**Predicting Accident Severity for the City of Seattle**

Phillip Asberry

September 21, 2020

**1. Business Understanding**

* 1. **Background**

The city of Seattle is the seat of King County Washington and the largest city of the state of Washington. With a population of 608,660 (according to the 2010 census) and a metropolitan area of 4 million, Seattle is one of the fastest growing regions in the US. As growth continues, traffic is a major concern.

Traffic pattern analysis and optimization is critical for the efficient flow of goods and services. With this information, Seattle can better facilitate the needs of its citizens. One element of traffic analysis is accident avoidance and mitigation. Accidents are an unfortunate part of any urbanized region. Along with the human cost, accidents can exacerbate traffic which can impede emergency personnel response. Predicting when and where potential accidents can happen would help the region of Seattle maintain growth encouraging commerce to proceed unencumbered.

* 1. **Problem**

A combination of weather conditions, season, day of the week, location, and vehicles involved can help make the determination of an accident’s severity. This project aims to predict the likelihood of severe accidents that could lead to traffic disruption.

* 1. **Interest**

This project would potentially be of interest to all parties associated with regional transportation in the Seattle metropolitan. With better understanding SDOT could predict when to deploy assets such as salt trucks. The mayors of the region could better lobby the state and federal governments for funds to improve infrastructure.

**2. Data Understanding**

**2.1 Sourcing Data**

The data for this project was provided by the SDOT Traffic Management Division, Traffic Records Group. This data set includes all types of collisions from the year 2004 to present.

**2.2 Data Analysis and Feature Selection**

Upon inspection of the data, the data was incomplete in many columns. The choice was made to simply drop underrepresented columns and fill in missing data with the mean or mode where appropriate. In addition, all columns that provide descriptions and codes were dropped as well. The intent for the feature set was to focus on a small but expressive set of data points.

Columns **SPEEDING, PEDROWNOTGRNT, EXCEPTRSNCODE, EXCEPTRSNDESC, INATTENTIONIND** and **INTKEY** were dropped from feature consideration because they only showed up in at most 3% of the 194,673 records.

**ADDRTYPE, and JUNCTIONTYPE** were dropped in favor of keeping the **X** and **Y** coordinates of the accident.

**JUNCTIONTYPE**, **SDOT\_COLDESC**, **SDOT\_COLCODE**, **ST\_COLCODE** and **ST\_COLDESC** were viewed by the project as descriptive columns that provided annotations for the row. Beyond documentation, they were not deemed important features and were dropped.

The following features described in subsequent paragraphs composed the feature matrix X.

The columns **X** and **Y** represent the geographical coordinates of the incident. The data was scaled via Min/Max normalization with the missing data replaced with the mode of each column. The assumption to use the most popular **X, Y** tuple for accidents without coordinates stemmed from the assumption most accidents in the city probably happened in a rather common area for all accidents.

The assumption to use the mode for missing values was used throughout the dataset.

**VEHCOUNT, PERSONCOUNT, PEDCOUNT, PEDCYLCOUNT** were treated in the same fashion as the coordinate columns **X** and **Y**. The features were Min/Max scaled and null values were replaced with the mode.

**VEHCOUNT** described the number of vehicles involved in the accident.

**PERSONCOUNT** described the total persons involved in the accident.

**PEDCOUNT** described the total pedestrians involved in the accident.

**PEDCYLCOUNT** described the total number of bicycles involved in the accident.

The **INCDATE** column contains timestamp data about the incident. The date information from the column was parsed and split into to new columns **‘Incident Month’** and **‘Incident Day of Week’** representing the month of the year and the respective day of the week the accident occurred. The valid month values are January, February, March, April, May, June, July, August, September, October, November, and December. The valid days of the week are Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday.

The **ROADCOND** column represents a categorical list of conditions for the roads at the time of the accident. The list of values was Dry, Wet, Unknown, Ice, Snow/Slush, Other, Standing Water, Sand/Mud/Dirt and Oil. The decision was made to simplify the model by segregating the road conditions into two categories, Fair or Hazardous. This decision was made to lessen the dimensionality of the matrix features. All missing or unknown values were substituted with the mode.

The **WEATHER** column was also reduced categorically to two options, Clear and Inclement. The previous categories were Clear, Overcast, Unknown, Other, Partly Cloudy, Raining, Snowing, Fog/Smog//Smoke, Sleet/Hail/Freezing Rain, Blowing Sand/Dirt, or Severe Crosswind. The values Clear, Overcast and Partly Cloudy were assigned to Clear, the others were delegated to Inclement weather. All unknown and null values were assigned to the mode of the column values.

The **LIGHTCOND** column contained the values Daylight, Unknown, Dusk, Other, Dark – Street Lights On, Dark – Street Lights, Dark – Street Lights Off, and Dark – Unknown Lighting. With the goal of dimensionality reduction, the values were reduced to two values, Day and Night. All values Daylight, Dusk, Dawn, were consider Day and all values prefixed with Dark were considered Night. All other unknown and null values were assigned to the mode.

**UNDERINFL** indicated whether any party involved were under the influence of alcohol. These values originally composed of a combination of 0 and No for negative cases and 1 and Yes in positive cases. The values were streamlined into either Yes or No. If a value was not present, the value was No. The assumption being if a driver were involved in an accident while intoxicated, it would have been noted. Driving under the influence in America is a serious offence.

**3. Modeling**

**3.1 Methodology**

Using pandas *get\_dummies()* to one-hot encode the categorical data described prior, a 194,673 x 36 feature matrix was the result.

The project philosophy for model building lay in a 90% split for Training and Dev sets and a 10% test set. Within the 90% Training/Dev split, the data was further composed into 3 folds. Using K-fold cross-validation 1/3 of the data served as a Dev/Validation set while the other 2/3 were used for model training.

The main reason for this approach to training our models is the relatively small size of our dataset. Dedication of at least 128,400 training and 64200 dev/cross-validation sets per fold were deemed adequate for training. The approximately 19,400 left for testing provided a decent sample to score the models.

The other issue that the project had to acknowledge was the imbalanced nature of the data. Due to this imbalance, two strategies were employed. The model was first developed with the imbalanced dataset and then again with an oversampling of the minority class.

The binary nature of the problem domain, (SEVERITY 2 – injury, SEVERITY 1 – prop damage) meant a classification approach to a solution was appropriate.

The first of the classification models used was a simple Logistic Regression. Its simple straight forward nature would provide a baseline for all other algorithms used during modeling.

The next algorithm deployed in modeling was a Support Vector Machine with a Linear Kernel. The support vector machine was used to improve upon the simple Logistic Regression for its versatility in learning complex decision boundaries. This would be useful for complex data sets of higher dimensionalities.

Third, the project made use of ensemble learning with both Random Forest Classification and Gradient Boosting. Decision Trees are a particularly good modeling approach to problems that consist of at lot of categorical data, the Seattle accident data is one such data set.

Lastly, the project took advantage of Neural Networks to model the problem of the Seattle accident dataset. Like Support Vector Machines, Neural Networks provide the ability to learn complex decision functions.

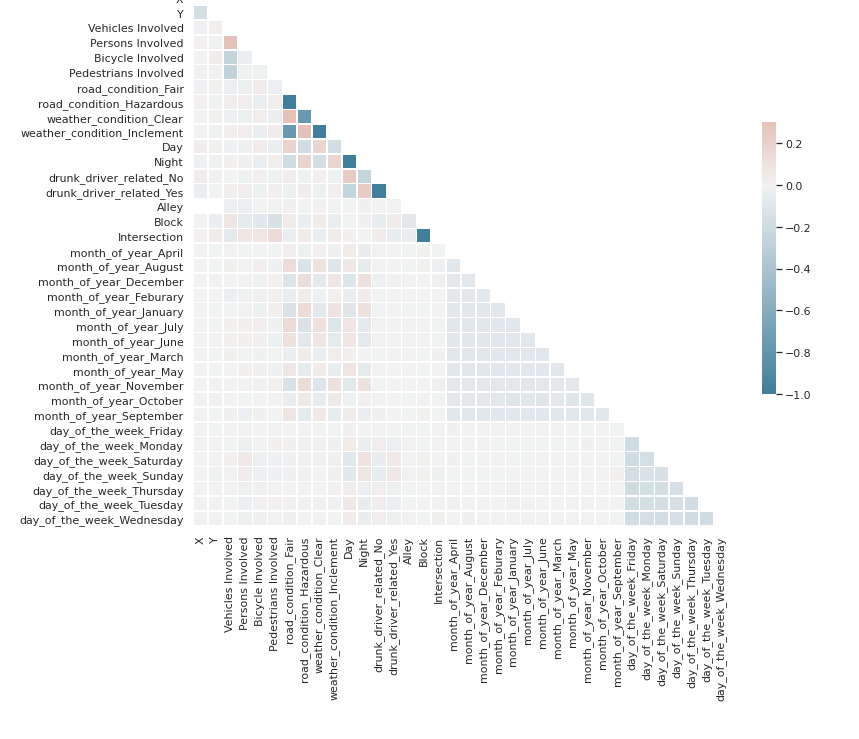
**3.2 Data Exploration**

The chart below displays the correlations between the features of the input matrix X. Taking a cursory glance at the correlations between features yields common sense explanations.

The Persons Involved in the accident has a positive correlation with the number of vehicles involved in the incident. This is a conclusion while confidently laid out in the data, would seem to follow with most drivers understanding of the roadways.

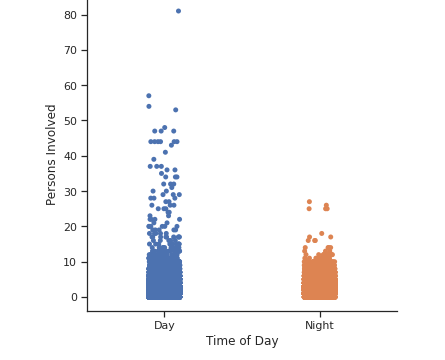
Born of the data set is the observation that clear whether positively correlates to fair road conditions while inclement weather yields higher chances of hazardous road conditions. Both sets of conditions can affect the situation of traffic and in turn the likelihood of traffic incidents. Since Seattle is known for its rain, knowing the odds of inclement weather can give insight into future road conditions.

The data exhibits most drunk driving accidents occur during the night. Drunken accidents most likely occur at night since that is the time most people have dinner, celebrate and gather socially. Though not evident in our crash analysis, it is accepted in most Western societies’ alcohol consumption increases with large gatherings.

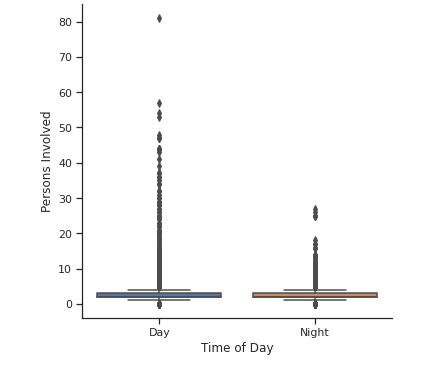
****

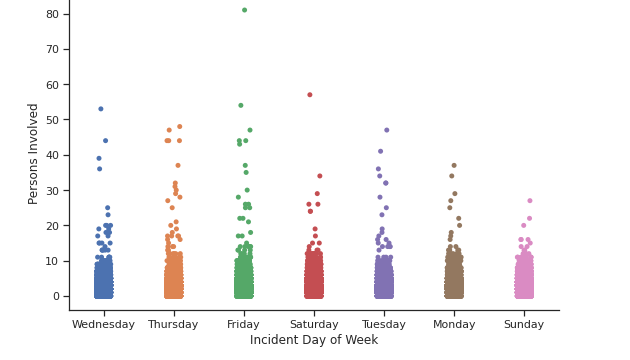
Accidents during the Fall and Winter correlate positively with driving at night with hazardous road conditions. Late fall and winter nights tend to have more snowy and icy conditions. Conversely, summer accidents tend to occur on clear weather days on fair road conditions. Summers in the United States consist of family vacations, 4th of July, Juneteenth, and Labor Day weekends. All the events mentioned increase road usage throughout the U.S.

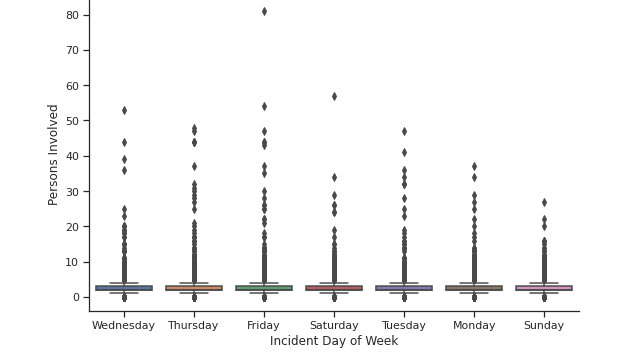
The analysis of the data showed that accidents with the most persons involved occurred during the day. The largest accident in the data set had 80 persons involved.



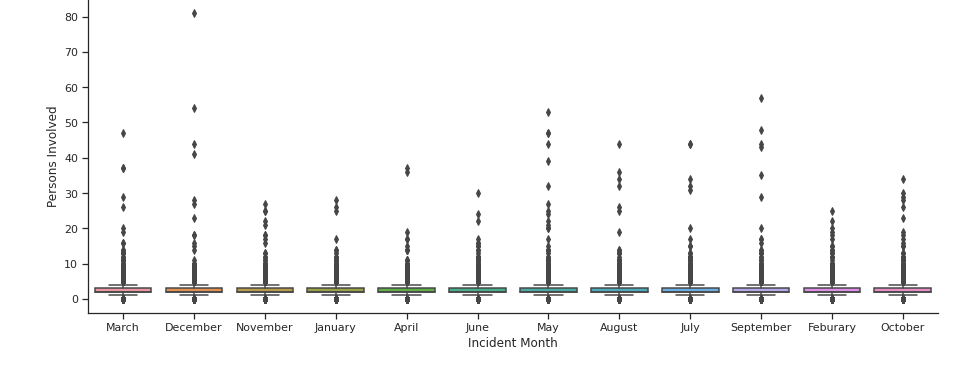
While the occurrence of large accidents occurred during daylight hours, both night and day had a mean of 2 persons per accident.



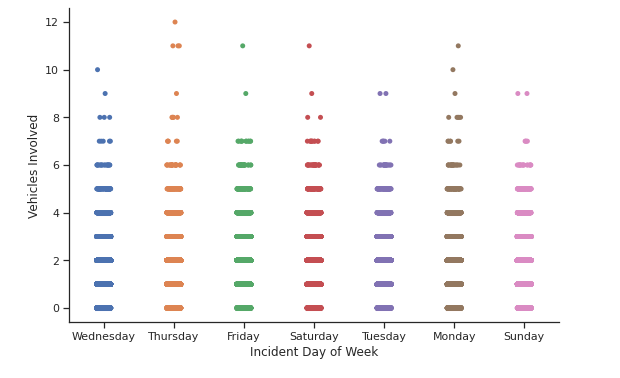
****

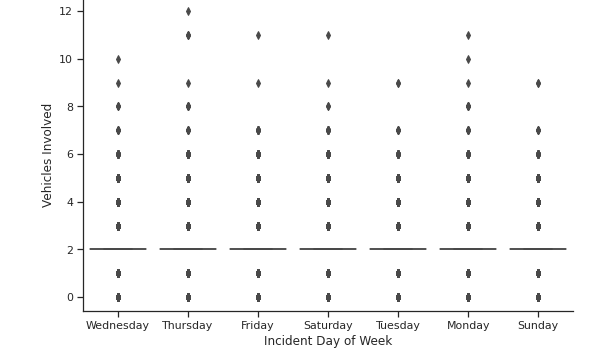
****

According to the records in the Seattle accident dataset, persons involved and the likelihood of being involved in an accident was steady regardless of the day of the week or month.

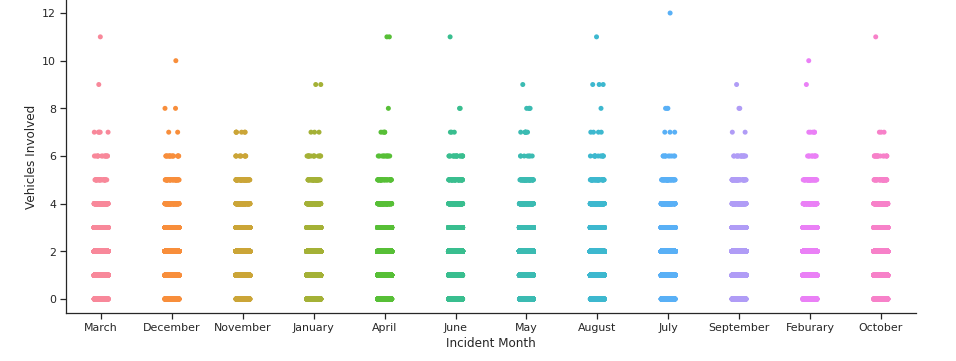


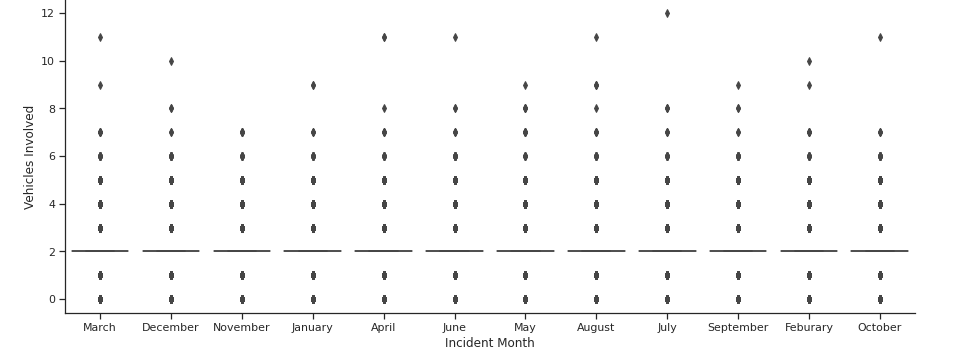
When the data looked at vehicles involved, no day had an advantage in number of vehicles per accident. While the mean number of vehicles per accident was 2, an outlier event exhibited a probability of an occurrence on any given day of the week.



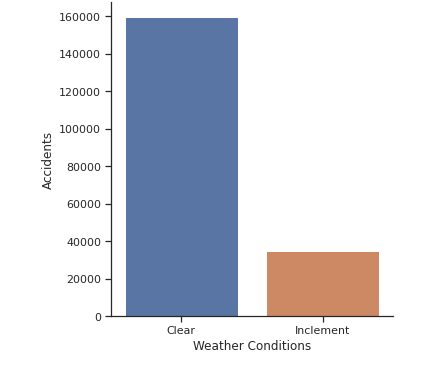


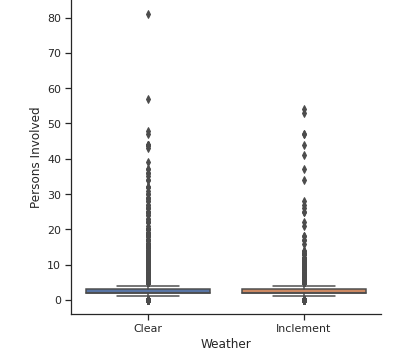
Vehicle per accident numbers showed a similar distribution when extrapolated on a monthly timetable. The mean vehicles per accident was 2 and every month had the same potential to have an outlier event.



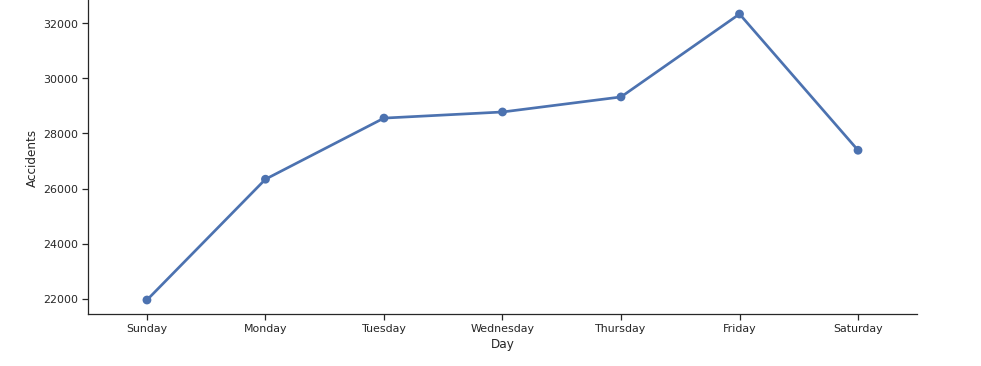


Surprisingly, the data showed that the largest and the most accidents occurred when the weather was clear. Conventional wisdom would have guessed that inclement weather would attribute more to accident size and frequency. A possible answer that is not supported by the dataset could be traffic speeds are decreased (the assumption being drivers are more careful) during inclement weather mitigating the severity of any accident occurrence.





Sunday has the least amount of accidents while Friday has had the most accidents during the timeframe of the dataset. The rise in accidents on Friday could be due to Friday had the second highest occurrence of drunk drivers. Sundays lack of incidents could in part be due to the American tradition of Sunday being a day of rest and family.



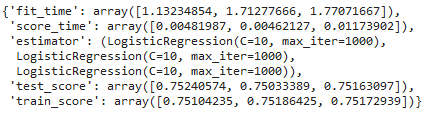
**4. Results**

**4.1 Overview**

All modeling for the problem domain was done using the sci-kit learn Python library. The data set was broken into a training, dev/validation, and a test set for model evaluation and building. K-fold (k = 3) validation was the preferred method of evaluating our dev/validation set against our training examples.

**4.2 Logistic Regression**

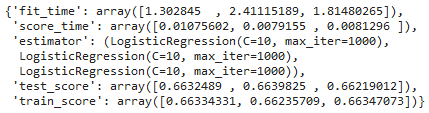
The Logistic Regression of the data set resulted in the following results.

****

The training set accuracy is low, (highest value was .7518) which signals that the dataset is experiencing some bias and is slightly underfitting the data. The validation accuracy (described as test\_score) is approximately the same as the training set scores. This data underscores the underfitting hypothesis. Finally, when given the unseen observations set aside in the test set, the final predictive accuracy of the model was 71.45%. The ROC of AUC was 0.7145 and the F1 score was .36390.

A low F1 score indicates our model has both inferior precision and recall.

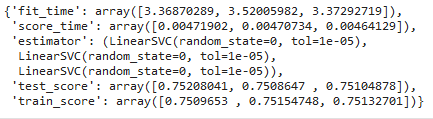
To combat the under representation of the minority class, the next round of Logistic Regression oversampled the minority class. The results are below.



With the oversampling, both the training and dev set accuracies decreased. This observation will be something observed in all oversampled models performed by the project. Contrary to the lower training and dev set accuracies, when provided with data unseen by the model the accuracy was 71.42%, not as low as the accuracies of the other sets of data. The ROC of AUC was consistent with the undersampled data at 0.715. The model did have a higher F1 score of 0.4972. The oversampled data lead to a much better algorithm if F1 was the primary metric of analysis.

**4.3 Support Vector Machine with a Linear Kernel**

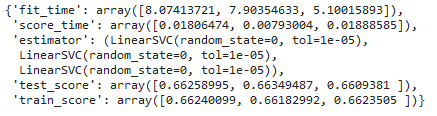
Performance of the SVM with the unbalanced dataset can be seen below.



The linear kernel of the support vector machine had similar performance as the Logistic Regression. The average accuracy of the training set was 75% and the validation set average was also 75%. When given the unseen data from the test set the model generalizes to an accuracy of 75%. The ROC of AUC was 0.71394 with a F1 score of 0.345. By the metric of F1 score, this model performs poorly.

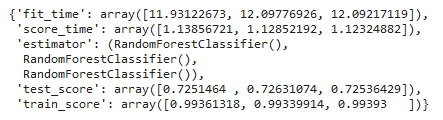
Oversampling the smaller class, the training and validation set accuracies dropped just like in the earlier model of the Logistic Regression. Like the oversampled Logistic Regression, the accuracy was approximately 71.9%. The ROC of AUC was 0.7193 and the F1 was 0.4934 indicating a much better model than the unbalanced dataset model.

Here are the results.



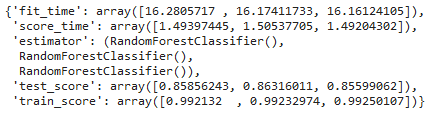
**4.4 Random Forest Classification**

The Random Forest Classifier outperformed all the other previous classification models on the training set accuracy. The average accuracy score was 99%. Unfortunately, the dev/validation set accuracy was not as high. The average accuracy of the dev/validation set was around 72%, indicating that the random forest model has overfit the training data.



The model did generalize a slightly better than the validation set would imply, the average accuracy was 73%. The ROC of AUC was the highest at .999 with an F1 score of 0.444. By the metrics of F1 and ROC, the random forest classifier performed very well.

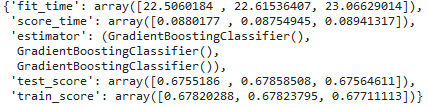
Unlike the other models, oversampling the minority class did not decrease validation/dev set accuracy. As observed below, the oversampling would on the surface appear to create a model that would outperform the other models base on the accuracy metric.



The eventual model had the lowest accuracy of 68.35 %. The ROC of AUC was remarkably high at 99.97%. Also, the F1 score was higher than its unbalanced brethren at 0.49541

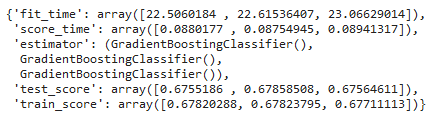
**4.5 Gradient Boosting Classification**

The gradient boosting algorithm seemed to suffer the same problems with bias as the Support Vector Machine and the Logistic Regression.



The accuracy on the unseen data from the test set was 75.63%. The ROC of AUC was 0.75899 and the F1 score was 0.3788

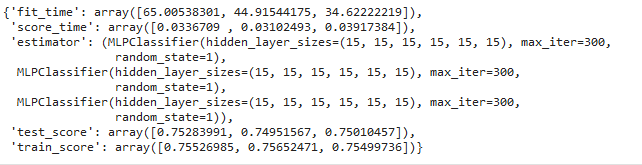
With the addition of oversampling the minority class the performance was in line with the other models. The training score and validation/dev score decreased.



The accuracy of the model decreased to just 71.75% with a ROC of AUC of 0.75898 and the highest F1 score of all the models at 0.5398

**4.6 Deep Neural Network**

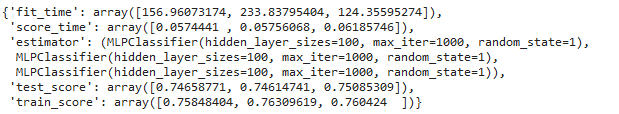
With the other models like Logistic Regression and Support Vector Machine with a Linear Kernel exhibiting signs of potential high bias, the notion of a deep neural network took root. The impetus for such a model stems from the idea the model would have an opportunity to learn new features as the network would be large. Unfortunately, the performance of our deep network classifier was no better than the Logistic Regression or SVM.



The accuracy was 75.249% with a ROC of AUC of 0.7366 and F1 score of 0.388.

**4.7 Shallow Neural Network**

The hidden layers were reduced in favor of having one hidden layer with 100 hidden nodes. Again, the performance was not better than previous attempts.



The accuracy of the model was 75.25% with a ROC of AUC of 0.72433 and an F1 score of 0.39.

**4.8 Other Model Optimizations**

It is worth briefly noting two other model optimization strategies. First was the attempt to fit the data to a higher order polynomial. Due to the system constraints on processing the data, the feature vector was only able to be raised to the order of 2. The second dimensional polynomial performed on par with the non-second-degree features.

Finally, PCA reduction did not help improve the predictive capabilities either. The model could not be reduced and still maintain 95% variance.

**5 Final Thoughts**

**5.1 Conclusions**

The data presented in this report is helpful in identifying factors that can precipitate traffic accidents in the city of Seattle. Factors like season, time of year and day of the week were shown to be just as important as obvious markers as vehicles and persons. City officials can use the insights learned in this report to plan emergency response, traffic planning, and staffing of emergency personnel.

The classification models in the project made predictions that were better than random chance but have plenty of room for improvement. A larger collection of data in the dataset would be a start. More data would give an opportunity to train larger models.

The fine tuning of metrics and potentially finding more numerical data points could make the models more robust. An example of such data points to collect in the future include but not limited to age of cars involved, street of incident, and age of driver. Another change to the models in the project would be shifting the allocations of the Train/Dev/Test split. Using more data for validation and tuning of hyperparameters could have created more predictive models.

In conclusion no model is completely accurate, but hopefully they can be useful.