**Predictive Analysis – Severity of Car Accident**

A. Introduction

A.1. Description & Discussion of the Background

Problem: Many people in general are unaware about the "Impact of traffic" on the accident rate and the severity of those accidents, thereby making them vulnerable to such accidents. When someone ventures out on a road , he / she should be fully aware of the risks, challenges instead of going with assumptions.

Hence this project is intended to generate Supervised ML models to predict the "severity" of the accident based on different predictor variables/attributes.

Audience: Driving population in general, who can gain value by being aware on the potential risks while driving to a particular location on certain road, light , weather conditions, car speeding etc. This would enable them to take informed decisions and choose the route / direction accordingly. More so, these insights shall facilitate the people to be able to plan out their journey and tasks effectively thereby being able to achieve the intended agenda. Rather than being stuck up in an accident site and miss out on the opportunities.

B. Data Description

The dataset used for building this ML model is Data-Collisions.csv. The label of this dataset is "Severity" which is classified as "Injury Collision" or "Property Damage Only Collision".

Many attributes can be used to train the model. In this case, we use the following attributes/features for model training and evaluation.

Location, Weather Condition, Light conditions, Road conditions, Junction, People Count, Vehicle count. These features shall influence the predictability for the target (Severity).

In order not to create a biased ML model, we have balanced the data in this dataset. More so, we have done feature engineering in order to improve the predictability of the model (dropping off the irrelevant features, cleaning up empty observations etc.)

We have evaluated multiple Supervised ML algorithms and selected the one which offers the highest prediction accuracy. Details are laid out in the Methodology section.

Our objective is to build, train, test different models to predict the class of severity. Finally deploy the one with the highest Prediction Accuracy. Please refer to the next section for detailed methodology adopted.

C. Methodology

For my analysis and research, I used the following ML algorithms namely

1. K nearest neighbour.

2. Decision Tree

3. Logistic Regression

4. Support Vector Machines

For code repository and sharing, I used GitHub. The snapshot of the dataset looks like

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **LOCATION** | **SEVERITYCODE.1** | **COLLISIONTYPE** | **PERSONCOUNT** | **VEHCOUNT** | **JUNCTIONTYPE** | **WEATHER** | **ROADCOND** | **LIGHTCOND** |
|  |  |  |  |  |  |  |  |  |
| 5TH AVE NE AND NE 103RD ST | 2 | Angles | 2 | 2 | At Intersection (intersection related) | Overcast | Wet | Daylight |
| AURORA BR BETWEEN RAYE ST AND BRIDGE WAY N | 1 | Sideswipe | 2 | 2 | Mid-Block (not related to intersection) | Raining | Wet | Dark - Street Lights On |
| 4TH AVE BETWEEN SENECA ST AND UNIVERSITY ST | 1 | Parked Car | 4 | 3 | Mid-Block (not related to intersection) | Overcast | Dry | Daylight |

My master data has the following predictor variables such as Location, Collissiontype, PersonCount, Vehcount, JunctionType, Weather, RoadCond, LightCond. Since the SeverityCode is dependent on multiple dependent variables, hence it was necessary to drop other columns.

C.1. Data Normalisation

To run the analysis, I normalised the data by dropping out many irrelevant attributes from the data set, cleaned up the null/NaN entries, deleted redundancy and converted data types to float. Also to use the scikit-learn library, we have to convert the Pandas data frame to a Numpy array.

array([[0, 2, 2, 1, 4, 8, 5],

[9, 2, 2, 4, 6, 8, 2],

[5, 4, 3, 4, 4, 0, 5],

[4, 3, 3, 4, 1, 0, 5],

[0, 2, 2, 1, 6, 8, 5]])

The shape after normalisation was

(182660 rows, 9 cols)

For instance

COLLISIONTYPE SEVERITYCODE.1

Angles 1 0.606056

2 0.393944

Cycles 2 0.877193

1 0.122807

Head On 1 0.566416

2 0.433584

Left Turn 1 0.604312

2 0.395688

Other 1 0.737533

2 0.262467

Parked Car 1 0.938818

2 0.061182

Pedestrian 2 0.898261

1 0.101739

Rear Ended 1 0.568180

2 0.431820

Right Turn 1 0.793786

2 0.206214

Sideswipe 1 0.865004

2 0.134996

Name: SEVERITYCODE.1, dtype: float64

| **LOCATION** | **SEVERITYCODE.1** | **COLLISIONTYPE** |
| --- | --- | --- |
| **5TH AVE NE AND NE 103RD ST** | **2** | **0** |
| **AURORA BR BETWEEN RAYE ST AND BRIDGE WAY N** | **1** | **9** |
| **4TH AVE BETWEEN SENECA ST AND UNIVERSITY ST** | **1** | **5** |

Similarly JUNCTIONTYPE, WEATHER, ROADCOND, LIGHTCOND were too normalised.

SEVERITYCODE.1 int64

COLLISIONTYPE int64

PERSONCOUNT int64

VEHCOUNT int64

JUNCTIONTYPE int64

WEATHER int64

ROADCOND int64

LIGHTCOND int64

dtype: object

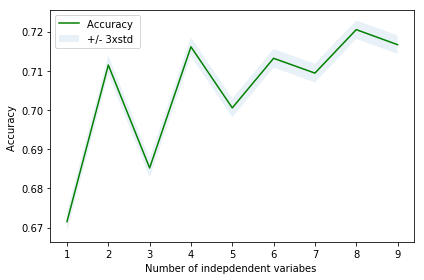
C.2. Building Supervised Learning classification models

The next important is to build , train and test the Supervised learning models and test those. Therefore the I split the dataset into a Training Set and a Test Set into 80-20 ratio respectively.

Which basically means that 80% of the normalised data set shall be used for training the models and rest 20% out of sample Test set shall be used for testing the model and accuracy.

**K-Nearest Neighbour (K-NN)**

Considering the feature space of “Location, Collissiontype, PersonCount, Vehcount, JunctionType, Weather, RoadCond, LightCond”, I tried to fit the Training data into this model to predict the accuracy. As you see, for different values of K, the prediction accuracy varies from 0.67 to 0.72. Basically, it depicts how closely the actual labels and the predicted labels are matched in the test set.



In this case, the best accuracy with this model is 0.720 with k=8.

**Decision Tree**

Similarly, using the same Feature space, the DecisionTrees's Accuracy: 0.740.

**Support Vector Machines (SVM)**

****

If we consider only Weather and ROADCOND, then the severity of the car

accident is just “Property damage only collision” regardless of the

Weather or Road conditions.

Also, using SVM

F1 score (accuracy)

0.6455598376907591

Jaccard Index for accuracy

0.7086116281616117

**Logistic Regression (LR)**

Using LR, the log loss is 0.5808543783835642.

The lower the log loss, the better the certainty. In this case, the probability of car accident happening considering the defined feature set is .58.

D. Results

The results in order of descending order of prediction accuracy

Decision Tree: 0.740

K-NN: 0.72

SVM: F1 score (0.64), Jaccard Index (0.70)

LR: 0.58 (log loss)

Taking into consideration the define data set, there is clear a high prediction of a severe car accident . However, this severity level is "Property damage only collission" and not an "Injury Collission".

F. Conclusion

According to the results, the Decision Tree Algorithm is most effective in predicting the accuracy of a car accident with highest severity. The K-NN and SVM models are stable too and point in the same direction with a similar error ratio as for Decision Tree. Hence the results are rationally approaching the actual value of the target.

It is recommended to deploy the Decision tree model for this specific case of Predictive analysis.

G. References:

* [1] <https://pandas.pydata.org/>