Predicting the Severity of Possible Traffic Accidents in Seattle City

Rong Chen

(Dated: October 12, 2020)

Based on the traffic accident data of Seattle city during 2004 and 2020, I trained several popular classification models including support vector machine (SVM), K-nearest neighbor (KNN), logistic regression (LR), decision tree (DT), random forest (RF) and extreme gradient boosting (XGBoost). These models are able to reasonably predict the severity of possible traffic accidents in Seattle city, therefore may be used to build real time traffic accident alarming systems for vehicles.

CONTENTS

I. INTRODUCTION

Predicting whether a traffic accident may happen and its severity if happened is a very important topic [1, 2]. For a drivers, if his/her car can be equipped with a real-time alarming system which can alert the possible severity of the accident if happened, it can no doubt greatly reduce the risk of a traffic accident, thus prevent the driver, the passengers and the pedestrians from getting injured and even save their lives. Besides, such a system can also be

incorporated into the navigation system, so that it can pick a route with the lowest risk to the destination.

Furthermore, more and more cars (especially those electric-powered cars) nowadays are equipped with some kind of 'auto-pilot' function based on AI, and in the future it is very possible that cars can automatically drive to the destination without a driver. For those cars, the ability to predict the possible severity or risk of possible traffic accidents in real time is of central importance. Because without such an ability, an auto-drive car is just like a moving coffin for the passengers and a moving killer for the pedestrians.

Therefore, no matter for traditional cars or auto-drive cars, we inevitably need to develop an efficient real time method to predict the severity of a possible traffic accident, in order to alarm the risk to the driver and the passengers.

In this report, I develop such a method to predict the severity of possible traffic accidents, based on classification (the label is 'severity') which is a type of supervised machine learning. Since this report is mainly for illustration purpose, I limit the scope within Seattle city. This is the report of the Capstone project — Car accident severity, for IBM Data Science Professional Certificate.

II. DATA

I use the example dataset provided in the week 1 of this capstone project, which is the data of the severity of traffic accidents occurred in Seattle city from 2004 January 1 to 2020 April 28. The raw data contains about 190000 records of traffic accidents. In each record, there are 38 columns. The meaning of each column can be found in the metadata also provided in the week 1 of the capstone project. The column "SEVERITYCODE" is the label which we will predict, it can be either 1 or 2, where code 1 means property damage and 2 means injury. The rest 37 columns can be taken as features in the machine learning and they are listed in Table I.

After deleting records including vague values such as 'NAN', 'N/A', 'unknown', 'nan', etc, we are left with about 160000 records. These data are what we will be using in this report. In Fig. 1, I randomly plot 2% locations for both severity 1 and 2 accidents on the Seattle map. The smaller green points indicate the location of severity 1 accidents and the bigger red points indicate severity 2.

TABLE I: Raw data

Label	Features (37)
SEVERITYCODE	X, Y, OBJECTID, INCKEY, COLDETKEY, REPORTNO, STATUS, ADDRTYPE, INTKEY, LOCATION, EXCEPTRSNCODE, EXCEPTRSNDESC, SEVERITYCODE.1, SEVERITYDESC, COLLISIONTYPE, PERSONCOUNT, PEDCOUNT, PEDCYLCOUNT, VEHCOUNT, INCDATE, INCDTTM, JUNCTIONTYPE, SDOT_COLCODE, SDOT_COLDESC, INATTENTIONIND, UNDERINFL, WEATHER, ROADCOND, LIGHTCOND, PEDROWNOTGRNT, SDOTCOLNUM, SPEEDING, ST_COLCODE, ST_COLDESC, SEGLANEKEY, CROSSWALKKEY, HITPARKEDCAR

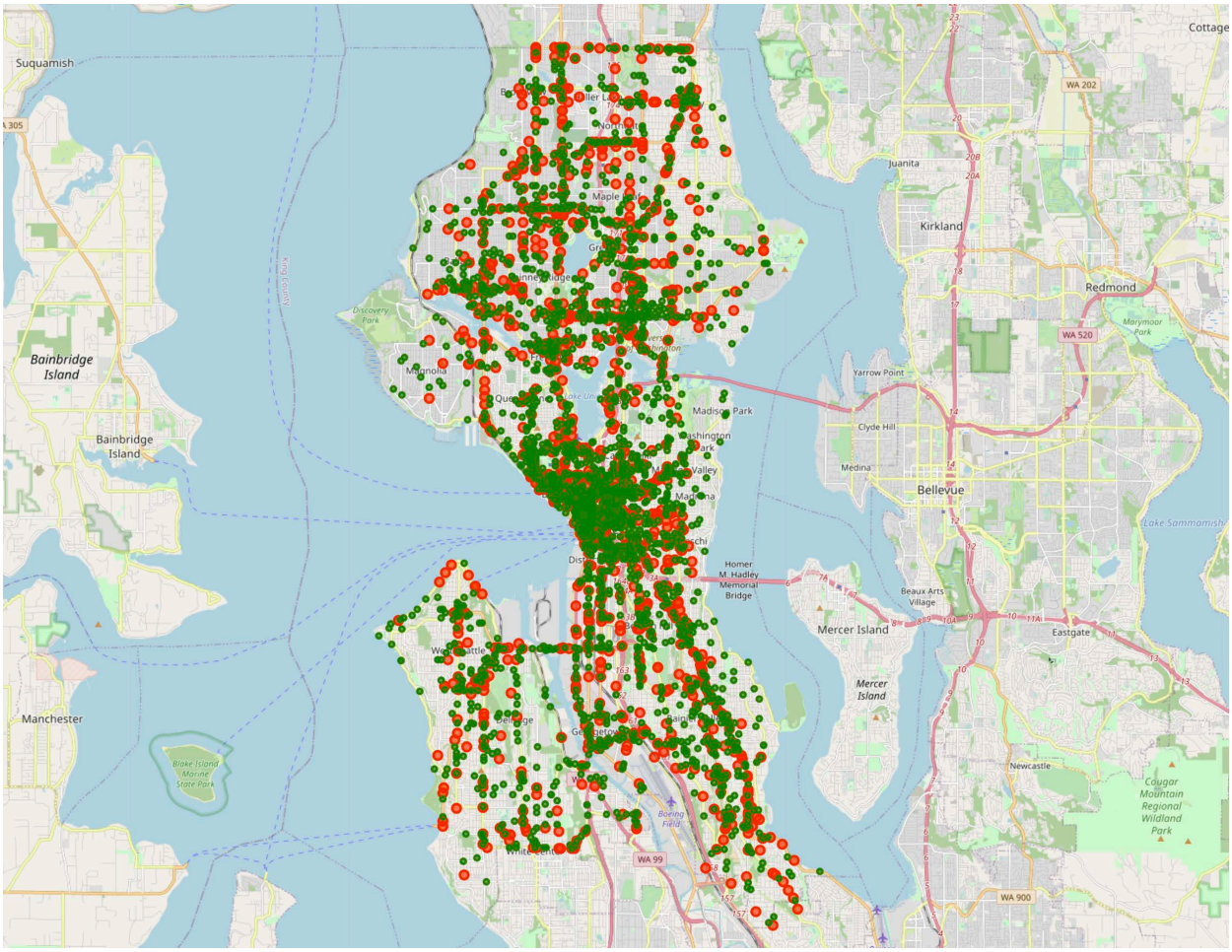


FIG. 1: Locations of severity 1 and 2 accidents in Seattle.

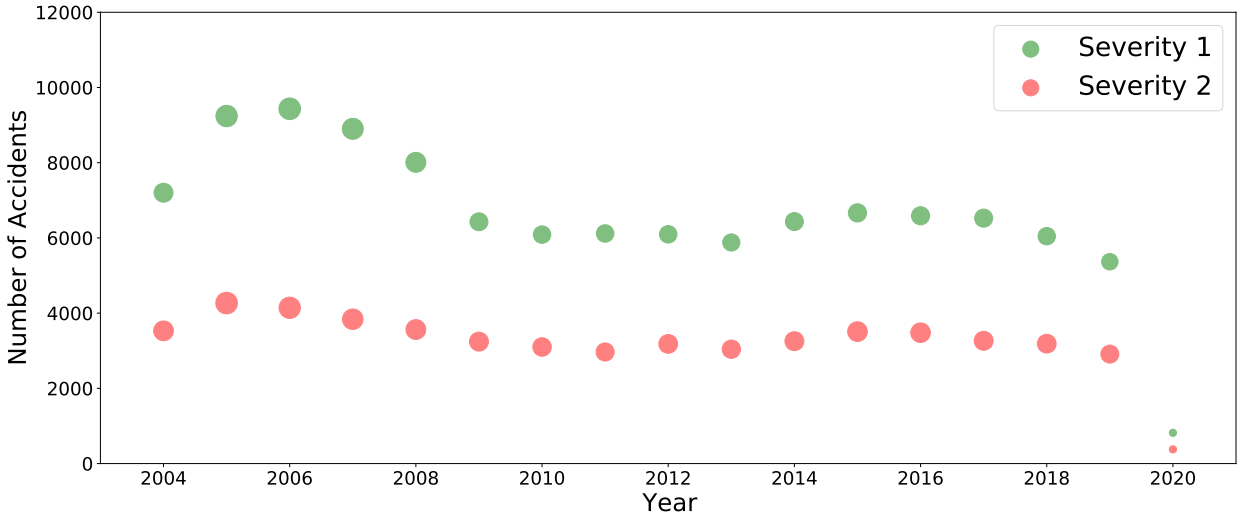


FIG. 2: Number of traffic accidents in Seattle from 2004 January to 2020 April.

There are some features need to be mentioned. "X" and "Y" represents the longitude and the latitude of the accident location, "ADDRTYPE" represents whether the accident location is intersection or block, "INCDATE" and "INCDTTM" represent the date and time of the accident, "INATTENTIONIND" indicates whether the driver was focusing on driving or not, "UNDERINFL" represents whether the driver was under influence of alcohol or not, "WEATHER", "ROADCOND", and "LIGHTCOND" represent the weather, road and light condition, "SPEEDING" represents whether the car is speeding or not.

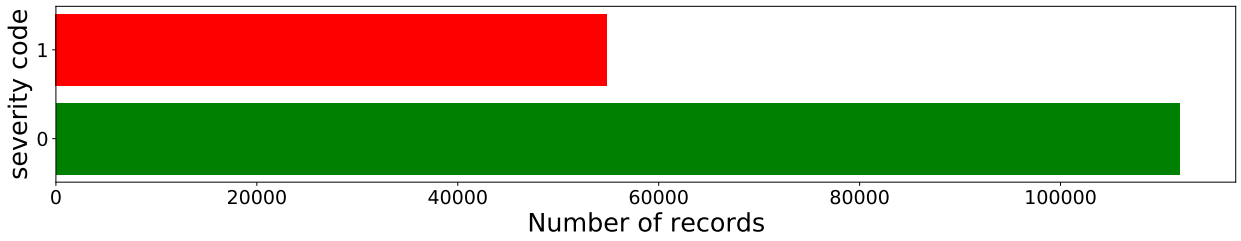


FIG. 3: Imbalance of severity code label.

However, as shown in Fig. 3, we see that among all the records, 2/3 are severity 1 and only 1/3 are severity 2. So the data are not balanced in terms of severity label. So when train the model, as described in the next section, we need to balance the data in the training set, otherwise the models we come up with will be biased and therefore useless.

III. METHODOLOGY

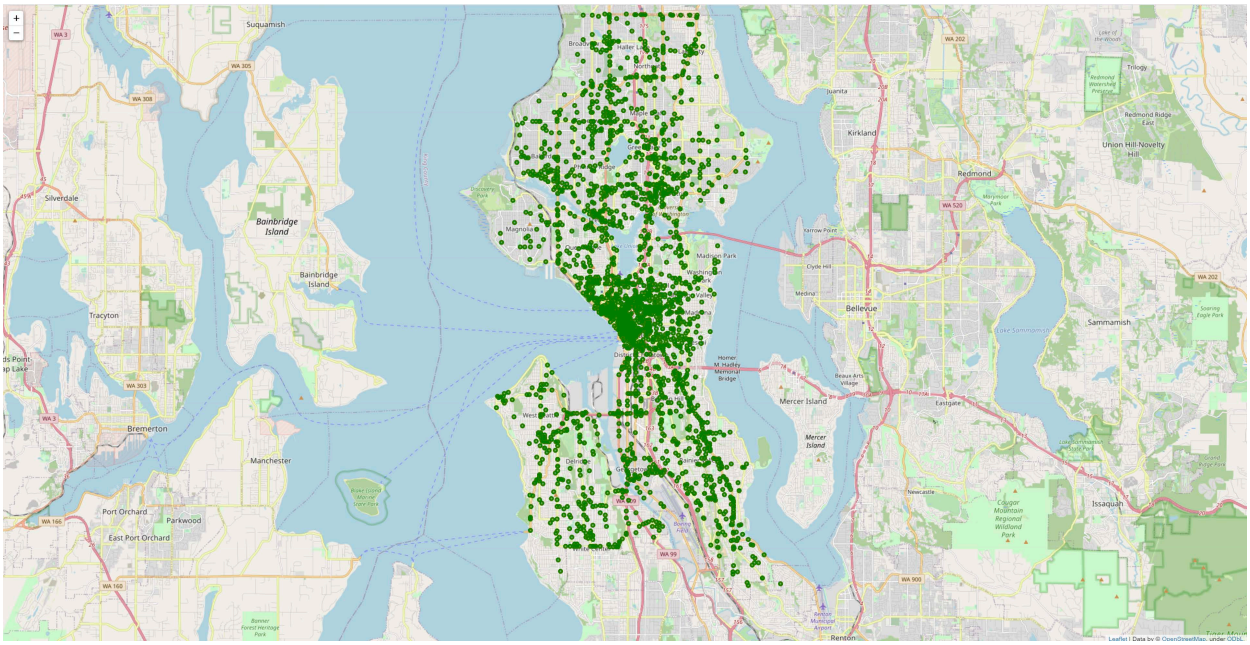
In machine learning, we need to pick features from the data, then use the training set to train the coefficients coupled with the features for particular models, finally we use the trained model to predict the labels in the test set.

So the first step is to pick relevant features for the machine learning. In Fig. 1 and Fig 4 we do find some patterns of the locations which means location must be included as a one of the features in the machine learning. However, we find that the severity 1 and 2 locations have some overlaps and are somewhat similar, therefore location alone is not enough to separate severity 1 and 2. We need more features. Although the severity of a traffic accident for sure related with factors such as how many people involved including the pedestrians, which direction the car is hit from, etc, we can not use those as features. Because those information cannot be known before the accident. We can only use the information which can be collected before the accident occurs, such as the location of the car, the date and time, the weather condition, the road condition, the light condition, etc. After deleting features which cannot be obtained before the accident, we are left with features as shown in Table II.

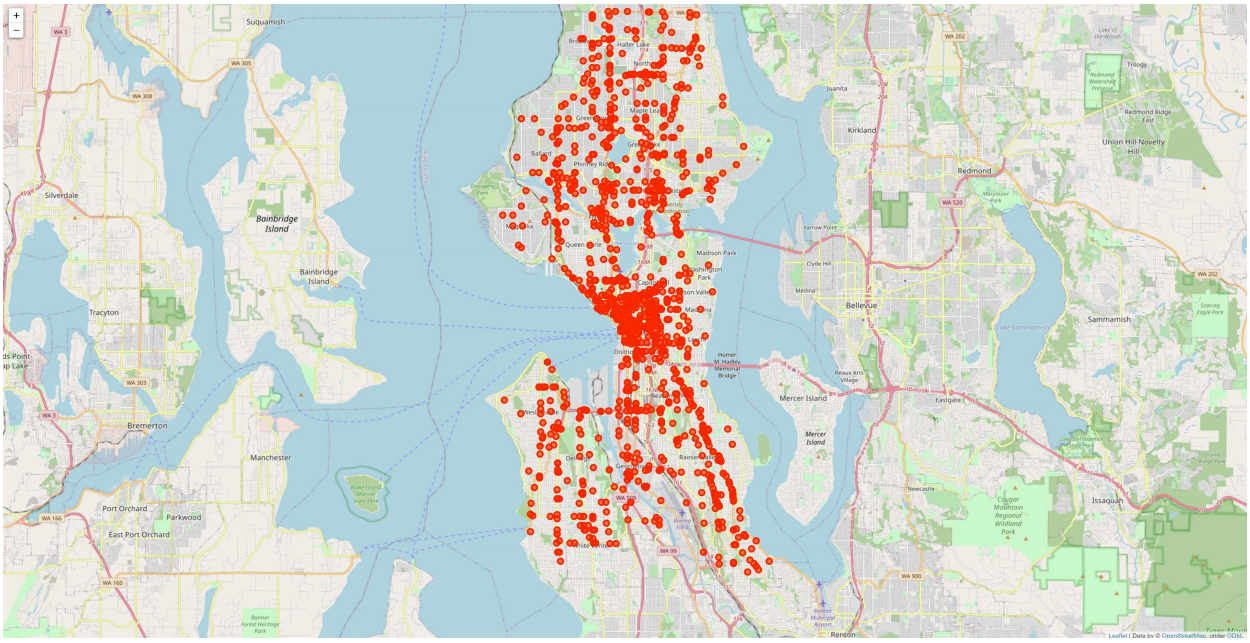
TABLE II: Relevant features for prediction

Label	Features (11)
SEVERITYCODE	X, Y, ADDRTYPE, INCDATE, INCDTTM, INATTENTIONIND, UNDERINFL WEATHER, ROADCOND, LIGHTCOND, SPEEDING

In Fig 5, we plot average number of severity 1 and 2 accident happened during each hour of a day. The number severity 2 accidents are almost always half the number of severity 1 accidents. On the hour of the day plot [3], from 1 AM to 4AM the number of accidents significantly decreased, obviously it is the result that most people are sleeping during that period of time, this period is the 'sleeping hours' of Seattle city. From 5AM to 8AM, the number of accidents increases, this can be interpret as people are going to work and so the traffic are heavier than before which means more accidents can occur. From 10AM to 5PM, the number of accidents continues to increase but with a smaller slope compared with the 5-9AM rush hour period. This can be interpret as, during that time most people are already at work. Many works are more or less related with driving to some extent. Even if people



(a) Locations of severity 1 accidents in Seattle.



(b) Locations of severity 2 accidents in Seattle.

FIG. 4: Locations of severity 1 and 2 accidents in Seattle.

work at office they or their company still need other people to deliver some stuff for them. So overall the demand of all kinds traffic is high, so the there can be more number of accidents than rest of day. Even so, since most people obey the traffic rules and it is day time so at least the light condition can be much better than in the night, so over there are roughly

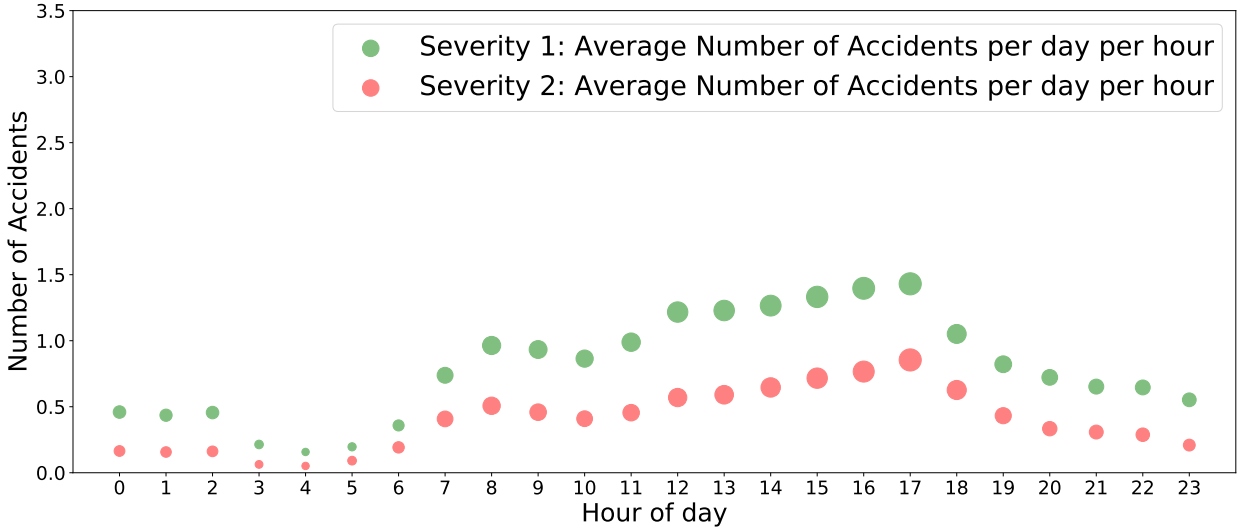


FIG. 5: Average number of accidents for severity 1 and 2 during each hour of a day.

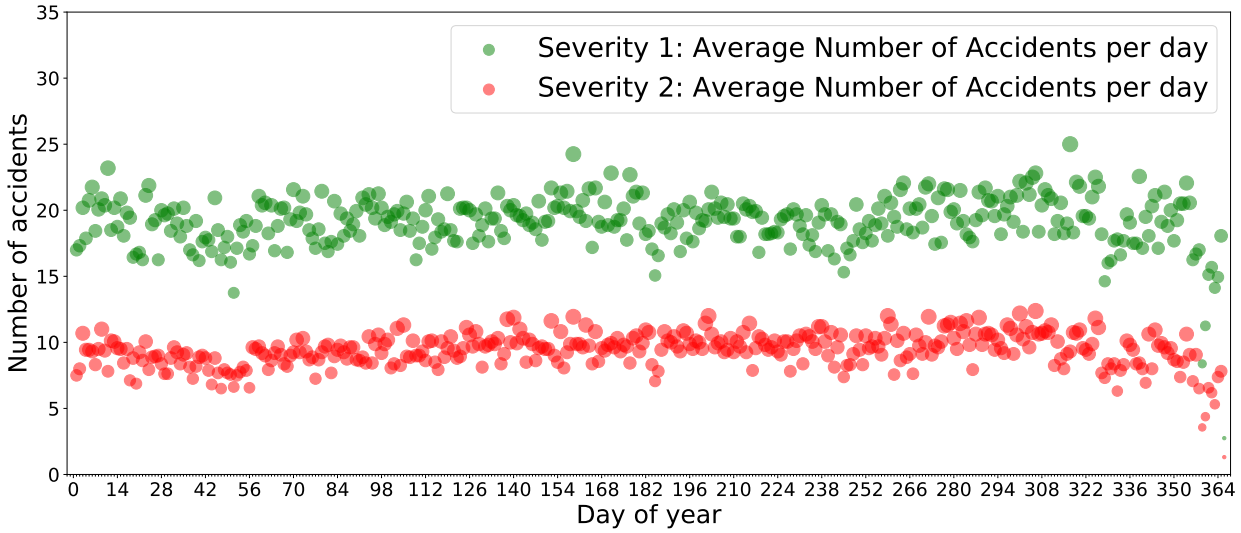


FIG. 6: Average number of accidents for severity 1 and 2 during day of year

like 1.5 – 2 accidents in the Seattle city per hour during that period of time. From 5PM to 11PM, the number of accidents decease because most people have already reached home so the traffic volume decreases. During midnight from 11PM to 2AM, the number of accidents remain the same level and is low.

We can see that from Fig. 6 and Fig. 7, the accident number decreases as the time towards the end of year which means holidays and less traffic. Fig. 8 show that Friday is the day when the accident number is higher than other days of the week, while Sunday has the lowest accident number on average. From Monday to Friday, the number of accidents slowly increases,

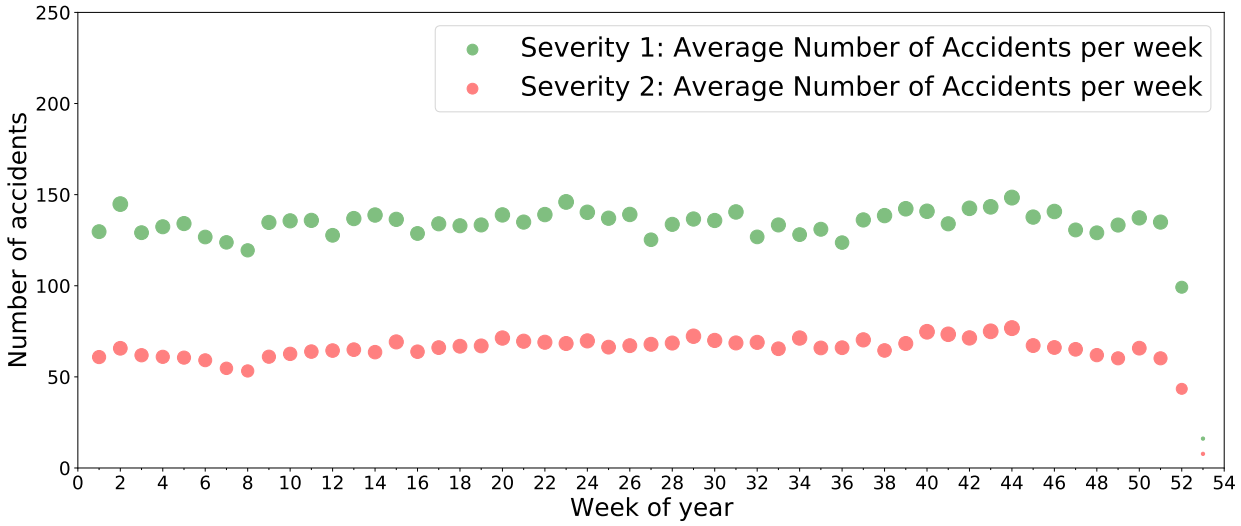


FIG. 7: Average number of accidents for severity 1 and 2 during week of year.

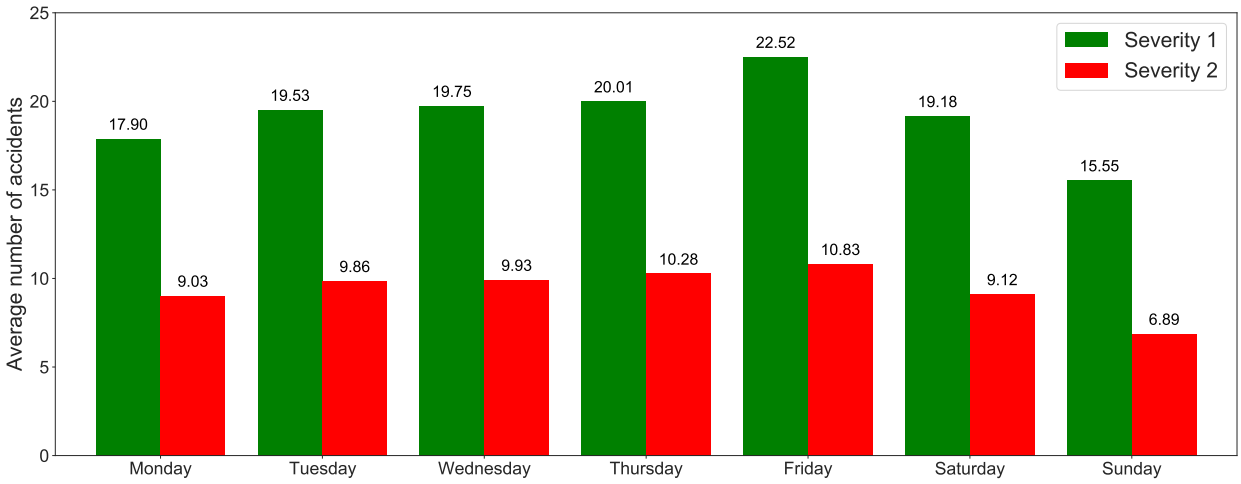


FIG. 8: Average number of accidents on weekdays.

while From Friday to Sunday this number decreases. Fig. 9 shows the average number of accidents during each month. We find that February has the lowest number of accidents, probably because the temperature is low and the traffic become less intense. October, on the other hand, has the highest number of accidents.

For features 'WEATHER', 'ROADCOND', 'LIGHTCOND' and 'ADDRTYPE', we use one hot method to convert each of their string values as a feature which can be either 0 or 1. We see that the most accidents occurs when the road condition is dry, weather condition is clear, light condition is daylight. It does not necessary means that daytime, clear weather and dry road can cause accidents, it can simply means that there are other factors that can make

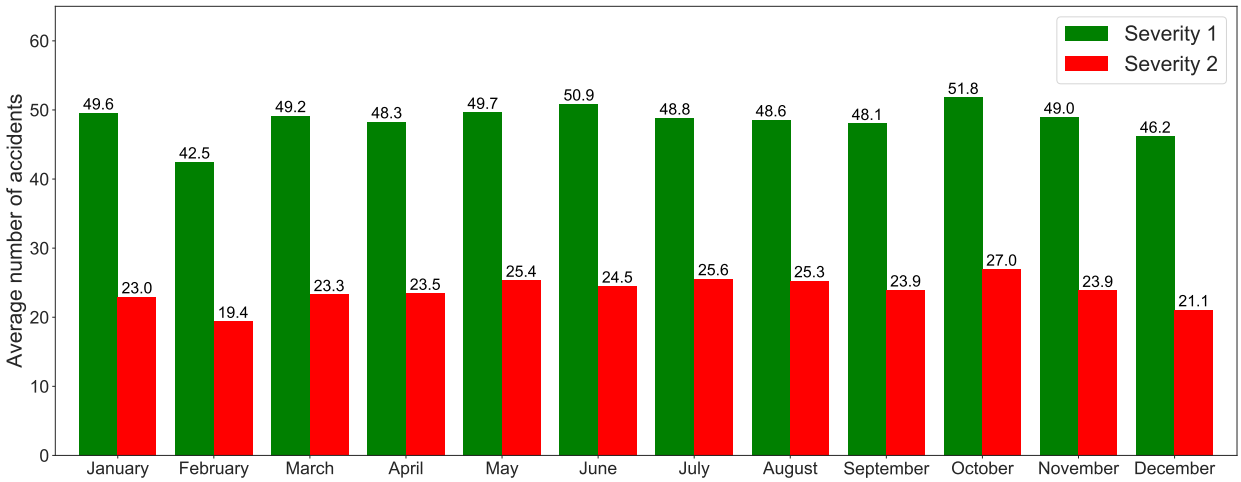


FIG. 9: Average number of accidents in each month.

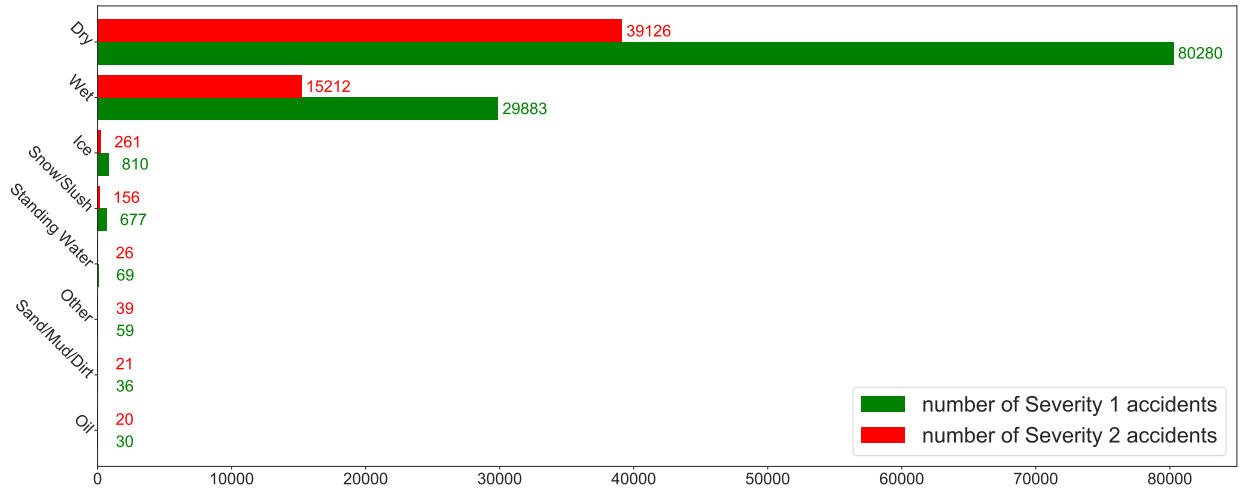


FIG. 10: Number of accident under different road conditions.

accidents occur. We also find under wet road condition, raining or overcast weather, dark with street on conditions, there are also many accidents occur. But there are always twice number of severity 1 accidents than severity 2, ratio between severity 1 and 2 is 2:1. That means severity of an accident does not very much depend on these road, weather, and light conditions. There must be other factors which can determine the severity of an accident. We noticed that, when the road condition is ice, snow/slush, weather condition is snowing, light condition is no street lights, the number of severity 2 accidents is three times or more the number of severity 1, the ratio is around or above 3:1. This seem to indicate that under these conditions perhaps people are more cautious and so severity 2 accidents occur less likely than

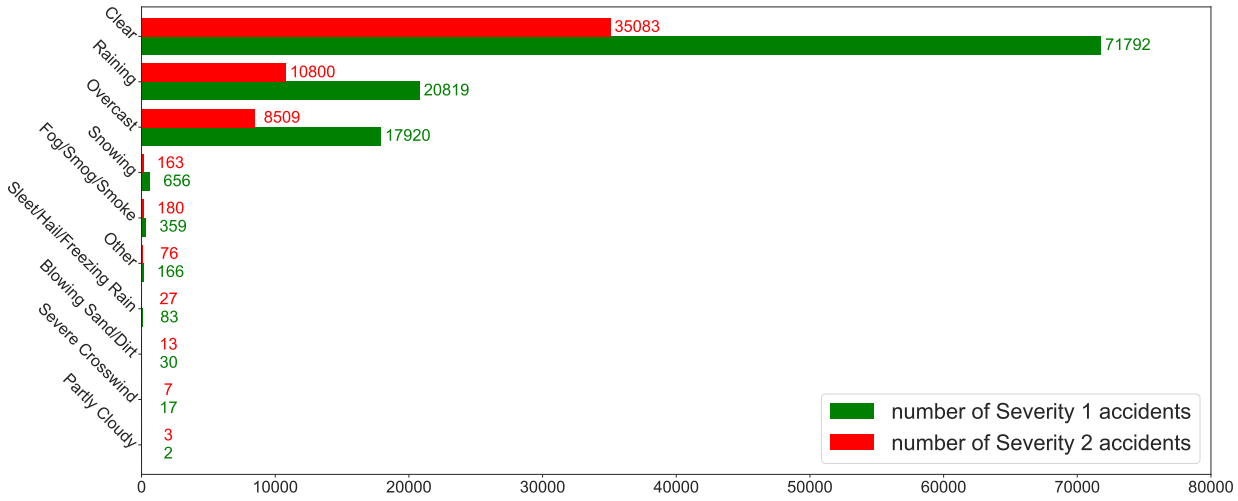


FIG. 11: Number of accident under different weather conditions.

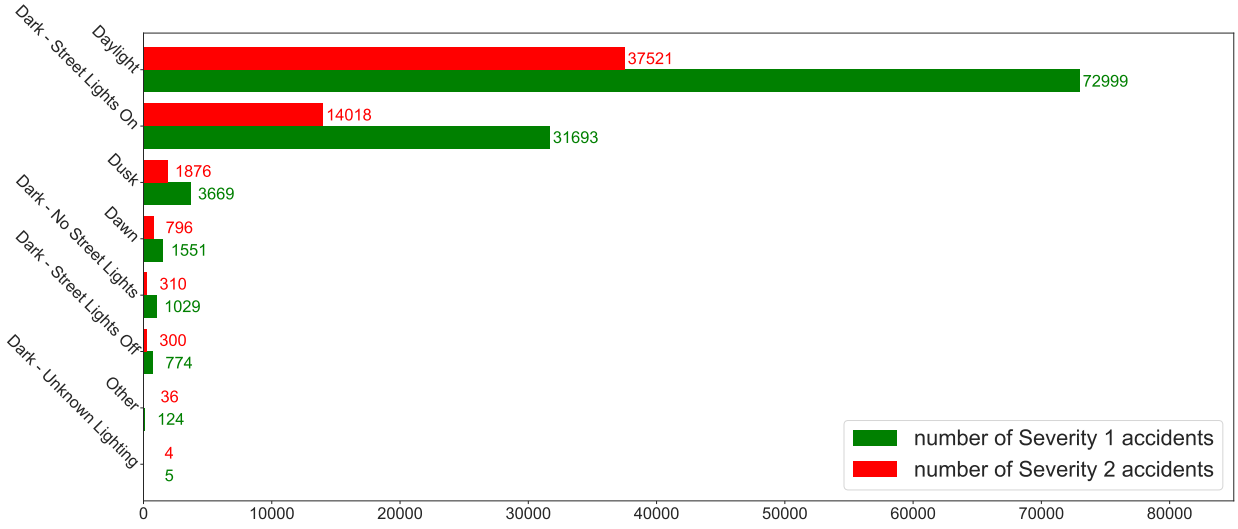


FIG. 12: Number of accident under different light conditions.

severity 1 ones. Or it can be interpreted as under these conditions, accidents are more likely to occur, but most of them are severity 1 than severity 2. So people still need to be cautious.

In Fig. 13, we plot the number of severity 1 and 2 accidents under different address type, and also under conditions such as whether speeding or not, paying attention or not, and whether under influence of alcohol or not. In Fig. 13, 0 means no and 1 means yes. We find that for address type, when at intersections, the ratio between severity 1 and 2 become almost 1:1 instead of the 2:1. This means at intersection the risk of severity 2 accidents significantly increases, so both the driver, the passengers, and the pedestrians need to pay extra attentions.

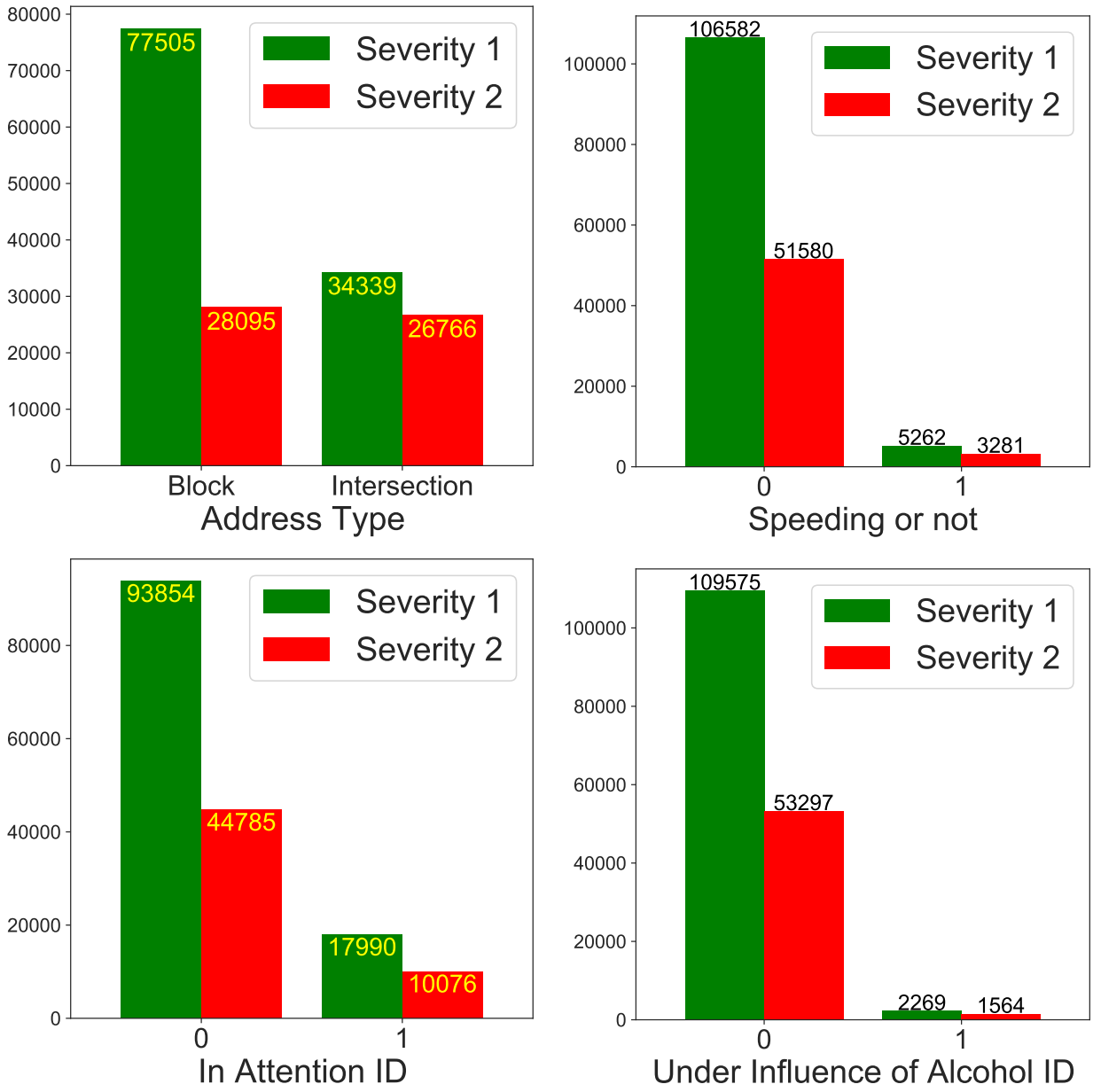


FIG. 13: The number of severity 1 and 2 accidents under different address type, and conditions such as whether speeding or not, paying attention or not, and whether under influence of alcohol or not.

Also, under influence of alcohol, the ratio between severity 1 and 2 are close to 1.5:1 instead of 2:1, which means the severity 2 accidents are more likely to happen than severity 1.

Table III list the useful features after cosine and sine conversion for time and location and the one hot process. Those are the 39 features we will be using in the models.

The second step is that, before training the data using different models, we need to do

TABLE III: Useful features after feature engineering

Label	Features (39)
SEVERITYCODE	INATTENTIONIND, UNDERINFL, SPEEDING, ADDRTYPE_Block, ADDRTYPE_Intersection, WEATHER_Blowing Sand/Dirt WEATHER_Clear, WEATHER_Fog/Smog/Smoke WEATHER_Other, WEATHER_Overcast, WEATHER_Partly Cloudy, WEATHER_Raining, WEATHER_Severe Crosswi WEATHER_Sleet/Hail/Fre WEATHER_Snowing, ROADCOND_Dry, ROADCOND_Ice, ROADCOND_Oil, ROADCOND_Other, ROADCOND_Sand/Mud/Dirt ROADCOND_Snow/Slush, ROADCOND_Standing Wate ROADCOND_Wet, LIGHTCOND_Dark - No St LIGHTCOND_Dark - Stree LIGHTCOND_Dark - Stree LIGHTCOND_Dark - Unkno LIGHTCOND_Dawn, LIGHTCOND_Daylight, LIGHTCOND_Dusk, LIGHTCOND_Other, dayofyear_cos, dayofyear_sin, minuteofday_cos, minuteofday_sin, Longitude_cos, Longitude_sin, Latitude_cos, Latitude_sin

some feature engineering on the data. For the longitude X and latitude Y, since the surface of the earth is a mostly a sphere, the polar coordinates which can describe the locations have a period of 2π, so instead of using X and Y directly, we use the value of the cosine and sine of X and Y [1] which is more reasonable. Since we do not have altitude data for the accidents' locations, we neglect this feature. Similarly, since the day of the year, the hour of the day, and the minute of hour also have periods such as 365, 24, and 60, we also use their cosine and sine instead. After all, time such as 23:59PM and 0:01AM are almost the same, there is really not a 23 and 0 difference.

Then we do feature scaling on all the values of the features so that they are distributed around their average values with variance 1. In Fig. 14 we show the correlation matrix between all the columns in Table III. It looks like the severity is more correlated with "ADDRTYPE" (interaction or block) than other features. Also, as expected, some features, eg, dry road condition and clear weather are highly correlated, and dry road and raining weather are highly negatively correlated.

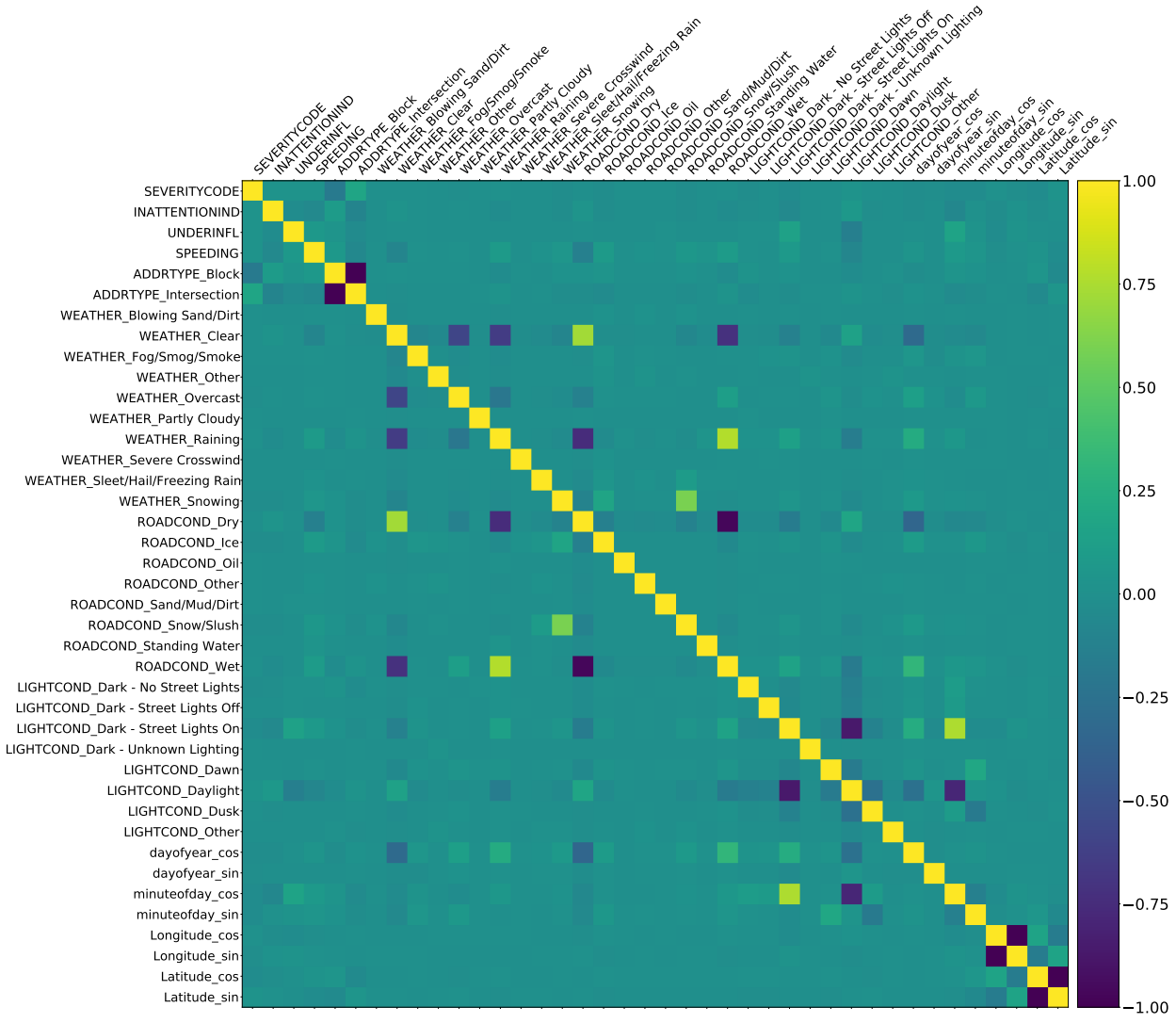


FIG. 14: Correlation matrix

The third step is to split the data into training set and test set and then balance the training set. As usual, we use 80% of the data as training set and 20% as testing set [4]. However, as is shown in Fig. 3, we see that among all the records, 2/3 are severity 1 and only 1/3 are severity 2. So the data are not balanced in terms of severity label. To proceed, we do up sampling (we also performed down sampling but it does not perform better than up sampling, so we keep using up sampling) for the severity 1 records in the training set, such that now there are the same number of severity 1 and 2 data in the training set. Note that we keep the test data unchanged, there is no need to balance the data in the test set.

With the training data ready, we will use several popular supervised machine learning algorithms for classification to train our data. We use scikit-learn python library, and the

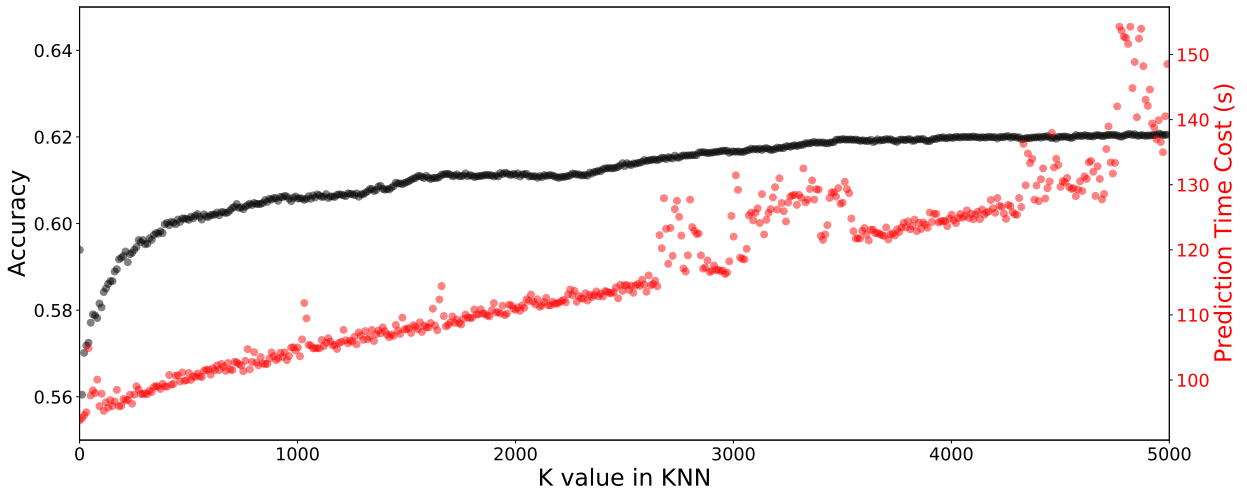


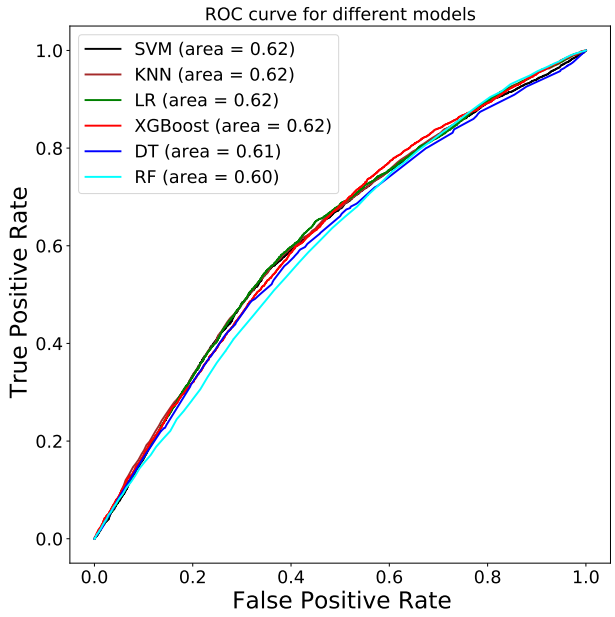
FIG. 15: Accuracy and the corresponding prediction time for the testing test .vs. K in KNN.

algorithms are support vector machine (SVM), K-nearest neighbor (KNN), logistic regression (LR), decision tree (DT), random forest (RF) and extreme gradient boosting (XGBoost) to train our model. Among them, RF and XGBoost are based on DT, they are typically less sensitive to imbalanced data, but they may over fit the data a little bit. KNN is a type of so called lazy learning, it does not really training any model, every time we have a new data point we need to find its nearest neighbors in the training set and then make prediction. About KNN model, we plot test accuracy .vs. K number in Fig. 15. We find that $K = 500$ is likely the elbow point for accuracy which can be taken as the optimal choice of K for KNN. So I simply use $K = 500$ in this report. LR is a widely used algorithm, it is fast and can give the weight of each features. SVM is powerful algorithm and involves many matrices operations to minimize the cost functions in order to achieve the optimal model for making predictions.

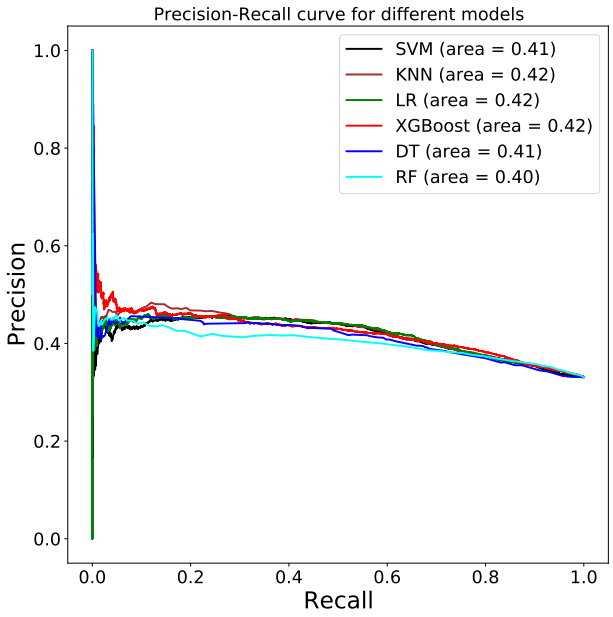
IV. RESULTS

In this section, I report the results of the selected machine learning models. The results reported in this section will be based on using 50% of the data and then do a 80%/20% training/testing set splitting. The reason to choose to use 50% of the data will be explained in the next section.

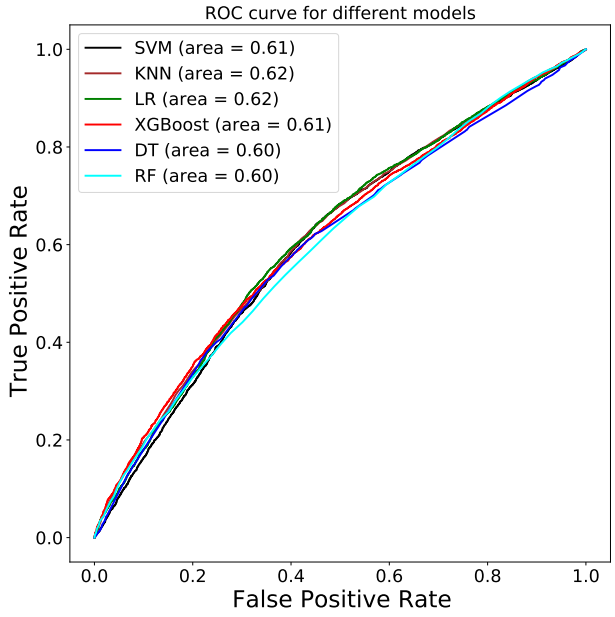
Based on record with the features listed in Table III, we need to predict whether it can be a severity 1 accident or a severity 2 accident, so this is a 2-class problem. Therefore, in Fig.



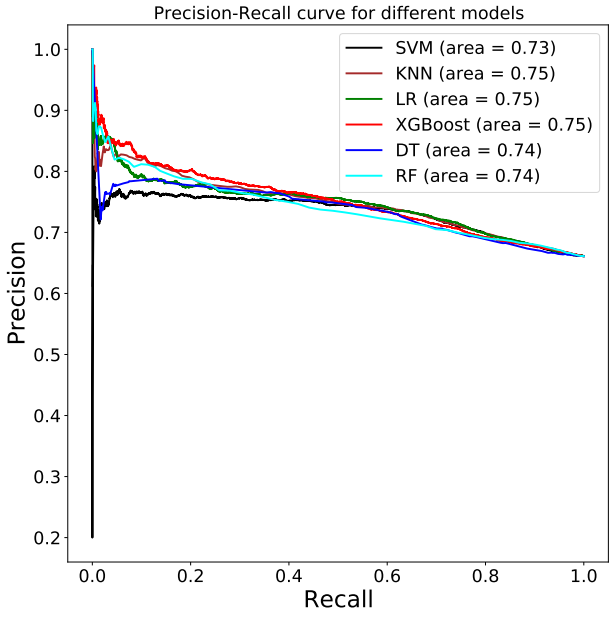
(a) Severity 2 as positive and 1 as negative.



(b) Severity 2 as positive and 1 as negative.



(c) Severity 1 as positive and 2 as negative.



(d) Severity 1 as positive and 2 as negative.

FIG. 16: ROC curves and Precision-Recall curves for different models.

16, I plot the receiver operator curve (ROC) and Precision-Recall curves [5] to measure the performance for different models under difference thresholds for classification. For this 2-class problem there can be two choices. For choice 1, we can choose severity 1 as positive examples and severity 2 as negative examples. For choice 2 we can choose severity 2 as positive and

severity 1 as negative. Fig. 16 includes the two choices. We find that no matter what choice we made, all the models give similar for ROC, all of them slightly bow towards (0,1) point and their area under curve (AUC) are above 0.6 which is good sign that the models have some skills. However, RF and DT performs relatively worse than other models given their AUC are among the lowest. Since our testing data is imbalanced, ROC may not be sufficient, because in this case a model simply bias towards the majority label (eg. towards predicating more negative samples and therefore generate a lot of fake true-negative samples) class with no skills may just seem to work reasonably. Therefore, for imbalanced classes, precision-recall curve can be useful. A good precision-recall curve should bow towards (1,1) point, and similar with ROC, the AUC is the bigger the better. For both choice 1 and choice 2, the curves are better than a merely a horizonal line which means no skills at all. The AUC for choice 2 are all around 0.74 which is higher than that for choice 1. This can be understand because in choice 2, majority class is severity 1 which means positive while choice 2 is the opposite, the models are better at predicting majority class than minority class for imbalanced data. In principle, we may choose the best threshold for each model. But since the ROC and precision-recall curve are very similar for each model, and none of the models are very close to the (0,1) point in ROC and (1,1) in precision/recall curve, therefore changing the threshold will not improve the F-1 score by very much. So we simply use the default threshold for each model, eg, in LR the default threshold is 0.5. Note that, since we did not use all the features in the raw data, because we can only pick features which can be obtained before the accident, it is reasonable that we do not expect our model performs extremely good in predicting severity 1 and 2.

TABLE IV: The accuracy of the built model using different evaluation metrics

Model	Training set			Test set		
	Jaccard index	F1-score	Accuracy	Jaccard index	F1-score	Accuracy
SVM	0.32	0.62	0.61	0.32	0.62	0.61
KNN	0.42	0.60	0.60	0.32	0.62	0.61
LR	0.41	0.60	0.60	0.32	0.62	0.61
XGBoost	0.46	0.63	0.63	0.32	0.62	0.61
DT	0.46	0.63	0.63	0.32	0.61	0.59
RF	1.00	1.00	1.00	0.19	0.62	0.65

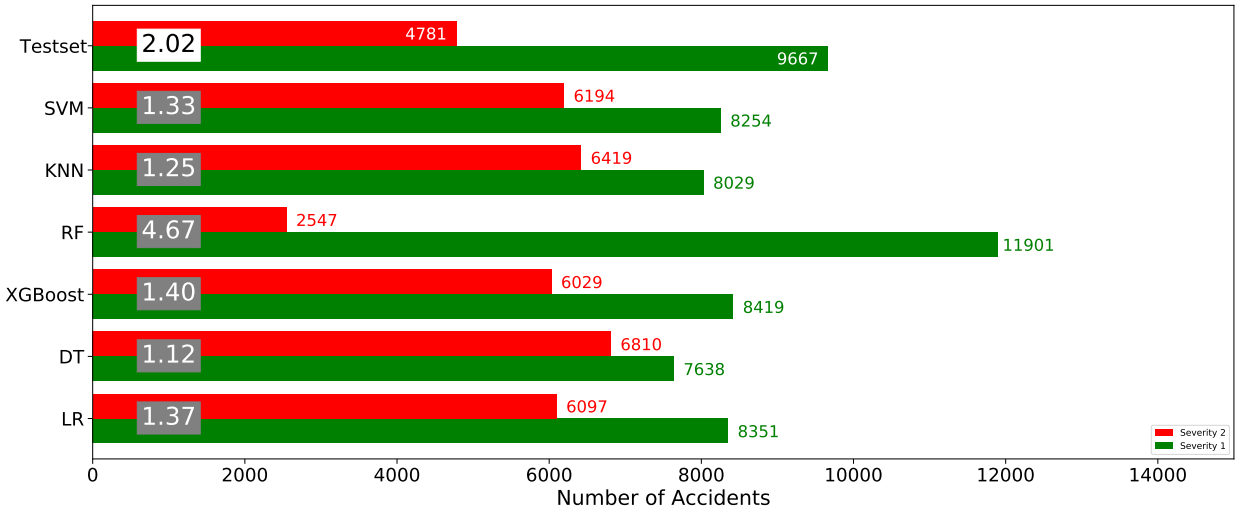


FIG. 17: Model predicted number of severity 1 and 2 accidents and the theoretical number of accidents in the testset.

In Table IV, we report the Jaccard index, F1-score, and accuracy of the models we used. we find that in terms of prediction, SVM, KNN, LR, XGBoost performs similarly with accuracy 0.61. DT performs little worse than them with accuracy 0.59. While RF shows the highest accuracy 0.65, its Jaccard index is pretty low with 0.19, which means RF may simply biased towards the majority class. Also, since RF performs perfectly in the training test but not so in the testing set, it indicates that RF has over fitting issue in this problem. In Fig 17, we plot the number of severity 1 and 2 accidents predicted by each model for the testing set. For the testing set, as the white label indicates, the theoretical ratio between the number of severity 1 accidents and severity 2 accidents is 2.02. We find that XGBoost, LR, and SVM are above 1.30, KNN is slightly worse than them. However, DT and RF performs worst. For DT, the ratio is 1.12 which means DT seem to predict the same number of severity 1 and 2 accidents. For RF, it predicting way much more severity 1 accidents than severity 2 ones, and the ratio becomes 4.67. So obviously RF is clearly the worst among the models we used.

V. DISCUSSION

In fact, when we are training the data, we find that it is not the more the data the better. In this problem, we choose 10%, 20%, 30%, ..., 100% of the data and then do a 80%/20% training/testing set splitting, we find that the test accuracy of each model are very similar,

they are all slightly above 0.6 (ideal value is 1).

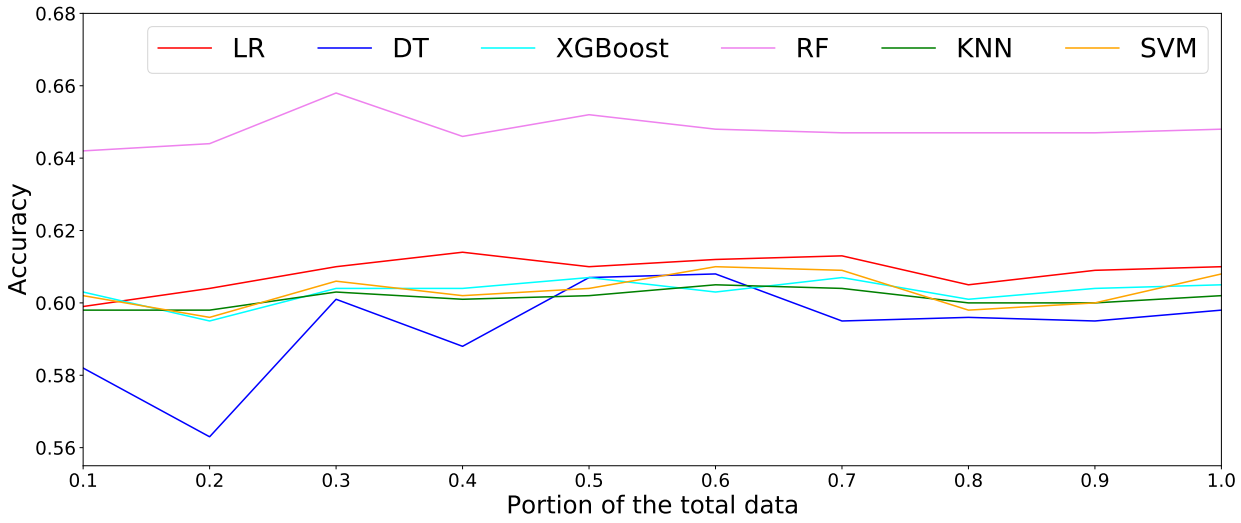


FIG. 18: Accuracy

In Fig. 18 we show the testing accuracy .vs. the amount of training data we used. Clearly, none of the model show a clear rising tendency of accuracy along with the increasing amount of the data. Besides, as pointed out in the previous section, the higher accuracy of RF is simply due to over fitting. For DT, its accuracy is always the worst. Overall, using 100% of the data does not really have advantage over say, using 50% of the data or even 20%, especially considering the computational cost for training SVM and testing KNN are much higher than other models. In terms of the accuracy, we can conclude RF and DT are the worst among the models.

In Fig. 19 we show the time cost for training and testing a fixed amount of data of all the models. Among the algorithms used, LR is the fastest algorithm and its training computational cost is about $\mathcal{O}(m)$ given m is the number of training points. RF and XGBoost are based on DT, their training computational cost are somewhere around $\mathcal{O}(m \log m)$, so they are still fast. For SVM, I use 'rbf' kernel which I believe it uses gaussian function as the similarity function (for SVM using linear kernel which means no kernel is used, the similarity functions are simply the data points themselves, so it is similar with LR). However, since SVM requires m by m matrices multiplications, its training computational cost is somewhere between $\mathcal{O}(m^2)$ and $\mathcal{O}(m^3)$ therefore it is expensive. For KNN, it does not really need to training any data, the training stage is basically simply read all the training data into memory, but the testing stage is expensive. If we have n testing data, for each of them, we need to

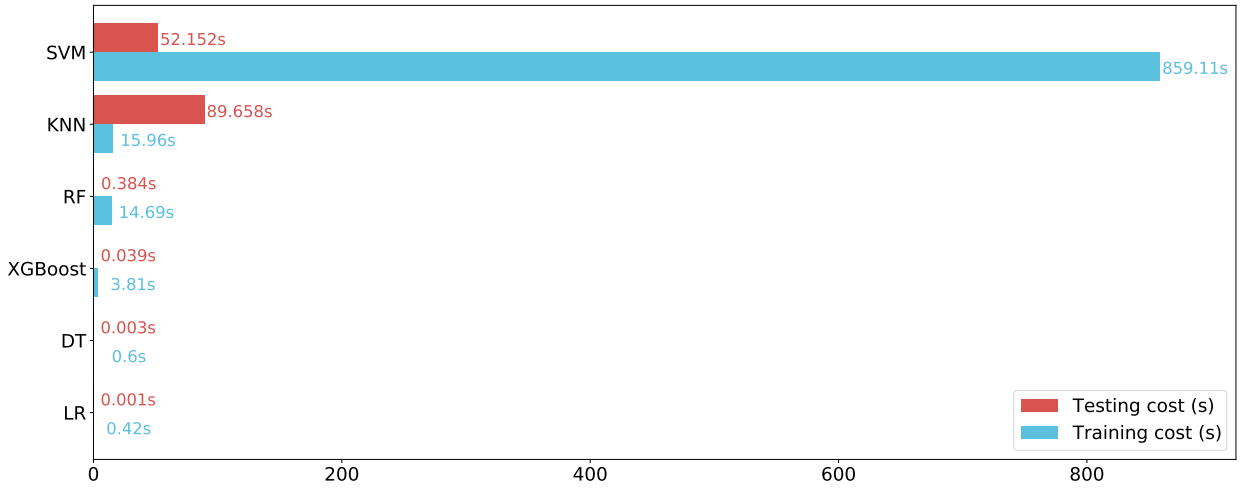


FIG. 19: Training and test time cost for different models based on about 80000 examples (50% of the total data) including 64000 training examples and 16000 testing examples.

calculate its distance between each of m training data, these m distances need to be stored in the memory, and then we choose the most proper K neighbors. So the total testing cost for KNN is $\mathcal{O}(m \times n)$ which is very high. From Fig. 15, we see that actually the cost (the red dots) also linearly increases with the K number of neighbors in KNN. For other algorithms the total testing cost are all around $\mathcal{O}(m)$ because they really trained models therefore no need for the testing data to scan the whole training set again.

It needs to be pointed out that, the time cost is based on about 80000 records of data, 64000 for training and 16000 for prediction and testing. Particularly, the testing part is what we care most after training. The testing time cost divided by 16000 is the time cost for one record, so actually all the model can make prediction within milliseconds. However, all the calculations are performed on my laptop, therefore, if we were to run the models on the mobile device with much less computation power, it may take longer for predicting each record.

Among all the model, LR uses the linear combination of each feature, then put it into sigmoid function to minimize the cost function. So it can give us the coefficient for each feature. In Fig. 20 we represent the absolute value of the coefficients for each feature as a bar plot. We find that the address type (whether it is block or intersection) is most important feature. As has discussed for Fig. 13, at intersections people need to be aware of the possibly of severity 2 accidents. Also, whether the driver is in attention, and is speeding or not, under the

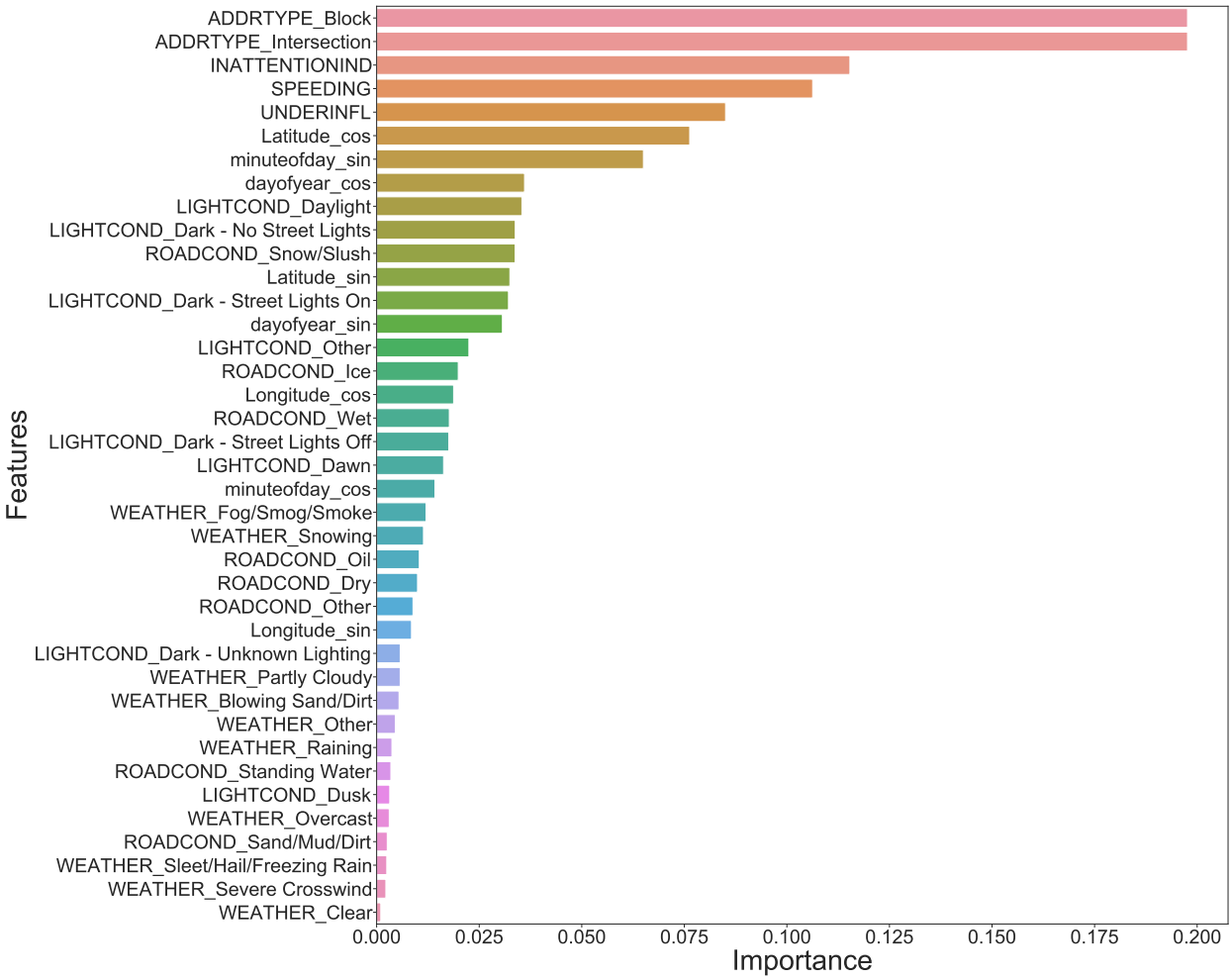


FIG. 20: Importance of the features (LR) based on training about 128000 accidents (80%) from the whole data.

influence of the alcohol or not, location, the time of day, the day of the year, light condition, road condition are among the very important features. These are the key features that the drivers, passenger, pedestrians, as well as the car manufactory, and the traffic department of the city need to be aware of.

VI. CONCLUSION

In this report, I use the historical traffic accident data of Seattle city, after selecting relevant features and doing feature engineering, I trained several models to predict the severity of possible traffic accidents in Seattle city with over 60% accuracy.

Among the models, overall, if we consider the combination of the accuracy, the training

and test time cost, LR is the winner for this problem based on the features we used as listed in Table. III. XGBoost the second. SVM uses more time for training while KNN uses more time for making prediction. RF and DT are not good enough due to over fitting and relatively low accuracy issue.

The trained models can be easily made into a software package for the car, in order to make real time traffic alarm for the driver. If the computing device on the car is powerful enough, we may use LR, XGBoost, SVM and KNN together to do real time prediction based on the already-trained model. However, if we were to make an App to run it on iphone or andoid phones, for real time analysis, LR model is clearly the best choice because it is fast (so it consumes less power) and reasonably accurate.

---

[1] A. Hébert, T. Guédon, T. Glatard, and B. Jaumard, CoRR **abs/1905.08770** (2019), arXiv:1905.08770.

[2] S. Moosavi, M. H. Samavatian, S. Parthasarathy, R. Teodorescu, and R. Ramnath, Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems 10.1145/3347146.3359078 (2019).

[3] In the 'INCDTTM' column, there are about 20000 records only have date and there are no hour, minute and second provided, those data did not provide hour of day information, so I removed them. So the total number of records reduced from about 160000 to 140000, after I plot the hour of day figure, I multiply the values by 16/14. Other figures such as month of year, week of year, day of year, etc are not affected. Whether deleting those 20000 records or not, does not really affect the rest part the report.

[4] In principle we need to split the data into three parts [6], usually 60% of the data as training set, 20% as cross validation set, 20% as testing set. We use training set and cross validation set to train and improve the model, then use test set to test the model.

[5] J. Brownlee, How to use roc curves and precision-recall curves for classification in python (2020).

[6] A. Ng, Machine Learning (2020).