In [55]:
```
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
```

In [56]:
```
df = pd.read_csv('Algerian_forest_fires_cleaned.csv')
```

In [57]:
```
df.isnull().sum()
```

Out[57]:
```
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
```

In [58]:
```
df.head()
```

Out[58]:

	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

In [59]:
```
df['Classes'] = np.where(df['Classes'].str.contains('not fire'),0,1)
```

In [60]:
```
df.head()
```

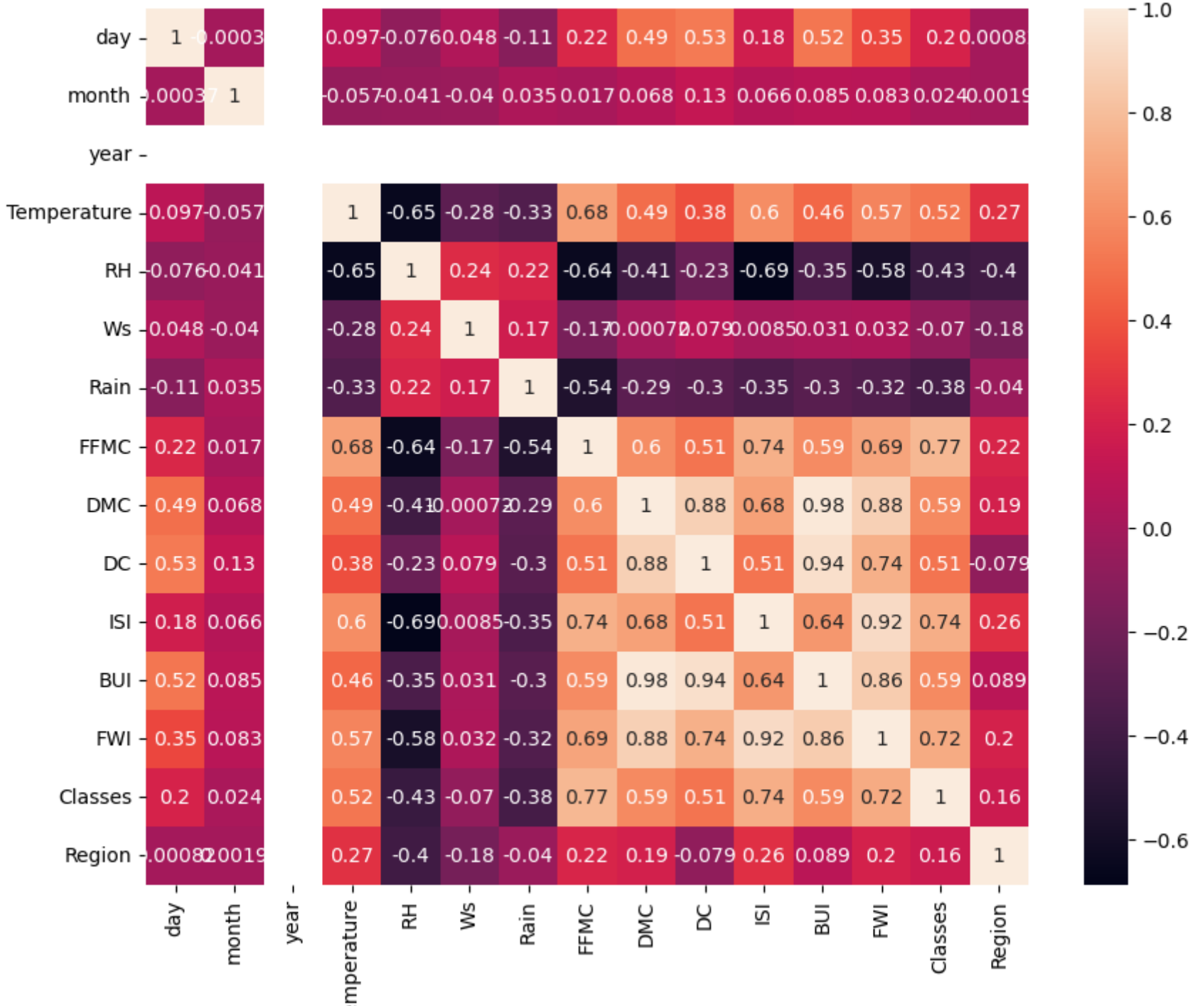
Out[60]:

	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	0	0
1	2	6	2012	29	61	13	1.3	64.4	4.1	7.6	1.0	3.9	0.4	0	0
2	3	6	2012	26	82	22	13.1	47.1	2.5	7.1	0.3	2.7	0.1	0	0
3	4	6	2012	25	89	13	2.5	28.6	1.3	6.9	0.0	1.7	0.0	0	0
4	5	6	2012	27	77	16	0.0	64.8	3.0	14.2	1.2	3.9	0.5	0	0

In [61]:
```
df.corr()
```

	year	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI	Classes	Region
year	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
Temperature	0.097227	-0.056781	NaN	1.000000	-0.651400	-0.284510	-0.326492	0.676568	0.485687	0.376284	0.603871	0.459789	0.566670	0.5160
RH	-0.076034	-0.041252	NaN	-0.651400	1.000000	0.244048	0.222356	-0.644873	-0.408519	-0.226941	-0.686667	-0.353841	-0.580957	-0.4321
Ws	0.047812	-0.039880	NaN	-0.284510	0.244048	1.000000	0.171506	-0.166548	-0.000721	0.079135	0.008532	0.031438	0.032368	-0.0699
Rain	-0.112523	0.034822	NaN	-0.326492	0.222356	0.171506	1.000000	-0.543906	-0.288773	-0.298023	-0.347484	-0.299852	-0.324422	-0.3790
FFMC	0.224956	0.017030	NaN	0.676568	-0.644873	-0.166548	-0.543906	1.000000	0.603608	0.507397	0.740007	0.592011	0.691132	0.7694
DMC	0.491514	0.067943	NaN	0.485687	-0.408519	-0.000721	-0.288773	0.603608	1.000000	0.875925	0.680454	0.982248	0.875864	0.5856
DC	0.527952	0.126511	NaN	0.376284	-0.226941	0.079135	-0.298023	0.507397	0.875925	1.000000	0.508643	0.941988	0.739521	0.5111
ISI	0.180543	0.065608	NaN	0.603871	-0.686667	0.008532	-0.347484	0.740007	0.680454	0.508643	1.000000	0.644093	0.922895	0.7351
BUI	0.517117	0.085073	NaN	0.459789	-0.353841	0.031438	-0.299852	0.592011	0.982248	0.941988	0.644093	1.000000	0.857973	0.5866
FWI	0.350781	0.082639	NaN	0.566670	-0.580957	0.032368	-0.324422	0.691132	0.875864	0.739521	0.922895	0.857973	1.000000	0.7192
Classes	0.202840	0.024004	NaN	0.516015	-0.432161	-0.069964	-0.379097	0.769492	0.585658	0.511123	0.735197	0.586639	0.719216	1.0000
Region	0.000821	0.001857	NaN	0.269555	-0.402682	-0.181160	-0.040013	0.222241	0.192089	-0.078734	0.263197	0.089408	0.197102	0.1623

In [62]:
```
plt.figure(figsize=(10, 8))
sns.heatmap(df.corr(), annot=True)
plt.show()
```


day
month
year
Temperature
RH
Ws
Rain
FFMC
DMC
DC
ISI
BUI
FWI
Classes
Region
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

In [64]:
```
df.columns
```

Out[64]:
```
Index(['day', 'month', 'year', 'Temperature', 'RH', 'Ws', 'Rain', 'FFMC',
       'DMC', 'DC', 'ISI', 'BUI', 'FWI', 'Classes', 'Region'],
      dtype='object')
```

In [70]:
```
x=df.drop(['day','month','year','RH','Ws','Rain','Region','FWI'],axis=1)
x
```

Out[70]:

	Temperature	FFMC	DMC	DC	ISI	BUI	Classes
0	29	65.7	3.4	7.6	1.3	3.4	0
1	29	64.4	4.1	7.6	1.0	3.9	0
2	26	47.1	2.5	7.1	0.3	2.7	0
3	25	28.6	1.3	6.9	0.0	1.7	0
4	27	64.8	3.0	14.2	1.2	3.9	0
...	...	...	...	...	...	...	...
238	30	85.4	16.0	44.5	4.5	16.9	1
239	28	41.1	6.5	8.0	0.1	6.2	0
240	27	45.9	3.5	7.9	0.4	3.4	0
241	24	79.7	4.3	15.2	1.7	5.1	0
242	24	67.3	3.8	16.5	1.2	4.8	0

243 rows × 7 columns

In [71]:
```
y=df['FWI']
y
```

Out[71]:
```
0      0.5
1      0.4
2      0.1
3      0.0
4      0.5
      ...
238    6.5
239    0.0
240    0.2
241    0.7
242    0.5
Name: FWI, Length: 243, dtype: float64
```

In [72]:
```
from sklearn.model_selection import train_test_split
```

In [76]:
```
X_train,X_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=42)
```

In [77]:
```
from sklearn.preprocessing import StandardScaler
scaler= StandardScaler()
x_train_sc= scaler.fit_transform(X_train)
x_test_sc= scaler.transform(X_test)
```

In [78]:
```
from sklearn.linear_model import LinearRegression
Re = LinearRegression()
Re.fit(x_train_sc, y_train)
```

Out[78]:
```
LinearRegression
LinearRegression()
```

In [79]:
```
Re.intercept_,Re.coef_
```

Out[79]:
```
(7.420103092783506,
 array([ 0.0115502 , -0.82144452,  0.38710554, -0.32954227,  5.16228839,
         3.67198149,  0.22867322]))
```

In [82]:
```
y_pred= Re.predict(x_test_sc)
```

In [83]:
```
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
```

In [84]:
```
print("mse: ", mean_squared_error(y_test,y_pred))
print("mae: ", mean_absolute_error(y_test,y_pred))
print("r2: ", r2_score(y_test, y_pred))
```

```
mse:  0.24520857432707494
mae:  0.3629907882995959
r2:  0.9919095564115894
```

In [86]:
```
score= r2_score(y_test, y_pred)
```

In [88]:
```
r2_adjusted= 1-((1-score)*len(y)-1)/(len(y)-x.shape[1]-1)
```

In [129]:
```
r2_adjusted
```

Out[129]:
```
0.9958894562043243
```

In [130]:
```
import pickle
```

In [131]:
```
pickle.dump(scaler, open("scaler_fire.pkl",'wb'))
pickle.dump(Re, open("regressor_fire.pkl",'wb'))


model_regressor = pickle.load(open("regressor_fire.pkl", 'rb'))
model_scaler = pickle.load(open("scaler_fire.pkl", 'rb'))
```

In [132]:
```
X_new =pd.DataFrame({
    'Temperature': [27, 33, 20, 25, 28],
    'FFMC': [65, 84, 42, 39, 50],
    'DMC': [3.4, 4.1, 2.5, 1.3, 3.5],
    'DC': [7.6, 14.5, 6.8, 13.5, 9.5],
    'ISI': [1.3, 1, 0.3, 2.5, 1.6],
    'BUI': [3.4, 3.8, 3.6, 2.9, 3.0],
    'Classes': [1, 0, 1, 0, 1]
})
```

In [133]:
```
X_new_sc=model_scaler.transform(X_new)
```

In [134]:
```
y_new_pre= model_regressor.predict(X_new_sc)
```

In [135]:
```
y_new_pre
```

Out[135]:
```
array([ 0.54965451, -1.28655849,  0.70182137,  2.80819315,  1.67875394])
```

In [ ]: