SEATTLE COLLISION DATA ANALYSIS

1. INTRODUCTION AND BUSINESS PROBLEM

In recent years many municipalities around the world gained access to large sets of traffic / road accident data that is still relatively underutilized. This project showcases how insights gained from road accident data can be used for the benefit of various stakeholders.

The key objective of the report is to predict the severity of the road accident based on the various data points such as weather and road condition, accident coordinates and etc. Analysis is based on the data provided by Seattle municipality and hence would only be relevant to Seattle though the model can still be extended (with relevant modifications) to any other city.

Predictive model can be utilized by individuals who are wondering how severe the road accident can be in a given area with specific road conditions and etc.

Predictions can also be leveraged by municipalities in city planning, e.g. improving road infrastructure in the problematic areas (with high proportion of severe accidents), increasing medical service coverage in such areas and etc.

2. DATA DESCRIPTION

As mentioned previously we will be leveraging data provided by Seattle municipality on road accident / collisions. Data contains a large number of features (37 without the label column). The features contain various info on road and weather conditions, coordinates of the accident, accident code provided by the authorities and etc. The label column is “SEVERITYCODE” which is populated with 1 (meaning property damage only) and 2 (meaning injury). The total number of entries is 194,673

The dataset is mildly imbalanced – ~30% of positive and ~70% of negative results. Hence, we would be required to balance the training sets (would be covered in more details later).

We will use data provided to build a predictive model to predict SEVERITYCODE. We would first clean data and remove redundant features. We would then try out different machine learning methods to build the most accurate / appropriate predictive model.

After first stab at the data cleaning we have decided to remove 19 redundant / non-relevant features. Below you can find reasons for removal:

- Duplication: “SEVERITYCODE.1", “SEVERITYDESK”, "INCDATE"
- Columns with too many missing data entries: "INTKEY", "INATTENTIONIND","EXCEPTRSNCODE","EXCEPTRSNDESC", "SPEEDING", "PEDROWNOTGRNT", “SDOTCOLNUM"
- Columns that have useless info (e.g. ID): "REPORTNO", "INCKEY", "COLDETKEY", "OBJECTID"
- Columns with categorical info that have too many unique values (impractical for one-hot encoding): "ST_COLDESC","SDOT_COLDESC", 'SEGLANEKEY', 'CROSSWALKKEY', "LOCATION",

In the end we are left with the following features: 'X', 'Y', 'STATUS', 'ADDRTYPE', 'COLLISIONTYPE', 'PERSONCOUNT', 'PEDCOUNT', 'PEDCYLCOUNT', 'VEHCOUNT', 'INCDTTM', 'JUNCTIONTYPE',

'SDOT_COLCODE', 'UNDERINFL', 'WEATHER', 'ROADCOND', 'LIGHTCOND', 'ST_COLCODE', 'HITPARKEDCAR'

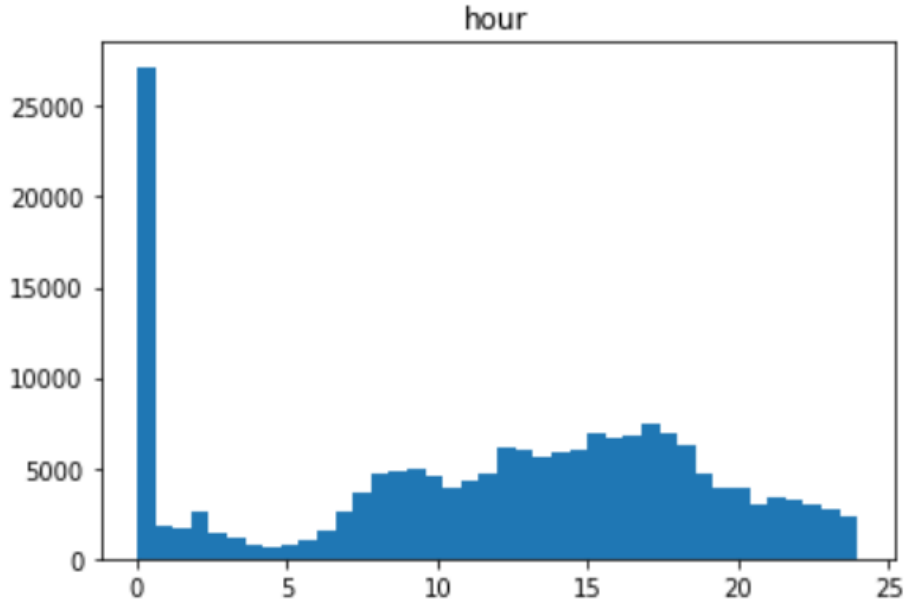
We have also removed rows with Nan losing 14,606 rows (~7.5%) which shouldn't affect our results much as this hasn't impacted the distribution between positive and negative results.

3. EXPLORATORY DATA ANALYSIS, DATA CLEANING AND FEATURE ENGINEERING

In order to kickstart our analysis we have first embarked on basic feature engineering.

Date and time of the accident was provided in a bulk form and hence we have created several separate time-related features, i.e. **hour, month, year** and **weekday** using Datetime capabilities from Pandas library. We then plotted distribution histograms of the numerical features to check for any anomalies. As seen from the graph below we have an unexpected spike for 0 hour. After further analysis it was deduced that 00:00 seems to be the default value that was inputted when granular time-related data was unavailable / missing. We have decided to drop these rows from the dataset as we thought that hour might be a very critical feature impacting the overall predictions and hence 00:00 values might be misleading. As a result, we have lost additional ~24k entries (or 12% of data).

Figure 1. Frequency distribution of road accidents per hour



We have further modified the time-related data to convey the notion of time cyclicality. For example, the model would tend to think that the difference between 00:01 and 23:59 is quite dramatic, however, that is not the case in real life. For that we broke down every cyclical feature into ***cos*** and ***sin*** components.

The next step was to convert categorical features into dummy / indicator variables in order to make the data more interpretable. The following categorical columns were identified and one-hot encoded: 'STATUS', 'ADDRTYPE', 'COLLISIONTYPE', 'JUNCTIONTYPE', 'UNDERINFL', 'WEATHER', 'ROADCOND', 'LIGHTCOND', 'HITPARKEDCAR', 'ST_COLCODE', 'SDOT_COLCODE'.

Our final processed dataset had 167 columns / features and 155,403 entries. We have also had a quick look into correlation matrix for numerical features and chi-square metric for categorical

features to potentially identify the features that are statistically insignificant and hence can be potentially removed.

Figure 2. Correlation matrix for numerical features

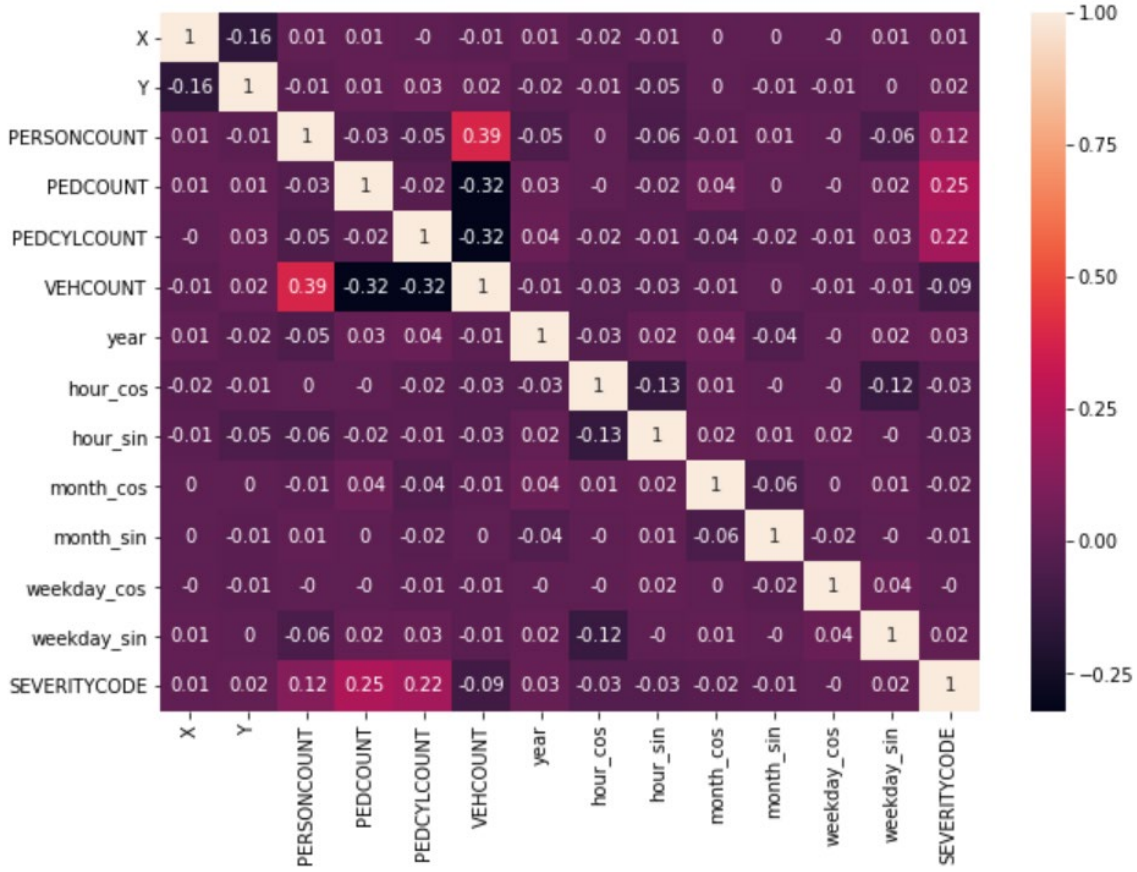
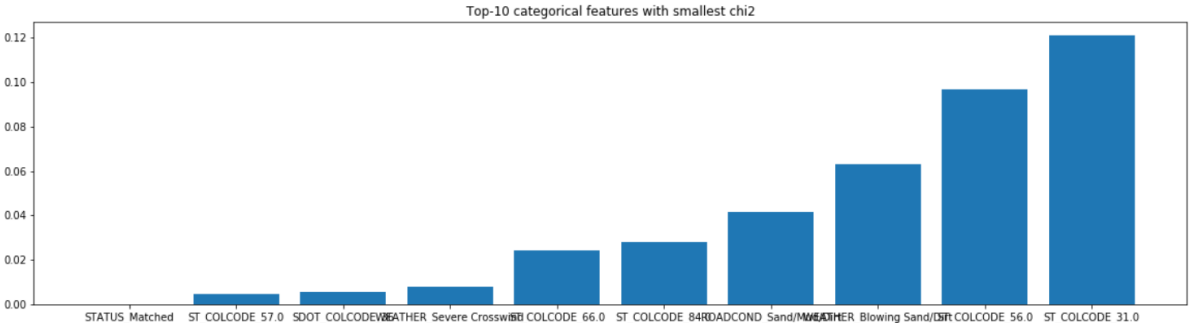


Figure 3. Top-10 categorical features with smallest chi-squared metric



It seems that our dataset can be further optimized by removing the statistically insignificant features though for now we have run the models on the full dataset.

Finally, we wanted to also present a two-dimensional histogram visualizing the frequency of severe and non-severe road accidents. It seems like there is a subtle difference of severe accidents occurring more frequently in city centres and other congested areas.

Figure 4. Distribution of SEVERE road accidents per coordinates

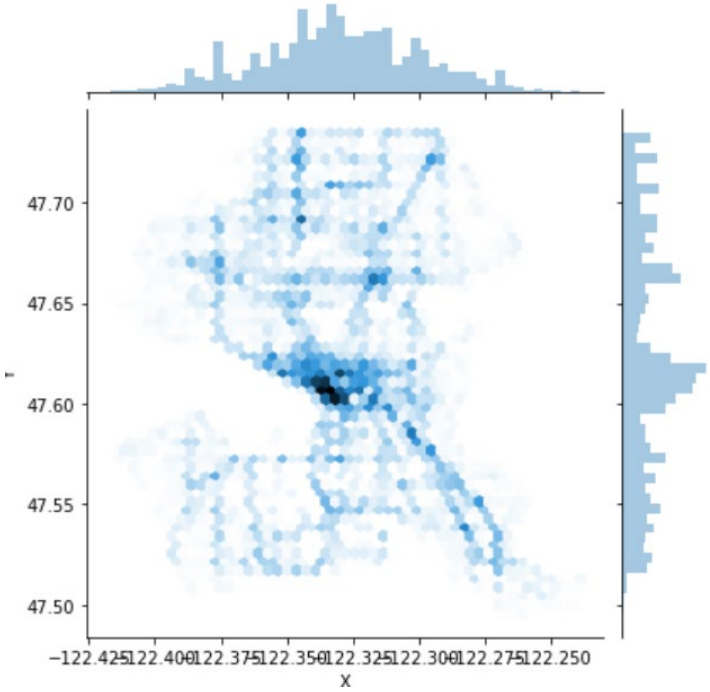
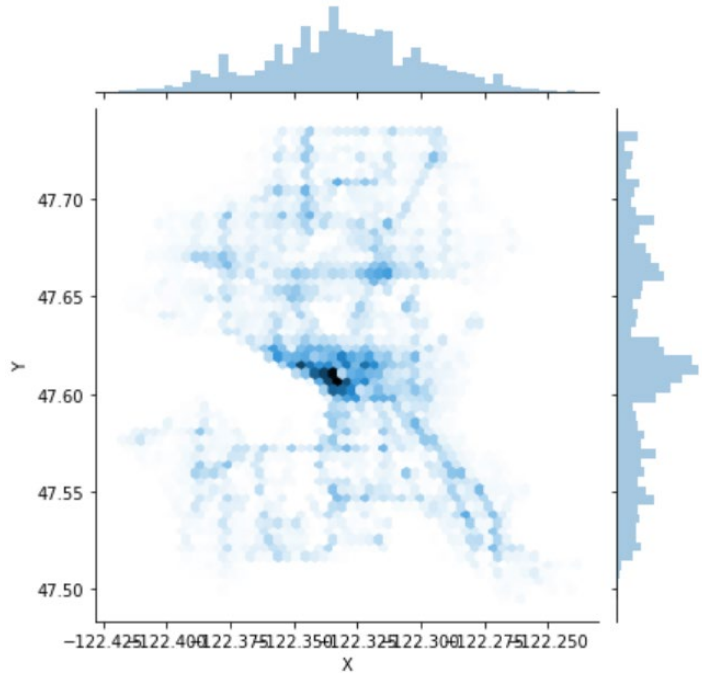


Figure 5. Distribution of NON-SEVERE road accidents per coordinates



4. METHODOLOGY

In order to build predictive models we have tried out a number of various machine learning models. In particular, we have tried out ***Multivariable Logistic regression, Random Forest*** and ***XGBoost models***. The summary of methodology is provided below:

1. Split the cleaned data into train and test sets

2. **Balance out the training set only** by random undersampling to have a 50:50 split for positive and negative results
3. Standardize train and test data
4. Create a subset of training data for Grid Search purposes (in order to shorten computing time)
5. Initialize the machine learning model
6. Run Randomized Grid Search on subset of the data (see step 3) to help tune the hyperparameters
7. Initialize and run the model with tuned hyperparameters from the previous step on full training data
8. Predict results for imbalanced test dataset and output results

As the test dataset is imbalanced the accuracy score would be quite meaningless on the test set so we have to define alternative metrics. Particularly, we have focused on **F1 score, Precision, Recall and AUC ROC score** that are more representative for imbalanced datasets.

We have also tried out the ***Deep Learning*** model with 1 hidden layer of 128 perceptrons with ReLu activation function and a dropout layer with rate of 20% to prevent overfitting.

5. RESULTS AND DISCUSSION

Below we can find a summary table of results for different machine learning algorithm:

	Multivariable Logistic Regression	**Random Forest**	**XGBoost**	**Deep Learning (1 hidden layer of 128 neurons)**
Accuracy	0.675	0.683	0.688	0.694
Balanced accuracy (average of recall)	0.709	0.713	0.716	0.711
F1 score	0.606	0.61	0.614	0.607
Recall	0.798	0.792	0.792	0.755
Precision	0.489	0.496	0.501	0.507
AUC ROC	0.791	0.797	0.799	0.795
Confusion matrix				
# of True Positive	7760	7700	7704	7347
# of True Negative	13230	13532	13675	14222
# of False Positive	1968	2028	2024	2381
# of False Negative	8123	7812	7678	7131

As seen from the table above, XGBoost model is marginally better than the other options though the difference is quite negligible. As mentioned before, we are dealing with the imbalanced dataset here and hence increasing F1 score, Recall / Precision and AUC ROC scores are what we should aim for. XGBoost maximizes these metrics and we have found the most optimal hyperparameters by running a randomized grid search. The further steps in regards to improving our results are described in the next section.

6. CONCLUSION AND FURTHER STEPS

The purpose of this report was to develop a model to predict severity of road accident based on a number of provided features. As described above we have tried several models and came to a conclusion that XGBoost algorithm works marginally better based on F1 score, balanced accuracy and precision / recall scores.

The results can be further improved by trying out a number of additional machine learning algorithms such SVM and K Nearest Neighbors and etc.

There is also a room for improvement in a feature engineering space. For example, one can try extracting address info from the coordinates (by leveraging geopandas) and creating several more features like district, street name and etc. This might help to localize the areas that have a large proportion of severe injuries.

We can also try dropping a number of features that proved to be statistically insignificant and hence irrelevant to our models to optimize / improve the model performance

Finally, one can make an attempt to leverage more data points. For instance, Seattle municipality has provided a larger set of similar data on the website which can drastically improve the predictive power of our model.