preProcessing_01

November 22, 2020

- @Author : Pramil Paudel, Sumit Bhattarai
- Development Env : Jupyter Lab
- Module : Preprocessing
- Summary : This module will create a data using some data modulation technique and create an intermediate data to process further.

NOTE : This notebook is used to modify raw data into preprocessed form. Main steps done here are looking into the datatype, data, no of NAs and combined it with USA geographical data so that we can get better visualization later.

[1]:
```
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from pylab import rcParams
from mpl_toolkits.mplot3d import Axes3D
from pandas.plotting import scatter_matrix
from IPython.display import Image
import pydotplus
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn import preprocessing
from sklearn import svm
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
from sklearn import preprocessing
from sklearn.tree import export_graphviz
from sklearn.externals.six import StringIO
from sklearn.cluster import DBSCAN
from sklearn import metrics
from sklearn.datasets import make_blobs
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.ensemble import RandomForestRegressor
from sklearn.neural_network import MLPRegressor
```

```
from sklearn.datasets import make_regression
from fbprophet import Prophet
print("Loaded Successfully -- -- -- -- -- -- -- -- -- -- -- -- --")
```

```
Loaded Successfully -- -- -- -- -- -- -- -- -- -- -- -- --
```

```
/Users/patthar/opt/anaconda3/lib/python3.7/site-
packages/sklearn/externals/six.py:31: DeprecationWarning: The module is
deprecated in version 0.21 and will be removed in version 0.23 since we've
dropped support for Python 2.7. Please rely on the official version of six
(https://pypi.org/project/six/).
  "(https://pypi.org/project/six/).", DeprecationWarning)
```

1 1. DATA LAODING TO DF

Defining input/output directory

[4]:
```
# This input path is not the part of project directory as it contains very␣
↪large dataset.
input_path_source = "../../../Raw_Data/"

# Following two path are part of project directory
input_path_raw = "../../data/raw/"
output_path_preprocessing = "../../data/pre_processing/"
```

[5]:
```
road_accident_data = pd.read_csv(input_path_source+"usa_total_accodent.csv")
road_accident_data.describe()
```

[5]:

	TMC	Severity	Start_Lat	Start_Lng	End_Lat
count	2.478818e+06	3.513617e+06	3.513617e+06	3.513617e+06	1.034799e+06
mean	2.080226e+02	2.339929e+00	3.654195e+01	-9.579151e+01	3.755758e+01
std	2.076627e+01	5.521935e-01	4.883520e+00	1.736877e+01	4.861215e+00
min	2.000000e+02	1.000000e+00	2.455527e+01	-1.246238e+02	2.457011e+01
25%	2.010000e+02	2.000000e+00	3.363784e+01	-1.174418e+02	3.399477e+01
50%	2.010000e+02	2.000000e+00	3.591687e+01	-9.102601e+01	3.779736e+01
75%	2.010000e+02	3.000000e+00	4.032217e+01	-8.093299e+01	4.105139e+01
max	4.060000e+02	4.000000e+00	4.900220e+01	-6.711317e+01	4.907500e+01

	End_Lng	Distance(mi)	Number	Temperature(F)
count	1.034799e+06	3.513617e+06	1.250753e+06	3.447885e+06
mean	-1.004560e+02	2.816167e-01	5.975383e+03	6.193512e+01
std	1.852879e+01	1.550134e+00	1.496624e+04	1.862106e+01
min	-1.244978e+02	0.000000e+00	0.000000e+00	-8.900000e+01
25%	-1.183440e+02	0.000000e+00	8.640000e+02	5.000000e+01
50%	-9.703438e+01	0.000000e+00	2.798000e+03	6.400000e+01
75%	-8.210168e+01	1.000000e-02	7.098000e+03	7.590000e+01
max	-6.710924e+01	3.336300e+02	9.999997e+06	1.706000e+02

```
       Wind_Chill(F)   Humidity(%)  Pressure(in)  Visibility(mi)  \
count   1.645368e+06  3.443930e+06  3.457735e+06    3.437761e+06
mean    5.355730e+01  6.511427e+01  2.974463e+01    9.122644e+00
std     2.377334e+01  2.275558e+01  8.319758e-01    2.885879e+00
min    -8.900000e+01  1.000000e+00  0.000000e+00    0.000000e+00
25%     3.570000e+01  4.800000e+01  2.973000e+01    1.000000e+01
50%     5.700000e+01  6.700000e+01  2.995000e+01    1.000000e+01
75%     7.200000e+01  8.400000e+01  3.009000e+01    1.000000e+01
max     1.150000e+02  1.000000e+02  5.774000e+01    1.400000e+02

       Wind_Speed(mph)  Precipitation(in)
count     3.059008e+06       1.487743e+06
mean      8.219025e+00       1.598256e-02
std       5.262847e+00       1.928262e-01
min       0.000000e+00       0.000000e+00
25%       5.000000e+00       0.000000e+00
50%       7.000000e+00       0.000000e+00
75%       1.150000e+01       0.000000e+00
max       9.840000e+02       2.500000e+01
```

[4]:
```
road_accident_data.head()
```

[4]:
```
     ID    Source    TMC  Severity           Start_Time             End_Time  \
0  A-1  MapQuest  201.0         3  2016-02-08 05:46:00  2016-02-08 11:00:00
1  A-2  MapQuest  201.0         2  2016-02-08 06:07:59  2016-02-08 06:37:59
2  A-3  MapQuest  201.0         2  2016-02-08 06:49:27  2016-02-08 07:19:27
3  A-4  MapQuest  201.0         3  2016-02-08 07:23:34  2016-02-08 07:53:34
4  A-5  MapQuest  201.0         2  2016-02-08 07:39:07  2016-02-08 08:09:07

   Start_Lat  Start_Lng  End_Lat  End_Lng  …  Roundabout Station   Stop  \
0  39.865147 -84.058723      NaN      NaN  …       False   False  False
1  39.928059 -82.831184      NaN      NaN  …       False   False  False
2  39.063148 -84.032608      NaN      NaN  …       False   False  False
3  39.747753 -84.205582      NaN      NaN  …       False   False  False
4  39.627781 -84.188354      NaN      NaN  …       False   False  False

  Traffic_Calming Traffic_Signal Turning_Loop Sunrise_Sunset Civil_Twilight  \
0           False          False        False          Night          Night
1           False          False        False          Night          Night
2           False           True        False          Night          Night
3           False          False        False          Night            Day
4           False           True        False            Day            Day

  Nautical_Twilight Astronomical_Twilight
0             Night                 Night
1             Night                   Day
2               Day                   Day
```

```
3               Day                   Day
4               Day                   Day

[5 rows x 49 columns]
```

2 1. LOOKING INTO MORE DETAILS OF FEATURES

[5]:
```
road_accident_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3513617 entries, 0 to 3513616
Data columns (total 49 columns):
ID                        object
Source                    object
TMC                       float64
Severity                  int64
Start_Time                object
End_Time                  object
Start_Lat                 float64
Start_Lng                 float64
End_Lat                   float64
End_Lng                   float64
Distance(mi)              float64
Description               object
Number                    float64
Street                    object
Side                      object
City                      object
County                    object
State                     object
Zipcode                   object
Country                   object
Timezone                  object
Airport_Code              object
Weather_Timestamp         object
Temperature(F)            float64
Wind_Chill(F)             float64
Humidity(%)               float64
Pressure(in)              float64
Visibility(mi)            float64
Wind_Direction            object
Wind_Speed(mph)           float64
Precipitation(in)         float64
Weather_Condition         object
Amenity                   bool
Bump                      bool
```

```
Crossing                bool
Give_Way                bool
Junction                bool
No_Exit                 bool
Railway                 bool
Roundabout              bool
Station                 bool
Stop                    bool
Traffic_Calming         bool
Traffic_Signal          bool
Turning_Loop            bool
Sunrise_Sunset          object
Civil_Twilight          object
Nautical_Twilight       object
Astronomical_Twilight   object
dtypes: bool(13), float64(14), int64(1), object(21)
memory usage: 1008.6+ MB
```

The main purpose of this section is to find out the co-relation between different attributes of the data and selecting the required number of features. We will start from the total columns present in the data and find out what are the important features to consider. The details of each of the data attributes/columns are available in the READ.me file

Following columns are removed from original data with following assumption

- Source : Since source is the API path by which data is collected, we decided to remove it as it doesn't provide any significance to the accident.
- TMC : TMC is dropped from the data as it is Traffic Message Channel and has no relation to our intended study.
- Distance : The distance is caused by the accident and it doesn't have any relation to the accident.
- Description : Since we are not analysing the data based on description of accident.
- Number : The street Number is also dropped as we are not analysing based on street number.
- Civil_Twilight : The day/Night information will be analysed based on only sunrise and sunset.
- Nautical_Twilight : The day/Night information will be analysed based on only sunrise and sunset.
- Astronomical_Twilight : The day/Night information will be analysed based on only sunrise and sunset.

[6]:
```
road_accident_df_1 = road_accident_data.
 ↪drop(["Source","TMC","Distance(mi)","Description","Number","Civil_Twilight","Nautical_Twili
road_accident_df_1.shape
```

[6]:
```
(3513617, 39)
```

That was primary removal of features based on the project motive. But if there are large number of NaN values captured in any of the feature it will make data skewed. The main idea to fill these data is either to drop them or populate some default values.

[7]:
```
road_accident_df_1.isnull().sum(axis = 0)
```

[7]:
```
ID                         0
Severity                   0
Start_Time                 0
End_Time                   0
Start_Lat                  0
Start_Lng                  0
Street                     0
Side                       0
City                     112
County                     0
State                      0
Zipcode                 1069
Country                    0
Timezone                3880
Airport_Code            6758
Weather_Timestamp      43323
Temperature(F)         65732
Wind_Chill(F)        1868249
Humidity(%)            69687
Pressure(in)           55882
Visibility(mi)         75856
Wind_Direction         58874
Wind_Speed(mph)       454609
Precipitation(in)    2025874
Weather_Condition      76138
Amenity                    0
Bump                       0
Crossing                   0
Give_Way                   0
Junction                   0
No_Exit                    0
Railway                    0
Roundabout                 0
Station                    0
Stop                       0
Traffic_Calming            0
Traffic_Signal             0
Turning_Loop               0
Sunrise_Sunset           115
dtype: int64
```

Looking at the number of NaN in each columns, “Wind Chill(F)”, “Wind_Speed(mph)”, “Precipitation(in)” contain large number of NaN which will make data unreliable so these columns are dropped.

[8]:
```
road_accident_df_2 = road_accident_df_1.
↪drop(["Wind_Chill(F)","Wind_Speed(mph)","Precipitation(in)"],axis=1)
```

Now we moved towards null removal which will not cost more than couple thousands of data

2.0.1 Other Features in the project will be discarded as per requirement by the model in the model creation and execution itself. Some of the features are kept to have make visulalization more clear and accurate

3 2. NULL REMOVAL

Lets remove null and nan data if it is in the any column of selected feature

[9]:
```
road_accident_df_2 = road_accident_df_2.dropna()
road_accident_df_2.shape
```

[9]:
```
(3402756, 36)
```

4 3.ADDING FIPS CODE FOR BETTER VISULIZATION

There is standard code provided by USA CENSUS site for easeness data is pulled from https://raw.githubusercontent.com/plotly/datasets/master/laucnty16.csv

[10]:
```
usa_fips_code = pd.read_csv(input_path_raw+"USA_FIPS_Code.csv")
usa_fips_code.head()
```

[10]:

	LAUS Code	State FIPS Code	County FIPS Code	\
0	CN0100100000000	1	1	
1	CN0100300000000	1	3	
2	CN0100500000000	1	5	
3	CN0100700000000	1	7	
4	CN0100900000000	1	9	

	County Name/State Abbreviation	Year	Labor Force	Employed	Unemployed	\
0	Autauga County, AL	2016	25,649	24,297	1,352	
1	Baldwin County, AL	2016	89,931	85,061	4,870	
2	Barbour County, AL	2016	8,302	7,584	718	
3	Bibb County, AL	2016	8,573	8,004	569	
4	Blount County, AL	2016	24,525	23,171	1,354	

	Unemployment Rate (%)
0	5.3
1	5.4
2	8.6
3	6.6
4	5.5

4.0.1 3.1 Convert fips code to standard 2 and 3 digit code

[11]:
```
def convert_to_two_digit_code(code):
    c = "0000"+str(code)
    return(c[-2:])
```

[12]:
```
usa_fips_code['State FIPS Code'] = usa_fips_code["State FIPS Code"].astype(str)
usa_fips_code['State FIPS Code'] = usa_fips_code["State FIPS Code"].
 ↪apply(convert_to_two_digit_code)
usa_fips_code.head()
# usa_fips_code["State FIPS Code"].unique
```

[12]:

	LAUS Code	State FIPS Code	County FIPS Code	\
0	CN0100100000000	01	1	
1	CN0100300000000	01	3	
2	CN0100500000000	01	5	
3	CN0100700000000	01	7	
4	CN0100900000000	01	9	

	County Name/State Abbreviation	Year	Labor Force	Employed	Unemployed	\
0	Autauga County, AL	2016	25,649	24,297	1,352	
1	Baldwin County, AL	2016	89,931	85,061	4,870	
2	Barbour County, AL	2016	8,302	7,584	718	
3	Bibb County, AL	2016	8,573	8,004	569	
4	Blount County, AL	2016	24,525	23,171	1,354	

	Unemployment Rate (%)
0	5.3
1	5.4
2	8.6
3	6.6
4	5.5

[13]:
```
def convert_to_three_digit_code(code):
    c = "00000"+str(code)
    return(c[-3:])
```

[14]:
```
usa_fips_code['County FIPS Code'] = usa_fips_code["County FIPS Code"].
 ↪astype(str)
usa_fips_code['County FIPS Code'] = usa_fips_code["County FIPS Code"].
 ↪apply(convert_to_three_digit_code).astype(str)
usa_fips_code.head()
```

[14]:

	LAUS Code	State FIPS Code	County FIPS Code	\
0	CN0100100000000	01	001	
1	CN0100300000000	01	003	
2	CN0100500000000	01	005	

```
3  CN0100700000000              01              007
4  CN0100900000000              01              009

  County Name/State Abbreviation  Year  Labor Force     Employed  Unemployed  \
0             Autauga County, AL  2016  25,649       24,297       1,352
1             Baldwin County, AL  2016  89,931       85,061       4,870
2             Barbour County, AL  2016   8,302        7,584         718
3                Bibb County, AL  2016   8,573        8,004         569
4              Blount County, AL  2016  24,525       23,171       1,354

   Unemployment Rate (%)
0                    5.3
1                    5.4
2                    8.6
3                    6.6
4                    5.5
```

[15]:
```
def convert_to_two_digit_code(code):
    c = "0000"+str(code)
    return(c[-2:])
```

4.0.2 3.2 Rename these two columns

Renaming of the column is done for better handlin of the colmn in the code

[16]:
```
usa_fips_code = usa_fips_code.rename(columns={"State FIPS Code":
 ↪"State_FIPS_Code","County FIPS Code":"County_FIPS_Code"})
usa_fips_code.head()
```

[16]:
```
         LAUS Code State_FIPS_Code County_FIPS_Code  \
0  CN0100100000000              01              001
1  CN0100300000000              01              003
2  CN0100500000000              01              005
3  CN0100700000000              01              007
4  CN0100900000000              01              009

  County Name/State Abbreviation  Year  Labor Force     Employed  Unemployed  \
0             Autauga County, AL  2016  25,649       24,297       1,352
1             Baldwin County, AL  2016  89,931       85,061       4,870
2             Barbour County, AL  2016   8,302        7,584         718
3                Bibb County, AL  2016   8,573        8,004         569
4              Blount County, AL  2016  24,525       23,171       1,354

   Unemployment Rate (%)
0                    5.3
1                    5.4
2                    8.6
3                    6.6
```

```
4                    5.5
```

4.0.3 3.3 Modify County Columns in the original data set

[17]:
```
def create_county_state(countyName,state):
    return countyName + " County, "+state
```

[18]:
```
road_accident_df_2.shape
```

[18]:
```
(3402756, 36)
```

[19]:
```
road_accident_df_2['countyState'] = create_county_state(road_accident_df_2.
↪County,road_accident_df_2.State)
```

[20]:
```
usa_fips_code = usa_fips_code.rename(columns={"County Name/State Abbreviation":␣
↪"countyState"})
usa_fips_code.head()
```

[20]:

	LAUS Code	State_FIPS_Code	County_FIPS_Code	countyState	Year	\
0	CN0100100000000	01	001	Autauga County, AL	2016	
1	CN0100300000000	01	003	Baldwin County, AL	2016	
2	CN0100500000000	01	005	Barbour County, AL	2016	
3	CN0100700000000	01	007	Bibb County, AL	2016	
4	CN0100900000000	01	009	Blount County, AL	2016	

	Labor Force	Employed	Unemployed	Unemployment Rate (%)
0	25,649	24,297	1,352	5.3
1	89,931	85,061	4,870	5.4
2	8,302	7,584	718	8.6
3	8,573	8,004	569	6.6
4	24,525	23,171	1,354	5.5

[21]:
```
usa_fips_code_df1 = usa_fips_code.
↪filter(["State_FIPS_Code","County_FIPS_Code","countyState"], axis=1)
usa_fips_code_df1.head()
```

[21]:

	State_FIPS_Code	County_FIPS_Code	countyState
0	01	001	Autauga County, AL
1	01	003	Baldwin County, AL
2	01	005	Barbour County, AL
3	01	007	Bibb County, AL
4	01	009	Blount County, AL

[22]:
```
usa_fips_code_df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3219 entries, 0 to 3218
```

```
Data columns (total 3 columns):
State_FIPS_Code     3219 non-null object
County_FIPS_Code    3219 non-null object
countyState         3219 non-null object
dtypes: object(3)
memory usage: 75.6+ KB
```

[23]:
```
merged_df_0 = pd.merge(road_accident_df_2, usa_fips_code_df1,␣
↪how="left",on="countyState")
```

[24]:
```
merged_df_0.head()
```

[24]:

	ID	Severity	Start_Time	End_Time	Start_Lat	\
0	A-1	3	2016-02-08 05:46:00	2016-02-08 11:00:00	39.865147	
1	A-2	2	2016-02-08 06:07:59	2016-02-08 06:37:59	39.928059	
2	A-3	2	2016-02-08 06:49:27	2016-02-08 07:19:27	39.063148	
3	A-4	3	2016-02-08 07:23:34	2016-02-08 07:53:34	39.747753	
4	A-5	2	2016-02-08 07:39:07	2016-02-08 08:09:07	39.627781	

	Start_Lng	Street	Side	City	County	…	\
0	-84.058723	I-70 E	R	Dayton	Montgomery	…	
1	-82.831184	Brice Rd	L	Reynoldsburg	Franklin	…	
2	-84.032608	State Route 32	R	Williamsburg	Clermont	…	
3	-84.205582	I-75 S	R	Dayton	Montgomery	…	
4	-84.188354	Miamisburg Centerville Rd	R	Dayton	Montgomery	…	

	Roundabout	Station	Stop	Traffic_Calming	Traffic_Signal	Turning_Loop	\
0	False	False	False	False	False	False	
1	False	False	False	False	False	False	
2	False	False	False	False	True	False	
3	False	False	False	False	False	False	
4	False	False	False	False	True	False	

	Sunrise_Sunset	countyState	State_FIPS_Code	County_FIPS_Code
0	Night	Montgomery County, OH	39	113
1	Night	Franklin County, OH	39	049
2	Night	Clermont County, OH	39	025
3	Night	Montgomery County, OH	39	113
4	Day	Montgomery County, OH	39	113

[5 rows x 39 columns]

[25]:
```
merged_df_0.shape
```

[25]:
```
(3402756, 39)
```

[26]:
```
merged_df_1 = merged_df_0.dropna()
merged_df_1.shape
```

[26]:
```
(3232251, 39)
```

[27]:
```
merged_df_1.head()
```

[27]:
```
     ID  Severity           Start_Time             End_Time  Start_Lat  \
0  A-1         3  2016-02-08 05:46:00  2016-02-08 11:00:00  39.865147
1  A-2         2  2016-02-08 06:07:59  2016-02-08 06:37:59  39.928059
2  A-3         2  2016-02-08 06:49:27  2016-02-08 07:19:27  39.063148
3  A-4         3  2016-02-08 07:23:34  2016-02-08 07:53:34  39.747753
4  A-5         2  2016-02-08 07:39:07  2016-02-08 08:09:07  39.627781

   Start_Lng                     Street Side          City      County  …  \
0 -84.058723                     I-70 E    R        Dayton  Montgomery  …
1 -82.831184                  Brice Rd     L  Reynoldsburg    Franklin  …
2 -84.032608             State Route 32    R  Williamsburg    Clermont  …
3 -84.205582                     I-75 S    R        Dayton  Montgomery  …
4 -84.188354  Miamisburg Centerville Rd    R        Dayton  Montgomery  …

  Roundabout Station   Stop Traffic_Calming Traffic_Signal Turning_Loop  \
0      False   False  False           False          False        False
1      False   False  False           False          False        False
2      False   False  False           False           True        False
3      False   False  False           False          False        False
4      False   False  False           False           True        False

   Sunrise_Sunset            countyState  State_FIPS_Code  County_FIPS_Code
0           Night  Montgomery County, OH               39               113
1           Night    Franklin County, OH               39               049
2           Night    Clermont County, OH               39               025
3           Night  Montgomery County, OH               39               113
4             Day  Montgomery County, OH               39               113

[5 rows x 39 columns]
```

5 4.WRITING O/P TO PREPROCESSING FOLDER

[28]:
```
merged_df_1.to_csv(output_path_preprocessing + "preprocessed_us_accident_data.
 ↪csv")
print("A file is written in pre_proccessing folder of the data !!! with name␣
 ↪preprocessed_us_accident_data.csv")
```

```
A file is written in pre_proccessing folder of the data !!! with name
preprocessed_us_accident_data.csv
```