regression-on-algerian-forest-fire

April 20, 2023

0.0.1 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 attribues and 1 output attribue (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

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

[2]:
```
dataset = pd.read_csv('Algerian_forest_fires_dataset_UPDATE.csv',header=1)
```

[3]:
```
dataset.head()
```

[3]:
```
   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
```

[4]:
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 246 entries, 0 to 245
Data columns (total 14 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
dtypes: object(14)
memory usage: 27.0+ KB
```

0.0.2 Data Cleaning

[5]:
```
dataset[dataset.isnull().any(axis=1)]
```

[5]:
```
                              day month  year Temperature   RH   Ws Rain   \
122  Sidi-Bel Abbes Region Dataset   NaN   NaN         NaN  NaN  NaN   NaN
167                              14    07  2012          37   37   18   0.2

     FFMC   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 converted into two sets bsed on the region from 122 ijndex we can make new colum which hold that two region

1. Bejaia Region Dataset
2. Sidi-Bel Abbes Region Dataset

[6]:
```
dataset.loc[:122,'Region'] = 0
dataset.loc[122:,'Region'] = 1
df = dataset
```

[7]:
```
df.head()
```

[7]:
```
   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
```

[8]:
```
df.tail()
```

[8]:
```
     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        1.0
242    0    not fire        1.0
243  0.2    not fire        1.0
244  0.7    not fire        1.0
245  0.5  not fire          1.0
```

[9]:
```
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
```

[10]:
```
df[["Region"]] = df[["Region"]].astype(int)
```

[11]:
```
df.isnull().sum()
```

[11]:
```
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
```

hear only few data content a null value point so we drop that data into dataset

[12]:
```
df = df.dropna().reset_index(drop = True)
```

[13]:
```
df.isnull().sum()
```

[13]:
```
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
```

[14]:
```
df.iloc[[122]]
```

[14]:
```
     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
```

[15]:
```
### removing 122 index
df=df.drop(122).reset_index(drop=True)
```

[16]:
```
df.columns
```

[16]:
```
Index(['day', 'month', 'year', 'Temperature', ' RH', ' Ws', 'Rain ', 'FFMC',
       'DMC', 'DC', 'ISI', 'BUI', 'FWI', 'Classes  ', 'Region'],
      dtype='object')
```

[17]:
```
### Removing extra space
df.columns = df.columns.str.strip()
df.columns
```

[17]:
```
Index(['day', 'month', 'year', 'Temperature', 'RH', 'Ws', 'Rain', 'FFMC',
       'DMC', 'DC', 'ISI', 'BUI', 'FWI', 'Classes', 'Region'],
      dtype='object')
```

[18]:
```
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   FFMC         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    int32
dtypes: int32(1), object(14)
memory usage: 27.7+ KB
```

[19]:
```
df.head()
```

[19]:

	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

Change the required column as int or float data type

[20]:
```
df[['day','month','year','Temperature','RH','Ws']] =␣
↪df[['day','month','year','Temperature','RH','Ws']].astype(int)
```

[21]:
```
df[['Rain','FFMC','DMC','DC','ISI','BUI','FWI']] =␣
↪df[['Rain','FFMC','DMC','DC','ISI','BUI','FWI']].astype(float)
```

[22]:
```
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    int32  
 1   month        243 non-null    int32  
 2   year         243 non-null    int32  
 3   Temperature  243 non-null    int32  
 4   RH           243 non-null    int32  
 5   Ws           243 non-null    int32  
 6   Rain         243 non-null    float64
 7   FFMC         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    int32  
dtypes: float64(7), int32(7), object(1)
memory usage: 22.0+ KB
```

[23]:
```
df.head()
```

[23]:

	day	month	year	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI	Classes	Region
0	1	6	2012	29	57	18	0.0	65.7	3.4	7.6	1.3	3.4	0.5	not fire	0
1	2	6	2012	29	61	13	1.3	64.4	4.1	7.6	1.0	3.9	0.4	not fire	0
2	3	6	2012	26	82	22	13.1	47.1	2.5	7.1	0.3	2.7	0.1	not fire	0
3	4	6	2012	25	89	13	2.5	28.6	1.3	6.9	0.0	1.7	0.0	not fire	0
4	5	6	2012	27	77	16	0.0	64.8	3.0	14.2	1.2	3.9	0.5	not fire	0

[24]:
```
df.describe()
```

[24]:

	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
```

[25]:
```
df.to_csv('Algerian_forest_fires_cleand__dataset_UPDATE.csv')
```

0.1 Exploratory Data Analysis (EDA)

[26]:
```
df_clean = pd.read_csv('Algerian_forest_fires_cleand__dataset_UPDATE.csv')
df_copy = df_clean.drop(['day','month','year'],axis=1)
```

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

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

[28]:
```
df_copy.drop(['Unnamed: 0'],axis=1,inplace=True)
```

[29]:
```
df_copy.head()
```

[29]:
```
   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
```

[30]:
```
## categores in class
df[['Classes']].value_counts()
```

[30]:
```
Classes
fire             131
not fire         101
fire               4
fire               2
not fire           2
not fire           1
not fire           1
not fire           1
dtype: int64
```

[31]:
```
df_copy['Classes'] = np.where(df_copy['Classes'].str.contains('not fire'),0,1)
```

[32]:
```
df_copy[['Classes']].value_counts()
```

[32]:
```
Classes
1          137
0          106
dtype: int64
```

[33]:
```
plt.style.use('seaborn')
df_copy.hist(bins=50,figsize=(20,15))
plt.plot()
```

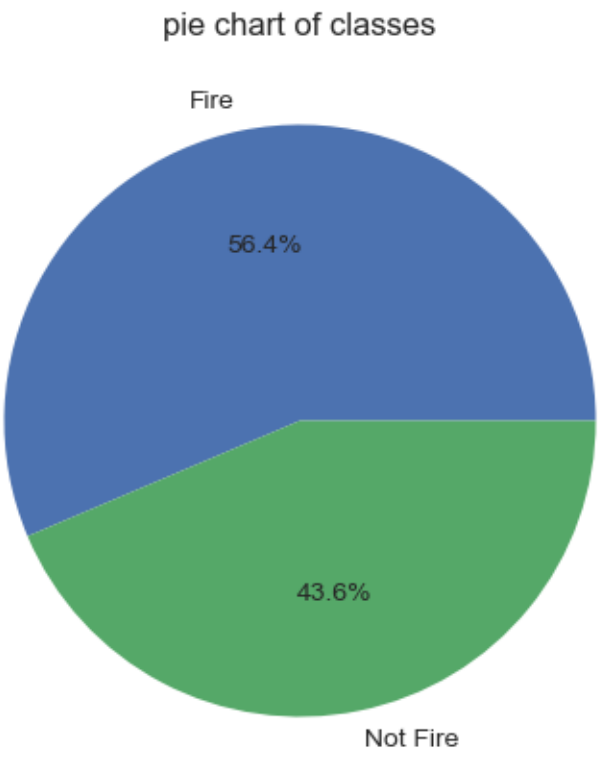
```
C:\Users\barav\AppData\Local\Temp\ipykernel_22096\3859349735.py:1:
MatplotlibDeprecationWarning: The seaborn styles shipped by Matplotlib are
```

```
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')
```

[33]: []

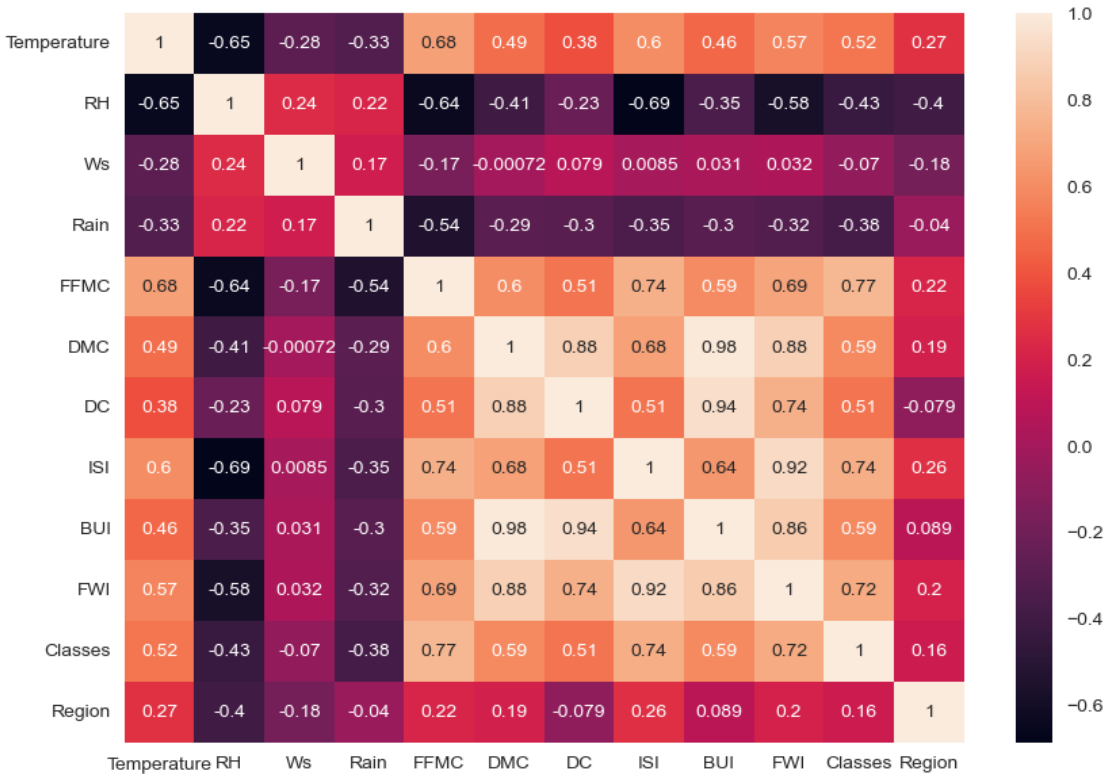
[34]:
```
### percentage for pie chart
percentage = df_copy['Classes'].value_counts(normalize=True)*100
```

[35]:
```
classlabels=["Fire","Not Fire"]
plt.figure(figsize=(5,5))
plt.pie(percentage,labels=classlabels,autopct='%1.1f%%')
plt.title('pie chart of classes')
plt.show()
```



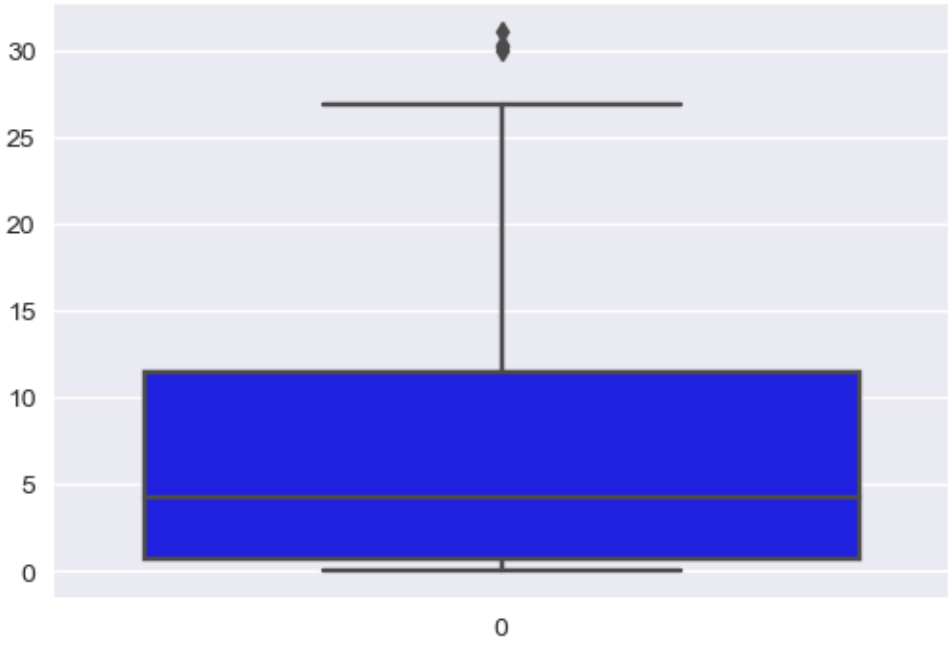
0.1.1 correlation

[36]:
```
plt.figure(figsize=(10,7))
sns.heatmap(df_copy.corr(),annot=True)
plt.show()
```



[37]:
```
plt.figure(figsize=(6,4))
sns.boxplot(df_copy['FWI'],color='blue')
```

[37]: <Axes: >



[38]:
```
df_copy.head()
```

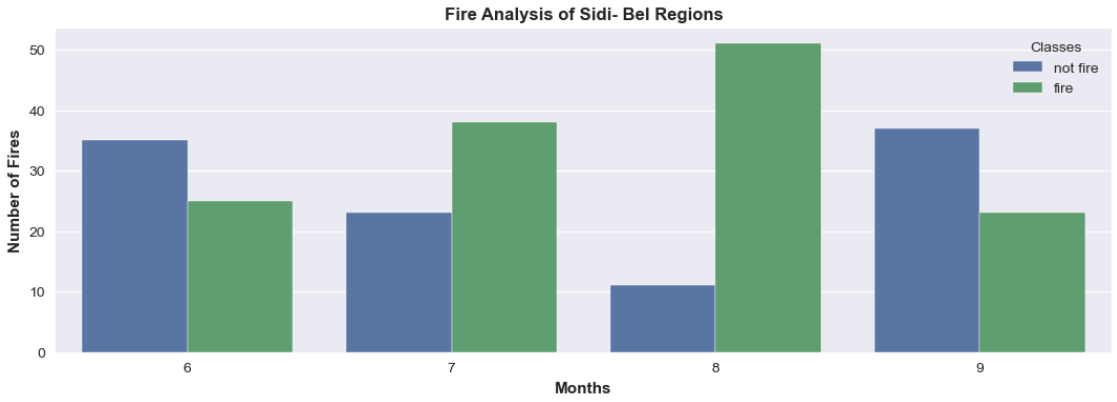
[38]:

	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

[39]:
```
df['Classes']=np.where(df['Classes'].str.contains('not fire'),'not fire','fire')
```

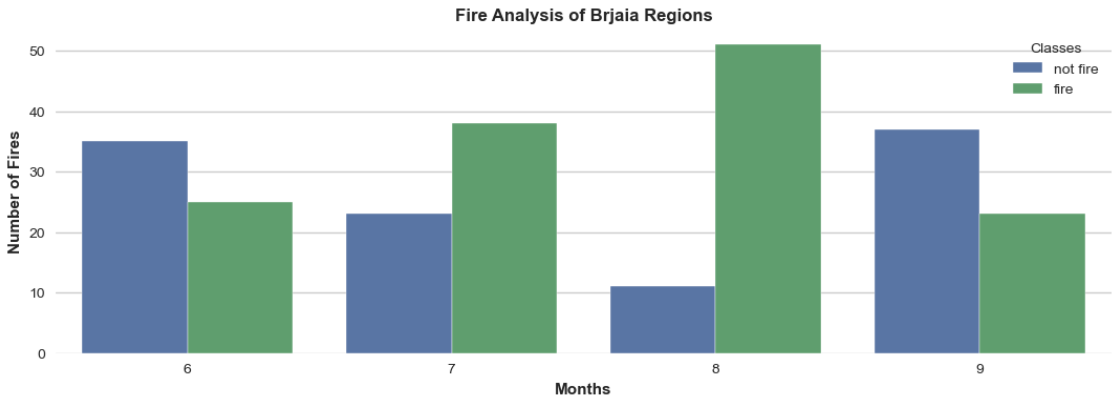
[40]:
```
## Monthly Fire Analysis
dftemp=df.loc[df['Region']==1]
plt.subplots(figsize=(13,4))
sns.set_style('whitegrid')
sns.countplot(x='month',hue='Classes',data=df)
plt.ylabel('Number of Fires',weight='bold')
plt.xlabel('Months',weight='bold')
plt.title("Fire Analysis of Sidi- Bel Regions",weight='bold')
```

[40]:
```
Text(0.5, 1.0, 'Fire Analysis of Sidi- Bel Regions')
```



[41]:
```
## Monthly Fire Analysis
dftemp=df.loc[df['Region']==0]
plt.subplots(figsize=(13,4))
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')
```

[41]: 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

0.2 Model Training

[42]:
```
df_copy.head()
```

[42]:

	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

[43]:
```
X = df_copy.drop(['FWI'],axis=1)
y = df_copy['FWI']
```

[44]:
```
X.head() ,y.head()
```

[44]:
```
(   Temperature  RH  Ws  Rain  FFMC  DMC   DC  ISI  BUI  Classes  Region
 0           29  57  18   0.0  65.7  3.4  7.6  1.3  3.4        0       0
 1           29  61  13   1.3  64.4  4.1  7.6  1.0  3.9        0       0
 2           26  82  22  13.1  47.1  2.5  7.1  0.3  2.7        0       0
 3           25  89  13   2.5  28.6  1.3  6.9  0.0  1.7        0       0
 4           27  77  16   0.0  64.8  3.0 14.2  1.2  3.9        0       0,
 0    0.5
 1    0.4
 2    0.1
 3    0.0
 4    0.5
 Name: FWI, dtype: float64)
```

0.2.1 Train Test Split

[45]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split =␣
 ↪train_test_split(X,y,test_size=.25,random_state=42)
```

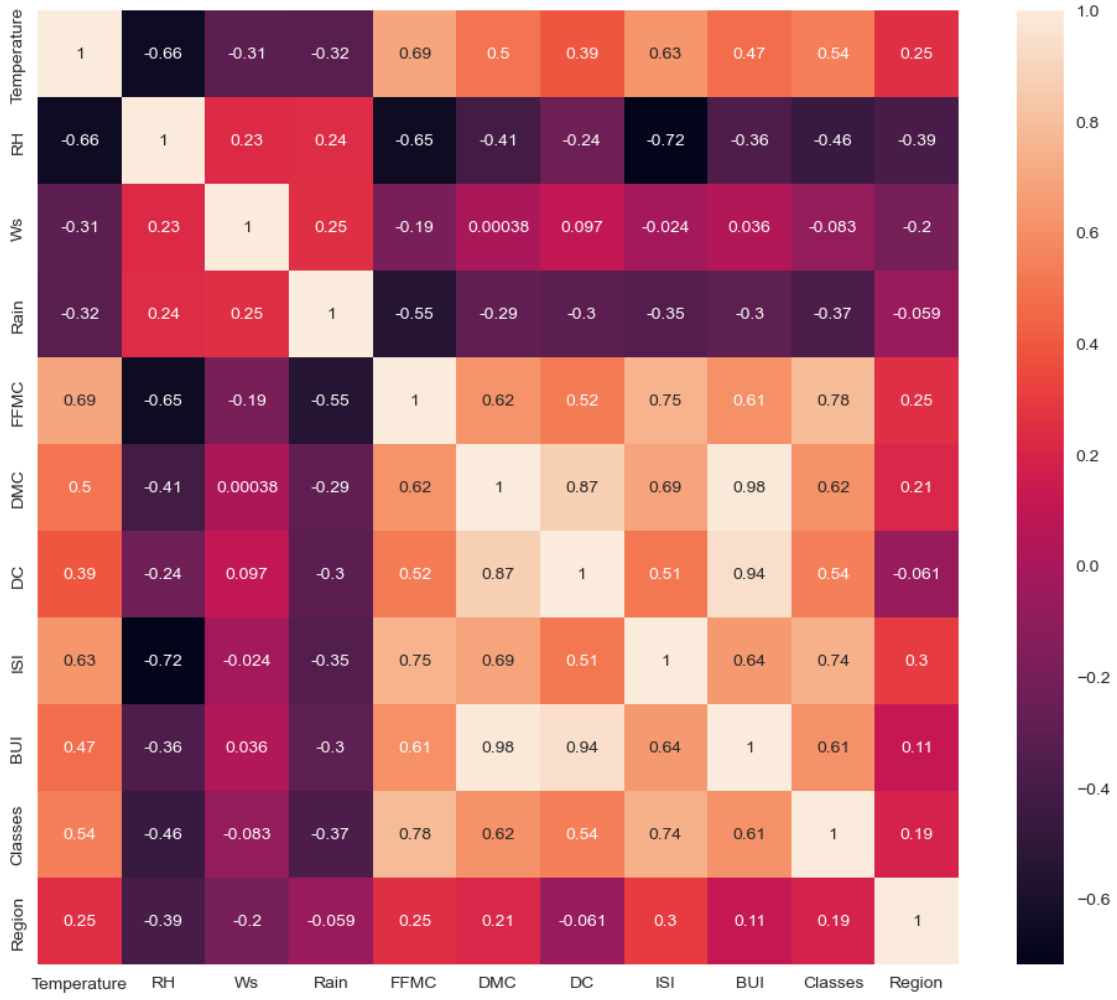
[46]:
```
X_train.shape, X_test.shape
```

[46]:
```
((182, 11), (61, 11))
```

0.2.2 Feature Selection

[47]:
```
plt.figure(figsize=(12,10))
corr = X_train.corr()
sns.heatmap(corr,annot=True)
```

[47]:
```
<Axes: >
```



[48]:
```python
def corrleation(dataset, threshold):
    col_corr = set()
    corr_matrix = dataset.corr()
    for i in range(len(corr_matrix.columns)):
        for j in range(i):
            if abs(corr_matrix.iloc[i, j ]) >threshold:
                colname = corr_matrix.columns[i]
                col_corr.add(colname)
    return col_corr
```

[49]:
```python
## threshold -- Domain experise
corr_feature = corrleation(X_train,0.85)
```

[50]:
```python
corr_feature
```

[50]: {'BUI', 'DC'}

[51]:
```
## drop feature when correlation is more then 0.85
X_train.drop(corr_feature,axis=1,inplace=True)
X_test.drop(corr_feature,axis=1,inplace=True)
X_train.shape, X_test.shape
```

[51]:
```
((182, 9), (61, 9))
```

0.2.3 Feature Scaling Or Standardization

[52]:
```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

[53]:
```
X_train_scaled
```

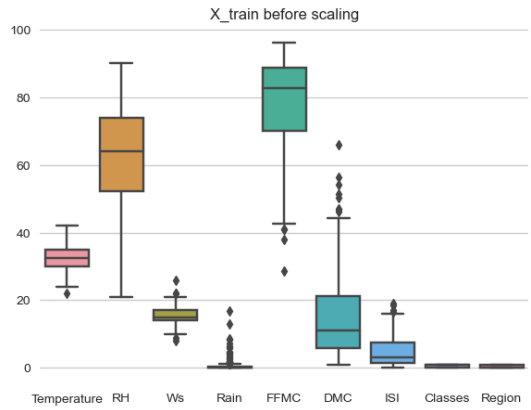
[53]:
```
array([[-0.84284248,  0.78307967,  1.29972026, …, -0.62963326,
        -1.10431526, -0.98907071],
       [-0.30175842,  0.64950844, -0.59874754, …, -0.93058524,
        -1.10431526,  1.01105006],
       [ 2.13311985, -2.08870172, -0.21905398, …,  2.7271388 ,
         0.90553851,  1.01105006],
       …,
       [-1.9250106 ,  0.9166509 ,  0.54033314, …, -1.06948615,
        -1.10431526, -0.98907071],
       [ 0.50986767, -0.21870454,  0.16063958, …,  0.5973248 ,
         0.90553851,  1.01105006],
       [-0.57230045,  0.98343651,  2.05910739, …, -0.86113478,
        -1.10431526, -0.98907071]])
```

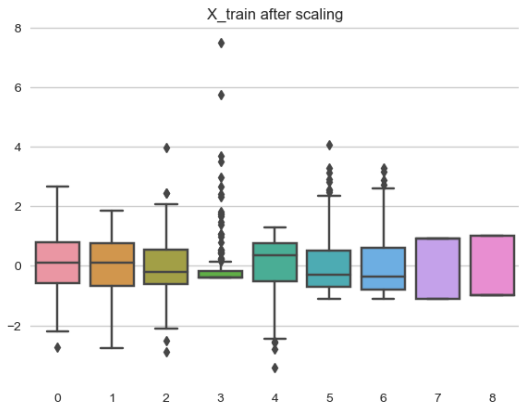
0.2.4 Box Plots To understand Effect Of Standard Scaler

[54]:
```
plt.subplots(figsize = (15,5))
plt.subplot(1,2,1)
sns.boxplot(data=X_train)
plt.title('X_train before scaling')
plt.subplot(1,2,2)
sns.boxplot(data=X_train_scaled)
plt.title('X_train after scaling')
```

```
C:\Users\barav\AppData\Local\Temp\ipykernel_22096\3323087420.py:2:
MatplotlibDeprecationWarning: Auto-removal of overlapping axes is deprecated
since 3.6 and will be removed two minor releases later; explicitly call
ax.remove() as needed.
  plt.subplot(1,2,1)
```

[54]:
```
Text(0.5, 1.0, 'X_train after scaling')
```





0.2.5 Linear Regression Model

[55]:
```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error,r2_score
```

[56]:
```
linreg = LinearRegression()
linreg.fit(X_train_scaled,y_train)
```

[56]:
```
LinearRegression()
```

[57]:
```
y_pred = linreg.predict(X_test_scaled)
mae = mean_absolute_error(y_test,y_pred)
score = r2_score(y_test,y_pred)
print(f"Mean absolute error: {mae}\nR2 score: {score}")
```

```
Mean absolute error: 0.5468236465249993
R2 score: 0.9847657384266951
```

0.2.6 Lasso Regression

[58]:
```
from sklearn.linear_model import Lasso
from sklearn.metrics import mean_absolute_error,r2_score
```

[59]:
```
lasso = Lasso()
lasso.fit(X_train_scaled,y_train)
```

[59]:
```
Lasso()
```

[60]:
```
y_pred = lasso.predict(X_test_scaled)
mae = mean_absolute_error(y_test,y_pred)
score = r2_score(y_test,y_pred)
print(f"Mean absolute error: {mae}\nR2 score: {score}")
```

```
Mean absolute error: 1.1331759949144085
R2 score: 0.9492020263112388
```

0.2.7 Ridge Regression model

[61]:
```
from sklearn.linear_model import Ridge
from sklearn.metrics import mean_absolute_error,r2_score
```

[62]:
```
ridge = Ridge()
ridge.fit(X_train_scaled,y_train)
```

[62]:
```
Ridge()
```

[63]:
```
y_pred = ridge.predict(X_test_scaled)
mae = mean_absolute_error(y_test,y_pred)
score = r2_score(y_test,y_pred)
print(f"Mean absolute error: {mae}\nR2 score: {score}")
```

```
Mean absolute error: 0.5642305340105719
R2 score: 0.9842993364555513
```

0.2.8 Elasticnet Regression

[64]:
```
from sklearn.linear_model import ElasticNet
from sklearn.metrics import mean_absolute_error,r2_score
```

[65]:
```
elastic = ElasticNet()
elastic.fit(X_train_scaled,y_train)
```

[65]:
```
ElasticNet()
```

[66]:
```
y_pred = ridge.predict(X_test_scaled)
mae = mean_absolute_error(y_test,y_pred)
score = r2_score(y_test,y_pred)
print(f"Mean absolute error: {mae}\nR2 score: {score}")
```

```
Mean absolute error: 0.5642305340105719
R2 score: 0.9842993364555513
```

heare i would like to go with Ridge Regression because there r2_score is max. now create a pickle file of Ridge Regression model

0.2.9 Pickle

[67]:
```
import pickle
pickle.dump(scaler,open('scaler.pkl','wb'))
pickle.dump(ridge,open('ridge.pkl','wb'))
```