CAR ACCIDENT SEVERITY PREDICTION

APPLIED DATA SCIENCE CAPSTONE PROJECT

ARVIND N

GITHUB LINK: HTTPS://GITHUB.COM/ARVIND-2021/COURSERA_CAPSTONE

Introduction | Business Understanding

Road accidents could be considered an old topic but with the progress of cars and the capabilities of the technology they carry, it is even more important to have models available to predict and mitigate their occurrences. However, predicting the severity of a car crash is not an easy task. Even tough it is possible; precision levels will vary significantly depending on the data available and how well the problem has been modeled.

A machine learning model is required to predict the severity of an accident given the conditions like weather, road and visibility conditions. When conditions are bad, our job is to alert drivers about the increased risk of a car accident.

Description of Data

Data set used for the capstone project is from Seattle DoT. The data comes from records of all collisions provided by SPD and captured in traffic Records. Data includes all types of collisions which occurred at the intersection or mid-block of a segment. Timeframe of the dataset is from 2004 and it's automatically updated on a weekly basis. The dataset comes with a pdf file containing a clear definition for each of the available features.

Our goal is to predict the severity of a crash. In the dataset, the target variable is called 'SEVERITYCODE' because it is used measure the severity of an accident. To achieve our goal, we will go through the following steps:

1. Feature Exploration (with data cleaning)
2. Dimensionality reduction
3. Model building
4. Optimization and final model selection

Considering that our problem is a classification one, we will use an F1 score to evaluate the performance of the different models that we will train. However, we will also take a close look at accuracy, precision and recall since they all provide valuable insights into understanding how the model is performing.

Finally, we will explain and give details on what are some possible next steps to further improve the overall performance and interpretability of the chosen model.

Methodology

Data Cleaning

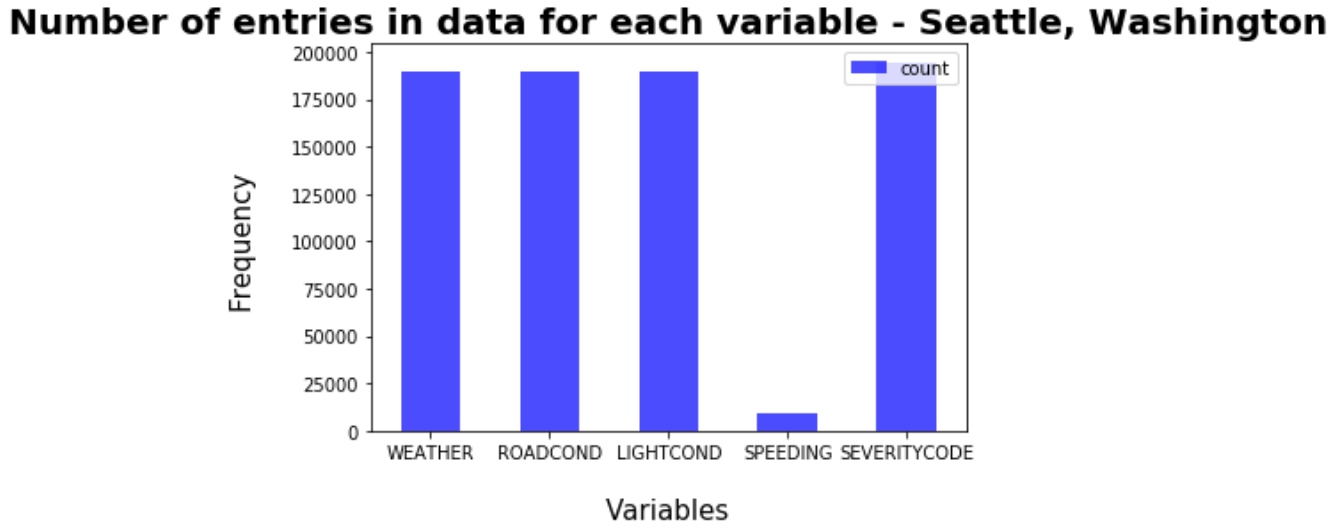
This is a machine learning project which uses classification to predict a categorical variable. First, the total dataset was high variation in the lengths of almost every column of the dataset. The dataset had a lot of empty columns which could have been beneficial had the data been present there. These columns included pedestrian granted way or not, walk key and hit parked car.

Feature Selection

Initially, a total of 4 features were selected for this project along with the target variable being Severity Code.

Name of Feature	Feature Explanation
WEATHER	Weather condition during time of collision
ROADCOND	Road condition during the collision
LIGHTCOND	Light conditions during the collision
SPEEDING	Whether the car was above the speed limit at the time of collision

Below chart shows variation in each of our selected feature in the dataset.



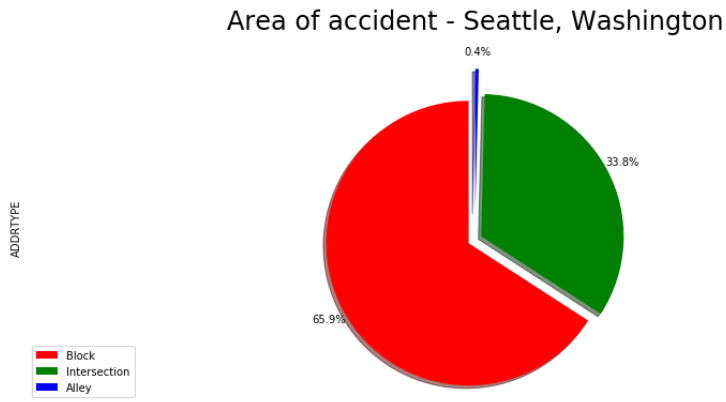
Looking at the number of entries for "speeding" feature, it is decided to drop this feature from our model as the missing values would cause issues in building an unbiased model. Final feature selection is as mentioned below

Name of Feature	Feature Explanation
WEATHER	Weather condition during time of collision
ROADCOND	Road condition during the collision
LIGHTCOND	Light conditions during the collision

Exploratory Data Analysis

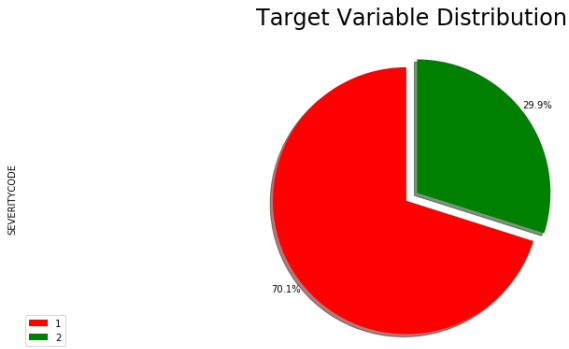
Generally, considering the effect of these variables in car accidents are important hence these variables were selected. A few pictorial depictions of the dataset were made in order to better understand the data.

Below pie-chart shows where are the accidents frequently occurring. Majority of the accidents in our dataset have occurred in and around “blocks”



Unbalanced Dataset

Let us look at the distribution of the target variables between Physical Injury and Property Damage Only. As the dataset is supervised but an unbalanced dataset where the distribution of the target variable is in almost 1:2.5 ratio in favor of property damage. It is very important to have a balanced dataset when using machine learning algorithms.



Our target variable SEVERITYCODE is only 42% balanced. In fact, severity code in class 1 is nearly three times the size of class 2. We can fix this by down sampling the majority class.

Machine leaning Models

We will build the following models:

K-Nearest Neighbor (KNN)

K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (based on distance). KNN will help us predict the severity code of an outcome by finding the most similar data point within k distance.

Decision Tree

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 result is a tree with decision nodes and leaf nodes. Decision tree model gives us a layout of all possible outcomes so we can fully analyze the consequences of a decision. In context, the decision tree observes all possible outcomes of different weather conditions.

Logistic Regression

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable. Because our dataset only provides us with two severity code outcomes, our model will only predict one of those two classes. This makes our data binary, which is perfect to use with logistic regression.

Results

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 for the model where the highest elbow bend exists is at 25. The balanced data was used to predict and fit the k-Nearest Neighbor classifier.

Below is the score from K-NN model

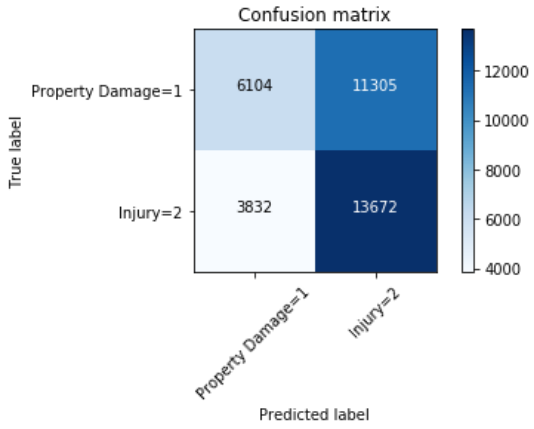
	KNN
Jaccard Index	0.56
F1- Score	0.54

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 ‘7’. The balanced data was used to predict and fit the Decision Tree Classifier.

Below is the score from Decision tree model and the confusion matrix

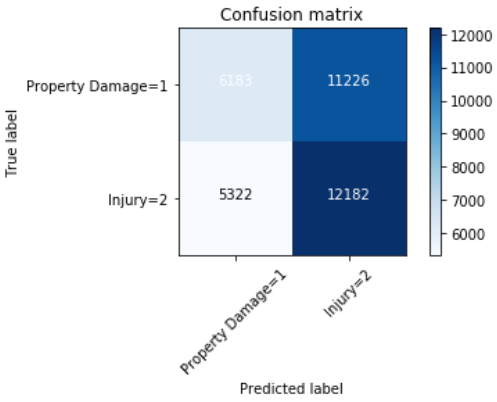


	Decision tree
Jaccard Index	0.57
F1- Score	0.55

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 ‘6’ whereas the solver used was ‘liblinear’. The balanced data was used to predict and fit the Logistic Regression Classifier.

Below is the score from Decision tree model and the confusion matrix



	Logistic regression
Jaccard Index	0.53
F1- Score	0.51
LogLoss	0.68

Discussion

In our dataset, we had categorical data that was of type 'object'. This is not a data type that we could have fed through an algorithm, so label encoding was used to created new classes that were of type int8; a numerical data type. As mentioned earlier, class 1 was nearly three times larger than class 2. The solution to this was down sampling the majority class with sklearn's resample tool. We down sampled to match the minority class exactly with same values each.

Once we analyzed and cleaned the data, it was then fed through three ML models; K-Nearest Neighbor, Decision Tree and Logistic Regression. Although the first two are ideal for this project, logistic regression made most sense because of its binary nature.

Evaluation metrics used to test the accuracy of our models were jaccard index, f-1 score and logloss for logistic regression. Choosing different k, max depth and hyperparameter C values helped to improve our accuracy to be the best possible. Below is the summary of results from all 3 models.

	KNN	Decision tree	Logistic regression
Jaccard Index	0.56	0.57	0.53
F1- Score	0.54	0.55	0.51
LogLoss			0.68

Conclusion

When comparing all the models by their f1-scores, we can have a clearer picture in terms of the accuracy of the three models individually 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 similar for k-Nearest Neighbor and Decision tree model. Furthermore, the average f1-score of the two models are very close but for the Logistic Regression it is lower by only 0.04. It can be concluded that any of the models can be used side by side for the best performance.

Recommendation:

When comparing these scores to the benchmarks within the industry, they perform well but not as good as the benchmarks. These models could have performed better if a few more things were present and possible.

- A balanced dataset for the target variable. More instances recorded of all the accidents taken place in Seattle, Washington

- Less missing values within the dataset for variables such as Speeding and Under the influence