SUPERVISING LEARNING PROJECT – DATA ANALYSIS OF KSI COLLISIONS TORONTO

Presented to Prof. Merlin James Rukshan Dennis

Professor of Supervised Learning

ICET, Centennial College

Supervised Learning (SEC. 001)

Prepared by Group 7:

Ripudaman - 301221570

Karan Maria - 301238368

Mahpara Rafia Radmy - 301176893

Ronald Saenz Huerta - 301218602

Manipal Sidhu - 300859319

Aug 18, 2022

Table of Contents

[**Executive summary** 3](#_Toc111734832)

[Objective 3](#_Toc111734833)

[Key Findings 3](#_Toc111734834)

[**Overview of Solution** 4](#_Toc111734835)

[**Data exploration and findings** 5](#_Toc111734836)

[Tools and library 5](#_Toc111734837)

[Visualizations 5](#_Toc111734838)

[**Feature selection** 10](#_Toc111734839)

[Correlation with Target Class 10](#_Toc111734840)

[**Data modeling** 12](#_Toc111734841)

[**Model Building** 13](#_Toc111734842)

[Logistic regression 13](#_Toc111734843)

[Decision tree classifier 14](#_Toc111734844)

[Random forest classifier 15](#_Toc111734845)

[K Neighbors Classifier 16](#_Toc111734846)

[SVM 17](#_Toc111734847)

[Hard Voting Classifier 18](#_Toc111734848)

[Soft Voting Classifier 19](#_Toc111734849)

# **Executive summary**

## Objective

The main objective of this project was to predict the conditions responsible for a fatal accident.

The key stakeholder of the project will be the police and emergency or first responders who if supplied with better information on ideal conditions which leads to the accident could more effectively save lives.

There are various types of features present in the dataset like the physical and environmental condition of the accident location, geometric coordinate, collision vehicle, driver condition, time of day, and many more which can play an important role in defending the probability of survival.

## Key Findings

Most features do have much impact on the probability of survival.

1. The key features which have an impact are “AUTOMOBILE Collision”, “Pedestrian”, “TRUCK Collision” and “Initial Impact Type Pedestrian”.
2. The Total number of people who died is 16.86k while the total number of accidents recorded is 6002.
3. Overall, the count of aggressive collisions has reduced in recent years.

# **Overview of Solution**

After applying different algorithms soft voting classing and Random Forest performed the best getting an accuracy of 80%.

We found some columns like “**Injury**” type which are making are model overfit as that and the Target label was very same.

We also found the data set to be highly unbalanced.

***Accuracy with the unbalanced dataset:***

***All Data Accuracy:*** *0.8646377770426729*

***All Data Precision:*** *0.05207835642618251*

***All Data Recall:*** *0.6374269005847953*

***All Data ROC AUC Score:*** *0.7523322939754811*

*Show the balance data with target Class*

*0 13022*

*1 2093*

***Accuracy with Up-scaled dataset:***

***All Data Accuracy:*** *0.6719781907541085*

***All Data Precision:*** *0.6896022116418369*

***All Data Recall:*** *0.6661226911950152*

***All Data ROC AUC Score:*** *0.6721921264619507*

***Show the balance data with the target Class***

***0 13022***

***1 13022***

# **Data exploration and findings**

## Tools and library

We used different tools and libraries to do the data exploration and analysis. There are the some of them:

1. Python
2. Pandas
3. NumPy
4. Seaborn
5. PowerBI
6. Sklearn
7. Scipy
8. Git-Hub
9. Google Colab

## Visualizations

Graphical user interface, text

Description automatically generatedA picture containing graphical user interface

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Chart, pie chart

Description automatically generated Chart, pie chart

Description automatically generated

Chart, pie chart

Description automatically generated

A picture containing text, crossword puzzle

Description automatically generated A picture containing text, crossword puzzle

Description automatically generated

Timeline

Description automatically generated with medium confidence Chart

Description automatically generated

A picture containing graphical user interface

Description automatically generated

Chart

Description automatically generated

Chart

Description automatically generated

Map

Description automatically generated

Chart, bar chart

Description automatically generated

# **Feature selection**

## Correlation with Target Class

Chart

Description automatically generated with medium confidence

ACCLASS 1.000000

TRUCK 0.114711

PEDESTRIAN 0.100861

SPEEDING 0.089580

TRSN\_CITY\_VEH 0.048213

ALCOHOL 0.021518

HOOD\_ID 0.015462

POLICE\_DIVISION 0.007411

REDLIGHT -0.000108

month -0.001364

PASSENGER -0.003197

DISABILITY -0.004044

MOTORCYCLE -0.012923

EMERG\_VEH -0.015988

AG\_DRIV -0.029194

HOUR -0.037810

CYCLIST -0.078454

AUTOMOBILE -0.084198

Based on the correlation matrix the columnsTRUCK and PEDESTRIAN are most correlated to the probability of survival This makes sense as getting hit by something as big as a tuck will likely be more lethal and the same logic can be derived for pedestrians.

We removed features 'INDEX\_','ObjectId','ACCNUM', 'X', 'Y','STREET1', 'STREET2', 'LATITUDE', 'LONGITUDE' where Index and object are unique and thus not useful for prediction and coordinate and location data will be duplicate of other from a prediction standpoint.

We also assumed that columns with too many null values in them (more than 40%) will actually not have or negative effect on the model and thus dropped from the prediction model.

After having a group discussion, we decided to drop 'TIME', 'YEAR', 'DATE', 'WARDNUM', 'INITDIR', 'INVAGE', 'INJURY' where columns like Injury, Year, and so on we did not make our model generalize better and thus were not good for prediction as it was leading to overfitting.

# **Data modeling**

We did some steps to prepare the data for the model. There are some of them:

* + We check for Null values.
  + We check the data set if it’s balanced and found it to be highly unbalanced.
  + We extracted the months and years from the date to find seasonal trends.
  + Removed duplicate columns like neighborhood and Division.
  + We assumed that columns with very few values will not affect our model much.
  + They were too many classes in some columns which were grouped to improve performance.

For example, before to fit our data, we check that the data is not balanced, for that reason, we used ***resample*** library to balance using **up-sample minority class** and **down-sample minority class**.

Table

Description automatically generated

# **Model Building**

In total as a group, we tried 5 different models which are Logistic regression, Decision tree classifier, Random Forest classifier, SVM, and K Nearest neighbor. After that to improve the accuracy we used ensemble models like Hard Voting Classifier, and Soft Voting Classifier.

## Logistic regression

The best parameters found for Logistic regression are *max\_iter=1000, multi\_class='multinomial', random\_state=0* giving the scores as follows:

***Test Precision:*** *0.6658116526200073*

***Test Recall:*** *0.6864374763883642*

***Test F1 Score:*** *0.6759672619047619*

***Test ROC AUC Score:*** *0.6652327897164303*

***Test Accuracy Score =*** *0.6655788059128431*

***Test Confusion Matrix =***

*[[1650 912]*

*[ 830 1817]]*

***Test Classification Report =***

*precision recall f1-score support*

*0 0.67 0.64 0.65 2562*

*1 0.67 0.69 0.68 2647*

*accuracy 0.67 5209*

*macro avg 0.67 0.67 0.67 5209*

*weighted avg 0.67 0.67 0.67 5209*

Chart, treemap chart

Description automatically generated Chart, line chart

Description automatically generated

## Decision tree classifier

The best parameters found for Decision tree are *class\_weight='balanced', max\_depth=20, n\_estimators=250, random\_state=42* giving the scores as follows:

***Test Precision:*** *0.6730369754881596*

***Test Recall:*** *0.612013600302229*

***Test F1 Score:*** *0.6410763751483973*

***Test ROC AUC Score:*** *0.6524158555765633*

***Test Accuracy Score =*** *0.6517565751583797*

***Test Confusion Matrix =***

*[[1775 787]*

*[1027 1620]]*

***Test Classification Report =***

*precision recall f1-score support*

*0 0.63 0.69 0.66 2562*

*1 0.67 0.61 0.64 2647*

*accuracy 0.65 5209*

*macro avg 0.65 0.65 0.65 5209*

*weighted avg 0.65 0.65 0.65 5209*

Chart, treemap chart

Description automatically generated

Chart, line chart

Description automatically generated

## Random forest classifier

The best parameters found for Random Forest are *class\_weight='balanced', max\_depth=25, max\_leaf\_nodes=20, random\_state=42* giving the score as follows:

***Test Precision:*** *0.7871396895787139*

***Test Recall:*** *0.8432304038004751*

***Test F1 Score:*** *0.8142201834862386*

***Test ROC AUC Score:*** *0.8065072882311728*

***Test Accuracy Score =*** *0.8066825775656324*

***Test Confusion Matrix =***

*[[321 96]*

*[ 66 355]]*

***Test Classification Report =***

*precision recall f1-score support*

*0 0.83 0.77 0.80 417*

*1 0.79 0.84 0.81 421*

*accuracy 0.81 838*

*macro avg 0.81 0.81 0.81 838*

*weighted avg 0.81 0.81 0.81 838*

Chart

Description automatically generated

Chart

Description automatically generated

## K Neighbors Classifier

The best parameters found for K Neighbors Classifier are *n\_neighbors=2* giving the scores as follows:

***Test Precision:*** *0.7701149425287356*

***Test Recall:*** *0.6365795724465558*

***Test F1 Score:*** *0.6970091027308193*

***Test ROC AUC Score:*** *0.7223665248323906*

***Test Accuracy Score = 0.7219570405727923***

***Test Confusion Matrix =***

*[[337 80]*

*[153 268]]*

***Test Classification Report =***

*precision recall f1-score support*

*0 0.69 0.81 0.74 417*

*1 0.77 0.64 0.70 421*

*accuracy 0.72 838*

*macro avg 0.73 0.72 0.72 838*

*weighted avg 0.73 0.72 0.72 838*

Chart, treemap chart

Description automatically generated

Chart, line chart

Description automatically generated

## SVM

The best parameters found for SVM (SVC) are *gamma='auto', probability=True, random\_state=42* giving the scores as follows:

***Test Precision:*** *0.6277056277056277*

***Test Recall:*** *0.6888361045130641*

***Test F1 Score:*** *0.6568516421291052*

***Test ROC AUC Score:*** *0.6381830402661244*

***Test Accuracy Score = 0.6384248210023866***

***Test Confusion Matrix =***

***[[245 172]***

***[131 290]]***

***Test Classification Report =***

*precision recall f1-score support*

*0 0.65 0.59 0.62 417*

*1 0.63 0.69 0.66 421*

*accuracy 0.64 838*

*macro avg 0.64 0.64 0.64 838*

*weighted avg 0.64 0.64 0.64 838*

Chart, treemap chart

Description automatically generated

Chart, line chart

Description automatically generated

## Hard Voting Classifier

The best score for Hard Voting Classifier is as follows:

***Test Precision:*** *0.7314410480349345*

***Test Recall:*** *0.7957244655581948*

***Test F1 Score:*** *0.7622298065984072*

***Test ROC AUC Score:*** *0.7503802183906082*

***Test Accuracy Score*** *= 0.7505966587112172*

***Test Confusion Matrix*** *=*

*[[294 123]*

*[ 86 335]]*

***Test Classification Report*** *=*

*precision recall f1-score support*

*0 0.77 0.71 0.74 417*

*1 0.73 0.80 0.76 421*

*accuracy 0.75 838*

*macro avg 0.75 0.75 0.75 838*

*weighted avg 0.75 0.75 0.75 838*

Chart, treemap chart

Description automatically generated

## Soft Voting Classifier

The best score for Soft Voting Classifier is as follows:

***Test Precision****: 0.7782608695652173*

***Test Recall:*** *0.850356294536817*

***Test F1 Score:*** *0.8127128263337116*

***Test ROC AUC Score:*** *0.8028759889950272*

***Test Accuracy Score*** *= 0.8031026252983293*

***Test Confusion Matrix*** *=*

*[[315 102]*

*[ 63 358]]*

***Test Classification Report*** *=*

*precision recall f1-score support*

*0 0.83 0.76 0.79 417*

*1 0.78 0.85 0.81 421*

*accuracy 0.80 838*

*macro avg 0.81 0.80 0.80 838*

*weighted avg 0.81 0.80 0.80 838*

Chart, treemap chart

Description automatically generated

Chart

Description automatically generated