**PREDICTING POTENTIAL ACCIDENTS AND SEVERITY BASED ON DIFFERENT DRIVING FACTORS**

**QUANG LA**

**October 4, 2020**

1. **Introduction**
   1. **Background**

According to a report by the National Highway Traffic Safety Administration (NHTSA), there were 36,096 fatalities in 2019, which averages to approximately 100 people died from motor vehicle traffic accidents. While this number is a steep decline from the record year of 2016 (more than 37,000 fatalities), the number of death still presents tremendous emotional suffering and potential economic hardship for the victims’ family. While still years away from safe and 100% autonomous vehicles, Machine Learning has been employed to aid and adjust car settings to avoid accidents in different driving settings.

* 1. **Problem**

This project aims to combine various driving factors and predict how likely and severe a car accident will happen. These data contains relevant information such as location, weather, road condition, type of streets, etc.

* 1. **Interest**

This model will benefit many car manufacturers, especially ones who plans to develop a fully autonomous vehicles like Tesla. The system can predict probability of an accident from current driving settings, and convey their prediction to human/AI driver to make appropriate changes to the vehicle settings.

1. **Data Acquisition and Cleaning**
   1. **Data Sources**

Data for this project is provided from Seattle Department of Transportation. The set includes all traffic accidents in the area from 2004, with relevant information such as location, weather, road, number of vehicles, etc.

* 1. **Data Cleaning**

The dataset was downloaded in .CSV format and contains 38 columns with 194,673 rows. Fortunately, our target label column SEVERITYCODE did not have any missing values. There are three columns with irrelevant information and more than 50% missing values, and were removed from the dataset. For columns with less than 3% missing values, we will remove the null rows from the dataset.

For features, depending on the column, we will either remove the null values, or replace it with a meaningful value. For example, we will remove rows where there're missing values in X and Y. On the other hand, the missing value in Speeding will be replaced with "N" to show the accident did not involve speeding, in contrast to the current available values of "Y". The Y/N values will then be changed into 1/0 for the predictive models. One special case is the UNDERINFL which has all 4 values 0, 1, Y, N, and Blank. For this, we will convert all Y value to 1, N to 0, and all blanks assumed to be 0.

After cleaning, our dataset now has 184,167 rows, which is 94% of the original dataset. This is an acceptable figures, and the remaining rows should be sufficient to train our predictive models.

* 1. **Feature Selection**

Of the remaining 27 columns, some contains elaborated description of another and would not offer any gained information for our Machine Learning Model. Example of these are SEVERITYDESC (description of SEVERITYCODE), JUNCTIONTYPE (for ADDRTYPE), and LOCATION (address from X and Y geospatial coordinates). There were also columns contain number/ID that was assigned by SDOT and did not contribute to our project. Examples of these columns are REPORTNO, SDOT\_COLCODE, ST\_COLCODE, etc. All the columns above were removed from the dataset. After removal, we have 15 columns remaining.

It should be noted that PERSONCOUNT, PEDCOUNT, PEDCYLCOUNT, and VEHCOUNT are quantity columns and can be tempting to add as features. However, the information from columns are after the fact (only accounted for after the accident already happened), and would not benefit our model to predict the accidents before they happen.

* 1. **Feature Engineering**

Our dataset has X and Y columns to represent the geographic coordinate of the accident sites. While it’d be nice to convert these coordinates into zip codes, the data size makes it a very slow and time-consuming task. Current available package in python depends on various map service to reverse geocoding these coordinates, and there’s a limit how many queries can be performed daily. Alternatively, we will use K-mean package to cluster the coordinates into 7 zones to represent the 7 council districts in Seattle.

For date and time columns, we will break them down into category. For example, INCDATE involves date of accident, and we can break them down into 4 quarters to see which time of the year has higher probability of car accidents. Similarly for INCDTTM, it can be changed into early morning, morning, afternoon, evening, night, and late night.

* 1. **Final Data Configuration**

After feature engineering, we perform a Correlation Matrix as the last step to check for any correlated columns (Pearson Correlation Coefficient > 0.9). The table below shows that none of the column is correlated to other.

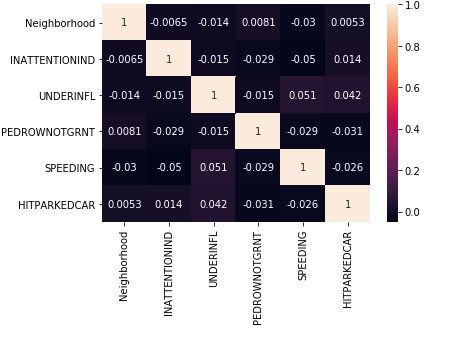


Figure 1. Correlation Matrix of Feature Columns

Our final data set will include:

* Label: SEVERITYCODE
* Features: Neighborhood, ADDRTYPE, QUARTERS, PERIODS, INATTENTIONIND, UNDERINFL,

WEATHER, ROADCOND, LIGHTCOND, PEDROWNOTGRNT, SPEEDING, and HITPARKEDCAR

One Hot Encoding technique will be applied to Neighborhood, ADDRTYPE, QUARTERS, PERIODS, WEATHER, ROADCOND, and LIGHTCOND.

1. **Data Analysis**
   1. **Relationship between Accidents and Location**

We start exploring relationship between accidents and location by graphing the accident counts for each neighborhood zone created from K-Means method above.

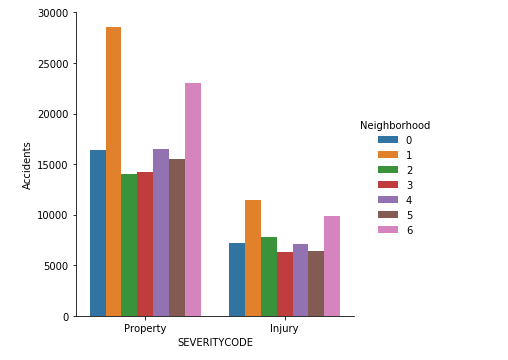


Figure 2. Number of Traffic Accidents per Zone

From figure 2, we can see that zone 1 has the highest number of accidents. Using Folium package, we see that zone 1 is within Seattle downtown area, where the speed limit is low due to high volume of pedestrian (worker and/or tourists).

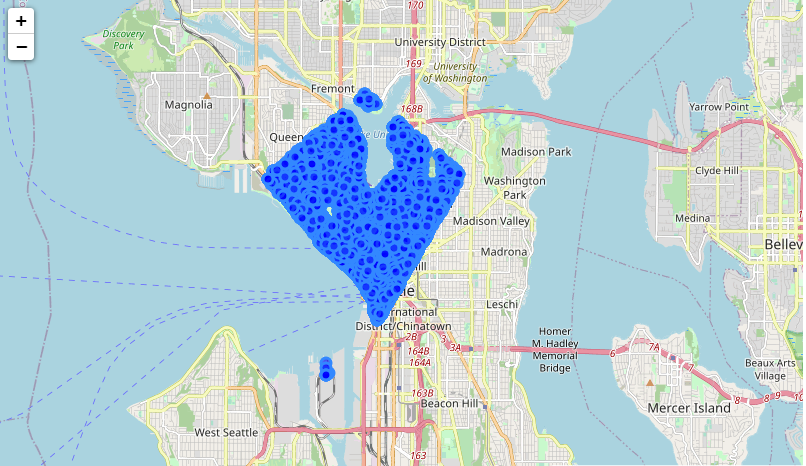


Figure 3. Zone 1 Accidents

Zone 6 is the second highest area with car accidents, which can also be explained by their proximity to zone 1 and the downtown area. They are also near the two main highways connect downtown to suburban, which increased the risk of traffic accident during rush hours.

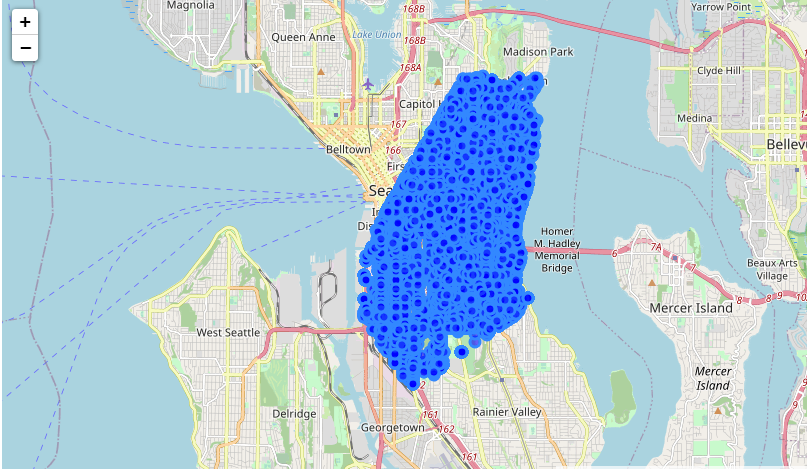


Figure 4. Zone 6 accidents

* 1. **Relationship between Accidents and Date/Time**

With regard to quarters of the year, the accident count doesn’t vary much between each. Figure below shows quarter 3 and 4 have more accidents, which may be due to higher volume of vacation traveling for summer (Q3), and holiday celebration/drinking driving for winter (Q4).

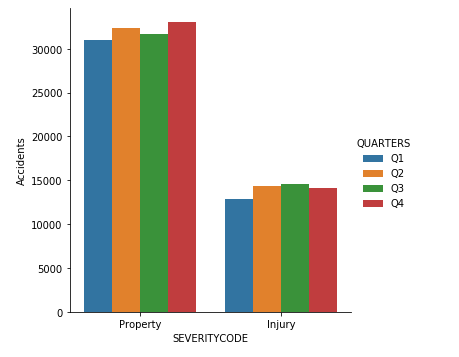


Figure 5. Traffic Accidents by Quarter of the Year

On the other hands, data for period of the day is more conclusive. The afternoon period from 4pm to 8pm has highest number of accidents, which coincides with rush hours during weekdays. Meanwhile, evening has more accidents with injuries, which is a reasonable considering driving and driver conditions in real life.

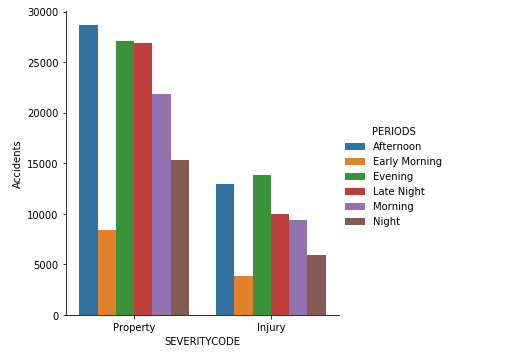


Figure 6. Traffic Accidents by Period of the Day

* 1. **Relationship between Accidents and Human Factors**

One surprised information from this dataset is relationship between accidents and various human factors such as inattention, under influence, and speeding. The dataset showed majority of the accidents did not involve the factors listed above (Figure 7). Additional data might need to be collected since the finding is somewhat contrast to the statistics collected by various government agencies.

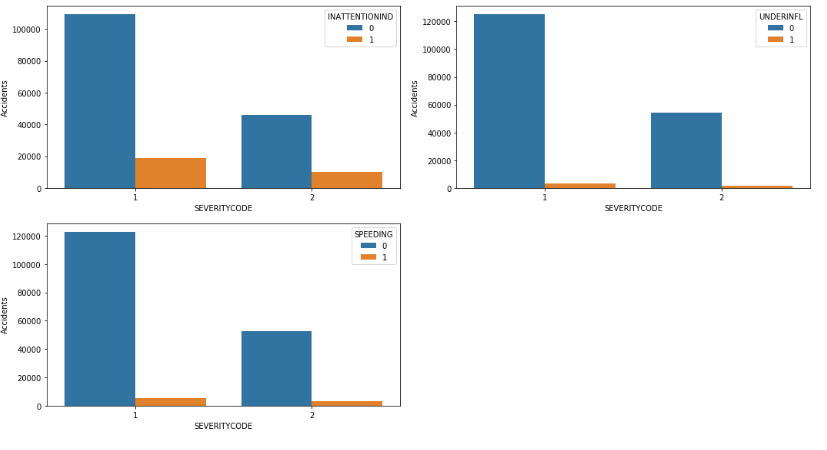


Figure 7. Traffic Accidents relationship with Human Driver Factor

* 1. **Relationship between Accidents and Address Types**

It seems block has more traffic accidents in 3 categories of severity codes, with only a slight different behind Intersection for low-injury (Figure 8). This makes sense, as even though intersection might seem to have more potential for accidents, there are also devices and regulation such as traffic light and stop sign to reduce the risk. On the other hand, block address possibly have more random factors such as pedestrian crossing, parking car, etc. that produces more car accidents.

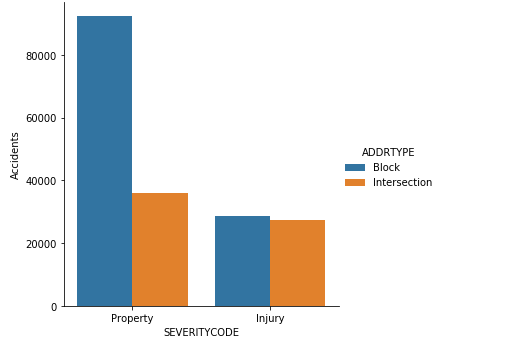


Figure 8. Accident for Different Address Types

1. **Predictive Modeling**

We will employ K-Nearest Neighbor (KNN), Decision Tree, Support Vector Machine (SVM), and Logistic Regression and compare their performances.

* 1. **K-Nearest Neighbor**

We will use Grid Search to find the best k value for KNN model. It is important to note that KNN method from Sklearn library works best with less than 20 columns, hence we will need to use Principal Component Analysis (PCA) to reduce the dataset dimensionality.

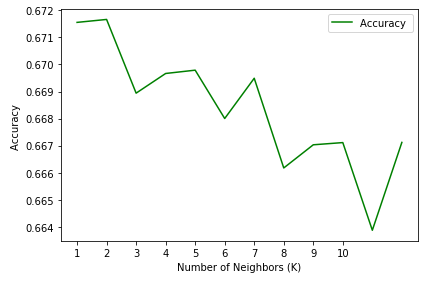


Figure 9. Accuracy Scores for KNN Model

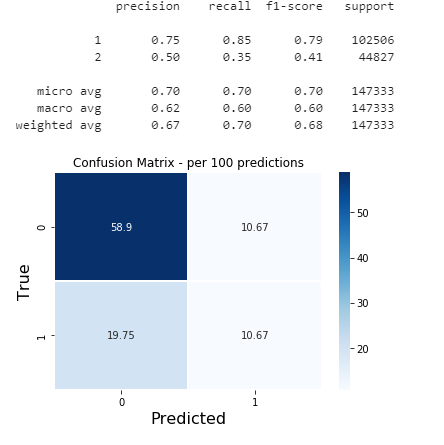


Figure 10. Confusion Matrix for KNN model

The charts above showed the best k value is 7, after PCA reduced to 5 features from our initial dataset.

* 1. **Decision Tree**

We use similar Grid Search technique for Decision Tree, and find the best max\_depth value at 5.

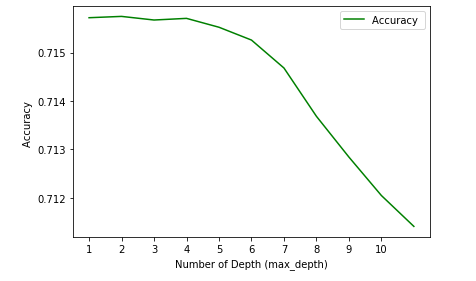


Figure 11. Decision Tree Grid Search

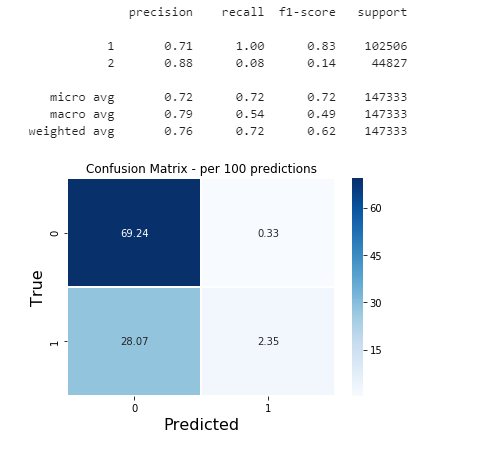


Figure 12. Confusion Matrix for Decision Tree

* 1. **Support Vector Machine**

Sklearn library supports different kernels for SVM, but we will use linear to prioritize speed in handling large dataset (more than 50,000 rows). In fact, instead of the regular SVC class with linear kernel, we will use LinearSVC from SVM library to better scale this large dataset.

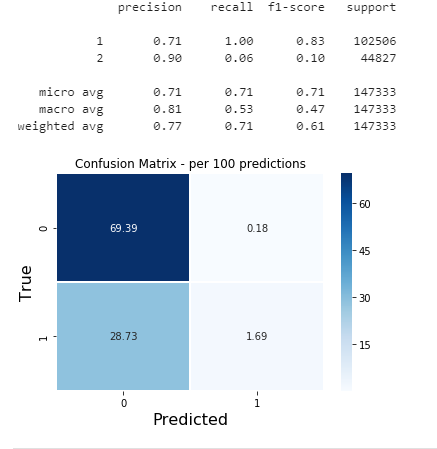


Figure 13. Confusion Matrix for SVM

* 1. **Logistic Regression**

Similar to some of the models above, we continue to employ Grid Search to find the best parameter for Logistic Regression model. The result was newton-cg as solver, and inverse of regularization strength of 0.001.

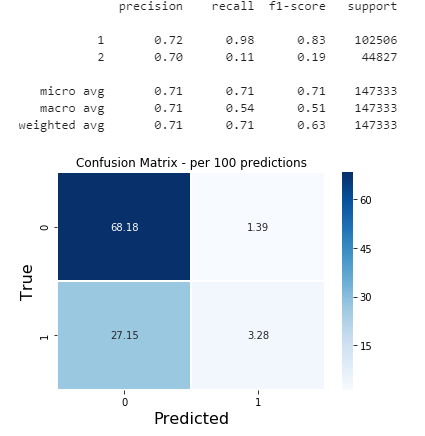


Figure 14. Confusion Matrix for Logistic Regression

* 1. **Models Performance Comparison**

To compare the performance of models above, we will apply them toward the test dataset and compare various accuracy metrics. Based on F-1 score, K-nearest Neighbor has the best performance and by a large margin (a 7.5% different from the next best model).

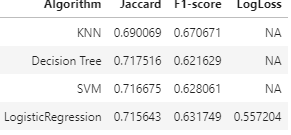


Figure 15. Accuracy Metrics for Predictive Models

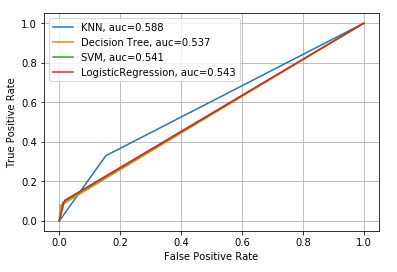


Figure 16. ROC Curves for Predictive Models

The ROC curve was also used to compare performance of our models. As life loss in any traffic accidents always has more devastating effect than financial/property loss, we will SEVERITYCODE 2 as positive label for the ROC curve. In a sense, we want a model with higher True Positive Rate – that is it can correctly identify a situation where injury/fatalities are more likely to happen. From figure 16, it looks like KNN model is the best suitable model for our objective. The highest AUC value further reinforce our conclusion.

1. **Conclusion**

For this project, we attempt to build a Predictive Model for potential traffic accidents. With data from Seattle Department of Transportation, I was able to select some important features to use for the model training sets. In all, the twelve selected features represent all location, time, weather, and human factor that will heighten the risk of an accident. Among four models built for this project, K-nearest Neighbor has the best performance in all meaningful accuracy metrics. It is important to note that in some case, human and/or AI driver might find predicted probability from Logistic Regression more useful and flexible when they can take input and determine the correct course of action based on past experience/training. Obviously, this will require more collected data for further model tuning.

1. **Future Direction**

For this project, I was able to achieve an accuracy F1-score of 67% with KNN predictive model. However, there are still questions about this dataset and whether more can be collected in order to improve model performance. For example, as discussed above, columns representing human factors such as inattention, under influence, and speeding did not really align with reality where they should heighten risk of vehicle accidents.

On the other hand, this project can also tackle a more challenging but useful of multiple classification. There are columns such as PERSONCOUNT and VEHCOUNT that can determine the severity of an accident further. Instead of SEVERITYCODE 1 for property-only, we can define the label to a value of 3 when there are 3 or more vehicle in VEHICOUNT. Similarly, a FATALCOUNT to account for fatality will get a new value of SEVERITYCODE to account for a much higher level of severity. Predictive models with these new labels can offer better input in certain situation, where certain courses of action can reduce the seriousness of potential accident instead of offer general guidance without any specific benefit to the drivers.