Tarun Abichandani
Applied Data Science Capstone
Week 3 Full Report

Machine Learning using Seattle's Collision data

Introduction:

The problem that occurs currently is that there is no system that alerts the drivers when they plan a trip on Google Maps or similar platforms. There is an expected time shown to the drivers but nothing alerts them about the severity of the driving conditions based on road conditions, time of the day or weather conditions.

Once a model is trained and tested based on the shared data provided, a small blip on the side on the estimated time can be shown that rates the severity between 1 and 2. If the severity is 2, the user/driver can be suggested to be cautious while driving or delay the trip if possible.

Data:

The collision data that is shared can be used to train the model that will help in predicting the severity level based on different factors. There will be a lot of pre-processing involved in finalizing the columns that will make the feature and as taught in the course, NaN will be replaced by either mean or dummy variables. Dummy variables will also be used to convert strings such as weather conditions, light conditions and road conditions.

After the feature is ready to be fed in the model, 3 Machine Learning techniques will be used to train and test the model (KNN, SVM and Logistic Regression) and the efficiency of the predicted results will be calculated by Jaccard and F1 index.

The data used for this capstone includes columns severity code, weather, road conditions, light conditions, and speeding. The data will be converted into X feature and a model will be trained to predict severity code label based on Machine Learning techniques.

Methodology:

The methodology used for this course is pretty straight forward. Whatever was taught in the other courses such as Data Analysis using Python, Machine Learning with Python was used to get the results for the Seattle's collision dataset.

The data was read and converted into Data Frame using the Pandas library. Once the data was explored, shape of the data and data types were checked.

The next steps included exploring the fields that were chosen to be part of the feature for the Machine Learning model. The fields included ROADCOND, LIGHTCOND, WEATHER, SPEEDING, and SEVERITYCODE.

Once each of these fields were explored, the SPEEDING column had plethora of NaN values. First these values were converted into string “N” and then Y/N were replaced by 1/0 respectively.

The WEATHER, LIGHTCOND, ROADCOND had around 2.5% of data missing from each column and no clear majority label was seen in any of the fields.
For WEATHER, only 57% of the weather was clear and hence could not replace the missing values with clear weather. Same for LIGHTCOND, only 59% of the light condition data had Daylight (not a majority) and hence Daylight could not replace the missing values. Only 63% of the ROADCOND data was dry and hence Dry could not replace the missing values as well.

After careful consideration, it was decided to delete the rows with any missing data and that changed the shape of the date from 194673 rows to 189337 (reduction by only 2.7% of the total data).

The next step included creating dummy variables for all three factors (weather, roadcond, lightcond) using pandas. Once the dummy variables were created, certain columns were dropped either because they didn’t affect the complete dataset as a whole (less than 10k) or they were not relatable as the correlation (such as others and unknown columns). The dummy variables data frames were concatenated with the original feature and the original variables were dropped leaving behind dummy variables, speeding, and severity code columns.

Since the shape of the data was still huge was KNN and SVM, a random set of 50000 datasets was used for training and testing the machine learning model. The index was reset and the data was standardized for the Train Test Split.

X and y were designated accordingly and the dataset was split into testing and training data. 90% of the 50000 data points were used for training the machine learning model and 10% of the data set was used to test the trained model. Since the number of severity code 2 is less as compared to the severity code 1, Imbalance learning package was used to balance the training dataset.

K Nearest Neighbor (KNN) Model
The model was trained initially using the value of k as 3. Then the data was put in a for loop to find out the best k for training the final model. The best k came out to be 14 and that k was used to train the KNN model

Support Vector Machine (SVM) Model
SVM model was trained using the X_train_res, y_train_res data and was used to predict the yhat for the SVM predictions.

Logistic Regression (LR) Model
LR model was also trained using the X_train_res, y_train_res data and was used to predict the yhat for the LR predictions.

Results:
The results were calculated using the Jaccard index and the F-1 score for the KNN and the SVM model. The LR model had an extra index i.e. the LogLoss component.

The results depict that approximately 70% of the times, the model correctly predicted the Severity Code in all three Machine Learning Models when evaluated using the **Jaccard index**.

KNN: 68.44% SVM: 47.10% LR: 46.76%

F-1 score evaluations are as follows:

KNN: 57.80% SVM: 46.29% LR: 45.63%

The Log Loss evaluations for the Logistic Regression model are as follows:

LR: 66.56%

Discussion:
Even though the predictions are 70% true as per Jaccard index for the KNN model, a few places to improve the predictions aka train the model better:

- Rather than taking random sample of 50000 datasets, maybe take 100000 or all of the data set. Taking all of the data set was tried but the jupyter notebook took way longer to predict any results for the best K in the KNN model. Hence random sample of 50000 was taken for the models.
- A few columns were dropped when converting columns into Dummy Variables. This was done to limit the number of columns to between 10-15. The columns such as others, unknown, with significantly less values were dropped since the correlation was not relatable.

Conclusion:
Even though there are certain limitations right now in the current model (limited datasets, few columns dropped), it serves as a significant starting point for the companies that provide navigation services to their users. Based on the conditions of the weather expected, the companies can use this ML model to predict the severity code and show a little warning on the side of the directed route if the prediction comes out to be 2. The driver will be cautious if they see a warning and will be cautious while driving or might even delay the travel plans. This can definitely help in reducing the number of accidents and saving lives.