Predicting Car Accident Severity using Seattle City Data: A Machine Learning-Based Approach

Sumanta Sannigrahi

Sep 30, 2020

1. Introduction

To reduce the frequency of car collisions in a community, an algorithm must be developed to predict the severity of an accident given the current weather, road and visibility conditions. When conditions are bad, this model will alert drivers to remind them to be more careful.

1.1 Background

With the increasing use of motorized vehicles, came a vast pressure on the network of highways. In large cities such as Seattle, this often leads to traffic congestion and as a result, accidents. Those incidents may leave other road users in peril, requiring action from emergency services to restore safety. Each accident is different in terms of severity, it is therefore often unknown how many emergency vehicles should be sent out. In high-severity cases, emergency aircrafts and fire brigade involvement may be required too. It is advantageous for those institutions to accurately predict how many vehicles and manpower are required to mitigate the impact of car accidents. A pre-emptive assessment of car accident severity leads to more complete information on the means required to escort each casualty to safety and restore the infrastructure**.**

1.2 Problem

This study aims to predict the severity of a car accident according to historical data. Emergency services often receive incomplete information when informed about collisions. Therefore, they struggle to assess how many resources they should allocate to mitigate the accident. Based on historical data, machine learning tools may aid them in resource allocation by predicting collision severity according to data provided by civilians. This study aims to bridge the gap between past accidents and future emergency assistance.

1.3 Interest

Obviously, Emergency services would reap large benefits from early information on the severity of incoming accidents. These benefits mainly apply to more efficient resource planning, optimization of personnel use and more effective mitigation. Furthermore, road users are stakeholders too, since they benefit from faster resolution and experience enhanced safety when involved in accidents themselves. This will help the road safety department to predict the severity based on road condition, light condition and many other data points discussed in the section below. This will not only help the city of Seattle but all other cities which can benefit from this project.

1.4 Data acquisition and cleaning

1.5 Data sources

Data on collisions within the city of Seattle is available through an Open Data initiative hosted by Seattle GeoData (Collisions, n.d.). It is noted this dataset is the example dataset provided for this capstone project. Since it provides an extensive set of attributes, it was

decided to use it as main data source for this project.

1.6 Data acquisition and cleaning

The dataset is available as comma-separated values (CSV) files, KML files, and ESRI shapefiles that can be downloaded from the Seattle Open GeoData Portal. The data is also available from RESTful API services in formats such as GeoJSON.
Data was download from the dataset to our project directory and I looked at the data types and the dimensionality of the data. We can see that the dataset contains 221,737 records and 40 variables.

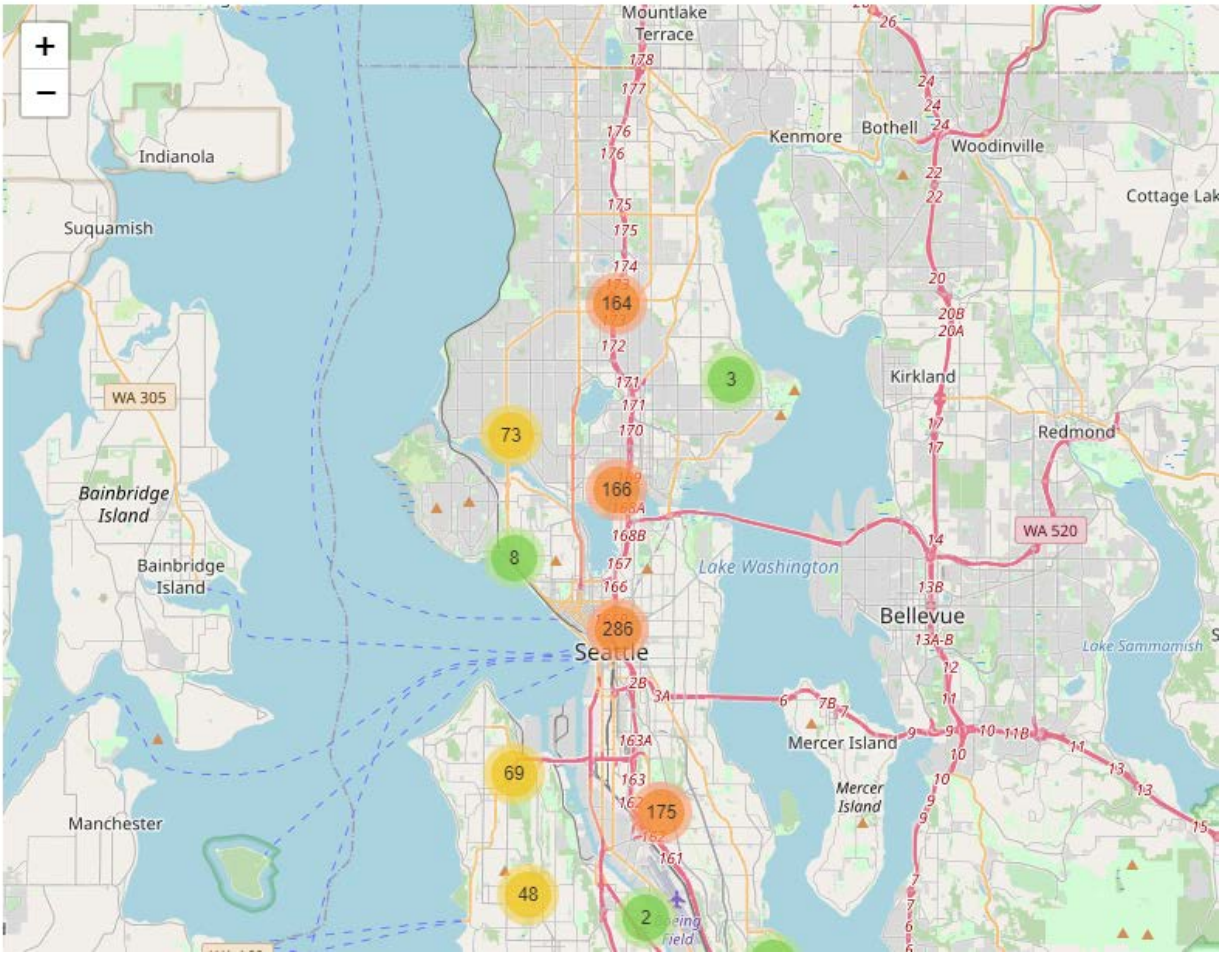
The metadata of the dataset can be found from the website of the Seattle Department of Transportation. On reading the dataset summary, we can determine the description of each of the fields and their possible values.

The data contains several categorical fields and corresponding descriptions which could help us in further analysis. We make an attempt at understanding the data in terms of the fields that we shall take into account for later stages of model building.

There are 2,21,737 observations and 40 variables in this data set. 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 the accidents, such as address type, weather, road condition, and light condition.

Target data - SEVERITYCODE
Other important variables include:
• ADDRTYPE: Collision address type: Alley, Block, Intersection
• LOCATION: Description of the general location of the collision
• PERSONCOUNT: The total number of people involved in the collision helps identify severity involved
• PEDCOUNT: The number of pedestrians involved in the collision helps identify severity involved
• PEDCYLCOUNT: The number of bicycles involved in the collision helps identify severity involved
• VEHCOUNT: The number of vehicles involved in the collision identify severity involved
• JUNCTIONTYPE: Category of junction at which collision took place helps identify where most collisions occur
• WEATHER: A description of the weather conditions during the time of the collision
• ROADCOND: The condition of the road during the collision
• LIGHTCOND: The light conditions during the collision
• SPEEDING: Whether or not speeding was a factor in the collision (Y/N)
• SEGLANEKEY: A key for the lane segment in which the collision occurred
• CROSSWALKKEY: A key for the crosswalk at which the collision occurred
• HITPARKEDCAR: Whether or not the collision involved hitting a parked car
This section describes the data requirements for this study. The data set of choice is introduced, as well as the steps for data cleaning and feature selection. The latter consists of a list of features that serve as input to predictive modelling, to be used by machine learning algorithms later

The X and Y fields denote the longitude and latitude of the collisions. We can visualize the first few non-null collisions on a map.



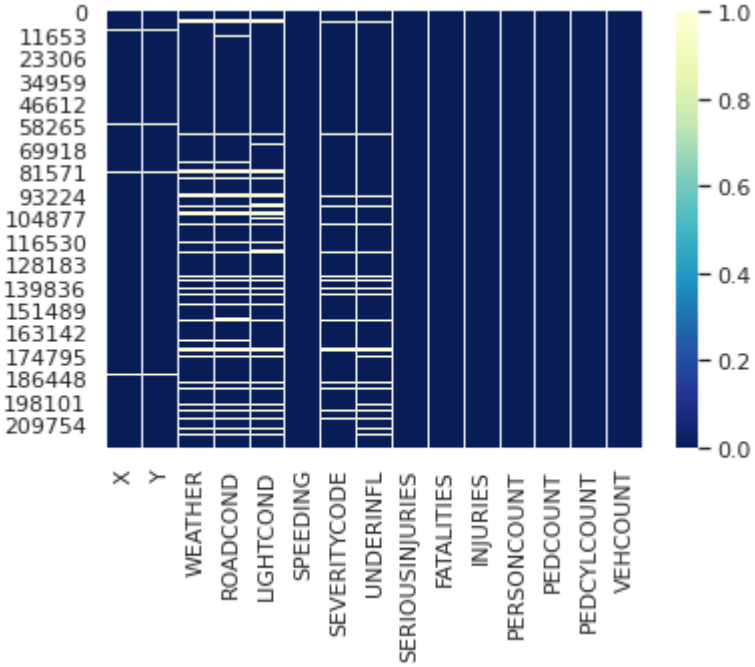
The **WEATHER** field contains a description of the weather conditions during the time of the collision. The ROADCOND field describes the condition of the road during the collision. The LIGHTCOND field describes the light conditions during the collision. The SPEEDING field classifies collisions based on whether speeding was a factor in the collision. Blanks indicate cases where the vehicle was not speeding.

The **SEVERITYCODE** field contains a code that corresponds to the severity of the collision. and SEVERITYDESC contains a detailed description of the severity of the collision. From the data it can be conclude that there were 349 collisions that resulted in at least one fatality, and 3,102 collisions that resulted in serious injuries. The following table lists the meaning of each of the codes used in the SEVERITYCODE field:

SEVERITYCODE Value	Description
1	Accidents resulting in property damage
2	Accidents resulting in injuries
2b	Accidents resulting in serious injuries
3	Accidents resulting in fatalities
0	Data Unavailable i.e. Blanks

The **UNDERINFL** field describes whether a driver involved was under the influence of drugs or alcohol. The values "0" and "N" denote that the driver was not under any influence while 1 and Y that they were. The **PERSONCOUNT** and **VEHCOUNT** indicate how many people and vehicles were involved in a collision respectively

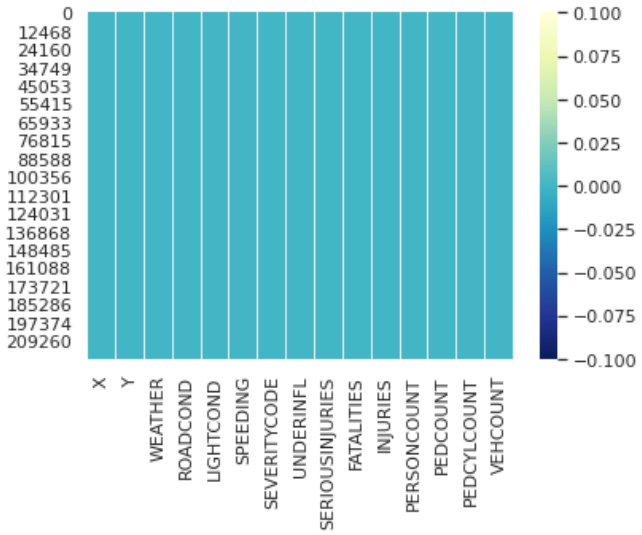
As the dataset is sourced from a database table, several unique identifiers and spatial features are present in the database which may be irrelevant in further statistical analysis. These fields are OBJECTID, INCKEY, COLDETKEY, INTKEY, SEGLANEKEY, CROSSWALKKEY, and REPORTNO. Other fields such as EXCEPTRSNCODE, SDOT_COLCODE, SDOTCOLNUM and LOCATION and their corresponding descriptions (if any) are categorical but have many distinct values that shall not be that much useful for analysis. The INCDATE and INCDTTM denote the date and the time of the incident but may not be of use in further analyses. The data needs to be pre-processed or cleaned.



Data before cleaning.

After dropping irrelevant columns and null values and performing data cleaning, we got dataset with 171,380 rows and 15 columns.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 221738 entries, 0 to 221737
Data columns (total 15 columns):
X                 214260 non-null float64
Y                 214260 non-null float64
WEATHER           195097 non-null object
ROADCOND          195178 non-null object
LIGHTCOND         195008 non-null object
SPEEDING          9936 non-null object
SEVERITYCODE      221737 non-null object
UNDERINFL         195307 non-null object
SERIOUSINJURIES   221738 non-null int64
FATALITIES        221738 non-null int64
INJURIES          221738 non-null int64
PERSONCOUNT       221738 non-null int64
PEDCOUNT          221738 non-null int64
PEDCYLCOUNT       221738 non-null int64
VEHCOUNT          221738 non-null int64
dtypes: float64(2), int64(7), object(6)
memory usage: 25.4+ MB
```



Data after cleaning

1.7 Feature selection

Data cleaning resulted in a data set with 171,504 rows and 15 attributes/Column. Only a smaller subset of those features are viable for machine learning algorithms. First, the data set includes metadata for each recorded collision such as an incident key and a unique identifier. Those were all dropped.

Features regarding date and time were dropped, as they incorporated several problems. Date and time were often not fully filled out, causing time to often be incomplete. If one were to be interested in investigating the relationship between time of day and the likelihood of collisions, this would not be possible with this data.

Since emergency services are the problem owner for this study, we assume the machine learning model will have to forecast collision severity based on incoming civilian reports. Therefore, attributes related to codes and descriptions provided by officials are not viable. Furthermore, civilians are unable to distinguish between injuries, serious injuries and fatalities. Therefore, they are not included within the selected features.

The data set yields a multitude of features with binary data. This includes whether the collision occurred due to speeding or inattention, whether the person was under the influence of drugs or alcohol or whether a parked car was hit. It was chosen to include speeding and intoxication/drug use as features, since data exploration unveiled those cases are prone to higher accident severity. Severe cases require more extensive effort and resources by emergency services. It is therefore paramount our model encapsulates such information. Intoxication or drug use may not always readily be assessed by civilians, yet it is assumed high levels of substance abuse reflect in driver behavior.

The final set of features includes 13 attributes, which yield the following characteristics: 1) no expert view is required to assess their value and 2) they are not highly correlated to other features within the selection. Table 1 provides an overview of the feature selection process.

Kept features	Dropped features	Reason for dropping features
PERSONCOUNT, VEHCOUNT PEDCOUNT, PEDCYLCOUNT, INJURIES, SERIOUSINJURIES, FATALITIES	CROSSWALKKEY JUNCTIONTYPE COLDTKEY REPORTNO	Dropped features that cannot be assessed by civilians.
WEATHER, ROADCOND, LIGHTCOND	-	Included to assess environmental conditions.
SEVERITYCODE-	INCDATE, INCDTTM	Incomplete, too expensive to transform.
SPEEDING, UNDERINFL	INATTENTIONIND , PEDROWNOTGRNT, HITPARKEDCAR	Opted for two binary variables that imply higher severity.
-	COLLISIONTYPE, SDOT_COLCODE, SDOT_COLDESC, SDOTCOLNUM, ST_COLCODE, ST_COLDESC, SEGLANEKEY,	(Meta)data provided by Seattle's emergency services, excluded as civilians cannot assess them.

2. Exploratory Data Analysis

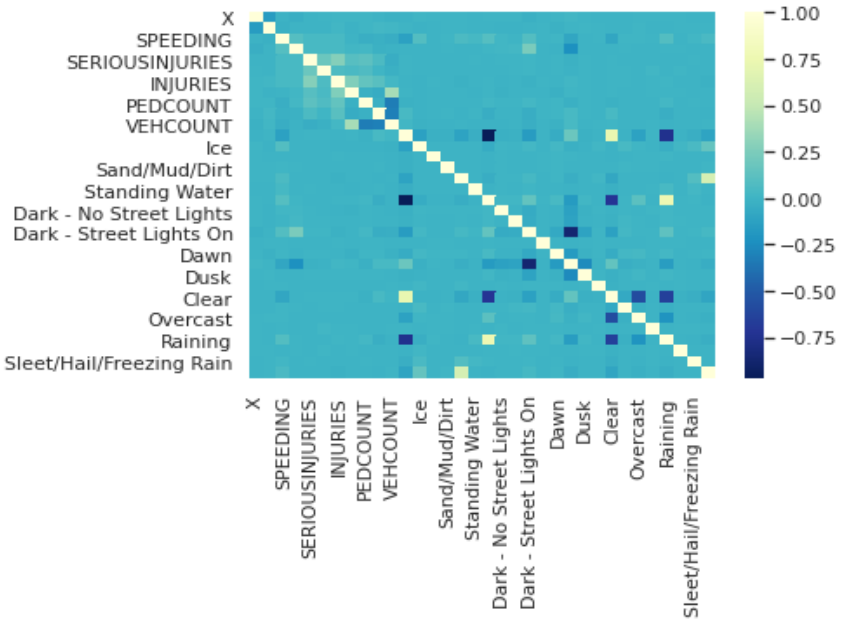
I have used Jupyter Notebook to do the data analysis. To generate the table and graph for the dataset, we imported Python libraries (Pandas, Numpy, Matplotlib, and Seaborn) and Tensor flow

First we imported the data through pd.read_csv. We noticed that it had 221737 rows and 40 columns. Therefore, we narrowed it down to 8 columns ('Severity', 'X', 'Y', 'Location', 'Vehcount', 'Weather', 'Roadcond', 'Lightdcond') and delete the missing values, which made the final dataset with 171540 observations and 13 variables

After fixing other data inconsistencies, now a one-hot encoding of the WEATHER, ROADCOND, and LIGHTCOND fields was done as they are categorical. Shuffling of the dataset is also necessary as it is an unbalanced dataset. Sample data after hot coding is shown below:

	0	1	2	3	4
X	-122.326	-122.361	-122.33	-122.279	-122.313
Y	47.6104	47.6561	47.6042	47.5152	47.6591
SPEEDING	0	0	0	0	0
SEVERITYCODE	1	1	1	1	1
UNDERINFL	0	0	0	0	0
SERIOUSINJURIES	0	0	0	0	0
FATALITIES	0	0	0	0	0
INJURIES	0	0	0	0	0
PERSONCOUNT	2	2	0	2	2
PEDCOUNT	0	0	0	0	0
PEDCYLCOUNT	0	0	0	0	0
VEHCOUNT	2	2	2	2	2
Dry	0	1	1	1	1

Finding the correlation among the features of the dataset helps understand the data better. For example, in the heatmap shown below, it can be observed that some features have a strong positive / negative correlation while most of them have weak / no correlation.



Correlation Matrix

The datasets x and y are constructed. The set x contains all the training examples and y contains all the labels. Feature scaling of data is done to normalize the data in a dataset to a specific range.

After normalization, they are split into x_train, y_train, x_test, and y_test. The first two sets shall be used for training and the last two shall be used for testing. Upon choosing a suitable split ratio, 80% of data is used for training and 20% of is used for testing. Sample data after splitting into training and testing data

```
array([[-0.87565152, -1.78787493, -0.23628704, -0.23847975, -0.1064020
        -0.04123484, -0.57981629, -0.35955477, -0.21020289, -0.1856599
         0.05721618,  0.62623344, -0.07949479, -0.0169053 , -0.0183929
        -0.06973573, -0.02391113, -0.60720945, -0.08993346, -0.0807113
        -0.61619157, -0.01024523, -0.12010997,  0.71342445, -0.1855306
        -0.01565095,  0.74472739, -0.05646148, -0.43411752, -0.0076361
        -0.48292009, -0.01207437, -0.02521821, -0.06952435],
       [-1.03965304,  1.46479561, -0.23628704, -0.23847975, -0.1064020
        -0.04123484, -0.57981629, -0.35955477, -0.21020289, -0.1856599
         0.05721618, -1.59684861, -0.07949479, -0.0169053 , -0.0183929
        -0.06973573, -0.02391113,  1.6468782 , -0.08993346, -0.0807113
        -0.61619157, -0.01024523, -0.12010997, -1.40169011,  5.3899458
        -0.01565095, -1.34277324, -0.05646148, -0.43411752, -0.0076361
         2.07073597, -0.01207437, -0.02521821, -0.06952435],
```

3.Modelling and Evaluation

3.1 Decision Tree Classifier

Decision Tree makes decision with tree-like model. It splits the sample into two or more homogenous sets (leaves) based on the most significant differentiators in the input variables. To choose a differentiator (predictor), the algorithm considers all features and

does a binary split on them (for categorical data, split by category; for continuous, pick a cut-off threshold). It will then choose the one with the least cost (i.e. highest accuracy), and repeats recursively, until it successfully splits the data in all leaves (or reaches the maximum depth).

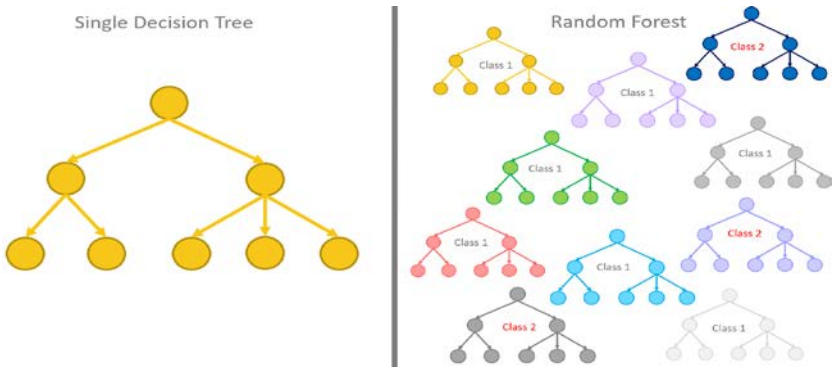
Information gain for a decision tree classifier can be calculated either using the Gini Index measure or the Entropy measure, whichever gives a greater gain. A hyper parameter Decision Tree Classifier was used to decide which tree to use, DTC using entropy had greater information gain; hence it was used for this classification problem.

	precision	recall	f1-score	support
1	1.00	1.00	1.00	22504
2	1.00	1.00	1.00	11068
2b	1.00	1.00	1.00	633
3	1.00	1.00	1.00	71
accuracy			1.00	34276
macro avg	1.00	1.00	1.00	34276
weighted avg	1.00	1.00	1.00	34276

Classification Report for DTC

3.2 Random Forest Classifier

Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction (see figure below).



Random Forest Classifier is an ensemble (algorithms which combines more than one algorithm of same or different kind for classifying objects) tree-based learning algorithm. RFC is a set of decision trees from randomly selected subset of training set. It aggregates the votes from different decision trees to decide the final class of the test object. Used for both classification and regression.

Like DTC, RFC requires an input that specifies a measure that is to be used for classification, along with that a value for the number of estimators (number of decision trees) is required. A hyperparameter was used to determine the best choices for the above-mentioned parameters. RFC using entropy as the measure gave the best accuracy when

trained and tested on pre-processed accident severity dataset.

	precision	recall	f1-score	support
1	1.00	1.00	1.00	22504
2	1.00	1.00	1.00	11068
2b	1.00	1.00	1.00	633
3	1.00	0.99	0.99	71
accuracy			1.00	34276
macro avg	1.00	1.00	1.00	34276
weighted avg	1.00	1.00	1.00	34276

Classification Report for RFC

3.3 Logistic Regression Classifier

In Logistic Regression, we wish to model a dependent variable(Y) in terms of one or more independent variables(X). It is a method for classification. This algorithm is used for the dependent variable that is Categorical. Y is modeled using a function that gives output between 0 and 1 for all values of X. In Logistic Regression, the Sigmoid (aka Logistic) Function is used.

Logistic Regression is a classifier that estimates discrete values (binary values like 0/1, yes/no, true/false) based on a given set of an independent variables. It basically predicts the probability of occurrence of an event by fitting data to a logistic function. Hence it is also known as logistic regression. The values obtained would always lie within 0 and 1 since it predicts the probability. The chosen dataset has more than two target categories in terms of the accident severity code assigned, one-vs-one (OvO) strategy is employed.

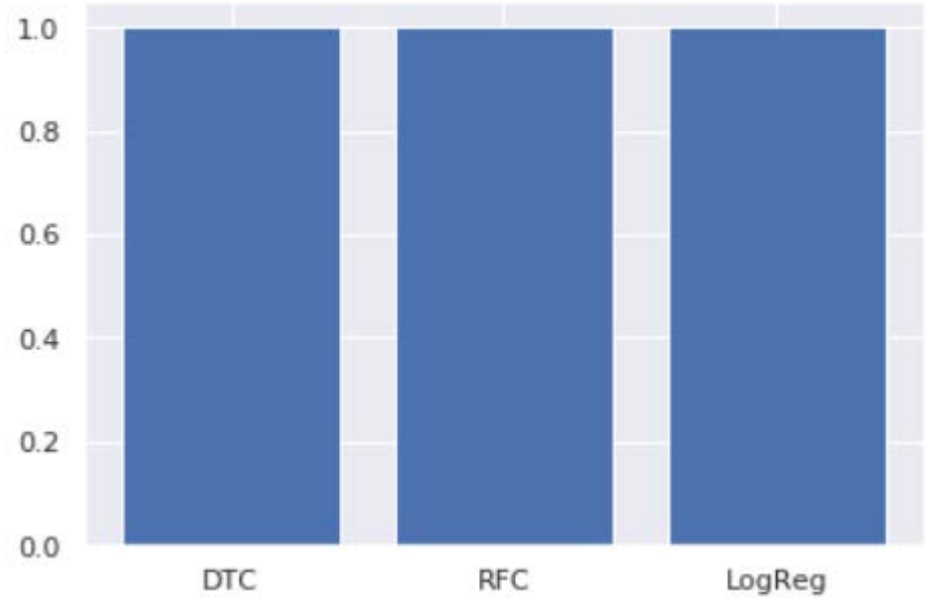
	precision	recall	f1-score	support
1	1.00	1.00	1.00	22504
2	1.00	1.00	1.00	11068
2b	1.00	0.99	1.00	633
3	1.00	0.99	0.99	71
accuracy			1.00	34276
macro avg	1.00	0.99	1.00	34276
weighted avg	1.00	1.00	1.00	34276

Classification Report for LogReg Classifier

4. Results

To check the prediction accuracy, we have used the Decision Tree, Random Forest, Logarithmic regression Models. The main reason for using these models is, since for the given data set some features have a strong positive / negative correlation while most of them have weak / no correlation. To predict properly we needed to find the how closely these factors were correlated and how they contributed to the severity prediction.

The accuracies of all models applied was 100% which means we can accurately predict the severity of an accident. A bar plot is plotted below with the bars representing the accuracy of each model.



5. Discussion

In the beginning of this notebook, we had categorical data that was of type 'object'. This is not a data type that we could have fed through an algorithm, so label encoding was used to created new classes that were of type int8. a numerical data type.
At the start of our analysis, I was trying to figure out the severity and frequency of road accidents based on weather conditions, road conditions, and other factors. Even though our data was a good size, there were number of missing elements and we needed to clean the data in order to get a good result. We had to drop many variables because there were too many missing elements, but I think few variables were important and are contributory factor that should be considered. For the sake of the project we have only used data that are contributing to the accident such as weather, light condition, road condition etc. however the data also captures the areas where the accidents have taken place , which can be of great help to the police department if analyzed properly.
Choosing different max depth and hyper parameter C values helped to improve our accuracy to be the best possible.

Conclusion

Initially, the classifiers had a prediction accuracy of 66%-71%, however, upon going back to the data preparation phase, minor tweaking and taking additional fields in the dataset improved the overall accuracy of all models.

The accuracy of the classifiers is excellent, i.e. 100% for the first three models, From analysis of the model we can say that the model has trained well and fits the training data and performs well on the testing set as well as the training set. We can conclude that this

model can accurately predict the severity of car accidents in Seattle.

References:

— Collisions. (n.d.). Retrieved from https://data seattlecitygis.opendata.arcgis.com/datasets/collisions
— City of Seattle open data portal- https://data.seattle.gov/ https://data.seattle.gov/browse?category=Transportation&provenance=official
— Kaggle.com
— Github.com
— Medium.com
— IBM