travelers. This observation leads to another possible application of the data into further research on the APIs used by MapQuest

and Bing Traffic, how they collect data, or even how they define an accident. In terms of using this information for time-series modeling, it will introduce an extra element of variance in the data.

4.4 Number of Accidents per Hour in CA

Going along with the previously explored accidents per year, is the accidents per hour. Before plotting the data, I was expecting to see a bimodal histogram, peaking at the two rush hours of 8 am and 5 pm, given the nature of commuting times and personal experience with traffic.

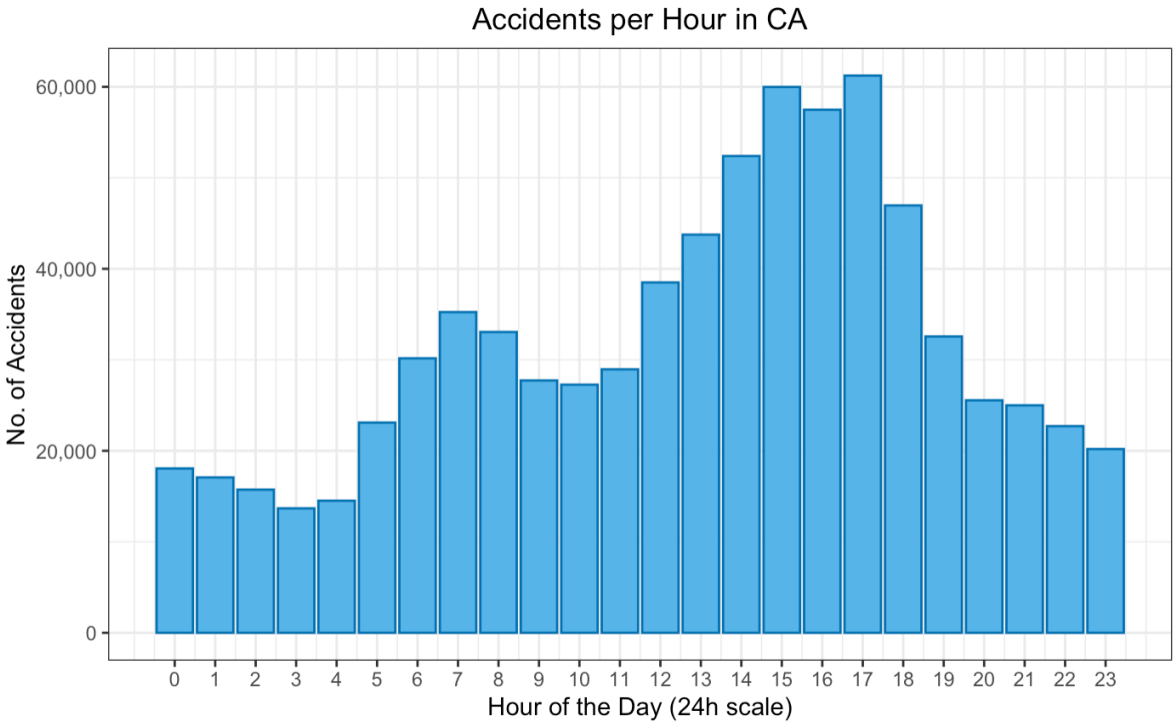

Figure 4.3: Distribution of the Aggregate Number of Accidents per Hour in the State of CA.

Figure 4.3 follows that intuition with a caveat: there is a minor spike in accidents around midnight. Most accidents, in aggregate, seem to occur between 3 pm to 5 pm, which is typically the end of the workday for most Americans. There is also a spike in the number of accidents around 6 am to 8 am, when most people would be commuting to work. One factor that could lead to increased accidents towards the end of the day is fatigue. While there may be an influx of vehicles on the road during the morning and afternoon, the major difference

between these hours is the human factor. Another interesting point this figure reveals is the spike in accidents seen at midnight. This observation may not seem like a significant difference; however, to read the graph correctly, we should take into account the number of cars on the road at these hours and then have a percentage of accidents in relation. It would also be interesting to see the data on how fatal the accidents were at these times. It is reported that while the most common times for accidents may be during peak commuter times, fatal accidents occur between 8 pm to midnight [14]. While the dataset contains a "Severity" variable, it does not indicate the severity level regarding safety or injury.

4.5 Time-Series Visualization

The final exploratory figure is a time-series visualization of the number of accidents that occur each month and the aggregate over each day of the week.

Figure 4.4 shows the two stacked trends of accidents by month and by days of the week. The line chart shows an increase in accidents towards the later months of October through December, met with a steep decline in January. These trends are aggregated throughout the whole dataset, not controlling for a particular year. Generally, this visualization would follow the intuition of accidents occurring more towards the holidays, when traveling increases, i.e., Thanksgiving, Christmas, and New Year. Additionally, if this data were representative of the United States as a whole, it may lead to further investigation of the relationship with weather conditions during these times; however, because this is only representative of data in California, the "clear" and "cloudy" weather conditions make up almost 90 percent of the data, and have insufficient evidence to make any causal claims.

The bottom plot is an extension of the top and provides more details regarding the number of accidents by the specific day of the week. The overall trend follows the above monthly plot; however, it is interesting to note the specific spikes seen on recurring days of the week, such as Thursdays. The data shows that Thursdays in December are when most accidents occur. Generally, most accidents happen on Fridays, and the weekends see a consistent decrease in the number of accidents. This differs from the data reported by the

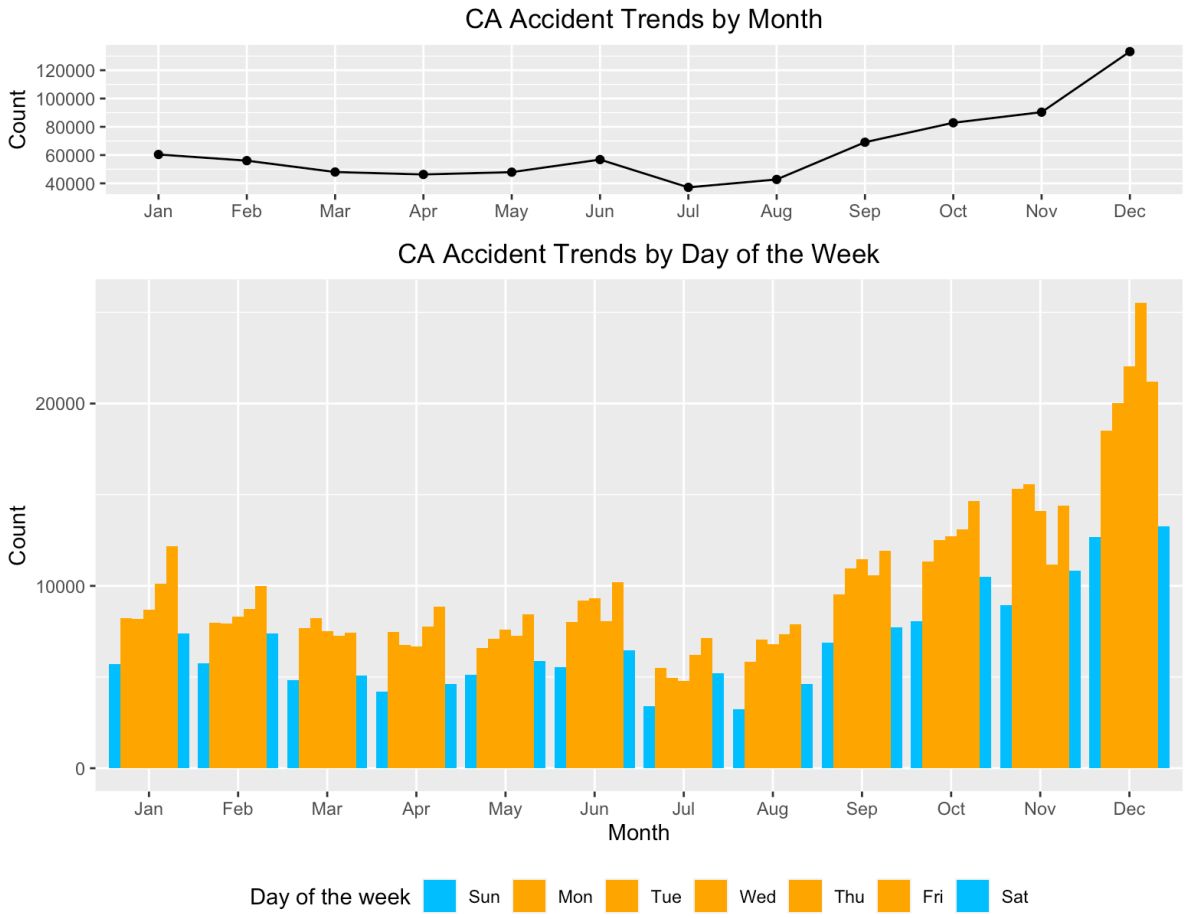

Figure 4.4: Time-Series Visualization of Accident Trends in the State of CA by Month and by Days of the Week.

National Highway Traffic Safety Administration data, which reports that the most dangerous day of the week to drive is Saturdays [15]. Even though this data may not follow the national reports, it is still instrumental in the time-series analysis. One exciting application of this information is it follows poisson data over the count of accidents. While that is not a primary focus of this paper, it is an interesting application that can be explored for future work.

CHAPTER 5

The Models

5.1 Logistic Regression for Severity Prediction

In order to predict the severity of accidents in our data, the definition of what is "severe" must be clear. The Severity variable is indicative of the impact an accident had on traffic, represented by an integer between one and four. One indicates the least impact on traffic, and four indicates a long delay in traffic. The first step in building the model is to evaluate the response variable. When tabling the four severity levels, it becomes apparent that the levels are unbalanced, as revealed by Table 5.1.

Severity (level)	Number of Accidents
1	4,926
2	737,434
3	19,764
4	8,853

Table 5.1: Count of the Number of Accidents per Severity Level in the State of CA.

To remedy the unbalanced levels, a solution would be to bucket the values as "not severe" and "severe". In order to do this, a new variable *Severity_Binary* is created by labeling all severity = 1 or 2 as a 0, indicating "not severe"; and labeling all severity = 3 or 4, indicating "severe". When this step was done, the new *Severity_Binary* variable was split into 742,360 "not severe" labels and 28,617 "severe" labels. While this is indeed more balanced than the four split levels, it is still not optimally balanced. The next step in building the model is splitting the data into a training set and a testing set.

Train-Test-Split: There is a 70/30 split of the data, creating a training dataset of 539,683 observations and a testing dataset with 231,294 observations. While there are many different splits the data can take on, for example, an 80/20 split, the reason it was split by 70/30 was to allow for enough testing observations given the large amount of data. Additionally, upon the first iteration of modeling on 80 percent of the training data, the results were unable to be produced because of a lack of computing power. When the split was done more modestly, it required less computation power to generate the model.

Resampling: Resampling is a method used in order to remedy imbalanced data, in this case, the unbalanced response variable *Severity_Binary*. Generally, an imbalance in training data leads to bias that can influence the model due to the skewness of the majority vs minority class. Randomly resampling the training data can be done by under-sampling, deleting examples from the majority class; or by oversampling, duplicating examples from the minority class [16]. For this case, resampling involves a new transformed version of the training data in which there is a combination of over- and under-sampling using the function *ovum.sample()* in the R library ROSE [17] specifying the method argument as "both". After applying this function, the response variable on the new training data is 270,142 "not severe" and 269,541 "severe" cases.

Logistic Regression: Once the data has been properly cleaned, transformed, and split, it is time to build the logistic model. Because the response variable is binary, the baseline model for severity prediction is logistic regression. Logistic regression is a predictive analysis appropriate to conduct when the dependent variable is binary. It uses nominal, ordinal, or ratio level independent variables to explain the relationship between one dependent binary variable [18]. The function used to create the baseline model in R is the *glm()* function. First, a full model was created to include all relevant variables in the resampled training dataset. Then, the full model is taken in by the *step()* function, to select a formula-based model by AIC in a stepwise algorithm [19]. The *step()* function has the ability to perform forward or backward stepwise regression, as well as both. Forward stepwise begins with a null model, then starts adding the most significant variables one after the other until all variables under consideration are included in the model. Conversely, backward stepwise begins with the

full mode, the starts removing the least significant variables until all statistically significant variables are included in the model [20]. Because the model being used for stepwise regression is the full model, backward stepwise is the preferred method for variable selection. After applying the step function on the model, the Akaike Information Criterion (AIC) selection criterion recommended to remove *Start_Lng* and *Precipitation* variables from the model. The fitted logistic regression model can by explained by the following equation:

$$\hat{\theta}(\mathbf{x}) = \frac{1}{1 + exp(-(\hat{\beta}'\mathbf{x}))}$$

Where $\hat{\theta}(\mathbf{x})$ denotes the binary response variable and $\beta' = (\beta_1, \beta_2, ..., \beta_{15})'$ predictor variables.

Table 5.2 shows the R output of the final logistic regression model. The estimates of the predictor variables are representative of the log of odds for an accident being severe or not severe. However, the log of odds is not easy to interpret, which is why we exponentiate it to get the odds ratio. The results on Table 5.2 are helpful in noting which variables are statistically significant in relation to the severity of an accident. Variables with at least one asterisk indicate there is evidence of a real associate between the predictor and response variable, at a p-value of less than 0.05. For example, the negative coefficient from the *Temperature* variable indicates with all else held constant, one unit increase in the temperature, means it is less likely for the accident to be highly severe.

Interpretations are best made from the odds-ratios, seen in Figure 5.3. Generally, the way to interpret odds-ratios are that if the estimate is less than 1, then the odds are decreased for an outcome; and if the odds are greater than one, then the odds are increased. In contrast to the results seen from Table 5.2, if the odds-ratio is 1, it means there is no association between the predictor and the response variable, meaning the results are not statistically significant. Figure 5.3 shows that *Temperature*, *Day*, and *Hour* are not statistically signifiant to predicting severity.

For variables with different levels, such as *Weather Condition*, the results are all in comparison to a reference group, in this case, clear weather. Interpretations of significant

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	1842.4663	5.8260	316.25	0.0000***
Start_Lat	-0.0066	0.0017	-3.83	0.0001***
SideR	0.0153	0.0110	1.39	0.1639
Temperature	-0.0038	0.0006	-6.77	0.0000***
Wind_Chill	0.0257	0.0006	46.41	0.0000***
Humidity	0.0088	0.0002	41.64	0.0000***
Pressure	-0.0131	0.0050	-2.59	0.0095**
Wind_Speed	0.0255	0.0007	34.92	0.0000***
Year	-0.9127	0.0029	-318.18	0.0000***
Month	-0.1293	0.0010	-128.39	0.0000***
Day	0.0007	0.0004	1.76	0.0785.
Wday	-0.0299	0.0018	-16.25	0.0000***
Hour	-0.0032	0.0006	-5.28	0.0000***
Traffic_SignalTRUE	-0.0538	0.0141	-3.81	0.0001***
JunctionTRUE	0.2853	0.0093	30.69	0.0000***
Weather_Conditioncloudy	0.0119	0.0080	1.49	0.1369
Weather_Conditionlow_visibility	-0.0409	0.0160	-2.56	0.0104*
Weather_Conditionrain	-0.1188	0.0167	-7.10	0.0000***
Weather_Conditionsnow	0.5839	0.0781	7.48	0.0000***
Weather_Conditionwindy	0.2995	0.1369	2.19	0.0286**

Table 5.2: Final Logistic Model Output for Binary Severity Prediction

predictors from Figure 5.3 include:

- All else held constant, if an accident occurs at a Traffic Signal, there is a 5% decrease in the odds of it being severe.
- All else held constant, if an accident occurs at a Junction, there is a 33% increase in the odds of the it being severe.

· All else held constant, there is a 79% and 35% increase in the odds of a severe accident if it is respectively snowing or windy outside, as opposed to the weather being clear. Additionally, there is a 4% and 11% decrease in the odds of a highly severe accident if the conditions show low visibility or rain.

	Estimate	2.5 %	97.5 %
(Intercept)	Inf	Inf	Inf
Start_Lat	0.99	0.99	1.00
SideR	1.02	0.99	1.04
Temperature	1.00	1.00	1.00
Wind_Chill	1.03	1.02	1.03
Humidity	1.01	1.01	1.01
Pressure	0.99	0.98	1.00
Wind_Speed	1.03	1.02	1.03
Year	0.40	0.40	0.40
Month	0.88	0.88	0.88
Day	1.00	1.00	1.00
Wday	0.97	0.97	0.97
Hour	1.00	1.00	1.00
Traffic_SignalTRUE	0.95	0.92	0.97
JunctionTRUE	1.33	1.31	1.35
Weather_Conditioncloudy	1.01	1.00	1.03
Weather_Conditionlow_visibility	0.96	0.93	0.99
Weather_Conditionrain	0.89	0.86	0.92
Weather_Conditionsnow	1.79	1.54	2.09
Weather_Conditionwindy	1.35	1.03	1.76

Table 5.3: Odds Ratio Estimates and Confidence Intervals of Logistic Regression Model.

5.1.1 Results

Once the final model has been established on the training data, it is used to make predictions and evaluate the testing data. The *predict()* function in R can be used to predict the probability of an accident being categorized as not severe, or severe, given the predictors. After storing the predicted probabilities, there is a selection criteria set for what would be categorized as Severe vs. Not Severe. In this case, the selection criteria is a prediction over 50% will be categorized as 1=Severe, and less than 50% will be categorized as 0=Not Severe. The results on the testing data are then evaluated using a confusion matrix.

A **Confusion Matrix** is a performance metric for a machine learning problem where output can be two or more classes. Labels on the horizontal represent the actual values, and labels on the vertical axis represent the predicted values. A perfect confusion matrix will have zeros on the off-diagonals, indicating that the predicted values are equal to the actual values, however, this would likely also lead to a model that is overfitting [21]. The 1 values in Table 5.4 represent a Severe response, and the 0 values represent a Not Severe response. The top left quadrant represents the True Positive values, meaning accidents that were predicted to be severe, were indeed. The bottom right quadrant represents the True Negative values, showcasing all accidents that were not categorized as severe. The off-diagonals show the areas of improvement of the model. The top right quadrant indicates a False Positive: the model falsely predicted the accident to be severe. Finally, the bottom left quadrant indicates a False Negative: the model incorrectly predicted an accident to not be severe, when it actually was.

Table 5.4: Confusion Matrix of Actual vs. Predicted Severity.

	1	0
1	6460	39362
0	2112	183360

The confusion matrix in Table 5.4 is useful for measuring other key performance metrics of the model such as Precision, Recall, Accuracy, and ROC curves. The focus is mainly on

model Accuracy, which reveals the general percentage accurately predicted for both severe and not severe accidents.

$Accuracy = \frac{6460+183360}{Total} = 82\%$ of all classes (Severe and Not Severe) have been predicted correctly.

Finally, the **Receiver Operating Characteristic (ROC)** curve is used to diagnose the ability of the model to accurately classify the response. The ROC curve places the false positive rate vs the true positive rate, meaning that the closer the curve is to the top left corner indicates better performance; whereas a curve that is closer to the dashed 45-degree diagonal indicates less accuracy (False Positive = True Positive). Figure 5.1 shows that the final model generally does a good job of predictive accuracy.

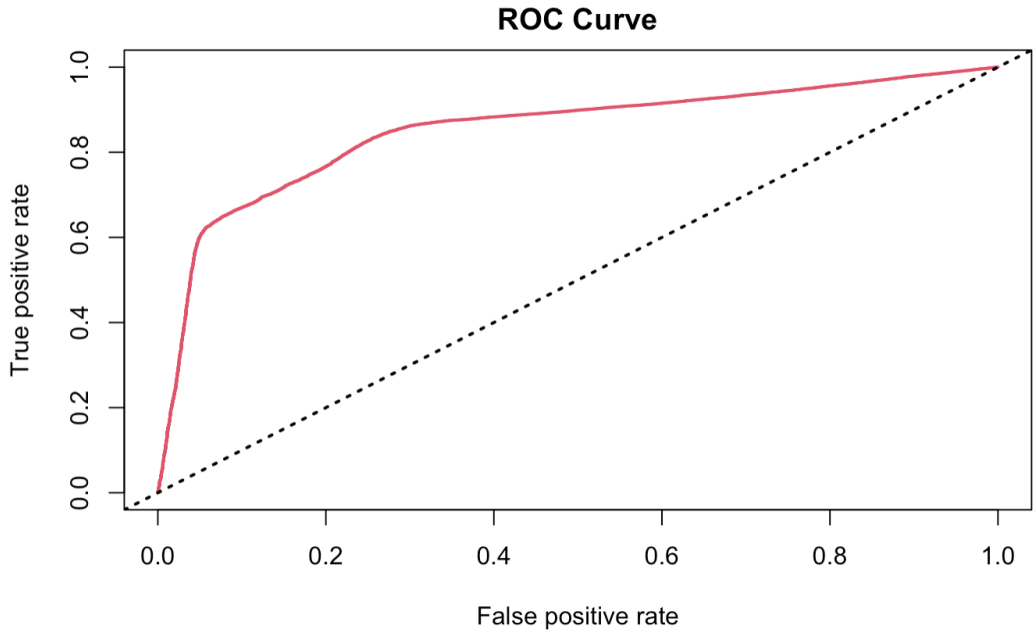

Figure 5.1: Receiver Operating Characteristic (ROC) Curve for the Final Logistic Regression Model.

The main takeaways from the final model of severity prediction is that the data can be used for risk severity prediction with a high accuracy. The variables that play the largest role in the statistical significance of predicting the severity of an accident, according to this model, are whether or not the the accident occurred at a traffic signal, junction, or is dependent on the weather condition.

5.2 ARMA Model for Time-Series

The final area of research is a time-series analysis of the number of accidents. Similar to the steps taken to build the severity prediction model, it is necessary to prepare the data. The data frame I begin with is the previously created clean data filtered for California. However, because the time-series model will only be looking at the number of accidents, the data must be pivoted such that there are two columns: a Date column that contains one observation for each day of the year and a column representing the total number of accidents corresponding to the specific date. For the next step, I grouped the data by Date and summarized it by the, *n()*, number of accidents. This created a data frame of 2005 observations and two variables. Once the data have been pivoted, the final step is to convert it to a time series. To do this, I used the *xts_to_ts()* function in R and set the frequency argument to 365 to work with daily counts. After this final step had been completed, I checked if the data was ready to be used for analysis by running the *is.ts()* function, which tests if an object is a time series, in this case, the output was true, and the data was ready for use.

The two models for time-series forecasting studied for the research question are Auto Regressive Moving Average (ARMA) and Auto-Regressive Integrated Moving Average (ARIMA). The difference between the two is the integrated distinction, which measures how many non-seasonal differences are needed to achieve stationarity [22]. The data is stationary in a time series when it does not depend on the time at which the series is observed, meaning there is not a strong trend or seasonality observed. To fit a stationary model, such as an ARMA, the first step is to detect the stationarity of the data and remove any trend or seasonality effect. At first glance, Figure 5.2 shows a clear upward trend in the number of accidents as time goes on. This was expected after the exploratory analysis was done in Table 4.1, showing a rapid increase in the number of accidents between 2018 and 2019, then again from 2020 to 2021. While the data documentation does not comment on this, we can assume it stems from a change in how the APIs collected data. While there is a noticeable trend, there is no clear seasonality seen in the data at first glance.

Since there is a trend in the data, removing the trend before modeling is necessary. One

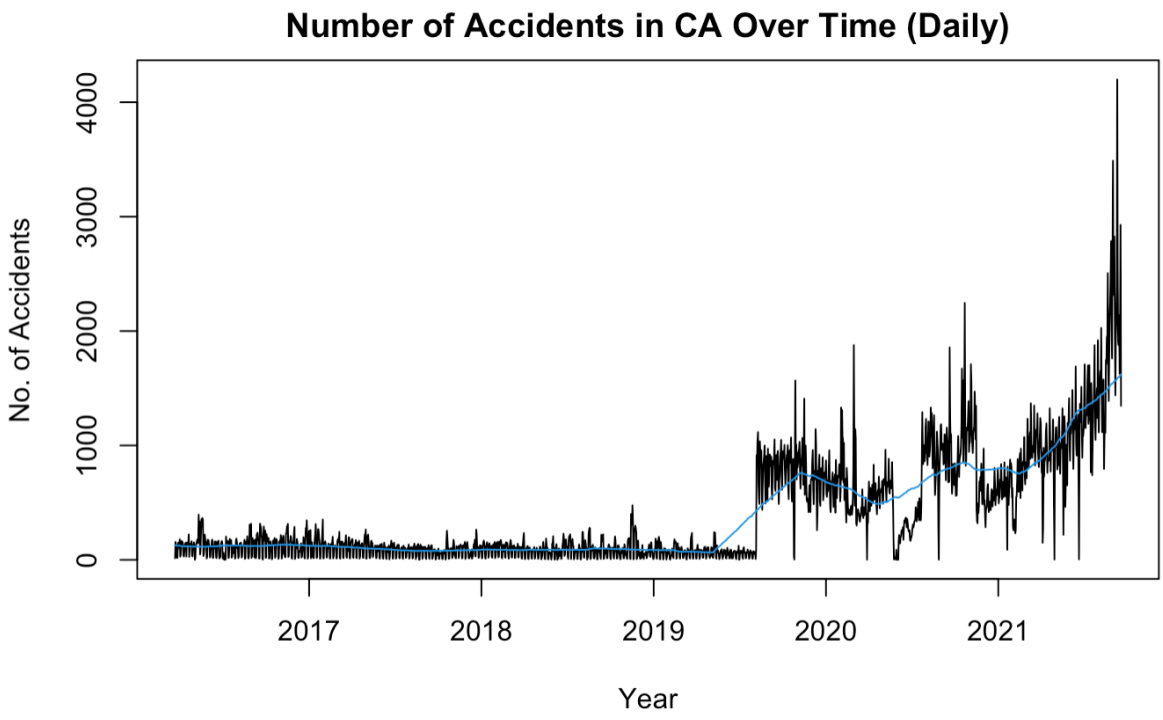

Figure 5.2: Time-Series Representation of Accidents in CA from 2016 to 2021.

method of seeing the number of differences that need to be taken on the data is looking at the **Auto Correlation Function**, or ACF. ACF shows the correlation between observations at the current time and previous time spots, otherwise denoted as the Lag. The purpose of the ACF plot is to reveal if the time series is random, identify seasonality, and uncover hidden patterns in the data. In addition to looking at the ACF plots, we can do a more formal test for autocorrelation using the **Box-Pierce test**. The Box-Pierce test statistic examines the null hypothesis of independence in a given time series. A p-value less than 0.05 means that we reject the null hypothesis and indicate that the time series contains autocorrelation. The function to execute this in R is the *Box.test()* function, which, when run on the time series object, outputs a p-value $< 2.2e-16$, which is well under 0.05, justifying the results seen in the ACF plots.

Figure 5.3 shows the ACF plots on zero to third-order differences in the data. The raw data shows a slow and steady decline towards zero above the threshold, indicated by the horizontal blue line. The First Differences plot removes a lot of the autocorrelations from

what is seen in the raw data; however, the second and third-order plots do not show much change after. The issue with taking the first-order difference to remove the linear trend is the significant loss of information, and the interpretation becomes near impossible due to the data transformation.

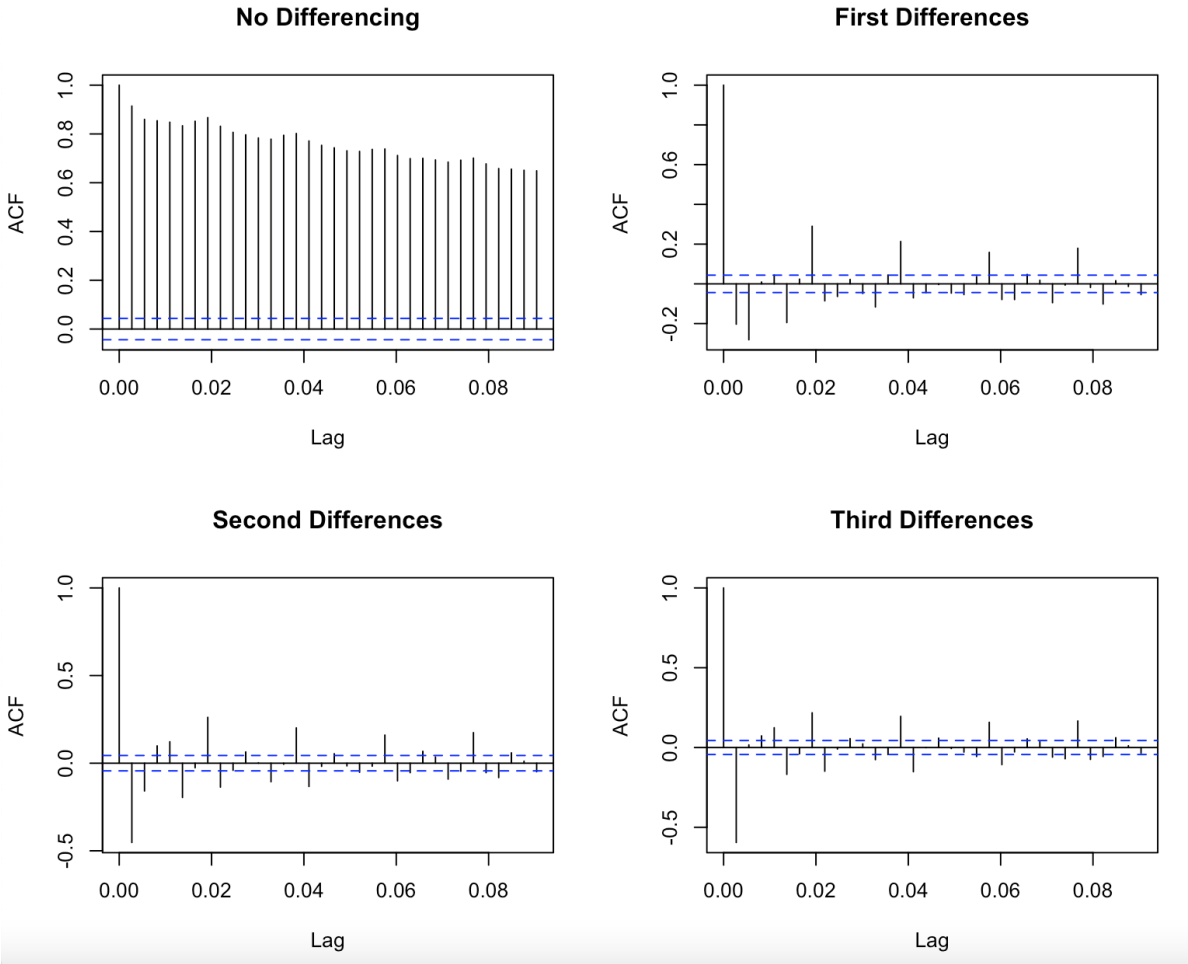

Figure 5.3: Auto Correlation Function (ACF) Plot of full Time-Series data in CA.

An alternative method to remove the trend is to model the trend. This is done by taking the vector of observations, known as our original time-series object, and subtracting the fitted trend vector, fitted by kernel smoothing. This results in the residuals. The next step is to treat the residuals like they are the data and model on the residuals.

Figure 5.4 shows three plots related to the residual's time-series object. The three graphics included are the time plot (top), the ACF plot (bottom left), and a histogram of the

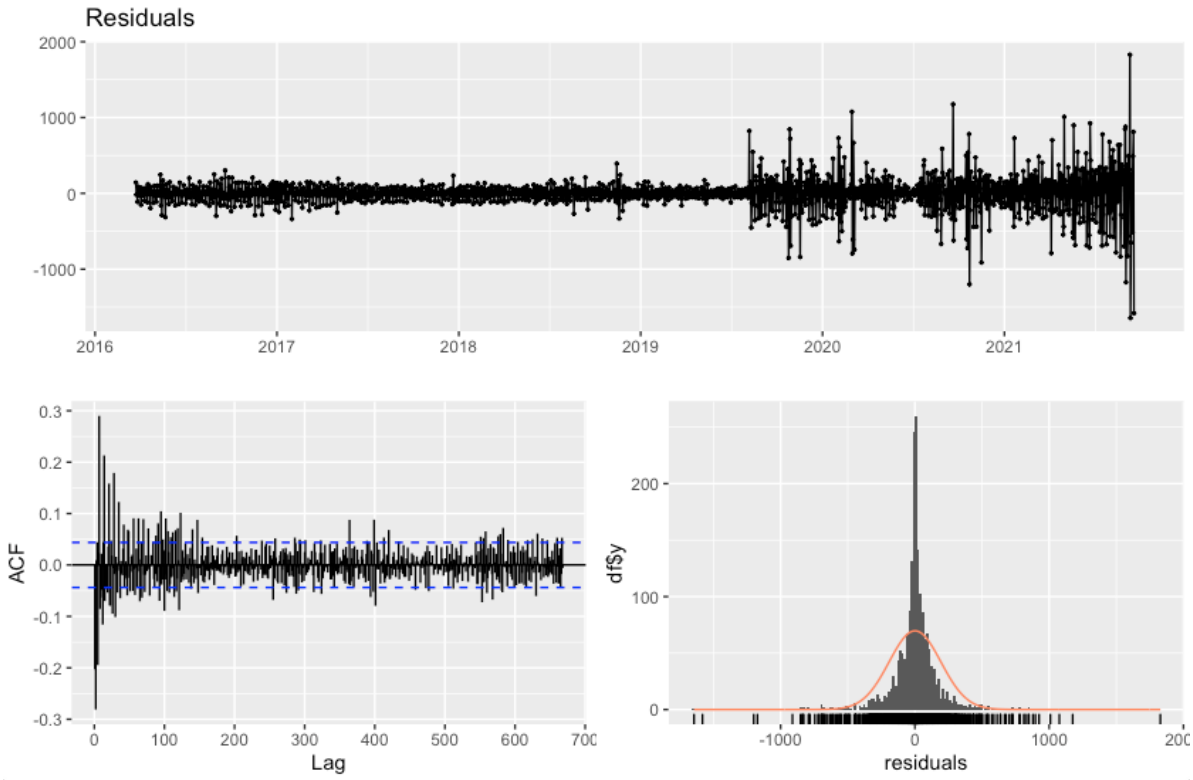

Figure 5.4: Residuals Diagnostic Plots: Time-Series, ACF, and Distribution.

residuals with an overlaid normal distribution for comparison (bottom right). The main areas of focus from this output are the bottom two plots. Ideally, we are looking for no autocorrelation and normally distributed residuals. The ACF plot shows that we still see some autocorrelation; however, the ACF of the series is decaying to zero much quicker than in the original series. The residuals show that they are normally distributed, which is ideal moving forward. In order to do the modeling, there are two methods primarily used. The first is a manual model, which uses information gathered from the diagnostic plots, and the second is an automatic model. For this paper, the most efficient option is the automatic model. To do this, I use the *auto.arima()* function in R, that comes from the *forecast* package [23]. The *auto.arima()* function returns the best ARIMA model according to the lowest AIC value. The function searches for the best possible models within the order constraints provided.

5.2.1 Results

The final model produced by the *auto.arima()* function was an ARMA model of the order:

$$ARMA(5, 0, 5)$$

This output can be broken into two parts, the Auto-Regressive model and the Moving Average model. As previously stated, the ARMA comprises the two separate AR and MA models. It is possible to build a model of solely one or the other; however, in this case, the final model works together to produce ARMA. The order of the *auto.arima()* function takes in three parameters. First is the AR parameter, which tells how much yesterday's value influences today. Second is the Integrated term, described as the order of differences. Lastly is the MA parameter, which accounts for some other unobserved noise and how that affects the data. The AR and MA values are chosen by minimizing the AIC after differencing the data d times. In this case, *AR(5)* means the current value is based on the previous five values. Additionally, *MA(5)* means the five lagged values have a significant direct effect on the present-day number of accidents. Finally, the zero middle term indicates that the data is stationary.

The final model is visualized in Figure 5.5 zoomed in from a window of time that starts in 2019, and forecasts the number of accidents until 2023. The window of time that begins in 2019 was to visually eliminate data from 2016-2018 when there were fewer accidents. While this model can be quickly optimized simply by having uniform data, it does a pretty good job of showing a prediction of the number of accidents. The goal of this paper was to perform a time-series analysis on the data, which has yet to be done, and this result is a great starting place and baseline for research moving forward.

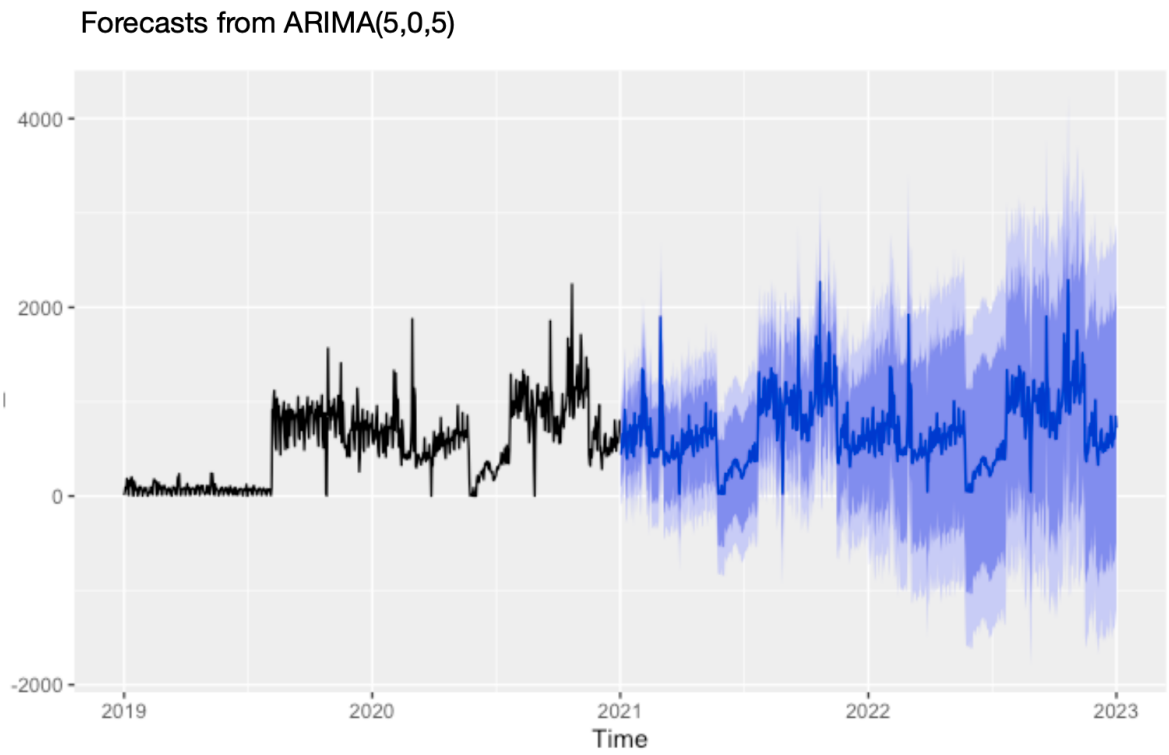

Figure 5.5: Time-Series Forecast (window from 2019 to 2023)

CHAPTER 6

Conclusion

This paper attempted to explain and predict the severity of an accident by using a logistic regression model with a binary indication of severity as the response variable, and conduct a time-series analysis that forecasts the number of accidents based on historical data. In addition to the trained severity prediction model that performed with 82% accuracy on the testing data, the model revealed three key features that significantly affected severity. Accidents at a traffic signal were 5% less likely to have a severe impact on traffic, while accidents taking place at junctions saw a 33% increase in the odds of it being severe. Regarding weather conditions, in California, the model reveals a 79% increase in the odds of a severe accident if it is snowing outside and a 35% increase if it is windy. Both of these odds are in relation to the baseline of clear weather. These results follow a general intuition of what would affect traffic; however, it is always worth noting when the data gives evidence to a general hypothesis.

The second objective of the paper focused on a time-series analysis of the number of accidents. Time-series analysis has yet to be explored on the specific dataset and gives new insights into various data applications. The greatest challenge in producing the time-series analysis and model was the inconsistency seen in the data. The unexplained variations in the number of recorded accidents per year introduce extra noise in the data that may not be necessary if the data was uniform. Since there was no specific remedy for this, modeling continued by taking the necessary precautions and evaluating the assumptions. The key to modeling the data was to eliminate the trend by fitting the trend and being left with the residuals. Once the output graphics of the residuals showed that the data was ready to be modeled, the best model selected was an ARMA(5,0,5), which produced the lowest AIC from

various combinations. This model did a fine job forecasting the number of accidents for a set time frame; however, it could continually be optimized. The results seen are helpful to get the ball rolling for future work using this dataset.

CHAPTER 7

Limitations and Future Work

This paper explored a large dataset with close to three million observations spanning almost six-year with forty-seven measurements that pertained to accident data across the United States. The greatest strength of this dataset is the many applications it can be used for. This paper focused explicitly on severity prediction regarding how much impact an accident has on traffic, and on a time-series analysis of the number of accidents. The most considerable weakness of the data came from the inconsistent APIs and sources for data collection. There is a noticeable and unexplained jump in the number of accidents recorded between 2018 to 2019 and 2020 to 2021. This discrepancy came up most in the time-series analysis and modeling of unexplained variation in the data. With this weakness aside, the two research questions were explored and executed using statistical methods and techniques related to logistic regression and autoregressive moving average prediction models.

The other three main identified limitations in addition to the data collection were the imbalanced data for the POIs, a lack of more descriptive variables, and the availability on the type of vehicles for which the data was obtained. First, the exploratory data analysis showed that several POI variables were very unbalanced, which could introduce bias to the final model if not dealt with properly. In addition, an analysis of "severity" in this specific study means the severity of impact on traffic. Suppose there was an additional variable that shows the severity of the accident regarding safety, such as a measurement of fatality or measurement of injury from the accident. This information could be used to do further research on risk severity prediction for the safety of road vehicles rather than just the impact on traffic. Finally, the last limitation of this dataset is that it does not come from AVs. The motivation behind this study was to use the insights found as a basis for future research

on EVs and AVs for the means of more efficient and safe transportation. However, it is understandable that this is not available because most companies in the space are currently still developing the Level 5 autonomous technology, so the data is harder to obtain via open source.

Due to the scope and potential, this data has many future applications. For example, future work can be done by getting Census data to control for various population statistics. Similarly, the data can be narrowed down for each County in a state or widened to seek a broad generalization of all accidents within the United States. A potential area that can be explored is to create a dashboard that allows any user to customize the scope for general research. Regarding time-series analysis, the models built can be further developed to model real-time accident prediction in combination with a real-time impact of an accident on traffic. Finally, another exciting area that can be explored is policy implications concerning the future of EVs and AVs. One of the pain points for EV/AV adaptation is the unknown about liability and regulations.

This topic can not be ignored as it is the future of transportation. The role of autonomy is increasing in society in almost every industry, and it will be the future of how communities run. Future research should be done on how autonomous vehicles and infrastructure can prevent overall traffic and create a continuous flow. Eventually, a perfect traffic flow would mean that no one would ever have to drive a car themselves or even own one. There are ideas that allude to various companies owning all cars, and passengers would call their ride on an app and input their destination. Regardless of the rate at which the future is evolving; research like this is essential to ensure the highest possible safety and to use prediction models to prevent randomness.

REFERENCES

[1] Horowitz Belsky, Weinberg. How many car accidents does the average person have? https://www.belsky-weinberg-horowitz.com/how-many-car-accidents-does-the-average-person-have/, year = 2022,.

[2] U.S. Department of Transportation. State by state crash data and economic cost index. https://www.transportation.gov/research-and-technology/state-state-crash-data-and-economic-cost-index, year = 2020.

[3] Miller Kory Rowe. Motor vehicle crashes cost the u.s. nearly 1 trillion/year. https://www.mkrfirm.com/blog/motorcycle-accidents/motor-vehicle-crashes-cost-u-s-nearly-1-trillionyear/, year = 2021,.

[4] Department of Energy. The history of the electric car. www.energy.gov/articles/history-electric-car: :text=Here in the U.S., the,spark interest in electric vehicles, year = 2014,.

[5] Feilding Cage. The long road to electric cars in the us. https://graphics.reuters.com/AUTOS-ELECTRIC/USA/mopanyqxwva/, year = 2022.

[6] Wired Brand Lab. A brief history of autonomous vehicle technology. https://www.wired.com/brandlab/2016/03/a-brief-history-of-autonomous-vehicle-technology/,.

[7] The 6 levels of vehicle autonomy explained. www.synopsys.com/automotive/autonomous-driving-levels.html, year = 2022.

[8] Investigation of automated vehicle effects on driver's behavior and traffic performance. *Transportation Research Procedia*, 15:761–770, 2016. International Symposium on Enhancing Highway Performance (ISEHP), June 14-16, 2016, Berlin.

[9] Mapquest. https://www.mapquest.com/.

[10] Microsoft bing. https://docs.microsoft.com/en-us/bingmaps/rest-services/traffic/, title = Traffic API.

[11] Sobhan Moosavi, Mohammad Hossein Samavatian, Srinivasan Parthasarathy, and Rajiv Ramnath. A countrywide traffic accident dataset. *CoRR*, abs/1906.05409, 2019.

[12] Zeeshan-ul-hassan Usmani. What is kaggle, why i participate, what is the impact? https://www.kaggle.com/getting-started/44916.

[13] State population by rank. https://www.infoplease.com/us/states/state-population-by-rank.

[14] Moses. When and where most car accidents occur. https://www.moseslawsc.com/blog/2021/july/when-and-where-most-car-accidents-occur, year = 2021.

[15] NHTSA. Nhtsa releases 2020 traffic crash data. https://www.nhtsa.gov/press-releases/2020-traffic-crash-data-fatalities, year = 2022.

[16] Jason Brownlee. Random oversampling and undersampling for imbalanced classification. https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification, year = 2020,.

[17] ovun.sample: Over-sampling, under-sampling, combination of over- and under-sampling. https://www.rdocumentation.org/packages/ROSE/versions/0.0-4/topics/ovun.sample,.

[18] Anshul Saini. Conceptual understanding of logistic regression for data science beginners. https://www.analyticsvidhya.com/blog/2021/08/conceptual-understanding-of-logistic-regression-for-data-science-beginners/, year = 2021.

[19] Step: Choose a model by aic in a stepwise algorithm. https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/step.

[20] Understand forward and backward stepwise regression. https://quantifyinghealth.com/stepwise-selection/.

[21] Sarang Narkhede. Understanding confusion matrix. https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62, year = 2018.

[22] Arma model. https://www.statisticshowto.com/arma-model.

[23] Auto.arima: Fit best arima model to univariate time series. https://www.rdocumentation.org/packages/forecast/versions/8.16/topics/auto.arima.