```
# IMPORTANT: RUN THIS CELL IN ORDER TO IMPORT YOUR KAGGLE DATA SOURCES
# TO THE CORRECT LOCATION (/kaggle/input) IN YOUR NOTEBOOK,
# THEN FEEL FREE TO DELETE THIS CELL.
# NOTE: THIS NOTEBOOK ENVIRONMENT DIFFERS FROM KAGGLE'S PYTHON
# ENVIRONMENT SO THERE MAY BE MISSING LIBRARIES USED BY YOUR
# NOTEBOOK.

import os
import sys
from tempfile import NamedTemporaryFile
from urllib.request import urlopen
from urllib.parse import unquote, urlparse
from urllib.error import HTTPError
from zipfile import ZipFile
import tarfile
import shutil

CHUNK_SIZE = 40960
DATA_SOURCE_MAPPING = 'us-accidents:https%3A%2F%2Fstorage.googleapis.com%2Fkaggle-data-sets%2F199387%2F5793796%2Fbundle%2Farchive.zip%3FX-Goog-Algorithm%3DGOOG4-RSA-SHA25

KAGGLE_INPUT_PATH='/kaggle/input'
KAGGLE_WORKING_PATH='/kaggle/working'
KAGGLE_SYMLINK='kaggle'

!umount /kaggle/input/ 2> /dev/null
shutil.rmtree('/kaggle/input', ignore_errors=True)
os.makedirs(KAGGLE_INPUT_PATH, 0o777, exist_ok=True)
os.makedirs(KAGGLE_WORKING_PATH, 0o777, exist_ok=True)

try:
  os.symlink(KAGGLE_INPUT_PATH, os.path.join("..", 'input'), target_is_directory=True)
except FileExistsError:
  pass
try:
  os.symlink(KAGGLE_WORKING_PATH, os.path.join("..", 'working'), target_is_directory=True)
except FileExistsError:
  pass

for data_source_mapping in DATA_SOURCE_MAPPING.split(','):
    directory, download_url_encoded = data_source_mapping.split(':')
    download_url = unquote(download_url_encoded)
    filename = urlparse(download_url).path
    destination_path = os.path.join(KAGGLE_INPUT_PATH, directory)
    try:
        with urlopen(download_url) as fileres, NamedTemporaryFile() as tfile:
            total_length = fileres.headers['content-length']
            print(f'Downloading {directory}, {total_length} bytes compressed')
            dl = 0
            data = fileres.read(CHUNK_SIZE)
            while len(data) > 0:
                dl += len(data)
                tfile.write(data)
                done = int(50 * dl / int(total_length))
                sys.stdout.write(f"\r[{'=' * done}{' ' * (50-done)}] {dl} bytes downloaded")
                sys.stdout.flush()
                data = fileres.read(CHUNK_SIZE)
            if filename.endswith('.zip'):
              with ZipFile(tfile) as zfile:
                zfile.extractall(destination_path)
            else:
              with tarfile.open(tfile.name) as tarfile:
                tarfile.extractall(destination_path)
            print(f'\nDownloaded and uncompressed: {directory}')
    except HTTPError as e:
        print(f'Failed to load (likely expired) {download_url} to path {destination_path}')
        continue
    except OSError as e:
        print(f'Failed to load {download_url} to path {destination_path}')
        continue

print('Data source import complete.')
```

```
Downloading us-accidents, 684855912 bytes compressed
[==================================================] 684855912 bytes downloaded
Downloaded and uncompressed: us-accidents
Data source import complete.
```

```
#importing the library
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
import tensorflow as tf


#loading the dataset
df=pd.read_csv('/kaggle/input/us-accidents/US_Accidents_March23.csv',nrows=10000)


df
```

	ID	Source	Severity	Start_Time	End_Time	Start_Lat	Start_Lng	End_Lat	End_Lng	Distance(mi)
0	A-1	Source2	3	2016-02-08 05:46:00	2016-02-08 11:00:00	39.865147	-84.058723	NaN	NaN	0.01
1	A-2	Source2	2	2016-02-08 06:07:59	2016-02-08 06:37:59	39.928059	-82.831184	NaN	NaN	0.01
2	A-3	Source2	2	2016-02-08 06:49:27	2016-02-08 07:19:27	39.063148	-84.032608	NaN	NaN	0.01
3	A-4	Source2	3	2016-02-08 07:23:34	2016-02-08 07:53:34	39.747753	-84.205582	NaN	NaN	0.01
4	A-5	Source2	2	2016-02-08 07:39:07	2016-02-08 08:09:07	39.627781	-84.188354	NaN	NaN	0.01
...	...	...	...	...	...	...	...	...	...	...
9995	A-9996	Source2	2	2017-01-06 16:01:06	2017-01-06 16:30:34	38.701267	-121.077751	NaN	NaN	0.00
9996	A-9997	Source2	2	2017-01-06 16:14:00	2017-01-06 16:43:38	36.981407	-122.011192	NaN	NaN	0.01
9997	A-9998	Source2	3	2017-01-06 16:08:58	2017-01-06 16:38:48	37.326691	-121.940720	NaN	NaN	0.01
9998	A-9999	Source2	3	2017-01-06 16:25:01	2017-01-06 16:54:51	37.930088	-122.324036	NaN	NaN	0.01
9999	A-10000	Source2	3	2017-01-06 16:22:04	2017-01-06 16:51:29	38.574406	-121.577354	NaN	NaN	0.01

10000 rows × 46 columns

```
#getting the preliminary information about the dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 46 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   ID                     10000 non-null  object
 1   Source                 10000 non-null  object
 2   Severity               10000 non-null  int64
 3   Start_Time             10000 non-null  object
 4   End_Time               10000 non-null  object
 5   Start_Lat              10000 non-null  float64
 6   Start_Lng              10000 non-null  float64
 7   End_Lat                0 non-null      float64
 8   End_Lng                0 non-null      float64
 9   Distance(mi)           10000 non-null  float64
 10  Description            10000 non-null  object
 11  Street                 10000 non-null  object
 12  City                   10000 non-null  object
 13  County                 10000 non-null  object
 14  State                  10000 non-null  object
 15  Zipcode                9999 non-null   object
 16  Country                10000 non-null  object
 17  Timezone               9999 non-null   object
 18  Airport_Code           9999 non-null   object
 19  Weather_Timestamp      9969 non-null   object
 20  Temperature(F)         9915 non-null   float64
 21  Wind_Chill(F)          1426 non-null   float64
 22  Humidity(%)            9876 non-null   float64
 23  Pressure(in)           9965 non-null   float64
 24  Visibility(mi)         9904 non-null   float64
 25  Wind_Direction         9969 non-null   object
 26  Wind_Speed(mph)        8226 non-null   float64
 27  Precipitation(in)      1123 non-null   float64
 28  Weather_Condition      9923 non-null   object
 29  Amenity                10000 non-null  bool
 30  Bump                   10000 non-null  bool
 31  Crossing               10000 non-null  bool
 32  Give_Way               10000 non-null  bool
 33  Junction               10000 non-null  bool
 34  No_Exit                10000 non-null  bool
 35  Railway                10000 non-null  bool
 36  Roundabout             10000 non-null  bool
 37  Station                10000 non-null  bool
 38  Stop                   10000 non-null  bool
 39  Traffic_Calming        10000 non-null  bool
 40  Traffic_Signal         10000 non-null  bool
 41  Turning_Loop           10000 non-null  bool
 42  Sunrise_Sunset         10000 non-null  object
 43  Civil_Twilight         10000 non-null  object
 44  Nautical_Twilight      10000 non-null  object
 45  Astronomical_Twilight  10000 non-null  object
dtypes: bool(13), float64(12), int64(1), object(20)
memory usage: 2.6+ MB
```

```
#checking for missing values
df.isna().sum()
```

```
ID                          0
Source                      0
Severity                    0
Start_Time                  0
End_Time                    0
```

```
Start_Lat                   0
Start_Lng                   0
End_Lat                 10000
End_Lng                 10000
Distance(mi)                0
Description                 0
Street                      0
City                        0
County                      0
State                       0
Zipcode                     1
Country                     0
Timezone                    1
Airport_Code                1
Weather_Timestamp          31
Temperature(F)             85
Wind_Chill(F)            8574
Humidity(%)               124
Pressure(in)               35
Visibility(mi)             96
Wind_Direction             31
Wind_Speed(mph)          1774
Precipitation(in)        8877
Weather_Condition           77
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              0
Civil_Twilight              0
Nautical_Twilight           0
Astronomical_Twilight       0
dtype: int64
```

```
df.isna().mean()
```

```
ID                      0.0000
Source                  0.0000
Severity                0.0000
Start_Time              0.0000
End_Time                0.0000
Start_Lat               0.0000
Start_Lng               0.0000
End_Lat                 1.0000
End_Lng                 1.0000
Distance(mi)            0.0000
Description             0.0000
Street                  0.0000
City                    0.0000
County                  0.0000
State                   0.0000
Zipcode                 0.0001
Country                 0.0000
Timezone                0.0001
Airport_Code            0.0001
Weather_Timestamp       0.0031
Temperature(F)          0.0085
Wind_Chill(F)           0.8574
Humidity(%)             0.0124
Pressure(in)            0.0035
Visibility(mi)          0.0096
Wind_Direction          0.0031
Wind_Speed(mph)         0.1774
Precipitation(in)       0.8877
Weather_Condition       0.0077
Amenity                 0.0000
Bump                    0.0000
Crossing                0.0000
Give_Way                0.0000
Junction                0.0000
No_Exit                 0.0000
Railway                 0.0000
Roundabout              0.0000
Station                 0.0000
Stop                    0.0000
Traffic_Calming         0.0000
Traffic_Signal          0.0000
Turning_Loop            0.0000
Sunrise_Sunset          0.0000
Civil_Twilight          0.0000
Nautical_Twilight       0.0000
Astronomical_Twilight   0.0000
dtype: float64
```

```
null_columns=['End_Lat','End_Lng','Precipitation(in)','Wind_Chill(F)']
df=df.drop(null_columns,axis=1)
```

```
df
```

0	A-1	Source2	3	2016-02-08 05:46:00	2016-02-08 11:00:00	39.865147	-84.058723	0.01	Right lane blocked due to accident on I-70 Eas...
1	A-2	Source2	2	2016-02-08 06:07:59	2016-02-08 06:37:59	39.928059	-82.831184	0.01	Accident on Brice Rd at Tussing Rd. Expect del...
2	A-3	Source2	2	2016-02-08 06:49:27	2016-02-08 07:19:27	39.063148	-84.032608	0.01	Accident on OH-32 State Route 32 Westbound at ...
3	A-4	Source2	3	2016-02-08 07:23:34	2016-02-08 07:53:34	39.747753	-84.205582	0.01	Accident on I-75 Southbound at Exits 52 52B US...
4	A-5	Source2	2	2016-02-08 07:39:07	2016-02-08 08:09:07	39.627781	-84.188354	0.01	Accident on McEwen Rd at OH-725 Miamisburg Cen...
...	...	...	...	...	...	...	...	...	...
9995	A-9996	Source2	2	2017-01-06 16:01:06	2017-01-06 16:30:34	38.701267	-121.077751	0.00	Accident on Brackenwood Pl at Melrose Way.
9996	A-9997	Source2	2	2017-01-06 16:14:00	2017-01-06 16:43:38	36.981407	-122.011192	0.01	Accident on Water St at Benito Ave.
9997	A-9998	Source2	3	2017-01-06 16:08:58	2017-01-06 16:38:48	37.326691	-121.940720	0.01	Accident on I-880 Northbound at Exits 1A 1B 1C...
9998	A-9999	Source2	3	2017-01-06 16:25:01	2017-01-06 16:54:51	37.930088	-122.324036	0.01	Accident on I-80 Eastbound at Exit 15 Cutting ...
9999	A-10000	Source2	3	2017-01-06 16:22:04	2017-01-06 16:51:29	38.574406	-121.577354	0.01	Right hand shoulder blocked due to accident on...

10000 rows × 42 columns

```
df=df.dropna(axis=0).reset_index(drop=True)
```

```
df
```

				06:49:27	07:19:27				Westbound at ...	
1	A-4	Source2	3	2016-02-08 07:23:34	2016-02-08 07:53:34	39.747753	-84.205582	0.01	Accident on I-75 Southbound at Exits 52 52B US...	
2	A-5	Source2	2	2016-02-08 07:39:07	2016-02-08 08:09:07	39.627781	-84.188354	0.01	Accident on McEwen Rd at OH-725 Miamisburg Cen...	Mi C
3	A-6	Source2	3	2016-02-08 07:44:26	2016-02-08 08:14:26	40.100590	-82.925194	0.01	Accident on I-270 Outerbelt Northbound near Ex...	W
4	A-7	Source2	2	2016-02-08 07:59:35	2016-02-08 08:29:35	39.758274	-84.230507	0.00	Accident on Oakridge Dr at Woodward Ave. Expec...	W
...	...	...	...	...	...	...	...	...	...	
8125	A-9995	Source3	2	2017-01-06 16:00:55	2017-01-06 16:30:43	37.066490	-121.219147	0.01	Accident on CA-152 Pacheco Pass Hwy Westbound ...	F
8126	A-9997	Source2	2	2017-01-06 16:14:00	2017-01-06 16:43:38	36.981407	-122.011192	0.01	Accident on Water St at Benito Ave.	
8127	A-9998	Source2	3	2017-01-06 16:08:58	2017-01-06 16:38:48	37.326691	-121.940720	0.01	Accident on I-880 Northbound at Exits 1A 1B 1C...	
8128	A-9999	Source2	3	2017-01-06 16:25:01	2017-01-06 16:54:51	37.930088	-122.324036	0.01	Accident on I-80 Eastbound at Exit 15 Cutting ...	
8129	A-10000	Source2	3	2017-01-06 16:22:04	2017-01-06 16:51:29	38.574406	-121.577354	0.01	Right hand shoulder blocked due to accident on...	

8130 rows × 42 columns

```
df.isna().sum()
```

```
#checking for unique values in each columns
{column:len(df[column].unique()) for column in df.columns if df.dtypes[column]=='object'}
```

```
#removing unnecessary column

unneeded_columns=['ID','Description','Street','City','Zipcode','Country']
```

```
df=df.drop(unneeded_columns,axis=1)
```

```
df
```

	Source	Severity	Start_Time	End_Time	Start_Lat	Start_Lng	Distance(mi)	County	State	Time:
0	Source2	2	2016-02-08 06:49:27	2016-02-08 07:19:27	39.063148	-84.032608	0.01	Clermont	OH	US/Eas
1	Source2	3	2016-02-08 07:23:34	2016-02-08 07:53:34	39.747753	-84.205582	0.01	Montgomery	OH	US/Eas
2	Source2	2	2016-02-08 07:39:07	2016-02-08 08:09:07	39.627781	-84.188354	0.01	Montgomery	OH	US/Eas
3	Source2	3	2016-02-08 07:44:26	2016-02-08 08:14:26	40.100590	-82.925194	0.01	Franklin	OH	US/Eas
4	Source2	2	2016-02-08 07:59:35	2016-02-08 08:29:35	39.758274	-84.230507	0.00	Montgomery	OH	US/Eas
...	...	...	...	...	...	...	...	...	...	
8125	Source3	2	2017-01-06 16:00:55	2017-01-06 16:30:43	37.066490	-121.219147	0.01	Santa Clara	CA	US/Pa
8126	Source2	2	2017-01-06 16:14:00	2017-01-06 16:43:38	36.981407	-122.011192	0.01	Santa Cruz	CA	US/Pa
8127	Source2	3	2017-01-06 16:08:58	2017-01-06 16:38:48	37.326691	-121.940720	0.01	Santa Clara	CA	US/Pa
8128	Source2	3	2017-01-06 16:25:01	2017-01-06 16:54:51	37.930088	-122.324036	0.01	Contra Costa	CA	US/Pa
8129	Source2	3	2017-01-06 16:22:04	2017-01-06 16:51:29	38.574406	-121.577354	0.01	Yolo	CA	US/Pa

8130 rows × 36 columns

```
def get_years(df,column):
    return df[column].apply(lambda date:date[0:4])
def get_months(df,column):
    return df[column].apply(lambda date:date[5:7])
```

```
df['Start_time_month']=get_months(df,'Start_Time')
df['Start_time_year']=get_years(df,'Start_Time')
df['End_time_month']=get_months(df,'End_Time')
df['End_time_year']=get_years(df,'End_Time')
df['Weather_timestamp_month']=get_months(df,'Weather_Timestamp')
df['Weather_timestamp_year']=get_years(df,'Weather_Timestamp')
```

```
df=df.drop(['Start_Time','End_Time','Weather_Timestamp'],axis=1)
```

```
df
```

	Source	Severity	Start_Lat	Start_Lng	Distance(mi)	County	State	Timezone	Airport_Code	Temperature(F)	...	Sunrise_Sunset	Civil_Twilight	Nautical_
0	Source2	2	39.063148	-84.032608	0.01	Clermont	OH	US/Eastern	KI69	36.0	...	Night	Night	
1	Source2	3	39.747753	-84.205582	0.01	Montgomery	OH	US/Eastern	KDAY	35.1	...	Night	Day	
2	Source2	2	39.627781	-84.188354	0.01	Montgomery	OH	US/Eastern	KMGY	36.0	...	Day	Day	
3	Source2	3	40.100590	-82.925194	0.01	Franklin	OH	US/Eastern	KCMH	37.9	...	Day	Day	
4	Source2	2	39.758274	-84.230507	0.00	Montgomery	OH	US/Eastern	KDAY	34.0	...	Day	Day	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8125	Source3	2	37.066490	-121.219147	0.01	Santa Clara	CA	US/Pacific	KCVH	48.2	...	Day	Day	
8126	Source2	2	36.981407	-122.011192	0.01	Santa Cruz	CA	US/Pacific	KWVI	52.0	...	Day	Day	
8127	Source2	3	37.326691	-121.940720	0.01	Santa Clara	CA	US/Pacific	KSJC	51.1	...	Day	Day	
8128	Source2	3	37.930088	-122.324036	0.01	Contra Costa	CA	US/Pacific	KCCR	44.1	...	Day	Day	
8129	Source2	3	38.574406	-121.577354	0.01	Yolo	CA	US/Pacific	KSAC	46.0	...	Day	Day	

8130 rows × 39 columns

```
def onehot_encode(df,columns):
    df=df.copy()
    for column in columns:
        dummies=pd.get_dummies(df[column],prefix=column)
        df=pd.concat([df,dummies],axis=1)
        df=df.drop(column,axis=1)
    return df
```

```
df=onehot_encode(df,columns=['County','State','Timezone','Airport_Code','Wind_Direction','Weather_Condition'])
```

```
df
```

	Source	Severity	Start_Lat	Start_Lng	Distance(mi)	Temperature(F)	Humidity(%)	Pressure(in)	Visibility(mi)	Wind_Speed(mph)	...	Weather_Condition_Mist	W
0	Source2	2	39.063148	-84.032608	0.01	36.0	100.0	29.67	10.0	3.5	...	0	
1	Source2	3	39.747753	-84.205582	0.01	35.1	96.0	29.64	9.0	4.6	...	0	
2	Source2	2	39.627781	-84.188354	0.01	36.0	89.0	29.65	6.0	3.5	...	0	
3	Source2	3	40.100590	-82.925194	0.01	37.9	97.0	29.63	7.0	3.5	...	0	
4	Source2	2	39.758274	-84.230507	0.00	34.0	100.0	29.66	7.0	3.5	...	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
8125	Source3	2	37.066490	-121.219147	0.01	48.2	62.0	30.05	10.0	11.5	...	0	
8126	Source2	2	36.981407	-122.011192	0.01	52.0	59.0	30.05	10.0	6.9	...	0	
8127	Source2	3	37.326691	-121.940720	0.01	51.1	50.0	30.04	10.0	5.8	...	0	
8128	Source2	3	37.930088	-122.324036	0.01	44.1	63.0	30.04	10.0	5.8	...	0	
8129	Source2	3	38.574406	-121.577354	0.01	46.0	71.0	30.09	10.0	8.1	...	0	

8130 rows × 215 columns

```
def get_binary_column(df,column):
    if column=='Source':
        return df[column].apply(lambda x:1 if x=='MapQuest' else 0)
    else:
        return df[column].apply(lambda x:1 if x=='Day' else 0)


df['Source']=get_binary_column(df,'Source')
df['Sunrise_Sunset']=get_binary_column(df,'Sunrise_Sunset')
df['Civil_Twilight']=get_binary_column(df,'Civil_Twilight')
df['Nautical_Twilight']=get_binary_column(df,'Nautical_Twilight')
df['Astronomical_Twilight']=get_binary_column(df,'Astronomical_Twilight')
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
df
```

	Source	Severity	Start_Lat	Start_Lng	Distance(mi)	Temperature(F)	Humidity(%)	Pressure(in)	Visibility(mi)	Wind_Speed(mph)	...	Weather_Condition_Mist	W
0	0	2	39.063148	-84.032608	0.01	36.0	100.0	29.67	10.0	3.5	...	0	
1	0	3	39.747753	-84.205582	0.01	35.1	96.0	29.64	9.0	4.6	...	0	
2	0	2	39.627781	-84.188354	0.01	36.0	89.0	29.65	6.0	3.5	...	0	
3	0	3	40.100590	-82.925194	0.01	37.9	97.0	29.63	7.0	3.5	...	0	
4	0	2	39.758274	-84.230507	0.00	34.0	100.0	29.66	7.0	3.5	...	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
8125	0	2	37.066490	-121.219147	0.01	48.2	62.0	30.05	10.0	11.5	...	0	
8126	0	2	36.981407	-122.011192	0.01	52.0	59.0	30.05	10.0	6.9	...	0	
8127	0	3	37.326691	-121.940720	0.01	51.1	50.0	30.04	10.0	5.8	...	0	
8128	0	3	37.930088	-122.324036	0.01	44.1	63.0	30.04	10.0	5.8	...	0	
8129	0	3	38.574406	-121.577354	0.01	46.0	71.0	30.09	10.0	8.1	...	0	

8130 rows × 215 columns

```
#Spliting and Scaling the dataset
y=df['Severity'].copy()
x=df.drop('Severity',axis=1).copy()
```

```
y.unique()
```

```
array([2, 3, 1, 4])
```

```
x
```

	Source	Start_Lat	Start_Lng	Distance(mi)	Temperature(F)	Humidity(%)	Pressure(in)	Visibility(mi)	Wind_Speed(mph)	Amenity	...	Weather_Condition_Mist	We
0	0	39.063148	-84.032608	0.01	36.0	100.0	29.67	10.0	3.5	False	...	0	
2	0	39.627781	-84.188354	0.01	36.0	89.0	29.65	6.0	3.5	False		0	

```
y=y-1
```

```
y.unique()
```

```
array([1, 2, 0, 3])
```

```
x=x.astype(float)
```

```
x
```

	Source	Start_Lat	Start_Lng	Distance(mi)	Temperature(F)	Humidity(%)	Pressure(in)	Visibility(mi)	Wind_Speed(mph)	Amenity	...	Weather_Condition_Mist	We
0	0.0	39.063148	-84.032608	0.01	36.0	100.0	29.67	10.0	3.5	0.0	...	0.0	
1	0.0	39.747753	-84.205582	0.01	35.1	96.0	29.64	9.0	4.6	0.0	...	0.0	
2	0.0	39.627781	-84.188354	0.01	36.0	89.0	29.65	6.0	3.5	0.0	...	0.0	
3	0.0	40.100590	-82.925194	0.01	37.9	97.0	29.63	7.0	3.5	0.0	...	0.0	
4	0.0	39.758274	-84.230507	0.00	34.0	100.0	29.66	7.0	3.5	0.0	...	0.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
8125	0.0	37.066490	-121.219147	0.01	48.2	62.0	30.05	10.0	11.5	0.0	...	0.0	
8126	0.0	36.981407	-122.011192	0.01	52.0	59.0	30.05	10.0	6.9	0.0	...	0.0	
8127	0.0	37.326691	-121.940720	0.01	51.1	50.0	30.04	10.0	5.8	0.0	...	0.0	
8128	0.0	37.930088	-122.324036	0.01	44.1	63.0	30.04	10.0	5.8	0.0	...	0.0	
8129	0.0	38.574406	-121.577354	0.01	46.0	71.0	30.09	10.0	8.1	0.0	...	0.0	

8130 rows × 214 columns

```
scaler=StandardScaler()
x=pd.DataFrame(scaler.fit_transform(x),columns=x.columns)
```

```
x
```

	Source	Start_Lat	Start_Lng	Distance(mi)	Temperature(F)	Humidity(%)	Pressure(in)	Visibility(mi)	Wind_Speed(mph)	Amenity	...	Weather_Condition_Mist	We
0	0.0	1.190386	3.218087	-0.027001	-1.440270	1.534788	-0.948900	0.409723	-1.169373	-0.103398	...	-0.027176	