KSI dataset analysis

COMP 247 Supervised Learning

Prepared by

D Souza, Delicia Lisha

Umana Linares, Oscar

Yeung, Lok Ki

Table of Contents

Executive summary

This end-to-end machine learning project developed using different machine learning models trained with the KSI dataset is solution that provides accurate predictions for the occurrence fatal collisions in the city of Toronto. The report explains diferent stages of the project, starting with data collection and preprocessing, followed by model training and evaluation. The final stage involved deploying the models on a Flask application, which was then consumed by a React frontend.

The KSI dataset was obtained from the toronto police official website, which contains over 17,488 collision records. After cleaning and preprocessing the data, several machine learning models were trained, including Random Forest, Support vector Classifier, AdaBoosting, and K-nearest neighbors. Each model was evaluated using a variety of performance metrics, and the best models ere selected for deployment.

The Flask application was used to serve the machine learning models and provide a Restful API for the React frontend to consume. The front end was developed using React and integrated with the Flask API, allowing users to input collision data and receive predictions for the outcome of a collision occurring at a given location. The backend and frontend were deployed used Render and Netlify respectively.

Overall, the end-to-end machine learning project provides a powerful tool for predicting fatal traffic collisions outcomes in the city of Toronto, which can help authorities take preventative measures and reduce the number of accidents and also the public to take precautions during certain conditions.

Introduction

This report aims to estimate if an accident is fatal or no-fatal. The Killed or Seriously Injured (KSI) dataset is extracted from Toronto police department website. It contains real life records collected from 2006 to 2020.

Overview

This document provides an overview of the KSI (Killed or Seriously Injured) dataset, including its data types and null values. The dataset contains both numeric and categorical columns, which are analyzed and visualized. The ACCLASS column is particularly examined, and the case distribution, LATITUDE, LONGITUDE, DISTRICT, ROAD_CLASS, LOCCOORD, Environment, and Incident time columns are also investigated.

Moreover, the Accident Information section focuses on IMPACTYPE, INVAGE, VEHTYPE, INVTYPE, and the road users involved in the accident. In the Data preprocessing section, the data cleansing, imbalance data, feature importance, encoding categories columns, scale numerical columns, and the final column list are discussed.

Finally, the Building up models section outlines four models used for this dataset, including the SVC, Random Forest, KNN, and ADA models.

Libraries Used

The code utilizes several libraries including imbalanced-learn for handling class imbalance, pandas and numpy for data manipulation and analysis, joblib for saving and loading models, and scikit-learn for implementing the AdaBoost classifier with a decision tree base estimator. In addition, custom libraries such as transformers, DataPipeline, and Common are used for data preprocessing, feature engineering, and storing configuration variables. It also uses Rebalancer for rebalancing the target variable, and Config for storing configuration variables. The Preprocessing library from the transformers module is used for data preprocessing, such as encoding. The train_test_split function from the model_selection module in the scikit-learn library is used to split the data into training and testing sets. The DecisionTreeClassifier library from the tree module in the scikit-learn library is used for creating a decision tree classifier. The SMOTE library from the over_sampling module in the imblearn library is used for oversampling data to address class imbalance. The Config library from the Common module in the utils package is used for configuration. The GradientBoostingClassifier library from the ensemble module in the scikit-learn library is used for creating a gradient boosting classifier. The GridSearchCV library from the model_selection module in the scikit-learn library is used for performing a grid search to find the best hyperparameters for the model. The KNeighborsClassifier library from the neighbors module in the scikit-learn library is used for creating a k-nearest neighbors classifier.

Data exploration and findings

Dataset field descriptions

Data type and null values

There are 57 columns in the dataset, forty-six of them have an object data type, while 11 of them are either float64 or int64. The dataset contains 17,488 records, and some of the columns contain missing values.

#	Column	Non-Null Count	Dtype	Unique Count
0	X	17488	float64	4710
1	Y	17488	float64	4283
2	INDEX_	17488	int64	17488
3	ACCNUM	17488	int64	6263
4	YEAR	17488	int64	16
5	DATE	17488	object	3735
6	TIME	17488	int64	1317
7	STREET1	17488	object	1789
8	STREET2	15896	object	2638
9	OFFSET	3028	object	445
10	ROAD_CLASS	17112	object	11
11	DISTRICT	17453	object	5
12	WARDNUM	16963	float64	72
13	LATITUDE	17488	float64	4283
14	LONGITUDE	17488	float64	4710

15	LOCCOORD	17393	object	8
16	ACCLOC	12038	object	10
17	TRAFFCTL	17454	object	11
18	VISIBILITY	17470	object	9
19	LIGHT	17488	object	9
20	RDSFCOND	17465	object	10
21	ACCLASS	17481	object	4
22	IMPACTYPE	17484	object	11
23	INVTYPE	17474	object	20
24	INVAGE	17488	object	21
25	INJURY	15877	object	6
26	FATAL_NO	773	float64	79
27	INITDIR	12551	object	6
28	VEHTYPE	14455	object	31
29	MANOEUVER	10083	object	17
30	DRIVACT	8822	object	14
31	DRIVCOND	8819	object	11
32	PEDTYPE	2921	object	17
33	PEDACT	2957	object	16
34	PEDCOND	2958	object	11
35	CYCLISTYPE	739	object	23
36	CYCACT	730	object	12
37	CYCCOND	729	object	11

38	PEDESTRIAN	7079	object	2
39	CYCLIST	1827	object	2
40	AUTOMOBILE	15860	object	2
41	MOTORCYCLE	1482	object	2
42	TRUCK	1060	object	2
43	TRSN_CITY_VEH	1069	object	2
44	EMERG_VEH	43	object	2
45	PASSENGER	6621	object	2
46	SPEEDING	2441	object	2
47	AG_DRIV	9039	object	2
48	REDLIGHT	1453	object	2
49	ALCOHOL	762	object	2
50	DISABILITY	475	object	2
51	HOOD_158	17488	object	159
52	NEIGHBOURHOOD_158	17488	object	159
53	HOOD_140	17488	object	141
54	NEIGHBOURHOOD_140	17488	object	141
55	DIVISION	17488	object	17
56	ObjectId	17488	int64	17488

Numeric columns

- **X, Y, LATITUDE** and **LONGITUDE** store coordinate information. X and Y columns will be discarded as **LATITUDE** and **LONGITUDE** are available.

- **ObjectId, INDEX_** are record identifiers, **ACCNUM** is an accident identifier, **WARDNUM** is a Ward ID, and **FATAL_NO** is a sequential number. These columns will be discarded as they seem unrelated to the prediction.
- **YEAR** and **TIME** store accident time information.

Categorical columns

- The labeling column is **ACCLASS** which is a categorical column with three possible values ‘Fatal’, ‘Non-Fatal Injury’ and ‘Property Damage Only’.
- **PEDESTRIAN, CYCLIST, AUTOMOBILE, MOTORCYCLE, TRUCK, TRSN_CITY_VEH, EMERG_VEH, PASSENGER, SPEEDING, AG_DRIV, REDLIGHT, ALCOHOL, DISABILITY** are Boolean columns, which means the value is either ‘yes’ or null. The ‘yes’ value will be replaced by 1 and null will be replaced by 0.
- **ROAD_CLASS** stores the road classification. The possible values are ‘Major Arterial', 'Local', 'Minor Arterial', 'Collector', 'Other', 'Pending', 'Laneway', 'Expressway', 'Expressway Ramp' and 'Major Arterial Ramp'.
- **DISTRICT** stores the city district, the possible values are ‘Toronto and East York’, ‘Etobicoke York’, ‘Scarborough’ and ‘North York’.
- **LOCCOORD** stores Location Coordinate, the possible values are 'Intersection', 'Mid-Block', 'Exit Ramp Westbound', 'Exit Ramp Southbound', nan, 'Mid-Block (Abnormal)', 'Entrance Ramp Westbound' and 'Park, Private Property, Public Lane'
- **ACCLOC** stores the collision location, the possible values are 'Intersection Related', nan, 'At Intersection', 'Non Intersection', 'At/Near Private Drive', 'Overpass or Bridge', 'Underpass or Tunnel', 'Private Driveway', 'Trail' and 'Laneway'. This column will be discarded as more than 30% of the values are null.

- **TRAFFCTL** stores Traffic Control Type, the possible values are 'No Control', 'Stop Sign', 'Traffic Signal', 'Pedestrian Crossover', 'Traffic Controller', nan, 'Yield Sign', 'School Guard', 'Traffic Gate', 'Police Control' and 'Streetcar (Stop for)'.
- **VISIBILITY** stores Environment Condition, the possible values are 'Clear', 'Rain', 'Other', 'Snow', 'Strong wind', 'Freezing Rain', 'Fog, Mist, Smoke, Dust' and 'Drifting Snow'
- **LIGHT** stores Light Condition, the possible values are 'Dark', 'Dusk', 'Daylight', 'Dark, artificial', 'Dawn', 'Dusk, artificial', 'Dawn, artificial', 'Daylight, artificial' and 'Other'.
- **RDSFCOND** stores Road Surface Condition, the possible values are 'Wet', 'Dry', 'Other', 'Slush', 'Ice', 'Loose Snow', 'Packed Snow', 'Spilled liquid' and 'Loose Sand or Gravel'.
- **IMPACTYPE** stores Initial Impact Type, the possible values are 'Approaching', 'Other', 'Cyclist Collisions', 'Pedestrian Collisions', 'Rear End', 'Angle', 'SMV Other', 'Turning Movement', 'Sideswipe' and 'SMV Unattended Vehicle'.
- **INVTYPE** stores Involvement Type, the possible values are 'Passenger', 'Driver', 'Vehicle Owner', 'Cyclist', 'Pedestrian', 'Truck Driver', 'Other Property Owner', 'Motorcycle Driver', 'Motorcycle Passenger', nan, 'Other', 'In-Line Skater', 'Driver - Not Hit', 'Moped Driver', 'Wheelchair', 'Pedestrian - Not Hit', 'Trailer Owner', 'Witness', 'Cyclist Passenger' and 'Moped Passenger'.
- **INVAGE** stores Age of Involved Party, the possible values are '50 to 54', '15 to 19', '55 to 59', '20 to 24', 'unknown', '45 to 49', '30 to 34', '35 to 39', '25 to 29', '40 to 44', '70 to 74', '60 to 64', '10 to 14', '75 to 79', '5 to 9', '85 to 89', '0 to 4', '65 to 69', '80 to 84', 'Over 95' and '90 to 94'.
- **INJURY** stores Severity of Injury, the possible values are 'Major', 'Minor', 'None', 'Minimal' and 'Fatal'. The column will be discarded since it is highly correlated to **ACCLASS**.
- **INITDIR** stores Initial Direction of Travel, the possible values are 'North', 'South', 'East', 'West' and 'Unknown'. The column will be discarded as it seems unrelated to the **ACCLASS**.

- **VEHTYPE** stores Type of Vehicle, the possible values are 'Other', 'Automobile, Station Wagon', 'Bicycle', 'Municipal Transit Bus (TTC)', 'Truck - Open', 'Motorcycle', 'Taxi', 'Passenger Van', 'Delivery Van', 'Moped', 'Pick Up Truck', 'Police Vehicle', 'Truck-Tractor', 'Truck - Closed (Blazer, etc)', 'Street Car', 'Truck - Dump', 'Bus (Other) (Go Bus, Gray Coach)', 'Intercity Bus', 'Construction Equipment', 'Truck (other)', 'Truck - Tank', 'Off Road - 2 Wheels', 'Other Emergency Vehicle', 'Fire Vehicle', 'School Bus', 'Tow Truck', 'Truck - Car Carrier', 'Ambulance', 'Railway Train' and 'Off Road - 4 Wheels'.
- **MANOEUVER** stores Motorcyclist Involved in Collision, the possible values are 'Going Ahead', 'Turning Right', 'Stopped', 'Turning Left', 'Overtaking', 'Reversing', 'Other', 'Slowing or Stopping', 'Pulling Away from Shoulder or Curb', 'Parked', 'Changing Lanes', 'Pulling Onto Shoulder or towardCurb', 'Making U Turn', 'Unknown', 'Merging' and 'Disabled'. The column will be discarded as over 40% of the values are missing.
- **DRIVACT** stores Apparent Driver Action, the possible values are 'Driving Properly', 'Lost control', 'Failed to Yield Right of Way', 'Improper Passing', 'Improper Turn', 'Exceeding Speed Limit', 'Disobeyed Traffic Control', 'Following too Close', 'Other', 'Improper Lane Change', 'Wrong Way on One Way Road', 'Speed too Fast For Condition' and 'Speed too Slow'. The column will be discarded as half of the values are missing.
- **DRIVCOND** stores Driver Condition, the possible values are 'Normal', 'Ability Impaired, Alcohol Over .08', 'Unknown', 'Inattentive', 'Had Been Drinking', 'Medical or Physical Disability', 'Ability Impaired, Alcohol', 'Fatigue', 'Other' and 'Ability Impaired, Drugs'. The column will be discarded as half of the values are missing.
- **PEDTYPE** stores Pedestrian Crash Type, the possible values are 'Normal', 'Ability Impaired, Alcohol Over .08', 'Unknown', 'Inattentive', 'Had Been Drinking', 'Medical or Physical Disability', 'Ability Impaired, Alcohol', 'Fatigue', 'Other', 'Ability Impaired, Drugs''Pedestrian involved in a

collision with transit vehicle anywhere along roadway', 'Pedestrian hit at mid-block', 'Unknown', 'Vehicle is going straight thru inter.while ped cross without ROW', 'Pedestrian hit on sidewalk or shoulder', 'Vehicle is going straight thru inter.while ped cross with ROW', 'Vehicle turns left while ped crosses with ROW at inter.', 'Vehicle turns left while ped crosses without ROW at inter.', 'Vehicle turns right while ped crosses with ROW at inter.', 'Vehicle hits the pedestrian walking or running out from between parked vehicles at mid-block', 'Other / Undefined', 'Pedestrian hit a PXO/ped. Mid-block signal', 'Vehicle turns right while ped crosses without ROW at inter.', 'Vehicle is reversing and hits pedestrian' and 'Pedestrian hit at private driveway', 'Pedestrian hit at parking lot'. The column will be discarded as more than 80% of the values are missing.

- **PEDACT** stores Pedestrian Action, the possible values are 'Crossing, no Traffic Control', 'Running onto Roadway', 'Crossing without right of way', 'On Sidewalk or Shoulder', 'Crossing with right of way', 'Other', 'Coming From Behind Parked Vehicle', 'Walking on Roadway Against Traffic', 'Crossing, Pedestrian Crossover', 'Pushing/Working on Vehicle', 'Person Getting on/off Vehicle', 'Walking on Roadway with Traffic', 'Playing or Working on Highway', 'Crossing marked crosswalk without ROW' and 'Person Getting on/off School Bus'. The column will be discarded as more than 80% of the values are missing.
- **PEDCOND** stores Condition of Pedestrian, the possible values are 'Unknown', 'Normal', 'Other', 'Inattentive', 'Had Been Drinking', 'Medical or Physical Disability', 'Ability Impaired, Alcohol', 'Ability Impaired, Alcohol Over .80', 'Ability Impaired, Drugs' and 'Fatigue'. The column will be discarded as more than 80% of the values are missing.
- **CYCLISTYPE** stores Cyclist Crash Type, the possible values are 'Cyclist turns right across motorists path', 'Motorist turned left across cyclists path.', 'Cyclist without ROW rides into path of motorist at inter, lnwy, dwy-Cyclist not turn.', 'Motorist without ROW drives into path of cyclist

at inter, lnwy, dwy-Driver not turn.', 'Cyclist loses control and strikes object (pole, ttc track)', 'Cyclist rode off sidewalk into road at midblock.', 'Motorist turns right at non-signal Inter.(stop, yield, no cont.,and dwy) and strikes cyclist.', 'Cyclist struck opened vehicle door', 'Cyclist turned left across motorists path.', 'Cyclist and Driver travelling in same direction. One vehicle sideswipes the other.', 'Motorist turning right on green or amber at signalized intersection strikes cyclist.', 'Cyclist makes u-turn in-front of driver.', 'Cyclist and Driver travelling in same direction. One vehicle rear-ended the other.', 'Motorist turning right on red at signalized intersection strikes cyclist.', 'Cyclist strikes pedestrian.', 'Insufficient information (to determine cyclist crash type).', 'Motorist reversing struck cyclist.', 'Cyclist struck at PXO(cyclist either travel in same dir. as veh. or ride across xwalk)', 'Motorist loses control and strikes cyclist.', 'Cyclist strikes a parked vehicle.', 'Motorist makes u-turn in-front of cyclist.' and 'Cyclist falls off bike - no contact with motorist.'. The column will be discarded as more than 90% of the values are missing.

- **CYCACT** stores Cyclist Action, the possible values are 'Failed to Yield Right of Way', 'Driving Properly', 'Other', 'Lost control', 'Disobeyed Traffic Control', 'Improper Turn', 'Improper Passing', 'Improper Lane Change', 'Following too Close', 'Speed too Fast For Condition' and 'Wrong Way on One Way Road'. The column will be discarded as more than 90% of the values are missing.
- **CYCCOND** stores Cyclist Condition, the possible values are 'Inattentive', 'Normal', 'Unknown', 'Had Been Drinking', 'Ability Impaired, Drugs', 'Ability Impaired, Alcohol Over .80', 'Medical or Physical Disability', 'Ability Impaired, Alcohol', 'Other' and 'Fatigue'. The column will be discarded as more than 90% of the values are missing.
- **DIVISION** stores Police Division, the possible values are 'D55', 'D22', 'D41', 'D52', 'D32', 'D12', 'D11', 'D23', 'D33', 'D43', 'D13', 'D53', 'D14', 'D42', 'D51', 'D31' and 'NSA'. This column will be discarded as it seems unrelated to **ACCLASS**.

Others

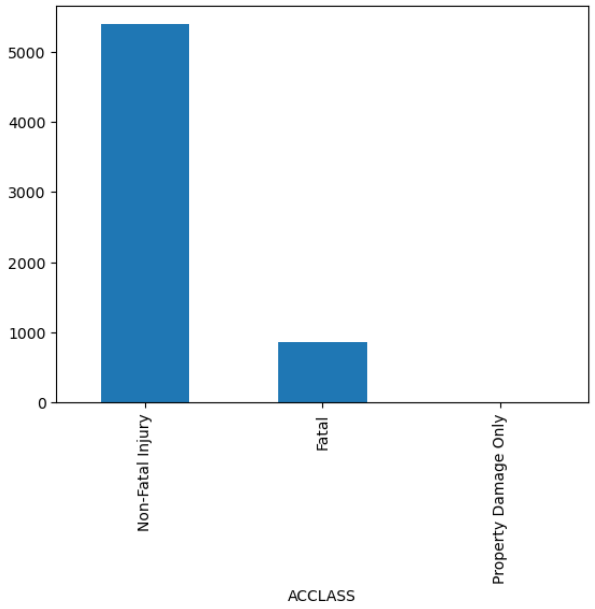
- **DATE** is the accident date. Day of week and Month information will be extracted from the column.
- **STREET1, STREET2, OFFSET, HOOD_158, NEIGHBOURHOOD_158, HOOD_140 and NEIGHBOURHOOD_140** stores location information. The columns will be discarded as it is correlated coordinate columns.

Visualization

ACCLASS

Since the number of fatal cases is much greater than that of non-fatal cases, the dataset is imbalanced.

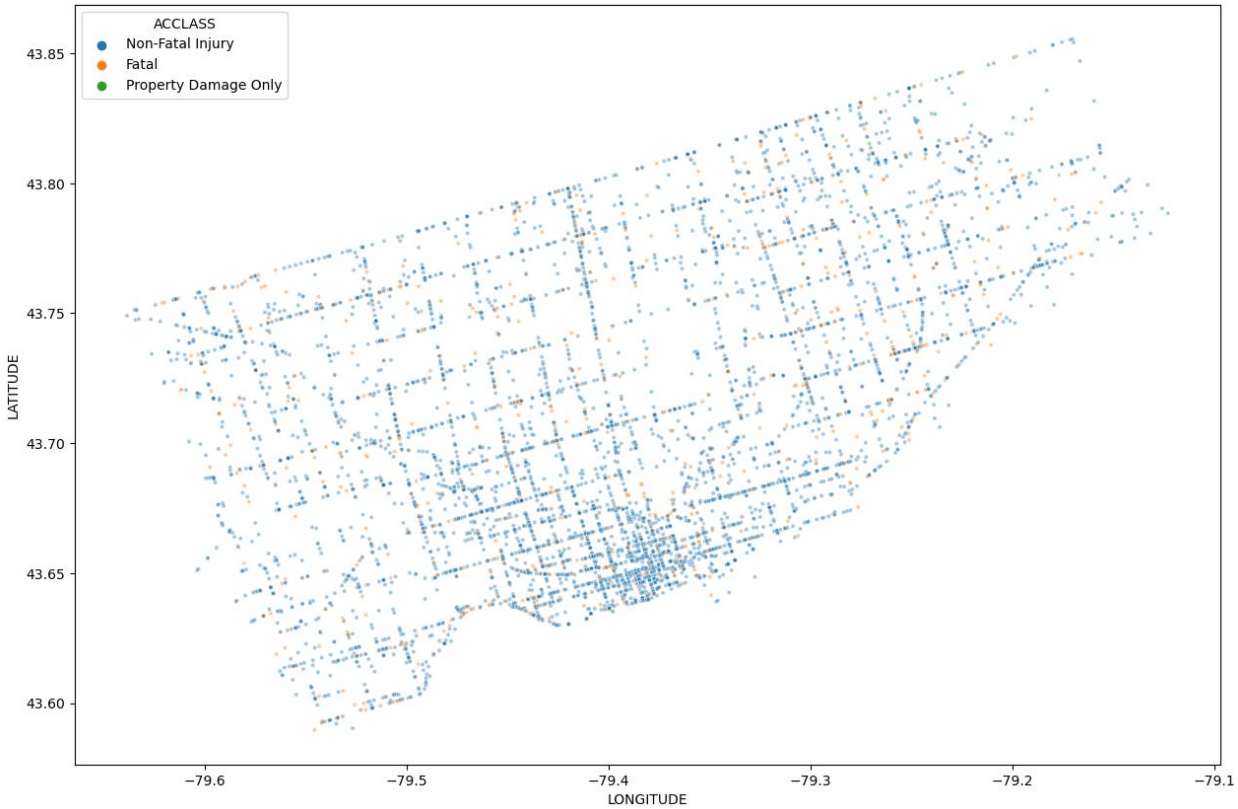
Frequency of values in **ACCLASS**



Case distribution

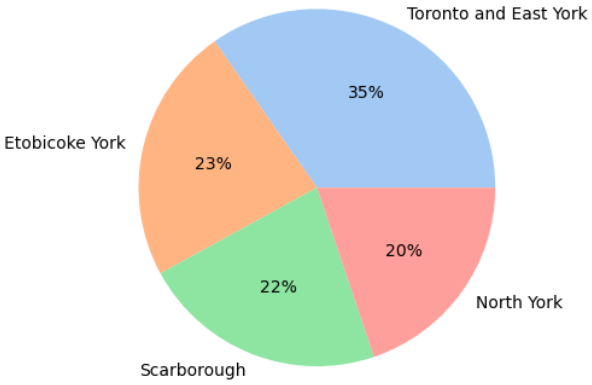
LATITUDE and LONGITUDE

More cases found in downtown Toronto.



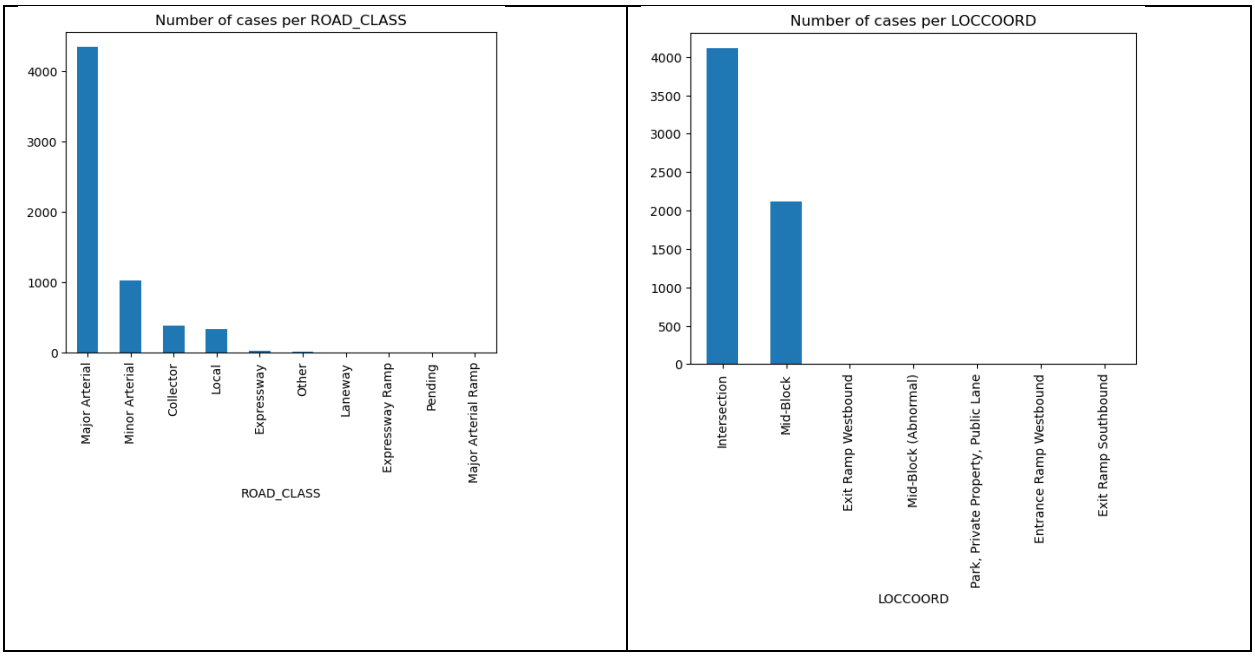
DISTRICT

1/3 of the cases occurs in Toronto and east york area



ROAD_CLASS and LOCCOORD

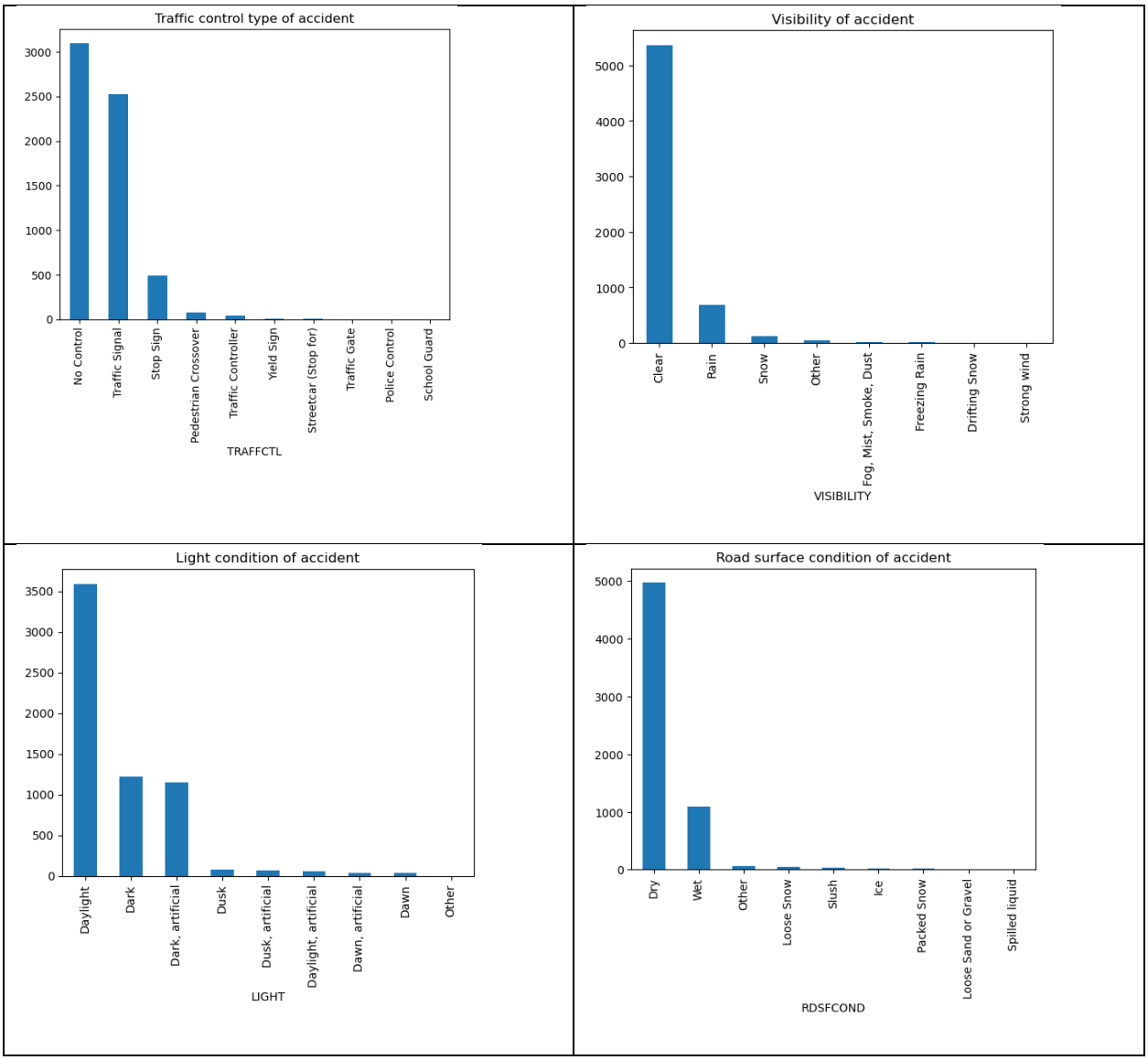
Most accidents occur at 'Major Arterial' and 'Intersection'



Environment

This session visualizes the number of cases that occur under different environmental conditions.

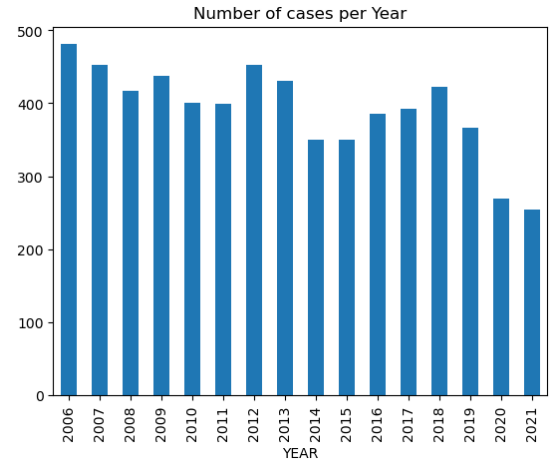
- TRAFFCTL: Traffic Control Type (Marjority are 'No Control' and 'Traffic Signal')
- VISIBILITY: Environment Condition (Marjority is 'Clear')
- LIGHT: Light Condition (Marjority is 'Daylight')
- RDSFCOND: Road Surface Condition (Marjority is 'Dry')

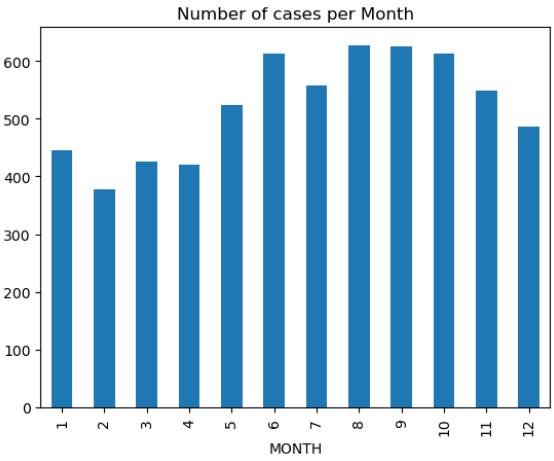


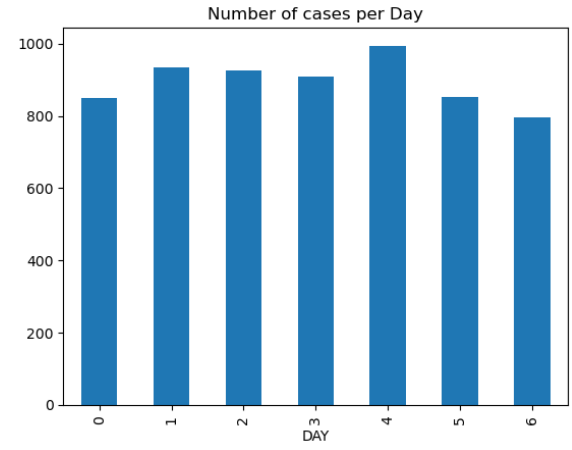
Incident time

This session visualizes the number of cases that occur at different time.

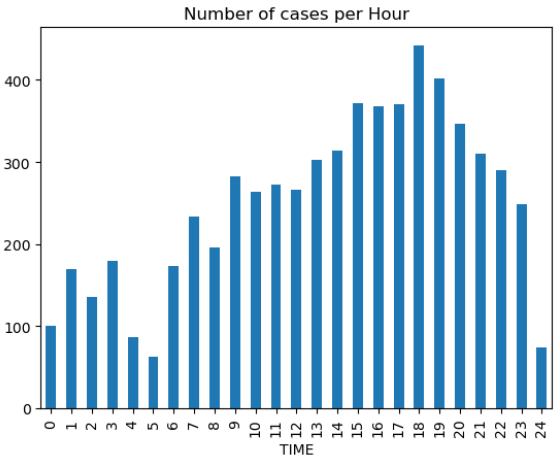
- The number of incidents decreases with year.
- There are less incidents in January, February, March, and April
- There are more incidents on Friday (4) and less incidents on Sunday (6)
- The number of incidents increase after 6 a.m. and peaks at 6 p.m.







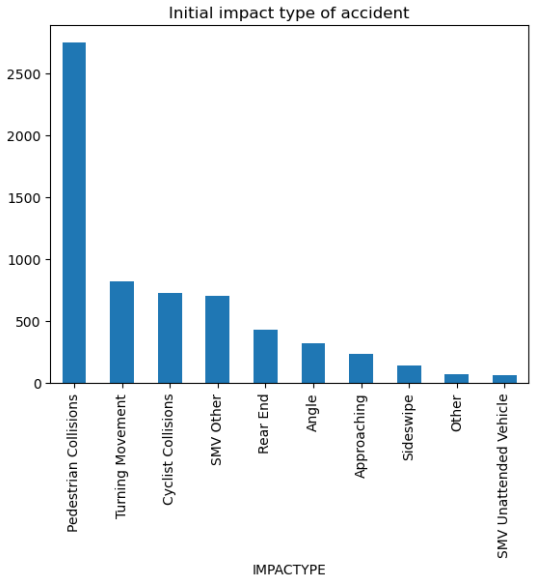
Day refers to week of day (0 for Monday and so on)



Accident Information

IMPACTYPE

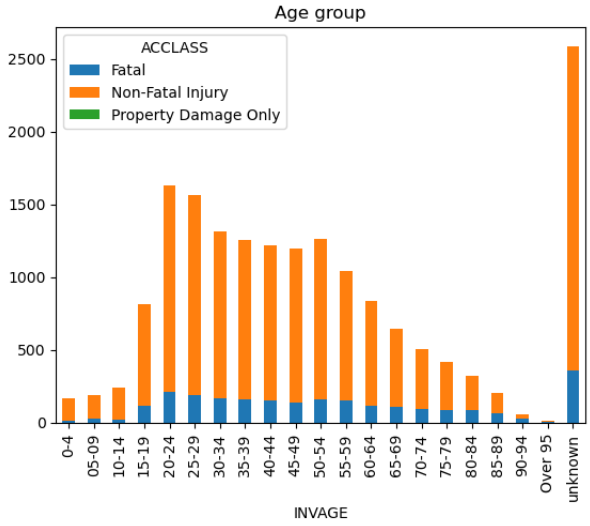
Number of cases per impact type. Top Initial Impact Type is Pedestrian.



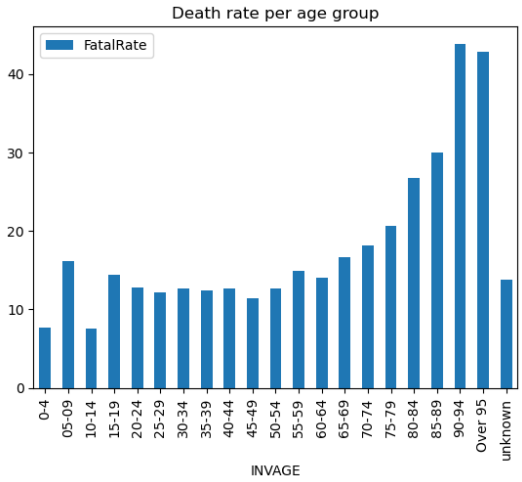
INVAGE

Age of Involved Party. As shown in the below diagram, many age values are missing and the death rate increase with age.

- Number of ‘Fatal’, ‘No-Fatal Injury’ and ‘Property Damage Only’ in each age group

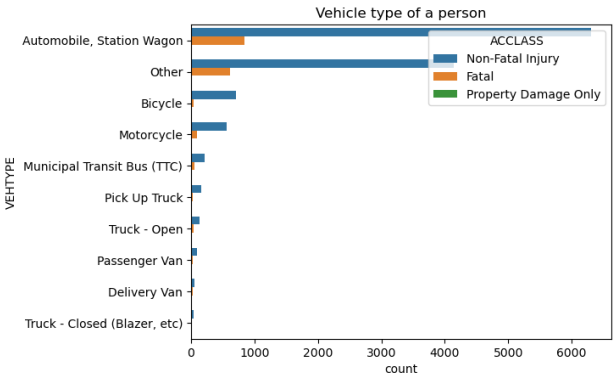


- Death ratio in each age group



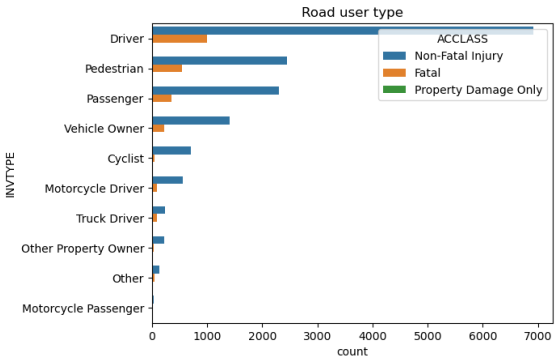
VEHTYPE

Type of Vehicle. As shown in the diagram, majority of the cases involve Automobile.



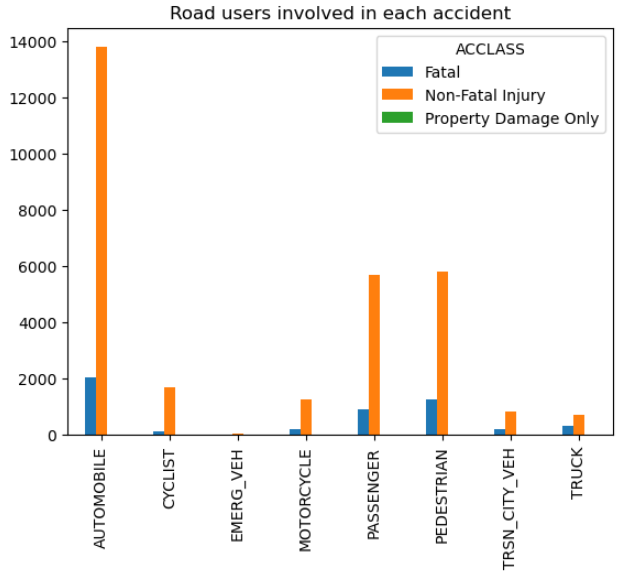
INVTYPE

Involvement Type. As shown in the diagram, most accidents involves driver, pedestrian, and passenger



Road users Involved in the accident

- Most cases involve automobile.
- If Truck involved in the accident, people involved are more likely to die.



Feature selection and Data modeling

Data cleansing

1. Remove irrelevant columns. i.e. the columns mentioned in **About KSI dataset** session.

X, Y, INDEX_, ACCNUM, STREET1, STREET2, OFFSET, WARDNUM, ACCLOC, INJURY, FATAL_NO, INITDIR, MANOEUVER, DRIVACT, DRIVCOND, PEDTYPE, PEDACT, PEDCOND, CYCLISTYPE, CYCACT, CYCCOND, HOOD_158, NEIGHBOURHOOD_158, HOOD_140, NEIGHBOURHOOD_140, DIVISION and **ObjectId**

2. Fill missing values.

   For **VEHTYPE** columns, the missing value will be filled by using the **VEHTYPE** of other records with same **ACCNUM**. For the numerical columns, SimpleImputer will be used to fill missing values with median.

3. Extract **Hour** from **TIME** column.
4. Extract **Month** and **DAY** (day of week) from **DATE** column.
5. Update Binary columns to indicate 'yes' by 1 and elsewise 0.
6. Update **ACCLASS** column to indicate Fatal case by 1 and elsewise, 0.

Imbalance data

SMOTE (Synthetic Minority Oversampling Technique) library will be used to handle imbalance data. The sampling_strategy will be set to minority to increase the size of the minority class.

Feature importance

RandomForestClassifier is used for finding out the feature importance. The **VISIBILITY, MOTORCYCLE, EMERG_VEH, REDLIGHT, ALCOHOL** and **DISABILITY** columns will be excluded as the percentage of importance is less than 1.

Row Labels	Sum of Importance	Percentage
INVAGE	0.21	20.58

INVTYPE	0.14	13.76
IMPACTYPE	0.09	8.83
LATITUDE	0.05	4.58
VEHTYPE	0.04	4.41
SPEEDING	0.04	3.75
TRUCK	0.04	3.51
YEAR	0.03	3.36
PEDESTRIAN	0.03	3.1
DAY	0.03	3.01
DISTRICT	0.03	2.93
LONGITUDE	0.03	2.76
TIME	0.03	2.65
AG_DRIV	0.03	2.6
LIGHT	0.03	2.57
ROAD CLASS	0.02	2.27
MONTH	0.02	2.14
TRAFFCTL	0.02	2.02
LOCCOORD	0.02	1.83
TOMOBILE	0.02	1.65
ASSENGER	0.02	1.62
RDSFCOND	0.01	1.32
TRSN_CITY_VEH	0.01	1.16
CYCLIST	0.01	1.1

VISIBILITY	0.01	0.82
REDLIGHT	0.01	0.71
ALCOHOL	0	0.37
MOTORCYCLE	0	0.34
DISABILITY	0	0.25
EMERG_VEH	0	0

Encoding categories columns

OneHotEncoder will be used to encode categories columns.

Scale numerical columns

StandardScaler library will be used to normalise the numerical columns.

Final column list

YEAR, DATE, TIME, ROAD_CLASS, DISTRICT, LATITUDE, LONGITUDE, TRAFFCTL, LIGHT, RDSFCOND, LOCCOORD, ACCLASS, IMPACTYPE, INVTYPE, INVAGE, VEHTYPE, PEDESTRIAN, CYCLIST, AUTOMOBILE, TRUCK, TRSN_CITY_VEH, PASSENGER, SPEEDING and AG_DRIV

Model building and fine tunning

ADA is the best model among all models tested in this report.

SVC

Library: sklearn.svm.SVC

Parameters

The following parameters are applied for SVC:

parameters	values
C	100
gamma	scale
kernel	rbf

C is a Regularization parameter. The default value is 1. The best parameter is 100 if given 1, 100, 200.

Kernel is the kernel type to be used in the algorithm. rbf performs good in terms of speed and accuracy in large dataset.

Performance measurement

The overall accuracy score of SVC is good. The precision and recall of predicting 0 is higher than that of 1, which means the classifier is good at predicting 0 but not good at predicting 1.

Training data

Accuracy

0.9997719844949456

Precision

0.9982993197278912

Recall

1

f1-socre

0.9991489361702128

classification_report

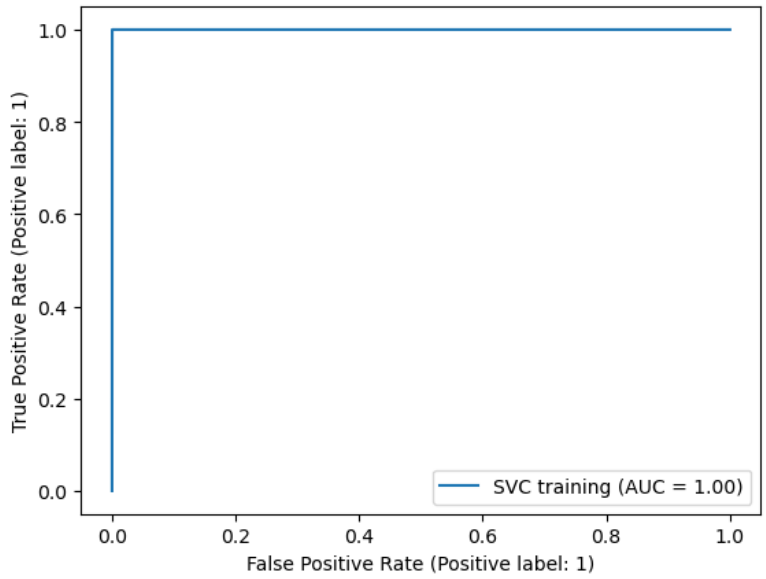
	precision	recall	f1-score	support
0	1	1	1	11396
1	1	1	1	1761
accuracy			1	13157
macro avg	1	1	1	13157
weighted avg	1	1	1	13157

Confusion matrix

[[11393 3]

[0 1761]]

ROC curve



Training data with SMOTE applied

Accuracy

0.9998683748683749

Precision

0.9997368190192122

Recall

1

f1-socre

0.99986839219127

classification_report

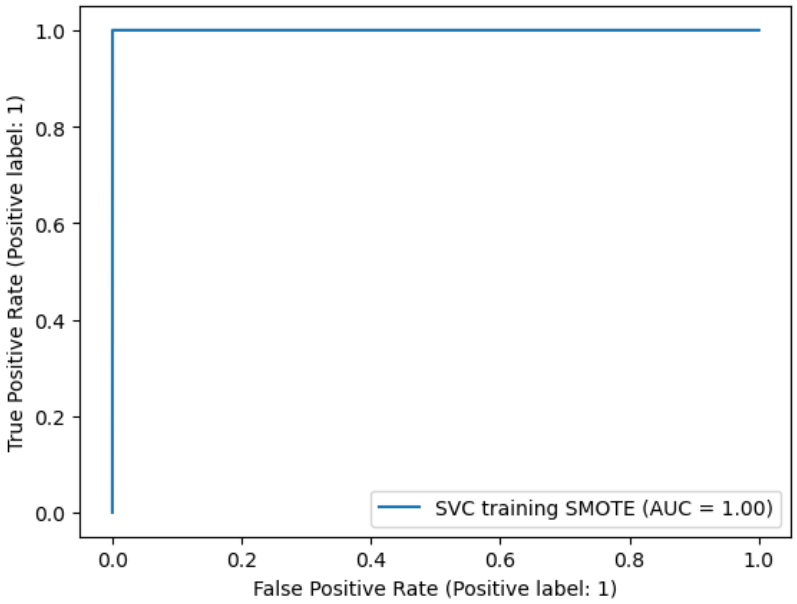
	precision	recall	f1-score	support
0	1	1	1	11396
1	1	1	1	11396
accuracy			1	22792
macro avg	1	1	1	22792
weighted avg	1	1	1	22792

Confusion matrix

[[11393 3]

[0 11396]]

ROC curve



Testing data

Accuracy

0.9109422492401216

Precision

0.6919060052219321

Recall

0.6022727272727273

f1-socre

0.643985419198056

classification_report

	precision	recall	f1-score	support
0	0.94	0.96	0.95	2850

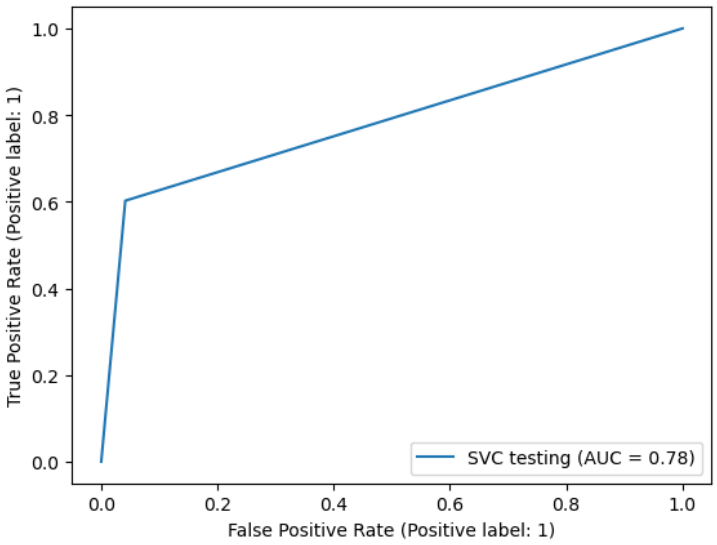
1	0.69	0.6	0.64	440
accuracy			0.91	3290
macro avg	0.82	0.78	0.8	3290
weighted avg	0.91	0.91	0.91	3290

Confusion matrix

[[2732 118]

 [175 265]]

ROC curve



Random Forest

Library: klearn.ensemble.RandomForestClassifier

Parameters

parameters	values
n_estimators	400
min_samples_split	35
min_samples_leaf	5
max_depth	18
algorithm	gini

For n_estimators, which indicate the number of trees in the forest, 400 is the best among 100, 200, 400, 600 and 1000.

For min_samples_split, 35 is the best among 5, 10, 15, 20, 25, 30, 35, 40 and 45.

For min_samples_leaf, 5 is the best among 1, 4, 7, 10, 13.

For max_depth, 18 is the best among 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 and 19.

For algorithm, gini is the best among gini and entropy.

Performance measurement

The accuracy of Random Forest is the worse among all four classifiers. The accuracy of predicting 0 is good while it is low for predicting 1. The recall of testing data is 0.3 for 1 which is relatively low. It means it is very likely to misclassify 1 as 0.

Training data

Accuracy

0.9135061184160523

Precision

0.7493995196156925

Recall

0.5315161839863713

f1-socre

0.6219269102990033

classification_report

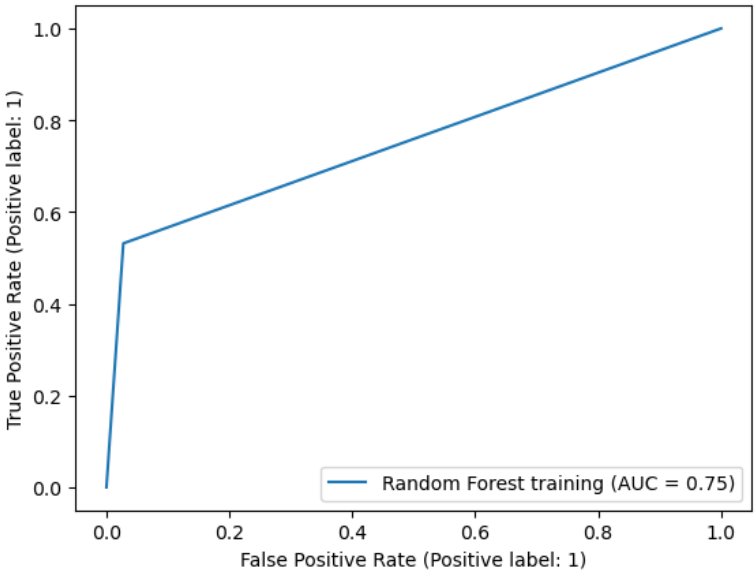
	precision	recall	f1-score	support
0	0.93	0.97	0.95	11396
1	0.75	0.54	0.62	1761
accuracy			0.91	13157
macro avg	0.84	0.75	0.79	13157
weighted avg	0.91	0.91	0.91	13157

Confusion matrix

[[11083 313]

[825 936]]

ROC curve



Training data with SMOTE applied

Accuracy

0.9454194454194454

Precision

0.9709593616626462

Recall

0.9183046683046683

f1-socre

0.9438982592225129

classification_report

	precision	recall	f1-score	support

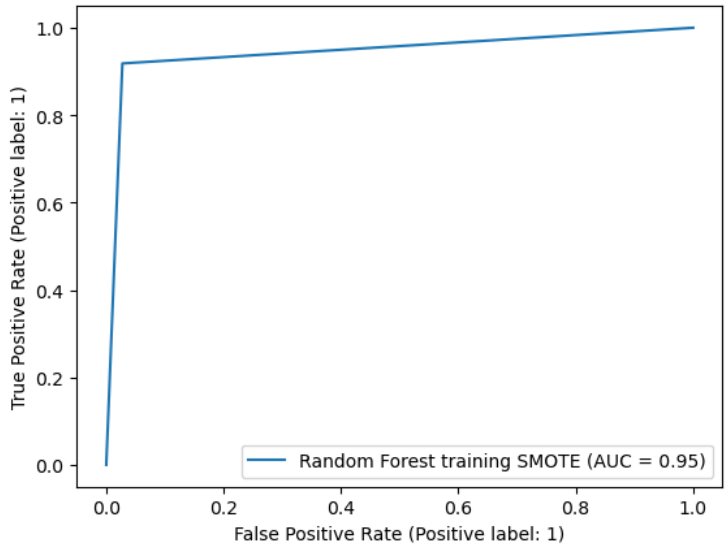
0	0.92	0.97	0.95	11396
1	0.97	0.92	0.94	11396
accuracy			0.95	22792
macro avg	0.95	0.95	0.95	22792
weighted avg	0.95	0.95	0.95	22792

Confusion matrix

[[11083 313]

 [931 10465]]

ROC curve



Testing data

Accuracy

0.8714285714285714

Precision

0.5333333333333333

Recall

0.3090909090909091

f1-socre

0.39136690647482014

classification_report

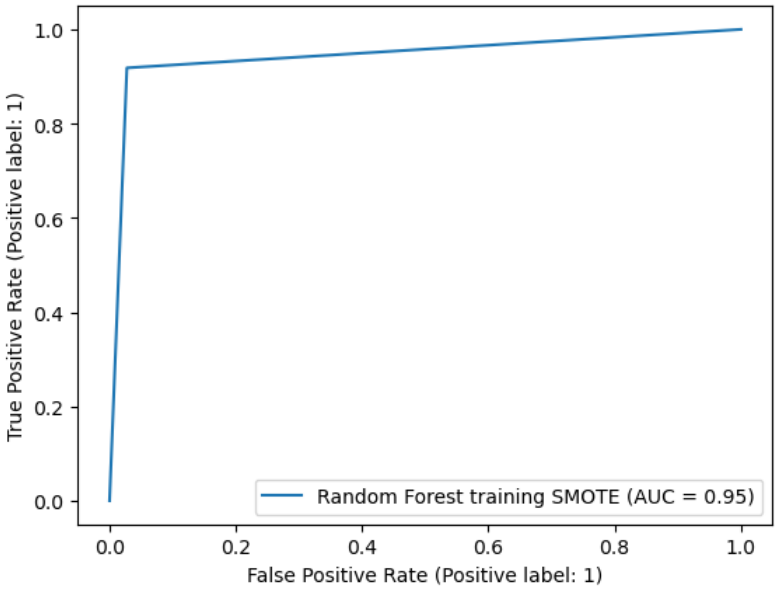
	precision	recall	f1-score	support
0	0.9	0.96	0.93	2850
1	0.53	0.31	0.38	440
accuracy			0.87	3290
macro avg	0.72	0.63	0.66	3290
weighted avg	0.85	0.87	0.85	3290

Confusion matrix

[[2731 119]

 [304 136]]

ROC curve



KNN

Library: sklearn.neighbors.KNeighborsClassifier

Parameters

parameters	values
weights	distance
metric	manhattan
n_neighbors	5
algorithm	kd_tree

For weights parameter, distance is the best out of uniform and distance. According to sklearn definition, Uniform means all points in each neighborhood are weighted equally, and distance means the classifier will weight points by the inverse of their distance.

For algorithms, kd_tree is better comparing to brute. Kd_tree performs better for larger dataset when compared to brute force.

For n_neighbors, 5 is the best out of 5, 9, 13 and 17.

Performance measurement

The overall accuracy score of KNN is good. The precision and recall of predicting 0 is higher than that of 1, which means the classifier is good at predicting 0 but not good at predicting 1. KNN's recall of predicting 1 is much better than that of SVC (0.6 for testing data), and thus KNN is a better model for predicting fatal cases.

Training data

Accuracy

1.0

Precision

1.0

Recall

1.0

f1-socre

1.0

classification_report

	precision	recall	f1-score	support
0	1	1	1	11396
1	1	1	1	1761

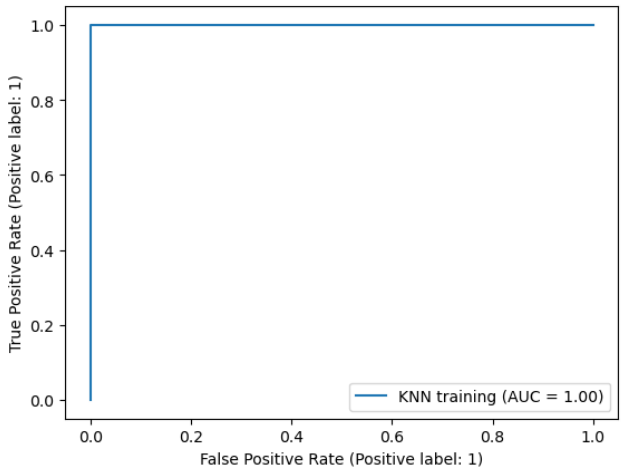
accuracy			1	13157
macro avg	1	1	1	13157
weighted avg	1	1	1	13157

Confusion matrix

[[11396 0]

[0 1761]]

ROC curve



Training data with SMOTE applied

Accuracy

1.0

Precision

1.0

Recall

1.0

f1-socre

1.0

classification_report

	precision	recall	f1-score	support
0	1	1	1	11396
1	1	1	1	11396
accuracy			1	22792
macro avg	1	1	1	22792
weighted avg	1	1	1	22792

Confusion matrix

[[11396 0]

[0 11396]]

ROC curve

Testing data

Accuracy

0.8951367781155015

Precision

0.5767366720516963

Recall

0.8113636363636364

f1-socre

0.6742209631728046

classification_report

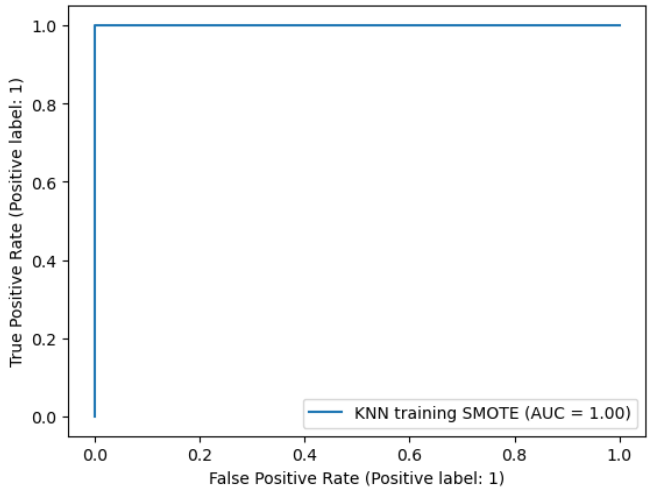
	precision	recall	f1-score	support
0	0.97	0.91	0.94	2850
1	0.58	0.81	0.67	440
accuracy			0.9	3290
macro avg	0.77	0.86	0.81	3290
weighted avg	0.92	0.9	0.9	3290

Confusion matrix

[[2588 262]

 [83 357]]

ROC curve



ADA

Library: sklean.ensemble.AdaBoostClassifier

Parameters

The following parameters are applied for AdaBoostClassier:

parameters	values
Base_estimator	DecisionTreeClassifier
n_estimators	50
learning_rate	1
algorithm	SAMME

The following parameters are applied for DecisionTreeClassifier

parameters	values

splitter	best
min_samples_split	45
min_samples_leaf	4
max_depth	19
criterion	Gini

Performance measurement

Accuracy score on training data dataset is perfect. Accuracy score on testing dataset is also good. The recall for 1 is 0.84 which is the highest among all other models.

Training data

Accuracy

1.0

Precision

1.0

Recall

1.0

f1-socre

1.0

classification_report

	precision	recall	f1-score	support
0	1	1	1	11396

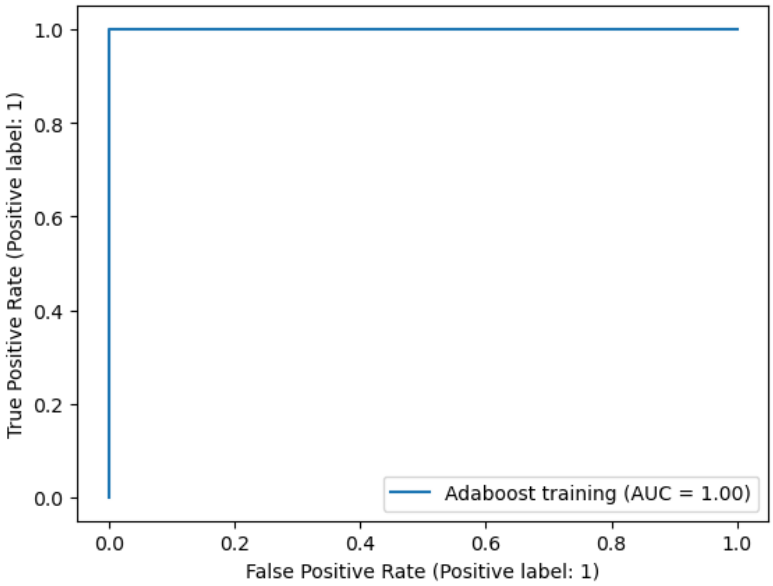
1	1	1	1	1761
accuracy			1	13157
macro avg	1	1	1	13157
weighted avg	1	1	1	13157

Confusion matrix

[[11396 0]

[0 1761]]

ROC curve



Training data with SMOTE applied

Accuracy

1.0

Precision

1.0

Recall

1.0

f1-socre

1.0

classification_report

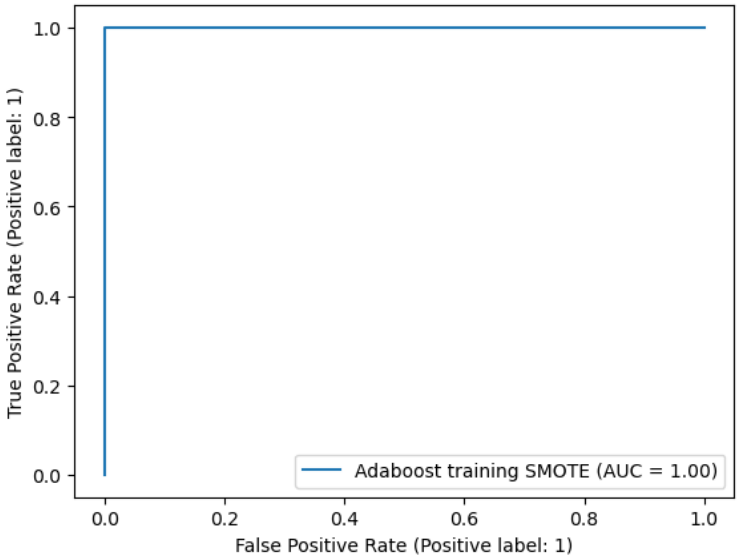
	precision	recall	f1-score	support
0	1	1	1	11396
1	1	1	1	11396
accuracy			1	22792
macro avg	1	1	1	22792
weighted avg	1	1	1	22792

Confusion matrix

[[11396 0]

[0 11396]]

ROC curve



Testing data

Accuracy

0.9775075987841946

Precision

0.9893048128342246

Recall

0.8409090909090909

f1-socre

0.9090909090909092

classification_report

	precision	recall	f1-score	support

0	0.98	1	0.99	2850
1	0.98	0.84	0.91	440
accuracy			0.98	3290
macro avg	0.98	0.92	0.95	3290
weighted avg	0.98	0.98	0.98	3290

Confusion matrix

[[2846 4]

[70 370]]

ROC curve