Memorandum
September 20, 2020

To: Seattle, WA city managers

From: Sean Moffat

Subject: Predicting accident severity in Seattle using supervised machine learning (SML)

1. Background and Introduction

Traffic accidents across the United States are a major issue, especially so in large cities, such as Seattle. Traffic accidents cause immediate direct damage to those involved and indirect and possibly long-lasting for those not directly involved. Of course, damages can range from minor scratches to most severe, loss of life.

Being able to accurately predict accident severity could be a useful tool for city managers and the rest of the population as well. With a shared understanding of traffic accident facts, the general public and city managers can develop data-driven policies to help mitigate accidents which should lead to fewer accidents with injuries.

This analysis will use supervised machine learning (SML) classification models to predict the severity of accidents, where severity is defined as an accident resulting in an injury. Additionally, I will use standard logistic regression to provide additional information / perspective on the relationship between accident severity (the dependent variable) and our feature set (the independent variables).

2. Data

For this case study, I am using the "Data-Collisions.csv" data set from the course website. This set contains traffic collision data from Seattle, WA covering accidents from January 1, 2014 – April 29, 2020. This set contains 38 features and after data cleaning and data preparation, I have 10 features left for modeling. Tables 1 and 2 below present information on the label and feature sets, respectively, that were used in the analysis. The data discussion is continued in 2.1) Data collection and preparation and 2.2) Data understanding.

Table 1. Label set

Original values of SEVERITYCODE	Feature engineered to be *binary* number
'Property Damage Only Collision'	0
'Injury Collision'	1

Table 2. Feature set

Ref.	Feature	Description	Type
01	COLLISIONTYPE_TO_NUM	Collision type.	Numeric unit interval (10 values)
02	PERSONCOUNT	The total number of people involved in the collision.	Continous
03	PEDCOUNT	The number of pedestrians involved in the collision.	Continous
04	PEDCYLCOUNT	The number of bicycles involved in the collision.	Continous
05	VEHCOUNT	The number of vehicles involved in the collision.	Continous
06	HITPARKEDCAR_TO_NUM	Whether or not the collision involved hitting a parked car.	Binary. 1 = yes, 0 = no
07	UNDERINFL_TO_NUM	Whether or not a driver involved was under the influence of drugs or alcohol.	Binary. 1 = yes, 0 = no
08	WEATHER_TO_NUM	A description of the weather conditions during the time of the collision.	Numeric unit interval (11 values)
09	ROADCOND_TO_NUM	The condition of the road during the collision.	Numeric unit interval (9 values)
10	LIGHTCOND_TO_NUM	The light conditions during the collision.	Numeric unit interval (9 values)

2.1. Data collection and preparation

I use pandas.read_csv() function to read the source csv file which I downloaded from the course website. With the data in a pandas dataframe I start my exploratory data analysis by checking data elements for:

- **Missing data**: My decision rule is to drop whole data columns if more than 40% of observations are missing and additionally, drop whole rows if data is missing from key features;
- **Convert date/time elements**: To take advantage of useful Python and pandas functions, it is important to store dates as date formats. From this, I was able to create other date features, such as year-month combination (useful later to summarize data); and
- **Data transformation / feature engineering**: At this step, I create new features from existing features. For text and alphanumeric features, I create new features that are numeric version of existing features; numeric features will be used in the modeling process.

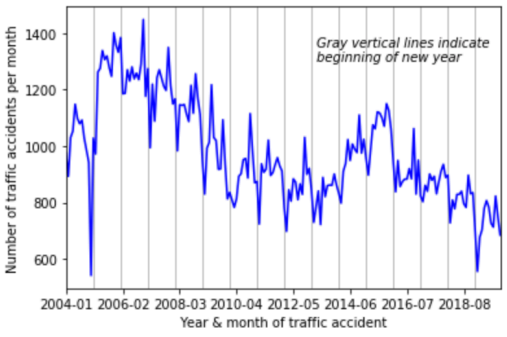
In the final step of data collection and preparation phase, I create two new dataframes, **train / validate** set and an **out-of-sample (OOS) test** set. I created these two dataframes from the existing dataframe by slicing data using incident date element. The train / validate dataframe will be ***used to train and test the SML models***; this set contains records with incident dates between Jan. 1, 2004 and Dec. 31, 2019 and has 188,061 records. The OOS test dataframe will be ***used to evaluate model performance*** by using data the model has not seen; this set was populated from records with records with incident date greater than or equal to Jan. 1, 2020, there are 1,255 records in this set.

2.2. Data understanding

To gain a better understanding of the cleaned and processed data, I create a few time series charts to illustrate accidents across time. I'm interested in understanding data in the **train / validate** data set, this data set contains 16 years of data, at the day level.

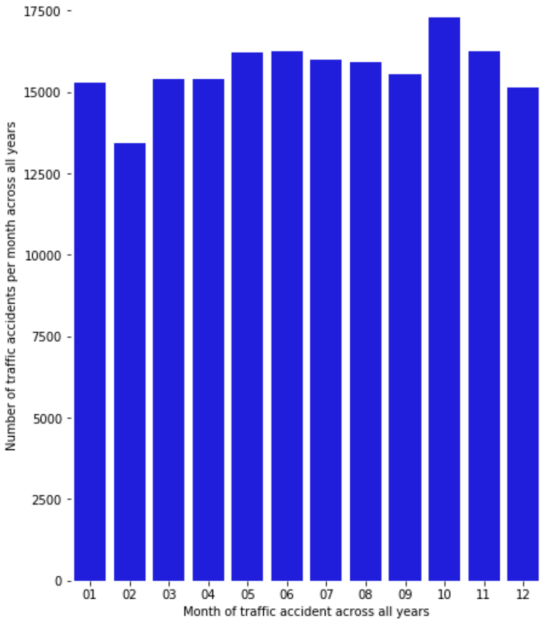
In Figure 1, one can see a distinct downward trend in traffic accidents at the year/month level, with an occasional up-tick, followed by another down trend. From this figure, it is difficult to visually determine if there's seasonality in the data.

Figure 1. Traffic accidents across time
General downtrend, with year end spikes with uptrend in 2014 & 2015



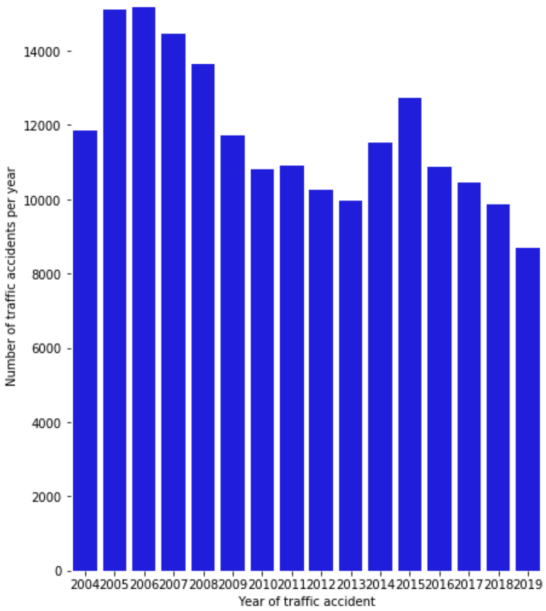
To help visually determine if there is a seasonal pattern, I created Figure 2, which plots total traffic accidents by month. Here, data suggests possible seasonality with traffic accidents in February and October.

Figure 2. Total traffic accidents by month between 2004 & 2019
Data suggests possible seasonality with traffic accidents in Feb. & Oct.



Lastly, I wanted to get a clear picture of traffic accidents volume by year. Figure 3 plots total traffic accidents by year. This chart tells one that in general, there has been a year-over-year (YoY) decline in traffic accidents, with 2014 and 2015 as exceptions.

Figure 3. Total traffic accidents by year
In general, there has been YoY decline in traffic accidents, with
2014 & 2015 as exceptions



3. Methodology

Recall, the focus of this capstone case study is to predict the severity of an accident; for this case study, I have defined severity as an accident involving an injury. I **use supervised machine learning (SML) as the analytical approach to predict the severity of an accident**. I selected SML in part because of the variety of models available, including logistic regression, k-nearest neighbors (k-NN), support vector machine and decision tree. I used logistic regression and k-NN for the case study.

Modeling will be done using the train / validation dataframe (df_train_validate) while **evaluation** will be conducted using the out-of-sample (OOS) test dataframe (df_oos_test). I cleaned and prepared both of these dataframes in the data collection and preparation section of the analysis, no additional preparation is needed for modeling or evaluation.

Tables 1 and 2 from the data section present information on the label and feature sets, respectively, that were used in the analysis.

Before I build a model, I split the train / validation dataframe into train and test sets and standardize the train and test feature sets. Train and test were stratified based on the label (SEVERITYCODE) to ensure I had a consistent makeup of label outcomes in each set.

3.1. Modeling

I ran two types of models for this analysis, standard logistic regression and SML classification models, only the SML models will be used for prediction. I've included standard logistic regression to provide additional information / perspective on the relationship between accident severity (the dependent variable) and our feature set (the independent variables). The additional information I'm looking for include sign on the estimated coefficient and the p-value. I'm using a significance level (denoted as α or alpha) of 0.05 to determine if the association between the dependent and independent variables are statistically significant. Somewhat surprisingly, the association between accident severity and weather is not statistically significant (p-value 0.164). Weather is only feature variable that is not statistically significant.

```
Regression with discrete dependent binary variable: modeling traffic accidents using non-standardized data
-------------------------------------------------------------------------------------------------------------

Optimization terminated successfully.
         Current function value: 0.538190
         Iterations 7

Estimate of Logit model, re: estimated coefficients and standard errors

                           Logit Regression Results
==============================================================================
Dep. Variable:     SEVERITYCODE_TO_NUM   No. Observations:               188061
Model:                           Logit   Df Residuals:                   188050
Method:                            MLE   Df Model:                           10
Date:                 Sun, 20 Sep 2020   Pseudo R-squ.:                 0.1204
Time:                         01:32:00   Log-Likelihood:            -1.0121e+05
converged:                        True   LL-Null:                   -1.1507e+05
Covariance Type:             nonrobust   LLR p-value:                     0.000
============================================================================================
                               coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------------------
const                    -2.1502      0.027    -79.569      0.000      -2.203      -2.097
COLLISIONTYPE_TO_NUM      0.0771      0.003     27.430      0.000       0.072       0.083
PERSONCOUNT               0.2407      0.005     49.041      0.000       0.231       0.250
PEDCOUNT                  2.8794      0.045     63.971      0.000       2.791       2.968
PEDCYLCOUNT               2.7896      0.046     60.888      0.000       2.700       2.879
VEHCOUNT                  0.1365      0.012     11.282      0.000       0.113       0.160
HITPARKEDCAR_TO_NUM      -1.5567      0.053    -29.399      0.000      -1.660      -1.453
UNDERINFL_TO_NUM          0.5912      0.025     23.892      0.000       0.543       0.640
WEATHER_TO_NUM            0.0060      0.004      1.393      0.164      -0.002       0.014
ROADCOND_TO_NUM           0.0159      0.003      4.802      0.000       0.009       0.022
LIGHTCOND_TO_NUM         -0.0365      0.003    -14.296      0.000      -0.041      -0.031
============================================================================================
```

The train / validate dataframe will be used to train and test the SML models; this set contains records with incident dates between Jan. 1, 2004 and Dec. 31, 2019 and has 188,061 records. Summary results from modeling are presented in Table 3, for complete details, please see the modeling section in the Jupyter notebook.

Table 3. Evaluation of SML models on train / validation sets

Evaluation metirc	Model 1: Logistic regression	Model 2: k-nearest neighbors (k-NN)
Accuracy classification score (train set)	0.749867	0.753171
Evaluation of SML models on validation set		
Jaccard similarity coefficient score	0.210730	0.288048
F1 score	0.348104	0.447262
Log loss	0.538100	Not applicable
Confusion matrix from validation set		
True positives	22.14%	33.94%
False negatives	77.86%	66.06%
False positive	2.19%	7.68%
True negatives	97.81%	92.32%

3.2. Evaluation

The out-of-sample (OOS) test dataframe will be *used to evaluate model performance* by using data the model has not seen; this set was populated from records with records with incident date greater than or equal to Jan. 1, 2020, there are 1,255 records in this set.

Through evaluation, I want to understand and quantify prediction accuracy of SML logistic regression and k-NN models I built using the train and validate data set. I accomplish this goal by applying these models to the OOS test dataframe. Summary results from evaluation are presented in Table 4 in the results section, for complete details, please see the evaluation section in the Jupyter notebook. Figures 4 and 5 present the confusion matrix for each model.

Figure 4. Normalized confusion matrix for SML logistic regression
Using GridSearchCV parameters

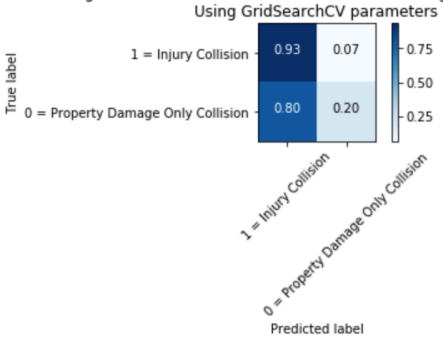
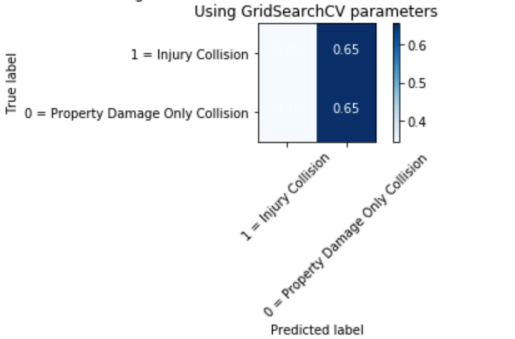


Figure 5. Normalized confusion matrix for SML k-NN
Using GridSearchCV parameters



4. Results

I considered two SML classification models, logistic regression and k-NN, to address the question of predicting severity of an accident. The evaluation of these models on out-of-sample test data suggests that the logistic regression is the best classifier, based on Jaccard score, F1 score. Summary evaluation metrics are reported in Table 4.

Table 4. Evaluation of SML models on out-of-sample (OOS) test set

Evaluation metirc	Model 1: Logistic regression	Model 2: k-nearest neighbors (k-NN)
Jaccard similarity coefficient score	0.343808	0.198276
F1 score	0.511692	0.330935
Log loss	0.859434	Not applicable
Confusion matrix from OOS test set		
True positives	93.47%	34.67%
False negatives	6.53%	65.33%
False positive	79.81%	34.77%
True negatives	20.19%	65.23%

5. Discussion

Future analysis enhancements can be grouped into three bucket. First, if I believe the "Property Damage Only Collision" collision outcome is overrepresented, I could use a sub-sample of data for modeling and evaluation. Second, to get more detailed outcomes I could use feature engineering to transform the label from a binary outcome into a multinomial, using an ***ordinal*** scale. This would allow me to get more severity detail, such as property damage and amount of dollar damage. Third, I could increase the number of parameters used in grid search with cross-validation. Fourth, I could consider additional SML classifier models such as:

* Support Vector Machine (SVM);
* Decision Tree; and
* Neighborhood Components Analysis (NCA)

And fifth, I could collect additional feature data elements and determine correlation between variables or run principal component analysis (PCA) as a way to reduce the number of features to the most impactful set.

6. Conclusion

In this analysis, I started statistical modeling with a standard logistic regression to provide additional information / perspective on the relationship between accident severity (the dependent variable) and our feature set (the independent variables). I then moved to SML without and with hyper-parameter tuning to predict accident severity.

From the standard logistic regression, I learned, somewhat surprisingly, the association between accident severity and weather is not statistically significant (p-value 0.164). Weather is only feature variable that is not statistically significant.

In regard to SML, the logistic regression was the best performing classifier, at 93.47% true positives. However, this performance appears to come at a price, with false positives ~80%.

That being said, in light of the SML logistic results, I plan to initiate additional work related to addressing possible overrepresentation "Property Damage Only Collision" collision outcome & trying additional estimation solvers.