

## 2.0-EDA And FE Algerian Forest Fires By Virat Tiwari

December 8, 2023

### 1 EDA AND FE BY VIRAT TIWARI

#### 1.1 Topic - Algerian Forest Fires Dataset

Data Set Information:

The dataset includes 244 instances that regroup a data of two regions of Algeria, namely the Bejaia region located in the northeast of Algeria and the Sidi Bel-abbes region located in the northwest of Algeria.

122 instances for each region.

The period from June 2012 to September 2012. The dataset includes 11 attributes and 1 output attribute (class) The 244 instances have been classified into fire(138 classes) and not fire (106 classes) classes.

Attribute Information:

1. Date : (DD/MM/YYYY) Day, month ('june' to 'september'), year (2012) Weather data observations
2. Temp : temperature noon (temperature max) in Celsius degrees: 22 to 42
3. RH : Relative Humidity in %: 21 to 90
4. Ws :Wind speed in km/h: 6 to 29
5. Rain: total day in mm: 0 to 16.8 FWI Components
6. Fine Fuel Moisture Code (FFMC) index from the FWI system: 28.6 to 92.5
7. Duff Moisture Code (DMC) index from the FWI system: 1.1 to 65.9
8. Drought Code (DC) index from the FWI system: 7 to 220.4
9. Initial Spread Index (ISI) index from the FWI system: 0 to 18.5
10. Buildup Index (BUI) index from the FWI system: 1.1 to 68
11. Fire Weather Index (FWI) Index: 0 to 31.1
12. Classes: two classes, namely Fire and not Fire

```
[54]: # Here we import some of the very important libraries that helpful in entire ↵  
      ↪eda and project as well
```

```
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns  
%matplotlib inline
```

```
[55]: # Here we import dataset that is not cleaned
# pd.read_csv ( ) function is used for reading the dataset

dataset=pd.read_csv('Algerian_forest_fires_dataset_UPDATE.csv' ,header=1)
```

```
[56]: # . shape ( ) function is used for getting the rows and columns of dataset
# rows , columns = 246 , 14

dataset.shape
```

```
[56]: (246, 14)
```

```
[57]: # . head ( ) function is iuse for initialing the five datapoints from the
↳ dataset

dataset.head()
```

```
[57]:   day month  year Temperature  RH  Ws Rain  FFMC  DMC  DC  ISI  BUI  FWI  \
0   01    06  2012           29  57  18    0  65.7  3.4  7.6  1.3  3.4  0.5
1   02    06  2012           29  61  13   1.3  64.4  4.1  7.6   1  3.9  0.4
2   03    06  2012           26  82  22  13.1  47.1  2.5  7.1  0.3  2.7  0.1
3   04    06  2012           25  89  13   2.5  28.6  1.3  6.9   0  1.7   0
4   05    06  2012           27  77  16    0  64.8   3  14.2  1.2  3.9  0.5

      Classes
0  not fire
1  not fire
2  not fire
3  not fire
4  not fire
```

```
[58]: dataset['month'].unique()
```

```
[58]: array(['06', '07', '08', '09', nan, 'month'], dtype=object)
```

```
[ ]:
```

```
[59]: # . info ( ) function gives the information of dataset

dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 246 entries, 0 to 245
Data columns (total 14 columns):
#   Column          Non-Null Count  Dtype
---  -
#   Column          Non-Null Count  Dtype
```

```

0   day          246 non-null   object
1   month        245 non-null   object
2   year         245 non-null   object
3   Temperature  245 non-null   object
4   RH           245 non-null   object
5   Ws           245 non-null   object
6   Rain         245 non-null   object
7   FFMFC        245 non-null   object
8   DMC          245 non-null   object
9   DC           245 non-null   object
10  ISI          245 non-null   object
11  BUI          245 non-null   object
12  FWI          245 non-null   object
13  Classes      244 non-null   object
dtypes: object(14)
memory usage: 27.0+ KB

```

## 1.2 Data Cleaning

```

[60]: # Check Missing Values
      # This is how we get the rows in which null values are present

dataset[dataset.isnull().any(axis=1)]

```

```

[60]:
      day month  year Temperature  RH  Ws Rain  \
122  Sidi-Bel Abbas Region Dataset  NaN  NaN      NaN  NaN  NaN  NaN
167      14    07  2012      37   37   18   0.2

      FFMFC  DMC    DC  ISI  BUI    FWI Classes
122   NaN   NaN   NaN  NaN  NaN   NaN     NaN
167  88.9  12.9  14.6  9  12.5  10.4  fire     NaN

```

The dataset is converted into two sets based on Region from 122th index, we can make a new column based on the Region

1 : “Bejaia Region Dataset”

2 : “Sidi-Bel Abbas Region Dataset”

Add new column with region

```

[61]: # Here we initialising the values 0 and 1 according to Region
      # Those regions are comes before 122 are denoted by 0
      # Those regions are comes after 122 are denoted by 1

dataset.loc[:122, "Region"]=0
dataset.loc[122:, "Region"]=1
df=dataset

```

```
[62]: # Again check the information of dataset like datatypes and datapoints etc

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 246 entries, 0 to 245
Data columns (total 15 columns):
#   Column          Non-Null Count  Dtype
---  -
0   day              246 non-null   object
1   month            245 non-null   object
2   year             245 non-null   object
3   Temperature      245 non-null   object
4   RH               245 non-null   object
5   Ws               245 non-null   object
6   Rain             245 non-null   object
7   FFMC             245 non-null   object
8   DMC              245 non-null   object
9   DC              245 non-null   object
10  ISI              245 non-null   object
11  BUI              245 non-null   object
12  FWI              245 non-null   object
13  Classes          244 non-null   object
14  Region           246 non-null   float64
dtypes: float64(1), object(14)
memory usage: 29.0+ KB
```

```
[63]: df.head()
```

```
[63]:   day month  year Temperature  RH  Ws Rain  FFMC  DMC  DC  ISI  BUI  FWI  \
0  01    06  2012           29  57  18    0  65.7  3.4  7.6  1.3  3.4  0.5
1  02    06  2012           29  61  13   1.3  64.4  4.1  7.6   1  3.9  0.4
2  03    06  2012           26  82  22  13.1  47.1  2.5  7.1  0.3  2.7  0.1
3  04    06  2012           25  89  13   2.5  28.6  1.3  6.9   0  1.7   0
4  05    06  2012           27  77  16    0  64.8   3  14.2  1.2  3.9  0.5

      Classes  Region
0  not fire      0.0
1  not fire      0.0
2  not fire      0.0
3  not fire      0.0
4  not fire      0.0
```

```
[64]: # Here we change the datatype of "Region" from floating to integer
# . astype ( ) function is used for changing the datatype values

df[['Region']] = df[['Region']].astype(int)
```

```
[65]: df.head()
```

```
[65]:   day month  year Temperature  RH  Ws Rain  FFMC  DMC  DC  ISI  BUI  FWI  \
0   01    06  2012           29  57  18     0  65.7  3.4  7.6  1.3  3.4  0.5
1   02    06  2012           29  61  13     1.3  64.4  4.1  7.6   1  3.9  0.4
2   03    06  2012           26  82  22    13.1  47.1  2.5  7.1  0.3  2.7  0.1
3   04    06  2012           25  89  13     2.5  28.6  1.3  6.9   0  1.7   0
4   05    06  2012           27  77  16     0  64.8   3  14.2  1.2  3.9  0.5

      Classes  Region
0  not fire      0
1  not fire      0
2  not fire      0
3  not fire      0
4  not fire      0
```

```
[66]: df.tail()
```

```
[66]:   day month  year Temperature  RH  Ws Rain  FFMC  DMC  DC  ISI  BUI  \
241  26    09  2012           30  65  14     0  85.4  16  44.5  4.5  16.9
242  27    09  2012           28  87  15     4.4  41.1  6.5   8  0.1   6.2
243  28    09  2012           27  87  29     0.5  45.9  3.5   7.9  0.4   3.4
244  29    09  2012           24  54  18     0.1  79.7  4.3  15.2  1.7   5.1
245  30    09  2012           24  64  15     0.2  67.3  3.8  16.5  1.2   4.8

      FWI  Classes  Region
241  6.5    fire      1
242   0  not fire      1
243  0.2  not fire      1
244  0.7  not fire      1
245  0.5  not fire      1
```

```
[67]: # Here we check the null values or missing values
```

```
df.isnull().sum()
```

```
[67]: day          0
      month       1
      year       1
      Temperature 1
      RH          1
      Ws          1
      Rain        1
      FFMC        1
      DMC         1
      DC          1
      ISI         1
```

```

BUI          1
FWI          1
Classes      2
Region       0
dtype: int64

```

```
[68]: # Here we drop or remove missing values
```

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

```
[69]: df.head()
```

```
[69]:
```

	day	month	year	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI	\
0	01	06	2012	29	57	18	0	65.7	3.4	7.6	1.3	3.4	0.5	
1	02	06	2012	29	61	13	1.3	64.4	4.1	7.6	1	3.9	0.4	
2	03	06	2012	26	82	22	13.1	47.1	2.5	7.1	0.3	2.7	0.1	
3	04	06	2012	25	89	13	2.5	28.6	1.3	6.9	0	1.7	0	
4	05	06	2012	27	77	16	0	64.8	3	14.2	1.2	3.9	0.5	

	Classes	Region
0	not fire	0
1	not fire	0
2	not fire	0
3	not fire	0
4	not fire	0

```
[70]: # Now we dataset have 0 null values
```

```
df.isnull().sum()
```

```
[70]:
```

day	0
month	0
year	0
Temperature	0
RH	0
Ws	0
Rain	0
FFMC	0
DMC	0
DC	0
ISI	0
BUI	0
FWI	0
Classes	0
Region	0

```
dtype: int64
```

```
[71]: # row [122] is blank row so we delete that row

df.iloc[[122]]
```

```
[71]:      day month year Temperature  RH  Ws Rain  FFMC  DMC  DC  ISI  BUI  \
122  day month year Temperature  RH  Ws Rain  FFMC  DMC  DC  ISI  BUI

      FWI  Classes      Region
122  FWI  Classes          1
```

```
[72]: # Here we remove the 122 row that is blank
# . drop ( ) function is used to remove the row

df=df.drop(122).reset_index(drop=True)
```

```
[73]: df.iloc[[122]]
```

```
[73]:      day month year Temperature  RH  Ws Rain  FFMC  DMC  DC  ISI  BUI  FWI  \
122   01     06  2012           32  71  12   0.7  57.1  2.5  8.2  0.6  2.8  0.2

      Classes      Region
122  not fire          1
```

```
[74]: # Here we check the all the columns

df.columns
```

```
[74]: Index(['day', 'month', 'year', 'Temperature', ' RH', ' Ws', 'Rain ', 'FFMC',
        'DMC', 'DC', 'ISI', 'BUI', 'FWI', 'Classes ', 'Region'],
        dtype='object')
```

```
[75]: # Fix Spaces in Columns
# . strip ( ) function removes the blank space from the columns

df.columns=df.columns.str.strip()
df.columns
```

```
[75]: Index(['day', 'month', 'year', 'Temperature', 'RH', 'Ws', 'Rain', 'FFMC',
        'DMC', 'DC', 'ISI', 'BUI', 'FWI', 'Classes', 'Region'],
        dtype='object')
```

```
[76]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 243 entries, 0 to 242
Data columns (total 15 columns):
#   Column          Non-Null Count  Dtype

```

```

---  -----  -----  -----
0   day          243 non-null  object
1   month        243 non-null  object
2   year         243 non-null  object
3   Temperature  243 non-null  object
4   RH           243 non-null  object
5   Ws           243 non-null  object
6   Rain         243 non-null  object
7   FFMFC        243 non-null  object
8   DMC          243 non-null  object
9   DC           243 non-null  object
10  ISI          243 non-null  object
11  BUI          243 non-null  object
12  FWI          243 non-null  object
13  Classes      243 non-null  object
14  Region       243 non-null  int64

```

dtypes: int64(1), object(14)

memory usage: 28.6+ KB

Changes the required columns as integer data type

```
[77]: df.columns
```

```
[77]: Index(['day', 'month', 'year', 'Temperature', 'RH', 'Ws', 'Rain', 'FFMC',
          'DMC', 'DC', 'ISI', 'BUI', 'FWI', 'Classes', 'Region'],
          dtype='object')
```

```
[78]: # This is how we covert Object Datatype into the Integer Datatype
```

```
df[['month', 'day', 'year', 'Temperature', 'RH', 'Ws']] = df[['month', 'day', 'year', 'Temperature', 'RH',
    ↳ astype(int)
```

```
[79]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 243 entries, 0 to 242
```

```
Data columns (total 15 columns):
```

```

#   Column          Non-Null Count  Dtype
---  -----  -----
0   day          243 non-null  int64
1   month        243 non-null  int64
2   year         243 non-null  int64
3   Temperature  243 non-null  int64
4   RH           243 non-null  int64
5   Ws           243 non-null  int64
6   Rain         243 non-null  object
7   FFMFC        243 non-null  object
8   DMC          243 non-null  object

```



```

9   DC                243 non-null    object
10  ISI                243 non-null    object
11  BUI                243 non-null    object
12  FWI                243 non-null    object
13  Classes            243 non-null    object
14  Region             243 non-null    int64
dtypes: int64(7), object(8)
memory usage: 28.6+ KB

```

```
[80]: df.head()
```

```

[80]:   day  month  year  Temperature  RH  Ws  Rain  FFMC  DMC   DC  ISI  BUI  \
0     1     6  2012           29  57  18     0  65.7  3.4   7.6  1.3  3.4
1     2     6  2012           29  61  13     1.3  64.4  4.1   7.6   1  3.9
2     3     6  2012           26  82  22    13.1  47.1  2.5   7.1  0.3  2.7
3     4     6  2012           25  89  13     2.5  28.6  1.3   6.9   0  1.7
4     5     6  2012           27  77  16     0  64.8   3  14.2  1.2  3.9

      FWI   Classes  Region
0  0.5  not fire      0
1  0.4  not fire      0
2  0.1  not fire      0
3   0   not fire      0
4  0.5  not fire      0

```

### Changing the other columns to float data datatype

```
[81]: objects=[features for features in df.columns if df[features].dtypes=='O']
```

```
[82]: objects
```

```
[82]: ['Rain', 'FFMC', 'DMC', 'DC', 'ISI', 'BUI', 'FWI', 'Classes']
```

```
[ ]:
```

```

[83]: for i in objects:
        if i!='Classes':
            df[i]=df[i].astype(float)

```

```
[84]: df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 243 entries, 0 to 242
Data columns (total 15 columns):
#   Column      Non-Null Count  Dtype
---  -
0   day         243 non-null   int64
1   month       243 non-null   int64

```

```

2   year          243 non-null    int64
3   Temperature   243 non-null    int64
4   RH            243 non-null    int64
5   Ws            243 non-null    int64
6   Rain          243 non-null    float64
7   FPMC          243 non-null    float64
8   DMC           243 non-null    float64
9   DC            243 non-null    float64
10  ISI           243 non-null    float64
11  BUI           243 non-null    float64
12  FWI           243 non-null    float64
13  Classes       243 non-null    object
14  Region        243 non-null    int64
dtypes: float64(7), int64(7), object(1)
memory usage: 28.6+ KB

```

```
[85]: objects
```

```
[85]: ['Rain', 'FFMC', 'DMC', 'DC', 'ISI', 'BUI', 'FWI', 'Classes']
```

```
[86]: df.describe()
```

```
[86]:
```

	day	month	year	Temperature	RH	Ws \
count	243.000000	243.000000	243.0	243.000000	243.000000	243.000000
mean	15.761317	7.502058	2012.0	32.152263	62.041152	15.493827
std	8.842552	1.114793	0.0	3.628039	14.828160	2.811385
min	1.000000	6.000000	2012.0	22.000000	21.000000	6.000000
25%	8.000000	7.000000	2012.0	30.000000	52.500000	14.000000
50%	16.000000	8.000000	2012.0	32.000000	63.000000	15.000000
75%	23.000000	8.000000	2012.0	35.000000	73.500000	17.000000
max	31.000000	9.000000	2012.0	42.000000	90.000000	29.000000

	Rain	FFMC	DMC	DC	ISI	BUI \
count	243.000000	243.000000	243.000000	243.000000	243.000000	243.000000
mean	0.762963	77.842387	14.680658	49.430864	4.742387	16.690535
std	2.003207	14.349641	12.393040	47.665606	4.154234	14.228421
min	0.000000	28.600000	0.700000	6.900000	0.000000	1.100000
25%	0.000000	71.850000	5.800000	12.350000	1.400000	6.000000
50%	0.000000	83.300000	11.300000	33.100000	3.500000	12.400000
75%	0.500000	88.300000	20.800000	69.100000	7.250000	22.650000
max	16.800000	96.000000	65.900000	220.400000	19.000000	68.000000

	FWI	Region
count	243.000000	243.000000
mean	7.035391	0.497942
std	7.440568	0.501028
min	0.000000	0.000000

```

25%      0.700000      0.000000
50%      4.200000      0.000000
75%     11.450000      1.000000
max      31.100000      1.000000

```

```
[87]: df.head()
```

```

[87]:   day  month  year  Temperature  RH  Ws  Rain  FFMC  DMC   DC  ISI  BUI  \
0     1     6  2012           29  57  18   0.0  65.7  3.4   7.6  1.3  3.4
1     2     6  2012           29  61  13   1.3  64.4  4.1   7.6  1.0  3.9
2     3     6  2012           26  82  22  13.1  47.1  2.5   7.1  0.3  2.7
3     4     6  2012           25  89  13   2.5  28.6  1.3   6.9  0.0  1.7
4     5     6  2012           27  77  16   0.0  64.8  3.0  14.2  1.2  3.9

      FWI      Classes  Region
0  0.5  not fire         0
1  0.4  not fire         0
2  0.1  not fire         0
3  0.0  not fire         0
4  0.5  not fire         0

```

```

[88]: # Here we save the updated csv file

df.to_csv('Algerian_forest_fire_dataset_cleaned (2).csv',index=False)

```

### 1.3 Exploratory Data Analysis

```

[89]: # Here we remove or drop day, month and year

df_copy=df.drop(['day','month','year'],axis=1)

```

```
[90]: df_copy.head()
```

```

[90]:   Temperature  RH  Ws  Rain  FFMC  DMC   DC  ISI  BUI  FWI      Classes  \
0           29  57  18   0.0  65.7  3.4   7.6  1.3  3.4  0.5  not fire
1           29  61  13   1.3  64.4  4.1   7.6  1.0  3.9  0.4  not fire
2           26  82  22  13.1  47.1  2.5   7.1  0.3  2.7  0.1  not fire
3           25  89  13   2.5  28.6  1.3   6.9  0.0  1.7  0.0  not fire
4           27  77  16   0.0  64.8  3.0  14.2  1.2  3.9  0.5  not fire

      Region
0         0
1         0
2         0
3         0
4         0

```

```
[91]: # Here we see the classes of categories
```

```
df_copy['Classes'].value_counts()
```

```
[91]: fire          131
      not fire      101
      fire           4
      fire           2
      not fire        2
      not fire        1
      not fire        1
      not fire        1
      Name: Classes, dtype: int64
```

```
[92]: # Encoding of the categories in classes
```

```
# Conversion of fire into 1 and not fire into 0
```

```
# np.where ( ) function is used for conversion and str.contains ( ) function is used for converting fire or not fire into 1 and 0
```

```
df_copy['Classes']=np.where(df_copy['Classes'].str.contains('not fire'),0,1)
```

```
[93]: df_copy.head()
```

```
[93]:   Temperature  RH  Ws  Rain  FFMC  DMC   DC  ISI  BUI  FWI  Classes  Region
0           29  57  18   0.0  65.7  3.4   7.6  1.3  3.4  0.5         0         0
1           29  61  13   1.3  64.4  4.1   7.6  1.0  3.9  0.4         0         0
2           26  82  22  13.1  47.1  2.5   7.1  0.3  2.7  0.1         0         0
3           25  89  13   2.5  28.6  1.3   6.9  0.0  1.7  0.0         0         0
4           27  77  16   0.0  64.8  3.0  14.2  1.2  3.9  0.5         0         0
```

```
[94]: df_copy.tail()
```

```
[94]:   Temperature  RH  Ws  Rain  FFMC  DMC   DC  ISI  BUI  FWI  Classes  \
238           30  65  14   0.0  85.4 16.0  44.5  4.5 16.9  6.5         1
239           28  87  15   4.4  41.1  6.5   8.0  0.1  6.2  0.0         0
240           27  87  29   0.5  45.9  3.5   7.9  0.4  3.4  0.2         0
241           24  54  18   0.1  79.7  4.3  15.2  1.7  5.1  0.7         0
242           24  64  15   0.2  67.3  3.8  16.5  1.2  4.8  0.5         0
```

```
      Region
238         1
239         1
240         1
241         1
242         1
```

```
[95]: df_copy['Classes'].value_counts()
```

```
[95]: 1    137
      0    106
      Name: Classes, dtype: int64
```

VISUALIZATION OF DATASET :-

```
[ ]: ## Plot desnity plot for all features

plt.style.use('seaborn')
df_copy.hist(bins=50,figsize=(20,15))
plt.show()
```

/tmp/ipykernel\_79/2347077609.py:3: MatplotlibDeprecationWarning: The seaborn styles shipped by Matplotlib are deprecated since 3.6, as they no longer correspond to the styles shipped by seaborn. However, they will remain available as 'seaborn-v0\_8-*<style>*'. Alternatively, directly use the seaborn API instead.

```
plt.style.use('seaborn')
```

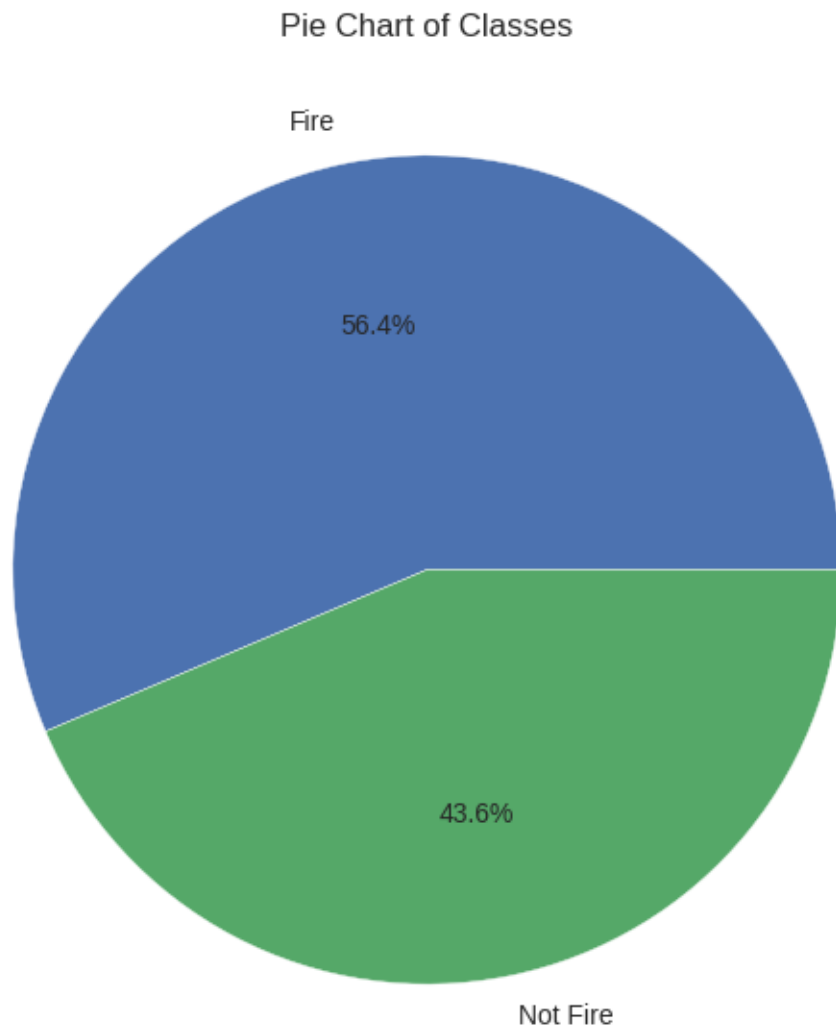


```
[ ]: ## Percentage for Pie Chart

percentage=df_copy['Classes'].value_counts(normalize=True)*100
```

```
[ ]: # plotting piechart

classlabels=["Fire","Not Fire"]
plt.figure(figsize=(12,7))
plt.pie(percentage,labels=classlabels,autopct='%1.1f%%')
plt.title("Pie Chart of Classes")
plt.show()
```



## 1.4 Correlation

```
[ ]: df_copy.corr()
```

```
[ ]:
```

	Temperature	RH	Ws	Rain	FFMC	DMC	\
Temperature	1.000000	-0.651400	-0.284510	-0.326492	0.676568	0.485687	
RH	-0.651400	1.000000	0.244048	0.222356	-0.644873	-0.408519	
Ws	-0.284510	0.244048	1.000000	0.171506	-0.166548	-0.000721	
Rain	-0.326492	0.222356	0.171506	1.000000	-0.543906	-0.288773	
FFMC	0.676568	-0.644873	-0.166548	-0.543906	1.000000	0.603608	
DMC	0.485687	-0.408519	-0.000721	-0.288773	0.603608	1.000000	
DC	0.376284	-0.226941	0.079135	-0.298023	0.507397	0.875925	
ISI	0.603871	-0.686667	0.008532	-0.347484	0.740007	0.680454	
BUI	0.459789	-0.353841	0.031438	-0.299852	0.592011	0.982248	
FWI	0.566670	-0.580957	0.032368	-0.324422	0.691132	0.875864	
Classes	0.516015	-0.432161	-0.069964	-0.379097	0.769492	0.585658	
Region	0.269555	-0.402682	-0.181160	-0.040013	0.222241	0.192089	

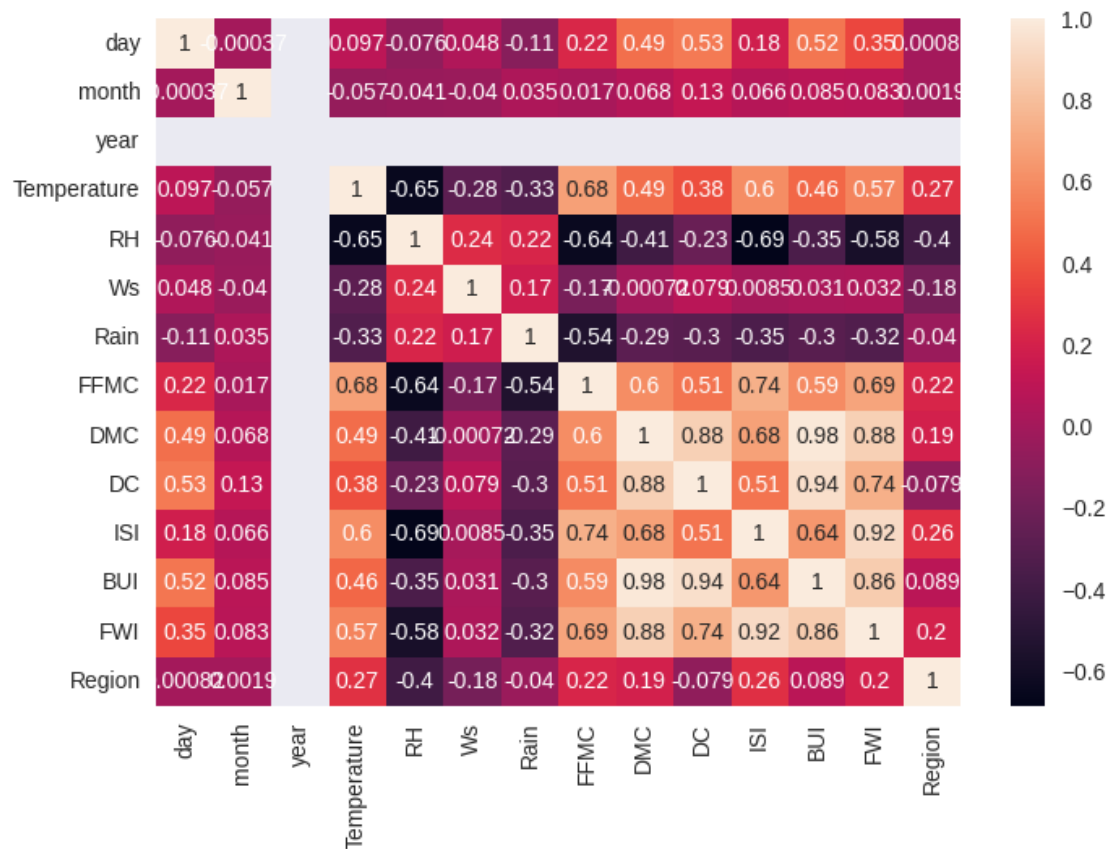
	DC	ISI	BUI	FWI	Classes	Region
Temperature	0.376284	0.603871	0.459789	0.566670	0.516015	0.269555
RH	-0.226941	-0.686667	-0.353841	-0.580957	-0.432161	-0.402682
Ws	0.079135	0.008532	0.031438	0.032368	-0.069964	-0.181160
Rain	-0.298023	-0.347484	-0.299852	-0.324422	-0.379097	-0.040013
FFMC	0.507397	0.740007	0.592011	0.691132	0.769492	0.222241
DMC	0.875925	0.680454	0.982248	0.875864	0.585658	0.192089
DC	1.000000	0.508643	0.941988	0.739521	0.511123	-0.078734
ISI	0.508643	1.000000	0.644093	0.922895	0.735197	0.263197
BUI	0.941988	0.644093	1.000000	0.857973	0.586639	0.089408
FWI	0.739521	0.922895	0.857973	1.000000	0.719216	0.197102
Classes	0.511123	0.735197	0.586639	0.719216	1.000000	0.162347
Region	-0.078734	0.263197	0.089408	0.197102	0.162347	1.000000

```
[ ]: sns.heatmap(df.corr(),annot=True)
```

```
/tmp/ipykernel_79/4277794465.py:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
```

```
sns.heatmap(df.corr(),annot=True)
```

```
[ ]: <AxesSubplot: >
```

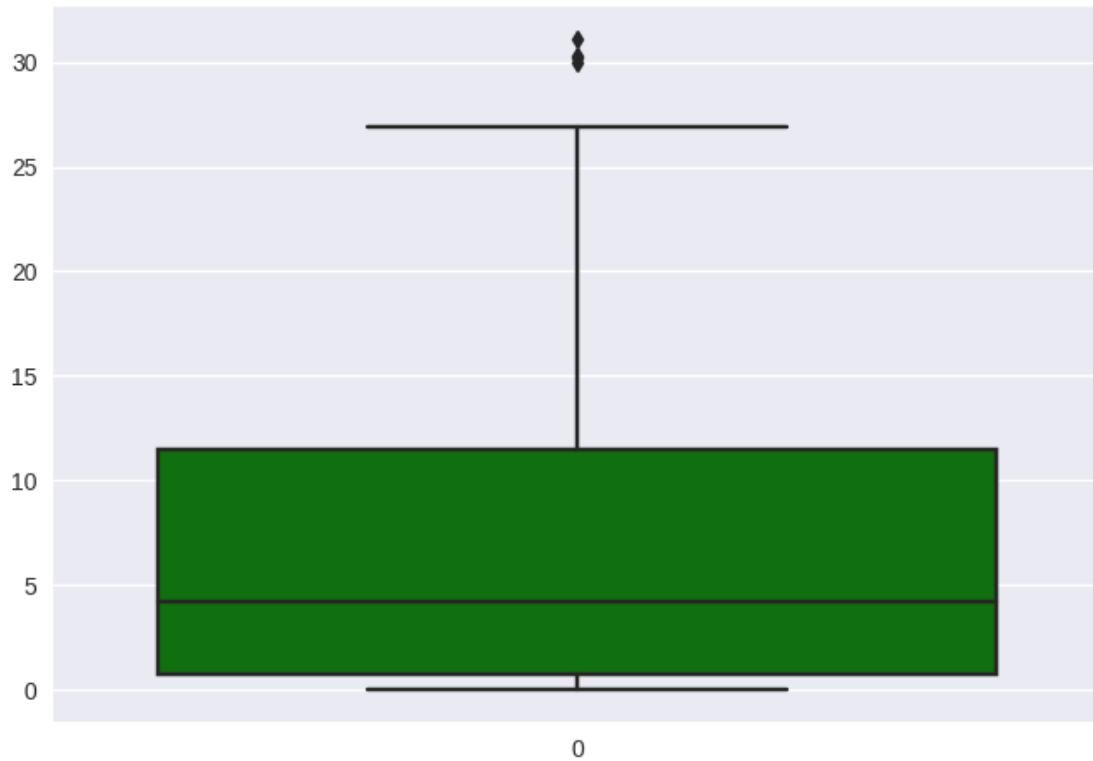


```
[ ]: ## Box Plots

sns.boxplot(df['FWI'],color='green')
```

```
[ ]: <AxesSubplot: >
```





```
[ ]: df.head()
```

```
[ ]:   day  month  year  Temperature  RH  Ws  Rain  FFMC  DMC  DC  ISI  BUI  \
0    1     6  2012         29  57  18   0.0  65.7  3.4  7.6  1.3  3.4
1    2     6  2012         29  61  13   1.3  64.4  4.1  7.6  1.0  3.9
2    3     6  2012         26  82  22  13.1  47.1  2.5  7.1  0.3  2.7
3    4     6  2012         25  89  13   2.5  28.6  1.3  6.9  0.0  1.7
4    5     6  2012         27  77  16   0.0  64.8  3.0 14.2  1.2  3.9
```

```
      FWI      Classes  Region
0  0.5  not fire         0
1  0.4  not fire         0
2  0.1  not fire         0
3  0.0  not fire         0
4  0.5  not fire         0
```

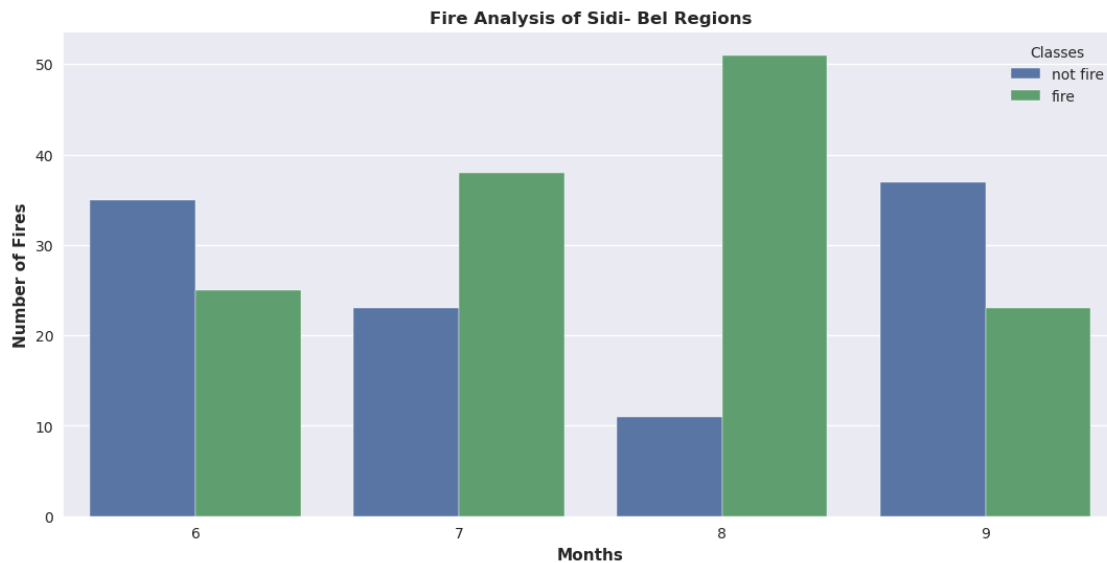
```
[ ]: df['Classes']=np.where(df['Classes'].str.contains('not fire'),'not fire','fire')
```

```
[ ]: ## Monthly Fire Analysis
```

```
dftemp=df.loc[df['Region']==1]
plt.subplots(figsize=(13,6))
```

```
sns.set_style('whitegrid')
sns.countplot(x='month',hue='Classes',data=df)
plt.ylabel('Number of Fires',weight='bold')
plt.xlabel('Months',weight='bold')
plt.title("Fire Analysis of Sidi- Bel Regions",weight='bold')
```

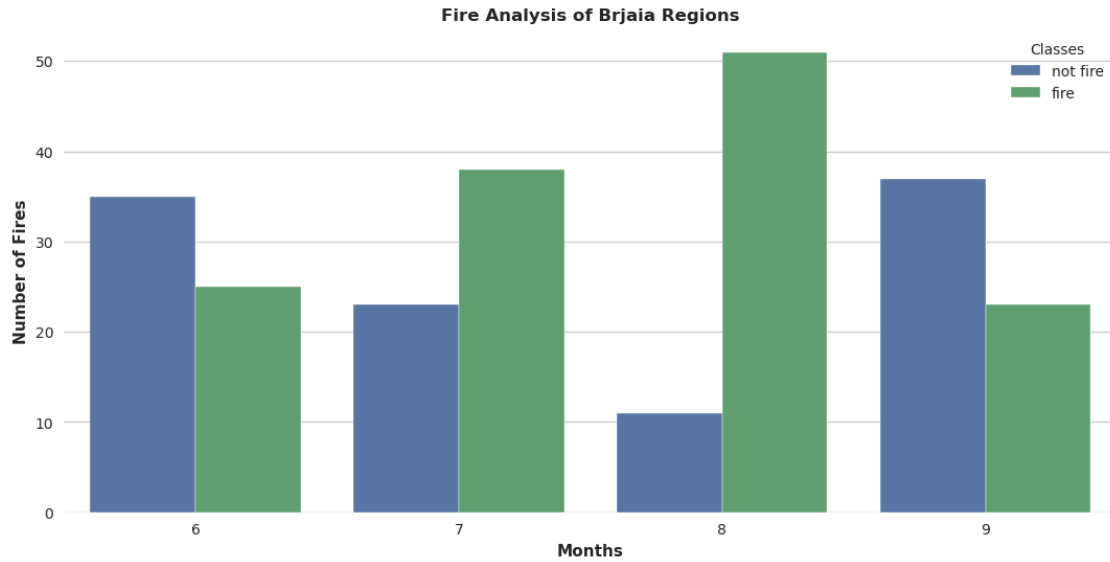
```
[ ]: Text(0.5, 1.0, 'Fire Analysis of Sidi- Bel Regions')
```



```
[ ]: ## Monthly Fire Analysis

dftemp=df.loc[df['Region']==0]
plt.subplots(figsize=(13,6))
sns.set_style('whitegrid')
sns.countplot(x='month',hue='Classes',data=df)
plt.ylabel('Number of Fires',weight='bold')
plt.xlabel('Months',weight='bold')
plt.title("Fire Analysis of Brjaia Regions",weight='bold')
```

```
[ ]: Text(0.5, 1.0, 'Fire Analysis of Brjaia Regions')
```



Its observed that August and September had the most number of forest fires for both regions. And from the above plot of months, we can understand few things

Most of the fires happened in August and very high Fires happened in only 3 months - June, July and August.

Less Fires was on September

THANK YOU SO MUCH !!

YOURS VIRAT TIWARI : )