Project: Predicting Accident Severity

I. Introduction: Business Problem

U.S. drivers' mileage has been climbing steadily since 2012. With more people on the roads, more accidents start to occur. The most recent data available shows that there are approximately 17 thousand traffic crashes each day nationwide. The significant increase of accident frequency leads to margins shrink for auto insurers. Insurance companies are processing thousands of claims, dealing with millions of losses every day. Although the number of car accidents seem to decrease for the last two years, the fact that around 36,560 motor vehicle deaths in 2018 is still a problem of concern. A drunk driver, speeding, bumpy road condition, or bad weather could all end up leading to a deadly accident. In fact, there are a lot of attributes that could cause a server car crush. By accurately predicting the possibility of getting into a car accident and how severe it would be, proper driving guidelines could be promoted to potentially change people's driving attitude and behaviors - bringing with it reducing cost for auto insurers, and fatalities rate.

II. Data Description

A. Data Source

The data used for this project is the example dataset (https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/Data-Collisions.csv) from Coursera course: applied data science capstone. It contains traffic collisions data provided by SPD and recorded by Traffic Records in Seattle from 2004 to present.

B. Feature Selections

The factors in this dataset that will influence the possibility of a car accident are:

- Collision address type: alley, block, or intersection
- Whether or not collision was due to inattention
- Whether or not a driver involved was under the influence of drugs or alcohol
- Whether or not speeding was a factor in the collision
- Weather conditions
- Road conditions
- Light conditions during the collision

This project aims to predict accident severity base on objective features: collision address, weather conditions, road conditions, and light conditions.

C. Data Cleaning

1. Remove collision that possibility caused by inattention, drugs or alcohol influence, and speeding.

The above features data contained problems of missing values and data formatting. All missing values were treated as "0", and "Y" values were treated as "1", converting these columns to two values: 0 and 1. The rows where contain "1" were dropped. The modified dataset only contained the rows where above features' value equal to 0, meaning the

collisions were not attributed from inattention, influence of drugs or alcohol, and speeding. Then, revised columns were removed for further process.

2. Check for missing values

The columns of collision address types, weather conditions, road conditions, and light conditions had the values of 0 or "unknown". Those values were treated as missing values and dropped.

3. Balance the data

The modified dataset had 87,019 data for severity 1, and 39,899 data for severity 2, indicating the dataset was unbalanced. Under-sampling method was used to balance the data.

4. Convert categorical data to numerical data

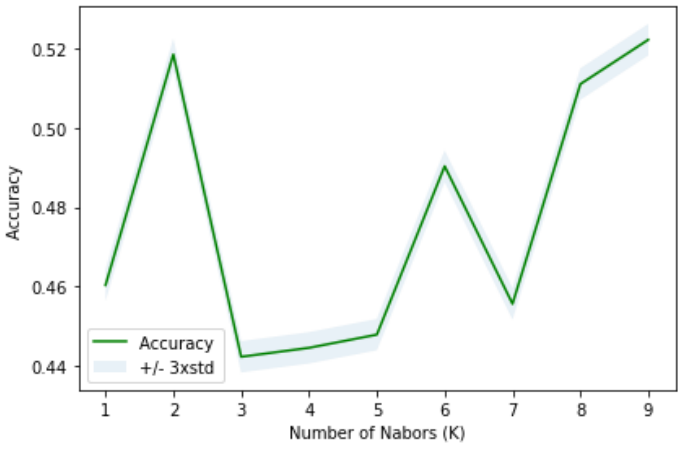
The columns data except for severity were all categorical. Converting them to numerical data would be a lot easier for further modeling process.

III. Methodology

The goal for this model is to accurately predict which level of severity the unknown collision cases should belong to, severity 1 or severity 2? This is a classification problem that determines the class label for an unlabeled test case given the dataset with predefined labels. I decided to use four various algorithms to build the model, calculate the accuracies of each model, and pick the one with the highest accuracy. I split the dataset into training set and testing set. All the following algorithms used the same set of training and testing data.

A. K-Nearest Neighbors

The first algorithm was KNN. In order to find the correct K value, I set the range of K from 1 to 10. Repeated the process by increasing K to see which K has the highest accuracy.



The best accuracy was with 0.52218 with K = 9. So I used K value as 9 to build the model.

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=None, n_neighbors=9, p=2,
           weights='uniform')
```

B. Decision Tree

High depth value could cause the problem of over fitting, so I set the depth value as 8 to build the model.

```
DecisionTreeClassifier(class_weight=None, criterion='entropy', max_depth=8,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, presort=False, random_state=1,
            splitter='best')
```

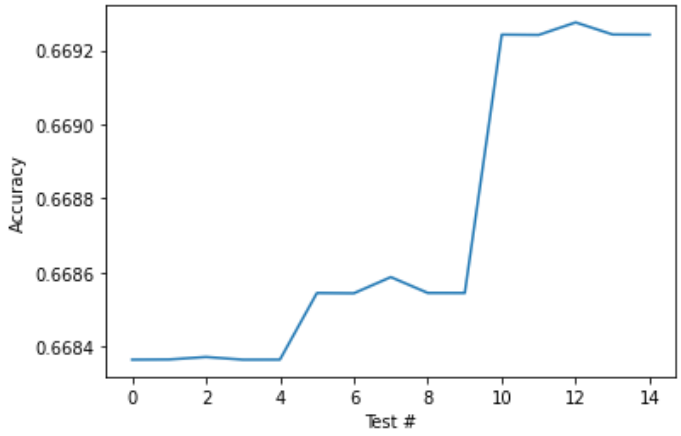
C. Support vector machine (SVM)

There are four types of kernelling in SVM model. I ran the model using each one of them (linear, polynomial, rbf, Sigmoid) and tested their accuracies. The result was that polynomial kernel had the highest accuracy score. The following graph shows the model built with polynomial kernel.

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
  kernel='poly', max_iter=-1, probability=False, random_state=None,
  shrinking=True, tol=0.001, verbose=False)
```

D. Logistic Regression

The last algorithms I tried was logistic regression. I ran 12 different tests with different c value and solver, and plotted the accuracy graph to find the best accuracies number.



It shows that test 12 has the highest accuracy.

```
LogisticRegression(C=0.001, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='warn',
          n_jobs=None, penalty='l2', random_state=None, solver='liblinear',
          tol=0.0001, verbose=0, warm_start=False)
```

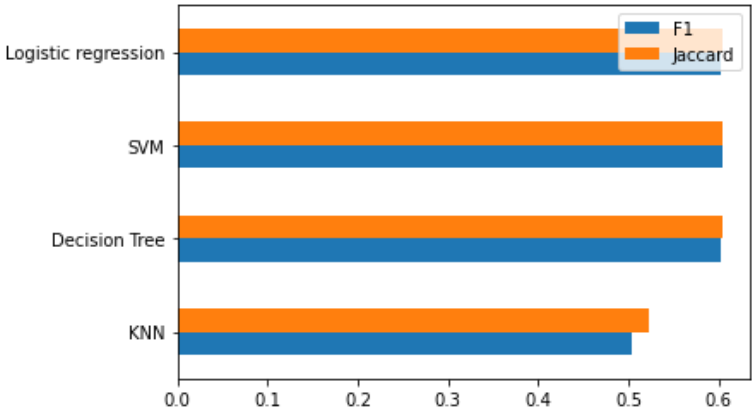
IV. Results

Different scoring methods were tested to find out the models' performance. I chose to use F1-score and Jaccard index to find out the accuracies of these four models built by different algorithms.

	F1	Jaccard
KNN	0.5040	0.52350
Decision Tree	0.6028	0.60457
SVM	0.6035	0.60526
Logistic regression	0.6030	0.60470

Log loss - logistic regression = 0.669.

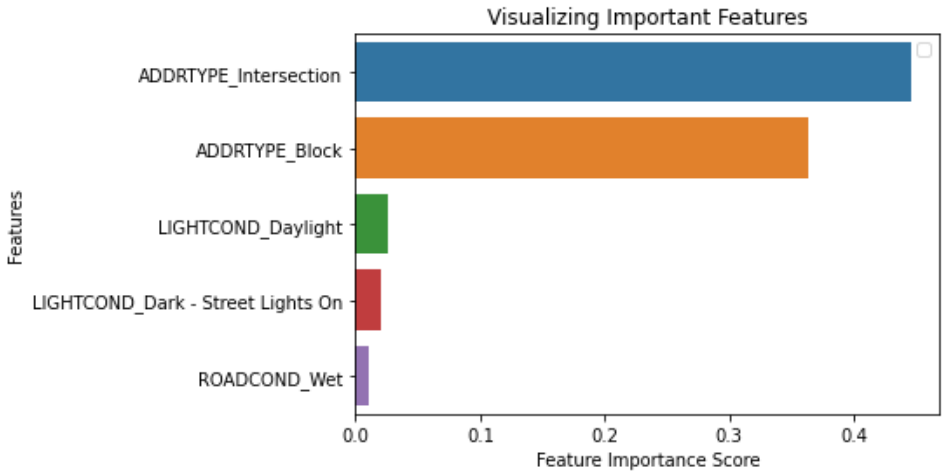
I plotted the horizontal bar graph for a better view.



Logistic regression machine learning algorithms has the highest accuracy score, where logloss is 0.669. Building a model using logistic regression can also predict the probability of the sample data and map the cases to a discrete class based on that probability, meaning that it will predict the probability of an unknown collision belonging to severity 1 or 2.

V. Discussion

The following graph shows the top 5 weighted features in the model.



"Intersection" and "Block" are the most important features of predicting a car accident, far ahead from the rest. Drivers should pay extra attention when they are driving close to intersection or block. Speed detectors or warning signs could be placed to reduce cars' speed.

I used logistic regression to build this machine learning model with accuracy of 0.669. Even though it's the highest score, there's still a lot of room for improvement. For more accurate prediction, the dataset can be added more detailed features like gender, age, visibility, etc.

VI. Conclusion

As one of the most common transportations right now, people's daily life is strongly related with car. The purpose of this project is to reduce claims for auto insurer and fatalities rate. Not only for insurance companies but also police departments will benefit from similar models to predict different accidents' severities.

VII. Reference

- fact+statistics: auto insurance
  https://www.iii.org/fact-statistic/facts-statistics-auto-insurance
- wiki

  https://en.wikipedia.org/wiki/Motor_vehicle_fatality_rate_in_U.S._by_year