Evaluating Vehicle Driver Classification: A Supervised Machine Learning Approach Through Comparative Analysis

by

Rudaba Zerin Rafa
15201030
Afrida Azim
16101190
Sowmik Ahmed
12201028
Md. Rayhan Hassan Mahin
20341048

A thesis submitted to
the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science or Computer Science and Engineering

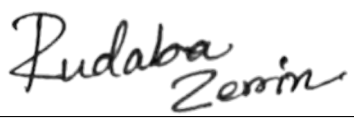
Department of Computer Science and Engineering
BRAC University
September 2020

Declaration

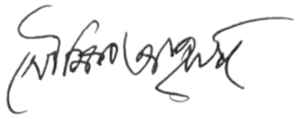
It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

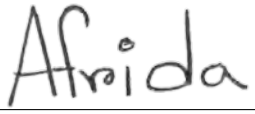
Student's Full Name & Signature:



Rudaba Zerin Rafa
15201030



Sowmik Ahmed
12201028



Afrida Azim
16101190

Md. Rayhan Hassan Mahin
20341048

Approval

The thesis/project titled "Evaluating Vehicle Driver Classification With Comparative Machine Learning Analysis" submitted by

1. Rudaba Zerin Rafa (15201030)
2. Afrida Azim (16101190)
3. Sowmik Ahmed (12201028)
4. Md. Rayhan Hassan Mahin (20341048)

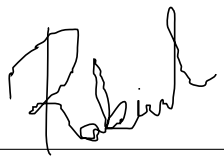
Of Summer, 2020 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on October 08, 2020.

Examining Committee:

Supervisor:
(Member)

Md. Khalilur Rhaman, PhD
Associate Professor
Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)



Md. Golam Rabiul Alam, PhD
Associate Professor
Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Mahbubul Alam Majumdar, PhD
Professor and Dean
School of Data and Sciences
Computer Science and Engineering
BRAC University

Abstract

Contemporary computing and applications of data science chime in unison when it comes to machine learning. Thus the boundaries of learning using AI is open for expansion, aiding those with limited resource and compact scope. Extensively, Supervised Learning has indigenous uses for predictive mapping. Using multi-class vehicle driver data as apparatus, this paper targets to analyze the classification accuracy, precision, predictability, rigidity and extrapolation of existing supervised learning algorithms in case of subjective mechanical data and objective qualitative data. Driver data is compiled focusing on the classical mechanics of driven vehicles and external affecting factors in every driving instance followed by processing for feature selection to split data for training and testing. Analyzing the variance of learning via different supervised machine learning algorithms, a shortlisted precedence of algorithm preference is prepared through comparative exploration. The exploratory extractions are then analyzed to reach optimal extrapolation of vehicle driver classification within the domain of driving style, adjudged distinctly as aggressive, normal and vague. Achieved insights would actively contribute to solidify drivers' conformity towards traffic laws and situational safe driving, aiming to secure a significant fall in road casualties.

Keywords: Comparative Analysis; Decision Tree; Driver Classification; Driver Evaluation; K-Nearest Neighbour, Logistic Regression; Machine Learning; Random Forest; Supervised Learning; Vehicle Driver Classification;

Acknowledgement

All praise goes to almighty Allah for whom we could continue our Research during the Covid-19 pandemic. We are thankful to almighty Allah who kept us safe from this worldwide crisis. Secondly, all guidance provided by our respectable Supervisor Md. Khalilur Rhaman who encouraged us to continue our research and helped us to think analytically. Through the whole research, he was available and provided research materials and helping hands by allocating necessary resources. Artificial intelligence and Machine learning is a mammoth topic but we tried our best to understand and explore the techniques and relevant applications. Thirdly, Thanks to the Computer Science and Engineering department for allowing us to present our work in front of the respected AI and ML jury panel. Last but not the least, special thanks to our parents who supported us in every aspect of our life. Without their motivation, we would not be able to progress and only because of their strong belief in us, we are now going to achieve our Bachelor degree in Computer Science and Engineering.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

AI Artificial Intelligence

BD Bangladesh

$BPWA$ Bangladesh Passengers' Welfare Association

$CART$ Classification and Regression Trees

GDP Gross Domestic Product

ML Machine Learning

NCA Neighborhood Component Analysis

PCA Principal Component Analysis

USD United States Dollar

Chapter 1

Introduction

1.1 The dire scenario of roads in BD

Road accidents is the name of the oldest yet ignored 'epidemic' spreading across the developing world, inflicting severe effects. For BD too, the speedy urbanization and rapidly growing transport and transit systems, road safe has turned to become an essential priority which cannot be avoided. Pointless acceleration, unjust overtaking, poor quality of roads, unfit vehicles, ineligible drivers and lack of road safety awareness is causing crashes and casualties on a regular basis. This is a sustainable development issue as this results in more than 3% loss of GDP in low to middle income nations. The calamity of road traffic accidents costs low and middle-income countries about USD 100 billion annually. In BD, road traffic crashes are a major public health concern resulting in over 21,000 fatalities each year. More than 7000 people have died in road accidents across the country in 2018 as found by a report published the BPWA. Additionally, more than 15000 faces injuries due to crashes that sum up to over 5500 in a calendar year. Reckless intent, unconscionable driving, unfit vehicles, premature or intoxicated drivers and ignorance of rules in reality is causing this onslaught of casualties and mishaps on roads everyday.

1.2 Objectives towards the goal

With a view to minimize these issues, this research is meant to serve as guide to predict and classify a driver and their respective vehicle driving instances by collaborating values of basic vehicular mechanics with external significant factors that provide machine-induced intelligent judgement on a labeled multi-class data model. This,in fact, will create scope for drivers to improve and amend their style by regularly receiving feedback and remarks through a continuous process of data collection and assessment. Furthermore, owners-stakeholders and regulatory personnel can be well-informed by accessing relevant driver data analysis. Additionally, considering public interest, the police can also access the records in cases of abnormal incidents or even accident investigations. Hence, stakeholders, passengers and extensively, the public can finally dream to put an end to the nightmare of road adversity, in a gradual manner.

1.3 Developing AI with ML techniques

Artificial Intelligence which implies that it can make an imitation of a human thinking and perform similar to human brain functionality. More accurately, it implies that machines can be empowered to think like a human without human interference and machines carries the recreation of artificial intelligence. Hence it is a process that it could be intelligence where human want all capabilities to add on machine. There's a misguided judgment that artificial Intelligence may be a system, but it is not a framework. Artificial intelligence implementation is a concept that comes from machine learning. Machine learning is a subset of artificial intelligence that provides some historical data to explore further. Without machine learning, AI can not exist since without machine preparing and it can not prepare human thinking capability in machine. It may be a simple concept machine takes information and learn from information. With all the learning the machine gives a predictions and accuracy similar to human . Artificial intelligence can not exist without machine learning. Machine learning empowers a computer framework to form predictions or take a few choices using historical data without being explicitly programmed. Machine learning employments a enormous amount of structured and semi-structured information so that a machine learning demonstrate can produce accurate result or allow predictions based on that information. Machine learning works with few sorts of historical data which are categorized as supervised, unsupervised and reinforcement learning. Supervised information implies which of the information has past labelled sign where unsupervised implies historical data isn't labelled; mostly worked as taking similar feature and make different clusters. Finally reinforcement learning implies a few of data is labelled and a few are not. With all the learning the machine gives a predictions and accuracy similar to human. So Artificial intelligence can not exist without machine learning. 'Aggressive driving' data has various features based on nature of the features it is named as 'Aggressive' , 'Normal' or 'Vague' so we are able say its predetermined and target attributed is labelled so its lies beneath supervised data. Here machine takes the defined data as inputs and gives the accuracy and prediction. Here we are able say AI usage here our machine gives prediction if it gives more accuracy and prediction at that point it can be said that Strong AI or if a machine learning gives comparatively less accuracy and wrong prediction at that point its weak AI.

Chapter 2

Literature Review

kMC-SVM is a rapid method of pattern recognition which is developed by combining SVM with k-means clustering and is applied to detect a driver's curve negotiating patterns whether moderate or aggressive [44]. It has been found that this method combined method results in lesser training time and additionally, facilitates improved class recognition in comparison with SVM method [44].
Bayesian theory was applied in case of estimating the probability of a driver being normal or aggressive in their vehicle driving. [31]. Moreover, Euclidean distance of the feature vector's pair of elements were calculated to derive the style of driving in case of the test datasets which ultimately resulted in reduced computational cost [31].
GPS, accelerometer, magnetometer, gyroscope, and compass sensors embedded along with the in-vehicle smartphone combined helped to achieve an upgraded system with a top notch domain [26]. By doing this, a better overview and monitoring of road conditions, such as bumps in the road, surface roughness, potholes etc. is unlocked and thus improves the accuracy in case of identifying driver behaviour [26]. Here, road anomalies are identified by the gyroscope which is cross-validated with the accelerometer sensor in order to achieve further confirmation [26]. When this results are collaborated with OBD-cable derived measurement values, the intelligence of the vehicle is enriched [26]. As a result, we can get better accuracy in terms of driving behavior classification [26].
The Bag of Words system is an efficient way to represent data from accelerometer with an aim to improve driver classification [39]. Thus, BoW should be utilizable in cases of intelligent applications such as driving style recognition [39].
SVC based driving-style classification approach can effectively differentiate the variations in individual's driving pattern [30]. This validates the hypothesis that each driver's driving style is inconsistent which varies due to different factors [30].
Experimental results show that the S3VM method performs better than the SVM method when we use a small amount of labeled data and a large amount of unlabeled data [19]. More specifically, this paper demonstrated that S3VM improves classification accuracy by about 10 per cent compared to SVM, in such situations [19].
The driving styles such as aggressive, inattentive, drunk and moderate driving style are compared and analyzed here [14]. It is mentioned that driver behaviour measurements such as DBQ solely rely on integrity of the person and it is suggested that real time ML applications to be used to attain results [14].

A 6 step methodology enables researchers to deal with data attained from sensors, data validation from experts using a data-driven method and unbalanced class distribution by generating an augmented dataset for training purposes, driving-style classification through various classifiers, and evaluation of the classifiers using ML metrics for example, accuracy, F1-score and statistical tests for analysis [45]. This classification of driving styles from data collected was in a naturalistic setting. The authors designed and implemented five computational models such as SVM, ANN, kNN and RF with the intent to determine the best evaluation performance for classification of different driving styles [45]. Resuls showed that SVM provided the best results for classification and identification of drivers; more so in case of aggressive driving [45].

To establish the underlying structure of driving styles, there is a need to acknowledge the driving style which is made up of the driving states proportioned θ where each and every driving state is portrayed as the combination of driving behaviours with proportion φ and secondly, in order to achieve more comprehensive data which included stimulating information from preceding vehicles like speed differential, range and other driving conditions are thereby considered in case of the proposed models for the betterment of recognizing different driving styles [36].

This proposed SVM model can recognize driving style accurately which is based on some trajectory attributes providing 91.7% accuracy rate [37]. This accuracy is much higher than other classifiers such as RF, MLP, and KNN [37]. The result implies that Support Vector Machine is optimal in case of driving style evaluation based on trajectory features [37].

CADSE is a contemporary driving style evaluation system which performs its evaluation by taking input of data from smartphone sensors [20]. Initially, this method puts emphasis on context awareness like traffic states and sensitivity of the care [20]. Next, it evaluates the drivers based on three separate continuous time periods and not just a single period of time [20]. Thirdly, in order to recognize driving styles, this system implements a wide range of experimental and mathematical features [20] Lastly, this system uses different machine learning algorithms for different task assigned to any driving evaluation system [20].

A comparative analysis based on accuracy, precision, recall and time complexity in case of four different classification algorithm was used to detect driving style [33]. Random Forest algorithm had the best performance of identification and solely stood above other algorithms in case of computational speed [33]. When the performance level of sub-classifiers was quite similar, the multi-model ensemble algorithm was dominant in case of improving identification performance of driving styles [33].

Physiological and psychological states are highly considered in case of analyzing driving events and behavior in applications of Machine Learning [41]. In case of analyzing the driver behavior, subjective-based values has been prioritized [41]. Studies implied that subjective vehicular data, environmental external factors and lastly, physiological and psychological input data combined together provides the best results [41]. Specificity, accuracy and recall are used to measure the performance of different ML algorithms in this case [41]. Driving events domains, when analyzed showed accuracy levels ranging from 73% to 98%, 82% to 96% are recall, 84% to 97% is the specificity range [41].

Road parameters such as number of lanes, segment length and ramp type along with traffic density and flow parameters were considered [38]. In case of measur-

ing potential errors among different homogeneous groups of data, the hierarchical model of Poisson-gamma gave better performance than non-hierarchical data models [38].SDCSM was an important feature and was positively correlated with PDO crash frequency meaning that a larger variance of speed genuinely increased the chance of crashes [38].

It has been found that lane width have variable effect of accidents in terms of severity [27]. The condition of pavement was found to be significant in case of accidents without death giving an approximate mean value of–0.0739 in case of inter-states while for state highways the pavement condition was more significantly related with accidents that caused casualties with the result of an approximate mean values of –0.0893 and –0.0926 for state-roads and highways respectively [27].

Research shows, the mean traffic speed and road casualties are strongly correlated [29]. The correlation of speed and road safety is stronger in research papers published after 2000 comparing to older ones [29]. The relationship curve of a driver's vehicular speed and their engagement in accidents in of the same shape as the traffic speed's relation with road safety [29].

In a weight based research, heavier vehicles are found to have lesser injury risk for the subjective driver but possess more threat of injury towards the objective vehicle of lower weight [32]. This implies that the safety index would significantly rise if all the vehicles are heavier [32]. Then again, heaver vehicles do not have much safety-probability in case of singular crash occasion whereas they pose increased threat towards pedestrians and non-motor vehicles like cycles [32]. Overall, the disadvantage of the objective vehicle can be greater than the advantage of the subjective vehicle meaning that it is not conclusive that an increase in average car weight ensures road safety as a whole [32].

For classification, 9 ensemble tree algorithms are used and analyzed comparatively for evaluation of secondary tasks of a driver while driving; these include KNN, SVM, RF, DT etc. [34]. Using unsupervised clustering, first, detecting the secondary task engagement of a driver was prioritized and then further classification of those secondary tasks in a hierarchical method was proposed to derive results [34].

By using hierarchical classification, detailed information and higher accuracy can be achieved at each level rather than single accuracy approach; additionally, using multiple classifiers provide more authentic accuracy than that of single classifier [34]. In this research, Decision Tree classifier gave the best accuracy in detecting secondary task engagement- 99.8% whereas other methods provided results of accuracy ranging from 66% to 96%. In case of identifying the type of secondary tasks, the RF classifier gave best accuracy of 82.2%, outperforming other classifier which gave accuracy between 55% and 79%. The accuracy of segregating different tasks were 77.7% for hand-held phone call, 82.2% for texting on phones and 97.1% for interacting with fellow passenger [34].

Reckless drivers could be identified by traits such as being a young male, easily distracted behind the wheel and sensation-seeking driver [35]. This group can also be considered at risk on the basis of their own self-reported crash involvement and points lost on their driver's license [35]. This reckless group had a higher tendency and frequency of dangerous accelerating on record, implying dangerous driving [35]. This serves as an indicator of risky driving when comprised with time and other mechanical measurements [35].

Chapter 3

Supervised Learning Algorithms

The answer the question of "how" to program computers, it can be out and out explained using Machine Learning. With the help of basic structure and experience gained, Machine Learning differs from statistics in the field of conclusion derivation. However, Machine Learning incorporates further skeptical properties of deducing proper computer architecture and suitable algorithms that can efficiently read, write, index, fetch and merge the data and provided solution to computational tractability effectively. Machine learning is based on development of intelligent software that can connect, interact and sustain the discipline of databases following the principles of statistical learning. The system gets more sophisticated by learning circumstantial change induced by data inputs or external information.

3.1 Supervised Learning

Supervised learning is based on algorithms where the classes are predetermined. Classes are of finite set of human-defined data where certain segments will be labeled in accordance. Machine learning is used to finding patterns among the respective segments and for making mathematical models which in turn would be evaluated according to the capacity of prediction in relation with the variance of the data as a whole. The two primary supervised models are classification models and regression models. Classifiers detect and map input space into the classes whereas regression models map the respective input space into a domain of real-values. The percentage of correct prediction is the prediction accuracy which eventually formulates the classifier's evaluation that is calculated. Among three calculating methods, one technique is to split the training set by using two-thirds for training and the other third for estimating performance. The other widely used technique is known as cross-validation where the training set is divided into mutually exclusive, equal-sized subsets in which the classifier is trained on the union of all the other subsets for each and every subset. The average of the error rate of each subset is therefore an estimate of the error rate of the classifier.
Algorithms with high inclination or bias produce less difficult models and refrain from overfit cases. Nonetheless, they fail to comprehend relevant details in the feature space. Conversely, algorithms with high variance produce complex functions that can portray the training dataset well and capture those details. However this sensitivity can create overfit. In expansion, a high variance is a direct impact of data in high dimensional space. Consequently, a large number of training samples

are essential so as to diminish the variance and misclassification[18].

3.2 Decision Tree

Decision Tree is a predictor that predicts the labels that are associated with instances by traveling from a root node towards the leaf of a tree. While going from root to leaf, in case of each node, the successor child is chosen by splitting the input space. This splitting is based on one of the featured of the node or by a set of predefined splitting rules. A leaf contains a specified label. Each leaf is assigned to a particular class that represents the most fitting target label. Also, a leaf might hold a vector representing the probabilty which would indicate the probability of the target label holding a certain value. Continuing on the path, instances are classified by iterating them from root to a leaf down the tree. This is done according to the outcome of the tests derived from traversal. Decision tree inducers are basically algorithms which can automatically construct a decision tree from any given dataset. Here, the goal is to find the optimal decision tree by minimization of the generalization error.

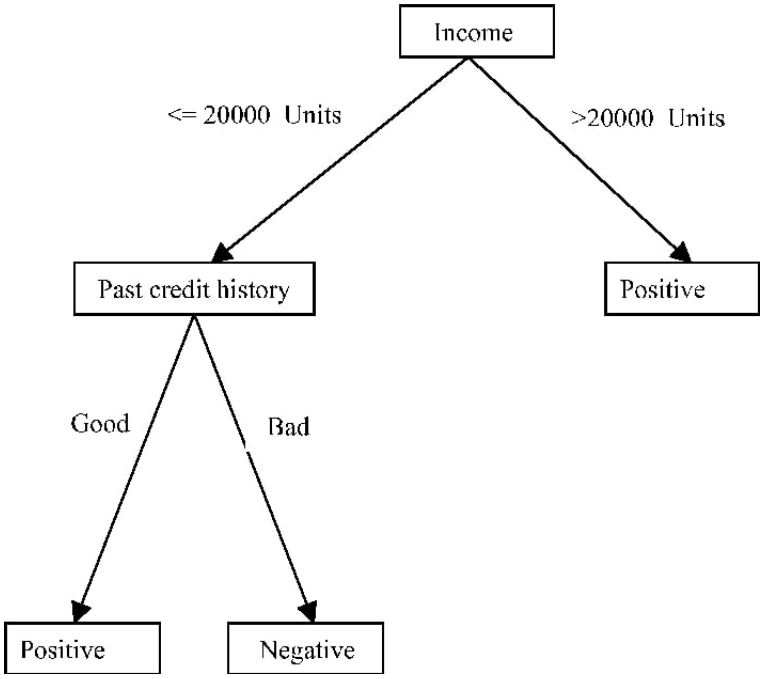


Figure 3.1: An example of Decision Tree[7]

The information gain is the measure utilized by the ID3 algorithm in request to choose the most beneficial attribute for separating from other attributes. Mutual Information or information gain targets decreasing the uncertainty of the information. In the event that ID3 utilizes information gain as a attribute selection method, C4.5 executes in accordance to gain ratio, which comprises in division of mutual information by the entropy of the tried attribute (the "split information" measure). The gain ration is hence no more than an augmentation of the information gain, which targets lessening its bias by restricting the expansion of the tree through penalization of variables that have numerous modalities[21].
Bootstrap Aggregating is an ensemble learning method designed in order to increase prediction-accuracy of other learning algorithms. This method is a technique that

combines several ML algorithms to obtain more accurate and static prediction than other alogithms working alone. Bagging can be applied to all kinds of classification algorithms but it is particularly useful for decision trees[25].
A critical point to be noted, decision tree doesn't back track for any kind of entropy alteration. Each property of our dataset is compared with the root node in each cycle and the tree encourages parts. Decision tree is based on each iteration of unused attribute with divide. Further attribute determination strategies are picked by a few analysts. It's found that there are nearly seven criteria presented; counting chi – square , gini index , entropy for information gain and so on.
CART used gini index in case of classification processing and splits the dataset intro a decision tree. We have used gini index approach to classify data.
The Gini index is defined by

$$G = \sum_{K=1}^{K} \hat{P_{mk}}(1 - \hat{P_{mk}})$$

3.3 k-Nearest Neighbor (kNN)

KNN or K-nearest neighbor is one kind of lethargic learning method based on instances. The classifier or model isn't worked during the time of training and results in a high cost and time-consumption during the period of grouping. Just training attributes along with their labels are put away at training time and classifier is work at the classification period for each test instance. It is a non-parametric technique for characterizing obscure data; for example it doesn't make any suspicions on underlying data distribution. An instance is represented by characteristic vector $a_1, a_2, ...a_n$, where a_i denotes the value of i^{th} attribute A_i of x [12]. Because of the basic simplicity, efficiency and easy-to-implement trait, KNN is a broadly used classification algorithm in the ML world. It is one of the top ten algorithms for data mining and has been widely implemented in different fields. KNN has a few limitations that impair its classification accuracy. As well as high time complexity, it has significant memory requirements[12].
k-Nearest Neighbor is based on the principle of close proximity of instances among the dataset that possess similar properties. This suggests that an unclassified instance can be determined by observing the value of nearby neighboring instances that are tagged with classification labels. The kNN algorithm locates the k nearest instances to the given query instance and thus decides the class by observing and critically identifying the most frequent class label. It is non-parametric due to its growth in the training sample and measures K most similar data distance. Usually it takes similar features to predict test data points and takes a distance measurement from training data and new data points. The number of K basically tells how numerous neighbors considered for similitude. Vital key point about KNN is it does more computation on testing tests.

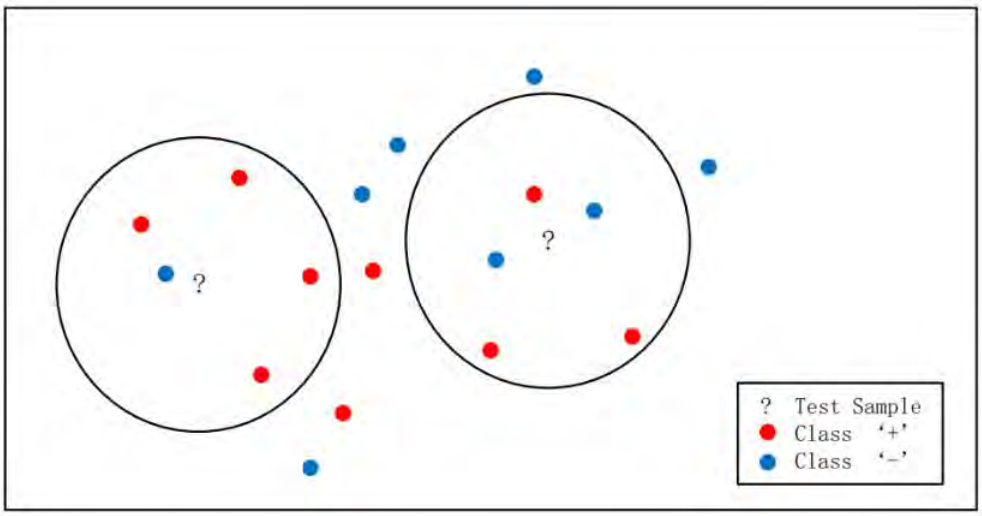


Figure 3.2: An example of KNN classification task with k=5[11]

Picking the value of K is fundamental so as to lessen misclassification. An enormous K value diminishes the variance cause by noisy data. However, it delivers a bias that overlooks smaller patterns that can be helpful for the correctness of classification[18]. Manhattan distance calculation is one approach of kNN.

$$Manhattan : D(x, y) = \sum_{i=1}^{m} |x_i - y_i|$$

3.4 Random Forest

RF is one kind of classifier that works on supervised models, quite similar to building decision trees. It comprises of a collection of decision trees in order to solve problems related to classification and regression [23]. A random forest is a classifier comprising of an assortment of decision trees, where each tree is built by applying an algorithm "A" on the training set "S" along with an additional random vector denoted by θ, where θ is examined from some distribution. The prediction of the random forest is acquired by a dominant part vote over the predictions of the individual trees. To indicate a specific random forest, we have to characterize the algorithm A and the dispersion over θ. There are numerous approaches to do this and here we depict one specific choice. We create θ as follows. In the first place, a random sub-sample is taken from S with substitutions, to be specific we test another training set S' of size m' utilizing the uniform dispersion over S. Secondly, we build a group sequentially I_1,I_2,..., where every I_t is a subset [d] of size k, which is created by inspecting consistently at random value from [d]. These different random factors structure the vector θ. At that point the algorithm A grows a decision tree (eg, utilizing the ID3 algorithm) in light of the example S', where at each splitting phase of the calculation the algorithm is confined to picking a feature that boosts the obtained gained from I. Then again, if k is negligible this limitation may forestall over-fitting. Another model is random split selection where at each node the split is picked capriciously from among the K best parts. It produces new training sets by randomizing the yields in the first training set. Another methodology is to choose the training set from a random set of weights on the models in the training set. There has been

composed various papers on "the random subspace" strategy which does a random determination of a subset of features to use to develop each tree. It is suggested to characterize enormous number of geometric highlights and search over a random selection of these for the prime split at every node.
The normal component in these methods is that for the k^{th} tree, a random vector Θ_k is produced, independent of the past random vectors Θ_1, ..., Θ_{k-1}. However, with a similar appropriation, a tree is developed utilizing the training set and Θ_k, resulting in a classifier $h(x, \Theta_k)$ where x is an input vector. For Example, in bagging the random vector Θ is created as the counts in N boxes coming about because of N darts thrown aimlessly at the boxes, where N is the quantity of models in the training set. Random split selection Θ comprises of a number of autonomous random integers between 1 and K. The nature and dimensionality of relies upon its utilization in tree construction. After a big number of trees are created, they vote in favor of the most popular class. We call these techniques random forest. All in all, Random Forest appears as an extended approach of decision trees. It makes a huge number of decision trees afterward work as gathering. This process is called ensemble. This ensemble method improves the predictive model performance. The process started with creating a bootstrap dataset by picking a random sample from the original dataset note that one sample can repeat in this bootstrapping process. After making a bootstrap dataset based on it numerous decision trees are made. At that point it'll get the forecast result from each choice tree, then in this step, voting will be performed for each anticipated result. The ultimate result is considered by most voted predictions as last.

3.5 Stochastic Gradient Descent (SGD)

There are two normal approaches to IPS or interior-point solution. 1. Barrier method and 2. Primal-dual interior point method. But despite the fact that it is more streamlined variant for the first issue, it is exceptionally normal in the AI issues. Gradient Descent which is also known as full-batch GD algorithm is basiccaly an actualized algorithm for the issues that are unconstrained [28].
Stochastic gradient based techniques, on the other hand, work with gradient estimations acquired from little subsamples (mini-batches) of training set. This can extraordinarily diminish computational necessities: on huge, excess dataset, basic stochastic gradient descent regularly passes advanced second-order batch method by order of magnitude.

$$J_t(w) = \lambda\Omega(w) + \frac{1}{k}\sum_{i=1}^{k} l(w, x_i^t, y_i^t)$$

Updates on SGD take the form:

$$w_{t+1} = w_t - \eta_t \nabla J_t(w_t)$$

There are a few methods we can look into such as mini batch, ADAM, momentum and variance-reduced mechanism. They possess different and unique qualities, favorable circumstances and deficiencies [28]. Mini-batch: All things considered, rather than playing out a perimeter update for each single training date, for the most part, we use the SGD for each mini-batch of respective training data.

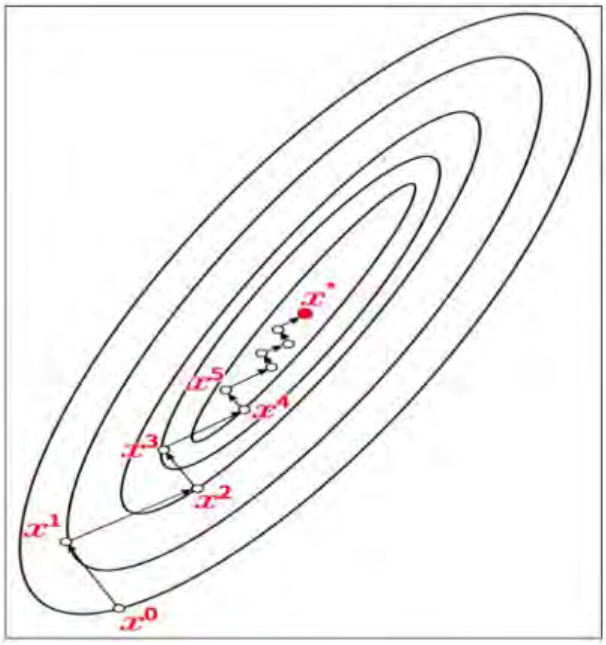


Figure 3.3: The searching path of SGD

Momentum: At the point when curves of the surface are steeper in one dimension than others, this technique has horrible performance since it would perform oscillations up and over the slopes. A basic method to defeat the shortcoming is to present a momentum term into the update iteration. Momentum reproduces the idea of inertia as per the laws of physics. This implies that in each iteration, the process of updating isn't just complied with the GD but also maintains a strong relationship with the direction of the last update of iteration, referring to the momentum.
AdaGrad: This can adjust the pace of training as per the dissimilar significant variables, and perform bigger updates for the frequent and smaller updates for frequent parameters. AdaDelta works specially well for sparse gradients.
SVRG: As of late, a few researchers find that SGD has slower convergance rate due to the inherent variance of data. SVRG has linear convergence outcomes in case of smooth and strongly convex loss. At each instance of SVRG, it is advised to keep an adaptation of estimated value as that is near the optimal x.

3.6 Support Vector Machine (SVM)

This algorithm rotates around the idea of a margin- either side of a hyperplane that differentiates two data classes. Maximizing the margin and in this way making the biggest conceivable separation between the isolating hyperplane and the occurrences on either side of it has been demonstrated to diminish an upper bound on the expected generalization error.
SVM segregates the input dataset into a sub-set of the training data which are called the Support Vectors. This current SVs are ideally separated between the closest members from separate classes by utilizing the function known as Kernels. Empirical Risk Minimization (ERM) is the method that minimizes training data's error. On the other hand, SRM limits the roof value of the anticipated risk. This is the reason SRM performs better since it has the capacity to generalize[23].

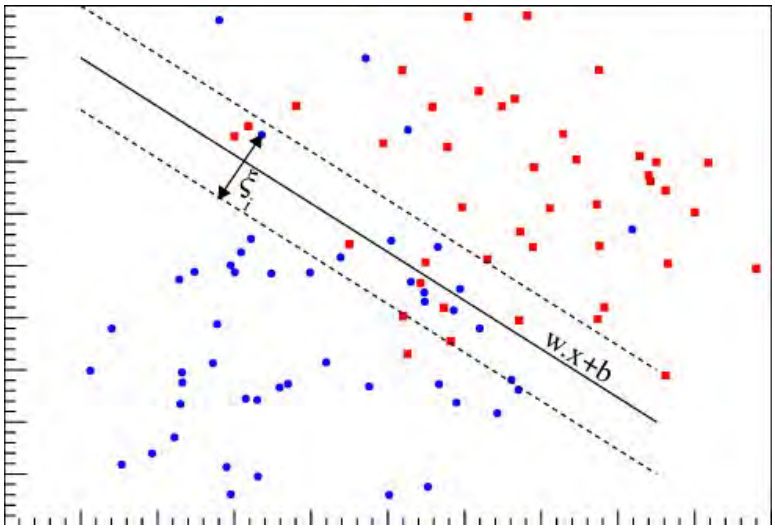


Figure 3.4: Soft margin SVM showing the (solid) maximal margin hyper-plane and (dashed) functional margins for two categories of events (dots and squares) in two dimensions[17]

On account of linearly separable data, when the ideal separating hyperplane is discovered, the points of data that are found on the hyperplane's margin are known as support vector points and a linear combination of these points are perceived as the solution. If the training data is considered as linearly-separable, (w, b) exists such that

$$w^T x_i + b \geq 1, x_i \in P$$
$$w^T x_i + b \leq 1, x_i \in N$$

by rule of

$$f_{w,b}(x) = sgn(w^T x + b)$$

where w is weight vector and b is bias.
When it's conceivable to linearly isolate two classes, an ideal separating hyperplane can be found by minimizing the Squared Norm of the separating hyperplane. This minimization can be set up as a convex quadratic programming (QP) issue.

$$Minimize_{w,b}\phi(w) = \frac{1}{2}(\|w\|)^2$$

subject to

$$y_i(w^r x = i + b) \geq 1, i = 1, \ldots, l$$

Support vector classifier takes a number of inputs from features and makes a hyperplane to classify. Originally support vector machine works for binary classification but as having three target attributes this entire handle lies in multinomial. Here we have three targets so preparing information made by three classifiers demonstrates is produced. After giving test samples, it finds the regarded coordinate prepared at that point after it appears positive and gives the probability of positive reaction.
Dual of the square hinge loss function-

$$min_{w,b,\xi}\frac{1}{2}\|w\|^2 + \frac{C}{m}\sum_{i=1}^{m}\xi_i^2$$

subject to

$$y_i((w, x_i) + b) \geq 1 - \xi_i$$

for all i and

$$\xi_i \geq 0$$

For classification problems, the convergence characteristic of w_C as $C \to \infty$. For regression we a theorem that implies- Consider any nonnegative and convex loss function. If $W_\infty \neq \emptyset$ then

$$lim_{C \to \infty} w_C = w_\infty$$

where

$$w_\infty = \arg min_{w \in W_\infty} \|w\|^2$$

If L2 loss is used, then $W_\infty \neq \emptyset$ This is due to the fact that w_∞ is a model without performing regularization hence overfitting tends to happen and also the performance faces decline. However, it is not easy to identify a C_{max} so that if $C \geq C_{max}$ then the model is close enough to w_∞ [42].
A popular approach to solving k-class pattern recognition problems is to consider the problem as a collection of binary classification problems. K-classifiers can be built, one for each class. The hyperplane can be constructed based on majority. Alternatively, $(k(k-1)/2)$ hyperplanes can be put to use along with some native voted approach. In the Support Vector literature, one-against-all method has been widely utilized to solve multi-class problems[2].

3.7 Naive Bayes

The Bayesian classification is another technique for the supervised learning method just as the statistical method for classification. It accepts a hidden probabilistic model and it permits catching uncertainty about the model in a principled manner by deciding probabilities of the results. The fundamental motivation behind the Bayesian classification is that it can take care of predictive problems [43].
Naive Bayes depends on one or more conditional probability which is intended for use in supervised induction assignments. It's an extraordinary type of Bayesian network depending on two suppositions. Here, the attributes are considered as independent irrespective of their classes which implies that no covered up characteristic impacts the model of expectation. Despite the fact that in real life it is hard to locate this sort of independent data, they perform truly well and scales with the given Dataset. NB is distinctive when it comes to treating discrete and numerical features. Naive Bayes gives a basic and proficient methodology. Since it expects that numeric values comply with a Gaussian Distribution, a few domains may not function admirably[23]. Typically the CP function is thought to me multinomial.
This sort of Bayesian Network is known as a Bayesian Multinomial Network or BMN. It works with discrete variables only and accordingly, if there is a constant variable then is must be put into discrete form, with a resulting loss of information. There are multiple BMNs: Naive Bayes, k-dependent classifier, tree-augmented Bayesian Network, and semi Naive Bayes. Within the sight of continuous variables, another option is to accept that continuous variables are obtained from a likely Gaussian distribution. The aforementioned type of network is known as conditional Gaussian network (CGN). It can manage both discrete and other continuous variables and,

thusly, it is a choice to work with mixed variables without the need to make discrete of the constant ones. An auxiliary limitation of this network is that a discrete feature cannot have continuous parent nodes. Despite the fact that the Gaussian estimation function for variables that are continuous is extremely solid, it provides a sensible guess to some practical and real-life distributions[5].

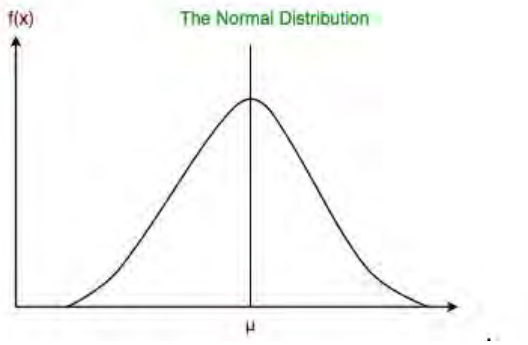


Figure 3.5: Gaussian Normal Distribution

This classification gives functional learning algorithm and can consolidate observed data. Bayesian classification gives helpful viewpoint to comprehension and assessing learning algorithms. It computes clear probabilities for hypothesis and it robust the noise in input data.
We consider an overall probability distribution of two values $P(\mathrm{x}_1, x_2)$.
By utilizing Bayes' rule, without loss of generality, we get this equation:

$$P(x_1, x_2) = P(x_1 \mid x_2)P(x_2)$$

Naïve Bayes is simple classification beneath supervised learning and regularly it takes after Bayes hypothesis which uses predictive analysis through probability. The main concept Bayes theorem is conditional probability. Conditional probability is the likelihood of one thing being true given that another thing is true

3.8 Logistic Regression

This is a predictive technique under machine learning and utilized when the target variable is categorical. As per rule, it calculates the likelihood of the target variable. Similar to NB, logistic regression [6] works by separating some set of weighted attributes from the input and joining them linearly, which implies that each feature is multiplied by a particular weight and then summed up.
The most significant distinction between NB and logistic regression is that the logistic regression is a biased and discrimination oriented classifier while the NB is a generative classifier.
The logistic regression hypothesis is

$$h_\theta(x) = g(\theta^T x)$$

Where function g is sigmoid function which is defined as

$$g(z) = \frac{1}{1 + e^{-}z}$$

Also, the cost function is defined as

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}[-y^i)\log(h_\theta(x^i)) - (1 - y^i\log(1 - h_\theta(x^i))]$$

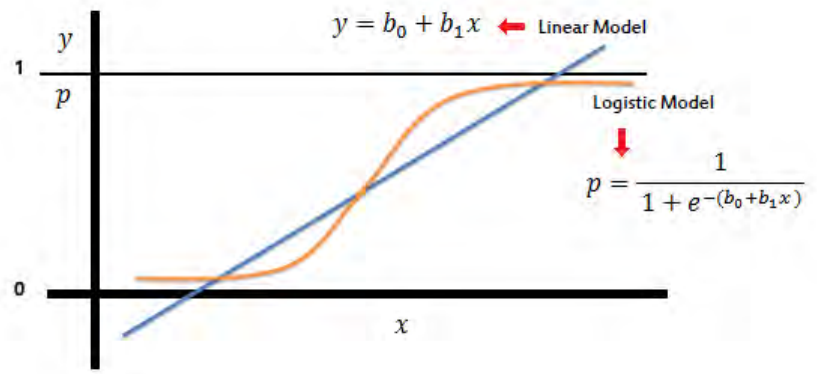


Figure 3.6: The logistic model

Multinomial calculated relapse (MNL) is normally directed utilizing greatest probability assessment, which is an iterative strategy. The primary cycle (called cycle zero) is the log probability of the invalid or void model; that is, a model without any indicators. At the following cycle, the indicators are remembered for the model. At every cycle, the log probability diminishes as the objective is to limit the log probability. At the point when the distinction between progressive cycles is exceptionally little, the model is said to have met, the emphasizing stops, and the last log probability (LR) measurement is figured[16].
The impact of any autonomous variable on the result can be tried utilizing the probability proportion (LR) measurement test. The invalid theory of this test is that the in- subordinate factors don't influence the needy variable. The invalid model is determined by getting the log probability of the perceptions with simply the reaction variable in the model from emphasis zero (for example model with block alone). The last fitted model is determined by getting the log probability of perceptions with all the autonomous factors in the model from the last emphasis after union[16].
To conclude, logistic regression is a kind of regression that predicts the probability of event of an occasion by fitting data to a logistic function [6]. Similarly the same number of type of regression examination, logistic regression utilizes a few predictor variables that might be numerical or categorical.

Chapter 4

Methodology

4.1 Description of dataset

We have gathered the information from https://www.kaggle.com/veeralakrishna/aggressive-driving-data. The data represented here are part for an examination investigating driver's expertise dependent on whether the driver confronted any impact with the preceding vehicle or not.
The collected data was separated into three categories. A structure was created to predict the driving style of a particular driver into "Aggressive", "Normal" or "Vague".
In the dataset these attributes were defined into numerical values. These categories were defined based on the length, speed, weight of both the vehicle of the particular driver himself and the preceding vehicle. Also there are all other attributes such as lane of the road that vehicle is moving, time gap between preceding vehicles.
There are some encompassing components that are referenced in that dataset which impacts on the driving style such as air temperature, precipitation type(clear, rain, snow), wind direction, wind speed, condition of road based on the weather(dry wet, visible track, snow covered) and lighting condition based on the time of the day(daylight, night, twilight). Carsten and Hibberd (2020) appeared in their study that speed limit ought to be controlled by street design and roadside department [46].
According to Das and Datta(2020), the traditional method of partner the impact of a solitary factor on the reaction variable isn't sufficient to describe the unpredictable idea of an accident event. As crash information contains a huge rundown of contributing variables, there was an issue of 'A larger number of highlights than data point' in tackling the research issue[40], this investigation helped to decide the impact of weather and street condition is important to assess driving style.

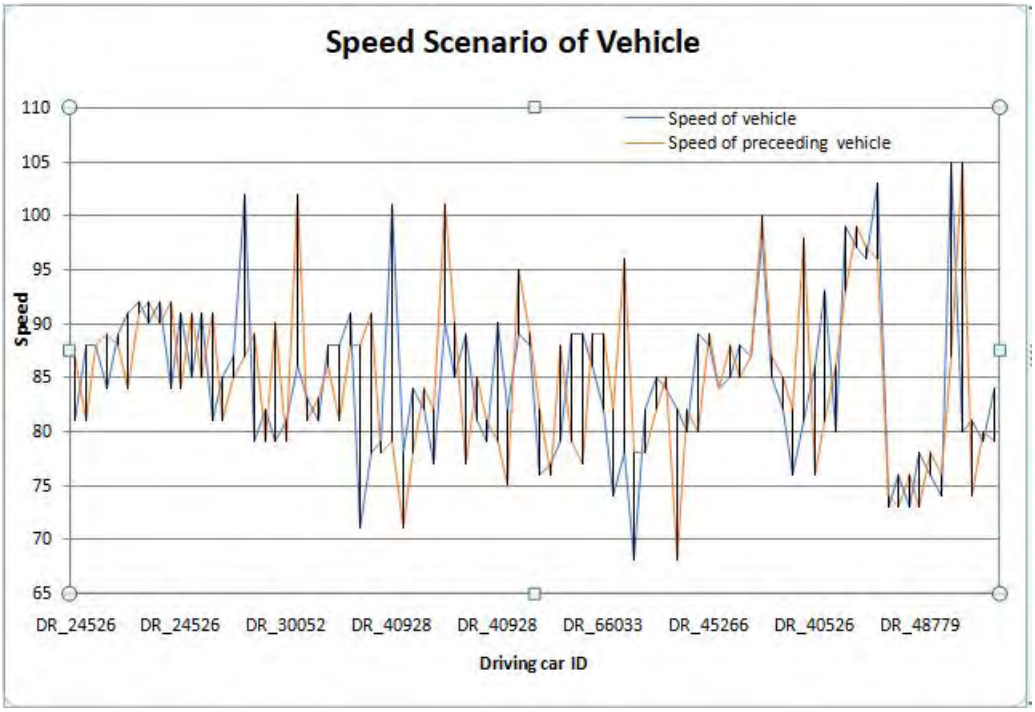


Figure 4.1: Speed Scenario of subjective vehicles and preceding vehicles

A research done by Jackett and Frith (2013) has built up a field technique to inspect the impact of street lighting on metropolitan night time crashes. They set up normal street luminance as the key portion reaction variable in crashes and recognized how various accident subsets can be affected by street lighting [10]. As indicated by the car accident information gathered in 2013-2017 from Louisiana Department of Transportation and Development (LADOTD) shows that around 55% of the accidents occurred in obscurity with various lighting conditions (Dark no light-28%, Dark light-17%, Dark light at Int-10% and at dusk and dawn 4%) [40].
A study done by Chakraborty and Gupta (2013) dealt with perceiving the example of driver's conduct during various weather conditions under realistic simulation. There they find the accident frequency within drivers up to 2years driving experience having 13.64% in clear, 15.91% in rainy and 14.29% in the foggy situation. Besides during foggy climates, high encountered driver's accident recurrence was higher [9]. Numerous specialists contemplated the effect of weather conditions on traffic well-

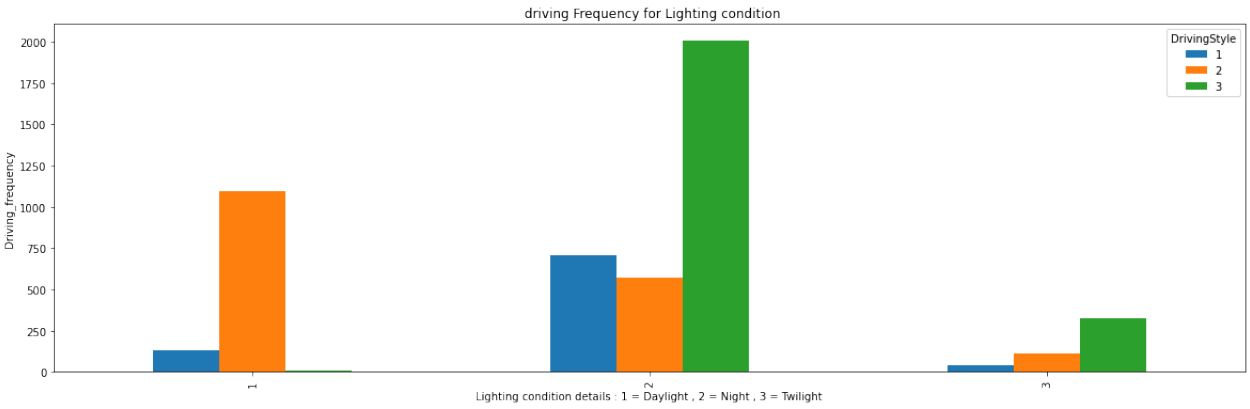


Figure 4.2: Lighting Condition

being in Canada in 1993, an investigation on rainfall related accidents in Calgary and Edmonton Cities reasoned that crash hazard is 70% higher in rainfall occasions than the accident danger in clear weather occasions [1]. These show the connection between driving examples to the weather and street condition.

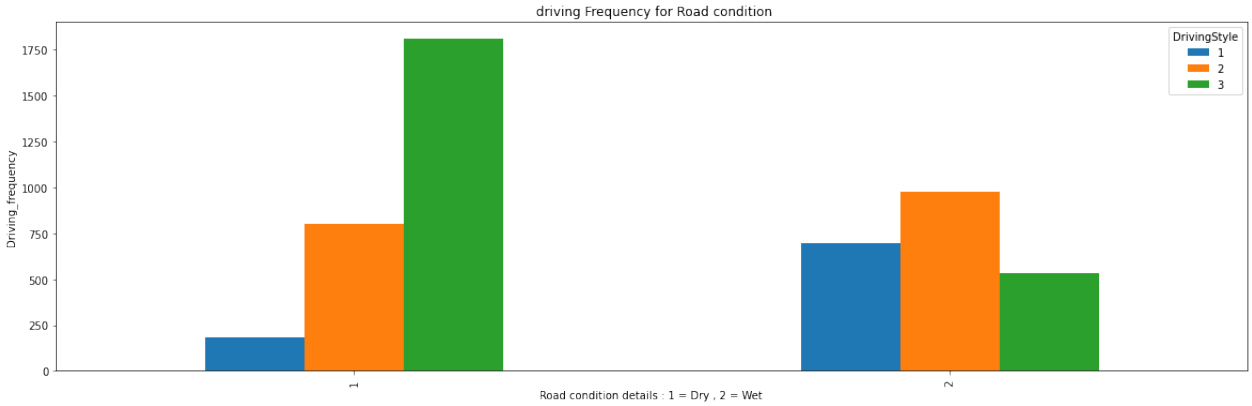


Figure 4.3: Road Condition due to Weather

4.2 Workflow

This is the workflow by which we have progressed with the data and applied machine learning techniques.

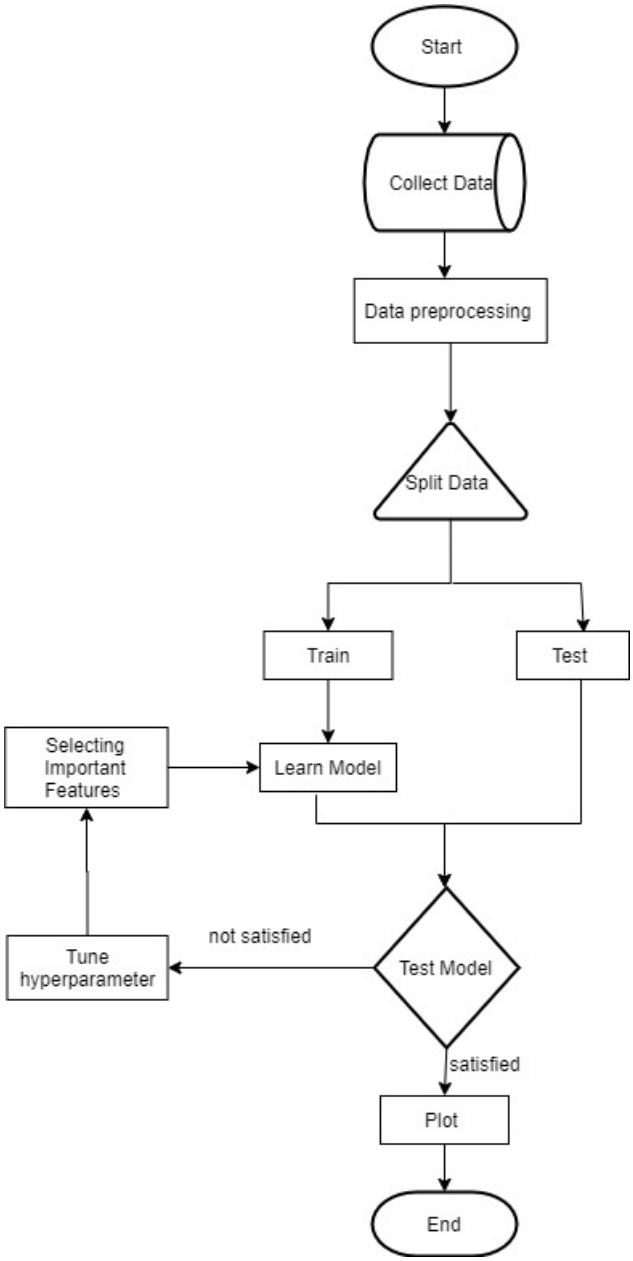


Figure 4.4: Workflow

4.3 Data pre-processing

The data we have collected has been handled through certain means before sending them through algorithms. Kotsiantis, Kanellopoulos and Pintelas(2006) stated that the data pre-processing effects on the execution of a supervised ML algorithm. In the event that there is a lot of unessential and repetitive data present or blaring and inconsistent information, at that point information disclosure during the training

stage is more troublesome [4]. In this project data normalization, feature selection and correlation matrix procedure to process the supervised data set.

4.4 Data type conversion

In our dataset there were several features which were represented as categorical data form such as Weather Detail (precipitation type), Condition of the road, Lighting Condition. Python cannot work with different data types, every data type needs to be the same for the evaluation process. For this purpose we have converted every data into one specific type dataset for improving correlation between data. All the data were converted into numerical form, specifically into integer and then we proceeded into the next stage.

4.5 Data Normalization

Normalization is a data pre-processing stage where data is scaled to required intervals [24]. Data scaling or normalization scales the data to study the considered insights and valuable connections. Here we have used min-max normalization where it changes the value of data in between 0 to 1. From [15],

$$Z = \frac{x - min(x)}{max(x) - min(x)}$$

4.6 Data Standardization

Standardization is a feature scaling technique where values are standard around the mean and with unit standard deviation. Standardization is a useful technique where data follows a Gaussian distribution. This technique is used when features of input dataset has large difference between their range. Any algorithm that tries to maximize the distance between the plane, if one feature has larger value than the other, it might try to dominate over the other feature. This is where Standardization can solve the complexity by giving all features the same impact on distance metric. It is also known as z-score normalization [24].

$$z = \frac{(value - min)}{standard\ deviation}$$

4.7 Feature Selection

Feature subset selection is the way toward recognizing and eliminating as much irrelevant and repetitive features as could be expected. This reduces the dimensionality of the data and enables learning algorithms to operate quicker and more viable [4]. In our algorithm we have used SelectKBest() method which selects features according to the k highest scores. We have used "score_func", "k", "mutual_info_classif" as parameters. The value for 'score_func' is 'f_classif", which is a default value and computes the ANOVA F-value for the provided sample. We used "k='all'" which indicates avoiding selection, for use in a parameter search. Furthermore we have

used mutual_info_classif which estimates mutual information for an individual target variable. To remove low variance of column we called VarianceThreshold and set the value of the parameter at $(.8\times(1-.8))$[8].

4.8 Correlation Matrix

Correlation matrix is a table which substantially shows the relationship among the data in a certain dataset. This helps us to understand the connection of attributes with the target variable that we use for prediction [22]. Here we created a correlation matrix with the attributes and worked on correlation analysis of this dataset. Figure of correlation matrix

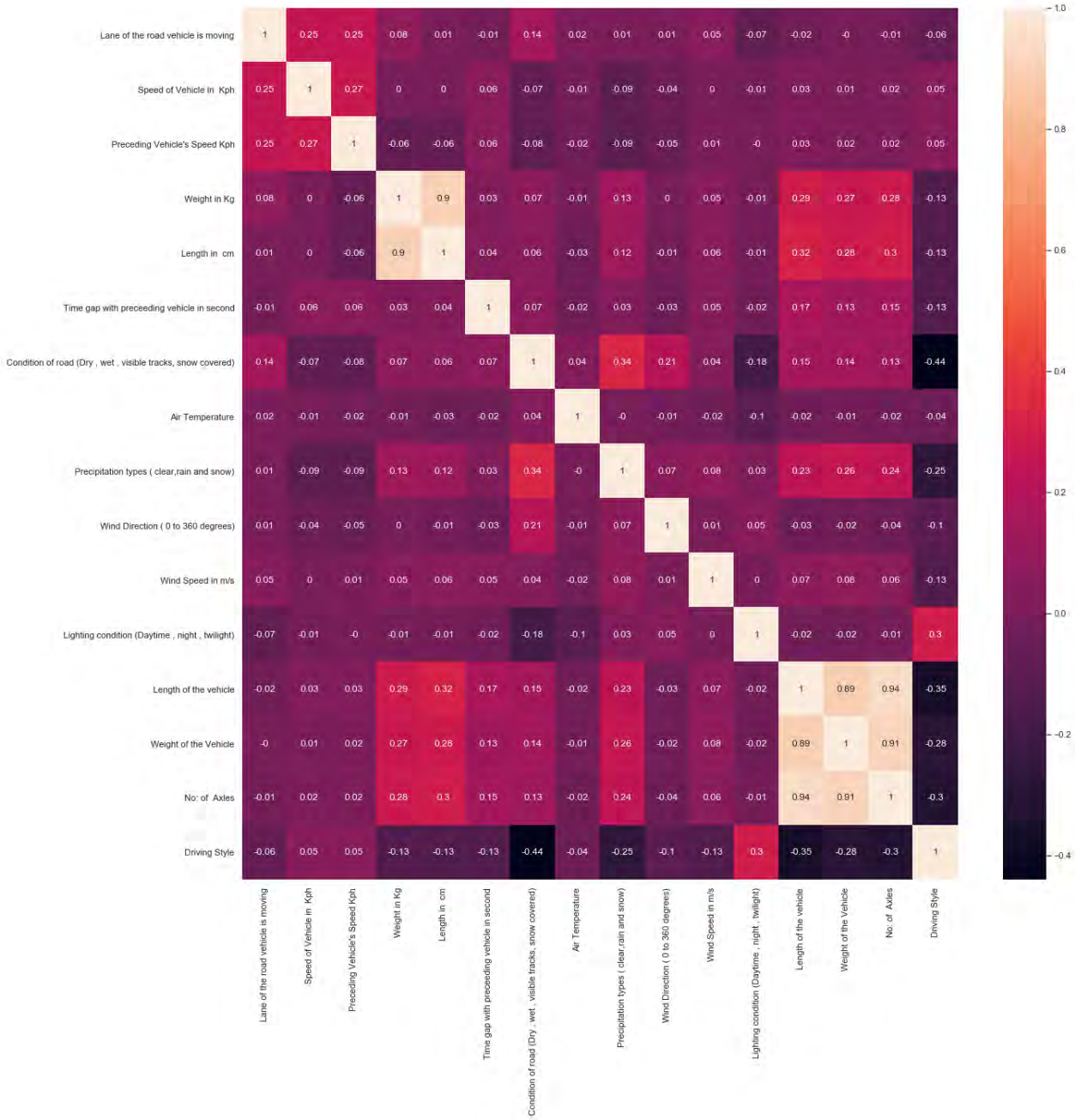


Figure 4.5: Co relation Matrix

Here we can see with the column Driving Style; there is a strong relation between Speed of the vehicle, Speed of the preceding vehicle and the Lighting Condition based on the time of the day (Daylight, Night and Twilight).

4.9 Data Splitting

Data splitting is a strategy to assess performance of machine learning. This technique takes the dataset and divides it into two groups, training and testing. Training dataset is the subset of data used to train the model and testing dataset is the set of data that is used to evaluate the model. The goal is to estimate the functioning of the machine learning model. Here in this study, we have set up training dataset to 70% and testing dataset to 30%.

Chapter 5

Implementation and Results

In spite of the fact that there are three strategies accessible for normalizing information Z score normalization, Decimal Scaling and min max normalization. Z score normalization turned suitable for some algorithms.
After all data is cleaned, like converting string column to float and string to integer, data is finally ready to jump into Algorithms using Scikit learn library. For our research, we chose the common supervised machine learning algorithms- Logistics regression, Decision Tree, Naïve Bayes, Random Forest, Support Vector Classifier and K-Nearest Neighbor.

5.1 Decision Tree

Executing the decision tree ,we set our parameters as "criterion= gini","splitter=best", "min-samples-split=2", which gave accuracy of 94.5%. Gini index or gini impurity is calculated for information gain of our classification problem which ensures best distribution of nodes for splitting. Another point for utilizing the gini list for decision trees since our objective with our dataset is .Analysts indicated that Gini Record suites as it were the yield or target variable is categorical. For each node 'splitter=best' whereas separating each node the Decision tree classifier considers best value whereas advance part. Here we did not utilize a pruning strategy since all the branches utilized is fundamental for a way better result still after giving a few pruning methods with 'ccp-alpha=1' the accuracy altogether diminished in 46.66%. The reason behind doing pruning is that 94.5% accuracy may look better accuracy but it might cause over-fit data as applying the pruning method reduces accuracy too much. We adjusted 'max-depth=7'; this process may handle the over-fitting problem . After all the adjustments we got 88.26% as final accuracy.

5.2 K-Nearest Neighbor

The main challenge was selecting the K value because from the startup we were unsure about which value may lead to better results so plotting an increasing number of K values against its error rate.
Here between range 0 to 5 shows less error value. As the curve starts to increase the error rate goes high. So we took the K value in between 0 to 5 which is 3.

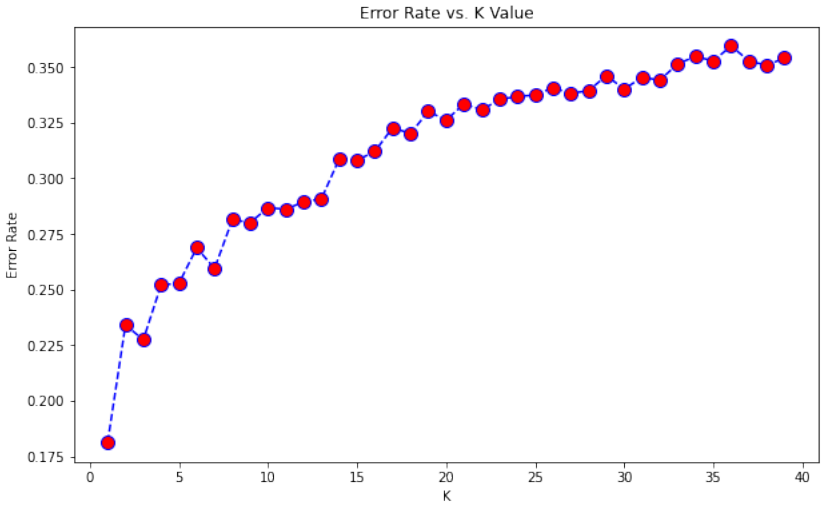


Figure 5.1: K value corresponding with error rate

Working on our dataset we adjusted parameters to improve accuracy which standard fit our predictive model; "n-neighbors=3" , "p=1", "weight=distance". Here three closest focuses are considered when calculating with Manhattan distance in spite of the other two conditions for estimation is Euclidean and Minkowski. Our optimized result appeared 86.0% with application of Neighbors component analysis. Whereas putting 'weight=uniform' implies accuracy 77.26% reason behind that it is found that our dataset has a few important features here and is able to assign higher weights. For this reason if assigned 'weight=uniform' that means giving equal distance and dominance of all neighbors which resulted in less accuracy. Where setting the value of K in "n-neighbours=3" accuracy diminished in 71% without neighbors component analysis method. If K value is increased more the more we will get less accuracy because the higher K value means consideration of more noise of data and leading to less accuracy. Here we used Manhattan distance measurement by 'p=1' because it lead to better accuracy .Relating to our dataset our target yield is categorical as well as a few include like street condition- "dry", "wet" and so on. In spite of the fact that best accuracy recorded by altering parameter, calculating distance by Euclidean, KNN classifier result dropped in 78.88% Manhattan distance is primarily calculated on real vector where calculating distance like squared way here our dataset gave better accuracy. So we will accept that our data points are not hypotenuse like Euclidean or maybe like distance is like grid ways.

5.3 Naïve Bayes

We have utilized Gaussian and Multinomial Naïve Bayes both gave exactness like 69.9% and 47.2% which is very poor to consider though making adjustments in hyper parameters. In two inquiries Gaussian naïve bayes gave a small way better result since continuous values associated with each feature are assumed to be distributed according to a Gaussian distribution. It gives a bell formed curve where values are about the mean of the point values as made clear[6].
Moving on the multinomial without experiencing Bayes accuracy because our knowledge unit has the highest correlation with each point like speed is dependent with

weight and length of the vehicle and other external instances like road condition, light condition, wind direction which affects the target given properties. Uncommonly Multinomial utilized to know the number of occurrences of a word in a given situation. We have considered Bernoulli Naïve bayes and result came about 47.8%. Bernoulli works well with the situation of the dataset feature like Positive or Negative, Victory or disappointment or any twofold nature at that point Bernoulli suits perfectly but as our dataset don't contain any double possibility so Bernoulli seems not fit effectively with higher accuracy. In common Naïve Bayes works best when dataset attributes are free from each other but our dataset is each attribute are exceedingly related to each other which lead to changed poor result.

5.4 Random Forest

We utilized a Random Forest classifier utilizing parameters like "bootstrap=True"; "max-depth=7"; come about 89.27% accuracy. Here bootstrap tests are connected when building decision trees and at least two tests for decision trees with hundred trees in forest with each tree depth is seven. Expanding the tree more means increasing max depth, it seems that more information is captured in the tree.. For this reason after applying "max-depth=8", accuracy hopped into 93.30%. Here is the point that very high amount accuracy may increase confusion about over-fitting so we finalized accuracy as 89.27%.

5.5 Logistic Regression

As we have three target variables our handle lies on Multinomial Logistics regression. Logistic regression mainly works with a logic function which could be a logarithm work which confines probability between 0 and 1. Here we have a few sets of features Speed, weight , length and so on from our dataset which is an autonomous variable X and Y is the subordinate variable which is calculated by summation of the set of x with its comparing coefficient from each X variable.
Estimated function then parsed to an exponential function called sigmoid function. We adjusted hyper parameters in " "max-iter=100000"; "solver="newton-cg"". Maximum iteration is considered 100000 since less thought of greatest emphasis gives less accuracy so the greatest cycle 100000 is taken to solve to converge. Here "solver=newton-cg" is considered to handle multinomial loss. We applied required adjustment and the extreme result for logistic regression appeared in 73.88%. Investigating the reason logistic regression works well when whole data is linearly separable with each other which can be seen by a simple set of features but here selected features are highly dependable with each other and perfect linear separable data is rarely found in the real world. Now optimizing the multinomial loss selecting optimizer algorithm may work i.g stochastic gradient descent algorithm. SGDclassifier calculates the loss function with updated weight and loss function. One special thing is added there is "Loss='log" and accuracy ended up at 71.73%.

5.6 Support Vector Classifier

Support vector classifier usually implemented by libsvm library. We used a standard scalar because a large number of input samples could have multivariate data with large numeric range. Standard scalar improves accuracy here because it makes the whole training and testing sample by scaling with mean of 0 and standard 1. Without using standard scalar accuracy dropped in 47.2% . The reason behind scaling because looking at the dataset each attribute ranges differs a lot. Describing dataset vehicle length, weight also preceding vehicle length and weight lies with different magnitude and units. This reason Support vector uses standardization to reduce feature dominance over each other and separates themselves properly by hyper plane. Including more regularization parameters acknowledges more missed classification on our dataset, expanding it into 'C=3' our score expanded into 83.67%. Here we set the regularization parameter as 'C=3', normally regularization is applied to avoid wrongful classification of data. LinearSVC normally use liblinear by minimizing squared hinge loss and takes less computational time which converge faster using 'Kernel=linear'. As it takes less time to compute so it can be defective to calculate the loss function.It can be a reason to less accuracy of using LinearSVC. We applied LinearSVC and got 72.2% accuracy.

Chapter 6

Result Analysis

We have selected four algorithms among all these algorithms we have worked with, where the accuracy rate is above 80%. These are the algorithms, Decision Tree, Random Forest tree, SVC and KNN. Before analyzing these algorithms we need to look into some parameter that works on measuring the performance rate of any machine learning algorithm.
If we look into the accuracy rate we can visualize the comparison among all these algorithms.

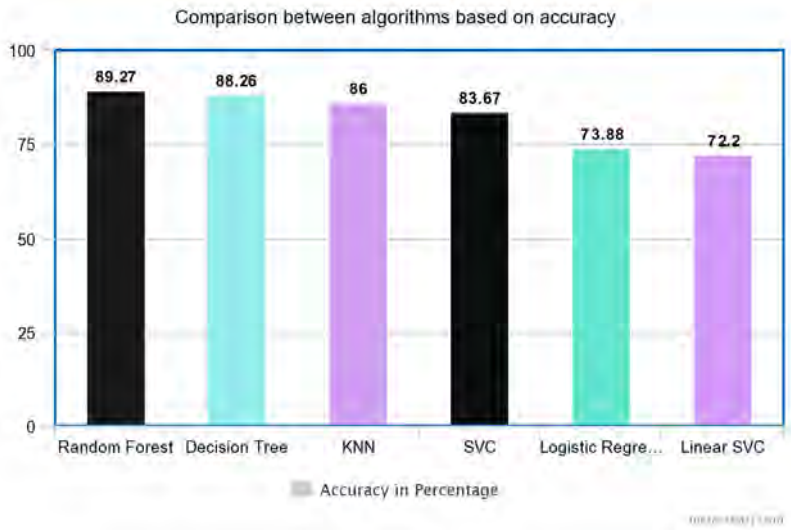


Figure 6.1: Representation of accuracy by applied algorithms

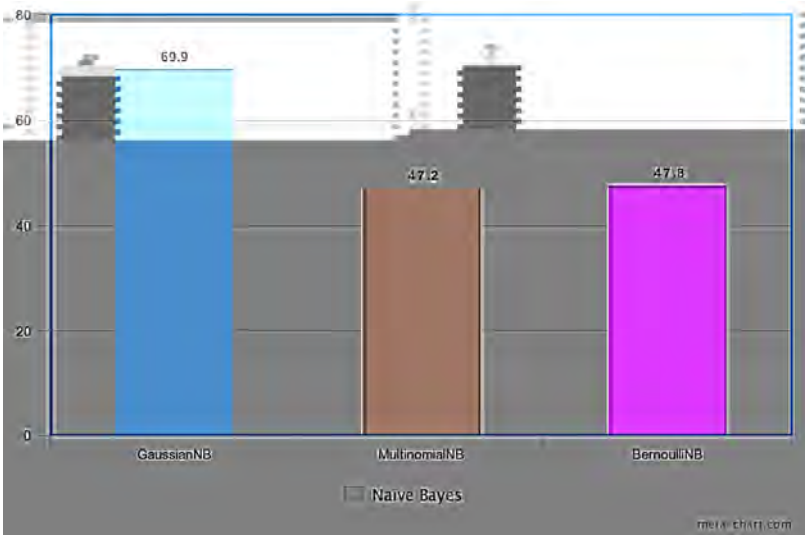


Figure 6.2: Representation of Naïve Bayes algorithm accuracy.

6.1 Confusion Matrix

Confusion matrix is a tabular formation to visualize the performance of the algorithm. Usually in binary classification, it contains two columns and two rows , where we can get four values; True positive, True positive, False positive and False negative. True positive shows the number of prediction where prediction is positive and it is true, false positive indicates the number of prediction where prediction is positive but it is false (Type 1 error), true negative indicates the number of prediction where prediction is negative and it is true and finally false negative indicates the number of prediction where prediction is negative but it is false (Type 2 error). For multiclass classification, we can get true and false value by visualizing data in the matrix and calculate the value by adding cell value.

6.2 Accuracy

Accuracy is one of the models to evaluate classification models. It is the number of correctly predicted data points among all the data. It determines the rate of correct classification. It is a ratio of correctly predicted value to the total value. The equation to determine accuracy value is for binary class is:

$$Accuracy = \frac{TruePositive + TrueNegative}{TruePositive + TrueNegative + FalsePositive + FalseNegative}$$

But in multi-class classification, we have to calculate average accuracy per class.

6.3 Precision and Recall

Precision and Recall score is another way to determine the rate of correct classification. Precision score represents the ratio of correctly predicted positive value. On the other hand, recall scores represent the ratio of correctly predicted positive value to the total predicted positive value that could have been made. Precision score only points out the correct positive prediction of all positive predictions whereas recalls can give us an indication of the missing positive prediction.

Precision for multi-class is

$$P = \frac{\sum_c TP_c}{\sum_c TP_c + \sum_c FP_c}$$

Recall for multi-class is

$$R = \frac{\sum_c TP_c}{\sum_c TP_c + \sum_c FN_c}$$

6.4 F1-score

F1 score is a way to provide a combination of both precision and recall into a single scoring system. It interprets the weighted average of precision and recall. So we can also visualize false positive and false negative values and consider those values. Accuracy score can give the best result under the circumstance of having the same false positive and false negative value. Otherwise we need to look for precision and recall score, here f1 can be very helpful.

$$F - Measure = \frac{2 \times Precision + Recall}{Precision + Recall}$$

6.5 Error calculation

Error calculation in machine learning is a very important process to identify the learning rate of a particular algorithm. It can measure how accurately an algorithm can predict the end result for previously unseen data. We have used the following methods to determine our calculation error.

6.6 Mean absolute error

Mean Absolute Error (MAE) is the result of measuring the difference between two continuous variables. It refers to the mean absolute value of each predicted error in every sample of the dataset. The equation for MAE is:

$$\sum_{i=1}^{n} \frac{|y_i - x_i|}{n}$$

Where y is predicted value, x is true value and n is number of data.

6.7 Mean squared error

MSE is the result of the average of squares of the errors. It gives better output as a loss function than MAE because unlike MAE, sum of square error is differentiable at any point and gives non-negative error. So by taking the mean of the sum of the square error will compute less total error value for a large number of data. The equation for MSE is:

$$MAE = \sum_{i=1}^{n} \frac{(y_i - x_i)^2}{n}$$

Where y is predicted value, x is true value and n is number of data.

6.8 Root mean squared error

RMSE is standard deviation of the prediction error. It gives high weight to large errors as errors were squared before they are averaged. The equation is:

$$MSE = \sqrt{\sum_{i=1}^{n} \frac{(y_i - x_i)^2}{n}}$$

Where y is predicted value, x is true value and n is number of data.

6.9 R2 score (Coefficient of determination)

R2 score is a measurement that determines the measure of how well the prediction approximates the real data. This score represents the quality of fit of a model. The equation for R2 score is:

$$R2\ Score = 1 - \frac{SSR}{SST}$$

Where SSR is sum squared regression(sum of error squared) and SST is total sum of squares.

Next, we move on into the algorithms' performance individually and evaluate them based on their performance.

6.10 Optimal Algorithms

6.10.1 Decision tree

Among all the algorithms, the decision tree gave us the highest accuracy value, which is 89.27%. Here, for this algorithm, we tried to figure out important features based on correlation with each other and only selected those features for reevaluating the whole algorithm, which aided us to achieve this high accuracy rate.

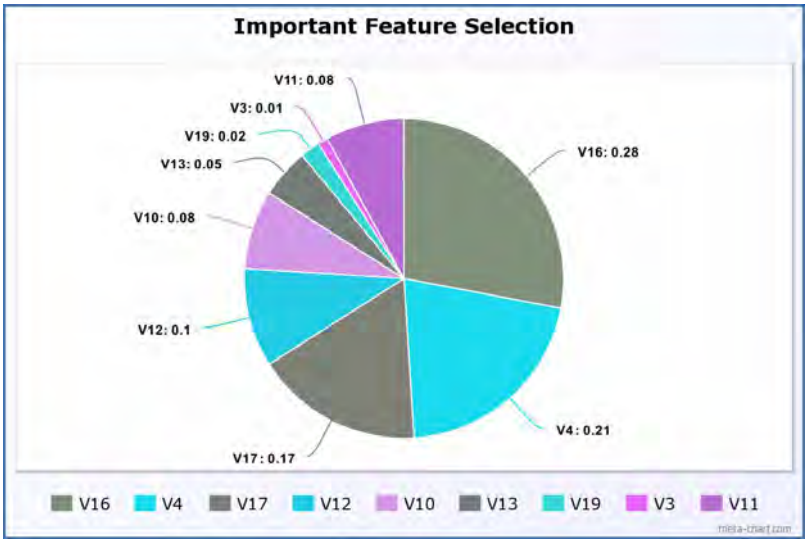


Figure 6.3: Important features in Decision Tree

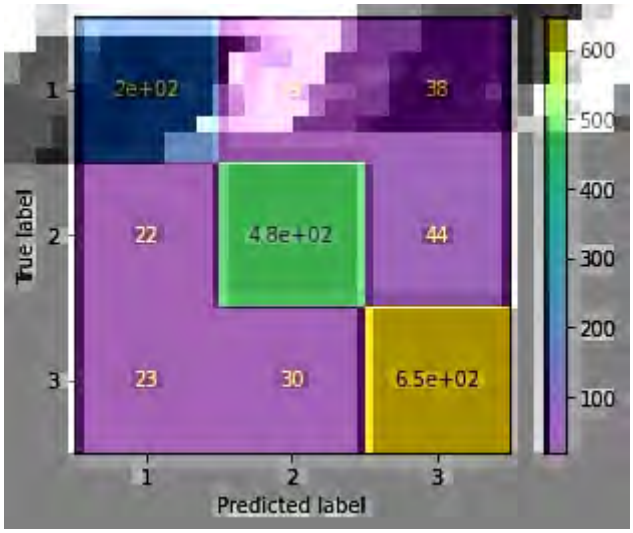


Figure 6.4: Confusion matrix of Decision Tree

If we look at the confusion matrix we can focus on more in depth analysis.

Aggressive	Normal	Vague
TP:196	TP:481	TP:647
TN: 1201	TN: 908	TN: 646
FP: 88	FP: 106	FP: 51
FN: 40	FN: 62	FN: 143

Table 6.1: Retrieved values from PL and TL for Decision Tree

From this fig we can figure out precision and recall score, which is respectively 0.8694 and 0.8595. The average of these two scores means f1 score is 0.8641, which is a very good score as it is pretty close to a perfect 1 score mark. Now if we look at error calculation, we can see the value of the MAE, MSE, RMSE and finally the R2 score is respectively 0.158, 0.2393, 0.4892 and 0.5620.

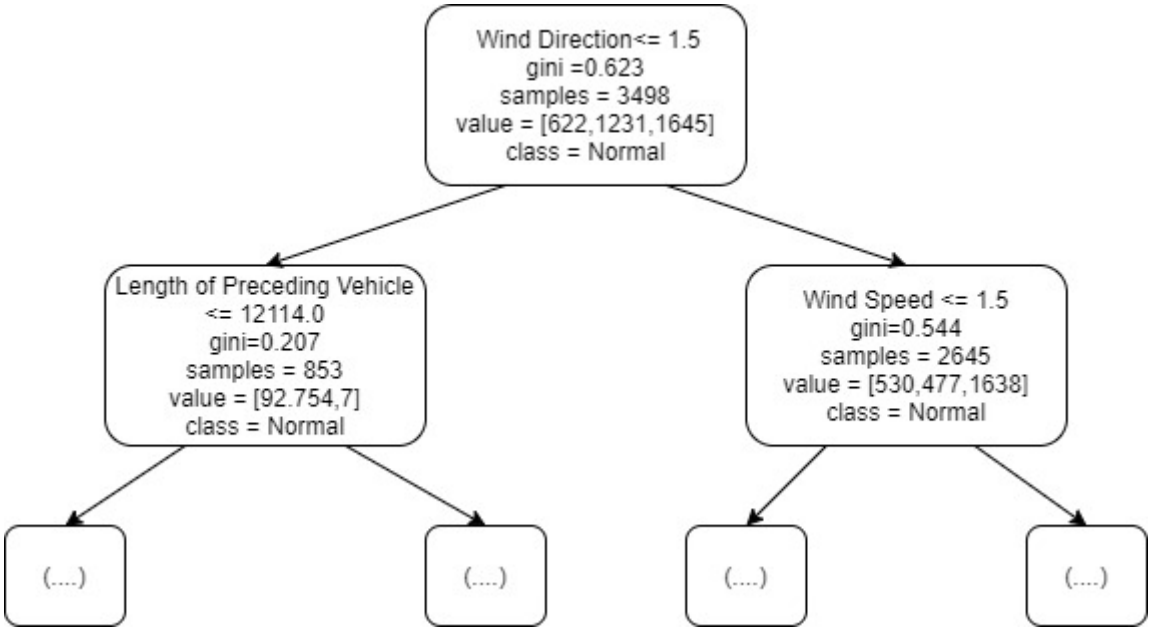


Figure 6.5: Representation of Decision Tree Classifier

Here root node wind direction includes a threshold value of 1.5 . Based on this condition the root node is split further. Here another we will see root node has 3498 samples over there which is divided in left child with 853 samples and right one is 2645 samples and value in root node implies how many samples the node got for each categories implies 622 for 'Aggressive' category , 1231 for 'Normal' category and 1645 for ' Vague' category. If we visualize the tree till final at that point we might see each of the leaf nodes are under a class always. Gini impurity always shows the chance of being incorrect in a node so when we reach till last node gini value will always be 0.0.

6.10.2 Random Forest

Random forest does a good job on accurately predicting data as well. The accuracy score on this one is 89.27%, which is pretty high. Here we have used the gini index for classification and depth value was 7. Confusion matrix for this algorithm is:

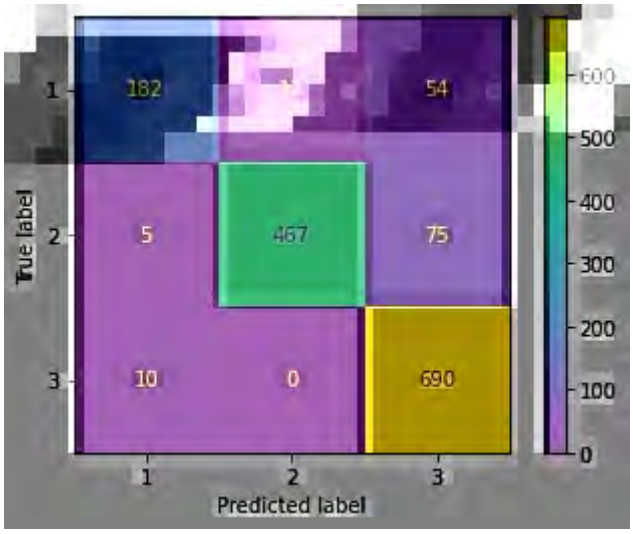


Figure 6.6: Confusion matrix of Random Forest

From this fig we can figure out precision and recall score, which is respectively 0.91 and 0.853.
If we decipher this we can get these values,

Aggressive	Normal	Vague
TP:192	TP:467	TP:690
TN:1232	TN:936	TN:671
FP:71	FP:80	FP:10
FN:15	FN:17	FN:129

Table 6.2: Retrieved values from PL and TL for Random Forest

The average of these two scores means f1 score is 0.87. In error calculation, the value of the MAE, MSE, RMSE and finally, the R2 score is respectively 0.15, 0.2353, 0.4851 and 0.5694.

6.10.3 KNN

Executing the KNN we got accuracy 69% the reason behind we extricated a V7 which signifies the ID of preceding vehicle . The reason for dropping this column is as a human presumption we thought is generally less important for the classifier model but accuracy appeared less for final consideration.
In spite of the fact that the scattered plot appears the numerous train inputs basic within the following classification field still the accuracy is less. After giving back the ID of the preceding vehicle included the result brought about in 71%
Then we tried different dimensionality reduction methods for this particular algorithm to improve the performance of this classification. Dimensionality reduction is

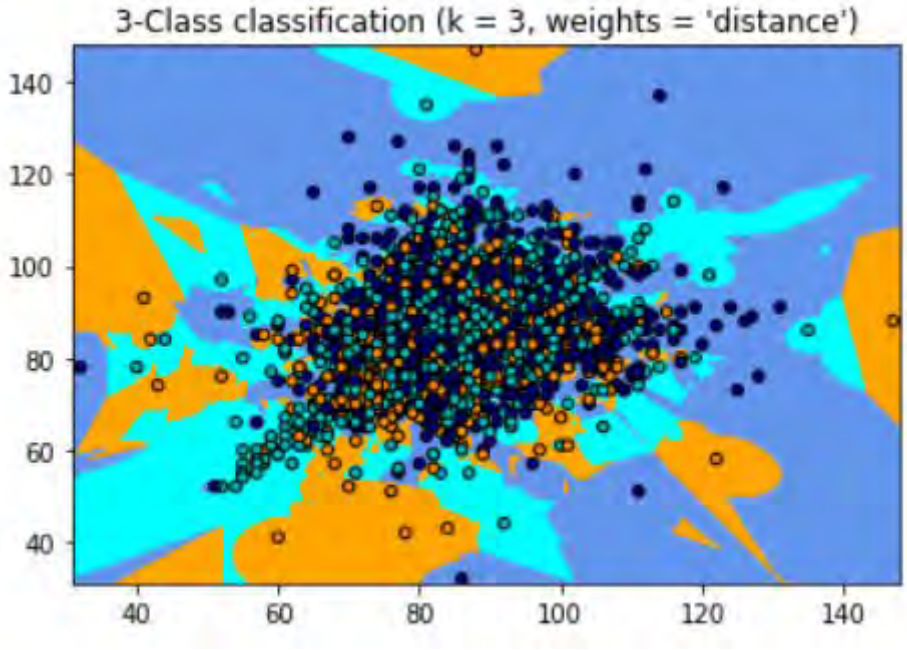


Figure 6.7: kNN visualization

a process of reducing a number of redundant data from a very large dataset. Because of having a higher dimension in the dataset, data classification process can be degraded. To solve this issue we used dimensionality reduction and figured out all the features that have more precedence over the whole classification process. There are two processes we have used to extract our data, PCA and NCA.

Principal Component Analysis(PCA)

PCA is an unsupervised, non-parametric technique where it extracts a lower dimensional space by analyzing variance-covariance structure. PCA reduces all the features that have a lower percentage of variation that each PC accounts for and makes it easier for learning the dataset. It makes maximum variability of the dataset by creating a base on the mean value of data and rotating the axes for more visibility. PCA can be computed as: [13]

$$PCA = \sum_{i=1}^{n} (Y_i - m)(Y_i - m)^T$$

Where n is the number if instance, Y is i-th instance and m is the mean vector of input data.
By using PCA we managed to get a 2D graph representing KNN classification, but the accuracy level increased by 72%, which was not more satisfactory than our initial prediction with 69%. Then we used the NCA method for classification, which gave us 86% accuracy rate. NCA also showed us correlation among the features of the dataset, where we can visualize which feature has stronger dependency among each other.

Neighborhood Component Analysis(NCA)

NCA is a non-parametric technique which selects features to maximize prediction accuracy of classification algorithms. It is a dimension reduction process but unlike PCA, NCA is a supervised method. PCA optimizes the loss variance whereas NCA works on overall classification accuracy. According to Goldberg, Roweis, Hinton and Salakhutdinov, by restricting A to be a non-square matrix of size d×D, NCA is able to work on linear dimensionality reduction. Here, the trained metric will be low rank, and the modified inputs will lie in R^d. Since the modification is linear, without loss of generality we only recognized the case d≤D. By making such a limitation, we can conceivably derive many further benefits beyond the already suitable technique for learning a KNN distance metric [3].
By using PCA we managed to get a 2D graph representing KNN classification, but the accuracy level increased by 72%, which was not more satisfactory than our initial prediction with 69%. Then we used the NCA method for classification, which gave us 86% accuracy rate. NCA also showed us correlation among the features of the dataset, where we can visualize which feature has stronger dependency among each other.

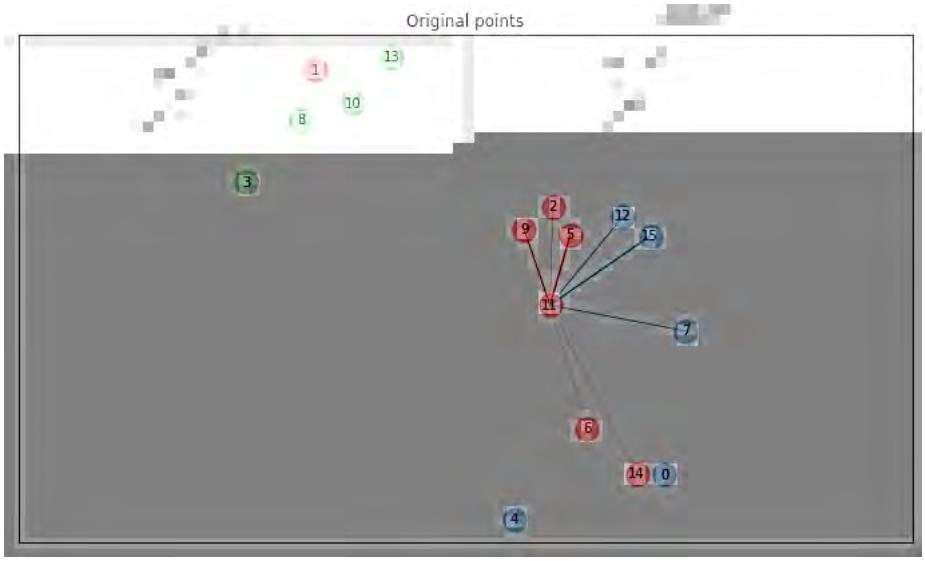


Figure 6.8: Points proportional with their distance

Before dividing the dimensionality reduction handle we took our 16 samples from 3 classes and plotted the point into original space. We centered on point 9, where we were able to see that the thickness or less thickness with other points. It primarily appears the distance among them. Then we used Neighbor component Analysis for learning those following points. After we plotted the points after change, we took embedding and found the closest neighbors.

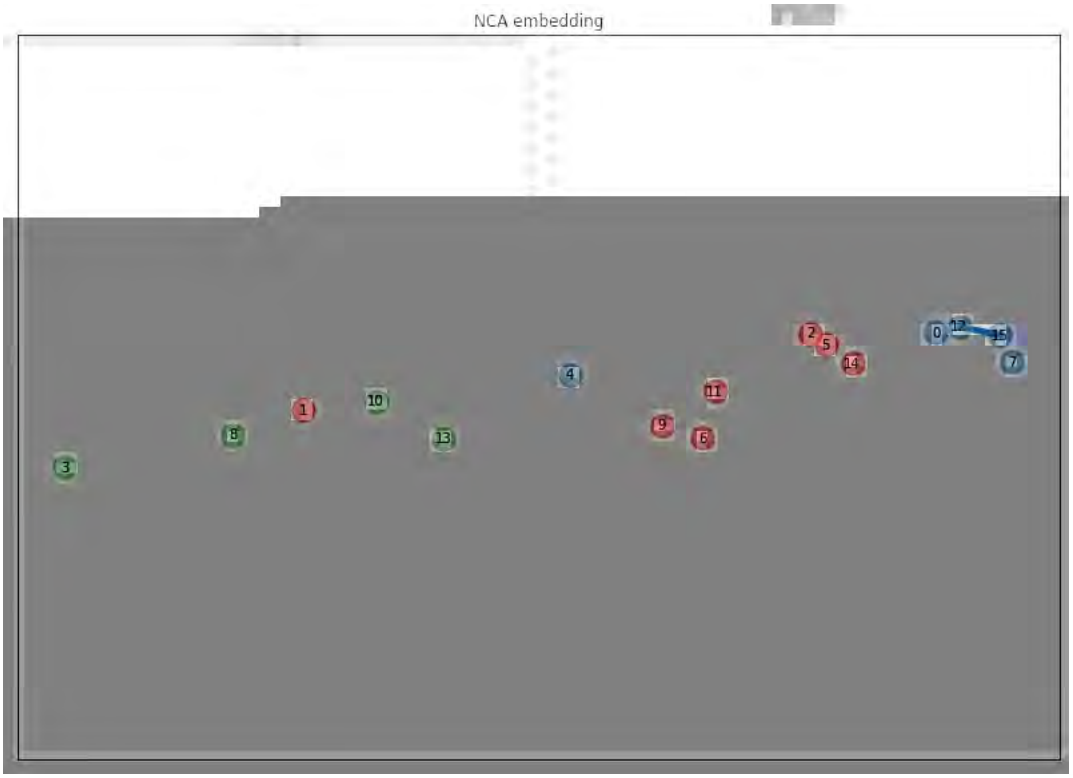


Figure 6.9: Neighbor component Analysis based on closest neighbors

As we know KNN decision boundary is controlled by the value of K which is quite problematic because of non-linearity. NCA has come up with a way better solution for this, since class decision boundaries are given by Nearest Neighbor classifier.

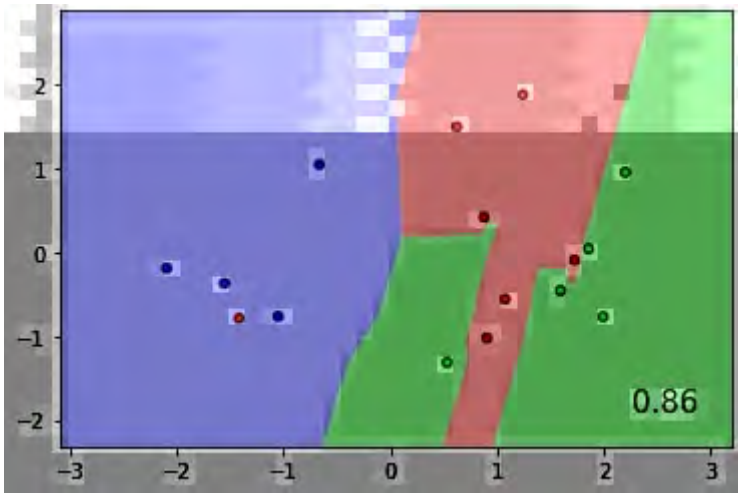


Figure 6.10: K- nearest neighbor (K=3) with Neighbor component Analysis

Here we can analyze that after applying the change of K nearest neighbor with considering Manhattan distance the figure appeared up like it, In spite of the fact that it was not perfectly linear in shape but accuracy is 86%. We might or might not get a way better linear shape by considering Euclidean or Minkowski distance but checking on accuracy below 80% which is less than our consideration. So for this particular scenario, we selected the NCA method for KNN classification. Now if we look at the confusion matrix we get,

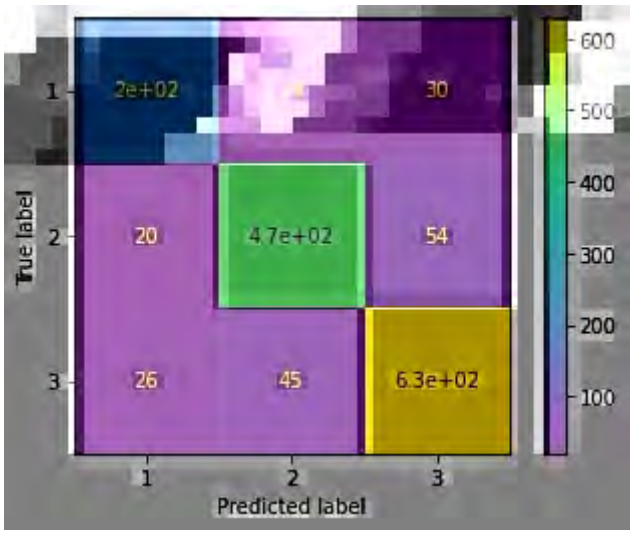


Figure 6.11: Confusion matrix of K-nearest neighbors

If we decipher this we can get these values,

Aggressive	**Normal**	**Vague**
TP:199	TP:473	TP:629
TN:1201	TN:884	TN:716
FP:54	FP:74	FP:71
FN:46	FN:69	FN:84

Table 6.3: Retrieved values from PL and TL for kNN

Here we can see, precision score and recall score is 0.8673 and 0.8499, and f1 score is 0.8527. If we focus on error calculation we can get 0.17 for MAE, 0.24467 for MSE, 0.4946 for RMSE and 0.5523 for R2 score.

6.10.4 SVC

Lastly we have SVC with 83.67% accuracy rate. Here we have used linearSVC method and SVC method. The linearSVC model in python uses the same parameter as SVC, but rather than implementing the term libsvm it uses liblinear. LinearSVC minimizes squared hinge loss whereas SVC minimizes regular hinge loss. Moreover in multiclass classification LinearSVC uses one-vs-rest and SVC uses one-vs-one Confusion matrix for this algorithm is:

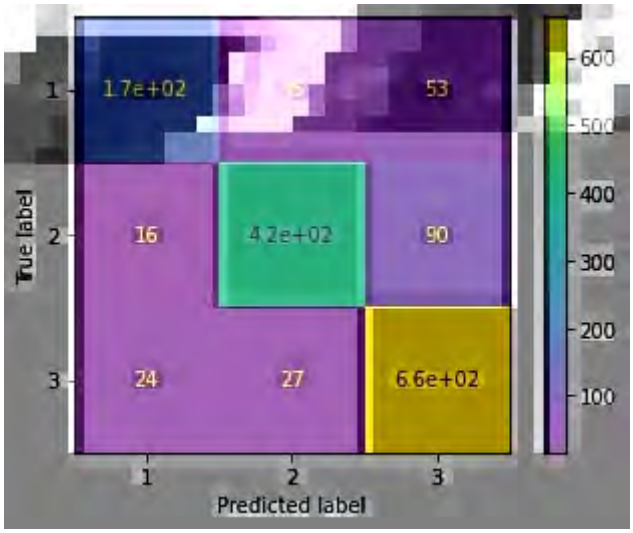


Figure 6.12: Confusion matrix of Support Vector Classifier

If we decipher this we can get these values,

Aggressive	Normal	Vague
TP:171	TP:424	TP:660
TN:1201	TN:908	TN:646
FP:88	FP:106	FP:51
FN:40	FP:62	FN:143

Table 6.4: Retrieved values from PL and TL for SVC

The precision and recall score for this one is 0.8349 and 0.796. The f1 score for SVC is 0.8202. Again for error calculation we get 0.2373 for Mean Absolute Error, 0.3653 for Mean Squared Error, 0.6044 for Root Mean Squared Error and 0.4418 for R2 score.

6.11 Comparative Analysis

If we look at this table we can see the comparison between these algorithms, we can make a comparative decision on which algorithm works best.

6.11.1 Error Calculation:

Algorithms	Mean absolute Error	Mean squared Error	Root mean squared Error	R2 Score
Random forest	0.15	0.2353	0.4851	0.5694
Decision Tree	0.1585	0.2393	0.4892	0.5620
KNN	0.17	0.2447	0.4946	0.5523
SVC	0.2373	0.3653	0.6044	0.4418

Table 6.5: Error Calculation

Here we can see the score for error calculation for different algorithms. We can see the order from least to highest Mean Absolute, Mean squared and Root mean squared errors are Random Forest, Decision Tree, KNN then SVC. Although there is no perfect score for mean absolute error, mean squared error, there we can measure the performance by the order of it. The larger the value of mean absolute error and mean squared error is, the more widely the data are scattered around the mean value. Same goes for Root mean squared error, it indicates that the lower root mean squared value can relatively predict the data accurately.

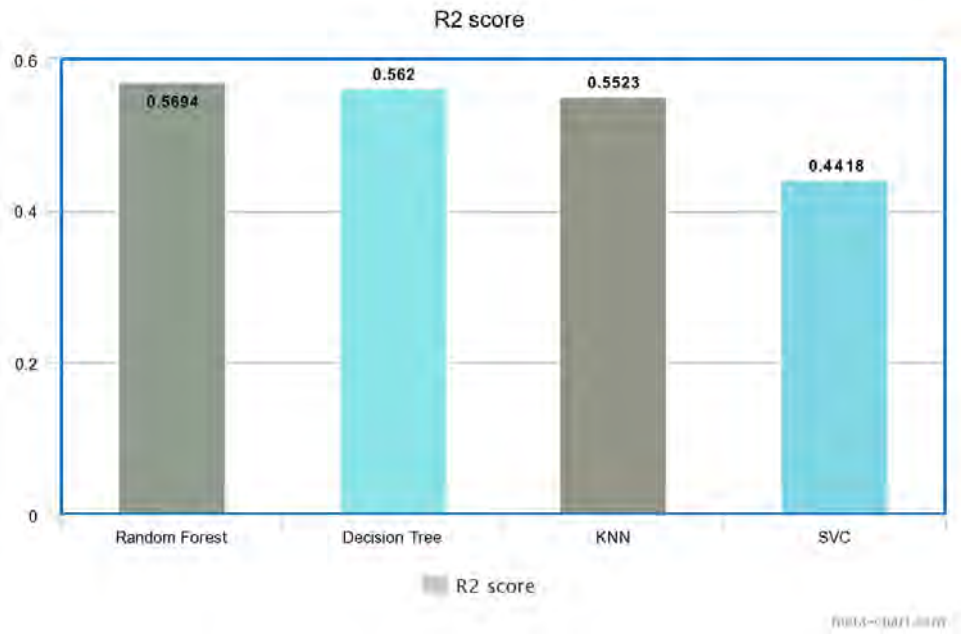


Figure 6.13: R2 score

On the other hand, R2 score which means coefficient for determination represents the value for better accuracy, higher value means the points are closer to the trend line. So from the table we can see that the Random Forest has a higher level of correlation than the others, then gradually Decision Tree, KNN and SVC algorithms.

6.11.2 Performance evaluation:

Algorithms	Accuracy	Precision	Recall	f1 score
Random forest	0.8927	0.9104	0.8525	0.8744
Decision Tree	0.8826	0.8694	0.8595	0.8641
KNN	0.86	0.8673	0.8499	0.8527
SVC	0.8367	0.8349	0.796	0.8202

Table 6.6: Performance Evaluation

To evaluate an algorithm, only accuracy is not enough. We need to compare precision score, recall score and f1 score for a better picture. Precision score is needed to show the results which are relevant. Here we can see Random Forest has the highest f1 score, then gradually the value decreases with Decision Tree, KNN and SVC. The least precision score is SVC which is still higher than 80%, which can be determined as satisfactory. Same goes for recall or sensitivity score. This is very important to determine the performance of the algorithm as well as it indicates the total relevant result correctly classified by the algorithm. We can see the same score pattern as precision and recall as well.

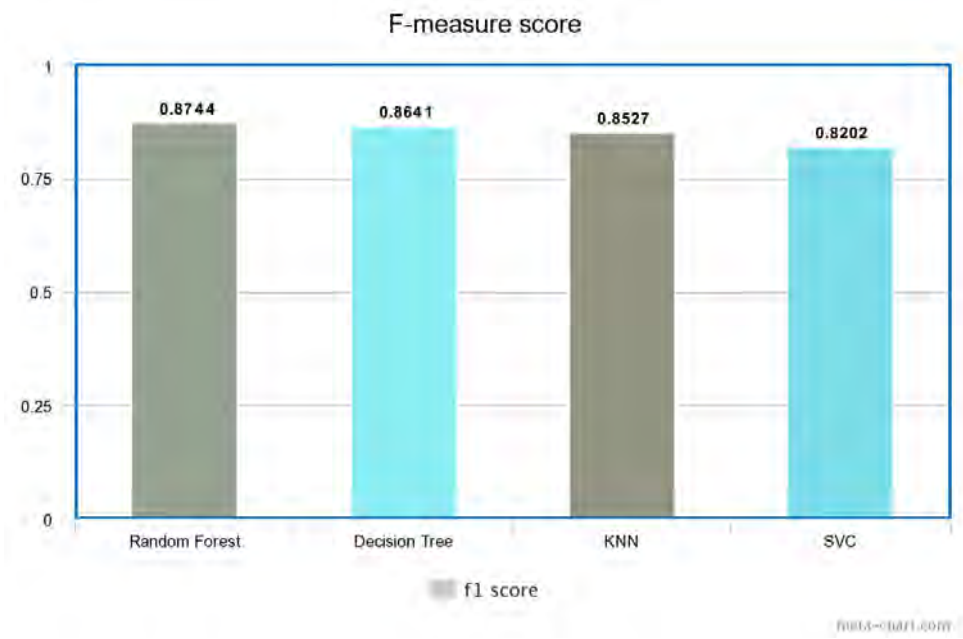


Figure 6.14: f1 score

Now if we focus on the F-measure score, we can see the similar consistency as well. So in this case we can say that there is a consistency among all these four parameters to determine algorithms performance.
If we look at these results, we can definitely see that the Random Forest makes the best prediction among all the others. Second most effective learning method is the Decision tree. Decision Tree can work with both categorical and numerical value, it can avoid over-fitting by pruning and it is a faster process for training and testing without even following any normalization and feature selection process. The most

important feature of this algorithm is, it is easier to interpret, which is beneficial for understanding data classification. Decision tree is built on a whole dataset whereas a random forest selects specific features or variables to make multiple Decision Trees and picks the best one to classify the dataset. Random Forests are less likely to over-fit than Decision Trees.
Next we have KNN, like the decision tree, KNN is also a non-parametric algorithm, this algorithm is a fast learner and easy to implement. But compared to random forest, random forest support feature interaction, which KNN cannot. This may have played an important role in our data model as well, since it is a very large model.
In classification problems, SVC is a very well-known algorithm. In high dimensional space, SVC is a very effective method, it is very effective in those cases where the number of dimensions are higher than number of cases. But in a large dataset, performance may degrade. In cases where the dataset has more noise or target classes overlap, which exist in our case, SVC can under-perform.

Chapter 7

Conclusion and Future Plan

7.1 Conclusion

In this research, we have presented an extended comparative study of driver evaluation using supervised machine learning algorithms. All these algorithms helped us to select a better method for evaluating driver's driving skills with proper classification method. Our studies find that the Random Forest algorithm works best with the attributes of our dataset due to its abilities of fast-processing and selecting best features to classify the dataset. We have also faced limitations with other algorithms as have not provided satisfactory results for our data model. With the goal to reducing road accidents and loss of precious lives in Bangladesh, we can apply this model to monitor a driver's and their respective vehicle's status and performance in a digitalised way so that the corresponding authority can take help from this system to minimize the number of accidents happening in our country. And as a bonus, ride sharing applications and relevant companies are also recommended to attain insights from our research to conduct quality business with more assurances given to the consumers. All in all, our study looks forward to contribute significantly to the rapidly changing dynamics of road and highway status in Bangladesh. As computer scientists, we continue our fight to save every life possible, specially, in terms of vehicular transit.

7.2 Future Plan

Driver evaluation systems with the help of machine learning have a lot of scope to improve. With the constant change of technology, we have to find smarter and more efficient ways to solve our problems. As we are working with driver classification we could make the classification more precise and accurate. With the help of Kalman Filter, we can improve our model further.
Kalman filter is an algorithm that is used to estimate true value, position or any object being measured that contains random error or variation. It is basically an optimal estimator that infers parameters from indirect, inaccurate and uncertain observations. Since it is a recursive process, we can use it to include new measurement and hence more accurate predictions.
In future, we want to use our research and Kalman filter, collaborating to create an embedded system diligently connected to the cloud and thus stakeholders can view the driver and vehicle status in real time. In terms of traffic analytics, with the help of this technique, we can incorporate it with our project by predicting traffic flow, vehicle density, average spacing, headway which can also help to resolve mass-traffic adversities. As of now, the expense to carry out this project is more than we can afford at undergraduate level. However, we sincerely look forward to carrying out the aforementioned project using an embedded system given the opportunities and necessary funding presented.

References

[1] J. Andrey and S. Yagar, "A temporal analysis of rain-related crash risk," *Accident Analysis Prevention*, vol. 25, no. 4, pp. 465 –472, 1993, ISSN: 0001-4575. DOI: https://doi.org/10.1016/0001-4575(93)90076-9. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0001457593900769.

[2] J. Weston and C. Watkins, "Multi-class support vector machines," Citeseer, Tech. Rep., 1998.

[3] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighbourhood components analysis," Cambridge, MA, USA: MIT Press, 2004.

[4] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas, "Data preprocessing for supervised learning," *International Journal of Computer Science*, vol. 1, pp. 111–117, Jan. 2006.

[5] A. Pérez, P. Larrañaga, and I. Inza, "Supervised classification with conditional gaussian networks: Increasing the structure complexity from naive bayes," *International Journal of Approximate Reasoning*, vol. 43, no. 1, pp. 1 –25, 2006, ISSN: 0888-613X. DOI: https://doi.org/10.1016/j.ijar.2006.01.002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0888613X0600003X.

[6] S. Kotsiantis, "Supervised machine learning: A review of classification techniques," *Informatica (Ljubljana)*, vol. 31, Oct. 2007.

[7] A. Ahmad, "Data transformation for decision tree ensembles," 2009.

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

[9] N. Chakrabarty and K. Gupta, "Analysis of driver behaviour and crash characteristics during adverse weather conditions," *Procedia - Social and Behavioral Sciences*, vol. 104, pp. 1048 –1057, 2013, 2nd Conference of Transportation Research Group of India (2nd CTRG), ISSN: 1877-0428. DOI: https://doi.org/10.1016/j.sbspro.2013.11.200. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877042813045916.

[10] M. Jackett and W. Frith, "Quantifying the impact of road lighting on road safety — a new zealand study," *IATSS Research*, vol. 36, no. 2, pp. 139 – 145, 2013, ISSN: 0386-1112. DOI: https://doi.org/10.1016/j.iatssr.2012.09.001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0386111212000325.

[11] D. Cheng, S. Zhang, Z. Deng, Y. Zhu, and M. Zong, "Knn algorithm with data-driven k value," Dec. 2014, pp. 499–512. DOI: 10.1007/978-3-319-14717-8_39.

[12] S. Taneja, C. Gupta, K. Goyal, and D. Gureja, "An enhanced k-nearest neighbor algorithm using information gain and clustering," Feb. 2014, pp. 325–329. DOI: 10.1109/ACCT.2014.22.

[13] N. Elavarasan and K.Mani, "A survey on feature extraction techniques," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 3, pp. 52–55, 2015.

[14] M. Gys Albertus Marthinus and M. Hermanus Carel, "A review of intelligent driving style analysis systems and related artificial intelligence algorithms.," *Sensors*, vol. 15, no. 12, pp. 30653 –30 682, 2015, ISSN: 1424-8220. [Online]. Available: https://ezp.sub.su.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edsdoj&AN=edsdoj.b7424db578544129ab029c535020a6a2&site=eds-live&scope=site.

[15] S. K. Patro and K. K. Sahu, "Normalization: A preprocessing stage," *ArXiv*, vol. abs/1503.06462, 2015.

[16] A. Abdulhafedh, "Incorporating the multinomial logistic regression in vehicle crash severity modeling: A detailed overview," *Journal of Transportation Technologies*, vol. 07, pp. 279–303, Jan. 2017. DOI: 10.4236/jtts.2017.73019.

[17] A. Bevan, R. Goñi, J. Hays, and T. Stevenson, "Support vector machines and generalisation in hep," *Journal of Physics: Conference Series*, vol. 898, Feb. 2017. DOI: 10.1088/1742-6596/898/7/072021.

[18] G. A. Valencia-Zapata, D. Mejia, G. Klimeck, M. G. Zentner, and O. K. Ersoy, "A statistical approach to increase classification accuracy in supervised learning algorithms," *CoRR*, vol. abs/1709.01439, 2017. arXiv: 1709.01439. [Online]. Available: http://arxiv.org/abs/1709.01439.

[19] W. Wang, J. Xi, A. Chong, and L. Li, "Driving style classification using a semisupervised support vector machine," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 5, pp. 650–660, 2017.

[20] M. Bejani and M. Ghatee, "A context aware system for driving style evaluation by an ensemble learning on smartphone sensors data.," *Transportation Research Part C: Emerging Technologies*, vol. 89, pp. 303–320, 2018, ISSN: 0968090X. [Online]. Available: https://ezp.sub.su.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edselc&AN=edselc.2-52.0-85042651929&site=eds-live&scope=site.

[21] H. Elaidi, Z. Benabbou, and H. Abbar, "A comparative study of algorithms constructing decision trees: Id3 and c4.5," in *Proceedings of the International Conference on Learning and Optimization Algorithms: Theory and Applications*, ser. LOPAL '18, Rabat, Morocco: Association for Computing Machinery, 2018, ISBN: 9781450353045. DOI: 10.1145/3230905.3230916. [Online]. Available: https://doi-org.ezp.sub.su.se/10.1145/3230905.3230916.

[22] S. Kumar and I. Chong, "Correlation analysis to identify the effective data in machine learning: Prediction of depressive disorder and emotion states," *International Journal of Environmental Research and Public Health*, vol. 15, p. 2907, Dec. 2018. DOI: 10.3390/ijerph15122907.

[23] G. M. Portal, A. G. Ghersi, P. S. Juárez, and R. G. Valenzuela, "Comparative analysis of supervised classifiers for classification of musical notes on mobile based applications," in *Proceedings of the 2nd International Conference on Vision, Image and Signal Processing*, ser. ICVISP 2018, Las Vegas, NV, USA: Association for Computing Machinery, 2018, ISBN: 9781450365291. DOI: 10.1145/3271553.3271575. [Online]. Available: https://doi-org.ezp.sub.su.se/10.1145/3271553.3271575.

[24] K. Sree and C. Bindu, "Data analytics: Why data normalization," *International Journal of Engineering and Technology(UAE)*, vol. 7, pp. 209–213, Sep. 2018. DOI: 10.14419/ijet.v7i4.6.20464.

[25] Y. Alsouda, S. Pllana, and A. Kurti, "Iot-based urban noise identification using machine learning: Performance of svm, knn, bagging, and random forest," in *Proceedings of the International Conference on Omni-Layer Intelligent Systems*, ser. COINS '19, Crete, Greece: Association for Computing Machinery, 2019, 62–67, ISBN: 9781450366403. DOI: 10.1145/3312614.3312631. [Online]. Available: https://doi-org.ezp.sub.su.se/10.1145/3312614.3312631.

[26] M.-S. Chen, C.-P. Hwang, T.-Y. Ho, H.-F. Wang, C.-M. Shih, H.-Y. Chen, and W. K. Liu, "Driving behaviors analysis based on feature selection and statistical approach: A preliminary study.," *The Journal of Supercomputing: An International Journal of High-Performance Computer Design, Analysis, and Use*, vol. 75, no. 4, p. 2007, 2019, ISSN: 0920-8542. [Online]. Available: https://ezp.sub.su.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edssjs&AN=edssjs.C00F4BC9&site=eds-live&scope=site.

[27] S. Chen, T. U. Saeed, M. Alinizzi, S. Lavrenz, and S. Labi, "Safety sensitivity to roadway characteristics: A comparison across highway classes.," *Accident Analysis and Prevention*, vol. 123, pp. 39 –50, 2019, ISSN: 0001-4575. [Online]. Available: https://ezp.sub.su.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edselp&AN=S0001457518308911&site=eds-live&scope=site.

[28] J. Du, "The frontier of sgd and its variants in machine learning," *Journal of Physics: Conference Series*, vol. 1229, p. 012 046, May 2019. DOI: 10.1088/1742-6596/1229/1/012046.

[29] R. Elvik, A. Vadeby, T. Hels, and I. van Schagen, "Updated estimates of the relationship between speed and road safety at the aggregate and individual levels.," *Accident Analysis and Prevention*, vol. 123, pp. 114–122, 2019, ISSN: 00014575. [Online]. Available: https://ezp.sub.su.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edselc&AN=edselc.2-52.0-85056884275&site=eds-live&scope=site.

[30] Y. Feng, S. Pickering, E. Chappell, P. Iravani, and C. Brace, "A support vector clustering based approach for driving style classification.," *International Journal of Machine Learning and Computing*, vol. 9, no. 3, pp. 344–350, 2019, ISSN: 20103700. [Online]. Available: https://ezp.sub.su.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edselc&AN=edselc.2-52.0-85067069664&site=eds-live&scope=site.

[31] W. Han, W. Wang, X. Li, and J. Xi, "Statistical-based approach for driving style recognition using bayesian probability with kernel density estimation," *IET Intelligent Transport Systems*, vol. 13, no. 1, pp. 22–30, 2019.

[32] A. Høye, "Vehicle registration year, age, and weight – untangling the effects on crash risk.," *Accident Analysis and Prevention*, vol. 123, pp. 1 –11, 2019, ISSN: 0001-4575. [Online]. Available: https://ezp.sub.su.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edselp&AN=S0001457518309242&site=eds-live&scope=site.

[33] Z. Li, K. Zhang, Chen B. (1, Y. Dong, and L. Zhang, "Driver identification in intelligent vehicle systems using machine learning algorithms.," *IET Intelligent Transport Systems*, vol. 13, no. 1, pp. 40–47, 2019, ISSN: 1751956X. [Online]. Available: https://ezp.sub.su.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edselc&AN=edselc.2-52.0-85059528019&site=eds-live&scope=site.

[34] O. A. Osman, M. Hajij, S. Karbalaieali, and S. Ishak, "A hierarchical machine learning classification approach for secondary task identification from observed driving behavior data.," *Accident Analysis and Prevention*, vol. 123, pp. 274 –281, 2019, ISSN: 0001-4575. [Online]. Available: https://ezp.sub.su.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edselp&AN=S000145751831114X&site=eds-live&scope=site.

[35] B. Palat, G. Saint Pierre, and P. Delhomme, "Evaluating individual risk proneness with vehicle dynamics and self-report data toward the efficient detection of at-risk drivers.," *Accident Analysis and Prevention*, vol. 123, pp. 140 –149, 2019, ISSN: 0001-4575. [Online]. Available: https://ezp.sub.su.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edselp&AN=S000145751830993X&site=eds-live&scope=site.

[36] . Qi G. (1, Wu J. (1, Y. Du, Jia Y. (1, N. Hounsell, N. Stanton, and Y. Zhou, "Recognizing driving styles based on topic models.," *Transportation Research Part D: Transport and Environment*, vol. 66, pp. 13–22, 2019, ISSN: 13619209. [Online]. Available: https://ezp.sub.su.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edselc&AN=edselc.2-52.0-85049906355&site=eds-live&scope=site.

[37] X. Qingwen, W. Ke, L. Jian John, and L. Yujie, "Rapid driving style recognition in car-following using machine learning and vehicle trajectory data.," *Journal of Advanced Transportation*, vol. 2019, 2019, ISSN: 0197-6729. [Online]. Available: https://ezp.sub.su.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edsdoj&AN=edsdoj.73935060cd8548b39e8646367415cb3b&site=eds-live&scope=site.

[38] C. Xu, X. Wang, H. Yang, K. Xie, and X. Chen, "Exploring the impacts of speed variances on safety performance of urban elevated expressways using gps data.," *Accident Analysis and Prevention*, vol. 123, pp. 29 –38, 2019, ISSN: 0001-4575. [Online]. Available: https://ezp.sub.su.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edselp&AN=S0001457518309758&site=eds-live&scope=site.

[39] M. R. Carlos, L. C. González, J. Wahlström, G. Ramírez, F. Martínez, and G. Runger, "How smartphone accelerometers reveal aggressive driving behavior?—the key is the representation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 8, pp. 3377–3387, 2020.

[40] S. Das and A. Dutta, "Extremely serious crashes on urban roadway networks: Patterns and trends," *IATSS Research*, 2020, ISSN: 0386-1112. DOI: https://doi.org/10.1016/j.iatssr.2020.01.003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0386111219301967.

[41] Z. Elamrani Abou Elassad, H. Mousannif, H. Al Moatassime, and A. Karkouch, "The application of machine learning techniques for driving behavior analysis: A conceptual framework and a systematic literature review.," *Engineering Applications of Artificial Intelligence*, vol. 87, 2020, ISSN: 09521976. [Online]. Available: https://ezp.sub.su.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edselc&AN=edselc.2-52.0-85073975446&site=eds-live&scope=site.

[42] J. Hsia and C. Lin, "Parameter selection for linear support vector regression," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–6, 2020.

[43] V. Koponen, "Conditional probability logic, lifted bayesian networks, and almost sure quantifier elimination," *Theoretical Computer Science*, 2020, ISSN: 0304-3975. DOI: https://doi.org/10.1016/j.tcs.2020.08.006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0304397520304461.

[44] J. Lee, Q. Lei, N. Saunshi, and J. Zhuo, "Predicting what you already know helps: Provable self-supervised learning," *ArXiv*, vol. abs/2008.01064, 2020.

[45] I. Silva and J. Naranjo, "A systematic methodology to evaluate prediction models for driving style classification.," *Sensors (Switzerland)*, vol. 20, no. 6, 2020, ISSN: 14248220. [Online]. Available: https://ezp.sub.su.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edselc&AN=edselc.2-52.0-85082146112&site=eds-live&scope=site.

[46] Y. Yao, O. Carsten, and D. Hibberd, "A close examination of speed limit credibility and compliance on uk roads," *IATSS Research*, vol. 44, no. 1, pp. 17 –29, 2020, ISSN: 0386-1112. DOI: https://doi.org/10.1016/j.iatssr.2019.05.003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0386111218300876.