HAMM-LIPPSTADT
UNIVERSITY OF APPLIED SCIENCES

Understanding the Customer Perception of Autonomous Vehicles Using Sentiment Analysis

Lippstadt Campus

Project Work
Bachelor of Engineering

submitted by
Pritish Sanjay Samant

Electronic Engineering
Mat.Nr.:2190651
<pritish-sanjay.samant@stud.hshl.de>

February 19, 2024

First supervisor:	Prof. Dr. Achim Rettberg
Second supervisor:	Charles Steinmetz

Contents

1 Abstract

In the context of self-driving cars, this research intends to assess the effectiveness of a random predictive algorithm for sentiment analysis. The distribution of feelings is examined using a Counter class on a dataset of tweets relating to self-driving cars. Random sentiments are created according to the observed distribution over a number of iterations, and accuracy is calculated by contrasting the anticipated emotions with the actual feelings. For the purpose of calculating the average accuracy, the procedure is done many times. The findings demonstrate how well the random model for forecasting captures sentiment trends and highlight its potential use in tasks involving sentiment analysis for self-driving car.

2 Introduction

People's thoughts on autonomous cars would be interesting to know in light of the COVID-19 pandemic and other issues affecting economies around the world. This paper uses a social media dataset, specifically the Twitter dataset, to do sentiment analysis to find out how people around the world feel about self-driving cars, especially in the event of an accident, and how they feel about accepting technology that makes safety easier. Companies can use this kind of analysis to find out how customers feel about autonomous vehicles and then make rules for them and market them in a way that fits those rules. A lot of progress has been made in research, which is pushing the government to make decisions about policies and plans to grow the market for self-driving cars. However, there aren't many reviews from customers [NHM+20]. Self-driving cars are becoming very popular very quickly, and new inventions are being made all the time. As a result, some inventions work well and prevent accidents, while others fail and make people dislike self-driving cars [GS22]. Using different machine-learning techniques, this paper will look at how customers feel about self-driving cars. This will help companies and researchers figure out what needs to be fixed and how to make self-driving cars. We may not need self-driving cars right now, but the fact that fossil fuels are running out and demand for other energy sources is growing is a big problem. That way, this kind of research will get the money it needs, and technology like this for self-driving cars will indeed be very important in the future if we want to fight climate change and the rising cost of fuel. People should not be against the idea of self-driving cars in the future [GS22], because of the money that will be spent on research and development and the benefits that will come from it. Self-driving cars have many benefits, such as more mobility, lower energy and pollution levels, shorter travel times, and low maintenance costs [Oth21]. Even though there are other problems with self-driving cars that most people won't be able to see or won't know much about, we will only be talking about how people think about them in this paper. To do this, we would need a dataset that includes what people think and feel about self-driving cars. Since the goal was the same, it was decided to scrape data from Twitter, mostly the username and the tweet. Machine learning is a big part of this because it uses keywords to look at this kind of data.

The rise in Autonomous vehicle technology development is gaining great attention in recent years as it anticipates bringing a huge change to transportations worldwide. Even though there are a lot of technical complexities about them and regulatory challenges prevent wider adoption of Autonomous vehicles, most importantly, knowing what the target audience feels and thinks about autonomous vehicles is critical for the general use of them. This project is geared towards unpacking the consumer perception of autonomous vehicles coupled with sentimental analysis. The notion of the autonomous vehicles, generally known as the self-driving or driverless cars, mimics a shift in industry that would eventually be realized to yield immense benefits like the improved road safety, rationality, as well as convenient compared to the traditional human-operated vehicles. Major automakers and the currently innovative companies in the tech industry are pouring a lot of their resources and time into autonomous vehicle research and development. Hence, the adoption of autonomous mobility is unstoppable. Although there are many issues to be dealt with in a society, including trust in technology, safety concerns, regulatory frameworks, and social impacts; these obstacles would need to be tackled.

As a case NLP (natural language processing), sentimental analytics play a crucial function to estimate the public attitude about automobile and their psychology. Sentiment analysis which looks at textual data from different places such as social media, online platforms, surveys and reviews can help and provide knowledge, opinions and feelings about the on the Autonomous vehicle technology and its prevalence in recent time. Besides classifying sentiment, it also bolsters the identification of a prominent idea behind the reviews, determines the level of gratification or dissatisfaction of users, and depicts the intensity of users' emotions, which make it possible to form a sophisticated vision of the customers' perception. The goal of this initiative is to contribute to the field of autonomous vehicles research by using a sentiment analysis approach to identify and understand consumer attitudes toward autonomous vehicles. In a very broad spectrum of textual data sources which include public discourse and consumer feedback, the project aims to uncover elements of unspoken data, patterns, trends, and sentiment drivers influencing the public opinion towards autonomous vehicles. Subsequently, such studies could guide the authorities in the field such as policymakers, industry players and researchers on which aspects of Autonomous vehicles promotion is to be emphasized and with what weight and how to deal with the public concerns.

To put it concisely, the significance of sentiment analysis, from the respondent's perspective, to shape the Autonomous vehicles technology future development and to enabling easier Autonomous vehicles technology integration into big time transportation systems cannot be underestimated. The undertaking has a plan to demonstrate the feelings, views and preferences of these vehicles users, which shall provide important data for the creation of trust, acceptance to all and eventually the international adoption of autonomous vehicles.

3 Background

3.1 Machine Learning

According to a study [WMZ09], "machine learning is defined as the field of study that focuses on enabling computers to imitate human learning processes and develop self-improvement techniques." This involves acquiring new knowledge and abilities, assessing existing knowledge, and continuously enhancing performance and accomplishments. It is in our nature to learn new things and get better at them by doing them. Little do we know about ourselves, and we can't do much for ourselves at birth. Soon, we'll be getting better at what we do and learning something new every day.

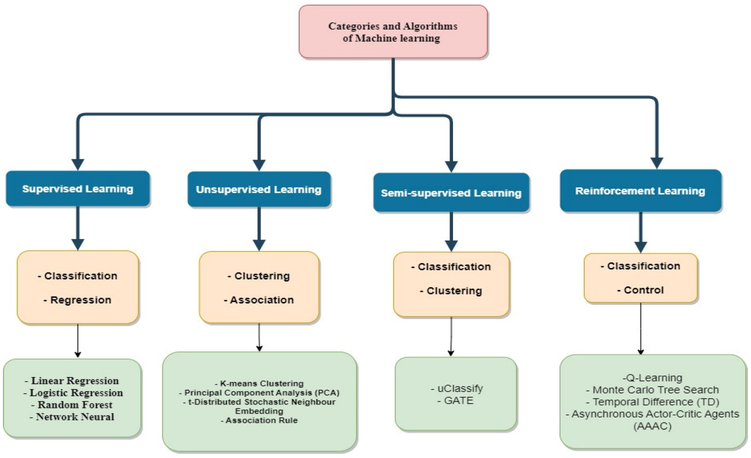

Figure 3.1: Machine Learning Algorithms[Tay23]

The same thing can be done by computers as well. Machine learning is possible because of the combination of computer science and statistics. This makes it possible for computers to learn how to do something on their own. Much like how our brains learn by doing, computers can also gain knowledge through experience. Suppose someone wants a computer to tell the difference between pictures of a cat and a dog. To start, with two pictures and ask people to name one as a dog and the other as a cat. If a computer is taught to learn, it will look at the data for statistical patterns that will help it tell the difference between things that look like cats and things that look like dogs. It might one day figure out on its own that cats have shorter noses and dogs come in a wider range of sizes, and then it could organize that information spatially. But it's very important that the computer, not the programmer, finds those patterns and sets up the sorting algorithm for the next set of data. One example of a simple algorithm that works well is figuring out the best distance between cats and dogs. When the computer looks at a new picture to see where it is on the line, it responds with "cat" or "dog." Of course, mistakes do happen. When a computer gets more information, it can better tune its algorithms and make more accurate predictions.

A lot of people are already using machine learning. It makes a lot of things possible, like text-to-speech, inbox stand filters, viewing suggestions for online shopping, credit card fraud detection, and a lot more. Researchers in both computer science and statistics are working together on machine learning projects at the University of Oxford. The goal is to make algorithms that can solve more complicated problems with less time and computing power. Machine learning could completely change every part of our lives, from how we use social media to how doctors diagnose us. There are two types of machine learning models: supervised and unsupervised. Being shown each one and the different kinds of plants they hold. It uses a group of functions derived from a list of example input-output pairs to connect an input to an output. This type of learning is called supervised learning. There is an example of a supervised learning model that could be used to guess a person's shoe size based on their age, given a dataset with two variables: age as the input and shoe size as the output. Even better, there are two subtypes of this type of learning: regression and classification. Different predictors are used by the regression model to find the target value. This can be used to figure out how two variables, one that is dependent and one that is independent, are related to each other. The output of a regression model stays the same over time. It is one of the most common types to use the linear regression model. It's enough to just find a line that fits the data. This idea can be expanded to include multiple linear regression, which finds the best-fitted plane, and polynomial regression, which finds the best-fitted curve. Each square above is a node in a decision tree. Usually, the more nodes there are, the more accurate the tree will be. In the field of assembled learning, random forests are like decision trees but with more features. Building many decision trees from bootstrap datasets of the original data is part of the process. Each tree step includes a random set of variables. Following this, the model chooses the mode of each prediction made by each decision tree. When the majority winds model is used, it lowers the chance of error caused by individual trees.

A neural network is a well-known model with many layers that are based on how the mind works. The circle shows a node, which is similar to how neurons are organized in the brain. The first layer stands for the input layer. The output layer is shown by the last layer, and the hidden layer is shown in the middle. The input goes through each node in the hidden layer, which is a function, on its way to the output, which is shown by the three circles. When you use classification, you get discrete results. A common type of classification model is logistic regression, which is similar to linear regression in that it looks at the chances of a limited set of outcomes. In supervised learning, two of the main methods used are dimension reduction and clustering. Clustering is the process of putting data points together. It is often used to put customers into groups, find fraud, and organize documents into groups. A lot of people use density-based clustering, K-means, hierarchical clustering, and mean shift clustering to put things into groups. Even though each method uses a different way to find clusters, they all have the same goal. Dimensionality reduction is the process of cutting down on the number of features in a set of features by this amount. Most of the ways to lower the number of dimensions can be put into one of two groups: feature extraction or feature elimination. A common way to reduce the number of dimensions is to use principal component analysis (PCA). Machine learning has applications in a wide range of industries, including manufacturing, finance, healthcare diagnostics, sales and marketing, and many more [SRK21].

3.2 Deep Learning

According to [SM19], "a neural network is a machine learning (ML) technique that is based on and looks like the brain and the human nervous system." Deep learning is the process of inducing mental-like learning in a computer. This field pertains to artificial intelligence and is occasionally referred to as deep neural learning or deep neural networking. Computers can acquire knowledge of patterns and abstract entities via deep learning.

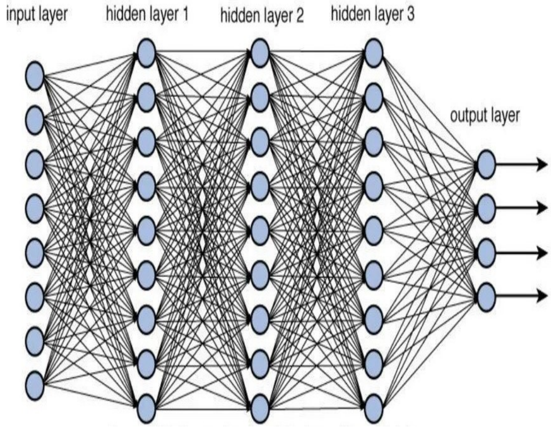

Figure 3.2: Deep Learning Structure[Par]

To gain insight into deep learning, consider a toddler acquiring knowledge about dogs. Dogs are introduced to toddlers through pointing to objects and receiving responses. The toddler employs a hierarchical structure to unpack the concept of a dog, which is inherently complex. Every tier within the hierarchy is constructed upon the insights acquired at the level beneath it. A neural network is a technological component that enables programs to discern the species of flower depicted in an image or the tune being sung. Deep learning is utilized not only in the identification of images within songs but also in speech recognition and translation software, as well as in autonomous vehicles. Superior learning and classification performance in domains including voice, handwritten character recognition, transfer learning, and other fields have been demonstrated using deep learning algorithms [Lau12]. While deep learning is not without its imperfections, deep learning models merely retain the information they were taught and learn by observing. Deep learning models that are trained on a limited or inconsequential dataset will acquire knowledge in a manner that is impractical for the given task.

deep learning is one of the branches of machine learning and it has been shown to possess a transformative impact on artificial intelligence (AI), particularly as was demonstrated in different domains. As deep learning raises its prominence in the autonomous vehicle space, it becomes the key factor which makes these machines smarter and more efficient in terms of decision making whereas looking at the bigger picture it aims to design safer and more competent transportation systems. Provided at a basic level, deep learning is structured on a neural network of the artificial kind mainly on those deep networks that include several layers to address sophisticated problems. These types of neural networks were unusually created to learn an assembly of data in a hierarchic manner, giving them the ability to distinguish sub-patterns and hidden features thereof.

The most vital point is that deep learning models have the specificity to features automatically extract from raw data, so they are the best for image recognition, natural language comprehension, and, specifically, presence detection in driverless vehicles. What is impressive about deep learning in automotive vehicles is that it has application capability in computer vision systems. Convolutional Neural Networks or CNNs feature as very effective types of Deep Neural Networks meant for visual information processing. This way, they can do object detection, lane detection, and recognize pedestrians. Through the use of huge datasets of images and video recordings, for example, such networks can train and identify various objects in real environments themselves, providing the vision that is needed for self-driving vehicles to navigate safely. Furthermore, deep learning is an essential component of the end-to-end learning methods for autonomous vehicles where machines can learn and understand all aspects of driving. Conventional procedure is characterized by actual building of algorithms in order to consequently interpret data from the sensors and to make necessary decisions. On the other hand, end-to-end learning makes use of deep neural networks that directly map sensor inputs with appropriated outputs, this means that the system can learn complex mappings and not these defined in a program. This methodology has the potential of refining the adaptability and strong resilience of automation in changing and demanding scenarios.

Nevertheless, deep neural nets efficiency in self-driven automobiles vitally depends on the availability of enormous, wide data sets assigned for the training. Training dataset's quality, as well as the number and variety are very important when it comes to generalization to the unknown and novel situations. Lastly, the deep learning models' interpretability is a barrier because their decision-making processes are elegant. They are different from the traditional algorithms, and it is difficult to understand them. Autonomous vehicle technology remains in the rapid development path towards fully autonomous vehicles; therefore, the deep learning techniques integration shows great potential that can improve perception, decision-making and the performance of these vehicles. The deep learning research community, along with development efforts of autonomous systems, is facing and overcoming challenges to unlock the potential of artificial intelligence in generating autonomous and robust transportation modes of the future.

4 Autonomous vehicles

Some driving vehicles can sense obstacles and stop when they need to, thanks to advanced technology. Autonomous vehicles, often known as smart cars or robocars, replace human drivers in part or entirely by using a range of sensors, computer processors, and databases, including maps [RJ15]. In addition, self-driving cars can read and react to traffic lights and road signs at intersections. But how did they do this? Some cars can get from point A to point B without a driver. The driver is the technology that is built into and on the car. Lighter sensors, ultrasonic sensors, cameras, and radar sensors are just some of the sensors that are used in the technology. The sensors are in the front and back of the vehicle. On top of that, lidar sensors and cameras are often found on the roof. Without them, the vehicle would be pretty much blind. Laser range finders are an important part of many vehicles. Lidar works in a way similar to radar. Lidar, on the other hand, uses many light rays that are picked up by the lidar sensor after being reflected by things in the environment. This is how the environment around you can be used to make a point cloud. Lidar, on the other hand, can't read road signs. Cameras read and understand street signs with the help of artificial intelligence. Cameras are needed for lane leaving as well as finding obstacles. When there is fog, neither lidar nor cameras work. In this case, radar is needed to warn of and stop impending collisions. Even so, one big problem with radar is that things have to be big enough to be picked up. So, people walking and riding bikes might be able to avoid being picked up by radar systems. Together with very accurate digital maps, GPS technology was used to figure out exactly where the vehicle was. Digital maps have a lot of information on them, like speed limits and signs for intersections. Geometry is also very important in places where GPS accuracy isn't always guaranteed, like in steep canyons and tunnels. For example, the diameter and speed of the wheels can be used to get a rough idea of where the vehicle will be in the future.

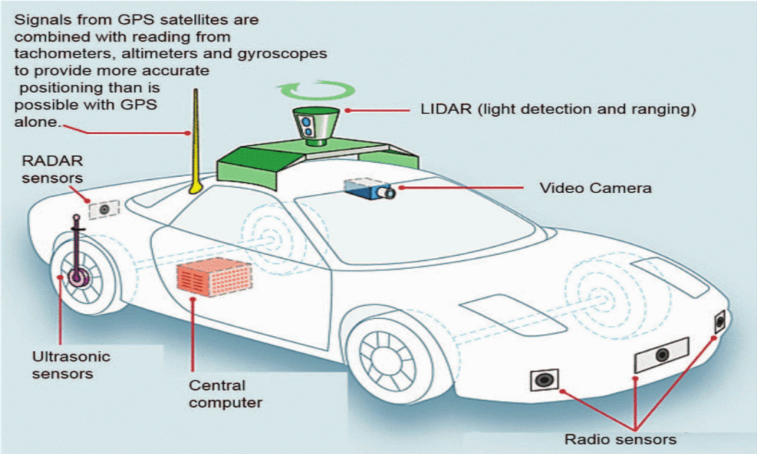
Signals from GPS satellites are combined with reading from tachometers, altimeters and gyroscopes to provide more accurate positioning than is possible with GPS alone.
LIDAR (light detection and ranging)
RADAR sensors
Video Camera
Ultrasonic sensors
Central computer
Radio sensors

Figure 4.1: Autonomous vehicle Architecture [RJ15]

The rise of Autonomous vehicles brings the revolution into transportation that is not only growing but also changing the way we move from one place to another. Called self-driving / driverless vehicles, autonomous vehicles incorporate cutting-edge sensors, software with machine learning / and AI, to get instruction and move around without human interaction. The increasing of autonomy for vehicles has been going through a very fast process recently and is being strongly sponsored by the growing of investments both from automotive as well as technology companies. Numerous manufacturers are in the first row of researchers and developers of Autonomous vehicles technology fighting to be the best and to bring entirely the idea of self-driving to life. The future of self-driving vehicles will change the path of conventional transport systems through various reportedly challenging problems. The emergence of automated vehicles is not only driven by the goal of increasing safety but also has a positive impact on reducing the number of traffic-related accidents. Supporting proponents, the rise of autonomous vehicles can lead to a reduction of the number of accidents in the streets that are attributed to human error, which is by the way the main cause of death on the roads. Autonomous vehicles not only employ real-time data but also their advancing algorithms can ensure quick and accurate responses to potential dangers. Also, they can do it way better than humans. Besides, self-driving cars open up a possibility to solve transportation problems in regard to efficacy, accessibility, and communication between the vehicles. Capable of exchanging information with vehicles and infrastructure, Autonomous vehicles can build the most effective routes that significantly cut traffic and move vehicles from the lanes, therefore improving traffic flow. As people will prefer to work closer to or even from home, commuting times may become shorter, less fuel will be expended in the process and emissions will reduce, thus contributing to a better and more environmentally friendly transportation system.

Notwithstanding the recent big jump in the level of self-driving vehicle technology, there are many unresolved challenges and hampering to its usage. There is probably no more crucial issue than the requirement to ensure the confidence and trust of members of the public in reliability and security of autonomous vehicles. Embarrassing crashes or breaking down in undesirable environments of Autonomous vehicles have contributed to safety and ethical concerns with passengers and led to be any speculations and fears on their footing. Similarly, through the existence of regulatory frames, legal issues also pose trouble for the development of autonomous cars on a large scale. The quandaries of who is responsible in cases of accidents, data privacy and cybersecurity as well as how ethics of artificial intelligence algorithms will be addressed remain to be fixed by the cooperation of politicians, private sector stakeholders, and research scholars whose goal should be to come up with the appropriate legal frameworks and rules. To make it clearer, self-driving cars would revolutionize the system of transport giving people the chance to be safe, productive and convenient at the closest places. It can be the case though that the implementation of this potential might be more complex thanks to different technological, regulatory, and societal complications. The insights about vehicle automation advantages and drawbacks if utilized effectively assist the stakeholders to manage well the benefits associated with technological transformation while minimizing the associated risks.

5 Sentiment Analysis

Being aware of the tone of discourse surrounding one's business or product proves advantageous in numerous respects. You can accomplish this by utilizing social media, product reviews, or email feedback. Individuals can discuss your brand and yourself in a variety of ways. Understanding the thoughts and emotions of others, whether expressed directly or indirectly, is a critical determinant of business success. Herein lies the importance of analyzing how individuals feel. In this paper, we define "sentiment analysis," explain why it is beneficial for organizations and businesses of all sizes, and demonstrate how it can be implemented. One method of sentiment analysis is machine learning, which uses renowned computer learning algorithms to categorize text according to its polarity using linguistic characteristics [SS17]. To examine and comprehend the tone of the text, sentiment analysis can be utilized. Alternatively stated, it is a method of examining the emotions that underlie a written work, such as a social media post. This may enlighten regarding the true nature of your organization's brand image, among other things. To illustrate, sentiment analysis could be employed to categorize tweets about a particular brand into distinct groups, such as positive, negative, or neutral. Alternatively, you can go even further by employing categories such as pleased, upset, anxious, or disappointed. Urgent text can be identified and sorted using sentiment analysis.

Sentiment analysis is frequently applied to product reviews, survey responses, and user experience. How does sentiment analysis function? Sentiment analysis first utilizes a blend of machine learning and natural language processing (NLP) to detect significant observations from the language used by individuals automatically. By examining the phrase orders and context, one can discern the specific words and phrases that were employed. "I adore this brand" is an easy example of this. Which is undeniably advantageous. However, complications do arise on occasion, as in the following tweet: "Your organization is killing it." Typically, the term "killing" connotes a negative connotation; however, this particular instance appears to be a positive tweet. This is because individuals fail to discern the intended significance of written material, such as irony or sarcasm. Sentiment analysis is plagued with numerous issues. However, as time passes, AI models become increasingly proficient at deciphering human speech in text.

Textual data sentiment analysis, which is another way the name opinion mining is referred to, falls under the category of the natural language processing (NLP) branch of information technology. It basically revolves around extracting, identifying, and analyzing subjective information. The sphere of autonomous vehicles is the best example of when a sentiment analysis feature holds a great importance for figuring out what people feel, think, and see on the emerging rises of given technology.

The basic purpose of emotion analysis is to identify the polarity of the mood in textual data and classify it into positive, negative or neutral. Generally, it may also involve the application of machine learning, statistical or linguistic approaches so as to basically

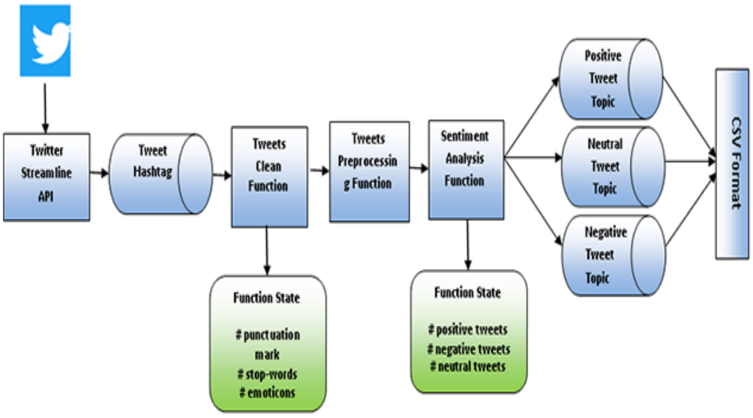

Figure 5.1: Sentiment Analysis Process [Gar20]

classify the text into different sentiments such as there is positive, negative, or neutral sentiment. Sentiment analysis enables the analysis of a wide spectrum of textual data sources including social media posts, online reviews, news articles, forum discussions, and surveys among other things. When it comes to self -driving cars, sentiment analysis has the ability to give a lot of information that will show the factors that affect the general acceptance or disapproval and awareness. Through the entailing of large sets of textual data from every type of source, researchers will be able to find the prevalent sentiments, highlight the main themes, and study the fundamental influences guiding customers' perception. As an instance, sentiments analysis may throw light on the views of the population about the safety, reliability, convenience, and ethical issues surrounding autonomous cars. Positive sentiments represent trust in technology, satisfaction with its deals, or support with the accomplishment of autonomous travel. On the other hand, such negative reactions may arise from people's worries about safety or their fears of unemployment due to autonomous vehicles, the uncertainty of the reliability of Autonomous vehicles and the issue of autonomous decision-making ethics.

On the other hand, sentiment analytics possesses the capability to measure the outcome of marketing campaigns, public relations activities, and policy schemes targeted towards technology adoption among individuals. Through long-term attitude monitoring, which identifies both negative and positive perceptions of varying causes and target audiences, the organization will be able to develop strategies that are sensitive to specific issues and build trust among future user groups as well. On the flip side, a lot is at stake with sentiment analysis in terms of the influence of reactions, the level of subjectivity, and ambiguity of language. In an environment in which contextual matters arise, sarcasm, irony and unlikely discrepancies of cultural profiles are some of the factors that can be attributed to a pertinent and accurate interpretation of sentiment from textual data. Also, a sentiment analysis tool may tend to produce bias or wrong conclusions, therefore the results will be mistaken.

6 Methodology

This Methodology includes two datasets. The first dataset(D1) includes tweets on self-driving cars that were scraped on Twitter using a Python script. It contains details on each tweet, including the time, tweet ID, tweet text, and sentiment. The tweets discuss a range of topics related to autonomous cars, such as innovations, employment prospects, safety issues, and technology improvements. This dataset offers a snapshot of the public's perceptions and conversations at that specific moment about autonomous cars. Researchers, experts, and professionals with an interest in sentiment analysis, public opinion, and trends relating to autonomous cars may find it useful [YWMW20]. The dataset(D1) includes a 4 columns consisting of 'Date', 'Tweet_Id', 'Tweet' and 'Sentiments' and this dataset consists of total of 78,000 tweets to classify into sentiments. Researchers may learn more about how people see autonomous cars and the prevalent opinions regarding this coming technology by examining the emotions conveyed in the tweets. The methodology repeats for the second dataset. The second dataset(D2) is more of a labelled dataset from a well known source Kaggle [dat].The dataset(D2) consists of 11,000 datapoints or texts to analyze. The dataset(D2) consists of total of 11 columns. Later the dataset(D2) is preprocessed to size down the columns to 4 columns consisting of 'sentiment', 'sentiment:confidence', 'sentiment_gold_reason' and 'text'. Results emerging from both the datasets are compared with each other to remove the bias that might arise out of a labelled dataset or a scraped dataset.

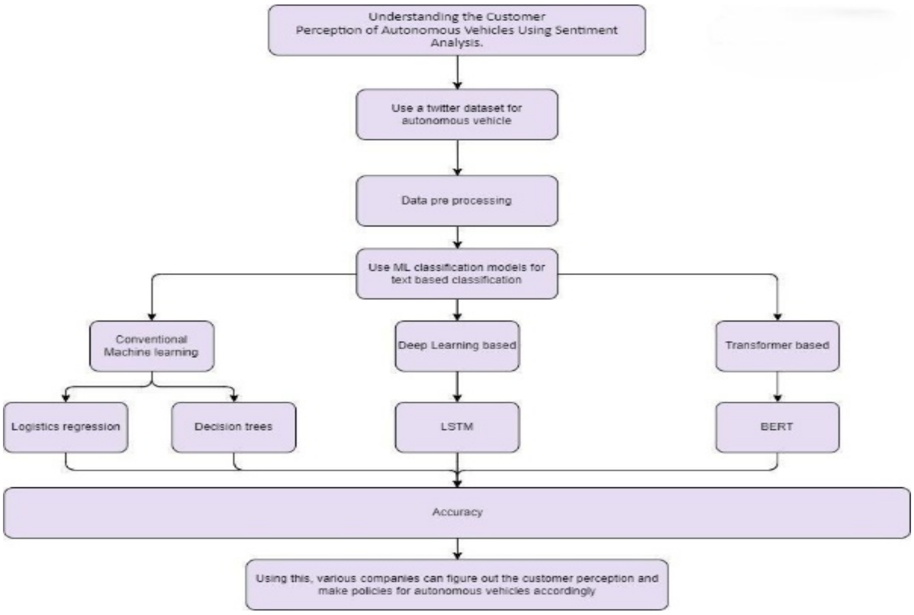
Understanding the Customer Perception of Autonomous Vehicles Using Sentiment Analysis.
Use a twitter dataset for autonomous vehicle
Data pre processing
Use ML classification models for text based classification
Conventional Machine learning
Deep Learning based
Transformer based
Logistics regression
Decision trees
LSTM
BERT
Accuracy
Using this, various companies can figure out the customer perception and make policies for autonomous vehicles accordingly

Figure 6.1: Sentiment Analysis Flow Chart

The methodology for this study involves several steps. The methodology for understanding the customer perception of Autonomous Vehicles Using sentiment analysis starts with importing the dataset in the environment. For example, consider a dataset which is scraped from twitter. The data is then preprocessed to a more suitable format for the study. The data preprocessing involves data cleaning, data transformation, feature engineering, dimensionality reduction, removing redundant values and optionally data augmentation. The dataset consisted of three types of sentiments, positive, negative and neutral. In this study, we focused on positive and negative sentiments texts only. For this reason, neural texts were completely dropped from the dataset for further evaluation.After preprocessing, different models are created based on the type of machine learning techniques. In this study, four types of algorithms were used; Logistics Regression, Decision trees, LSTM and BERT. All the models were created for binary classification as we mainly were focusing to classify sentiments between positive and negative sentiments only. After the models are created, the datasets are splitted into training and testing dataset. The data is trained on the four models of different algorithms and tested on the testing set. This gave out different accuracies for the individual models. These accuracies are then compared between the two datasets to weed out the bias. The results of the sentiment analysis are then interpreted to gain insights on how the various data techniques perform on sentiment analysis. The findings or results involves different accuracies for different algorithms or models for two different datasets, D1 and D2. Further visualization techniques are used to display the findings. In this study, python based libraries are used for visualizing results rather than the tools. Through this comprehensive methodology, the study aims to provide valuable insights into the customer perception of autonomous vehicles. This study will help the stakeholders in autonomous vehicles sectors to improve and educate customers in relation to customers or public concerns regarding Autonomous vehicles. Also, this study will encourage the companies to incorporate such techniques to gain insights into marketing perspective of autonomous vehicles.

It is crucial to remember that the dataset only represents a single moment in the past, and the feelings stated may not always reflect the whole range of views on autonomous cars. The dynamic nature of public opinion means that people tend to change their opinion from time to time and also it changes over a period of time. This study does not aim at studying public sentiments towards autonomous vehicles in current time or a specified period of time and date. Whereas, this study aims at the approach to identify sentiments of public towards autonomous vehicles at any given point of time. Nevertheless, this dataset is a useful tool for sentiment analysis, mining opinions, and gaining insight into the debate around autonomous cars. This can also be an useful tool for policy making by the governments. The government doesn't have to rely on industries or survey companies to determine the problems and concerns of public and make necessary policies accordingly.

7 Machine Learning Techniques for Sentiment Analysis

7.1 Conventional Machine Learning

7.1.1 Logistic regression

The logistic regression method, which comes from statistics, has been used in machine learning. When there are only two possible class values, this is the best way to solve the problem. Considering a scenario with the task of determining whether an email qualifies as spam or not. There is a need to set a classification threshold to use linear regression to solve this problem. People will mistakenly think that the data point is not malignant if the predicted continuous value is 0.5 and the actual class is malignant at 0.4. This classification mistake may have very bad effects in real-time. Logistic regression encompasses various forms, including binary logistic regression, which deals with a categorical response that can only assume one of two alternatives, such as "spam" or "not spam."

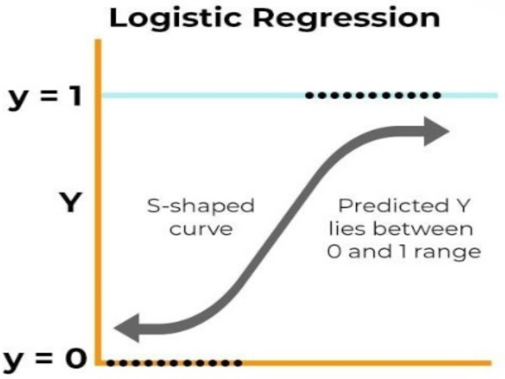

Figure 7.1: Logistics Regression Architecture[spi]

Multinomial logistic regression encompasses at least three unordered groups. Like guessing which fruits, vegetables, or nuts people will like best. Ordered lists of at least seven categories were used in ordinal logistic regression. Taking a look at the scale from 1 to 5 for movie ratings. The logistic function, upon which logistic regression is founded, is the source of the method's nomenclature. The logistic function, commonly known as the sigmoid function, is a curve that transforms any real number into a value ranging from 0 to 1. However, it never remains precisely at those two figures. Using this function, data, which is emails, will be put into groups based on whether they are spam or not, as shown in the previous example. If 'Z,' the result of the examples going through the sigmoid function, gets close to infinity, Y (predicted) will become 1. On the other hand, if 'Z' gets close to negative infinity, Y (0) will become 0. There are, of course, many other types of logistic regression and more complex extensions; this is just the most basic one. There will be irrational uses of the logistic regression technique as many academics rarely thoroughly examine the underlying theoretical models and assumptions [ZHTS19].

7.1.2 Decision trees

Tree-like structures known as decision trees assist individuals in the decision-making process by posing a sequence of inquiries until the appropriate response is obtained. The stage and level of complicated multi-stage decision-making difficulties are evident, making it easy for the decision-making group to collaborate and thoroughly analyze a range of elements that will help it make the best choice [LYZ22].

Decision trees is one of the most powerful machine learning technique yet a simple one. The structure of decision trees looks like a tree but working like a flow chart where each node represents a feature and each branch denotes a decision which is based on that feature and each leaf represents the result or classification. Decision tree works like splitting of a tree or branching from a tree where the outcome depends on the feature and the decisions it took in the process. Decision trees is mostly used for classification or decision making problems. Consider a decision making problem with a data, Decision trees will keep on splitting the data based on features until it reaches the final conclusion. While its splitting, it also trains repeatedly to learn in the process. Although it is simple, decision trees can handle complex data and both the classification types, binary and categorical. The issue with Decision trees is that is susceptible to overfitting. overfitting refers to over learning or over training of data which creates problems when it is used for a new data. However this problem can be mitigated in Decision tree classifier and therefore it is one of the most basic and simple yet powerful machine learning model which can be used for large and complex data as well.

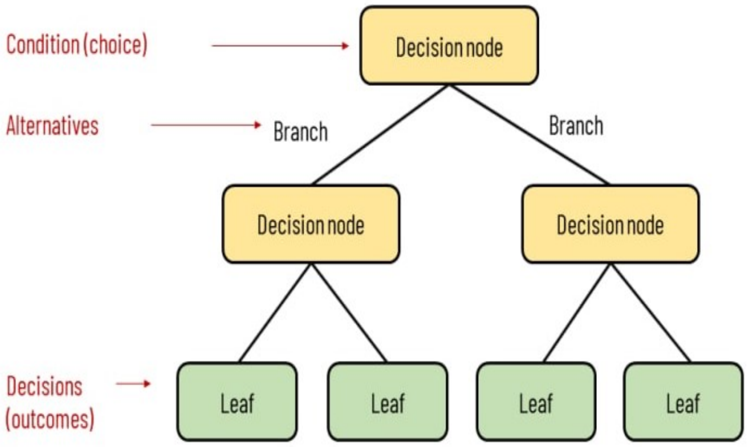

Figure 7.2: Decision Trees Architecture[Kos]

For instance, if it is to be estimated the range of a vehicle by considering its weight and the number of cylinders it possesses. Potentially, a decision tree would resemble this. Before proceeding, one can determine the vehicle's weight. What is its weight in pounds? If affirmative, one shall adopt a narrow range. If not, it is must to determine the number of cylinders. The range is increased when the number of cylinders is reduced to six. When the number exceeds six, the range becomes narrow. As a consequence, the range of a heavy vehicle will invariably be limited. The range of a light vehicle with fewer than six cylinders is greater than that of a light vehicle with more than six cylinders. The decision tree suggests that a new automobile featuring eight cylinders and a substantial weight possesses an inadequate range.

7.2 Deep Learning Based

7.2.1 LSTM

LSTM networks are a unique kind of RNN that have the ability to learn long-term dependencies [NSS20]. In deep learning, least significant tree (LSTM) networks are composed of layers. Long-short-term memory is the abbreviation for LSTM. They are extraordinarily complex recurrent neural architectures that retain information from the past with remarkable accuracy. When neural networks are fully connected and feedforward, it is considered good. However, their capability is limited to solving static problems, meaning that the output is solely determined by the input at hand.

Long Short-Term Memory (LSTM) networks are type of recurrent neural network (RNN), which are used to solve complex language related problems. Unlike traditional RNNs, LSTM is little different. LSTM remembers the previous information captured while on old steps and can also determine the irrelevant information and forget it.This feature helps LSTM with sequential data tasks like speech recognition, sentiment detection, and time-series analysis. From Figure 7.3, it can be observed that the LSTM has three gates, Forget gate, Input gate and Output gate. Each gate is responsible for the flow of information. The LSTM architecture also contains a memory cell which retains the necessary required information during the flow of data. This technique helps LSTM learn in a more powerful way by remembering the old required information as well the new information and training over it repeatedly. LSTM efficiency is high when it comes to short sequences. In long sequences there are chances that LSTM might not perform well. Nevertheless, LSTM is a powerful model and the architecture of LSTM has paved the way for more powerful architectures like Gated Recurrent Units (GRUs). Another advanced architecture based on LSTMs are transformer based models. This study has a smaller sequence compare to the real life data on which Large language models(LLM) are trained. This was the reason for LSTM performing better as compared to other methods for the two datasets.

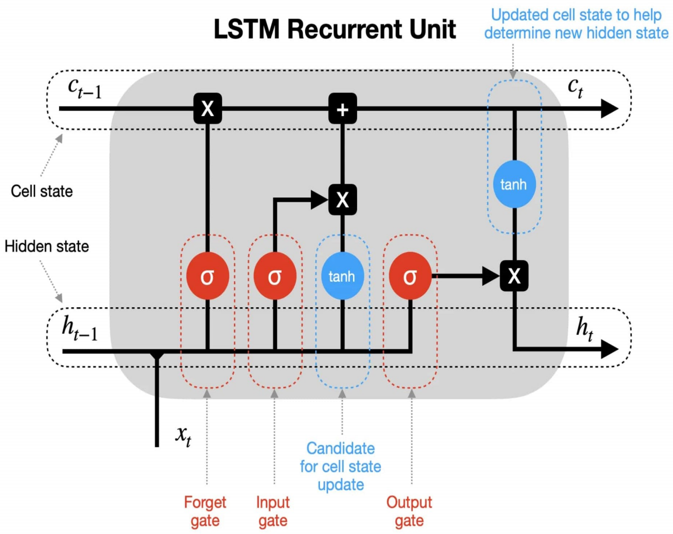

Figure 7.3: LSTM Architecture[Dob]

At time t, the outcome is contingent on what is introduced. However, certain issues require the use of time series, which are sequentially ordered events that occur over time. To improve the performance of fully connected feedforward networks, automatic backward connections were incorporated. These particular types of connections alter the network's response time. We implemented it by incorporating a memory of previous inputs using the output from time t-1 before the current moment.

7.3 Transformer Based

7.3.1 BERT

BERT is, so to speak, a language model capable of acquiring knowledge of particular language tasks. When using word(s) as input vectors in context-related classification tasks, the language model's bidirectionality helps produce better results [Kal20]. The ability to do so is attributed to its comprehension of language. It understands the relationships between words, allowing you to fine-tune it to perform well on a variety of language tasks. These models are trained in two distinct phases. The process of training the language model initially is known as pre-training. We then further fine-tuned the model. The ability of these models to comprehend language enables them to perform a multitude of tasks.

BERT is a transformer based architecture that uses different pre-trained models also called as Large language models(LLM). As mentioned earlier, most of the models only performs in one single direction. Whereas, BERT can go into either direction. That means it can understand words in front of it and if necessary, it can go backward as well. It will understand the meaning of the word in both direction and that's where the term bidirectionality is referred as. Unlike other models, this feature allows the BERT to understand the words in a sentence in a better way with respect to context and the meaning of the words associated with the sentence. Though BERT was developed by google, BERT's pre-trained models are open source and are available easily. This pre-trained model is trained on huge amount of data with the help of unsupervised learning and also fine tuned. Due to its ability to work on huge data with high accuracy to tackle various problems as well as its ability to work with complex data, BERT is one of the popular language models that can be used in Natural Language Processing(NLP). ChatGPT or BARD are some of the popular generative AI models that are based on transformer based Architecture.

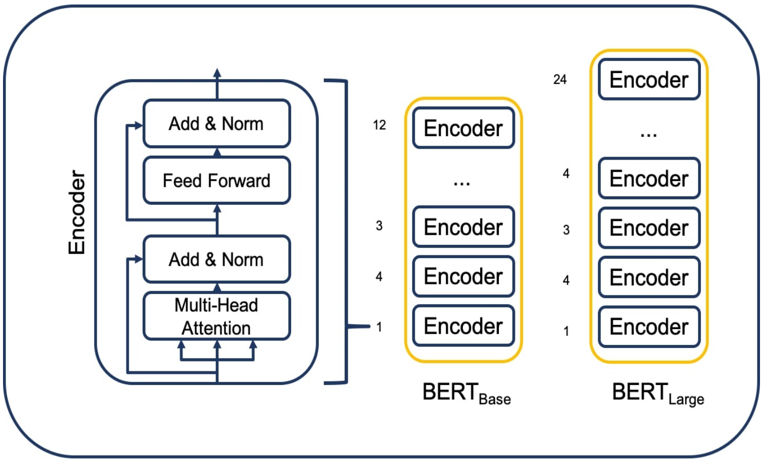

Figure 7.4: BERT Architecture[EMF]

Building a working BERT model requires considerable effort and an abundance of data. For this reason, it has been retrained for our benefit. Now, in order to utilize BERT, it is sufficient to fine-tune it for the specific task at hand. Transformers have provided encoders and decoders for our use. Thus, the encoder learns how the language is utilized, while the decoder performs the precise task at hand. Encoders were stacked upon one another, utilizing BERT. BERT consists, so to speak, exclusively of encoders. The encoder section is where the context is learned in the standard transformer architecture. This context is then transmitted to the decoder to assist it in completing the trained task. It was translated for the most part.

8 Implementation

The code [kag] sample shows how to train a model using the well-known deep learning framework Keras. The 'fit' function is specially used in this code to train a categorization model. First, a copy of the 'History' callback class is made; this class can be used to hold the model's training data. Throughout the training process, the 'History' object can record different metrics and losses. The number of initial samples handled in each iteration of training is determined by the value of the 'batch_size' parameter, which is set to 128. Depending on the amount of available memory and processing power, this number may be changed. To provide weights to various classes in the dataset, the dictionary 'class_weights' is created. Class 0 is given a weight of 1, and class 1 is given a weight of 1.6 divided by the prejudice in this instance.

```
from keras.callbacks import History
history = History()
batch_size = 128
# also adding weights
class_weights = {0: 1 ,
                1: 1.6/bias}

model.fit(X_train,
          y_train,
          epochs = 15,
          batch_size=batch_size,
          verbose = 1,
          class_weight=class_weights,
          validation_data=(X_test, y_test), callbacks=[history],
          steps_per_epoch=50)
```

The above code snippet is the sentiment analysis model for LSTM. This is done to correct any class disparity in the data set and guarantee that the model gives the minority class greater consideration while being trained. The training procedure is started by using the 'model.fit' function. The inputs include the training data ('X_train' and 'y_train'), as well as parameters for the validation data ('X_test' and 'y_test'), the 'History' callback object, the size of the batch, the verbosity straight, class weights, and the total number of epochs (15). The model can go over the data again and again during training for the set number of epochs. Using an optimization approach like stochastic gradient descent, the model can analyze batches of training samples, compute the loss, and change the weights of the model in each epoch.

```
y_pred = np.argmax(model.predict(X_test, batch_size=batch_size), axis=1)
df_test = pd.DataFrame({'true': y_test.tolist(), 'pred':y_pred})
df_test['true'] = df_test['true'].apply(lambda x: np.argmax(x))
```

```
print("confusion matrix",confusion_matrix(df_test.true, df_test.pred))
print(classification_report(df_test.true, df_test.pred))
```

The "history" object, which can keep track of the training metrics or losses for subsequent analysis, is sent via the "callbacks" option. 'steps_per_epoch', which controls how many steps (batches) are going to be analyzed in each epoch, is set to 50 as a final setting. The size of the training dataset may determine how this parameter is changed. This code snippet shows how to build up and train a model with certain parameters, such as class weights, batch size, and validation data, to solve class imbalances and enhance the training process.

The supplied code snippet relates to assessing predictions from a classification model and producing a confusion matrix. NumPy, a library written in Python for numerical calculations, is used in the code [TLT^{+}21]. The '_pred' variable is first assigned to the outcome of using the 'np.argmax' function on a certain axis in an array of values. The indexes of the highest values along the chosen axis are returned by the 'np.argmax' function. The given code does not provide the precise axis that was utilized in this instance. The anticipated class labels are then set to the variable 'f_test' by using the 'np.argmax' function with the variable 'x'. Next, the confusion matrix is calculated by executing the 'np.int' function on the word "confusion matrices" or combining it with the actual class labels ('true') and predicted class labels ('df_test.pred').

```
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs = range(1, len(acc) + 1)

plt.plot(epochs, acc, label='Training Acc', color='blue')
plt.plot(epochs, val_acc, label='Validation Acc', color='red')
plt.legend()
plt.title('Training and Validation Accuracy')
plt.show()
```

The number of instances of true positive, genuine negative, false positive, or false negative predictions is shown in the confusion matrix, which gives a summary of the model's performance. The code then calls the print function on the parameters true and 'df_test.pred' to output the confusion matrix. In conclusion, this code snippet computes anticipated class labels, produces a confusion matrix, and then prints the outcomes using NumPy methods. The confusion matrix offers important information about the model's accuracy and mistake rates for categorization. The above explanation is for LSTM.

The logistic regression code begins by creating a model based on 'LogisticRegression()' class. The approach used is binary classification which predicts binary results. The 'fit()' method is used to train and test the model. In this code, 'X_train' resembles the feature variables or training data and 'y_train' represents the target variable. The target variable is in format where the classes are in binary form. During training, the logistic regression model learns the relationship between the features and the target classes. At the end of the training, the 'predict()' method is used to generate predictions for the test dataset which is 'X_test'. These predictions are then stored in the 'prediction_lr'. This is further analyzed. Lastly, the ’model_Evaluate()’ function is used. This function evaluates the model’s performance and prints out a classification report consisting of metrics such as precision, recall, F1-score, and the confusion matrix. This model can be further improved by analysing the report and making the necessary changes in the model or by optimizing it. The below code, represents the ’model_Evaluate()’ function to be implemented within the logistic regression model.

```
lr = LogisticRegression()
lr.fit(X_train, np.argmax(y_train, axis=1))
prediction_lr = lr.predict(X_test)
model_Evaluate(lr)
```

The Decision tree classifier code is evaluated using scikit-learn. The code begins with importing all the required libraries. A decision tree classifier is created using the default settings and therefore has no changes in hyperparameters. Using the ’fit()’ method, the classifier is then trained on the training data that is ’X_train’ and ’y_train’. Then the ’predict()’ method is used to predict for the test dataset that is ’X_test’. The accuracy of the model is calculated by comparing the predicted values with the true values of the test dataset. This accuracy interprets the model’s performance. Finally, the ’model_evaluate()’ function is passed. Similar to the previous model, this model evaluates a classification report consisting of different metrics using the function ’model_evaluate()’.

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
import seaborn as sns

# For decision tree classifier
classifier = DecisionTreeClassifier()
classifier = classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

model_Evaluate(classifier)
```

The next algorithm BERT, is used for sentiment analysis using TensorFlow. The code begins with the preprocessing by importing the required libraries and using the balanced dataset which includes equal number of positive and negative samples. After that the dataset is mixed to randomize it. Then the BERT tokenizer ’AutoTokenizer’ is imported using the checkpoint function with checkpoint as ’bert-base-uncased’ and the BERT model ’TFAutoModel’. Missing values are removed from the dataset. Further, the dataset is splitted into training, validation and test sets. Moving ahead, a tokenization function is introduced using BERT tokenizer. ’get_model’ function is used further to create a model which has BERT incorporated into it using Keras. The said model consists of dense layer with ReLu activation function, dropout layer and dense layer with sigmoid activation function. This model is compiled with loss as binary cross-entropy and Adam as optimizer for binary classification. Function ’test_result’ evaluates the model’s performance on the test set by predicting probabilities and then generate a classification report. Overall, this code implements a pre-trained BERT model using TensorFlow to evaluate the accuracy for sentiment analysis for autonomous vehicles.

```
from transformers import AutoTokenizer, TFAutoModel
from IPython.display import clear_output

checkpoint = "bert-base-uncased"

tokenizer = AutoTokenizer.from_pretrained(checkpoint)
model = TFAutoModel.from_pretrained(checkpoint, output_hidden_states=True)
clear_output()
```

```
from sklearn.metrics import classification_report

def test_result(model):
    test_inputs = tokenization(dataset["TEST"])
    result_proba = model.predict([test_inputs.input_ids, test_inputs.attention_mask])
    result = np.array(result_proba).ravel().tolist()
    result = [1 if x > 0.5 else 0 for x in result]
    print(classification_report(targets[’TEST’], result))
    return result_proba, result
```

The above implementation works the same for both the datasets. Although, pre-processing works differently for both the datasets, the model creation and the evaluation part remains the same. Both the datasets when evaluated have different accuracies for different algorithms. The results can be observed and compared with each other and are mentioned in detail in Results section of this study.

9 Results

Here it is also seen that the final accuracy for this calculation in this dataset(D1) model is 0.504 and The offered code is intended to assess a random prediction model's average accuracy using a self-driving sentiment collection. The code makes use of the Pandas library to manipulate data, the random module to provide predictions at random, and the counter class obtained from the collection module to determine the sentiment distribution in the dataset. First, the code uses the read_csv function to import the dataset into a Pandas DataFrame.

```
import pandas as pd
import random
from collections import Counter

test_data = pd.read_csv('self_driving_final.csv', encoding='latin-1')

sentiment_counts = Counter(test_data['Sentiments'])


average_accuracy = sum(accuracies) / num_iterations

print("Average Accuracy:", average_accuracy)
```

The dataset's(D1) 'Sentiments' column is then combined with the Counter class to get the sentiments' distribution. The code then specifies the variables required for accuracy calculations and iterations. The accuracy values received after each iteration are recorded in an empty list called accuracies. The number of consecutive repetitions is set to 1000.

Based on the range of emotions acquired from the counter class, a random emotion is chosen for each data point within the dataset D1. The sentiment distribution values and keys are sent as parameters to the random.choices function, which then chooses one sentiment as the anticipated sentiment. The predicted sentiment is expanded to include this forecast. The algorithm then determines the number of accurate predictions by contrasting each predicted sentiment against its associated real sentiment after producing anticipated sentiments for the full dataset. Following that, the accuracy is calculated by dividing the overall correct predictions by the total number of attitudes that were predicted. The accuracy list now includes this accuracy value. The average accuracy is then calculated by adding together all of the accuracy values in the accuracy list and dividing that total by the total number of iterations. The result displays the average accuracy. This method generates random sentiment forecasts for an autonomous sentiment dataset D1 in order to assess the median precision of a randomized prediction model [ESB+21].Similar approach was used for Dataset D2 to analyse the public sentiment towards autonomous vehicles.

From Figure 9.1, the LSTM scored the highest accuracy of 89% amongst other algorithms.

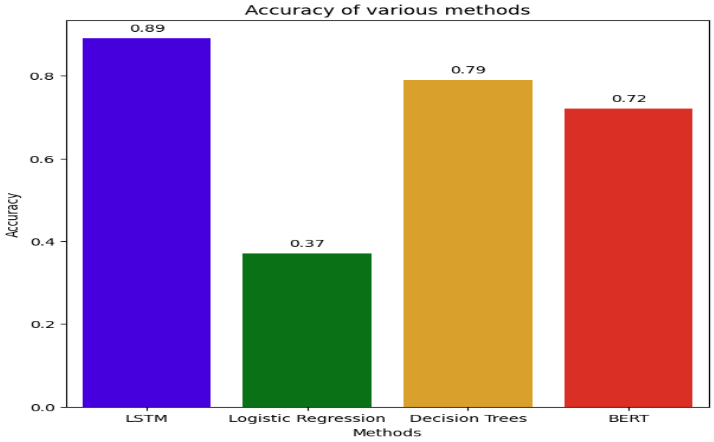

Figure 9.1: Accuracy of Various Methods on Dataset D1

Decision trees were not much behind LSTM, with an accuracy score of 79%. It can be seen that deep learning-based algorithms are better suited for text-based sentiment analysis. In fact, conventional machine learning works well as well. The transformer-based algorithm BERT couldn't work well, but a higher version of the algorithm, like GPT-4, could improve the results.

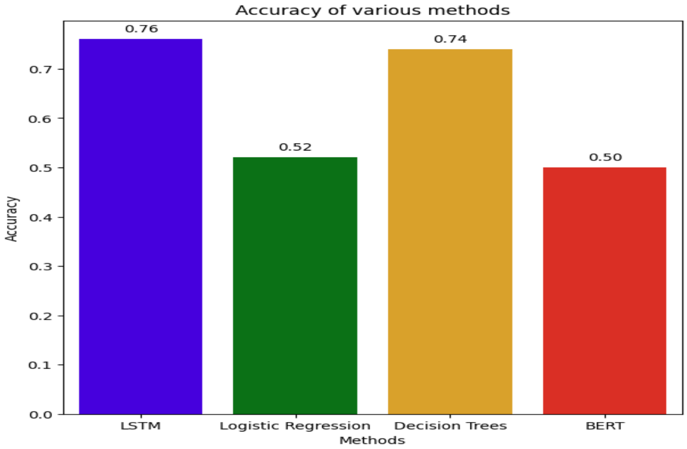

Figure 9.2: Accuracy of Various Methods on Dataset D2

From Figure 9.2, It was observed that , when analysis is performed on dataset D2, LSTM scored the highest accuracy of 76%. Decision tree classifier scored an accuracy of 74%. Whereas, Logistic Regression and BERT scored an accuracy of 52% and 50% respectively. In both the cases, it was observed that lexicon based algorithm that is LSTM scored the best amongst other algorithms followed by Decision tree. This study also proved the assertion that LSTM is a lexicon based model and it works better for analysing the polarity of sentiments.

	Accuracy %			
	LSTM	Logistic Regression	Decision Tree	BERT
Dataset D1	89%	37%	79%	72%
Dataset D2	76%	52%	74%	50%

Table 9.1: Comparing Accuracy on Dataset D1 and D2.

10 Future Work

Several promising techniques or methods can be explored to understand customers perceptions of autonomous vehicles using sentiment analysis.Using machine learning algorithms, specifically deep learning techniques, can be more thoroughly researched to achieve more accurate results. Doing this will be more beneficial in achieving the sentiments of customers for the respective automobile companies. These companies will be able to take decisions with regard to autonomous vehicles, creating a win-win situation for both companies and customers. This also makes it easier for technological advancements to progress at a faster rate than with a trial-and-error approach. Although theoretically, transformer-based algorithms should work really well, a lack of computing resources and a higher version of the transformer-based algorithm might change the scenario. Additionally, on Dataset(D2), datapoints were less compared to the expected size. In Dataset D2, a decent size of texts would have been beneficial for model like BERT to achieve high accuracy. The size of Dataset D2 is the major reason for low accuracy on BERT model. Data augmentation would also be a helpful method to overcome the biasness or to optimize the model in case of D2. With respect to datasets, two different datasets with similar size and specifications can be tested and compared to check for its accuracy towards predicting sentiments. To some extent, fine tuning of models can help in achieving a high accuracy. Not only sentiment analysis for autonomous vehicles, but the approach of this study can also used in different studies including predicting human sentiments, or understanding animals and their sentiments as well as it has the potential to understand product's market for different industries or companies. This study was based on text based analysis. Integration of multi-modal data like incorporating additional data apart from text like images or videos can help with more accurate scenarios to begin with. Another perspective is to have a longitude Analysis. Longitude Analysis means the study of sentiment over a period of time.

The other factor to take into account is the social study to be done on customers. Customers are hesitant to use autonomous vehicles, or some might have some reservations about the autonomous vehicle's safety and other issues. This can be addressed by companies or researchers using sentiment analysis of customers. The use of this study can be further extended to government agencies for policy drafting as well as in research field like robotics where the machine can be made to predict sentiment when its interacting with humans.

11 Conclusion

In this project, four machine learning algorithms were tested to determine customers perceptions of autonomous vehicles using sentient analysis. Two datasets were examined for the study. Dataset(D1) was the dataset scraped from twitter whereas Dataset (D2) was available via Kaggle. Both of these datasets had their own specifications. The algorithms were chosen based on the types of machine learning algorithms, like conventional machine learning, deep learning, and transformer-based machine learning. After comparing the accuracies of models or algorithms with two different datasets D1 and D2, It was observed that, a deep learning-based algorithm like LSTM worked the best for text-based sentiment analysis. Currently, the deep learning-based algorithm LSTM scores the best accuracy, followed by conventional machine learning algorithms like decision trees.Transformer-based algorithms like BERT and logistic regression has the lowest accuracy.

It was observed that using machine learning techniques and improving them, it is possible to determine the sentiment of general people towards autonomous vehicles. This sentiment analysis focuses on approach of working around the sentiment analysis of public opinion towards autonomous vehicles. As the opinion of public changes over time and is dynamic in nature, this stud does not focuses on the sentiments itself and does not correlate with the sentiments of the public towards autonomous vehicles. This will eventually help companies or researchers address concerns and the market related to autonomous vehicles. This approach can also help government with drafting policies as this is entirely a new concept and this can be considered as a starting point.

Bibliography

[dat] Sentiment Self-driving Cars - dataset by crowdflower — data.world. https://data.world/crowdflower/sentiment-self-driving-cars. [Accessed 14-02-2024].

[Dob] Saul Dobilas. LSTM Recurrent Neural Networks—How to Teach a Network to Remember the Past — towardsdatascience.com. https://towardsdatascience.com/lstm-recurrent-neural-networks-how-to-teach-a-network-to-remember-the-p [Accessed 14-02-2024].

[EMF] R Evtimov, A Maiwald, and M Falli. Bidirectional encoder representations from transformers (bert).

[ESB$^+$21] Y. El Jariri, F. L. Sefyani, A. Benhida, Z. Benkhaldoun, T. De France, D. Gillet, P. Mathias, K. Kolenberg, K. Chafouai, A. Elhabib, and M. Sabil. Python Interface for Spectroscopic Data Analysis. In K. Kinemuchi, C. Lovekin, H. Neilson, and K. Vivas, editors, *RR Lyrae/Cepheid 2019: Frontiers of Classical Pulsators*, volume 529 of *Astronomical Society of the Pacific Conference Series*, page 321, June 2021.

[Gar20] Neha Garg. Annotated corpus creation for sentiment analysis in code-mixed hindi-english (hinglish) social network data. *Indian Journal of Science and Technology*, 13(40):4216–4224, October 2020.

[GS22] Achal Shankar Gupta and Shilpi Sharma. *Analysis of Public Perception of Autonomous Vehicles Based on Unlabelled Data from Twitter*, page 59–67. Springer Nature Singapore, November 2022.

[kag] Sentiment Analysis with Twitter Data — kaggle.com. https://www.kaggle.com/code/pelinay/sentiment-analysis-with-twitter-data. [Accessed 14-02-2024].

[Kal20] Rohit Kumar Kaliyar. A multi-layer bidirectional transformer encoder for pre-trained word embedding: A survey of bert. In *2020 10th International Conference on Cloud Computing, Data Science Engineering (Confluence)*, pages 336–340, 2020.

[Kos] Yulia Kosarenko. How to Create Decision Trees for Business Rules Analysis. https://why-change.com/2021/11/13/how-to-create-decision-trees-for-business-rules-analysis/. [Accessed 15-02-2024].

[Lau12] Francis Quintal Lauzon. An introduction to deep learning. In *2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA)*, pages 1438–1439, 2012.

[LYZ22] Yifan Lu, Tianle Ye, and Jiali Zheng. Decision tree algorithm in machine learning. In *2022 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)*, pages 1014–1017, 2022.

[NHM$^+$20] Ilja Nastjuk, Bernd Herrenkind, Mauricio Marrone, Alfred Benedikt Brendel, and Lutz M. Kolbe. What drives the acceptance of autonomous driving? an investigation of acceptance factors from an end-user's perspective. *Technological Forecasting and Social Change*, 161:120319, 2020.

[NSS20] Baidya Nath Saha and Apurbalal Senapati. Long short term memory (lstm) based deep learning for sentiment analysis of english and spanish data. In *2020 International Conference on Computational Performance Evaluation (ComPE)*, pages 442–446, 2020.

[Oth21] Kareem Othman. Public acceptance and perception of autonomous vehicles: a comprehensive review. *AI and Ethics*, 1(3):355–387, August 2021.

[Par] Ravindra Parmar. Training Deep Neural Networks — towardsdatascience.com. `https://towardsdatascience.com/training-deep-neural-networks-9fdb1964b964`. [Accessed 14-02-2024].

[RJ15] M V Rajasekhar and Anil Kumar Jaswal. Autonomous vehicles: The future of automobiles. In *2015 IEEE International Transportation Electrification Conference (ITEC)*, pages 1–6, 2015.

[SM19] Ajay Shrestha and Ausif Mahmood. Review of deep learning algorithms and architectures. *IEEE Access*, 7:53040–53065, 2019.

[spi] Logistic Regression: Equation, Assumptions, Types, and Best Practices — spiceworks.com. `https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-logistic-regression/`. [Accessed 14-02-2024].

[SRK21] Soni Singh, K R Ramkumar, and Ashima Kukkar. Machine learning techniques and implementation of different ml algorithms. In *2021 2nd Global Conference for Advancement in Technology (GCAT)*, pages 1–6, 2021.

[SS17] H. Sankar and V. Subramaniyaswamy. Investigating sentiment analysis using machine learning approach. In *2017 International Conference on Intelligent Sustainable Systems (ICISS)*, pages 87–92, 2017.

[Tay23] Mohammad Mustafa Taye. Understanding of machine learning with deep learning: Architectures, workflow, applications and future directions. *Computers*, 12(5):91, April 2023.

[TLT$^+$21] Guillaume Tauzin, Umberto Lupo, Lewis Tunstall, Julian Burella Pérez, Matteo Caorsi, Wojciech Reise, Anibal Medina-Mardones, Alberto Dassatti, and Kathryn Hess. giotto-tda: A topological data analysis toolkit for machine learning and data exploration, 2021.

[WMZ09] Hua Wang, Cuiqin Ma, and Lijuan Zhou. A brief review of machine learning and its application. In *2009 International Conference on Information Engineering and Computer Science*, pages 1–4, 2009.

[YWMW20] Kum Fai Yuen, Yiik Diew Wong, Fei Ma, and Xueqin Wang. The determinants of public acceptance of autonomous vehicles: An innovation diffusion perspective. *Journal of Cleaner Production*, 270:121904, 2020.

[ZHTS19] Xiaonan Zou, Yong Hu, Zhewen Tian, and Kaiyuan Shen. Logistic regression model optimization and case analysis. In *2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)*, pages 135–139, 2019.

List of Figures

Affidavit

I <Pritish Sanjay Samant> herewith declare that I have composed the present paper and work by myself and without use of any other than the cited sources and aids. Sentences or parts of sentences quoted literally are marked as such; other references with regard to the statement and scope are indicated by full details of the publications concerned. The paper and work in the same or similar form has not been submitted to any examination body and has not been published. This paper was not yet, even in part, used in another examination or as a course performance. .
Lippstadt, February 19, 2024

Pritish Sanjay Samant