Predicting the severity of accidents

Introduction and background

The City of Seattle collects information on accidents including factors such as location, weather, injury type, and lighting conditions. This has led to data being available for thousands of accidents, however the problem is that this data is not being leveraged as well as it could be.

The idea for this capstone project is to use all the available data to build a machine learning model that can predict future accident severity based on known factors. The audience for such a solution would be drivers as they are concerned about their safety when driving. This model could help inform drivers of the increased risks of driving giving the presence of certain factors, which could help reduce the quantity and severity of accidents.

As noted, I will be using the accident data collected for the City of Seattle. While the data has 37 attributes, most of these attributes are not useful for predicting accident severity, for example the junction type and codes used for labeling the accident. Therefore only relevant factors with a high likelihood of predictability will be used. This includes the weather (overcast, raining, clear etc.), the road condition (wet, dry etc.), and lighting conditions (daytime, nighttime with or without streetlamps). Only three features are being used for this machine learning model as to avoid over fitting the model.

This data will be split into a training and testing set, where a portion of the data will be used to train and build the model, and the other will be used to test its effectiveness.

Methodology

After loading the data into Python, I began by getting a high-level look at the data using the '.head' function to understand what the dataset looked like. By doing so, I was able to see a few entries in the dataset and to see all the features included. This allowed me to narrow down which features would ultimately be included in the machine learning model, for example eliminating non-predictable features such as 'report number.' The next important step was to examine the dependent variable, which was severity code, to see if any changes needed to be made to the data. After visualizing the data using a simple bar graph, it was clear the data needed to be balanced to create a better machine learning model. Before balancing the data, there were 136,485 instances of severity code 1, and 58,188 instances of code 2. After shuffling the data and creating a new data frame, there was an equal amount of instances of both severity code 1 and 2, of 58,188. Balancing the dataset is very important as it ensures that the model is not biased.

Many machine learning techniques require that the variables be numerical in nature and not categorical, for example "dark – streetlights on." All three independent variables that were chosen – weather, road conditions and lighting conditions – were categorical variables, therefore

it was necessary to transform these into numerical values. The technique used is called label encoding, where each unique value in a column is converted to a number, for example wet gets attached 0, and dry gets attached a 1. Now that the values were all numerical, it was possible to begin building the machine learning model.

The dependent variable severity code was represented numerically by either a 1 or a 2, however it could also be looked at as classes, in that the accident was either in class 1, or 2. Logistic regression, which is a variation of linear regression, is useful for predicting the probability of a value belonging to a class, which is exactly what the model would be used for, therefore logistic regression was a good fit for building the machine learning model. The data was split into X, or the independent variables, and Y, the dependent variable. The data was then normalized using 'preprocessing' to change the values to a common scale, and then finally split into a training and testing set. Ultimately, the model was used to predict whether an accident would be a severity code 1 or 2, and the accuracy and effectiveness of this model will be discussed in the following section.

Results

After building the model, its very important to test and evaluate its performance to understand its ability to predict. To evaluate this model, I used the Jaccard index, which looks at the difference between the predicted labels and the actual values, the Logarithmic loss, which measures the performance of the model in terms of how the classifier performs, and finally the F1 score which is the average of precision and recall for a model. The values for the above noted are 0.527, 0.684, and 0.51, respectively. All three of these accuracy evaluations measure between 0 and 1, with the exception that 0 is a better accuracy score for Logarithmic loss.

Based on these scores, it can be said that the machine learning model using weather, road conditions, and lighting conditions as the independent variables is able to predict whether an accident will belong to class 1 or 2 with moderate accuracy. Prior to building this model, I had hoped for slightly better scores, however the shortcomings of this model will be discussed in the discussion section, including steps that could improve the model in the future.

Discussion

One recommendation to improve the predictability and the usefulness of the model would be to include a few more independent variables. This would include features such as location, person count and incident date. Including these features might improve the accuracy of the model, however more importantly it relates to the original purpose of the model which is to give drivers more information on the severity of an accident given certain conditions. Drivers would likely be curious as to how the location would affect the chances of an accident, for example on the highway versus a two-lane road. As well, it could be useful to know if the number of people in a car affects the likelihood of an accident or if certain days or times within a day are more

dangerous than others. Including these features could make the model more beneficial to its audience.

The second recommendation would be to perform more feature engineering on the dataset. Feature engineering is “the process of using data’s domain knowledge to create features that make machine learning algorithms work.” (Odegua). Certain feature engineering steps were taken in the building of the machine learning model, including extracting only certain features, balancing the data set to improve the machine learning model, and transforming categorical variables into numerical variables. That being said, to improve the model further more steps could be taken such as dealing with missing values, using an objective measure for selecting features such as evaluating them based on a score, and using a better method for transforming categorical variables into numerical ones, such as one-hot encoding.

Conclusion

Based on a large dataset of accidents in the City of Seattle containing 37 features ranging from the location to the weather to the person count, I was able to create a machine learning model which is able to predict the severity of an accident based off of three features, namely the weather, the road conditions and finally the lighting conditions. After evaluating the model, I was disappointed to find out its accuracy in predicting the severity code was only moderate, however I have reflected above on how the model could be improved in the future to increase in accuracy score and usefulness to the models audience which is drivers.

This has been an extremely useful project in applying and refining the skills I have learnt throughout the duration of the IBM Data Science course.

References

Odegua, R. (n.d.). *A Practical Guide to Feature Engineering in Python.* Retrieved from Heartbeat: https://heartbeat.fritz.ai/a-practical-guide-to-feature-engineering-in-python-8326e40747c8