Predicting the Severity of Traffic Accidents with Machine Learning

Darius Parker

Introduction: Business Understanding

The availability of cheap and abundant computation, cheap and abundant data as well as user friendly machine learning software packages has provided the opportunity for individuals and enterprises to explore data with powerful tools and efficiency that were once not available to solve real world problems.

Motor vehicle accidents are amounts the leading cause of injury and death in the US. The purpose of this paper is to answer if popular machine learning techniques may be applied to predict the servility of traffic accidents. The findings of this report will be targeted to stakeholders interested in finding patterns and trends to make decisions to improve traffic safety.

Data

The “Data-Collisions” example dataset provided in this course will be used for analysis. The Metadata is available by Clicking here. The dataset contains 37 features, and 194673 cases. Due to the large number of cases, SVM will not be applied. Features which may be suitable for KNN and logistic regression were selected and converted from string to integer categorical labels. Miscellaneous and redundant variables such as report numbers, incident id’s ect were excluded. Lastly, rows with missing values were removed. After cleaning, the dataset contained 183177 cases and the following 12 columns. Examples of some features are below:

SEVERITYDESC – Severity Description. This will be our target variable, with categories 'Injury Collision' or 'Property Damage Only Collision'.

COLLISIONTYPE – Collision type. Contains the following categories:

['Angles', 'Sideswipe', 'Parked Car', 'Other', 'Cycles', 'Rear Ended', 'Head On', 'Left Turn', 'Pedestrian', 'Right Turn']

JUNCTIONTYPE - Category of junction at which collision took place. Contains the following categories:

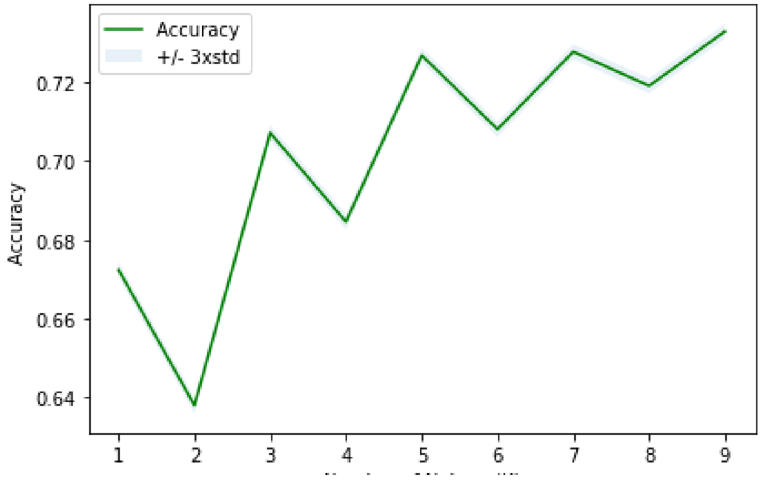
['At Intersection (intersection related)', 'Mid-Block (not related to intersection)', 'Driveway Junction', 'Mid-Block (but intersection related)', 'At Intersection (but not related to intersection)', 'Unknown', 'Ramp Junction']

List of all variables that will be used in the model below, see metadata for a detailed description:

['COLLISIONTYPE', 'JUNCTIONTYPE','PERSONCOUNT', 'PEDCOUNT', 'PEDCYLCOUNT', 'VEHCOUNT','WEATHER', 'ROADCOND', 'LIGHTCOND', 'SEVERITYDESC']

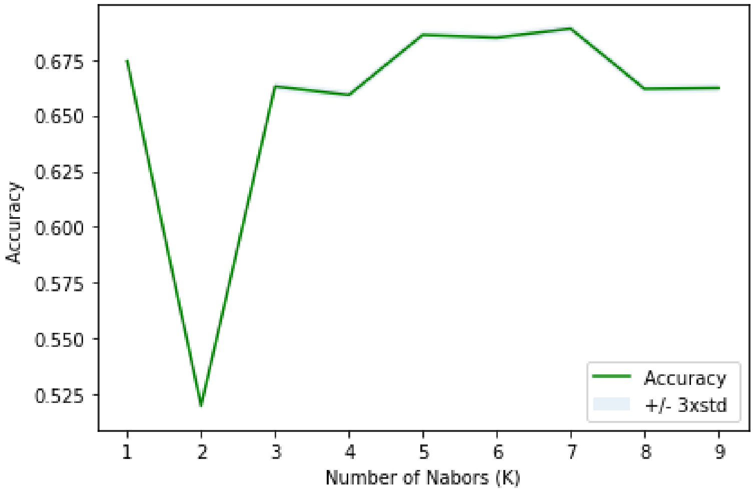
Modeling

First, a K Nearest Neighbor classification algorithm was used to predict our target variable, SEVERITYDESC, which may have a result of either “Property Damage Only Collision” or “Injury Collision”. For the first trial, all available features above were used. Using all features we can predict with .73 accuracy the severity of an accident, with k = 9.



However, an argument can be made that variables such as 'PERSONCOUNT', 'PEDCOUNT', 'PEDCYLCOUNT', and 'VEHCOUNT' should be removed. One does not need a machine learning model to predict if an accident involving pedestrians, bicyclist or multi car accidents may be severe. Also, PERSONCOUNT was removed as one can assume that the odds of an injury increase as there are more individuals involved in an accident who may be injured.

For the second trial, only features related to road conditions, weather and lighting were used. This model was less accurate than the former, at .68 accuracy, with k = 7.



```
The best accuracy was with 0.6889398405939513 with k= 7
```

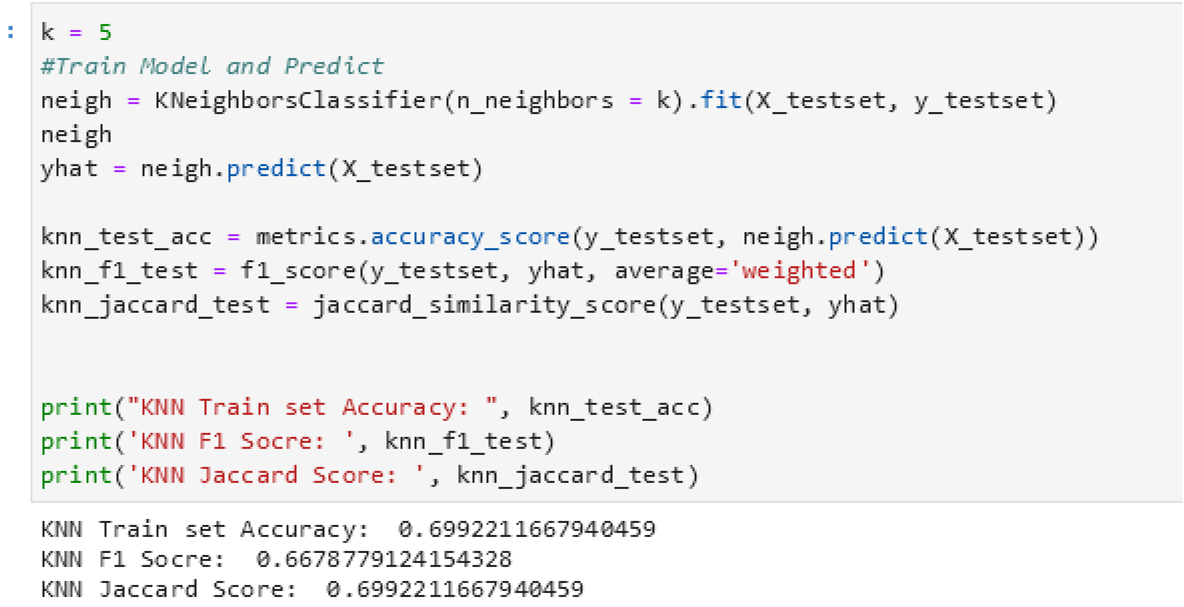
Next, see how this compares to using only the type of collision and type of intersection. Using 'COLLISIONTYPE', 'JUNCTIONTYPE' only, accuracy is slightly less than our model including all variables at .69 accuracy, with K = 5. Thus far, this was our best model given that all features in our first model are not needed. This model also had higher metrics for F1 score and jaccard score (see notebook for detailed scores for each trial).

Next, KNN results were compared to logistic regression. For simplicity, only the features from our best KNN were used to compare ('COLLISIONTYPE', 'JUNCTIONTYPE'). The logistic regression model did not perform well compared KNN algorithm. A trial was also ran using logistic regression and features from our second trial above ('WEATHER', 'ROADCOND', 'LIGHTCOND'). This also yielded poor results.

Results

Our two best models can be compared below. We see that our best model is the KNN algorithm with features related to the the type of collision (ie head on, rear end, side swipe ['COLLISIONTYPE]) and the type of location of the accident (such as drives ways, “mid – block”, intersections ect ['JUNCTIONTYPE']).

K-Nearest Neighbors:



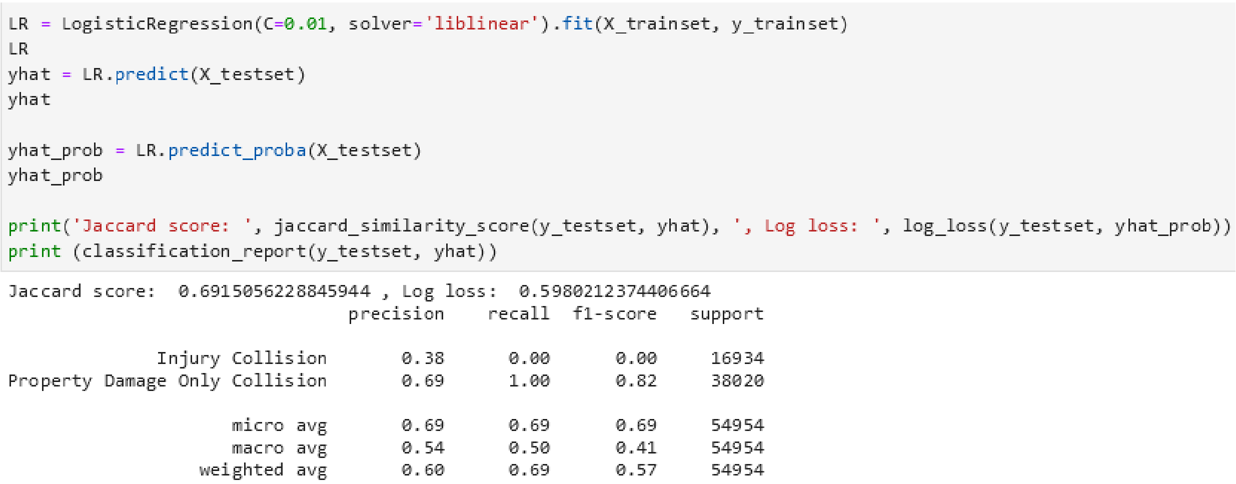
```
: k = 5
  #Train Model and Predict
  neigh = KNeighborsClassifier(n_neighbors = k).fit(X_testset, y_testset)
  neigh
  yhat = neigh.predict(X_testset)

  knn_test_acc = metrics.accuracy_score(y_testset, neigh.predict(X_testset))
  knn_f1_test = f1_score(y_testset, yhat, average='weighted')
  knn_jaccard_test = jaccard_similarity_score(y_testset, yhat)


  print("KNN Train set Accuracy: ", knn_test_acc)
  print('KNN F1 Socre: ', knn_f1_test)
  print('KNN Jaccard Score: ', knn_jaccard_test)
```

```
KNN Train set Accuracy:  0.6992211667940459
KNN F1 Socre:  0.6678779124154328
KNN Jaccard Score:  0.6992211667940459
```

Logistic Regression:



```
LR = LogisticRegression(C=0.01, solver='liblinear').fit(X_trainset, y_trainset)
LR
yhat = LR.predict(X_testset)
yhat

yhat_prob = LR.predict_proba(X_testset)
yhat_prob

print('Jaccard score: ', jaccard_similarity_score(y_testset, yhat), ', Log loss: ', log_loss(y_testset, yhat_prob))
print (classification_report(y_testset, yhat))
```

```
Jaccard score:  0.6915056228845944 , Log loss:  0.5980212374406664
                              precision    recall  f1-score   support

              Injury Collision       0.38      0.00      0.00     16934
Property Damage Only Collision       0.69      1.00      0.82     38020

                     micro avg       0.69      0.69      0.69     54954
                     macro avg       0.54      0.50      0.41     54954
                  weighted avg       0.60      0.69      0.57     54954
```

Discussion, Potential Improvements

In the process of selecting features available for each modeling trial, I noticed that many are of the nature such that a predictive model would not be needed to predict the severity of an accident (for example, an accident involving a pedestrian would likely cause an injury). This is comparable to an example presented in a previous IBM Data Science Course (example where the conclusion showed that larger house sizes correlated with higher price). The data features available which could be converted into categorical labels were limited, and a more in depth dataset could have been used. Also, it could be beneficial to create models involving the location of accidents, features such as the latitude and longitude location of the accident.

Conclusion

With a limited dataset, and “not so great” evaluation metrics for test datasets (see accuracy, Jaccard scores, F1 scores ect above), I would conclude that further exploration is needed in order to build a more reliable model. Additional machine learning algorithms could also be tested to determine if there are more effective ways of using machine learning to predict the severity of traffic accidents.