

IBM Applied Data Science Capstone Project

Car accident severity

Snehal Kolekar

3rd November 2020

Abstract

Globally, road traffic accidents are an important health concerns which needs to be tackled. A multidisciplinary approach is required to understand what causes them and to provide the evidence for policy support. Some of the key factors in determining the likelihood and severity of a road traffic accident can include-weather conditions, road and light conditions, individual negligence (driving under influence of drugs or alcohol, driving at high speed, etc). In this report, the Seattle road accident data is used which is recorded by the Seattle Department of Transport. The machine learning techniques are used here for modeling relative prevalence and influence of these factors in the severity of road traffic accidents in Seattle city. This will help the city planners for future road construction and also for emergency service providers for predicting the severity of accidents based on information provided at accident.

1. Introduction

Road traffic accidents are an important public health concern globally. Approximately 1.35 million people die in road crashes each year, on average 3,700 people lose their lives every day on the roads [1]. An additional 20-50 million suffer non-fatal injuries, often resulting in long-term disabilities. Therefore, it is necessary to study the factors causing them and finding the strategies to tackle the road accidents.

Traffic accidents not only cause millions of disabilities and deaths, and take a toll on the countries' finance, but are also considered a big problem in the way of improving public health. Numerous factors such as environmental, vehicle-related, host (driver, pedestrian), and their type of interaction affect characteristics of these accidents. In order to find strategies for minimizing traffic accidents, we need to understand the combination of numerous factors.

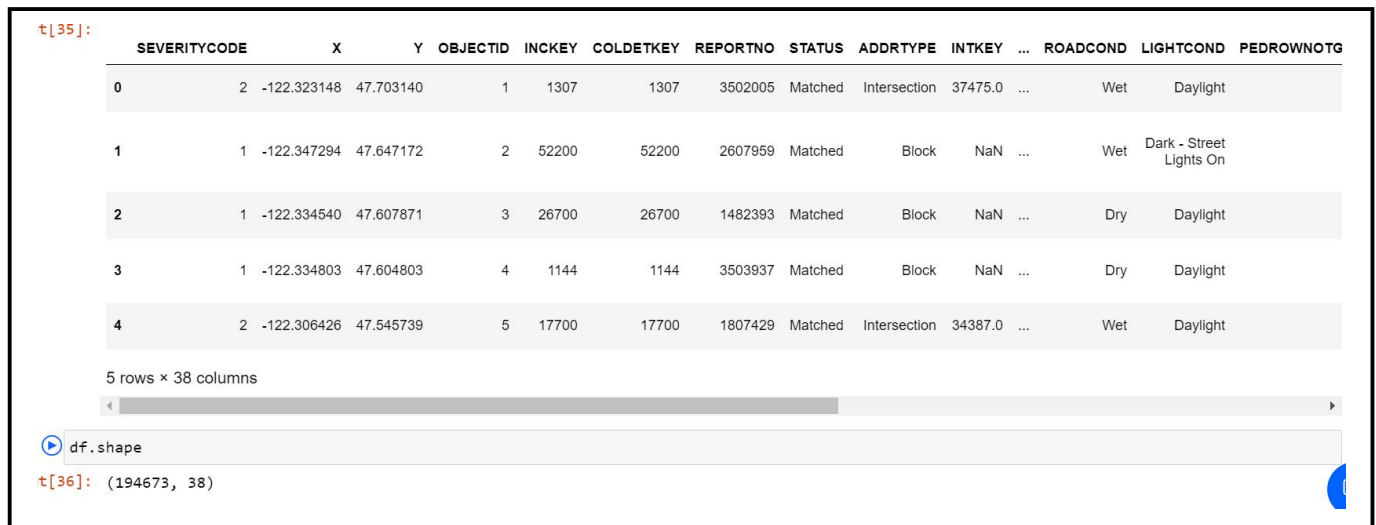
Our main aim is to find what are the main factors causing road accidents and can we predict the severity based on these factors? This study will be helpful for understanding the design of road network, finding the high risk zones, evaluating the severity in particular areas. So, the targeted group will be city planners and engineers

who can design the road network with consideration of preventative mechanisms for Traffic Accidents and Road Safety.

2. Data

2.1 Data Source

The example dataset is used here which is recorded by SDOT (Seattle Department of Transport). Data_Collisions.CSV file is downloaded & read into pandas read_csv function in Jupyter notebook [2].



The screenshot shows a Jupyter notebook cell with the following content:

```
t[35]:
```

	SEVERITYCODE	X	Y	OBJECTID	INCKEY	COLDKEY	REPORTNO	STATUS	ADDRTYPE	INTKEY	...	ROADCOND	LIGHTCOND	PEDROWNOTG
0	2	-122.323148	47.703140	1	1307	1307	3502005	Matched	Intersection	37475.0	...	Wet	Daylight	
1	1	-122.347294	47.647172	2	52200	52200	2607959	Matched	Block	NaN	...	Wet	Dark - Street Lights On	
2	1	-122.334540	47.607871	3	26700	26700	1482393	Matched	Block	NaN	...	Dry	Daylight	
3	1	-122.334803	47.604803	4	1144	1144	3503937	Matched	Block	NaN	...	Dry	Daylight	
4	2	-122.306426	47.545739	5	17700	17700	1807429	Matched	Intersection	34387.0	...	Wet	Daylight	

5 rows x 38 columns

df.shape

```
t[36]: (194673, 38)
```

Figure 1: Screenshot from Jupyter notebook showing first five rows and shape of dataset

Original dataset contains 194673 rows and 38 columns as shown in figure 1. Such a large dataset requires too much time & memory while training and testing models. Therefore, we are randomly selecting sample of original dataset which contains 19467 rows & 38 columns of original dataset as shown in figure 2.



```
7]: #selecting sample of large dataset
df = df.sample(frac=0.10)

df.shape

Out[37]: (19467, 38)
```

Figure 2: Screenshot of Jupyter notebook showing shape of randomly selected sample dataset

Since we would like to identify the factors that cause the accident and the level of severity, we will use SEVERITYCODE as our dependent variable (Y), and try different combinations of independent variables (X) to get the result. Since, the observations are quite large, we may need to filter out the missing value and delete the unrelated columns first. Then, we can select the factor which may have more impact on road accidents, such as address type, weather, road condition, and light condition.

2.2 Data Cleaning

Before building machine learning models, there is need of data wrangling/ data cleaning which is the process of converting data from the initial format to the format that may be better for analysis. There are some reasons why this original dataset is not suitable for further analysis, they are described in following subsections.

2.2.1 Identifying & dealing with missing data

For identifying missing values we are using Python's built-in-function isnull() as shown in Figure 3.

```
#evaluating missing data
missing_data = df.isnull()
missing_data.head(5)
```

	SEVERITYCODE	X	Y	OBJECTID	INCKEY	COLDETKEY	REPORTNO	STATUS	ADDRTYPE	INTKEY	...	ROADCOND	LIGHTCOND	PEL
0	False	False	False	False	False	False	False	False	False	False	...	False	False	True
1	False	False	False	False	False	False	False	False	False	True	...	False	False	True
2	False	False	False	False	False	False	False	False	False	True	...	False	False	True
3	False	False	False	False	False	False	False	False	False	True	...	False	False	True
4	False	False	False	False	False	False	False	False	False	False	...	False	False	True

5 rows × 38 columns

Figure 3: screenshot showing missing data in dataframe df

In above Figure 3, “TRUE” stands for missing value and “FALSE” stands for not missing value. Those missing values may hinder our further analysis as some of key predictive variables have missing entries. Including these missing data entries in the model is likely to bias the model and so we drop the affected rows. To check total number of missing data in each column we use isnull().sum() function as shown in figure 4.

```
#Check where there are NaNs in the dataframe
print(df.isnull().sum(axis=0))
```

SEVERITYCODE	0
X	519
Y	519
OBJECTID	0
INCKEY	0
COLDETKEY	0
REPORTNO	0
STATUS	0
ADDRTYPE	204
INTKEY	12869
LOCATION	266
EXCEPTSNCODE	10973
EXCEPTSNDESC	18889
SEVERITYCODE.1	0
SEVERITYDESC	0
COLLISIONTYPE	503
PERSONCOUNT	0
PEDCOUNT	0
PEDCYLCOUNT	0
VEHCOUNT	0
INCDATE	0
INCDTTM	0

Figure 4: counting missing data in each column

Based on the summary above, each column has 19467 rows of data, 19 columns containing missing data. Target variable "SEVERITYCODE" does not have any null or unknown values. But some of the key predictor variables have missing or unknown data such as WEATHER, ROADCOND. Including these data entries is likely to bias the model. So, we drop the affected rows.

2.2.2. Dealing with unnecessary columns

Several columns in the dataset are unrelated to causes or severity of accidents. Examples of such columns are OBJECTID, COLDETKEY. So, columns that are not included in building or testing models should be dropped.

2.2.3 Dealing with categorical values

Some columns contain categorical data. Such columns are WEATHER, ROADCOND, LIGHTCOND. Machine learning models do not handle categorical variables. So, it is necessary to convert these categorical data to numerical data via one hot encoding using PANDAS function `pd.get_dummies()`. Also, some columns such as SPEEDING, HITPARKEDCAR have the binary data in the (Y/N) format. In order to make these columns useful for machine learning models, it is necessary to convert these columns into numerical data such as 1 for Y and 0 for N.

2.2.4 dealing with unbalanced dataset

The dataset is highly imbalanced. As shown in figure 5., there are 13221 accidents of SEVERITYCODE = 1 & 5705 accidents of SEVERITYCODE = 2. If we train models to predict severity using this dataset where majority of accidents have one particular outcome (severitycode =1) then it is likely to have biased model. To avoid this problem, we will under sample the dataset having same number of values for SEVERITYCODE =1 and SEVERITYCODE = 2. Figure 6 shows us balanced dataset histogram.

```
: df['SEVERITYCODE'].value_counts()
Out[75]: 1    13221
         2     5705
         Name: SEVERITYCODE, dtype: int64
```

Figure 5: number of data values having SEVERITYCODE = 1 and 2

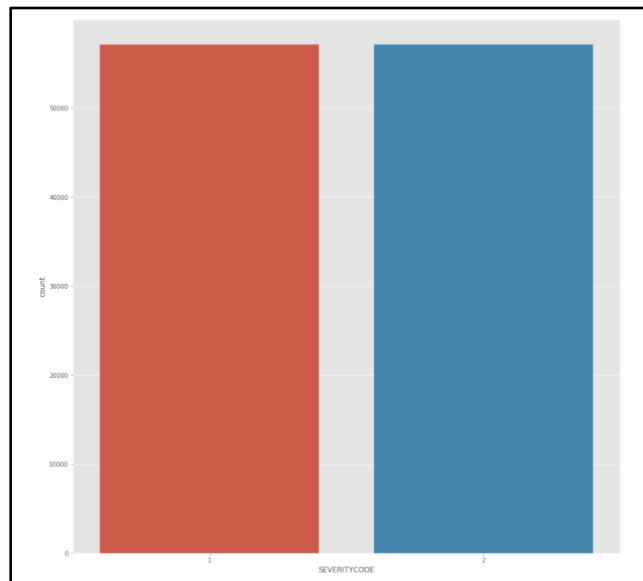


Figure 6: balanced dataset histogram

3. Exploratory data analysis

Before we begin building model, it is useful to plot some key features of the dataset in order to gain an intuitive understanding of Seattle car accident database. To visualize different factors related to car accidents severity, I will use python library-Matplotlib and generate the graphs by using matplotlib.pyplot function.

In figures 7-9, the distribution of light conditions, road conditions and weather conditions are shown both before and after resampling the data as described in subsection 2.2.4. In figure 7, the resampled dataset replicates trends of distribution of lighting conditions same as before sampling. It shows that the relative proportions of the most common lighting conditions “Daylight” and “Dark-Street Lights On” are kept in balance after resampling the dataset. This indicates that we have not significantly biased the dataset towards any particular set of light conditions.

In figure 8, we see that by resampling the data, we lose information corresponding to extremely rare road conditions – 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 distribution of most common road conditions – Dry and Wet are unchanged by resampling.

Finally, in figure 9 we have the distribution of weather conditions before and after resampling. It shows that random sampling has kept balance of most common weather condition distributions such as Clear, Raining and Overcast.

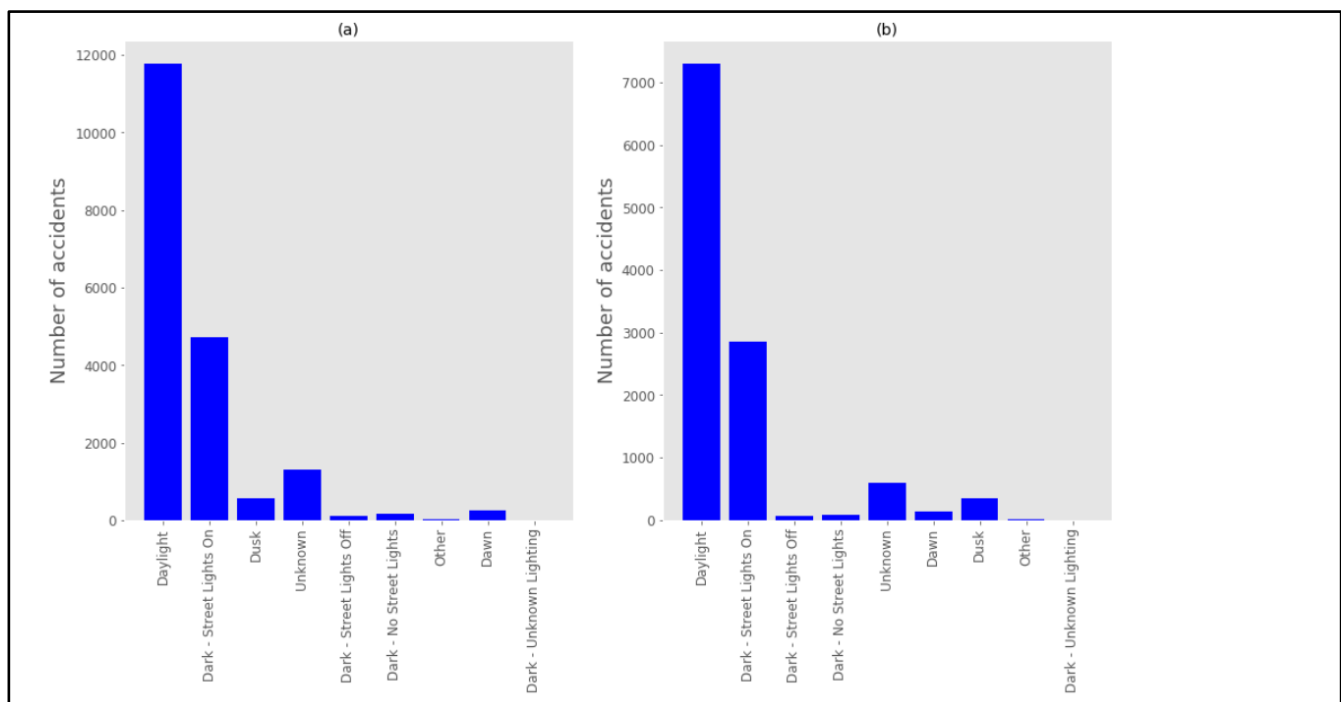


Figure 7: Distribution of light conditions recorded by SDOT before (a) and after (b) re-sampling

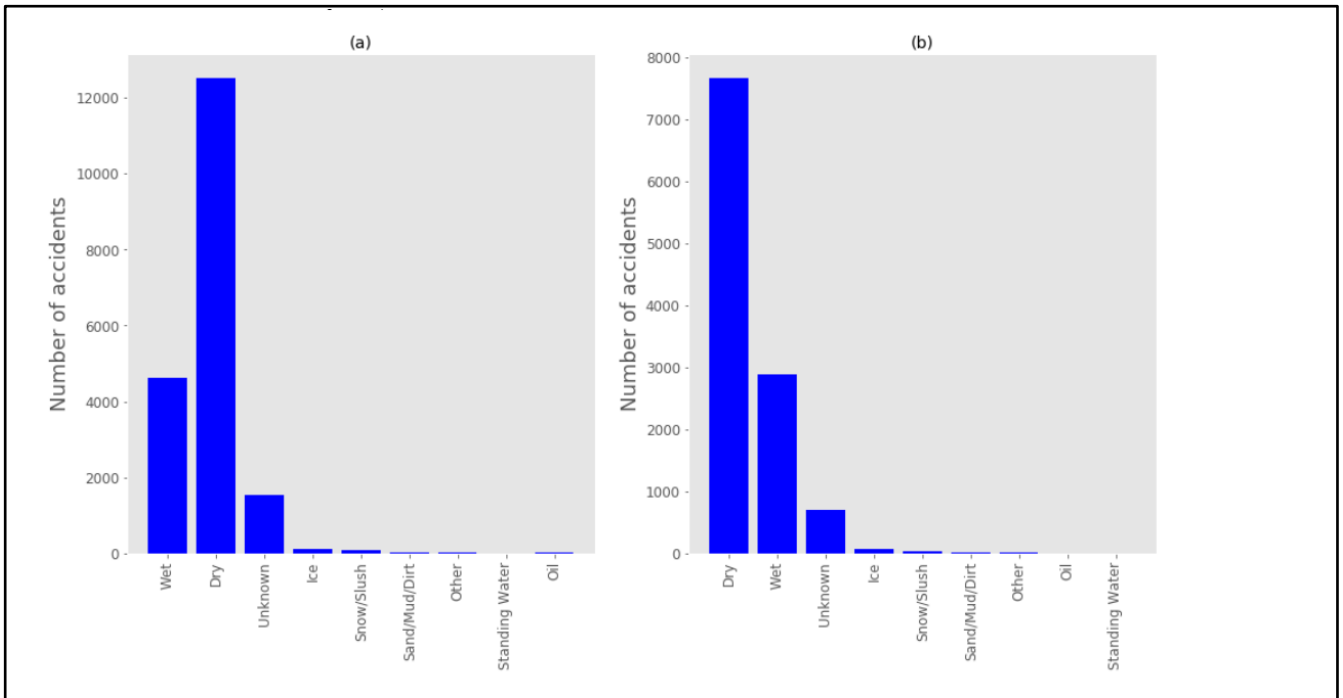


Figure 8: Distribution of road conditions recorded by SDOT before (a) and after (b) re-sampling

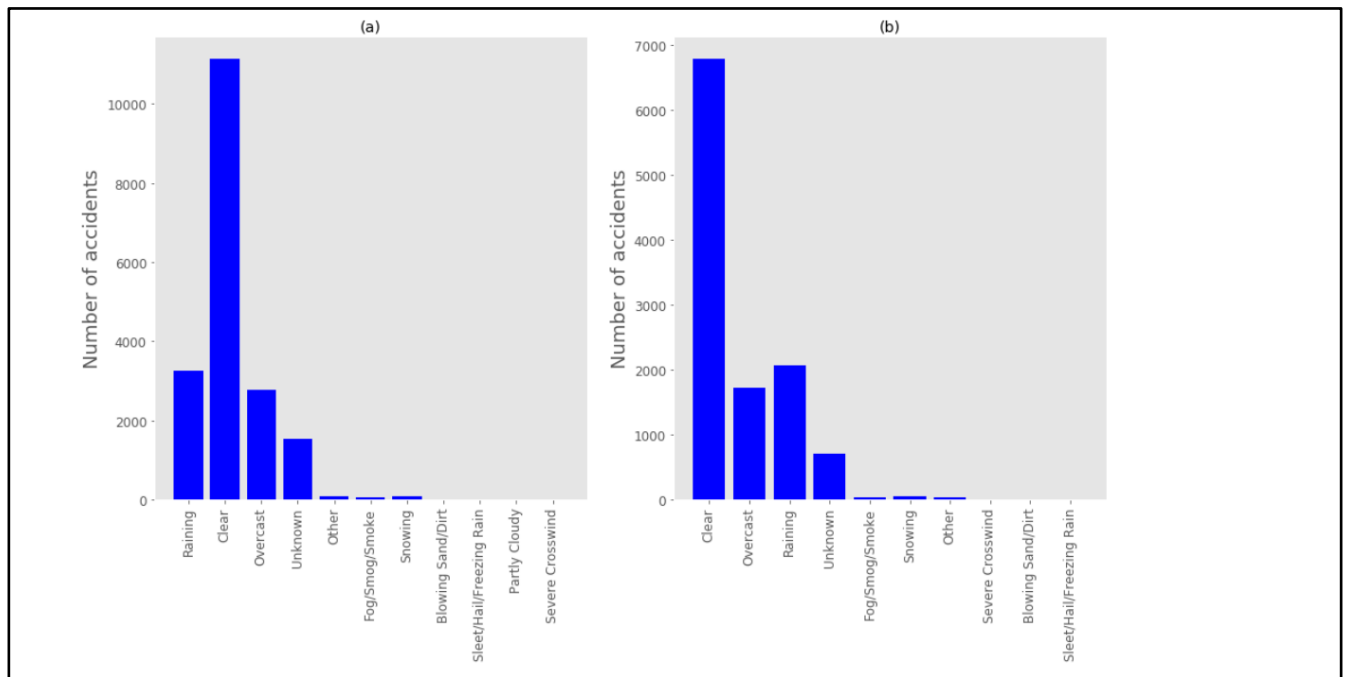


Figure 9: Distribution of weather conditions recorded by SDOT before (a) and after (b) re-sampling

We can illustrate that after cleaning and resampling the dataset, we got significantly balanced dataset which will not bias the model. Now, we are confident to use dataset for modeling.

4. Methodology

The primary object of this analysis is to train and test a model which predicts SEVERITYCODE, an integer 1 or 2 which describes whether an accident involved property damage or injury from information which may be available to emergency service providers at the time when an accident is reported. Having cleaned and balanced dataset, the final steps before modeling can be feature selection, standardizing the feature set and splitting dataset into training and testing datasets.

4.1 Feature Selection

After cleaning and balancing dataset, the final feature set used to predict the severity code includes following columns.

- PEDCOUNT, PEDCYLCOUNT, VEHCOUNT - these columns indicate how many pedestrians, cyclists, vehicles are involved in accidents.
- UNDERINFLU – indicates whether or not a driver involved was under the influence of drugs or alcohol.
- SPEEDING – indicates speeding is factor for collision or not.
- Data from WEATHER which is expanded in 9 columns showing weather conditions using one hot encoding.
- Data from ROADCOND which are expanded to 8 columns using one hot encoding.
- Data from LIGHTCOND which are expanded to 8 columns using one hot encoding
- Data from ADDRTYPE showing collision address type is expanded in 3 columns using one hot encoding

So, the final FEATURE dataset consists of 11410 rows & 33 columns.

4.2 Standardization

Having cleaned and balanced data, the final step before modelling is to standardize 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 important step to perform in order to ensure that the model is not biased towards or away from certain feature types.

4.3 Train-Test Split

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.

4.4 Models

Models that we can use to predict the car accident severity from the available data fall into two categories: regression models and classification models. The Seattle accident data have cleaned & prepossessed in such a way that we can build models belonging to either category to predict `SEVERITYCODE`.

4.4.1 Decision Tree Model

Decision tree models are built by iterating through 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=1` and `SEVERITYCODE=2` within parent sample, and then from each of these two “branches” identify the feature which most cleanly separates the data subsets and so on. The aim is usually to branch the tree until every leaf contains only `SEVERITYCODE = 1` or `SEVERITYCODE = 2` i.e. has no disorder/entropy. However, if there are many features in feature set, 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 Seattle accident dataset using the `DECISIONTREECLASSIFIER` in `SCIKIT LEARN`, using the “entropy” criterion. As the cleaned and balanced dataset is relatively small (11410 rows x 33 columns), no depth limit was imposed, and the decision tree classifier ran until every leaf was pure. At the initial branching level, a cut on `PEDCOUNT` is found to split the sample into subsamples a and b with entropies of 0.998 and 0.224. At the second branching level, branch a is split on `PEDCYLCOUNT` into subbranches with entropies of 0.933 and 0.345. Meanwhile, branch b is split on `VEHCOUNT` with entropies of 0.201 and 0.61.

The first three layers of the decision tree from parent sample are shown in figure 10 and the model’s confusion matrix is shown in figure 12. The decision tree model correctly predicts accidents with `SEVERITYCODE = 1,2` 67% and 63% respectively. The F1 scores for `SEVERITYCODE 1` and `2` are 0.66 and 0.64 respectively.

4.4.2 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.

SVM model was built from car accident dataset using C-Support Vector Classification method (SKLEARN.SVM.SVC), with a linear mapping kernel. The precision of the SVM model for predicting accident SEVERITYCODE=1,2 is 0.64 and 0.65 respectively and F1 scores of the model for both categories are 0.66 and 0.61 respectively. The confusion matrix of SVM model is shown in figure 12.

4.4.3 Logistic Regression Model

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 SCILIT LEARN with a regularization 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 test set. The confusion matrix providing insight into the accuracy of the model is shown in figure 12. We see that 68% of the time the model correctly predicts SEVERITYCODE=1 and 61% of the time the model correctly predicts SEVERITYCODE=2. The F1 score is 0.66 for SEVERITYCODE =1 and 0.62 for SEVERITYCODE =2, in the test subset.

4.4.4 K-Nearest Neighbor Model

The final model which we use in our analysis of the Seattle accident database uses the k-Nearest Neighbor (kNN) algorithm. The kNN is pattern recognition algorithm which maps an input dataset to multidimensional hyperspace and then attempts to classify a data point of unknown classification based on the classifications of k nearest neighbors. 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. Training the model with using too few neighbors (low k) increases the likelihood of chance matches, creating unstable decision boundaries, whereas training the model with too many neighbors (high k) can rob the model of discriminatory power. While there is no priority method of choosing the best k, it is often the case that model's predictive power is optimized for $k \sim \sqrt{N}$, where N is the number of samples in training dataset.

KNN model is built for $k = 1-60$ using the KNEIGHBORCLASSIFIER in SCIKITLEARN. The resulting model accuracy as a function of k is shown in figure 11. We see that best accuracy was with 0.65 with $k = 16$. The confusion matrix for KNN model (Figure 12) highlights that the KNN approach correctly categorizes

SEVERITYCODE = 1 69 % of the time and correctly categorizes SEVERITYCODE = 2, 62 % of the time. The F1 scores of the two categories are 0.67 and 0.64.

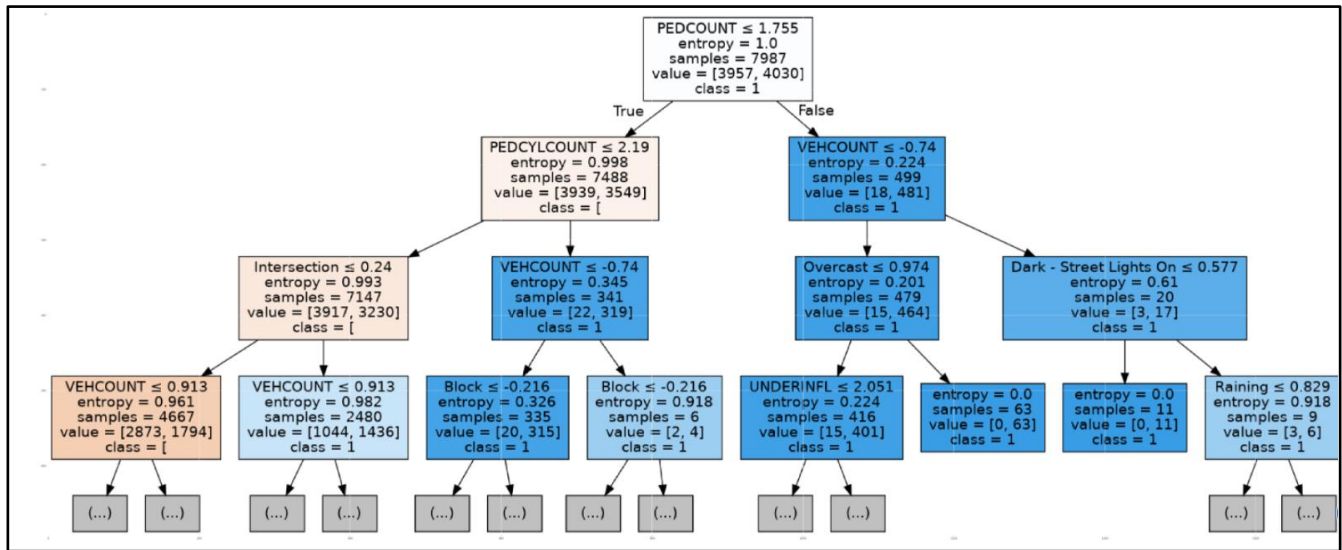


Figure 10: First three layers of Decision Tree Classification Model

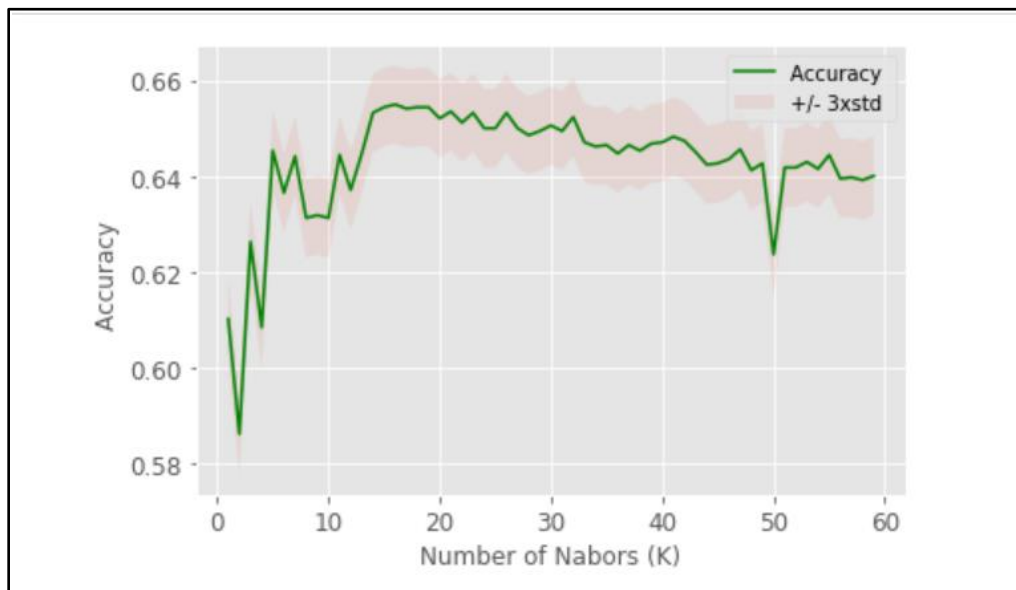


Figure 11: Comparison of the kNN model accuracy for k ranging from 1-60. We see that the model accuracy is optimized for k = 16, hence fit the optimal kNN model using the 16 nearest neighbors for each datum.

5. Result and Discussion

Now we can summarize the comparative performances of each of the four machine learning models in terms of correctly categorizing SEVERITYCODE based on

the feature set available to describe each accident. The confusion matrices for all four models are shown in figure 12. We see that both the KNN and SVM models correctly identify accidents with SEVERITYCODE = 1 69% of the time, while the Decision Tree model correctly identify accidents with SEVERITYCODE = 2 with 63 %. The SVM model has lower predictions of SEVERITYCODE = 2 accidents with 59% and the Decision Tree model has performed low for predicting SEVERITYCODE = 1 accidents with 67%.

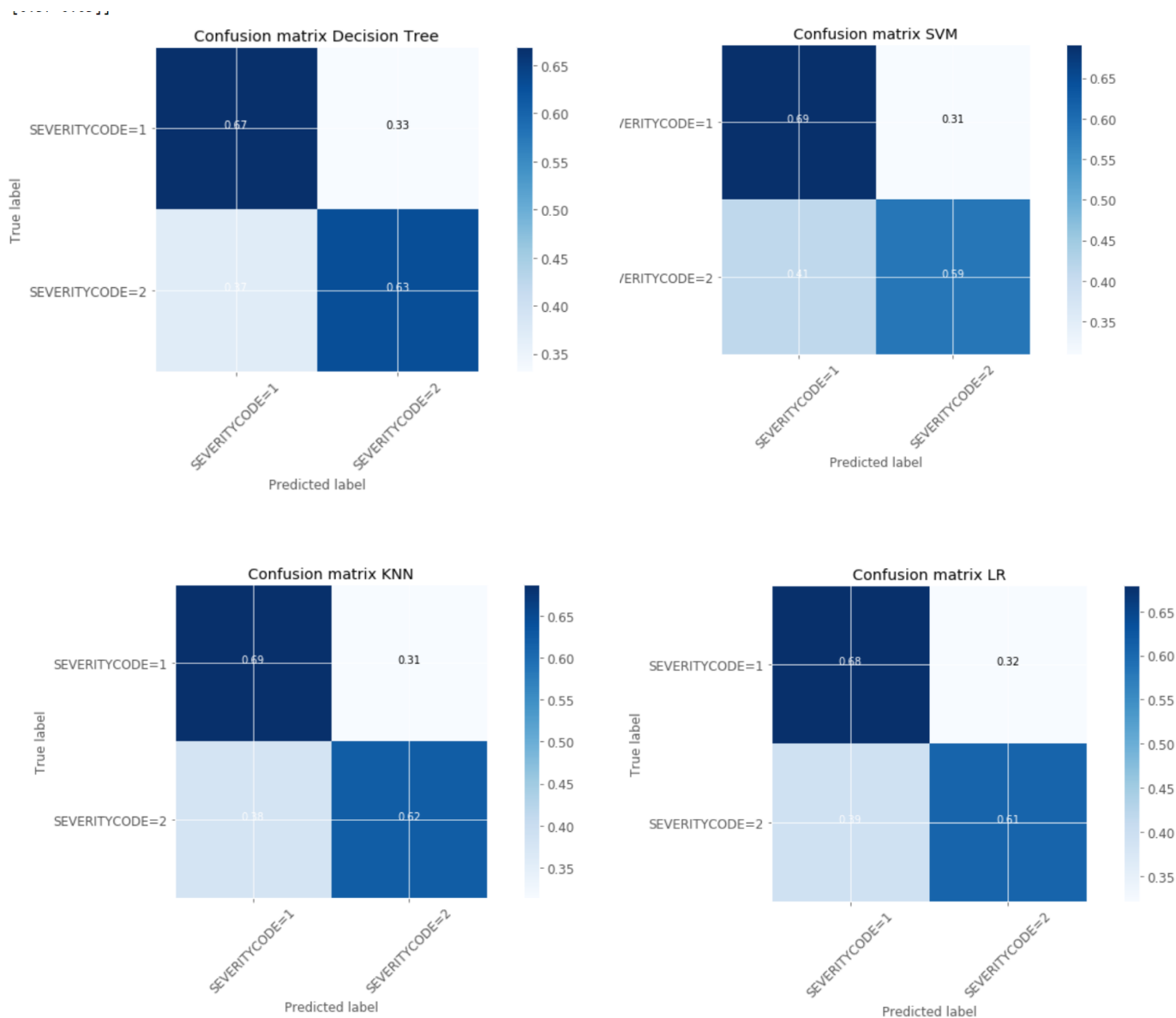


Figure 12: Confusion matrices for the Decision Tree, Support Vector Machine, Logistic Regression and K-nearest Neighbor models. Both SVM and KNN models have highest accuracy for predicting SEVERITYCODE = 1 accidents (69%), while for predicting SEVERITYCODE = 2 accidents Decision Tree has highest accuracy (63%).

In the figure 13, F1 score and Jaccard Similarity Index for each model is shown. The kNN model has highest F1 score with 0.65. The Decision Tree has second largest F1 score as 0.649. Logistic regression and SVM models have relatively lower F1 Score as 0.64 and 0.63. Jaccard Similarity Scores for the four models are 0.65,0.649,0.64and 0.643 respectively.

Algorithm	Jaccard	F1-score	Log Loss
KNN	0.654981	0.654585	NA
Decision Tree	0.649722	0.649583	NA
SVM	0.640082	0.639027	NA
Logistic Regression	0.643587	0.643076	0.604675

Figure 13: Report showing performance metrics of four machine learning models

6. Conclusions and Future Work

In this study, accident data from the Seattle Department of Transport were used to train and test the models for predicting the severity of an accidents based on information that might be available to emergency service providers 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 having accident information, we can predict the probability of the accident involving injury or property damage.

The kNN, Decision Tree and LR models performed well with F1 scores between 0.64-0.65. The SVM model has relatively lower performance with F1 score as 0.63. Given the nearly similar performances of the KNN and Decision Tree models, Decision Tree models is preferred as it has ability to return the rankings of most significant features via decision tree leaf levels.

Such a model can of course, be adopted 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 1 or 2. Moreover here we are used 10% of original data ,for future work we can use more dataset to predict severity of accidents. With improved record keeping in more recent years, with high quality of data it may be worth revisiting this work in future and modeling the accident data to see if the features which best predict accident severity have changed over time.

7. References

1. <https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries>
2. <https://www.coursera.org/learn/applied-data-science-capstone/supplement/Nh5uS/downloading-example-dataset>
3. https://github.com/snehalkolekar/Coursera_Capstone/blob/master/week%202%20assignment.ipynb
4. <https://towardsdatascience.com/methods-for-dealing-with-imbalanced-data-5b761be45a18>