Road Mishap Risk Assessment

Instructor: Prof. Ralph Lano

Recent Trends and Technologies

(Results Summary)

Monday, 6 July 2020

Created By:

1. Fakkiragouda J. Patil

Matrikel-Nr: 00013920

2. Manas Ranjan Chhotray

Matrikel-Nr: 00015920

Conference List

1. DATA ANALYTICS 2020

Date	October 25th, 2020
Place	Nice, France
Deadline of Submission	July 20 2020
Website Link	https://www.iaria.org/conferences2020/CfPDATAANALYTICS20.html

2. International conference on Machine learning Big data management Cloud and Computing (ICMBDC)

Date	November 30th, 2020
Place	Mumbai, India
Deadline of Submission	November 13th, 2020
Website Link	http://asar.org.in/Conference/14547/ICMBDC/

3. DATA 2020

Date	July 7 to 9, 2020
Place	Portugal (Online)
Deadline of Submission	May 20 2020
Website Link	http://www.dataconference.org/ImportantInformation.aspx

4. International Conference on Big data, Machine Learning and IOT

Date	November 27th, 2020
Place	Bengaluru, India
Deadline of Submission	November 13th, 2020
Website Link	http://irfsr.com/Conference/428/ICBMI/

5. International Conference on Artificial Intelligence and Soft Computing

Date	January 3rd and 4th, 2021
Place	Munich, Germany
Deadline of Submission	December 18th, 2020
Website Link	http://www.academicsworld.org/Conference2021/Germany/1/ICAISC/

Model Building

In this research different machine learning algorithms are employed to predict accident severity level at different scenarios such as etc.

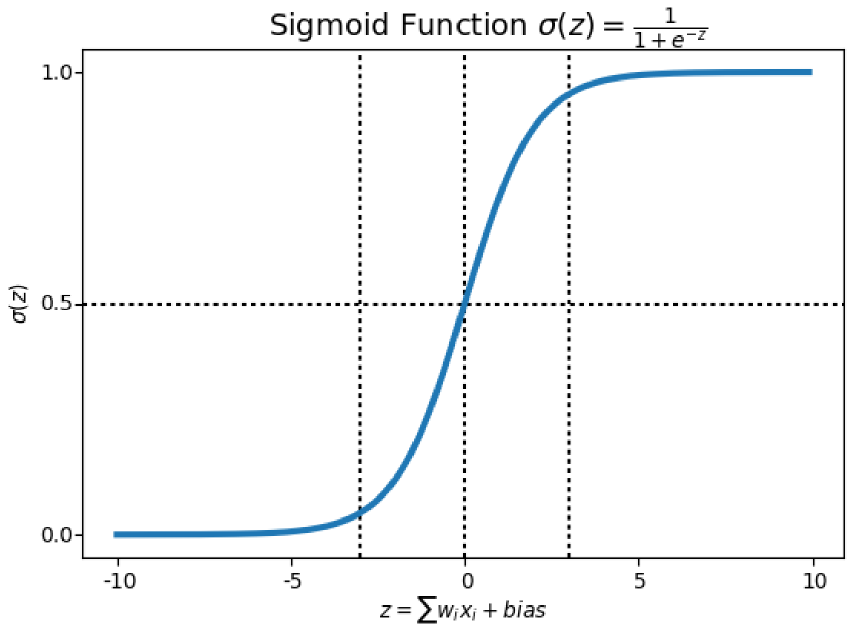
This classification problem will predict 3 output classes (1-Fatal,2-Serious and 3-Slight).

In this section different machine learning classifiers are briefly described which are tested and compared with each other as part of the research paper to predict accident severity level.

We Performed below activities as part Model Building for different algorithms:

1. Sampling the Data set
2. Splitting the Data samples into Training and Testing
3. Model with all features
4. Acquire the Feature Importance
5. Model with only the most Important Features
6. Hyper-Parameter Tuning
7. Performance Evaluation Table

Logistic Regression



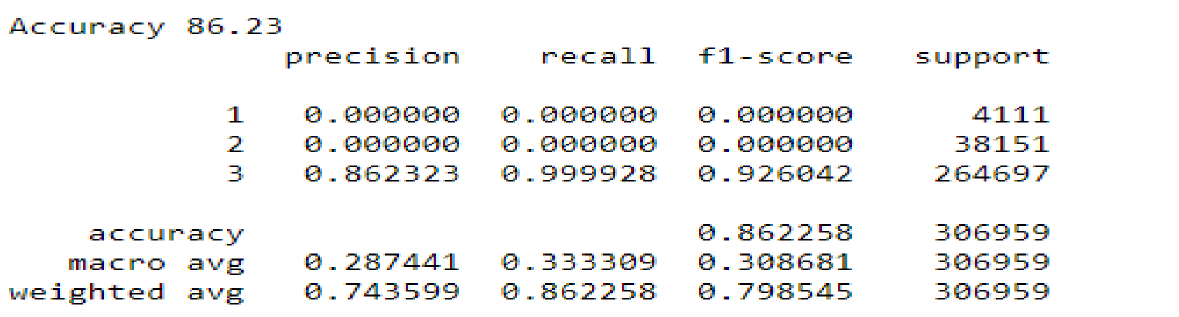
Logistic regression is a classification algorithm, used when the value of the target variable is categorical in nature. Logistic regression is most commonly used when the data in question has binary output, so when it belongs to one class or another, or is either a 0 or 1.

The Sigmoid Function:

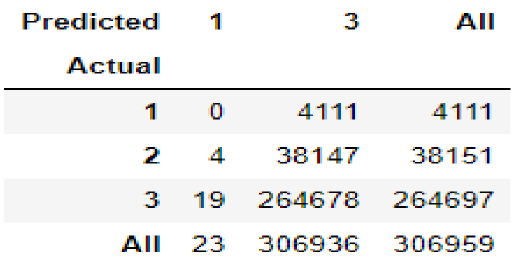
The sigmoid function/logistic function is a function that resembles an "S" shaped curve when plotted on a graph. It takes values between 0 and 1 and "squishes" them towards the margins at the top and bottom, labeling them as 0 or 1.The equation for the Sigmoid function is this:

$$y = 1/(1 + e^{-x})$$

Logistic Regression Data Experiment Results:

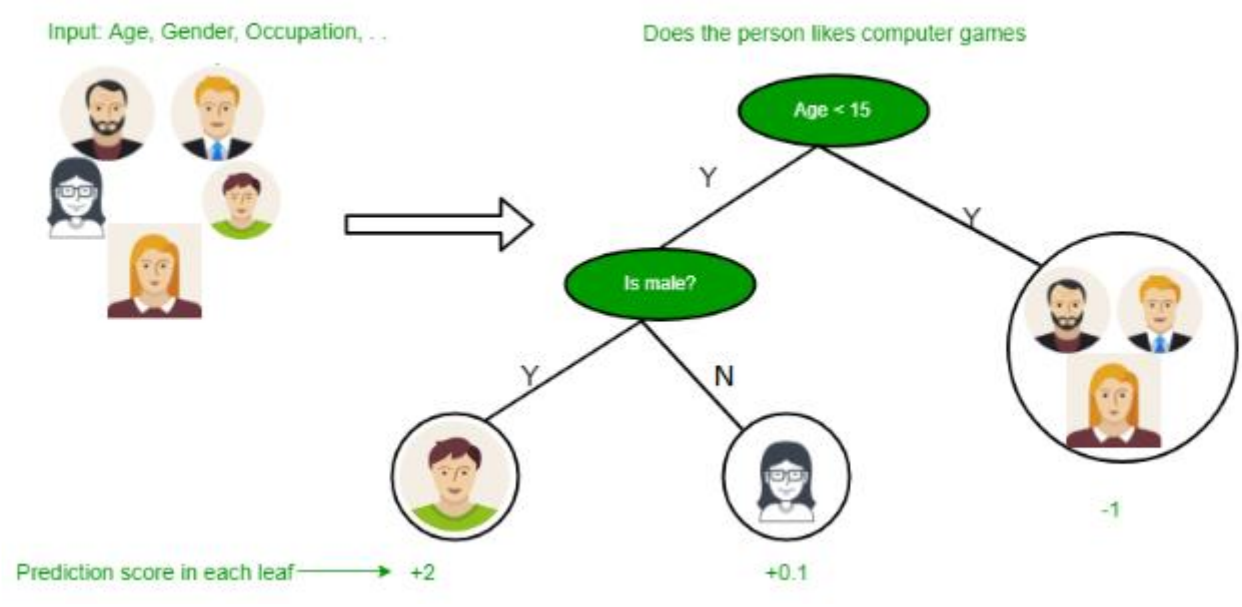


```
Accuracy 86.23
              precision    recall  f1-score   support

           1   0.000000  0.000000  0.000000      4111
           2   0.000000  0.000000  0.000000     38151
           3   0.862323  0.999928  0.926042    264697

    accuracy                       0.862258    306959
   macro avg   0.287441  0.333309  0.308681    306959
weighted avg   0.743599  0.862258  0.798545    306959
```



Predicted	1	3	All
Actual			
1	0	4111	4111
2	4	38147	38151
3	19	264678	264697
All	23	306936	306959

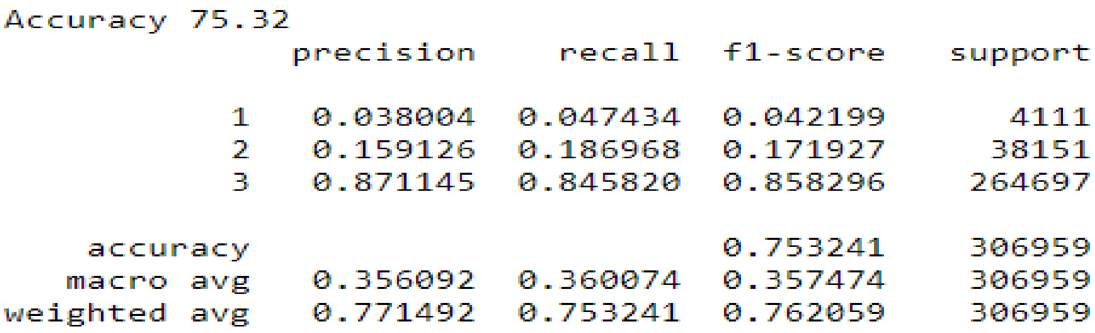
Decision Tree


Input: Age, Gender, Occupation, . .
Does the person likes computer games
Age < 15
Y
Y
Is male?
Y
N
Prediction score in each leaf
+2
+0.1
-1

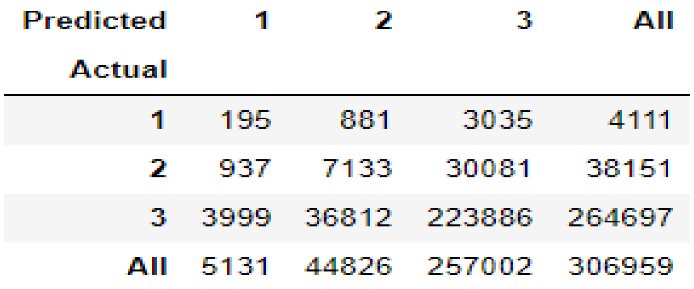
A decision tree is drawn upside down with its root at the top. In the image on the left, the green color oval shape represents a condition/**internal node**, based on which the tree splits into branches/ **edges**. The end of the branch that doesn't split anymore is the decision/**leaf**.

Although, a real dataset will have a lot more features and this will just be a branch in a much bigger tree, but you can't ignore the simplicity of this algorithm. The **feature importance is clear** and relations can be viewed easily. This methodology is more commonly known as **learning decision tree from data** and above tree is called **Classification tree** as the target is to classify does the person like Computer games or not. **Regression trees** are represented in the same manner, just they predict continuous values like price of a house. In general, Decision Tree algorithms are referred to as CART or Classification and Regression Trees.

Decision Tree Data Experiment results:



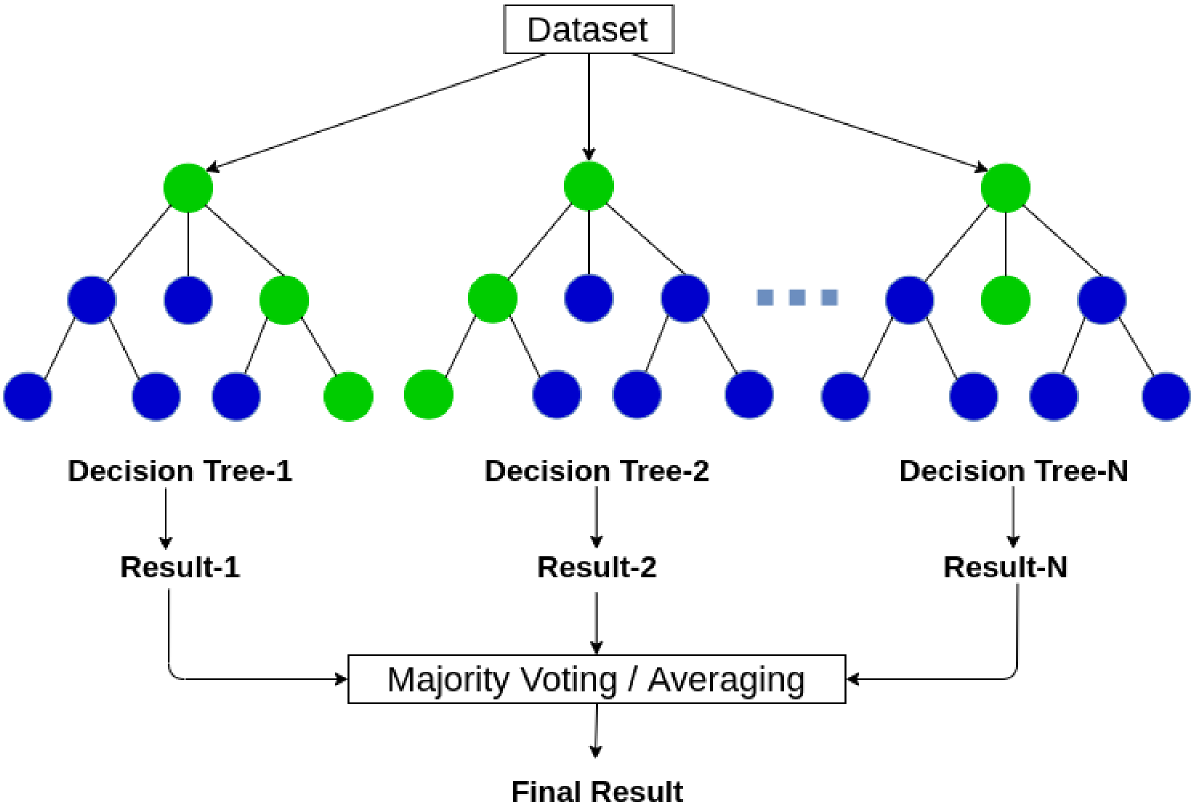
```
Accuracy 75.32
              precision    recall  f1-score   support

           1   0.038004  0.047434  0.042199      4111
           2   0.159126  0.186968  0.171927     38151
           3   0.871145  0.845820  0.858296    264697

    accuracy                       0.753241    306959
   macro avg   0.356092  0.360074  0.357474    306959
weighted avg   0.771492  0.753241  0.762059    306959
```



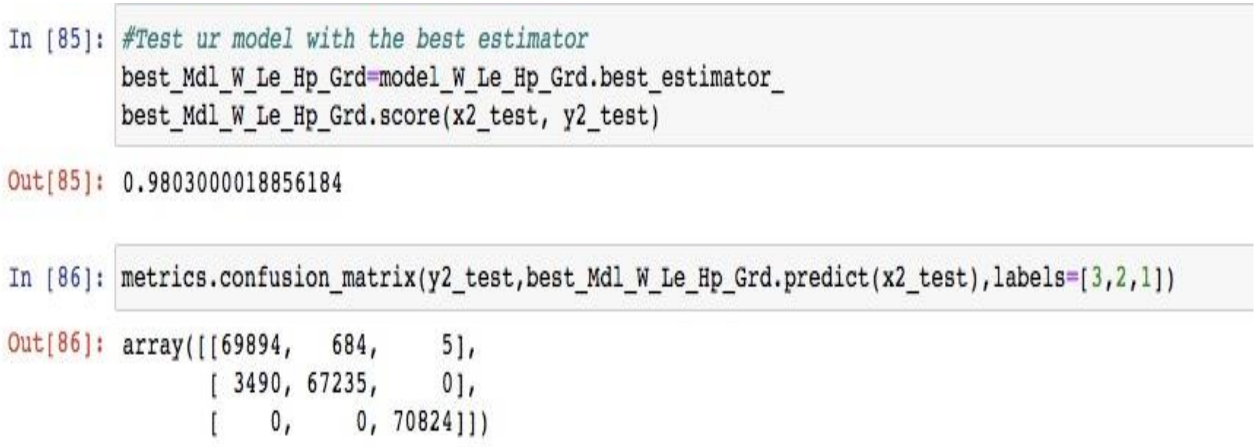
Predicted	1	2	3	All
Actual				
1	195	881	3035	4111
2	937	7133	30081	38151
3	3999	36812	223886	264697
All	5131	44826	257002	306959

Random Forest

Random Forest is a very useful algorithm for handling large data samples and can be used for both classification and regression. Bagging algorithms are used by RF's to create new training sets from the specific training set. It creates decision trees on random samples, gets a prediction from each tree and then gives the best prediction by voting. As the output comes from the votes of all the trees, Overfitting problem can be minimized. It normally gives a high accuracy in classification or prediction because a large number of trees give the final decision by voting. But the classification can be time consuming for a large sample because of the large number of trees. Figure below gives a brief visual idea of the working algorithms of a RF classifier where the prediction comes from voting of different training sets.



Random Forest Data Experiment results:



```
In [85]: #Test ur model with the best estimator
         best_Mdl_W_Le_Hp_Grd=model_W_Le_Hp_Grd.best_estimator_
         best_Mdl_W_Le_Hp_Grd.score(x2_test, y2_test)

Out[85]: 0.9803000018856184

In [86]: metrics.confusion_matrix(y2_test,best_Mdl_W_Le_Hp_Grd.predict(x2_test),labels=[3,2,1])

Out[86]: array([[69894,   684,     5],
                [ 3490, 67235,     0],
                [    0,     0, 70824]])
```