Research Paper on Road Accident of UK Traffic (1979-2019) DataSets Analysis Using Python Machine Learning Prediction

Conference Paper · January 2021

CITATION
1

READS
2,154

1 author:

Adnan Majeed
Computer Learning Center
15 PUBLICATIONS **26** CITATIONS

SEE PROFILE

Published: International Conference of Computer Science (Computer Learning Center Lahore Pakistan January 10th January 2021.

Research Paper on Road Accident of UK Traffic (1979-2019) DataSets Analysis Using Python Machine Learning Prediction

Data Cleaning Identify Missing Values Compare Results and produced best results and analysis

Author: Adnan Majeed (Master of Computer Science) Virtual University of Pakistan. Affiliation from Beaconhouse National University Lahore Pakistan, Lecturer at ILM Group of Colleges Lahore Pakistan. Unique Group of Colleges Lahore Pakistan Adnanmajeed82@gmail.com +92-300-4870104

Abstract

Traffic accident is main problem in UK area, but it is not reusable. Many people face many hurdles and problem in their daily lives. Somehow data is collected from UK Road accident and experiments and results performed using python data analytics. The objective of this report has been to performed machine learning algorithm. It was observed that using similar method in the literature study the best predicted model has been described in detail. The traffic forecast accidents has been a key issue for improving the transportation and public safety routes and their other link roads. The problem has occurred the proper class balanced and the space of heterogeneity is not provided. The relationship between independent variable and dependent variable is caused with each other. The recent study described the researcher performance and their data analytics is good to solve the problem. Forecast traffic.

This report has been critically analyzed the different researchers works and their different analysis in the same fields. This report work has been done through different researchers and different parts of the research group. In traffic accident in UK from 1979 to 2019 datasets the identifications of variables and their analysis is performed using machine learning algorithm.

Introduction

Traffic accident have been an important issue in UK for public health and safety. In 2001 to 2019 year the death toll rates of accident has been reached upto 1.50 million. The main feature to analyze the future accidents and their concerned matter have been analyzed with this datasets. Different security stakeholders such as police and different security officers are performed their duties but the travelers are idle to do their work.

Possibly the applications of real time safe route has been recommended by for the drivers by the security agencies. But with the rapid developed of data collection and different accident reports collected from the big cities of the UK is mainly deal with the concerned. In the last few years the traffic accident predictions has become more and more realistic, the rainfall data from the public transportation has been provided from the valuable information center.

Traffic accident analysis has been a very challenging issue, however some of the problems has been addressed and resolved with the safe public health precautions of road driving for drivers. The traffic accidents are a rare events but if we develop a strategy and labels them the accidents and their non-accident road events then it's avoidable. Machine learning algorithm of python of road accident has been tested very carefully. The populated cities of UK the traffic accident has been increasing on yearly basis. The best machine learning model has been provided in this report.
If the lower speed limit of the car then it's may be control in rural areas as well as urban cities.

In UK the low population density is high but the speed limit is also very high but the global model is not very accurate. The relationship between different factors of traffic accidents and their nonlinear complex performance cannot be realized.

Many years before many machine learning models and their methods has been tested data sets applied in accident datasets. But these machine learning model has been provide the facility to the Europe security agencies to control traffic accident and their other events to avoid null factors.
Different researches provide the traffic accident prediction and their datasets with limited functionality. This report introduces many exploration method, such as ARIMA model, time series model, and missing values detection model the check different angles of traffic accidents. Forecast model has been describe the binary classification problems.

Data

```
In [7]: mydata.columns

Out[7]: Index(['Accident_Index', 'Location_Easting_OSGR', 'Location_Northing_OSGR',
               'Longitude', 'Latitude', 'Police_Force', 'Accident_Severity',
               'Number_of_Vehicles', 'Number_of_Casualties', 'Date', 'Day_of_Week',
               'Time', 'Local_Authority_(District)', 'Local_Authority_(Highway)',
               '1st_Road_Class', '1st_Road_Number', 'Road_Type', 'Speed_limit',
               'Junction_Detail', 'Junction_Control', '2nd_Road_Class',
               '2nd_Road_Number', 'Pedestrian_Crossing-Human_Control',
               'Pedestrian_Crossing-Physical_Facilities', 'Light_Conditions',
               'Weather_Conditions', 'Road_Surface_Conditions',
               'Special_Conditions_at_Site', 'Carriageway_Hazards',
               'Urban_or_Rural_Area', 'Did_Police_Officer_Attend_Scene_of_Accident',
               'LSOA_of_Accident_Location'],
              dtype='object')
```

The data of datasets has described the column value , which described the data value and its type.

Libraries

```
import numpy as np
import pandas as pd
import seaborn as sns
from pylab import rcParams
from statsmodels.tsa.stattools import adfuller
from pandas import datetime
from pandas import read_csv
from pandas import DataFrame
from statsmodels.tsa.arima.model import ARIMA
import os
```

```
import random
mydata = pd.read_csv("Accident_DataSet.csv")
import matplotlib.pyplot as plt
```

In libraries section all python libraries has been included, such as numpy library and data frame libray and pandas library for visualizations.

Data Labeling

The machine learning algorithm we deal with the datasets that have contains the multiples labels and contain many columns. The labels describe the datasets more understandable for human readable.

```
mydata['Accident_Severity'].unique()
```

```
array([3, 2, 1], dtype=int64)
```

```
In [4]: mydata['Number_of_Vehicles'].unique()
```

```
Out[4]: array([ 2,  1,  4,  3,  5,  6, 13,  8,  7,  9, 12, 10, 11,
               18, 16, 14, 15, 25, 17, 29, 19, 20, 56, 61, 35, 21,
               26, 31, 27, 51, 49, 23, 24, 36, 59, 41, 47, 30, 33,
               43, 40, 38, 66, 192, 53, 73, 39, 75, 78, 44, 22, 82,
               34, 37, 88, 42, 32, 28, 67], dtype=int64)
```

```
In [5]: mydata['Number_of_Casualties'].unique()
```

```
Out[5]: array([ 1,  3,  2,  5,  4,  6,  7, 12, 10,  9, 16,  8, 25, 11, 33, 17, 14,
               45, 13, 15, 42, 22, 24, 29, 23, 20, 26, 18, 28, 35, 57, 39, 37, 43,
               36, 21, 44, 38, 34, 27, 60, 19, 56, 41, 52, 62, 32, 47, 50, 55, 40,
               51, 30, 63, 70, 53, 31, 61, 66, 80, 46, 48, 54, 75, 90, 79, 71, 67,
               68, 87, 93, 58, 59], dtype=int64)
```

```
In [6]: mydata['Light_Conditions'].unique()
```

```
Out[6]: array([ 1,  4,  6,  5, -1,  7], dtype=int64)
```

```
In [7]: mydata['Weather_Conditions'].unique()
```

```
Out[7]: array([ 8,  3,  2,  7,  9,  1,  4,  5,  6, -1], dtype=int64)
```

```
In [8]: mydata['Road_Surface_Conditions'].unique()
```

```
Out[8]: array([ 1,  3,  2, -1,  4,  5], dtype=int64)
```

Missing Value & Data Cleaning

We find missing value in the datasets of Accidents in which those value which contain Nan and NA value.

The datasets at the column, we can use the isull() method to fill pandas in the blanks with "NA" and "NaN", and we be able to check that missing values and "NA" are familiar as missing values and both of the Boolean responses are True.

```
In [7]: missing_values=["N/a", "na", np.nan]
        mydata = pd.read_csv("Accident_DataSet.csv")
```

```
C:\Users\Adnan\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3146: DtypeWarning: Columns (0,13,31) have mi
xed types.Specify dtype option on import or set low_memory=False.
  has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
```

```
In [8]: mydata
```

Out[8]:

	Accident_Index	Location_Easting_OSGR	Location_Northing_OSGR	Longitude	Latitude	Police_Force	Accident_Severity	Number_of_Vehicles	Num
0	197901A11AD14	NaN	NaN	NaN	NaN	1	3	2	
1	197901A1BAW34	198460.0	894000.0	NaN	NaN	1	3	1	
2	197901A1BFD77	406380.0	307000.0	NaN	NaN	1	3	2	
3	197901A1BGC20	281680.0	440000.0	NaN	NaN	1	3	2	
4	197901A1BGF95	153960.0	795000.0	NaN	NaN	1	2	2	
...	...	...	...	...	...	...	...	...	
8511620	2019984106919	312635.0	573392.0	-3.368899	55.047323	98	3	1	
8511621	2019984107019	337522.0	591682.0	-2.983499	55.215407	98	3	4	
8511622	2019984107219	318544.0	567087.0	-3.274645	54.991685	98	3	2	
8511623	2019984107419	336525.0	584226.0	-2.997491	55.148292	98	3	1	
8511624	201998QC01004	291367.0	608364.0	-3.715064	55.357237	98	2	1	

```
In [9]: mydata.isnull().sum()
```

Out[9]:

```
Accident_Index                                  0
Location_Easting_OSGR                       10263
Location_Northing_OSGR                      10263
Longitude                                 4887388
Latitude                                  4887388
Police_Force                                    0
Accident_Severity                               0
Number_of_Vehicles                              0
Number_of_Casualties                            0
Date                                            8
Day_of_Week                                     0
Time                                          923
Local_Authority_(District)                      0
Local_Authority_(Highway)                       0
1st_Road_Class                                  0
1st_Road_Number                                 0
Road_Type                                       0
Speed_limit                                    37
Junction_Detail                                 0
Junction_Control                                0
2nd_Road_Class                                  0
2nd_Road_Number                                 0
Pedestrian_Crossing-Human_Control               0
Pedestrian_Crossing-Physical_Facilities         0
Light_Conditions                                0
Weather_Conditions                              0
Road_Surface_Conditions                         0
Special_Conditions_at_Site                      0
Carriageway_Hazards                             0
Urban_or_Rural_Area                             0
Did_Police_Officer_Attend_Scene_of_Accident     0
LSOA_of_Accident_Location                 5190678
dtype: int64
```

```
In [10]: mydata.isnull().any
```

```
Out[10]: <bound method DataFrame.any of          Accident_Index  Location_Easting_OSGR  Location_Northing_OSGR  \
0                 False                   True                    True
1                 False                  False                   False
2                 False                  False                   False
3                 False                  False                   False
4                 False                  False                   False
...                 ...                    ...                     ...
8511620           False                  False                   False
8511621           False                  False                   False
8511622           False                  False                   False
8511623           False                  False                   False
8511624           False                  False                   False

         Longitude  Latitude  Police_Force  Accident_Severity  \
0             True      True         False              False
1             True      True         False              False
2             True      True         False              False
3             True      True         False              False
4             True      True         False              False
```

```
<bound method DataFrame.any of          Accident_Index  Location_Easting_OSGR
Location_Northing_OSGR  \
0                 False                   True                    True
1                 False                  False                   False
2                 False                  False                   False
3                 False                  False                   False
4                 False                  False                   False
...                 ...                    ...                     ...
8511620           False                  False                   False
8511621           False                  False                   False
8511622           False                  False                   False
8511623           False                  False                   False
8511624           False                  False                   False

         Longitude  Latitude  Police_Force  Accident_Severity  \
0             True      True         False              False
1             True      True         False              False
2             True      True         False              False
3             True      True         False              False
4             True      True         False              False
...            ...       ...           ...                ...
8511620      False     False         False              False
8511621      False     False         False              False
8511622      False     False         False              False
8511623      False     False         False              False
8511624      False     False         False              False

         Number_of_Vehicles  Number_of_Casualties   Date  ...  \
0                     False                 False  False  ...
1                     False                 False  False  ...
2                     False                 False  False  ...
3                     False                 False  False  ...
4                     False                 False  False  ...
...                     ...                   ...    ...  ...
8511620               False                 False  False  ...
8511621               False                 False  False  ...
8511622               False                 False  False  ...
8511623               False                 False  False  ...
8511624               False                 False  False  ...

         Pedestrian_Crossing-Human_Control  \
```

```
0                                   False
1                                   False
2                                   False
3                                   False
4                                   False
...                                   ...
8511620                             False
8511621                             False
8511622                             False
8511623                             False
8511624                             False

         Pedestrian_Crossing-Physical_Facilities  Light_Conditions  \
0                                          False             False
1                                          False             False
2                                          False             False
3                                          False             False
4                                          False             False
...                                          ...               ...
8511620                                    False             False
8511621                                    False             False
8511622                                    False             False
8511623                                    False             False
8511624                                    False             False

         Weather_Conditions  Road_Surface_Conditions  \
0                     False                    False
1                     False                    False
2                     False                    False
3                     False                    False
4                     False                    False
...                     ...                      ...
8511620               False                    False
8511621               False                    False
8511622               False                    False
8511623               False                    False
8511624               False                    False

         Special_Conditions_at_Site  Carriageway_Hazards  Urban_or_Rural_Area
\
0                             False                False                False
1                             False                False                False
2                             False                False                False
3                             False                False                False
4                             False                False                False
...                             ...                  ...                  ...
8511620                       False                False                False
8511621                       False                False                False
8511622                       False                False                False
8511623                       False                False                False
8511624                       False                False                False

         Did_Police_Officer_Attend_Scene_of_Accident  \
0                                              False
1                                              False
2                                              False
3                                              False
```

```
4                                                    False
...                                                    ...
8511620                                              False
8511621                                              False
8511622                                              False
8511623                                              False
8511624                                              False

         LSOA_of_Accident_Location
0                             True
1                             True
2                             True
3                             True
4                             True
...                            ...
8511620                       True
8511621                       True
8511622                       True
8511623                       True
8511624                       True
```

```
In [14]: mydata['Longitude'].isnull()

Out[14]: 0           True
         1           True
         2           True
         3           True
         4           True
                    ...
         8511620    False
         8511621    False
         8511622    False
         8511623    False
         8511624    False
         Name: Longitude, Length: 8511625, dtype: bool
```

```
In [13]: mydata['Latitude'].isnull()

Out[13]: 0           True
         1           True
         2           True
         3           True
         4           True
                    ...
         8511620    False
         8511621    False
         8511622    False
         8511623    False
         8511624    False
         Name: Latitude, Length: 8511625, dtype: bool
```

```
In [15]: mydata['LSOA_of_Accident_Location'].isnull()

Out[15]: 0          True
         1          True
         2          True
         3          True
         4          True
                    ...
         8511620    True
         8511621    True
         8511622    True
         8511623    True
         8511624    True
         Name: LSOA_of_Accident_Location, Length: 8511625, dtype: bool
```

```
In [16]: mydata['Location_Northing_OSGR'].isnull()

Out[16]: 0           True
         1          False
         2          False
         3          False
         4          False
                    ...
         8511620    False
         8511621    False
         8511622    False
         8511623    False
         8511624    False
         Name: Location_Northing_OSGR, Length: 8511625, dtype: bool
```

```
In [23]: mydata_drop.shape

Out[23]: (3320599, 32)
```

```
In [24]: mydata_drop.head()
```

Out[24]:

	Accident_Index	Location_Easting_OSGR	Location_Northing_OSGR	Longitude	Latitude	Police_Force	Accident_Severity	Number_of_Vehicles	Num
4883216	1999010SU0945	519490.0	203300.0	-0.271752	51.715661	1	3	1	
4883217	1999010SU0946	521740.0	201070.0	-0.239977	51.695136	1	3	2	
4883218	1999010SU0947	519610.0	203240.0	-0.270037	51.715096	1	3	2	
4883219	1999010SU0948	520090.0	202830.0	-0.263233	51.711309	1	2	2	
4883220	1999010SU0949	522640.0	200320.0	-0.227225	51.688200	1	3	4	

5 rows × 32 columns

```
In [26]: mydata_drop_with_condition = mydata.dropna(thresh=2)

In [27]: mydata_drop_with_condition.shape
Out[27]: (8511625, 32)

In [28]: mydata.shape
Out[28]: (8511625, 32)

In [29]: mydata_drop_column = mydata.dropna(axis=1)

In [31]: mydata.shape
Out[31]: (8511625, 32)

In [33]: mydata.duplicated()
Out[33]: 0          False
         1          False
         2          False
         3          False
         4          False
                    ...
         8511620    False
         8511621    False
         8511622    False
         8511623    False
         8511624    False
         Length: 8511625, dtype: bool
```

Describing Total Missing Values in the Datasets

```
In [19]: # Total missing values for each feature
         mydata.isnull().sum()
```

```
Out[19]: Accident_Index                                  0
         Location_Easting_OSGR                      10263
         Location_Northing_OSGR                     10263
         Longitude                                4887388
         Latitude                                 4887388
         Police_Force                                   0
         Accident_Severity                              0
         Number_of_Vehicles                             0
         Number_of_Casualties                           0
         Date                                           8
         Day_of_Week                                    0
         Time                                         923
         Local_Authority_(District)                     0
         Local_Authority_(Highway)                      0
         1st_Road_Class                                 0
         1st_Road_Number                                0
         Road_Type                                      0
         Speed_limit                                   37
         Junction_Detail                                0
         Junction_Control                               0
         2nd_Road_Class                                 0
         2nd_Road_Number                                0
         Pedestrian_Crossing-Human_Control              0
         Pedestrian_Crossing-Physical_Facilities        0
         Light_Conditions                               0
         Weather_Conditions                             0
         Road_Surface_Conditions                        0
         Special_Conditions_at_Site                     0
         Carriageway_Hazards                            0
         Urban_or_Rural_Area                            0
         Did_Police_Officer_Attend_Scene_of_Accident    0
         LSOA_of_Accident_Location                5190678
         dtype: int64
```

Statistical data Description

```
In [20]: mydata.describe()
```

Out[20]:

	Location_Easting_OSGR	Location_Northing_OSGR	Longitude	Latitude	Police_Force	Accident_Severity	Number_of_Vehicles	Number_of_Casu
count	8.501362e+06	8.501362e+06	3.624237e+06	3.624237e+06	8.511625e+06	8.511625e+06	8.511625e+06	8.51162
mean	4.333716e+05	3.009623e+05	-1.415332e+00	5.255985e+01	3.004689e+01	2.786522e+00	1.783262e+00	1.33155
std	1.063388e+05	1.686127e+05	1.398114e+00	1.434830e+00	2.629903e+01	4.473819e-01	7.282905e-01	8.2277
min	0.000000e+00	0.000000e+00	-7.536169e+00	4.991236e+01	1.000000e+00	1.000000e+00	1.000000e+00	1.00000
25%	3.699900e+05	1.777500e+05	-2.331407e+00	5.148910e+01	6.000000e+00	3.000000e+00	1.000000e+00	1.00000
50%	4.373100e+05	2.641500e+05	-1.375062e+00	5.224528e+01	2.300000e+01	3.000000e+00	2.000000e+00	1.00000
75%	5.218100e+05	3.982100e+05	-2.061900e-01	5.345665e+01	4.500000e+01	3.000000e+00	2.000000e+00	1.00000
max	9.999800e+05	1.213700e+06	1.762010e+00	6.080166e+01	9.800000e+01	3.000000e+00	1.920000e+02	9.30000

Length and Shape of Accident Datasets

```
In [23]: len(mydata)
Out[23]: 8511625

In [25]: mydata.shape
Out[25]: (8511625, 32)
```

Data Types

```
In [30]: mydata.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8511625 entries, 0 to 8511624
Data columns (total 32 columns):
 #   Column                                       Dtype
---  ------                                       -----
 0   Accident_Index                               object
 1   Location_Easting_OSGR                        float64
 2   Location_Northing_OSGR                       float64
 3   Longitude                                    float64
 4   Latitude                                     float64
 5   Police_Force                                 int64
 6   Accident_Severity                            int64
 7   Number_of_Vehicles                           int64
 8   Number_of_Casualties                         int64
 9   Date                                         object
 10  Day_of_Week                                  int64
 11  Time                                         object
 12  Local_Authority_(District)                   int64
 13  Local_Authority_(Highway)                    object
 14  1st_Road_Class                               int64
 15  1st_Road_Number                              int64
 16  Road_Type                                    int64
 17  Speed_limit                                  float64
 18  Junction_Detail                              int64
 19  Junction_Control                             int64
 20  2nd_Road_Class                               int64
 21  2nd_Road_Number                              int64
 22  Pedestrian_Crossing-Human_Control            int64
 23  Pedestrian_Crossing-Physical_Facilities      int64
 24  Light_Conditions                             int64
 25  Weather_Conditions                           int64
 26  Road_Surface_Conditions                      int64
 27  Special_Conditions_at_Site                   int64
 28  Carriageway_Hazards                          int64
 29  Urban_or_Rural_Area                          int64
 30  Did_Police_Officer_Attend_Scene_of_Accident  int64
 31  LSOA_of_Accident_Location                    object
dtypes: float64(5), int64(22), object(5)
memory usage: 2.0+ GB
```

Descriptive Statistics

The descriptive statistics verify the standard deviation and mean max and min values from the datasets index. It also detect the string values from the index that is very meaningful.

```
In [31]: import numpy as np
```

```
In [32]: mydata.describe(include=np.object)
```

Out[32]:

	Accident_Index	Date	Time	Local_Authority_(Highway)	LSOA_of_Accident_Location
count	8511625	8511617	8510702	8511625	3320947
unique	8511625	14975	1439	209	35596
top	1979220605046	21/12/1979	17:00	9999	E01000004
freq	1	1655	87688	3735552	4248

```
In [33]: mydata["Accident_Severity"].value_counts()
```

```
Out[33]: 3    6831813
         2    1542581
         1     137231
         Name: Accident_Severity, dtype: int64
```

```
In [37]: mydata["Weather_Conditions"].value_counts()
```

```
Out[37]:  1    6559767
          2    1151401
          8     307112
          4     131658
          5     124187
          9     104490
          7      63853
          3      53055
          6      13989
         -1       2113
         Name: Weather_Conditions, dtype: int64
```

Data Frames

```
In [39]: mydata.index
```

```
Out[39]: RangeIndex(start=0, stop=8511625, step=1)
```

```
In [40]: mydata.values
```

```
Out[40]: array([['197901A11AD14', nan, nan, ..., -1, -1, nan],
                ['197901A1BAW34', 198460.0, 894000.0, ..., -1, -1, nan],
                ['197901A1BFD77', 406380.0, 307000.0, ..., -1, -1, nan],
                ...,
                ['2019984107219', 318544.0, 567087.0, ..., 2, 2, nan],
                ['2019984107419', 336525.0, 584226.0, ..., 2, 2, nan],
                ['201998QC01004', 291367.0, 608364.0, ..., 2, 1, nan]],
               dtype=object)
```

```
In [45]: mydata["Accident_Severity"]

Out[45]: 0          3
         1          3
         2          3
         3          3
         4          2
                   ..
         8511620    3
         8511621    3
         8511622    3
         8511623    3
         8511624    2
         Name: Accident_Severity, Length: 8511625, dtype: int64
```

List.(mydata.items())

```
[('Accident_Index',
  0          197901A11AD14
  1          197901A1BAW34
  2          197901A1BFD77
  3          197901A1BGC20
  4          197901A1BGF95
                   ...
  8511620    2019984106919
  8511621    2019984107019
  8511622    2019984107219
  8511623    2019984107419
  8511624    201998QC01004
  Name: Accident_Index, Length: 8511625, dtype: object),
 ('Location_Easting_OSGR',
  0               NaN
  1          198460.0
  2          406380.0
  3          281680.0
  4          153960.0
                ...
  8511620    312635.0
  8511621    337522.0
  8511622    318544.0
  8511623    336525.0
  8511624    291367.0
  Name: Location_Easting_OSGR, Length: 8511625, dtype: float64),
 ('Location_Northing_OSGR',
  0               NaN
  1          894000.0
  2          307000.0
  3          440000.0
  4          795000.0
                ...
  8511620    573392.0
  8511621    591682.0
  8511622    567087.0
  8511623    584226.0
```

```
 8511624    608364.0
 Name: Location_Northing_OSGR, Length: 8511625, dtype: float64),
('Longitude',
 0           NaN
 1           NaN
 2           NaN
 3           NaN
 4           NaN
            ...
 8511620   -3.37
 8511621   -2.98
 8511622   -3.27
 8511623   -3.00
 8511624   -3.72
 Name: Longitude, Length: 8511625, dtype: float64),
('Latitude',
 0            NaN
 1            NaN
 2            NaN
 3            NaN
 4            NaN
            ...
 8511620    55.05
 8511621    55.22
 8511622    54.99
 8511623    55.15
 8511624    55.36
 Name: Latitude, Length: 8511625, dtype: float64),
('Police_Force',
 0           1
 1           1
 2           1
 3           1
 4           1
            ..
 8511620    98
 8511621    98
 8511622    98
 8511623    98
 8511624    98
 Name: Police_Force, Length: 8511625, dtype: int64),
('Accident_Severity',
 0          3
 1          3
 2          3
 3          3
 4          2
           ..
 8511620    3
 8511621    3
 8511622    3
 8511623    3
 8511624    2
 Name: Accident_Severity, Length: 8511625, dtype: int64),
('Number_of_Vehicles',
 0          2
 1          1
```

```
 2          2
 3          2
 4          2
           ..
 8511620    1
 8511621    4
 8511622    2
 8511623    1
 8511624    1
 Name: Number_of_Vehicles, Length: 8511625, dtype: int64),
('Number_of_Casualties',
 0          1
 1          1
 2          3
 3          2
 4          1
           ..
 8511620    1
 8511621    1
 8511622    1
 8511623    1
 8511624    1
 Name: Number_of_Casualties, Length: 8511625, dtype: int64),
('Date',
 0          18/01/1979
 1          01/01/1979
 2          01/01/1979
 3          01/01/1979
 4          01/01/1979
              ...
 8511620    18/05/2019
 8511621    30/05/2019
 8511622    21/06/2019
 8511623    29/06/2019
 8511624    21/04/2019
 Name: Date, Length: 8511625, dtype: object),
('Day_of_Week',
 0          5
 1          2
 2          2
 3          2
 4          2
           ..
 8511620    7
 8511621    5
 8511622    6
 8511623    7
 8511624    1
 Name: Day_of_Week, Length: 8511625, dtype: int64),
('Time',
 0          08:00
 1          01:00
 2          01:25
 3          01:30
 4          01:30
            ...
 8511620    01:00
```

```
 8511621    08:46
 8511622    15:30
 8511623    14:10
 8511624    12:45
 Name: Time, Length: 8511625, dtype: object),
('Local_Authority_(District)',
 0           11
 1           23
 2           17
 3            2
 4          510
           ...
 8511620    917
 8511621    917
 8511622    917
 8511623    917
 8511624    917
 Name: Local_Authority_(District), Length: 8511625, dtype: int64),
('Local_Authority_(Highway)',
 0              9999
 1              9999
 2              9999
 3              9999
 4              9999
             ...
 8511620    S12000006
 8511621    S12000006
 8511622    S12000006
 8511623    S12000006
 8511624    S12000006
 Name: Local_Authority_(Highway), Length: 8511625, dtype: object),
('1st_Road_Class',
 0          3
 1          6
 2          3
 3          3
 4          3
           ..
 8511620    4
 8511621    3
 8511622    4
 8511623    6
 8511624    3
 Name: 1st_Road_Class, Length: 8511625, dtype: int64),
('1st_Road_Number',
 0            4
 1            0
 2          112
 3          502
 4          309
           ...
 8511620    725
 8511621      7
 8511622    723
 8511623    710
 8511624    702
 Name: 1st_Road_Number, Length: 8511625, dtype: int64),
```

```
('Road_Type',
 0          1
 1          9
 2          9
 3         -1
 4          6
           ..
 8511620    6
 8511621    6
 8511622    6
 8511623    6
 8511624    6
 Name: Road_Type, Length: 8511625, dtype: int64),
('Speed_limit',
 0          30.0
 1          30.0
 2          30.0
 3          30.0
 4          30.0
            ...
 8511620    60.0
 8511621    60.0
 8511622    60.0
 8511623    30.0
 8511624    60.0
 Name: Speed_limit, Length: 8511625, dtype: float64),
('Junction_Detail',
 0          1
 1          3
 2          6
 3          3
 4          0
           ..
 8511620    0
 8511621    0
 8511622    3
 8511623    3
 8511624    0
 Name: Junction_Detail, Length: 8511625, dtype: int64),
('Junction_Control',
 0          4
 1          4
 2          4
 3          2
 4         -1
           ..
 8511620   -1
 8511621   -1
 8511622    4
 8511623    4
 8511624   -1
 Name: Junction_Control, Length: 8511625, dtype: int64),
('2nd_Road_Class',
 0         -1
 1         -1
 2         -1
 3         -1
```

```
 4         -1
           ..
 8511620   -1
 8511621   -1
 8511622    4
 8511623    6
 8511624   -1
 Name: 2nd_Road_Class, Length: 8511625, dtype: int64),
('2nd_Road_Number',
 0           -1
 1           -1
 2           -1
 3           -1
 4            0
           ...
 8511620      0
 8511621      0
 8511622    721
 8511623    723
 8511624      0
 Name: 2nd_Road_Number, Length: 8511625, dtype: int64),
('Pedestrian_Crossing-Human_Control',
 0         -1
 1         -1
 2         -1
 3         -1
 4         -1
           ..
 8511620    0
 8511621    0
 8511622    0
 8511623    0
 8511624    0
 Name: Pedestrian_Crossing-Human_Control, Length: 8511625, dtype: int64),
('Pedestrian_Crossing-Physical_Facilities',
 0         -1
 1         -1
 2         -1
 3         -1
 4         -1
           ..
 8511620    0
 8511621    0
 8511622    0
 8511623    0
 8511624    0
 Name: Pedestrian_Crossing-Physical_Facilities, Length: 8511625, dtype:
int64),
('Light_Conditions',
 0          1
 1          4
 2          4
 3          4
 4          4
           ..
 8511620    1
 8511621    1
```

```
 8511622    1
 8511623    1
 8511624    1
 Name: Light_Conditions, Length: 8511625, dtype: int64),
('Weather_Conditions',
 0          8
 1          8
 2          8
 3          8
 4          3
           ..
 8511620    1
 8511621    1
 8511622    1
 8511623    1
 8511624    1
 Name: Weather_Conditions, Length: 8511625, dtype: int64),
('Road_Surface_Conditions',
 0          1
 1          3
 2          3
 3          3
 4          3
           ..
 8511620    2
 8511621    2
 8511622    1
 8511623    1
 8511624    1
 Name: Road_Surface_Conditions, Length: 8511625, dtype: int64),
('Special_Conditions_at_Site',
 0         -1
 1         -1
 2         -1
 3         -1
 4         -1
           ..
 8511620    0
 8511621    0
 8511622    0
 8511623    0
 8511624    0
 Name: Special_Conditions_at_Site, Length: 8511625, dtype: int64),
('Carriageway_Hazards',
 0          0
 1          0
 2          0
 3          0
 4          0
           ..
 8511620    0
 8511621    0
 8511622    0
 8511623    0
 8511624    0
 Name: Carriageway_Hazards, Length: 8511625, dtype: int64),
('Urban_or_Rural_Area',
```

```
 0          -1
 1          -1
 2          -1
 3          -1
 4          -1
            ..
 8511620     2
 8511621     2
 8511622     2
 8511623     2
 8511624     2
 Name: Urban_or_Rural_Area, Length: 8511625, dtype: int64),
('Did_Police_Officer_Attend_Scene_of_Accident',
 0          -1
 1          -1
 2          -1
 3          -1
 4          -1
            ..
 8511620     1
 8511621     1
 8511622     2
 8511623     2
 8511624     1
```

Data Preprocessing

In the given datasets all accident record has been tested and managed according to the requirements. In this report all the official words has been tested carefully. We have been organized the total data set according to the main function. In the total data the variable has been tested very carefully and the previous accidents datasets has been tested briefly. It has been since tested many times the missing and null values. Since it has been described that the missing values can affect performance. We adopted the different methods that uses the average value of the feature columns to provide the required amount. We have been used different statistical method to describe the datasets that will not affect the mean.

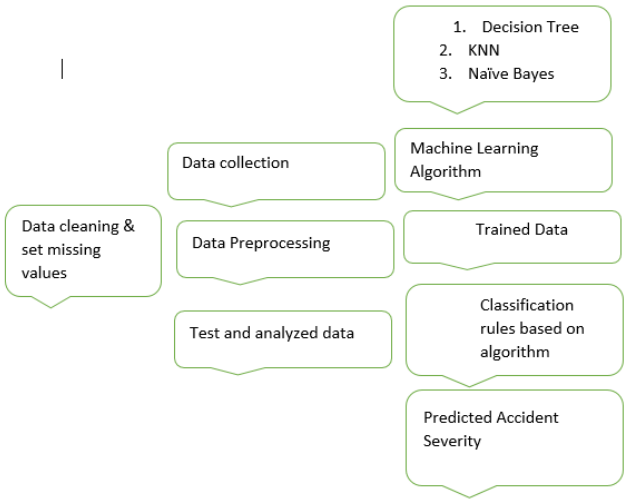


Figure 1 Data processing & data cleaning method

Final Datasets

Final datasets using python anaconda jupyter notebook software. We use UK accident datasets. This record has been described in detail. Datasets has been tested in many times, analyzed many times and different technical observation has been tested briefly. The datasets is 1979 – 2019 years period. The data sets has been saved as a Accident_Datasets.csv file.

Background

The report uses the python data set for machine learning analysis. The relationship between the accident besides numerous factors, and the weather, road conditions, driving behavior, etc. Machine learning model is an extensively used technique for learning the association between groups of variables. We can analyze the functional association between variables, after check the accuracy of the relationship and verify the null and missing values in the accident data set. In this report, single-level and multi-level analyses have been extensively conducted, and a expressive analysis has been conducted to investigate the influence of various factors.

Literature Review

We found that the literature studies the Support vector machine (SVM) model has been the best performance and the results. The studies describe that the 11 to 15 accident types has been the average of the SVM model has 0.73 and the average of the model is 0.63 to 0.67 score[].

The usually found that he accident cases has been placed mislabeled according to the evaluating the confusion matrix and the qualitative evaluation has been mislabeled.

The different set of 1000 labelled form the accident datasets has been the 3000 unlabeled that has been made publicly available. []

Crum at el. Has tested a face to face interview from the different approx. 500 different truck drivers at different 5 rest stops in between the united states and European countries, in this study the three index has been close calls insights and exhaustion the crash participation

The study from the Crum and Morrow[] describe that the statistical modeling of the trucking companies based on their safety records and their health. Crum and Morrow has been selected the sample from each of the three statistical quartile to check the poorest safety performance and the best safety performance form the middle of the two quartiles to investigate the quartile to identify the safety performance.

The recent studies from the research Stern et al. [] found the research of related to commercial motor vehicles and drivers. The main important problem is addressed that the running controlled experiment

by the different imposing treatments has tested. The different research has been design and different observational studies that associate with the effects of variables.

The studies of Bowden and Ragsdale [] found that the optimization algorithm has been scheduling the drivers performance during driving a car at the road, this performance algorithm has described that the to reduce the minimize the trip duration while subject to the level of other constraints. The maximum driving hours in the Europe and the united states that under the law and order. This algorithm assumed that the three processes has been best fit in the observation of driving hours Akerstedl et al[].

Xu et al[] has been developed a crash prediction model in different level of their road accident crash severity. This model of crash severity was considered and appreciated.

Olson et al[] has been described in their studies that the distracted driving the 203 to 205 different commercial drivers that involved in 4452 critical events such road accident. Road accident during driving many vehicles has been crashed near the departure lane that is alongwith 19888 time periods of the driving hours has been increases due to their driving behaviors. Many critical accident has been reports about 60% proceeding happened while driving the driving performance for non – driving tasks during the driving.

Wang et al [] found in their studies that the numerous issued has been increasing due to traffic conditions. They recommend that the traffic and road accident has been increased due to weather condition. The road is slipper and the bad road condition has been detected that is the cause of accident.

The recent studies of Pande et al[] describe that the road accident has detected from close to close end. That they found in their analysis ratio of 5:1 of non-crashes to crashes and its uses the random forest in their variables to select the different strategies to count the neural network inference. The found that in their studies the average speed of downstream and upstream is not a significant.

Sun and Sun [] using the matched case control design with the ratio of 5:1 to implement the Markov model which is not provide the states of the upstream and downstream of road accident. If we consider that the upstream is increasing on highway that its must be influence with the downstream. The drawback in their studies that they cannot find the exact ratio of accident. They cannot provide the exact road and weather conditions.

The studies found that [] many of the statistical testing has unbalanced and panel data sets used hourly records on crashes on highway. It has been deeply concerned with the real time datasets of weather condition is different form every end and year. The lower speed of driver during driving a car on national highway of European roads has been increasing the crashes of car and accident. Basagana et al (2015) found in their studies the weather condition is poor and its not according to the road conditions. It also detected in their datasets the heat weaves is not considering at all that directly associated with the drivers performance.

Parvareh et al(2018) found in their studies that the air temperature has been deeply influence on pedestrian and motorcycle accidents in Europe. Many of the statistical accident has been reported and tested in their studies but the drawback is that its not completely upto the mark. Edwards (1999) found that misreporing of weather or road surface conditions is also a common error and the police is reporting

the accident during the busy driving hours. The results has mismatched pairs analysis are the compared to the results and achieved though the traditional.

In Europe many metrological stations has based on strategic planner. The event of rainfall is concerned with road condition and weather condition that is limited to do the reporting. Shen et al(2020) found in their studies that journey may be small and the risk of death and traffic injuries has been increasing due to increasing amount of exposure.

Ashraf et al. (2019) postulate that but road accidents occur almost equally during the night and day, more fatalities are recorded during night-time travel.

Perrels et. (2015) suggest that the weather is a critical component in a driving a car comprise with road accidents.

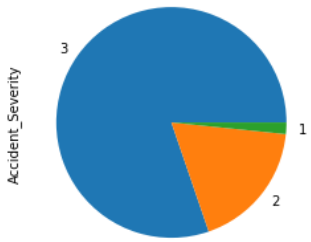
Naqvi et al(2020) investigated in the study that the effect of fuel prices in Britain found that the number of fatal crashes decreased by 0.4% for every 1% increase in fuel prices. They concluded that fuel prices mediate fatal crashes by reducing exposure as a resul of less travel and moderated driver behavior like speed reduction .

Levulytel et al(2017) analyzed that the pedestrian accidents in Europe is high and they identified that pedestrian crossing designs as a contributor to traffic accidents.

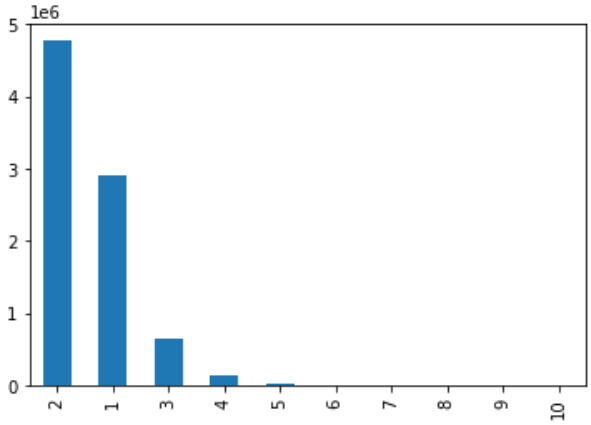
Botswana Pego(2009) found in in their studies that traffic police used two different forms to record accident data leading to irregularities.

Data Analysis

```
In [62]: mydata["Accident_Severity"].value_counts().head(10).plot(kind="pie")
Out[62]: <AxesSubplot:ylabel='Accident_Severity'>
```



```
In [64]: mydata["Number_of_Vehicles"].value_counts().head(10).plot(kind="bar")
Out[64]: <AxesSubplot:>
```



Results

A Appendix: Technical Details

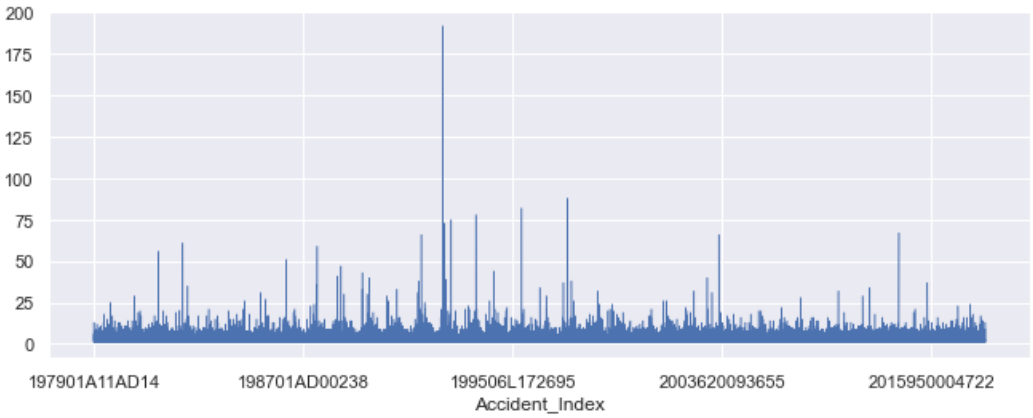
Data Visualization shows an important role in the time series analysis and data forecasting. Plots of the raw sample data can provide the appreciated analytic to identify time-based structures like trends, cycles, and seasonality that can influence the choice of model.

A.1 Creation of Data Frame and Additive Decomposition of Time-Series

```
import matplotlib.pyplot as plt

# Display figures inline in Jupyter notebook

import seaborn as sns

# Use seaborn style defaults and set the default figure size

sns.set(rc={'figure.figsize':(11, 4)})

mydata['Number_of_Vehicles'].plot(linewidth=0.5);
```

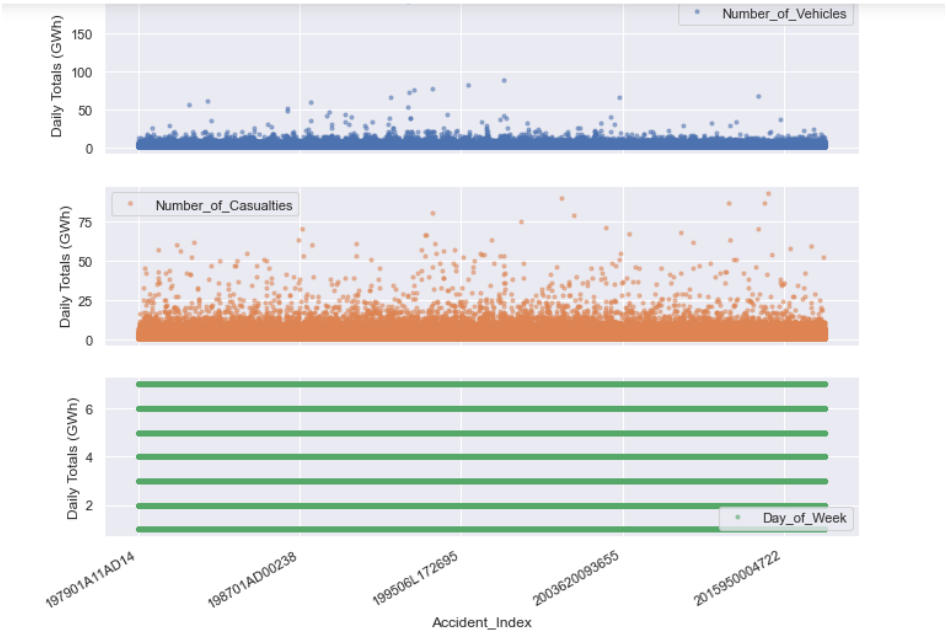
This model describe the Accident index and number of vehicles accident ratio.

```
In [22]: mydata['Number_of_Vehicles'].plot(linewidth=0.5);
```



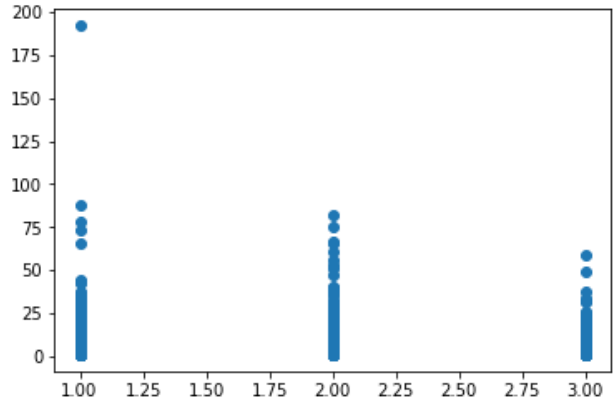

```
cols_plot = ['Number_of_Vehicles', 'Number_of_Casualties', 'Day_of_Week']

axes = mydata[cols_plot].plot(marker='.', alpha=0.5, linestyle='None', figsize=(11, 9), subplots=True)

for ax in axes:

    ax.set_ylabel('Daily Totals (GWh)')
```

This model describe the Number of vehicles and number casualties happened in day of week. In the previous research these models would not be shown. So we improve our research and investigate its more meaningful way.



In [15]:
```
import matplotlib.pyplot as plt
```

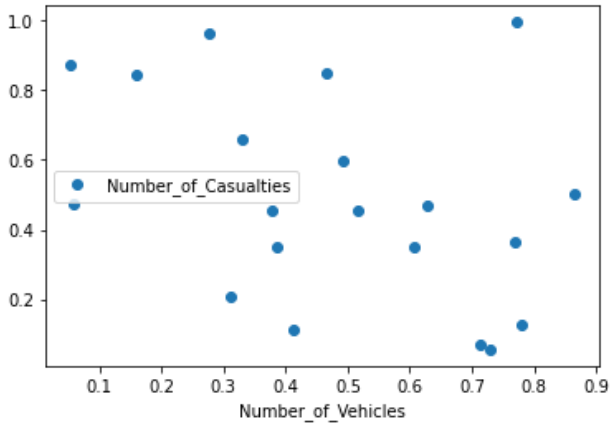
In [19]:
```
import matplotlib.pyplot as plt
plt.scatter(mydata['Accident_Severity'], mydata['Number_of_Vehicles'])
plt.show() # Depending on whether you use IPython or interactive mode, etc.
```



This model describe the accident severity and number of vehicles relationship. Its very important to view. In the previous research these models would not be shown. So we improve our research and investigate its more meaningful way.

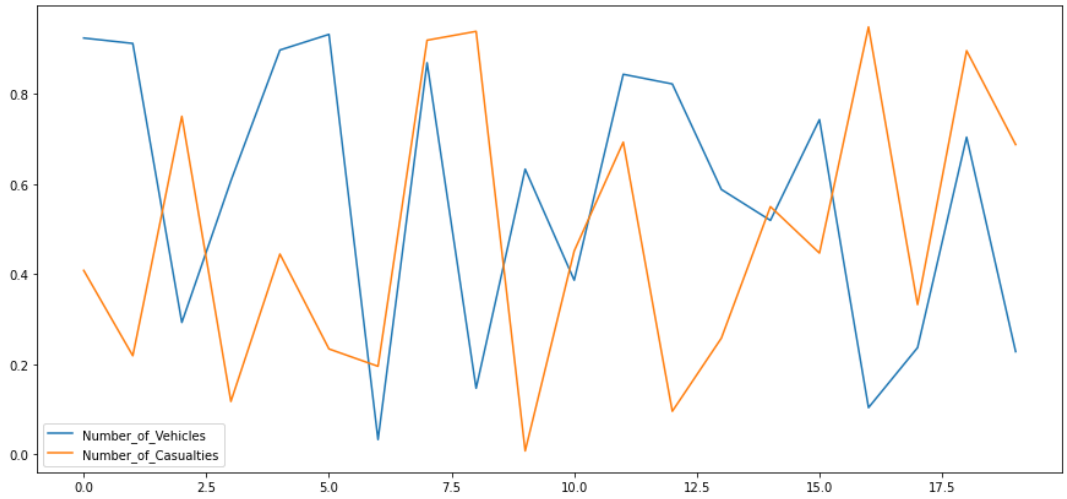
In [20]:
```
import pandas as pd
import numpy as np

#creating sample data
mydata={'Number_of_Vehicles':np.random.rand(20),
      'Number_of_Casualties': np.random.rand(20)}
mydata= pd.DataFrame(mydata)
mydata.plot(x='Number_of_Vehicles', y='Number_of_Casualties', style='o')
```

Out[20]: <AxesSubplot:xlabel='Number_of_Vehicles'>



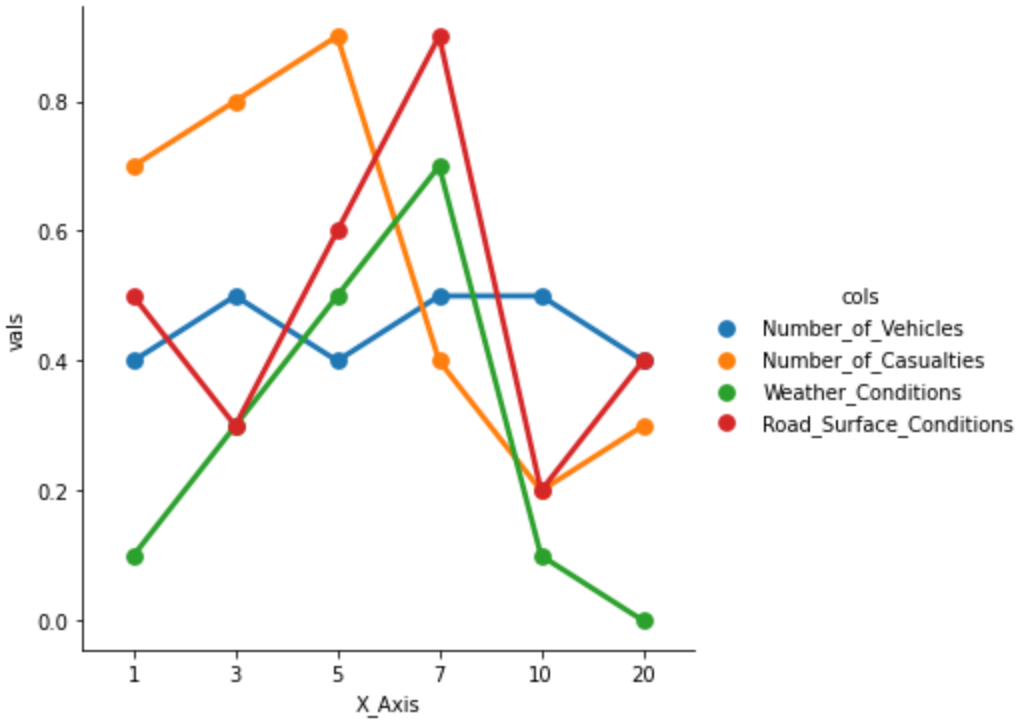
Testing the ARIMA Model. In the previous research these models would not be shown. So we improve our research and investigate its more meaningful way.

```
In [11]: from pylab import rcParams
         rcParams['figure.figsize'] = 15, 7
         mydata.plot()

Out[11]: <AxesSubplot:>
```



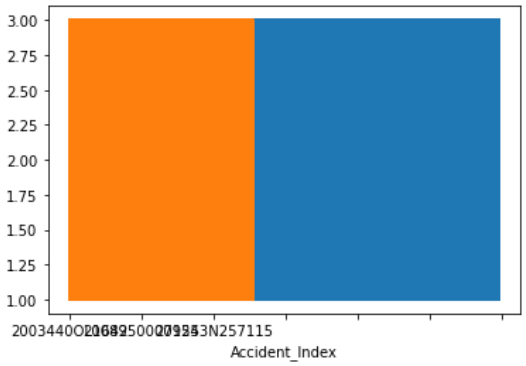
Model Fitting plot

```
mydata = pd.DataFrame({'X_Axis':[1,3,5,7,10,20],
                       'Number_of_Vehicles':[.4,.5,.4,.5,.5,.4],
                       'Number_of_Casualties':[.7,.8,.9,.4,.2,.3],
                       'Weather_Conditions':[.1,.3,.5,.7,.1,.0],
                       'Road_Surface_Conditions':[.5,.3,.6,.9,.2,.4]})
Print(mydata)
mydata = mydata.melt('X_Axis', var_name='cols',  value_name='vals')
g = sns.factorplot(x="X_Axis", y="vals", hue='cols', data=mydata)
```



A.4 MSE Calculation by ARIMA Model Fitting

```
In [24]:  #plotting the data
          train['Accident_Severity'].plot()
          valid['Accident_Severity'].plot()

Out[24]: <AxesSubplot:xlabel='Accident_Index'>
```



In [8]:
```
train = mydata[mydata["Accident_Severity"] == "Train"]
test = mydata[mydata["Accident_Severity"] == "Test"]
print(train.shape)
print(test.shape)
```

```
(0, 31)
(0, 31)
```

In [10]:
```
train_array = train["Number_of_Casualties"]
print(train_array.shape)

test_array = test["Number_of_Casualties"]
print(test_array.shape)
```

```
(0,)
(0,)
```

Conclusion

In the comprehension report many time series and ARIMA model has been tested. ARIMA model of forecasting using python has been tested successfully. The objective of this research is detect the different statistical analysis to achieve the better road accident prediction. Many machine learning algorithm has been tested and analyzed. Data is being so huge and every data point is being numeric. And each value of machine learning has been applied on given columns. Time series data forecasting has been tested.

References

[1] Thomas, & PETE and Morries (2013) Identifying the causes of Road Crashes in Europe. Annals of Advance Automotaive medicine / Annual Scientific concerence. 57.12-22

[2] Heinrich, H.W. Industrial Accident Prevention , A Scientific Approach McGraw-Hill.

[3] Ljung, M(2002) Driving Reliability and Error Analysis Method , Its Master Thesis University Linkoping.

[4] Zhuoning Yuan University of Lowa Prediction of Traffic Accidents though Heterogenous Urban datasets Case studies.

[5] Joaquin Abellan, & Juan De Ona 2013. Analysis of Traffic Accident severity using decision rules via decision trees. Expert System application 40,15(2013)

[6] Mikhail Belkin & Partha 2001. Laplacian Eigenmaps and Spectral techniques for clustering NIPS Vol 14, 585. 591.

[7] Ruth Bergel, M Debbarth, (2013) Explaining the Road Accident Risk Weather Effects , Accident Analysis & Prevention 60 (2013).

[8} F. Pedregosa, G Varoguax, Scikit-learn: Macine Learning in Python (2011) Journal of Machine learning research 12(2011) 2825-2830.

[9] National Weather Serrvices.(2017) Edition. National Digital Forecast Database

[10] JD Tamerius, R mantilla (2016) Precipitation Effects on Motor Vehicle Crashes Vary by Space time and environmental conditions, weather climate & Society 8, 4 (2016), 399-407

[11] Labib & Farhan (2019) Road Accident Analysis and Prediction of Accident Severity by using machine learning in Bangladesh 1-5/ICSC.2019

[12] N.V and P.A Nandurge (2017) Analyzing Road Accident Data using machine learning paradigms. International conference I-SMAC (Social Mobile analytics) 2017. Pp.604.610.

[13] D.Toshnival & S. Kumar (2016) A data mining approach to characterize road Accidents locations Journal of Modern Transportation Vol 24,PP 62-72

[14] Shawndra Hill and Beshah (2010) Mining road traffic accident data to improve safety. And role of road related factors and accident serverity in Ethipoia. Atificail intelligence development.

[15] M. Khalili and A Esmaeili. (2012) Determining the road defects impact on accident severity. Based on vehicle situation after accident apporoach on logistic regression 2012. International Conference of Statistics in Science 2012 pp 1-4

[16] Wang. R (2012) Adaboost for feature selection classification and its relation with SVM. A review physical procedia vol 25. Pp.800-807.

[17] Zhang,Z. (2016) introduction to machine learning k-nearest neigbors. Annal of translation medicine vol 4, pp.218.218

[18] Rish I . (2001) an empirical study of the Naïve Bayes Classifier, IJCAI 2001 work empirical method vol 3.

[19] M. Khalili and A. Esmaelili (2012) Determining the road defects impact on accident severity. Based on vehicle situation accident an approach of logistic regression. International conference on statistics in science and business Pp. 1-4

[20] A. Juspo and A Mamun (2017) An intelligent smartphone based approach using IOT for ensuring safe driving . international conference on electrical engineering and computer science pp.217-223.

[21] Syed Masiur and Khaled Predicting crash injury severity with machine learning algorithm. Synergized with clustering technique. Promising protocol journal of environmental research and public health.

[22] Fung, S.H. Yau K.W.K. (2006) Multiple Vehicle accidents in Hong Kong. 1157-1161.

[23] Ivan, J. Zajac. S.S. Factors of influencing injury severity of motor vehicle – crossing pedestrian crashes in rural connectricut. Anal pre 369-379.

[24] Kones Publications road accident big data mining & visualization using support vector machine wright university.

[25] Miaco Cai & Amir a review of data analytics applications in road traffic safety Descriptive predictive modeling MDPI.

[26] Soojung Hur, Imran Ashraf Catastrophic factors involved in road accidents underlying causes and descriptive analysis PLOS ONE.

[27] D & Mcnamara T (2014) The influence of Rainfall on Road Accidents in Urban Areas a weather road approach. Travel of behavior society pp15.21.

[28] Athanasosios Votsis (2015) weather conditions and weather information and Car crashes ISPRS international journals of Geo information

List of Figures

Figure 1 Data processing & data cleaning method