COMP 542 Machine Learning

**Project Proposal**

Due date: by 11:59 PM on Wednesday (Mar. 30, 2022)

**Title: US Traffic Accident analysis**

Names (Group No.) : Group 4: Matthew Choi, Hui Du

1. **Problem Spaces**

(1 ~ 1 and ½ page: The problems that you would like to address – The reason why you do this project. The problem should be from existing or current status not general problem)

Looking briefly at the data, a couple questions come up that can be answered by further data analysis.

Problems regarding accident frequency:

* Which major cities/states have a higher frequency of accidents compared to others?



* When adding severity of accidents as a parameter amongst this ranking, how does the ranking change?
* Can accident frequency be purely explained by population density or can they be explained by another/combination of different metrics?



* Besides population density, what can be a useful indicator of potential higher rates of accident frequency?



These are interesting problems that stem from looking purely at accident frequency. Using the data, we can look to find relationships between overall frequency of accidents, as well as specific frequencies of accidents rated on a scale.

Problems regarding accidents when considering time:

In addition to accidents occurring at higher rates in specific locations, we can look at the spread of frequencies of accidents that occured over a specific interval of time.

The data was gathered over the time span of 2016 - 2021. Looking at data over the spread of years, we can look at the relationship between location and time. More specifically, we can see if the data shows any relationship between times of high traffic in certain areas and spikes of accident frequency in those same areas, during the same time span, over the course of several years.

Looking more specifically, we can see if certain months across the US have higher frequencies of accidents or look to find relationships between variance in accident frequencies when accounting for different segments of certain months or different segments of any given week.

Weather conditions can also be an important parameter when used alongside data gathered regarding time to find relationships between common weather patterns in certain parts of the year and US, and spikes or dips in accident frequency in those areas.



Thus we can look to predict potential changes in accident frequency in certain locations based on the relationships that are found between weather, time, and location and how these relationships describe changes in accident frequency in the specific weather, time, and location parameters.



1. **Your Approaches**

(1 ~ 1 and ½ page: Describe what approaches you will be providing in the project to address found problems)

The whole data set is very large, which records 2016-2021 the US traffic accident in specific conditions, such as time, weather, severity… in 49 dimensions and more than 3 million data. Thus, in this paper, for considering the hardware, the problem may restrict into a specific year or just in a region. What’s more, there must be some data missing and noise in each feature. In this paper, the first step is dealing with these missing data and noise.

Data Preprocessing:

For the missing data value (number), using the mean value instead.

For the missing data attribute (non number), using LabelEncoder class transforms the value into an integer number, and uses the common number instead.



For the noise, finding the outlier by using cluster, using the mean value instead.

To avoid overfitting, using train\_test\_split to split the training set and testing set

To avoid the large difference of the feature, some features need to feature scaling by using StandardScaler.

Data visualization:

For each data set, using the correct graph to clearly show the result and comparison of each States, weather condition or the time. For example, the histogram is the best way to show the accident number in each State.

Data Modeling:

Delete the useless features.

There must still be too many features in the data set, so using PCA to reduce the dimension.

Using Xgboost[1] to model the data and predict the severity of the traffic accident.

Show the accuracy and feature importance at last.

Result:

Adjust the parameters and try to get more accuracy for the result.

Approach Intro:

Compared with GBDT[2], XGBoost has a different Obj function. The step is following:



1. Keep adding trees and keep feature splitting to grow a tree. Each time we add a tree, we are actually learning a new function f(x) to fit the residuals of the last prediction.
2. When we finish training to get k trees, we want to predict the score of a sample, which is actually based on the features of this sample, in each tree will fall to the corresponding a leaf node, each leaf node corresponds to a score
3. Finally, we only need to add up the scores of each tree to the predicted value of the sample.
4. **Related works**

(1 ~ 1 and ½ page: Describe the current status or previous relevant works done by others and compare them to your proposed work)

There are currently many Exploratory Data Analyses (EDA) exploring the relationships between certain data points.

Some works look to give a much broader summary of the different relationships that can be explored within the data, while others go much further into detail in describing patterns within small time-frames, with one study looking at accident severity levels purely within the month of May 2019. Our proposed work would be looking at broader trends described by the dataset.

One study looked to answer many questions regarding the topics discussed in section 1 of this proposal. Among these questions are considerations regarding streets (specific locations) that have the highest accident frequency when compared to the other streets described by the dataset. Other considerations include looking at which point within the 24 hour span of a day has the highest frequency of accidents, as well as how basic weather conditions affected accident frequency.

(Only mention algorithms here, no visualization method. ) For instance, there’s two main parts for this dataset prediction: 1. end-time prediction. 2. Severity Prediction. For each of these two, there’s several methods and algorithms[3]:

1. End-time prediction:

* Linear Regression
* 2. Support Vector Machine
* 3. Decision Tree
* 4. Gradient Boosting Tree
* 5. Random Forest
* 6. XGBoost
* 7. Multi-layer Perceptron Regression

2. Severity prediction:

* 1. Logistics Regression
* 2. Support Vector Machine
* 3. Decision Tree
* 4. Gradient Boosting Tree
* 5. Random Forest
* 6. XGBoost
* 7. Multi-layer Perceptron Classification

For the first:

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm | RMSE | R2 | Adj\_R2 |
| Gradient Boosting Tree | 2.422596 | 0.813159 | 0.812088 |
| Random Forest | 2.626825 | 0.780329 | 0.779071 |
| XGBoost | 2.695338 | 0.768721 | 0.767396 |
| Support Vector Machine | 3.253056 | 0.663106 | 0.661176 |
| Decision Tree | 3.417098 | 0.628272 | 0.626142 |
| Multi-layer Perceptron | 3.541191 | 0.600783 | 0.598496 |
| Linear Regression | 4.985955 | 0.208580 | 0.204046 |

For the second:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Accuracy(Balanced) | Accuracy | Precision(Weighted\_avg) | Recall(Weighted\_avg) | F1-score(Weighted\_avg) | Roc\_ovo(macro) | Roc\_ovr(weighted) |
| XGBoost | 0.556345 | 0.903029 | 0.891821 | 0.903029 | 0.892511 | 0.909246 | 0.923071 |
| Gradient Boosting Tree | 0.501361 | 0.895249 | 0.878865 | 0.895249 | 0.881477 | 0.892311 | 0.915979 |
| Random Forest | 0.433038 | 0.882051 | 0.865390 | 0.882051 | 0.858590 | 0.890837 | 0.904347 |
| Decision Tree | 0.528765 | 0.844401 | 0.848112 | 0.844401 | 0.846146 | 0.686906 | 0.719075 |
| Multi-layer Perceptron | 0.399015 | 0.851487 | 0.812838 | 0.851487 | 0.823419 | 0.841784 | 0.847812 |
| Support Vector Machine | 0.408555 | 0.843429 | 0.806123 | 0.843429 | 0.818009 | 0.781210 | 0.750664 |
| Logistics Regression | 0.277131 | 0.856905 | 0.828878 | 0.856905 | 0.801722 | 0.789534 | 0.820884 |

1. **Possible Outcomes**

(1/2 page: Describe final outcomes that you expect and will achieve from your project)

For the data preprocessing:

In this paper, the final data set should be without missing value and reduce the noise as much as possible.



The features should be reduced as much as possible by using PCA.

For visualization:

The graph should clearly show which one is bigger, which one gets more shares. For example, the ratio of visible rate should use a pie chart to show which one gets the biggest share.

For modeling and testing result:

The model should end in the acceptable time and accuracy (more than 70%).



And if there’s time left, in this paper, the comparison of other algorithms is necessary.

Also, the result should show which features have the dominance effect on the training.



Parameter adjustment:

Try more times for adjusting parameters in order to improve the result.

Reference:

[1]: <https://github.com/dmlc/xgboost>. Chen, Tianqi; Guestrin, Carlos (2016). "XGBoost: A Scalable Tree Boosting System". In Krishnapuram, Balaji; Shah, Mohak; Smola, Alexander J.; Aggarwal, Charu C.; Shen, Dou; Rastogi, Rajeev (eds.). *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*. ACM. pp. 785–794. [arXiv](https://en.wikipedia.org/wiki/ArXiv_(identifier)):[1603.02754](https://arxiv.org/abs/1603.02754)

[2]: Cheng Li. ["A Gentle Introduction to Gradient Boosting"](http://www.chengli.io/tutorials/gradient_boosting.pdf)

[3]: Kaggle: https://www.kaggle.com/code/aaronds/us-traffic-accidents-analysis-and-prediction/notebook

Font: Cambria, Font size12, Line space: Single Line, Margin: Normal (1’’ for top, bottom, left, and right), Column: Single