IBM Data Science Capstone Project
Car Accident Severity

Kanakala Yashwanth Sai

10th October 2020

Abstract

It is important to reflect on the stark human and economic cost of road traffic accidents, the first order consequences of which can include property damage, personal injury or death. Key aggravating factors in determining the likelihood and severity of a road traffic accident can include – but are not limited to – current weather, local road/visibility conditions, time of day and individual negligence (e.g. driving under the influence of alcohol/narcotics, driving without due care and attention, driving at excessive speed, etc). In this report I will use Machine Learning techniques to probe publicly-available data for 221,006 road traffic accidents over the past 17 years in the Seattle City Council area in order to model the relative prevalence and influence of these factors in the severity of road traffic accidents recorded in the City. It is envisaged that this model will allow road planners in the city to identify the conditions which cause severe accidents and influence their future road construction in response, and aid the decision-making of Emergency Services call handlers/dispatchers, who will be able to better-predict the severity of car accidents based on the information they receive when the accident is first reported.

1 Introduction

1.1 Context

Road traffic accidents are a major source of human and economic hardship in most advanced economies, with consequences which can range from minor property/vehicular damage, to major damage, personal injury or death. It is estimated that road traffic accidents cost the United States' economy $\sim \$810$billion per year, including costs due to property/vehicular damage, legal costs, medical bills and loss of earnings [1]. It is therefore of paramount importance that we understand the factors influencing the severity of road traffic accidents, in order that road/city planners can devise strategies to mitigate accident severity, and in order to target the efforts and resources of emergency first responders.

1.2 Problem

Intuitively, we might expect that some of the factors which influence the likelihood and severity of a road traffic accident include: the weather, local road contitions (i.e. highways, urban areas or rural roads), time of day (and the presence or absence of street lights), and the number and type of vehicles in the area. Additional factors which may influence the severity of road traffic accidents include those which can be traced to individual negligence, such as driving under the influence of alcohol/narcotics, driving without due care and attention or driving at excessive speed. While it is intuitive that a combination of these factors may be important, intuition alone cannot determine the relative significance of these factors. Determining the relative significance of these competing factors is necessary if we are to fully understand the causes of road traffic accidents and devise new strategies to minimise their occurrance and severity.

1.3 Relevant parties

The main target audiences for this work will be road/city planners and emergency service responders. For instance, it is hoped that by identifying and understanding the key factors influencing the severity of accidents, it may be possible for local authorities to mitigate against these factors. This could involve identifying key accident "black spots" in a city/region in order to introduce traffic-calming measures, or deciding whether to target resources at advertising campaigns to encourage safer driving (e.g. anti speeding or anti drink-driving campaigns). Furthermore, a model which can predict accident severity based on local and geographical conditions may be of use to emergency service call handlers, as it will enable them to prioritise the allocation of emergency services resources (i.e. fire department, police and ambulances) based on the information that is available at the time the accident is reported. They will be able to make a quick, evidence-based assessment of whether it is more appropriate to send light resources (motorcycle police/ambulances) or heavy resources (police squad cars, full-sized ambulances and fire trucks) as the first response to an accident, based on its predicted severity.

2 Data

2.1 Data Acquisition

Data were obtained for all road traffic accidents recorded in the Seattle municipal area between Jan 2004–Aug 2020 by the Seattle Department of Transport (SDOT). The data were obtained from the Seattle Open Data Portal (SODP: [2]) in Comma Separated Value (CSV) format and read in to a Pandas Dataframe for analysis using the Pandas read_csv function. In total, the dataset comprises 221,006 rows (one for each road traffic accident in Seattle during this period) and 40 providing information about the accident, including the accident severitycode (i.e. the target variable for this analysis). Further information about the properties of the dataset (and the analysis thereof) is available in an online Jupyter Notebook [3], however a list of the columns present in the dataset is shown in Table1 along with a brief description of each column's meaning. A full glossary of headings in the table is available online at SDOT [4]. Furthermore, to illustrate the format in which the raw data are recorded, the first row of the table is shown in Table2.

The target/dependent variable is severitycode which, in its original form, takes the values 0, 1, 2, 2b or 3. The definitions of these severity codes are provided in the "Attribute Information" metadata which accompany the data release [4] and are as follows:

- **0:** Unknown
- **1:** Property/vehicular damage
- **2:** Minor injury
- **2b:** Serious injury
- **3:** Fatality

2.2 Data Cleaning

In its original form, this dataset is not suitable for quantitative analysis. There are five principal reasons for this, which are explained in the following subsections.

2.2.1 Missing target data

The target variable is severitycode, however nearly 20,000 accidents (9.8% of the dataset) are missing this vital information. As the purpose of the model is to predict the severity of an accident from the other features in the dataset, clearly accidents with no value for this vital variable are of no use to us, and need to be excluded.

2.2.2 Missing predictor variables

The dataset contains missing entries, where one or more of the key predictor variables are absent or uninformative (e.g. 6.8% of accidents have “Unknown” listed in the weather column). Including these data entries in the model is likely to bias the model, and so we drop the affected rows. Some of the Y/N columns (e.g. pedrownotgrt, speeding) also have missing data: in these instances, we infer that if the condition was true, then it would have been noted in the original report and recorded as “Y”. We therefore infer that any accident with null entries in these columns is equivalent to “N”.

2.2.3 Dataset includes unnecessary columns

A number of columns in the dataset are superfluous (i.e. they contain information which is unrelated to the causes or severity of accidents) or are redundant (i.e. they simply replicate information which is already encoded in other columns). Examples of superfluous columns include objectid, inckey and coldetkey, which all identify the accident records with respect to other data held by SDOT which are not included in this dataset. Examples of redundant columns include severitydesc (which provides a textual description of the accompanying severitycode) and sdot_colcode/sdot_coldesc (which replicate the

Column Name	pandas data type	Description
X	float64	Longitude (deg.)
Y	float64	Latitude (deg.)
OBJECTID	int64	ESRI unique identifier
INCKEY	int64	Unique key for the incident
COLDETKEY	int64	Secondary key for incident
REPORTNO	object	Report number for incident.
STATUS	object	Meaning not defined (mostly null)
ADDRTYPE	object	Collision address type
INTKEY	float64	Identifier of intersection
LOCATION	object	Description of collision location
EXCEPTRSNCODE	object	Meaning not defined (mostly null)
EXCEPTRSNDESC	object	Meaning not defined (mostly null)
?SEVERITYCODE	object	Accident severity code: 0, 1, 2, 2b, 3
SEVERITYDESC	object	Description of severitycode

COLLISIONTYPE	object	Description of collision
PERSONCOUNT	int64	Number of people involved
PEDCOUNT	int64	Number of pedestrians involved
PEDCYLCOUNT	int64	Number of cyclists involved
VEHCOUNT	int64	Number of vehicles involved
INJURIES	int64	Number of minor injuries in accident
SERIOUSINJURIES	int64	Number of major injuries in accident
FATALITIES	int64	Number of fatalities in accident
INCDATE	object	Date of accident
INCDTTM	object	Date and time of accident
JUNCTIONTYPE	object	As addrtype: description of road type.
SDOT_COLCODE	float64	SDOT collision code decribing accident
SDOT_COLDESC	object	Description of SDOT_COLCODE
INATTENTIONIND	object	Whether collision was due to inattention (Y/N)
UNDERINFL	object	Whether one or more drivers were intoxicated
WEATHER	object	Description of weather conditions
ROADCOND	object	Description of road surface conditions
LIGHTCOND	object	Description of light conditions
PEDROWNOTGRNT	object	Whether collision involved a breach of pedestrian right of way (Y/N)
SDOTCOLNUM	float64	Unique key given by SDOT to incident
SPEEDING	object	Whether speed limit breached or not (Y/N)
ST_COLCODE	object	State collision code describing accident
ST_COLDESC	object	Description of ST_COLDESC
SEGLANEKEY	int64	Key for lane segment where accident occurred
CROSSWALKKEY	int64	Key for crosswalk where accident occurred
HITPARKEDCAR	object	Whether accident involved a parked car (Y/N)

Table 1: Description of column names and contents in SDOT Road Collisions database. Further information is available at the Seattle Open Data Portal ([2]). ?SEVERITYCODE is the target variable.

Column Name	Value
X	-122.34
Y	47.6254
OBJECTID	1
INCKEY	333240
COLDETKEY	334740
REPORTNO	3851889
STATUS	Unmatched
ADDRTYPE	Intersection
INTKEY	28743
LOCATION	9TH AVE N AND ROY ST
EXCEPTRSNCODE	*Null*

EXCEPTRSNDESC	NaN
[?]SEVERITYCODE	2
SEVERITYDESC	Injury Collision
COLLISIONTYPE	NaN
PERSONCOUNT	2
PEDCOUNT	0
PEDCYLCOUNT	0
VEHCOUNT	0
INJURIES	2
SERIOUSINJURIES	0
FATALITIES	0
INCDATE	2020/08/10 00:00:00+00
INCDTTM	8/10/2020
JUNCTIONTYPE	At Intersection (intersection related)
SDOT_COLCODE	11
SDOT_COLDESC	MOTOR VEHICLE STRUCK MOTOR VEHICLE, FRONT END ...
INATTENTIONIND	NaN
UNDERINFL	NaN
WEATHER	NaN
ROADCOND	NaN
LIGHTCOND	NaN
PEDROWNOTGRNT	NaN
SDOTCOLNUM	NaN
SPEEDING	NaN
ST_COLCODE	*Null*
ST_COLDESC	NaN
SEGLANEKEY	0
CROSSWALKKEY	0
HITPARKEDCAR	N

Table 2: First row of the dataframe, illustrating the typical format of the entries in the SDOT Road Collisions database. [?]SEVERITYCODE is the target variable.

information that is encoded in the st_colcode column). Columns which are not to be included in building/testing our model should be dropped.

Moreover, as the objective of this work is to construct a model for predicting severitycode, it is imperative that we exclude any columns which are implicit in the definition of the target variable. severitycode is determined according to whether or not an accident involves injury or death. To include these columns in the model would risk training a model which is only tautologically valid. We therefore must drop the columns injuries, seriousinjuries and fatalities from the dataset.

2.2.4 Presence of categorical data

Some columns, contain non-numerical data. In the case of weather, this corresponds to one of eleven text strings which define the weather conditions at the time of the accident (e.g. “clear”, “rain” “snow”). Machine learning models cannot work directly with categorical data, and so for this reason it will be necessary to re-cast these columns as numerical data via one-hot encoding using the pandas function pd.get_dummies(). This function can be used to add eleven new columns to the dataframe – one for each distinct weather type – populated with 1s or 0s, depending on the recorded weather type. Afterwards, the original column of categorical data (weather) is dropped from the dataframe[1].

Other columns do contain numerical data, but numerical data which do not correspond to any kind of sequence. For example, sdotcolcode, the SDOT collision code, is an integer which can take values from 0–84, but it does not make sense to think of, for example, sdotcolcode=62 (Vehicle Struck By Country Road or Construction Machinery) as being in any sense “greater” than sdotcolcode=43 (Vehicle Struck Stopped Train). In truth, this column corresponds to nominal categorical data, and thereforen was re-cast using onehot encoding.

Finally, some columns (e.g. speeding) contain binary data in string format (Y/N). In order to make these columns useful for model-building it will be necessary to transform these columns to numerical data: Y→1 and N→0. For the same reason it will also be necessary to re-cast the target variable itself (severitycode) from its original alpha-numeric state to a column of integer values: 0, 1, 2, 2b, 3 → 0, 1, 2, 3, 4. As previously stated, accidents with severitycode=0 correspond to an unknown accident outcome, and so these entries are dropped from the dataset before analysis.

2.2.5 Data are imbalanced and non-standardised

The numerical data are highly imbalanced: there are $\sim 345\times$ as many accidents with severitycode=1 as there are accidents with severitycode=4. If we train a model to predict accident severity using a dataset in which the vast majority of the accidents have one particular outcome, then it is highly likely that the model itself would be biased, and would preferentially classify unknown accidents with the dominant severitycode. To avoid this issue, we down-sample the dataset by randomly selecting N_4 accidents with severitycode=1, 2 or 3, where N_4 is the number of accidents with severitycode=4 (i.e. fatal accidents).

Furthermore, the data ranges are not standardised. For instance, after numerical encoding of binary data and one-hot encoding of nominal categorical data, most of the columns in the dataset have values 0 or 1. Other columns, however (such as longitude, x and latitude y) have numerical values which are two orders of magnitude larger. Fitting a model with such non-standardised data is likely to yield spurious results, as most machine learning algorithms assume that all features have numerical values centred around 0, with unit variance. After all other data cleaning steps were performed and an exploratory analysis of the dataset conducted, the features which were to form the basis of model-building were be standardised using standardscaler function from the sklearn.preprocessing package.

[1] A test of this approach was performed wherein instead of using one-hot encoding to create additional columns filled with 1s and 0s, the orignal weather column was re-encoded with an integer scale of values ranging from 0–3 for weather conditions of increasing severity, e.g. “Clear” and “Overcast”→0, “Rain”→1 and more extreme weather events (e.g. “Severe crosswind” and “Blowind Sand/Dirt”) were given values of 2 and 3. The precision and recall of the resulting models were nearly indistinguishable from those of the models produced using one-hot encoding.

2.3 Feature Selection

The dataset contains a column of type object which corresponds to the date and time of the incident, incdttm. In order to make use of this information, the pandas function to_datetime was used, and the original incdttm column was replaced with separate columns giving the integer day, date of month, month and year in which the accident occurred, as well as the time of the accident (rounded to the nearest hour).

After incorporating this temporal information, further inspection of the cleaned data showed that a number of columns were missing so much data that they had little predictive power, e.g. seglanekey, crosswalkkey, etc. These columns were therefore not used for model-building.

The final feature set used to predict severitycode therefore includes the following columns:

- x,y, the latitude and longitude of the accident
- pedcount, pedcyclcount, vehcount, which indicate how many pedestrians, cyclists or vehicles are involved
- inattentionind, underinfl, pedrownotgrnt, speeding, which indicate if irresponsible driving is a factor
- Data from weathercond, which are expanded to 11 columns describing weather conditions using one-hot encoding
- Data from lightcond, which are expanded to 8 columns using one-hot encoding
- Data from roadcond, which are expanded to 5 columns using one-hot encoding
- block, intersection, which indicate the type of road on which an accident occurs • Data

  from sdot_colcode, which are expanded to 40 columns using one-hot encoding
- Data from incdttm, which are reduced to:
  - month, a number between 1–12 giving the month of the year
  - weekend, a binary variable indicating whether the accident took place on a week day or at the weekend
  - hour_nearest, giving the time of day when the accident occurred, rounded to the nearest hour

After these final data cleaning, balancing and standardisation operations were performed, we are left with a dataset consisting of 1308 rows and 61 columns.

3 Exploratory data analysis

Before we begin building our model, it is useful to plot some of the key features of the dataset in order to gain an intuitive understanding of the Seattle car accident database.

We begin by mapping out the locations at which accidents occur, as well as the average accident severity at different locations in the city. Geographical data giving the boundaries of Seattle's distinct neighbourhoods were obtained in geojson format from the *seattle.io* github repository [5], and the python package shapely was used to determine which (polygonal) neighbourhood boundary each accident occurred within, given the longitude and latitude (x, y) information contained in the dataset.

A map of the Seattle metropolitan area was drawn-up using the folium package, and a choropleth layer added to represent the total number of accidents in each neighbourhood. The resulting map is shown in Fig.1. We see that road accidents are common in regions like the University District, near Seattle's Ship Canal Bridge (a key choke-point in the road network crossing the Lake Washington Ship Canal). Other areas with high accident rates are the Industrial District and Industrial District West, which straddle the Duwamish Waterway, and are again connected via a major bridge. We may therefore conclude that the areas around bridges are accident blackspots, possibly due to congestion associated with funnelling multiple distinct roads in to a single crossing.

In Fig.2 the mean accident severity in each neighbourhood is shown. There appears to be no *strong* dependence on the accident severity with geography, however there are hints that coastal peninsulas (e.g. Magnolia, West Seattle and Laurelhurst) see relatively few major accidents, whereas areas to the North of the city (Broadview, Haller Lake and Lake City) see a higher average accident severity. This may be associated with the increased opportunities to drive quickly in the suburban sprawl compared with the congested city centre.

In addition to showing where accidents occur, it is instructive to investigate the environmental conditions (e.g. road/lighting conditions) in which accidents occur, as well as the dates and times at which they occur. We see (Fig.3) that the number of daily road accidents in Seattle peaked between 2005–2006 at around 40 accidents per day, before droping over the subsequent four years to a plateau at ~ 25 accidents per day between 2010–2019. Moreover, we see quarterly variation in the rate of accidents in Seattle, with a lower accident rate in the spring/summer months (when days are longest) and an increased accident rate in the autumn/winter months (when the days are shortest). Note that Fig.3 appears to show a drastic drop-off in the daily accident rate in 2020. This is almost certainly due to the 2020

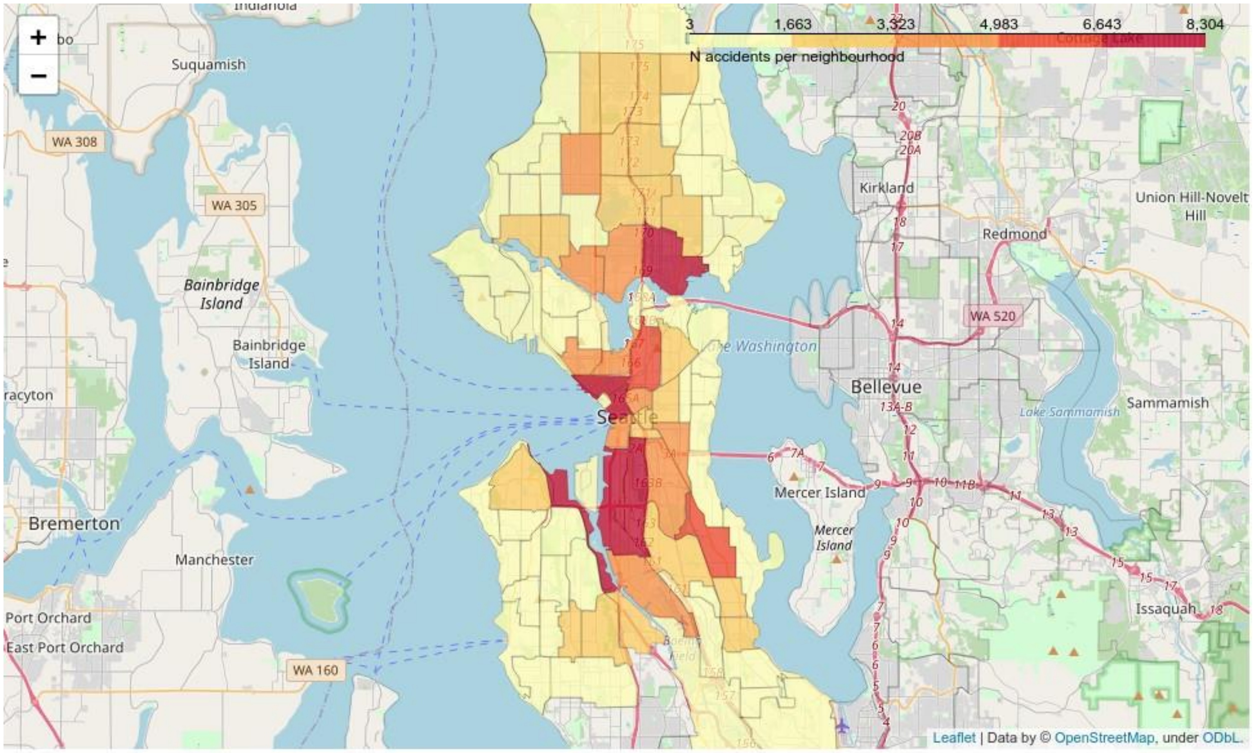

Figure 1: Total number of accidents recorded in each Seattle neighbourhood.

data being incomplete rather than a significant improvement in road conditions. To avoid biasing the model, the dataset used for analysis is terminated at December 2019.

In Figs4–6 the distributions of day of the week on which accidents occur, road conditions and lighting conditions are shown both before and after re-sampling the data as described in §2.2.5. In the raw dataset we see a trend whereby accidents occur least frequently at the weekend and on Mon, and that the daily rate of accidents steadily grows between Tues– Fri. The re-sampled dataset does not completely replicate this trend, having a daily accident rate which plateaus between Wed–Fri. However the higher-level split between weekend versus week day, which we use for training and testing the model, is unchanged by re-sampling. In Fig.5 we see that by re-sampling the data, we lose information corresponding to extremely rare road conditions (e.g. oil on the road). However the rarity of these conditions in the original dataset implies that they would not be good features to build the model around anyway. Importantly, the distributions of the four most common weather conditions are unchanged by this re-sampling, which indicates we have not significantly biased the dataset towards any particular set of weather conditions.

Finally, in Fig.6 we show the distribution of lighting conditions before and after resampling. As expected of random sampling, the relative proportions of the most common lighting conditions ("Daylight" and "Dark – Street Lights On") are kept in balance, with only the extremely rare and uninformative conditions such as "Dark – Unknown Lighting" being dropped from the re-sampled dataset as a result.

The correlation matrix corresponding to the cleaned and re-sampled dataset is shown in Fig.7. We see a number of intuitive and obvious correlations in the data, such as wet road

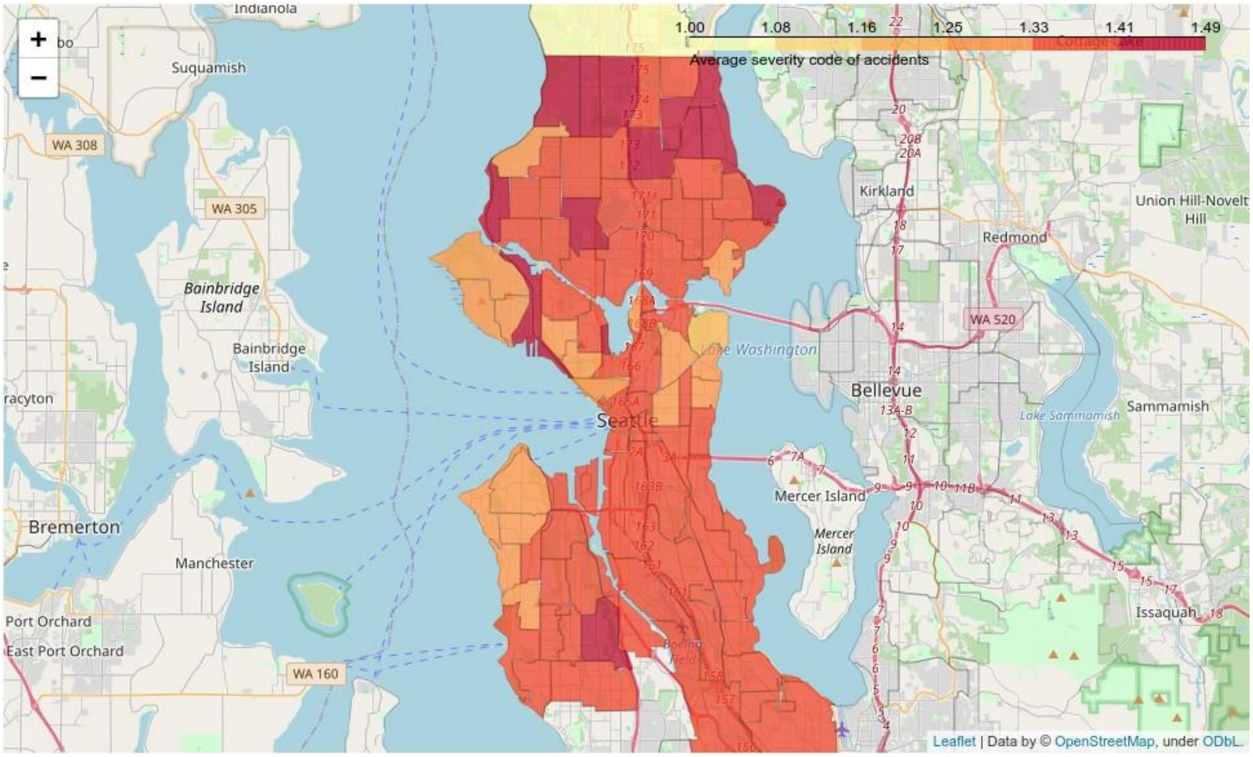

Figure 2: Mean severity of accidents occurring in the Seattle neighbourhood.

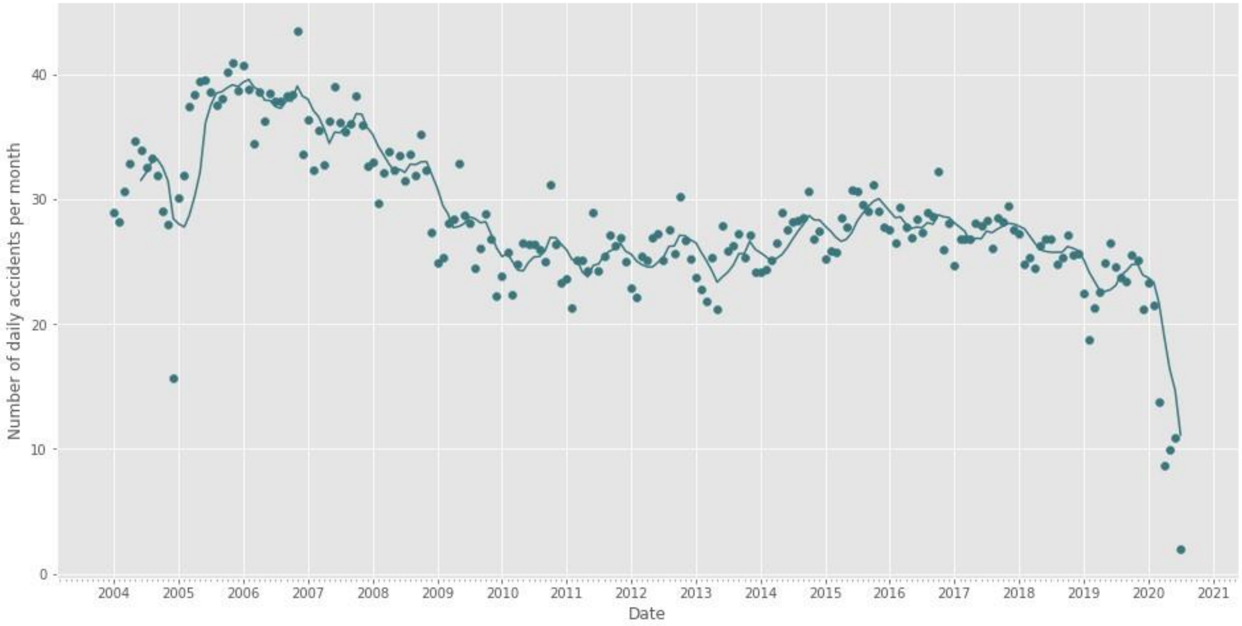

Figure 3: Daily accidents recorded in the SDOT database between 2004-2020, plotted at monthly intervals (green points). The six-month rolling average is shown as a solid green line. We see that the daily accident rate in Seattle peaked between 2005–2006, and has been roughly constant since 2010. We see also regular variations in the accident rate throughout the year. Note that the apparent steep drop-off in 2020 probably reflects delays in updating the database, and is unlikely to be real.

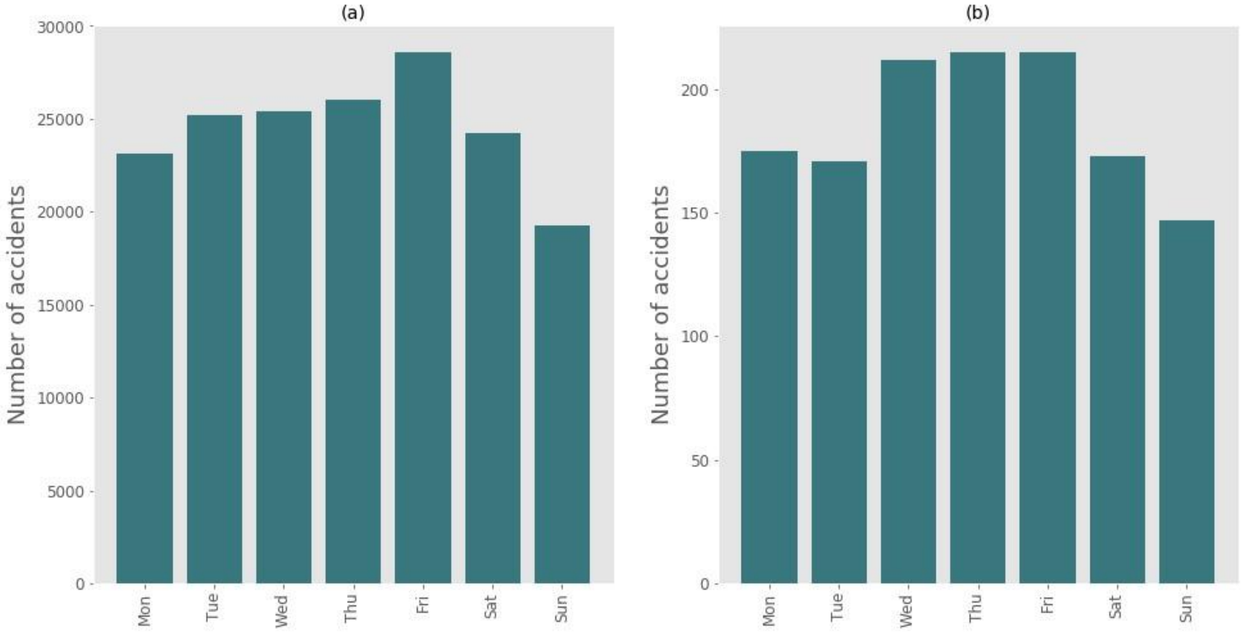

Figure 4: Distribution of the days of the week on which accidents occured before (a) and after (b) resampling.

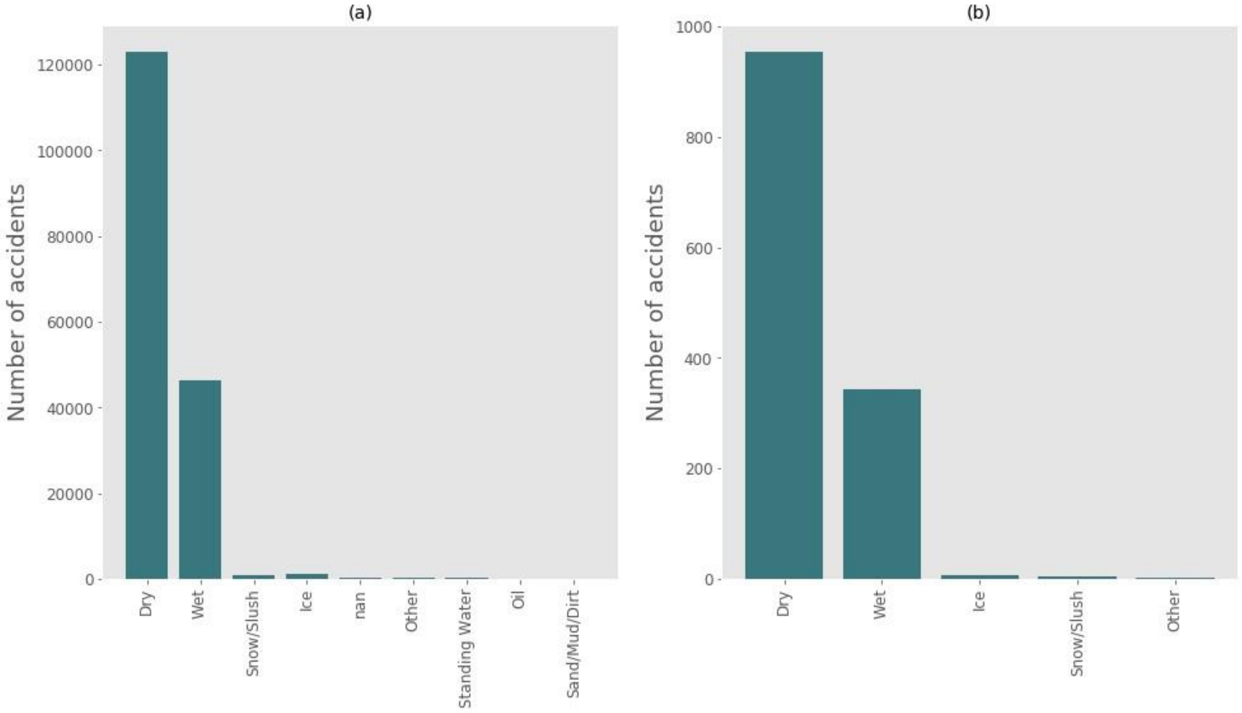

Figure 5: Distribution of road conditions recorded by SDOT before (a) and after (b) re-sampling.

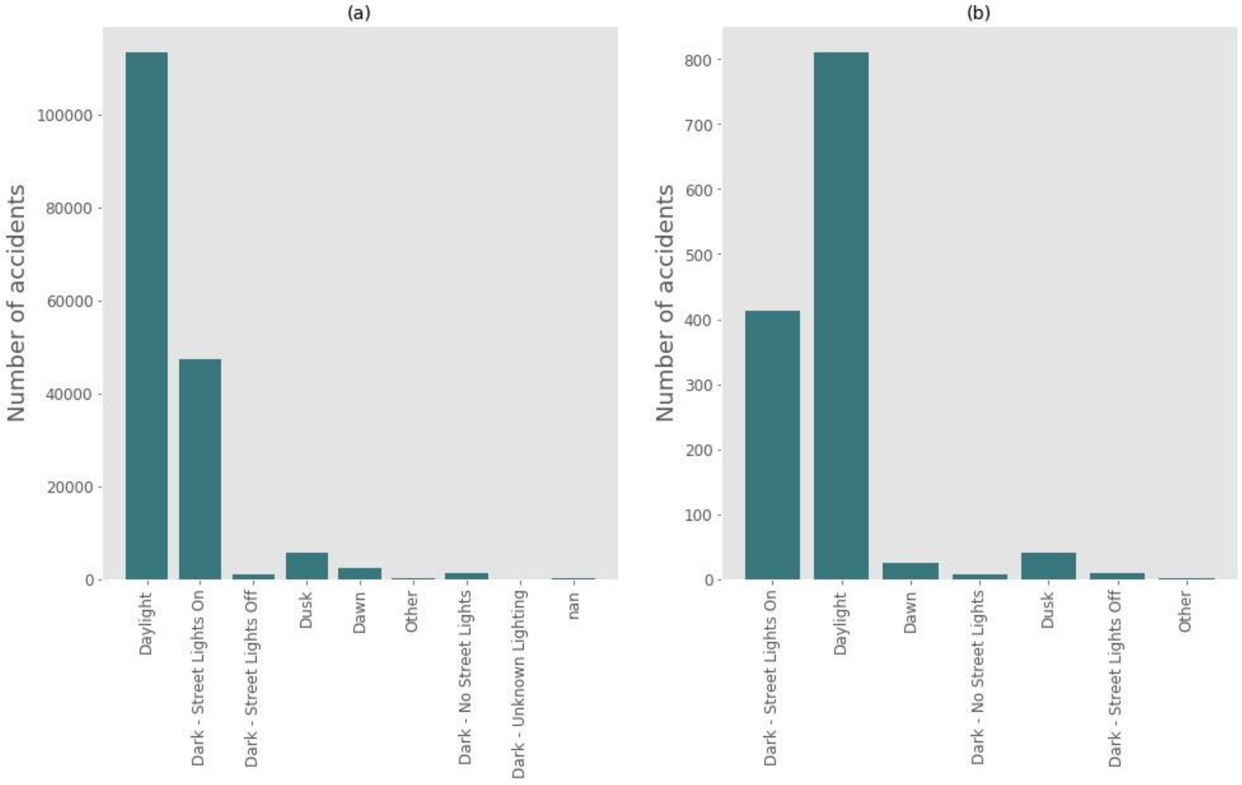

Figure 6: Distribution of lighting conditions recorded by SDOT before (a) and after (b) re-sampling.

conditions correlating with rainy weather conditions. The correlation matrix also provides an “at-a-glance” answer to the question of which types of collisions (encoded in sdot_colcode_XX) correlate most strongly with the severity of an accident. We see, for instance that sdot_colcode_24 (“From Opposite Direction – Both Moving – Head-On”) correlates strongly with severitycode, whereas sdot_colcode_14 (“From Same Direction – Both Going Straight – One Stopped – Rear End”) does not. That these kinds of intuitive correlations are visible in the correlation matrix of the re-sampled data strongly suggests that the feature engineering and data balancing described above has not significantly biased the data. We can be confident that the dataset is now suitable for modelling.

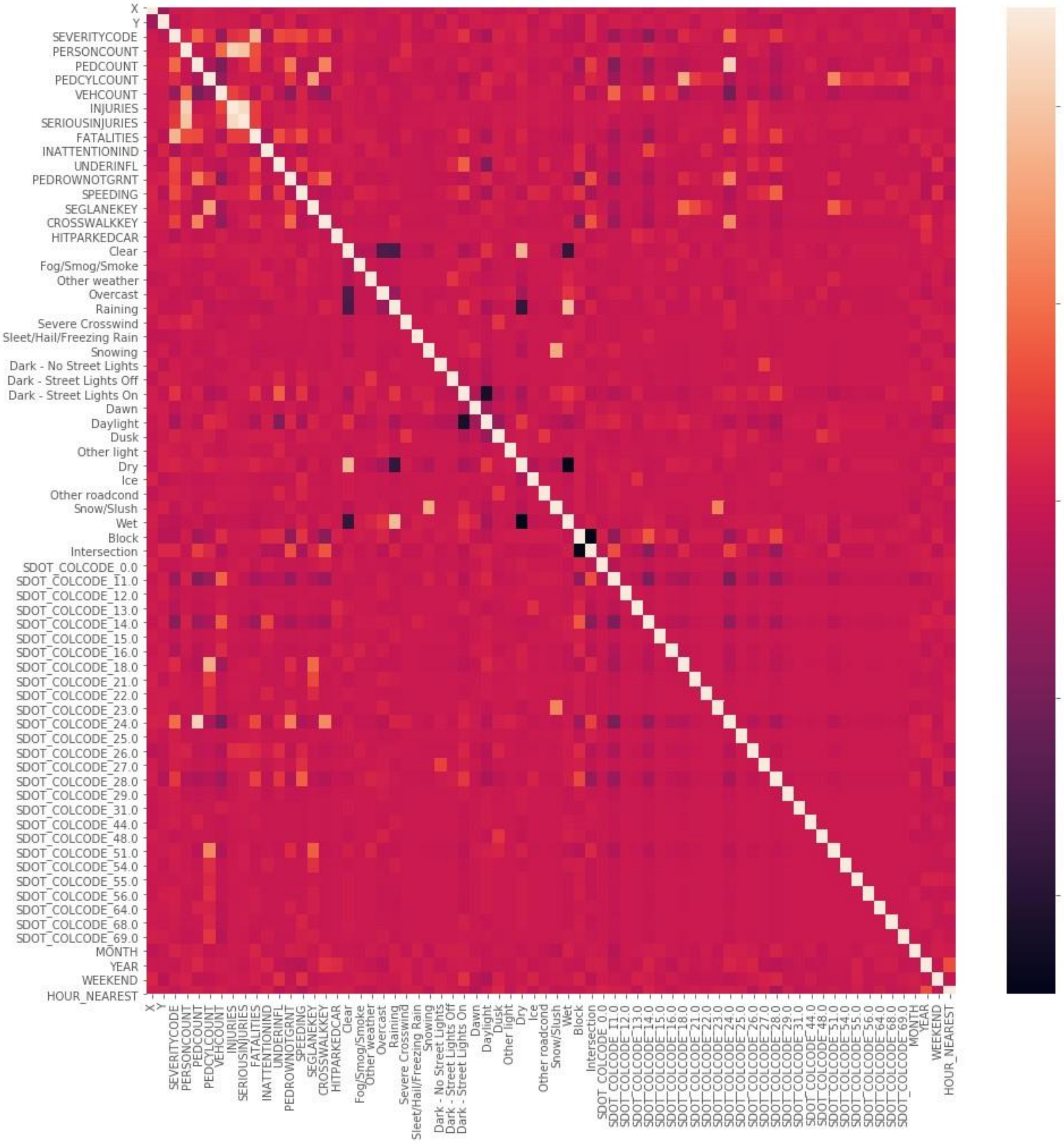

Figure 7: Correlation matrix for the SDOT dataset after cleaning and re-sampling the data. There are some intuitive correlations in the data, such as wet road conditions correlating with rain. The correlation matrix also makes clear

that the accident type with the highest severitycode is sdotcolcode=24, which corresponds to “From Opposite Direction – Both Moving – Head-On”.

4 Predictive modelling

To recap, the primary objective of this analysis is to train and test a model which predicts severitycode, an integer ranging from 1–3 which describes whether an accident involved property damage, injury or death, from information which may be available to emergency service call handlers at the time when an accident is reported. Having cleaned and balanced the data, the final step before modelling can commence is to standardise the feature set using the standardscaler package from scikit learn. standardscaler re-casts every feature in the feature set as having a distribution with zero mean and unit variance. This is an important step to perform in order to ensure that the model is not biased towards or away from certain feature types by the range numerical values assigned to those features.

Finally, the data were split in to training and testing subsets using the train_test_split function from scikit learn. The parameter test_size was set to 0.3, meaning that 70% of the balanced data were used for training the model and 30% of the data were reserved for testing.

Models that we can use to predict car accident severity from the avaiable data fall in to one of two categories: regression models and classification models. Regression models can be used to predict the value of a discrete or continuous target variable from the feature variable(s), while classification models predict the category to which a datum belongs given the information in its feature set. The Seattle accident data have been cleaned and engineered in such a way that we can build models belonging to either category to predict severitycode.

4.1 Logistic Regression model

Having converted severitycode into a binary variable (0 for no-injury and minor-injury collisions, 1 for major injury and fatality collisions) we can begin our model building with a simple logistic regression model to predict which of the two accident severity codes a given accident from the test set belongs to. Logistic regression (LR) models use the logistic function to model the probability of a binary target variable belonging to either class.

A logistic regression model was built from the training set using scikit learn with a regularisation value $C = 0.01$. The model was used to predict the severitycode for accidents in the test set, and these predicted values were compared with the (known) severitycodes of the data in the test set. The confusion matrix providing insight into the accuracy of the model is shown in Fig.11. We see that 70% of the time the model correctly predicts severitycode=1, and 84% of the time the model correctly predicts severitycode=0. The $F1$ score (the harmonic mean of the model’s precision and recall) is 0.77 for severitycode=0 and 0.76 for severitycode=1 in the test subset, indicating that the model correctly predicts the accident severity based on the available featues for around three quarters of accidents.

A comparison of the performance of this model with that of the other models will be given in §4.5.

4.2 Decision Tree model

The next model that we build and test is a decision tree model. Decision tree (DT) models are built by iterating thorugh the available feature set to identify which features – as well as which thresholds of those features – most cleanly separate the sample on the target variable (severitycode). The objective is to find the feature which most cleanly separates the sample between severitycode=0 and severitycode=1 within the parent sample, and then from each of these two “branches” identify the feature which most cleanly separates the data subsets, and so on. The objective is usually to branch the tree until every “leaf” contains only severitycode=0 or severitycode=1, i.e. has no disorder/entropy. However if there are many features in the feature set, the compute power needed to branch the decision tree down to zero entropy may be subject to the law of diminishing returns, and so in practice a maximum branching depth is sometimes specified. One of the key attractions of building a decision tree model is that it provides easy insight into which features in the feature set most cleanly separate the target variable between categories.

A decision tree model was built for the Seattle accident dataset using the decisiontreeclassifier in scikit learn, using the “entropy” criterion. As the cleaned and balanced dataset is relatively small ($1308 \text{ rows} \times 61 \text{ columns}$), no depth limit was imposed and the decision tree classifier ran until every leaf was pure (even if that means splitting until there is only one datum in the leaf). The final decision tree requires 30 layers of branching depth to achieve leaf purity. At the initial branching level, a cut on vehcount is found to split the sample into sub-samples a and b with entropies of 0.787 and 0.89. At the second branching level, branch a is split on pedcount in to sub-branches u and v, with entropies of 0.911 and 0.604, respectively. Meanwhile, branch b is split on speeding into sub-branches x and y, with entropies of 0.831 and 0.870, respectively. Sub-branches u, v, x and y are then split on additional features, and so on until leaf purity is achieved.

The first three layers of the decision tree from the parent sample are shown in Fig.8, and the model’s confusion matrix is shown in Fig.11. The decision tree model correctly predicts accidents with severitycode=0,1 68% and 65% of the time, respectively. The $F1$ scores for severitycode=0,1 are 0.65 and 0.67, respectively.

4.3 Support Vector Machine model

Support Vector Machine (SVM) is a form of supervised learning model which is used for data classification and regression analysis. SVM models seek to separate data on the target variable by mapping the dataset to a higher-dimensional space and hyperplanes which most cleanly separate the data in this higher-dimensional space.

An SVM model was built from the car accident dataset using the C-Support Vector Classification method (sklearn.svm.svc), with a linear mapping kernel. By choosing a linear mapping kernel, it is possible to de-project the best-fit support vectors back to the hyperspace of the original dataset, allowing the most constraining features to be ranked (Fig.9). Like the decision tree classifier, SVM determines that the number of pedestrians involved is a major indicator of the accident severity, as is excess speed. Unlike the decision tree, however, SVM finds the collision code to be a particularly good predictor of accident severity, with 5 of the top 10 most significant features being from the set of one-hot encoded

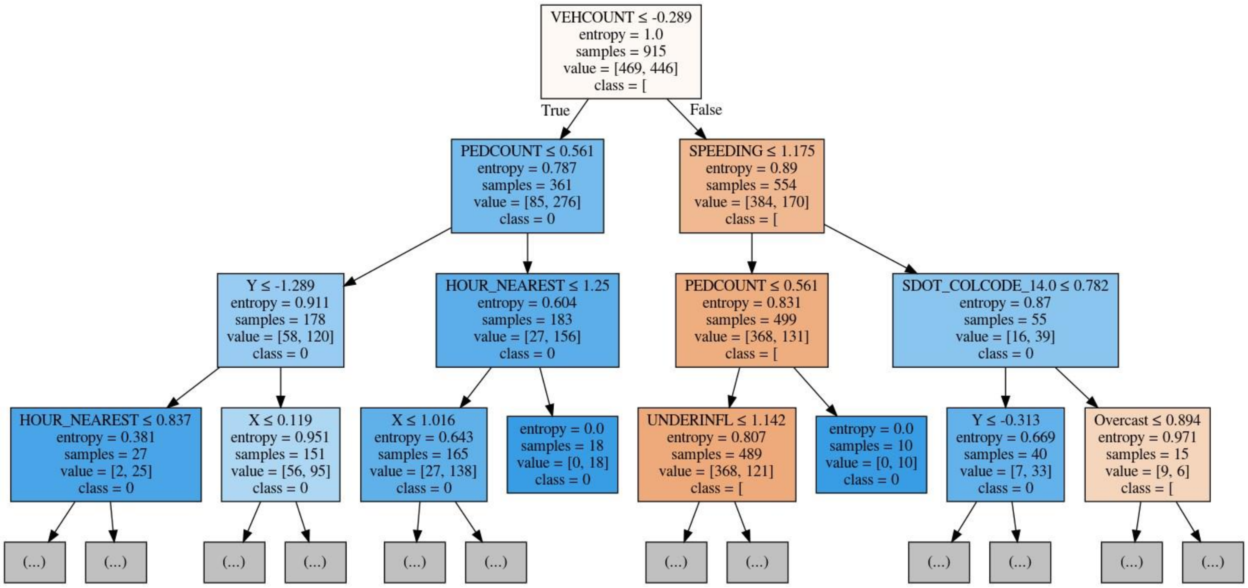

Figure 8: The top three branch levels of the decision tree classification model. We see that the decision tree determines vehcount to be the feature with the most diagnostic power over the accident severity at the cardinal level. Depending on which subset one investigates next, either pedcount or speeding are the next most useful features for classification. At the third level, the features y (longitude), hour_nearest, pedcount and sdot_coldode=24 are the most useful (sdot_colcode=24 refers to a head-on accident between two moving vehicles).

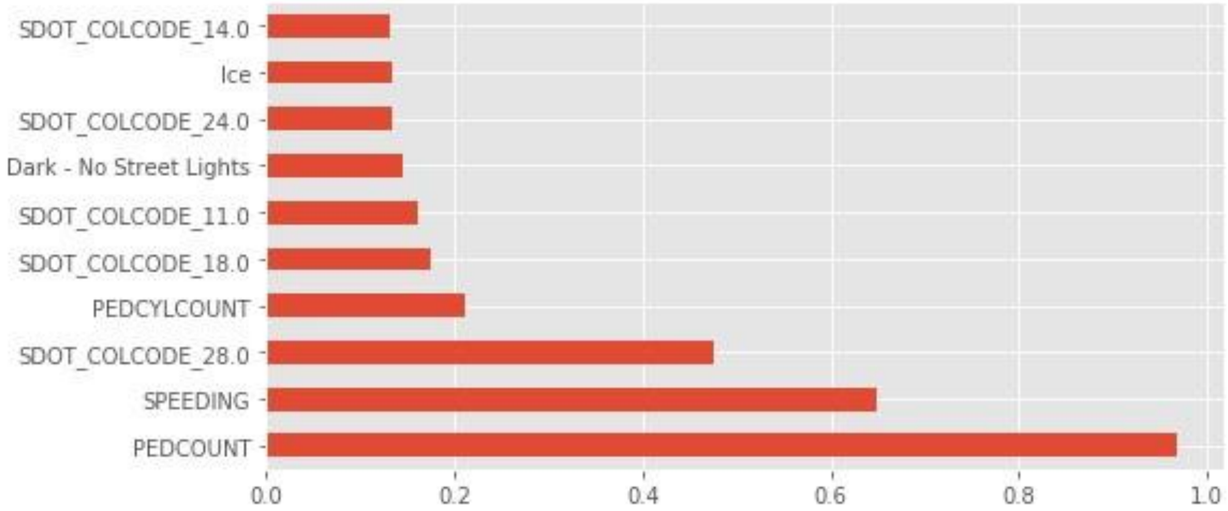

Figure 9: Ranked list of the top-ten most constraining features for determining accident severity from the SVM model, ordered from least to most-significant.

SDOT collision codes. Two of the collision codes associated with the most severe outcomes (24 and 28) refer to different types of head-on collisions, while codes 11 and 18 refer to side-swipe collisions. Collision code 14 refers to rear-end collisions, where the car in front is stationary. While icy road conditions contribute to a tiny fraction of the total number of accidents in the Seattle accident database (0.56%), SVM is nevertheless able to identify ice as a major contributor to the *severity* of road accidents.

The precisions of the SVM model for predicting accident severitycode=0,1 is 0.82 and 0.80, respectivley, and the $F1$ scores of the model for both categories are 0.76 and 0.75.

4.4 k-Nearest Neighbour model

The final model which we use in our analysis of the Seattle accident database uses the kNearest Neighbour (kNN) algorithm. kNN is a pattern-recognition algorithm which maps an input dataset to a multi-dimensional hyperspace and then attempts to classify a data point of unknown classification based on the classifications of is k-nearest neighbours in this hyperspace. The optimum choice of k is highly dependent on the dataset in question, and in practice it is usually necessary to train and test kNN models using a range of k, measuring the accuracy of each (i.e. the percentage of labels that are correctly categorised in the testing subset). Training the model with using too few neighbours (low k) increases the likelihood of chance matches, creating unstable decision boundaries, whereas training the model with too many neighbours (high k) can rob the model of discriminatory power. While there is no *a priori* method of cho$\sqrt{\text{os}}$ing the best k, it is often the case that the model's predictive power is optimised for $k \sim N$, where N is the number of samples in the training dataset. kNN models were built for $k = 1$–300 using the kneighborsclassifier in scikit learn. Aside from updating k after each iteration of model fitting and evaluation, all other parameters were left at their defaults (see scikit learn user manual for more details). The resulting model accuracy as a function of k is shown in Fig.10. We see that the model accuracy does not peak sharply at a given k, but oscillates at an accuracy of $\sim 73\%$ for $k = 5$–30. The model with the highest accuracy is that at $k = 25$, which is at least of similar order-of-magnitude to the square root of 915, the number of samples in the training set.

The confusion matrix for the SVM model (Fig.11) highlights that the kNN approach correctly categorises severitycode=0 73% of the time, and correctly categorises severitycode=1 66% of the time. The $F1$ scores of the two categories are 0.69 and 0.70, respectively, suggesting that the precision and recall of the model are comparable.

4.5 Comparative model performance

We now summarise the comparative performances of each of the four machine learning models in terms of correctly categorising severitycode based on the feature set avaiable to describe each accident. The confusion matrices for all four models are shown in Fig.11. We see that both the Logistic Regression and Support Vector Machine models correctly identify serious accidents (severitycode=1) 70% of the time, while the Decision Tree and k-Nearest Neighbour models correctly predict severitycode=1 accidents 64% and 62% of the time, respectively. Conversely, the kNN model predicts low-severity accidents (severitycode=0) with the highest accuracy (87%), while the LR and SVM models have

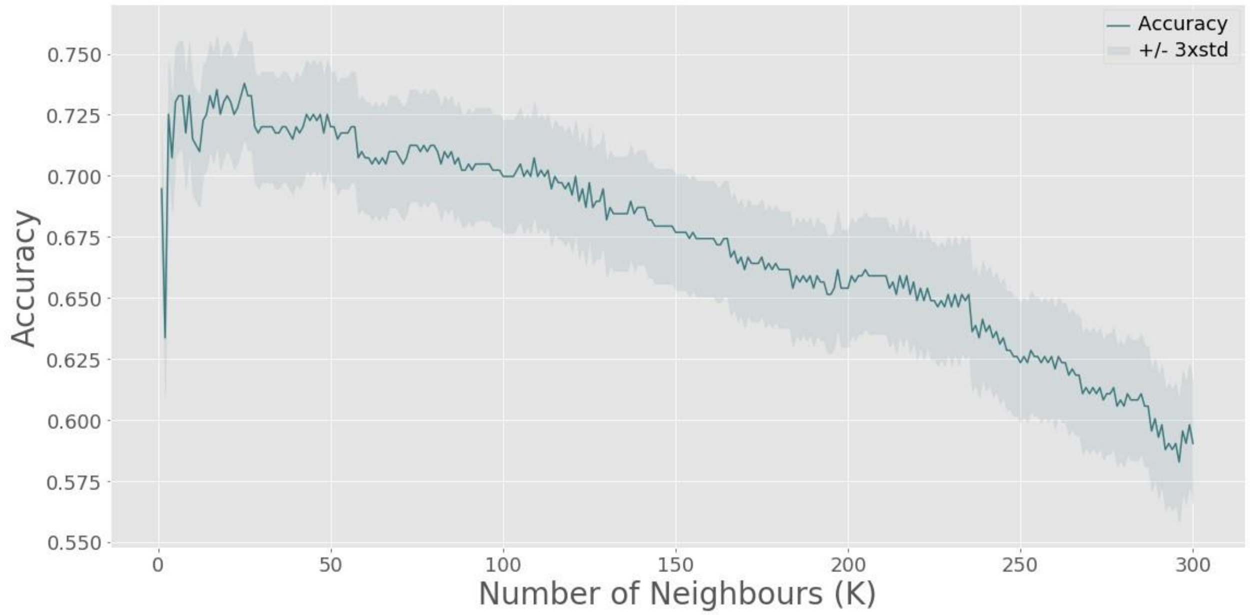

Figure 10: A comparison of the kNN model accuracy for $1 \le k \le 300$. We see that the model accuracy is optimised for $k=25$, and hence fit the optimal kNN model using the 25 nearest neighbours for each datum.

Model name	F1 Score	Jaccard Score	Notes
Logistic Regression	0.77	0.77	–
Decision Tree	0.67	0.66	Most constraining features: No. vehicles/No. pedestrians/Speeding
Support Vector Machine	0.76	0.76	Most constraining features: No. pedestrians/Speeding/Head-on (Y/N)
k-Nearest Neighbours	0.75	0.74	Performance optimised at k = 25

Table 3: Summary of the performance metrics of the four Machine Learning models trained and tested on the Seattle accident dataset.

an accuracy of 84% and 82%. The DT model predicts severitycode=0 accidents with the lowest accuracy (69%).

In Table3 the $F1$ score and Jaccard Similarity Index for each model is given. The LR and SVM models have the highest $F1$ scores, 0.77 and 0.76, respectively. The $F1$ score of the kNN model is comparable, being 0.75. The DT model performs most poorly, having an $F1$ score of 0.67. The Jaccard Similarity scores for the four models (defined as the fractional overlap in the true/predicted labels of the test data, given the model) are 0.77, 0.76, 0.74 and 0.66, respectively.

One key advantage of the SVM model over the LR and kNN models (notwithstanding their comparable $F1$ and Jaccard scores) is that the SVM model returns a ranked list of the most influential features in the feature set for determining accident severity.

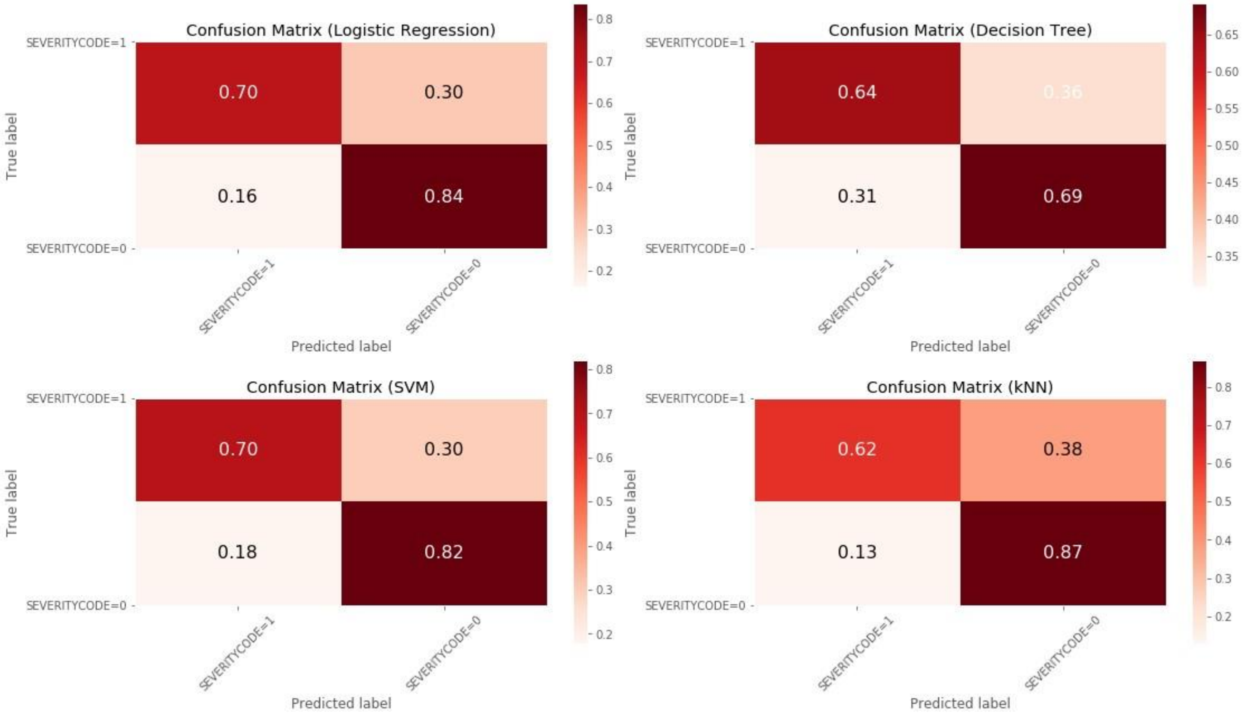

Figure 11: Confusion matrices for the Logistic Regression, Decision Tree, Support Vector Machine and k-Nearest Neighbour models. The kNN model has the highest accuracy for severitycode=0 accidents (87%), but only the third-highest accuracy for predicting severitycode=1 accidents (62%). The SVM model has the joint-highest severitycode=1 accuracy (70%) and the third-highest severitycode=0 accuracy (82%). The Logistic Regression model has the joint-highest severitycode=1 accuracy (70%) and the second-highest severitycode=0 accuracy (84%). The Decision Tree model performs worst for both severitycodes.

5 Conclusions and Future Work

In this study, accident data from the Seattle Department of Transport were used to train and test models for predicting the severity of an accident based on information that might be available to an emergency services call handler when an accident is reported to them. The main purpose of building this model is to allow the emergency services in Seattle to allocate their resources in such a way as to best deal with accidents when they do occur. By knowing where, when and in what conditions an accident happens, and by having a description of the accident (which can be cross-referenced against the Seattle Department of Transport collision code dictionary) this allows the probability of the accident involving serious injury or death to be evaluated. If an accident's severity is predicted to be high then it may be more advisable to dispatch ambulances and multiple police units to confront the situation, whereas if an accident is predicted to be of low severity, then it may be more appropriate to send more mobile units (e.g. police motorcycles and motorcycle ambulances) to investigate the incident initially.

The LR, kNN and SVM models all perform well, with $F1$ scores between 0.74–0.77. The DT model performs relatively poorly, having an $F1$ score of 0.67. Given the simlar performance of the LR, kNN and SVM models, the SVM model is preferred, as it has the key advantage of being able to return a ranked list of the most significant features in terms of their influence on the accident severity code (provided a linear mapping kernel is used). The SVM model highlights that accidents involving pedestrians and multiple vehicles often have severe consequences, as do those in which excess speed is a factor. By identifying and ranking the major causes of accident severity in this manner, it is hoped that town/city planners will be able to design new road infrastructure and target the introduction of traffic calming measures where they are most needed.

Such a model can, of course, be adapted to any road traffic network in any part of the world in which sufficient accident data are recorded. In future, the model could be improved to predict the accident severity on a continuum running from 1–4, rather than simply predicting a binary accident severity of 0 (minor) or 1 (major). The Seattle traffic dataset provides a rich feature set upon which to build the model, however the number of accidents which were missing one or more key variables was relatively high (including, for almost 10% of the accidents recorded, the target variable itself). As a result, the dataset which initially contained 221,006 entries was cut by around 20% before the process of data balancing/resampling began. With improved record keeping in more recent years, the quality of the dataset is improving all the time. Moreover, the models presented in this work treated data from 2004–2020 without prejudice to the year in which the accident occurred: as previously noted, however, the daily number of accidents in Seattle has declined markedly since 2006. It may be, therefore, that road conditions have changed appreciably in Seattle since then, and that as a result, including these older data may be biasing the model. In future, it may be worth revisiting this work and modelling the accident data in five-year chunks, to see if the features which best predict accident severity have changed over time.

References

[1] https://www.pbs.org/newshour/nation/motor-vehicle-crashes-u-s-cost-871-b illion-year-federal-study-finds

[2] http://data-seattlecitygis.opendata.arcgis.com/datasets/5b5c745e0f1f48e7a53acec63a0022ab_0

[3] https://github.com/apt-astro/IBM_Applied_Data_Science_Capstone/blob/master/Accident_Severity_Capstone.ipynb

[4] https://www.seattle.gov/Documents/Departments/SDOT/GIS/Collisions_OD.pdf

[5] https://github.com/seattleio/seattle-boundaries-data

[6] https://towardsdatascience.com/how-to-find-the-optimal-value-of-k-in-knn -35d936e554eb