**Classification of Road Accident Severity Using Machine Learning Techniques**

Thomas Staite – 6422510  
[staitet@uni.coventry.ac.uk](mailto:staitet@uni.coventry.ac.uk)  
*MSc Data Science & Computational Intelligence  
Coventry University*

Submitted on: 14/12/2018

***Abstract* – This paper discusses the preparation, application and evaluation of a number of Machine Learning algorithms to a data set, in an attempt to classify the severity of road traffic accidents, using data recorded by the UK Government and retrieved from the Kaggle website. Data preparation tasks; such as Class Imbalance testing and Dimension Reduction, have aided performance, reduced learning time, and increased reliability for our chosen algorithms. The selected techniques; Random Forests, K-Nearest Neighbours, and Support Vector Machines have been implemented in Python and their resulting Confusion Matrices have been analysed and compared using calculations such as Accuracy, Recall, and Precision to determine the optimal algorithm for this particular dataset. The results have been discussed with application suggestions proposed and the social, legal, and ethical considerations have been stated and discussed.**

I. INTRODUCTION

The Department for Transport reported that road accident causality levels in the UK for the year 2017 reached a total of 170,993 (down 6% on previous year) with 1,793 (0% change) of these causalities listed as fatalities. Since the recording of these statistics in 2007 where the causalities and fatalities were 247,780 and 2,946 respectively, the UK has undergone vast improvement in road safety management, traffic monitoring, and vehicle safety, which in turn has contributed to the decline in casualties over the last decade. (Department for Transport, 2017)

Although the reduction in traffic accidents over time is evident, the total number of casualties per year is still considered a high value, averaging 5 fatalities and 468 casualties per day. Further reduction of these values involves the exploration and analysis of current road accident data with the intentions to identify possible explanations as to why such accidents are still occurring. A focus on exploring the relationship between accident severity and driving environmental factors may uncover underlying factors contributing to the likelihood of an accident occurring. Feature-based learning which incorporates a selection of machine learning algorithms will enable us to build mathematical models that can aid us in uncovering the underlying factors. The implementation of the aforementioned algorithms would require the gathering, processing, and preparation of relevant data related to traffic accidents in the UK. The identified data is discussed in the next section. Training of the selected models will include altering a set of model parameters in an attempt to derive the best possible configurations for each model. Upon completion of the training and testing of the models, the optimal model, and its parameters, will be evaluated further and suitable use cases for real-life application will be suggested to a target audience.

A related article, published by So Young Sohn and Sung Ho Lee of Yonsei University, South Korea, explored the severity of traffic accidents in Korea, using fusion, ensemble and clustering algorithms. Their research found that “*In terms of classification accuracy, the Dempster–Shafer algorithm appears to improve the classification performance of not only individual algorithm such as the neural network and the decision tree but also fusion algorithms including the Bayesian and logistic fusion.”* (So Young Sohn, Sung Ho Lee, 2003). This research was backed up by published results which shows the marginal differences in the model accuracies. The data has been re-created below.

|  |  |
| --- | --- |
|  | Accuracy (%) |
| Decision Tree | 72.30 |
| Neural Network | 70.86 |
| Dempster-Shafer | 72.79 |
| Bayesian | 71.23 |
| Logistic fusion | 72.30 |

*Table 1. Re-creation of results*

Furthermore, “*ensemble algorithms such as bagging and arcing also showed improvement in classification accuracy”* providing accuracies of 72.70% and 74.78% respectively. (So Young Sohn, Sung Ho Lee, 2003)This article, although similar, looks at the problems associated with road accidents in Korea, a massively contrasting country to the UK, and so the results obtained here will not be a true representation for the same situation in this country. Ensemble algorithms in this article have proved to have the highest accuracy on the targeted dataset and so it would be logical to consider selecting an algorithm of this type for our own analysis.

Moreover, Zhuoning Yuan et al, published an article in August 2017 where they looked at the prediction of traffic accidents through heterogeneous urban data in the USA. Their particular study compared four classification models – Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), and Deep Neural Networks (DNN). An insight into the experiment and the obtained results conveys some interesting statistics. Of the four methods, Support Vector Machines performed the worst in terms of accuracy, averaging 64.12%. Decision Tree, Random Forest, and Deep Neural Network achieved average accuracies of 86.53%, 88.18%, and 91.90% respectively. The data they used consisted of “*415,153 motor vehicle crashes over 8 years containing 40 features related to traffic crash”*, with features being summarised into four categories – Temporal Factors, Weather Factors, Road Factors, and Human Factors. (Zhuoning Yuan et al, August 2017)

Likewise, with the article published by So Young Sohn and Sung Ho Lee, the country at focus for this article has many differences to the UK, including; but not limited to, highway codes, road safety, and vehicle rules, and so again the results obtained here will not be completely alike to our own study.

The order of our study is as follows – Section II will discuss the identified data set that will be used to train our models, Section III will provide details of the models that have been selected and applied to our data, Section IV explores the various data preparation techniques used to transform and prepare our data, Section V presents the results from the experiment, Section VI discusses the achieved results and identifies real-life use cases for our models, and finally Section VII will weigh-up the social, legal, and ethical issues relating to this study.

II. THE DATA SET

The dataset used in this project has been sourced from Kaggle user Dave Fisher-Hickey at <https://www.kaggle.com/daveianhickey/2000-16-traffic-flow-england-scotland-wales>, whom originally collected the data from the Department of Transport government website (<https://www.dft.gov.uk/traffic-counts/download.php>). The downloaded folder consisted of three CSV files containing data for over a million traffic accidents between the years 2005 to 2014. The three files were separated into yearly intervals which were ‘accidents\_2005\_to\_2007, ‘accidents\_2009\_to\_2011, and ‘accidents\_2012\_to\_2014’. To reduce the learning time of our models, and to ensure completion of the project before the submission date, only the most recent file (accidents\_2012\_to\_2014) has been selected for analysis. This particular file contained 464,497 instances with 33 features (including the target variable). The features included in this data have been listed in the appendix.

The feature ‘Accident\_Severity’ is the target variable, which can have any one of three values - 1 being Fatal, 2 being Series, and 3 being Slight.

In Section IV, the issues with class imbalance and feature selection are addressed with the aim to maximise the performance of the machine learning algorithms, while maintaining a relatively low training time.

III. CLASSIFICATION TECHNIQUES

*Random Forests (RF)* – a supervised learning algorithm that consists of an ensemble of Decision Trees, which are most commonly trained with the ‘bagging’ method. Decision Trees are created based on parameters such as the number of features to consider and the percentage of the raw training set to include – this can have duplicates. The trees are merged together to obtain a more accurate, stable prediction. (Niklas Donges, 2018)

Each individual Decision Tree splits the data by selecting the features that maximise the information gain at each level and repeats this until a ‘pure’ node is reached or no feature increases information gain.

A big advantage of random forests that gives it the credibility of being one of the most commonly used algorithms is the ability to use the model for both classification and regression problems. In our case, this model will be used for classification.

*K-Nearest Neighbours (KNN)* – a machine learning algorithm that determines the classification of a new instance by interpreting the “K” closest neighbours to this instance. (Vik Paruchuri, 2015) The closeness of instances is determined by the distance calculation – one of the most commonly used is the Euclidean distance, which can be defined as:

K-Nearest Neighbours is particularly useful when there are not many features to describe the instances. Particular advantages of the algorithm are the fast training times and the ability to learn complex target functions.

*Support Vector Machine (SVM) –* a machine learning algorithm whose objective is “*to find a hyperplane in an N-dimensional space (N – the number of features) that distinctly classifies the data points”* (Rohith Gandhi, 2018)

The basic ideas of SVM are as follows:

* A kernel function maps data onto a high dimensional space where it can be classified with linear decision surfaces.
* The optimal hyperplane that maximises the degree of separation between two classes is found.
* A hyperplane that is found for data that is not linearly separable will maximise the margin between the classes while minimizing the number of misclassifications.

SVM training is relatively easy as no local optima is found, and the ability to scale to high-dimensional data makes this algorithm a popular choice amongst Data Science professionals.

IV. DATA PREPERATION AND EXPERIMENT SETUP

Data exploration and pre-processing is a vital step in building and training machine learning algorithms. Our data must be consistent, accurate, and presented in the correct format in order for the models to be able to execute and predict, and generalise well.

When we had an initial look at the data, we found a number of issues that needed addressing. First and foremost, the feature ‘Junction\_Detail’ had no value stored for every instance of the dataset, and so was removed due to its redundancy. In addition to this, a small number of instances had no value for the ‘Time’ feature, and so we decided to remove these instances as the value for this feature could not be calculated or inferred from other areas of the dataset. We also felt that time may have relatively strong predicting power.

The location of each traffic accident was represented in two unique ways - the ‘Longitude’ and ‘Latitude’ of the accident, and then the 'Location\_Easting\_OSGR' and 'Location\_Northing\_OSGR'. These two sets of features, from a machine learning point of view, can be seen as duplicate columns (representing the same thing semantically) and so we decided to remove the latter set of features to make the dataset a little simpler.

Dimension Reduction occurred at the next stage to select only the features from the data that we believed would be important in determining the severity of an accident. The features ‘Police\_Force’, ‘Accident\_Index’, ‘Local\_Authority\_(District)’, ‘Local\_Authority\_(Highway)’, ‘Date’, ‘1st\_Road\_Number’, ‘2nd\_Road\_Number’, ‘Did\_Police\_Officer\_Attend\_Scene\_Of\_Accident’, and ‘LOSA\_Of\_Accident\_Location’ were chosen to be removed from the dataset as it was believed they would hold very little predicting power and would waste valuable time when training the models.

Resolving missing values involved using Pandas’ ‘fillna’ function, which sets a default value for all instances of a chosen column whom do not currently hold a value. There were four columns that had missing values for a portion of instances – for the ‘Junction\_Control’ column, we set the value as ‘UNK’, for ‘Road\_Surface\_Conditions’ we assumed ‘Normal’, for ‘Special\_Conditions\_at\_Site’ we assumed ‘None’, and for ‘Carriageway\_Hazards’ we also assumed ‘None’.

Time, originally in the format 24HH:MM, was split into two separate columns; one for ‘hours’ and one for ‘minutes’, as we felt that an integer datatype for this feature would be easier to work with than a ‘datetime’ data type. We also altered the datatypes for a number of columns to make them better represent the type of data that was being stored.

It is common practice to arrange the dataframe so that the features (x) are at the front of the dataframe with the target class (y) at the end. Re-arranging the dataset has allowed us to split the data into training and testing sets easier.

Outlier detection was performed to identify inconsistencies in the data and see if they could be justified using domain knowledge. Looking at the counts of the values for each column, we found that one particular instance had a value of 67 for the number of vehicles involved in the accident with a total of 70 casualties. An online search for the accident and the date it happened, revealed that this did in fact occur (<https://www.bbc.co.uk/news/uk-england-kent-23970047>), and so although it is an outlier, the data is accurate and can remain in the dataset.

Likewise, an instance containing only 2 vehicles but with 93 casualties seemed suspicious and so with an online search, we found that this accident involved a double decker bus carrying a large quantity of passengers (<https://www.bbc.co.uk/news/uk-england-beds-bucks-herts-29687707>) – again, this data appears accurate, and so will remain in the dataset. Outlier detection was completed on a number of columns in our dataset.

Naturally splitting the data by their Accident\_Severity class value will allow us to look at the shape of the dataframe for each of the target classes and check if any of these classes are being heavily over-/under- represented. We have obtained the following results from doing this:

Severity 1 (fatal) has shape: (5302, 22)

Severity 2 (serious) has shape: (66778, 22)

Severity 3 (slight) has shape: (392604, 22)

Class Imbalance is evidently present here and in order to process our data correctly, we must inform Under- and Over-sampling to tackle this issue. Under-sampling has been applied to scale the data to 80,000 instances where the accident severity is either 2 or 3 (40,000 for each class), whereas over-sampling with replacement has enabled us to duplicate data for the remaining class – again scaling to 40,000 instances.

A re-run of the sampling was completed for the Support Vector Machine algorithm where only 4,000 instances were selected for each class, this was done to tackle the issues with learning time for this particular model as the overall time available for the project was limited.

The application of One Hot Encoding (OHE) has converted categorical variables – like the day of the week, and whether it was an urban or rural location, into a form that can be passed into our selected machine learning models. SciKit’s implementation of Random Forests can only handle numerical inputs and so using OHE is a convenient way of representing our categorical data, numerically. An example of an alteration made by OHE is as follows:

Before OHE

|  |
| --- |
| Urban\_or\_Rural |
| Urban |
| Rural |
| Rural |
| Rural |
| Urban |
| Rural |
| Urban |

After OHE

|  |  |
| --- | --- |
| Urban\_or\_Rural:Urban | Urban\_or\_Rural:Rural |
| 1 | 0 |
| 0 | 1 |
| 0 | 1 |
| 0 | 1 |
| 1 | 0 |
| 0 | 1 |
| 1 | 0 |

The final step of preparation was to split the dataframe into training and testing data. An agreement was made to keep 80% of the data as training data, with the remaining 20% being recognised as the testing data.

Number instances X\_train dataset: (96000, 156)

Number instances y\_train dataset: (96000, 1)

Number instances X\_test dataset: (24000, 156)

Number instances y\_test dataset: (24000, 1)

V. EXPERIMENT RESULTS

To evaluate the performance of the algorithms selected, we have used the following metrics:

Accuracy – the fraction of total predictions our model was correct on, defined as

Precision – the fraction of relevant instances among the retrieved instances, defined as

Recall – the fraction of relevant instances that have been retrieved over the total amount of relevant instances, defined as

F-Score – the harmonic mean of precision and recall, defined as

Where,

And,

These metrics, along with the accompanying confusion matrix, will help us to evaluate how well the model has been able to generalise and classify the test data. The confusion matrix will give us an insight into the predictions that were made and will allow us to see whether the algorithms performed better on a specific class.

*Random Forests (RF)*

We have ran this model with varying parameters for ‘n\_estimators’ – the number of trees to be built before taking maximum voting or average of predictions. The range of values are [20, 50, 100, 150, 200, 220, 230, 250, and 300]. The accuracy report is as follows:

For 20 trees, accuracy: 82.9166666667 For 50 trees, accuracy: 84.1166666667 For 100 trees, accuracy: 84.4083333333 For 150 trees, accuracy: 84.4625  
For 200 trees, accuracy: 84.4458333333 For 220 trees, accuracy: 84.4666666667 For 230 trees, accuracy: 84.3916666667 For 250 trees, accuracy: 84.4041666667 For 300 trees, accuracy: 84.4541666667

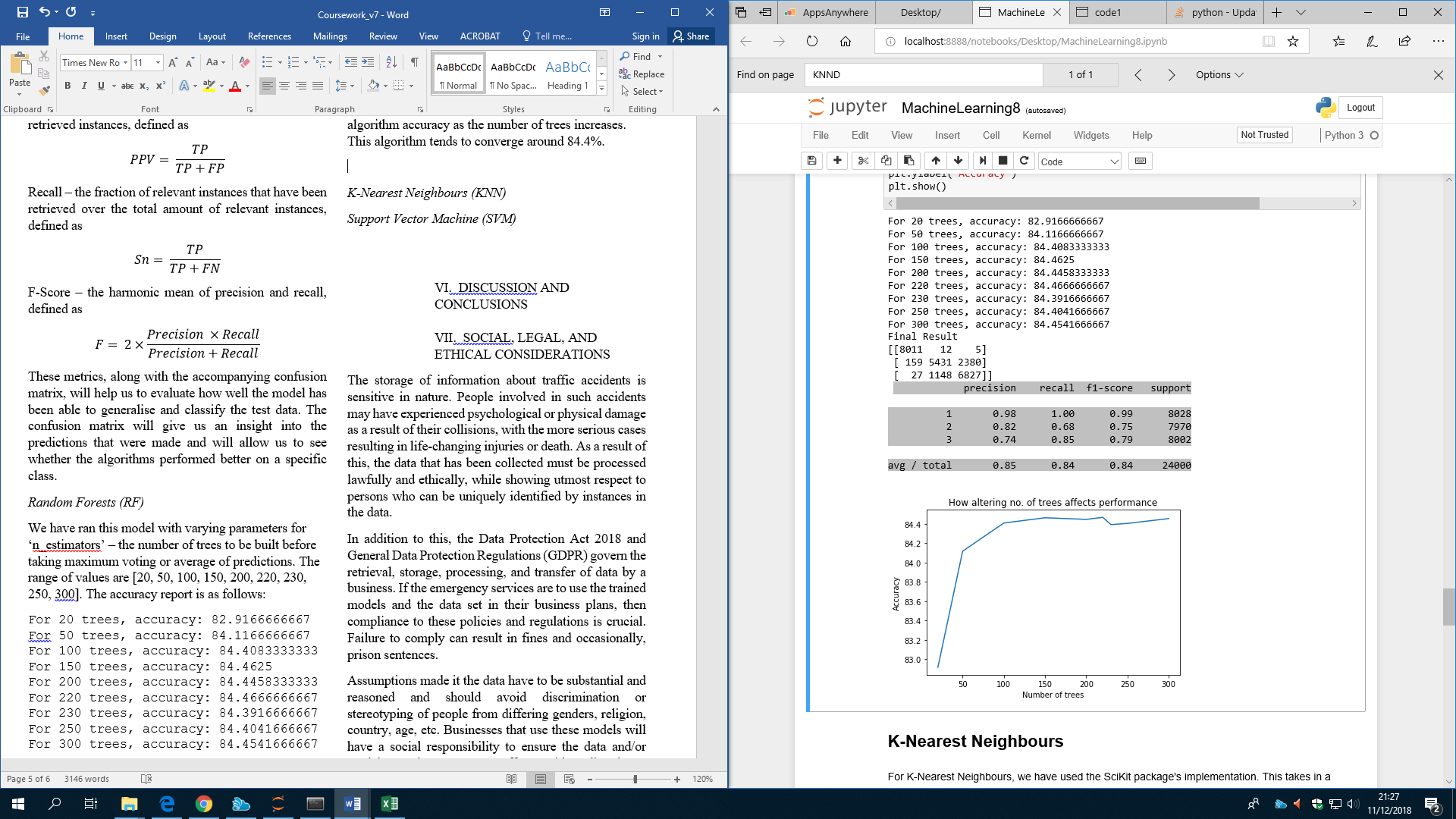
The confusion matrix for the final iteration is:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **Truth Data** | | |
|  |  |
| **Classifier Results** |  | Severity 1 | Severity 2 | Severity 3 |
| Severity 1 | 8011 | 12 | 5 |
| Severity 2 | 159 | 5431 | 2380 |
| Severity 3 | 27 | 1148 | 6827 |

And the Precision/Recall/F-measure report is:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F-Measure** |
| Severity 1 | 98% | 100% | 99% |
| Severity 2 | 82% | 68% | 75% |
| Severity 3 | 74% | 85% | 79% |
|  |  |  |  |
| **AVG / Total** | 85% | 84% | 84% |

A graph has been created to show the converging of the algorithm accuracy as the number of trees increases. This algorithm tends to converge with accuracy approximately 84.4%.



The precision for the first class is 98%, with recall 100%. These values are considerably high and it is likely that this is due to the duplication of instances when we implemented sampling with replacement. In addition to this, it is possible that instances in the first class are easier to distinguish compared to the other two classes.

The maximum accuracy that we achieved for this particular model is 84.46% when the number of trees was 150.

*K-Nearest Neighbours (KNN)*

We have run this model with varying parameters for K to see how the accuracy changes. The values we chose were [3, 5, 10, 15, 20, 30, and 50]. The accuracy report after running the algorithm 7 times is as follows:

For 3 Nearest Neighbours, accuracy: 0.743958333333  
For 5 Nearest Neighbours, accuracy: 0.72925  
For 10 Nearest Neighbours, accuracy: 0.690708333333  
For 15 Nearest Neighbours, accuracy: 0.675416666667  
For 20 Nearest Neighbours, accuracy: 0.671791666667  
For 30 Nearest Neighbours, accuracy: 0.664208333333  
For 50 Nearest Neighbours, accuracy: 0.655

The confusion matrix for the best iteration is:

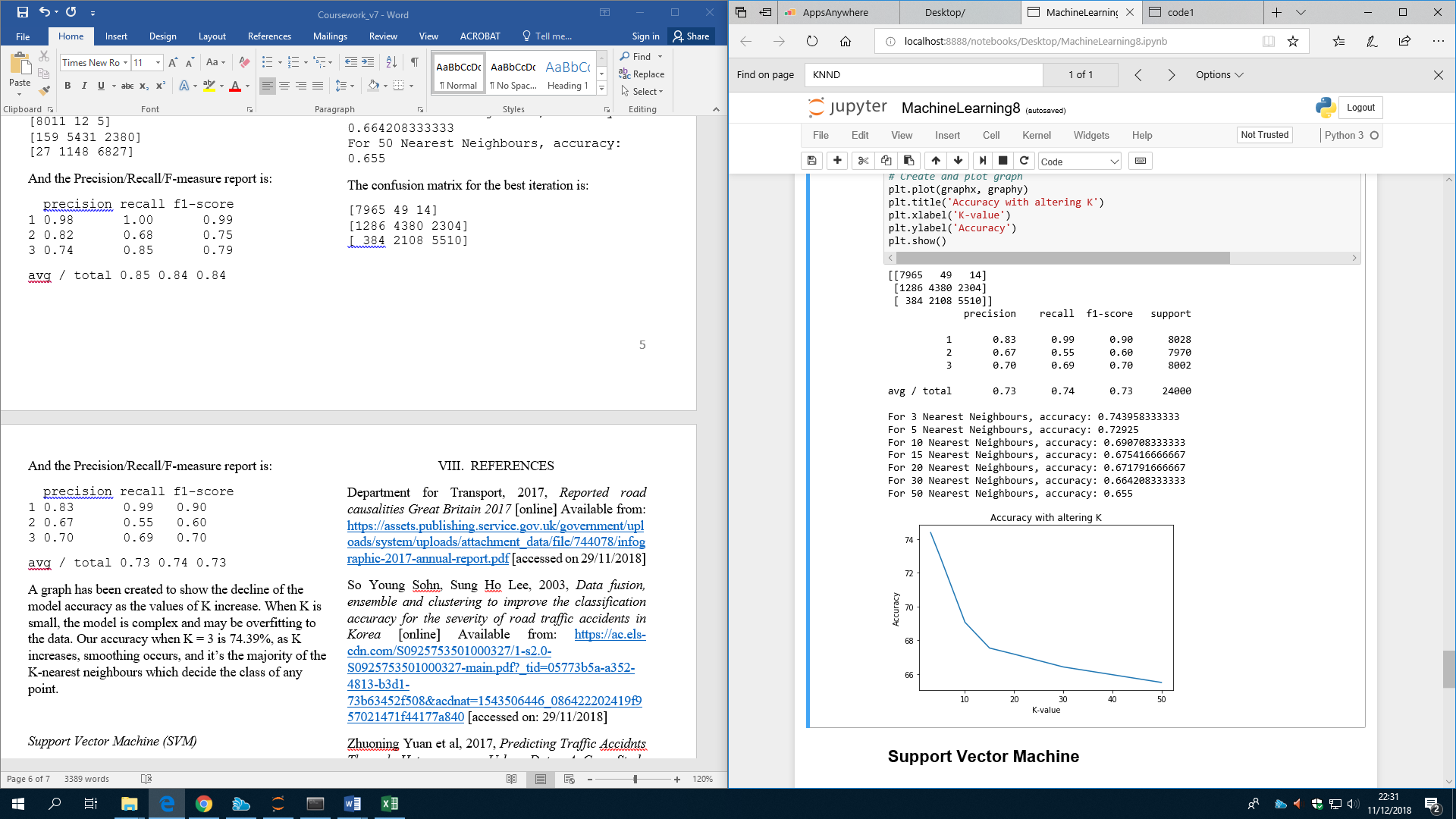
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **Truth Data** | | |
|  |  |
| **Classifier Results** |  | Severity 1 | Severity 2 | Severity 3 |
| Severity 1 | 7965 | 49 | 14 |
| Severity 2 | 1286 | 4380 | 2304 |
| Severity 3 | 384 | 2108 | 5510 |

And the Precision/Recall/F-measure report is:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F-Measure** |
| Severity 1 | 83% | 99% | 90% |
| Severity 2 | 67% | 55% | 60% |
| Severity 3 | 70% | 69% | 70% |
|  |  |  |  |
| **AVG / Total** | 73% | 74% | 73% |

Similarly to the Random Forest, the Recall and Precision for the first class is relatively high compared to the other two classes. The second and third class for this algorithm tends to have prediction accuracies slightly worse than what was achieved in the Random Forest.

A graph has been created to show the decline of the model accuracy as the values of K increase. When K is small, the model is complex and may be overfitting to the data. Our accuracy when K = 3 is 74.39%, as K increases, smoothing occurs, and it’s the majority of the K-nearest neighbours which decide the class of any point.



*Support Vector Machine (SVM)*

Non-separable features often become linearly separable after they are mapped to a high-dimensional feature space. This is done using a kernel. We have decided to use the Radial Basis Function for our implementation.

This algorithm takes in a value for the parameter 'C' - the penalty parameter C of the error term and a value for the parameter 'Gamma' - the kernel coefficient for the kernel that we are using.

We test a range of ‘C’ values – 0.001, 0.01, 0.1, 1, 10, and ‘Gamma’ values – 0.001, 0.01, 0.1, 1, using Grid Search and choose the values that perform the best on our data. These values were C = 10, and Gamma = 0.001.

The confusion matrix that was achieved for this algorithm is:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **Truth Data** | | |
|  |  |
| **Classifier Results** |  | Severity 1 | Severity 2 | Severity 3 |
| Severity 1 | 483 | 197 | 110 |
| Severity 2 | 271 | 256 | 275 |
| Severity 3 | 189 | 164 | 455 |

SVM Accuracy: 49.75

The Precision/Recall/F-Score report is:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F-Measure** |
| Severity 1 | 51% | 61% | 56% |
| Severity 2 | 41% | 32% | 36% |
| Severity 3 | 54% | 56% | 55% |
|  |  |  |  |
| **AVG / Total** | 49% | 50% | 49% |

In comparison to the other two algorithms, this algorithm performs considerably worse, with less than 50% total accuracy. This could be due to the fact that the data was scaled down massively to only 4,000 instances of each class in order to combat the high learning times required for this model. Training this algorithm on a cluster/supercomputer with many more instances for each class may improve the accuracy of the model while maintaining a satisfactory learning time.

The summary statistics for these three algorithms are as follows

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Random Forest** | **KNN** | **SVM** |
| Total Training Instances | 96000 | 96000 | 9600 |
| Total Testing Instances | 24000 | 24000 | 2400 |
| **Severity 1** |  |  |  |
| True Positives (TP) | 8011 | 7965 | 483 |
| True Negatives (TN) | 12258 | 9890 | 711 |
| False Positives (FP) | 17 | 64 | 307 |
| False Negatives (FN) | 186 | 1670 | 460 |
| Precision (%) | 98 | 83 | 51 |
| Recall (%) | 100 | 99 | 61 |
| F-Measure (%) | 99 | 90 | 56 |
| **Severity 2** |  |  |  |
| True Positives (TP) | 5431 | 4380 | 256 |
| True Negatives (TN) | 14838 | 13475 | 938 |
| False Positives (FP) | 2539 | 3590 | 546 |
| False Negatives (FN) | 1160 | 2157 | 361 |
| Precision (%) | 82 | 67 | 41 |
| Recall (%) | 68 | 55 | 32 |
| F-Measure (%) | 75 | 60 | 36 |
| **Severity 3** |  |  |  |
| True Positives (TP) | 6827 | 5510 | 455 |
| True Negatives (TN) | 13442 | 12345 | 739 |
| False Positives (FP) | 1175 | 2492 | 353 |
| False Negatives (FN) | 2385 | 2318 | 385 |
| Precision (%) | 74 | 70 | 54 |
| Recall (%) | 85 | 69 | 56 |
| F-Measure (%) | 79 | 70 | 55 |
|  |  |  |  |
| Average Accuracy (%) | 84.22 | 69.00 | 49.75 |

VI. DISCUSSION AND CONCLUSIONS

Comparing the three models, it is clear to see that Random Forest has by far been the best performer with an average accuracy of 84.22%, compared to 69% for KNN, and 49.75% for SVM. The total number of True Positives and True Negatives for RF were higher with a total of 20,269 / 24,000, compared to 17,855 / 24,000 for KNN, and 1,194 / 2,400 for SVM.

As for individual class performance, the Severity 1 class seems to have very high prediction success for RF and KNN, while it holds poor results for SVM. Severity class 2 and 3 have fairly competent results for RF and KNN, but also perform poorly with SVM.

Referring back to the study by Zhuoning Yuan et al which was discussed in the introduction, it was found that for their similar problem, SVM also performed the worst out of the 4 models they chose to implement, achieving only 64.12% accuracy. On top of this, it is apparent that their RF model was also much more suited to this type of problem as they also achieved a much higher accuracy of 88.18%. This goes to show that some models are better suited to specific dataset types and sizes and in our case, it is the RF model that is the optimal solution.

Further study and improvement on the work we have completed would involve repeating the learning process for the models but with a much bigger sample size and on a more powerful computer – either a computing cluster or a supercomputer. The expected result of doing this would be a better fitted model which in turn would result in higher accuracies as the model is able to train on more data. The learning time for these models will be minimized as the hardware used to train the models are more powerful and so would be able to complete more operations per time unit.

In addition to this, the collection of more specific data relating to the accidents may help in determining their severity. Features like the type of vehicle/s involved, information about the driver and passenger/s, and information on any vehicle faults that were identified, may hold predictive power that will enable our models to achieve better results – again, increasing the sample size and number of features would require the models to be ran on more powerful hardware.

A more thorough exploration of the locations; Longitude and Latitude features, of each accident along with the application of clustering techniques may also provide some information as to where accidents are more likely to happen and why. If certain areas or roads contain higher than average counts of accidents, then we could narrow down our research to these areas to try and reason as to why this is and suggest actions that could be taken to prevent future accidents occurring.

A real-life application of our work could be used in the emergency services sector. When a new traffic accident is reported to the emergency services via the telephone number ‘999’, the information collected from the phone operator may be fed into our RF model and the accident severity can be predicted. This information will help the assigned hospital to estimate how many ambulance units and staff to distribute to the scene of the accident – in effect, this will maximise resource allocation and will help to save costs to the hospital.

VII. SOCIAL, LEGAL, AND ETHICAL CONSIDERATIONS

The storage of information about traffic accidents is sensitive in nature. People involved in such accidents may have experienced psychological or physical damage as a result of their collisions, with the more serious cases resulting in life-changing injuries or death. As a result of this, the data that has been collected must be processed lawfully and ethically, while showing utmost respect to persons who can be uniquely identified by instances in the data.

In addition to this, the Data Protection Act 2018 and General Data Protection Regulations (GDPR) govern the retrieval, storage, processing, and transfer of data by an individual or business. If the emergency services are to use the trained models and the data set in their business plans, then compliance to these policies and regulations is crucial. Failure to comply can result in fines and occasionally, prison sentences.

Assumptions made in the data have to be substantial and reasoned and should avoid discrimination or stereotyping of people from differing genders, religion, country, age, etc. Businesses that use these models will have a social responsibility to ensure the data and/or models are by no means offence either directly or indirectly and should be able to provide reasoning as to why assumptions have been made.

VIII. REFERENCES

Department for Transport, 2017, *Reported road causalities Great Britain 2017* [online] Available from: <https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/744078/infographic-2017-annual-report.pdf> [accessed on 29/11/2018]

Niklas Donges, 2018, *The Random Forest Algorithm* [online] Available from: <https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd> [accessed on 30/11/18]

Rohith Gandhi, 2018, *Support Vector Machine – Introduction to Machine Learning Algorithms* [online] Available from: <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47> [accessed on: 30/11/18]

So Young Sohn, Sung Ho Lee, 2003, *Data fusion, ensemble and clustering to improve the classification accuracy for the severity of road traffic accidents in Korea* [online] Available from: <https://ac.els-cdn.com/S0925753501000327/1-s2.0-S0925753501000327-main.pdf?_tid=05773b5a-a352-4813-b3d1-73b63452f508&acdnat=1543506446_086422202419f957021471f44177a840> [accessed on: 29/11/2018]

Vik Paruchuri, 2015, *K nearest neighbours in Python: A tutorial* [online] Available from: <https://www.dataquest.io/blog/k-nearest-neighbors-in-python/> [accessed on: 30/11/18]

Zhuoning Yuan et al, 2017, *Predicting Traffic Accidents Through Heterogeneous Urban Data: A Case Study* [online] Available from: <http://urbcomp.ist.psu.edu/2017/papers/Predicting.pdf> [accessed on 30/11/18]

IX. Appendix

* Original features in the dataset

'Accident\_Index', 'Location\_Easting\_OSGR', 'Location\_Northing\_OSGR',  
'Longitude',  
'Latitude',  
'Police\_Force',  
'Number\_of\_Vehicles', 'Number\_of\_Casualties',  
'Date',  
'Day\_of\_Week',  
'Time',  
'Local\_Authority\_(District)', 'Local\_Authority\_(Highway)', '1st\_Road\_Class',  
'1st\_Road\_Number',  
'Road\_Type',  
'Speed\_limit',  
'Junction\_Detail',  
'Junction\_Control',  
'2nd\_Road\_Class',  
'2nd\_Road\_Number', 'Pedestrian\_Crossing-Human\_Control', 'Pedestrian\_Crossing-Physical\_Facilities', 'Light\_Conditions', 'Weather\_Conditions', 'Road\_Surface\_Conditions', 'Special\_Conditions\_at\_Site', 'Carriageway\_Hazards', 'Urban\_or\_Rural\_Area', 'Did\_Police\_Officer\_Attend\_Scene\_of\_Accident',  
'LSOA\_of\_Accident\_Location',  
'Year',  
'Accident\_Severity'

* Screenshots of the code

