CAPSTONE PROJECT

CAR ACCIDENT SEVERITY

SEATTLE

BY ANTONIO PALMA

Index

1.Introduction

1.1 Background

This data is regarding Seattle, Washington state. Total number of personal vehicles in Seattle in the year 2016 was 444,000 vehicles. The increase in car ownership rates can lead to higher numbers of accidents on the road because of a simple probability.

1.2 Problem

The National Highway Traffic Safety Administration of the USA suggests that the economic and social harm from car accidents can cost more than $800 billion per year. According to 2017 WSDOT data, a car accident occurs every 4 minutes and a person dies due to a car crash every 20 hours in the state of Washington while Fatal crashes went from 508 in 2016 to 525 in 2017, resulting in the death of 555 people. This project´s goal is to predict severity of an accident and how it can be reduced based on some factors.

1.3 Stakeholders

The reduction in severity of accidents can be beneficial to Seattle Authority which works towards improving those road factors and the car drivers who may take precaution to reduce the severity of accidents.

2. Understanding Data

2.1 Data Cleaning

The dataset has total observations of 194673 with variation in number of observations for every feature. First of all, the total dataset had high variation in the lengths of almost every column of the dataset. The dataset had a lot of empty columns which could have been beneficial had the data been present there. These columns included pedestrian granted way or not, segment lane key, crosswalk key and hit parked car. The models aim was to predict the severity of an accident, considering that, the variable of Severity Code was in the form of 1 (Property Damage Only) and 2 (Injury Collision) which were kept this way. Furthermore, the "Y" was given value of 1 whereas "N" and "no" value was given 0 for the variables Inattention, Speeding and Under the influence. For lighting conditions, Light was given 0 along with Medium as 1 and Dark as 2. For Road Condition, Dry was assigned 0, Mushy was assigned 1 and Wet was given 2. As for Weather Condition, 0 is Clear, Overcast is 1, Windy is 2 and Rain and Snow was given 3. 0 was assigned to the element of each variable which can be the least probable cause of severe accident whereas a high number represented adverse condition which can lead to a higher accident severity. Whereas, there were unique values for every variable which were either 'Other' or 'Unknown', deleting those rows entirely would have led to a lot of loss of data which is not preferred.

In order to deal with the issue of columns having a variation in frequency, arrays were made for each column which were encoded according to the original column and had equal proportion of elements as the original column. Then the arrays were imposed on the original columns in the positions which had 'Other' and 'Unknown' in them. This entire process of cleaning data led to a loss of almost 5000 rows which had redundant data, whereas other rows with unknown values were filled earlier.

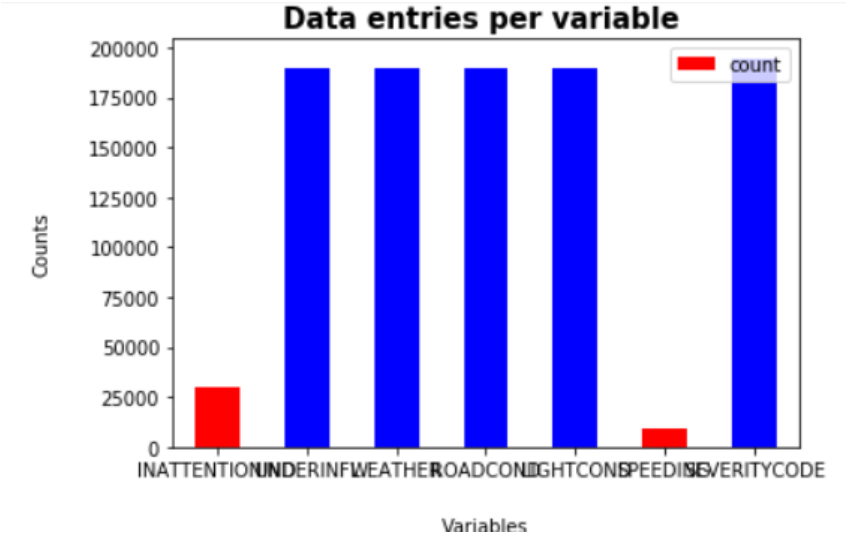

Fig. 1 – count of each variable

2.2 Variable Selection

A total of 6 variables were selected for this project along with the target variable being Severity Code, as per figure 1.

- Underinfl - wether or not the driver was under the influence (Y/N)
- Weather - Weather condition during time of collision (ovecast/rain/clear)
- Roadcond - road condition during collision (wet/dry)
- Lightcond - light condition during collision (lights on/dark with lights on)
- Speeding – whether driver was above speed limit or not
- InnattentionInd – whether or not collision was due to inattention

3. Methodology

The variables above were selected because they are commonly known as the top causes for car accidents.

After looking at data and making some cleaning, was noted that dataset is supervised but unbalanced and target variable was close to 2:1 ratio in favor of property damage. Because of this, and to try to have balanced data so machine learning algorithms are feasible I have used the function smote from imblearn library in order to balance the target variable in equal proportions and to have neutral classification model which is trained on equal instances of both the elements under severity of accidents.

The factor with most number of accidents under adverse conditions was adverse weather while adverse lighting condition was second. The factors that have less contribution were over-speeding and under the influence.

3.1 Machine Learning Models

k-Nearest Neighbor: is an algorithm for supervised learning where the data is 'trained' with data points corresponding to their classification. Once a point is to be predicted, it takes into account the 'K' nearest points to it to determine it's classification.

Decision Tree Analysis: The Decision Tree Analysis breaks down a data set into smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes.

Logistic Regression: Logistic Regression is a variation of Linear Regression, useful when the observed dependent variable, y, is categorical. It produces a formula that predicts the probability of the class label as a function of the independent variables.

Support Vector Machine (SVM) model was not used because is inaccurate for large data sets, like this dataset. Also, SVM works best with dataset filled with text and images.

4. Results

4.1 Decision Tree

Decision Tree Classifier from the scikit-learn library was used to run the Decision Tree Classification model on the Car Accident Severity data. The criterion chosen for the classifier was ‘entropy’ and the max depth was ‘6’. The post-SMOTE balanced data was used to predict and fit the Decision Tree Classifier.

4.1.1 Classification Report

	precision	recall	f1-score
0	0.68	0.69	0.68
1	0.38	0.36	0.37
micro avg	0.58	0.58	0.58
micro avg	0.53	0.53	0.53
weighted avg	0.57	0.58	0.58

4.1.2 Confusion Matrix

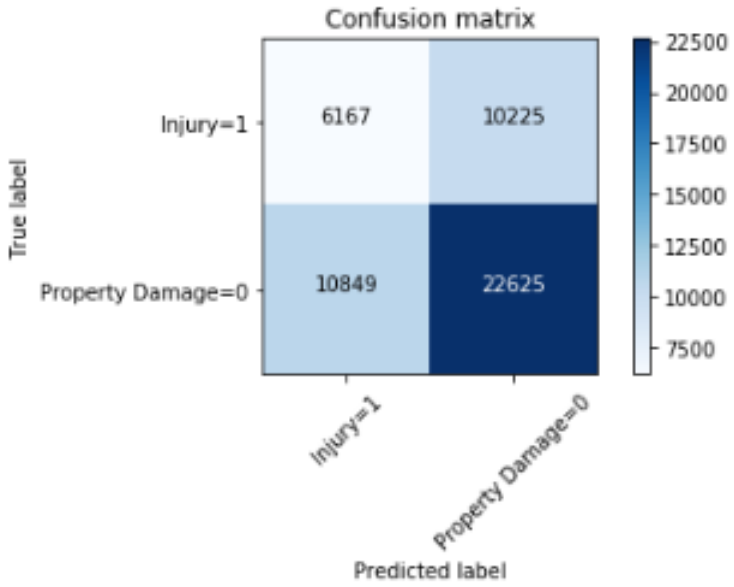

Fig 2 – Confusion matrix – Decision Tree

4.2 Logistic Regression

Logistic Regression from the scikit-learn library was used to run the Logistic Regression Classification model on the Car Accident Severity data. The C used for regularization strength was '0.01' whereas the solver used was 'liblinear'. The post-SMOTE balanced data was used to predict and fit the Logistic Regression Classifier

4.2.1 Classification Report

	precision	recall	f1-score
0	0.69	0.67	0.68
1	0.36	0.37	0.37
micro avg	0.58	0.58	0.58
micro avg	0.52	0.52	0.52
weighted avg	0.58	0.58	0.58

4.2.2 Confusion Matrix

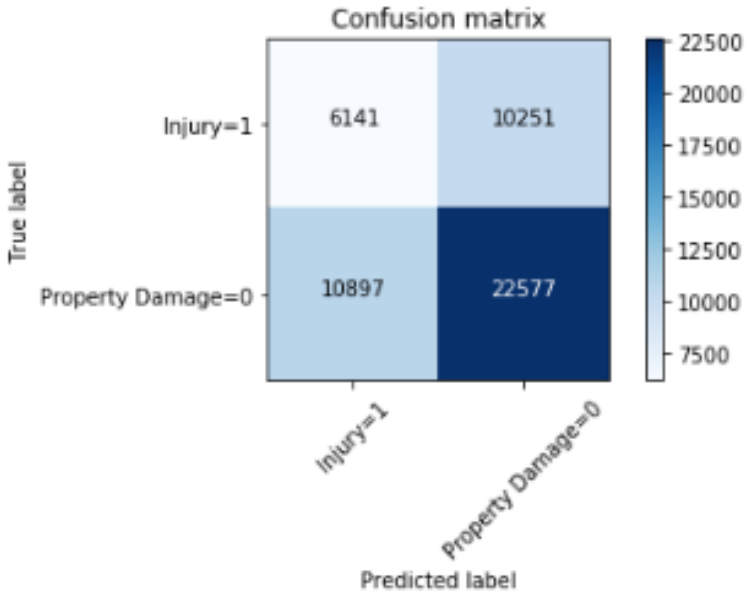

Fig 3 – Confusion matrix – Logistic Regression

4.3 k-Nearest Neighbor

k-Nearest Neighbor classifier was used from the scikit-learn library to run the k-Nearest Neighbor machine learning classifier on the Car Accident Severity data. The best K, as shown below, for the model is at 4.

4.3.1 Best Ks

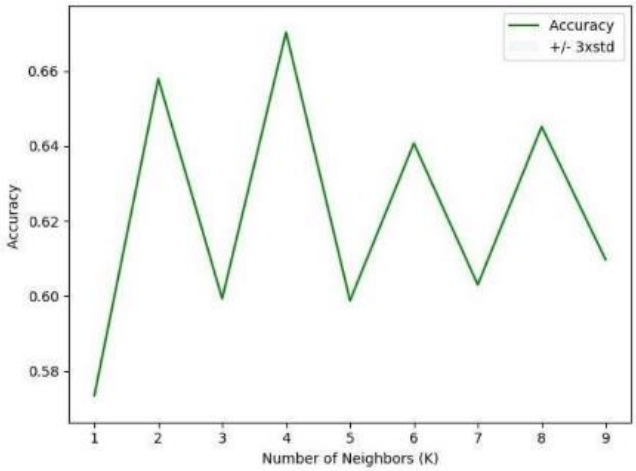

Fig 4 – Ks values

4.3.2 Classification Report

	precision	recall	f1-score
0	0.93	0.70	0.80
1	0.08	0.32	0.13
Accuraccy	0.67		
Macro avg	0.50	0.51	0.46
weighted avg	0.86	0.67	0.75

5. Discussion

Models	Avg. F1 score	Injury (1) vs property damage (0)	Precision	Recall
Decision Tree	0,58	0	0,68	0,69
		1	0,38	0,36
Log Regression	0,58	0	0,69	0,67
		1	0,36	0,37
KNN	0,75	0	0,93	0,70
		1	0,08	0,32

5.1 Average f1-Score

f1-score is a measure of accuracy of the model, meaning the harmonic mean of precision and recall.

Perfect precision and recall is shown by the f1-score of 1 and worst precision is value of 0 which means that either precision or recall is 0.

F1-score above is the average of individual f1-scores of the two elements of target variable, property damage and injury.

Comparing f1-scores of the three models, it is clear that KNN has the highest f1-score which means higher precision and recall.

Decision Tree model's f1-score and Logistic Regression f1 score are exactly the same at 0.58.

However, average f1-score doesn't clear show the true picture of models accuracy because of the different precision and recall of the different models on both elements of target variable. Hence, it is biased more towards the precision and recall of Property Damage due to its weightage in the model.

5.2 Precision

Precision is the percentage of results that are relevant, meaning how many of the selected items from the model are relevant.

It is the division of true positives by true positives and false positives.

The highest precision for Property Damage is for Logistic Regression, whereas for Injury it is the Decision Tree, but it can be seen that both values are very close.

The Precision is calculated individually in order to understand how accurate the model is at predicting Property Damage and Injury individually.

For Decision Tree model, precision of Property Damage is 0.68 and for Injury is 0.38 which is fairly good.

For Logistic Regression model, Property Damage is 0.69 and Injury is 0.36.

For KNN, Property Damage is 0.93, which is highly accurate, but for Injury is 0.08, which is really low.

In terms of precision, both Decision Tree and Logistic Regression models are very good.

5.3 Recall

Recall is the percentage of total relevant results correctly classified by the algorithm, meaning how many relevant items were selected.

It is the division of true positives by true positives and false negatives.

The highest value for Property Damage is when using the KNN model at 0.70 and for Injury is the Logistic Regression model at 0.37.

Recall for both Property Damage and Injury is very close for Decision Tree and Logistic Regression model. For KNN is slightly different with recall for Property

Damage is 0.70 and Injury is 0.32. The recall for Property Damage and Injury is the most balanced in terms of being good for both the outputs of the target variable.

6. Conclusion

Comparing all values of f1-scores, Precision and Recall for the 3 models, it is possible to have a clear picture in terms of accuracy of the models individually and as a whole and how well they perform for each output of the target variable.

When comparing these scores, we can see that the f1-score is highest for KNN at 0.75. However, when comparing precision and recall for each of the models, it is clear that KNN model performs poorly in the precision of 1 (0.08).

The variance is too high for this model to be selected as a viable option.

Looking at the other two models, both the Decision Tree and Logistic Regression have a more balanced precision and recall for 0 and 1.

Also, average f1-score of the two models ar exactly the same.

We can then infer that both models can be used for best performance.

When comparing these scores to the benchmarks within the industry, it can be seen that they perform well but not as good as the benchmarks. Maybe they could have performed better if:

- There was a balanced dataset for the target variable
- Less missing values in the original dataset for variables like Speeding and Under the influence

7. Recommendations

After evaluating the data and the output of the Machine Learning models, a few recommendations can be made.

Looking that almost all the accidents have occurred on a block or an intersection, more traffic lights, safety signs or better light conditions could be implemented to improve road conditions and reduce number of accidents.

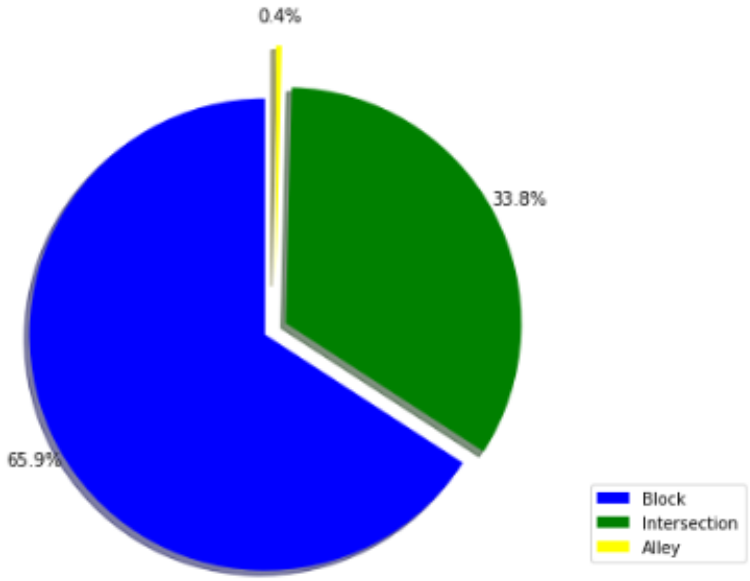

Fig 5 – accident area

Also, regarding the concentration of accidents, it can be seen that most of them are on the main roads, near the highway in the city center. The car drivers could also use this data to assess when to take extra precautions on the road under the given circumstances of light condition, road condition and weather, especially near or around I-5.

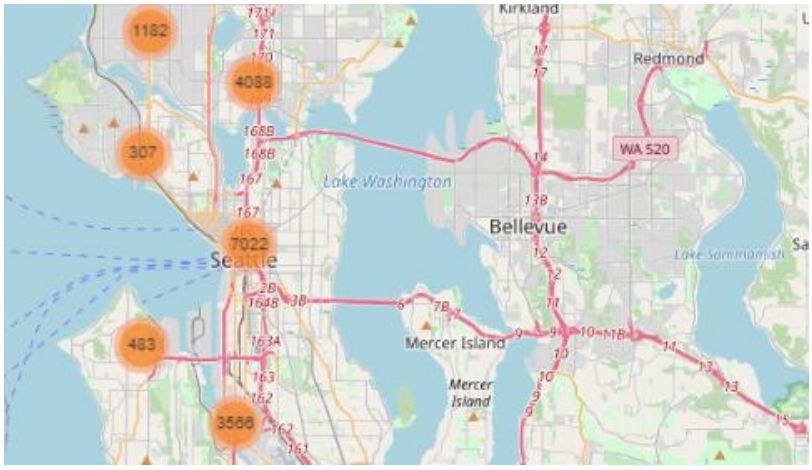

Fig 6 – concentration of accidents