```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib as plt
from sklearn.preprocessing import StandardScaler
from sklearn import svm
from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.model_selection import train_test_split, cross_val_score


df=pd.read_csv('forestfires.csv')
df
```

	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	...	monthfeb	monthjar
0	mar	fri	86.2	26.2	94.3	5.1	8.2	51	6.7	0.0	...	0	0
1	oct	tue	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	...	0	0
2	oct	sat	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	...	0	0
3	mar	fri	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2	...	0	0
4	mar	sun	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0	...	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	..
512	aug	sun	81.6	56.7	665.6	1.9	27.8	32	2.7	0.0	...	0	0
513	aug	sun	81.6	56.7	665.6	1.9	21.9	71	5.8	0.0	...	0	0
514	aug	sun	81.6	56.7	665.6	1.9	21.2	70	6.7	0.0	...	0	0
515	aug	sat	94.4	146.0	614.7	11.3	25.6	42	4.0	0.0	...	0	0
516	nov	tue	79.5	3.0	106.7	1.1	11.8	31	4.5	0.0	...	0	0

517 rows × 31 columns

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 517 entries, 0 to 516
Data columns (total 31 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   month          517 non-null    object
 1   day            517 non-null    object
 2   FFMC           517 non-null    float64
 3   DMC            517 non-null    float64
 4   DC             517 non-null    float64
 5   ISI            517 non-null    float64
 6   temp           517 non-null    float64
 7   RH             517 non-null    int64
```

```
 8   wind           517 non-null    float64
 9   rain           517 non-null    float64
 10  area           517 non-null    float64
 11  dayfri         517 non-null    int64
 12  daymon         517 non-null    int64
 13  daysat         517 non-null    int64
 14  daysun         517 non-null    int64
 15  daythu         517 non-null    int64
 16  daytue         517 non-null    int64
 17  daywed         517 non-null    int64
 18  monthapr       517 non-null    int64
 19  monthaug       517 non-null    int64
 20  monthdec       517 non-null    int64
 21  monthfeb       517 non-null    int64
 22  monthjan       517 non-null    int64
 23  monthjul       517 non-null    int64
 24  monthjun       517 non-null    int64
 25  monthmar       517 non-null    int64
 26  monthmay       517 non-null    int64
 27  monthnov       517 non-null    int64
 28  monthoct       517 non-null    int64
 29  monthsep       517 non-null    int64
 30  size_category  517 non-null    object
dtypes: float64(8), int64(20), object(3)
memory usage: 125.3+ KB
```

```
df.shape
```

```
(517, 31)
```

```
df.describe()
```

	FFMC	DMC	DC	ISI	temp	RH	win
count	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.00000
mean	90.644681	110.872340	547.940039	9.021663	18.889168	44.288201	4.01760
std	5.520111	64.046482	248.066192	4.559477	5.806625	16.317469	1.79165
min	18.700000	1.100000	7.900000	0.000000	2.200000	15.000000	0.40000
25%	90.200000	68.600000	437.700000	6.500000	15.500000	33.000000	2.70000
50%	91.600000	108.300000	664.200000	8.400000	19.300000	42.000000	4.00000
75%	92.900000	142.400000	713.900000	10.800000	22.800000	53.000000	4.90000
max	96.200000	291.300000	860.600000	56.100000	33.300000	100.000000	9.40000

8 rows × 28 columns

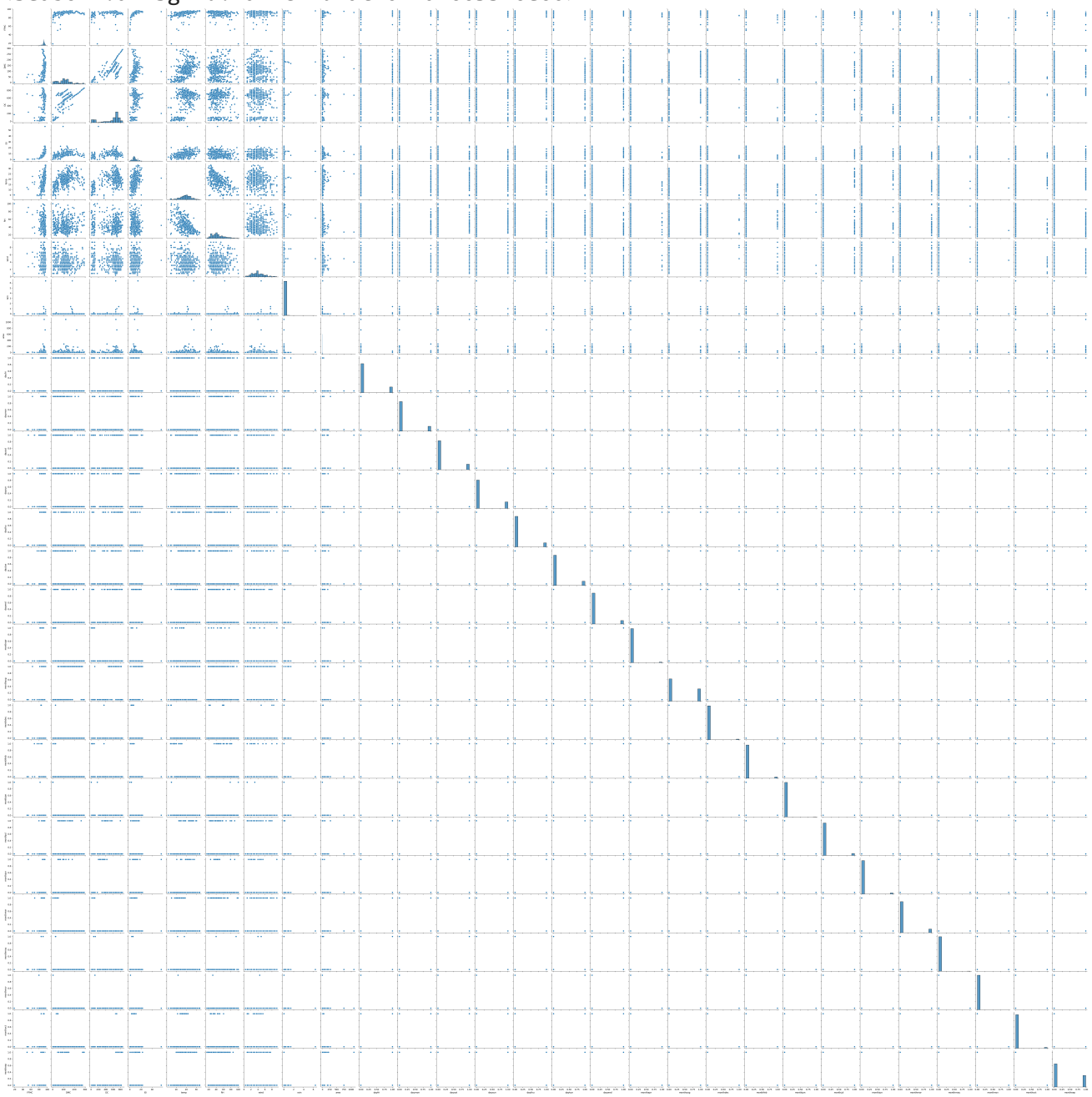
```
df.duplicated()
```

```
0      False
1      False
2      False
```

```
3      False
4      False
       ...
512    False
513    False
514    False
515    False
516    False
Length: 517, dtype: bool
```

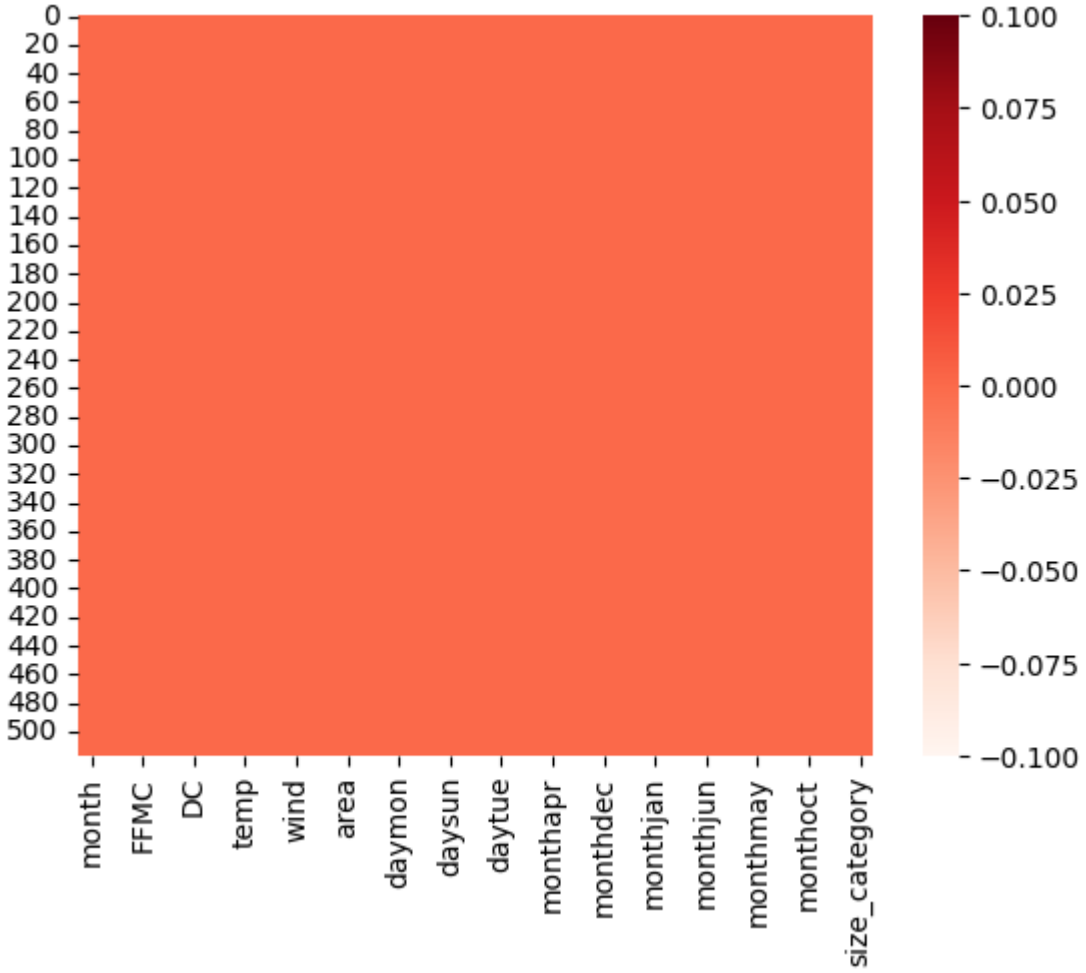
```
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x7a76653f6e60>
```



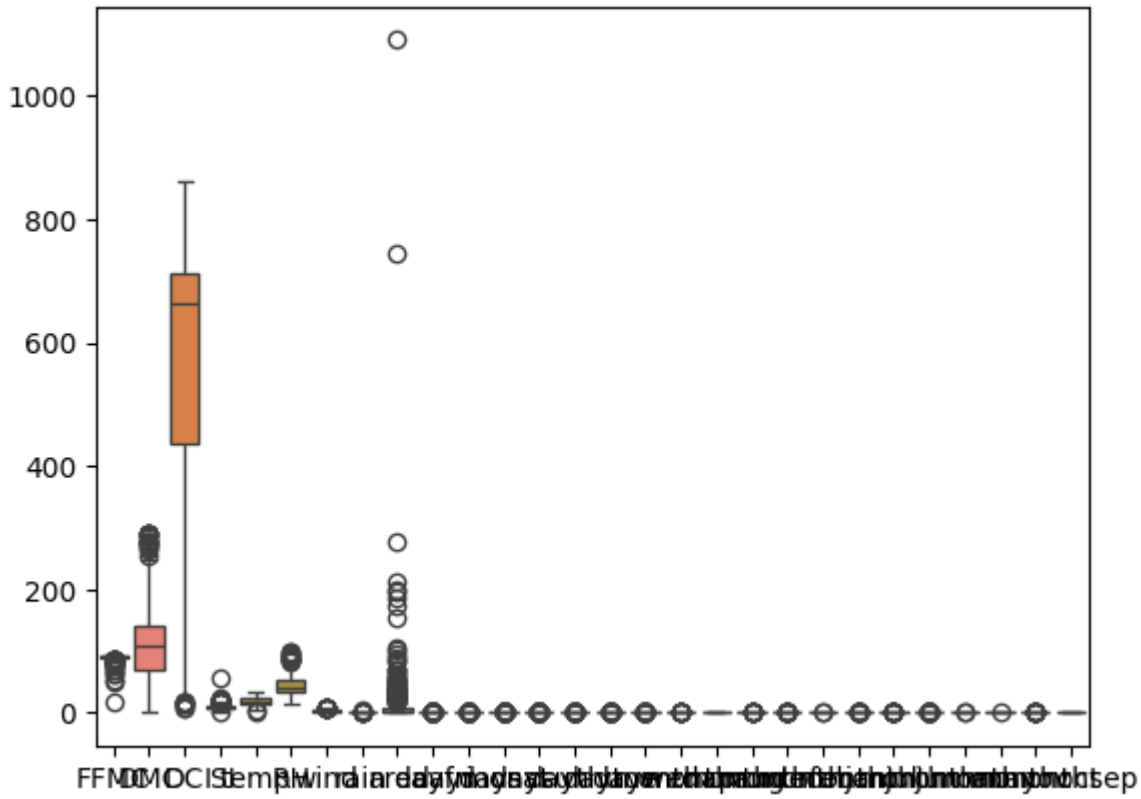
```
sns.heatmap(df.isnull(),cmap='Reds')
```

```
<Axes: >
```




```
sns.boxplot(data=df)
```

```
<Axes: >
```



```
df1=df.iloc[:,2:]
df1
```

	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area	dayfri	...	monthfeb	month
0	86.2	26.2	94.3	5.1	8.2	51	6.7	0.0	0.00	1	...	0	
1	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	0.00	0	...	0	
2	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	0.00	0	...	0	
3	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2	0.00	1	...	0	
4	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0	0.00	0	...	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
512	81.6	56.7	665.6	1.9	27.8	32	2.7	0.0	6.44	0	...	0	
513	81.6	56.7	665.6	1.9	21.9	71	5.8	0.0	54.29	0	...	0	
514	81.6	56.7	665.6	1.9	21.2	70	6.7	0.0	11.16	0	...	0	
515	94.4	146.0	614.7	11.3	25.6	42	4.0	0.0	0.00	0	...	0	
516	79.5	3.0	106.7	1.1	11.8	31	4.5	0.0	0.00	0	...	0	

517 rows × 29 columns

```
array = df1.values
X = array[:,0:28]
Y = array[:,28]
```

```
X
```

```
array([[86.2, 26.2, 94.3, ..., 0, 0, 0],
       [90.6, 35.4, 669.1, ..., 0, 1, 0],
       [90.6, 43.7, 686.9, ..., 0, 1, 0],
       ...,
       [81.6, 56.7, 665.6, ..., 0, 0, 0],
       [94.4, 146.0, 614.7, ..., 0, 0, 0],
       [79.5, 3.0, 106.7, ..., 1, 0, 0]], dtype=object)
```

```
Y
```

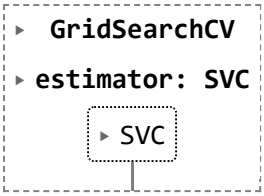
```
       'large', 'large', 'large', 'large', 'large', 'large', 'large',
       'large', 'large', 'large', 'large', 'large', 'large', 'large',
       'large', 'large', 'large', 'large', 'large', 'large', 'large',
       'large', 'large', 'large', 'large', 'large', 'large', 'large',
       'large', 'large', 'large', 'large', 'large', 'large', 'large',
       'large', 'large', 'large', 'large', 'large', 'large', 'large',
       'large', 'small', 'small', 'small', 'large', 'small', 'small',
       'small', 'small', 'small', 'small', 'small', 'small', 'small',
       'small', 'small', 'large', 'small', 'large', 'small', 'small',
       'small', 'large', 'small', 'small', 'small', 'large', 'small',
       'small', 'small', 'small', 'small', 'small', 'large', 'small',
       'large', 'large', 'small', 'large', 'large', 'large', 'large',
       'large', 'large', 'small', 'small', 'large', 'small', 'small',
       'small', 'small', 'small', 'small', 'large', 'small', 'large',
       'large', 'small', 'small', 'small', 'small', 'small', 'small',
       'small', 'small', 'small', 'small', 'small', 'small', 'small',
       'small', 'small', 'small', 'large', 'small', 'small', 'small',
       'small', 'small', 'small', 'small', 'small', 'large', 'large',
       'small', 'small', 'small', 'small', 'small', 'small', 'small',
       'small', 'large', 'large', 'large', 'small', 'small', 'small',
       'small', 'large', 'large', 'small', 'small', 'small', 'small',
       'small', 'large', 'small', 'large', 'small', 'small', 'small',
       'small', 'large', 'small', 'small', 'small', 'large', 'small',
       'small', 'small', 'large', 'small', 'small', 'small', 'small',
       'small', 'large', 'small', 'small', 'large', 'small', 'large',
       'small', 'small', 'small', 'large', 'large', 'small', 'large',
       'small', 'small', 'large', 'large', 'small', 'large', 'large',
       'large', 'large', 'small', 'large', 'small', 'large', 'large',
       'large', 'small', 'small', 'large', 'small', 'small', 'small',
       'small', 'large', 'small', 'small', 'small', 'large', 'small',
       'small', 'small', 'large', 'small', 'small', 'small', 'small',
       'small', 'small', 'large', 'large', 'small', 'small', 'small',
       'large', 'small', 'large', 'small', 'small', 'small', 'small',
       'small', 'small', 'small', 'small', 'large', 'small', 'large',
       'small', 'small', 'small', 'small', 'large', 'small', 'small',
       'small', 'small', 'small', 'large', 'small', 'small', 'small',
       'small', 'small', 'large', 'large', 'small', 'small', 'small',
       'small', 'small', 'large', 'small', 'small', 'small', 'small',
       'small', 'small', 'small', 'large', 'small', 'small', 'large',
       'large', 'small', 'large', 'small', 'large', 'large', 'small',
       'small', 'large', 'small', 'large', 'small', 'small', 'small',
       'small', 'large', 'small', 'small', 'large', 'large', 'small',
       'small', 'small', 'small', 'large', 'large', 'small', 'small',
       'large', 'large', 'large', 'small', 'small', 'small', 'small',
       'large', 'small', 'small', 'small', 'small', 'small', 'small',
       'small', 'large', 'large', 'large', 'small', 'small'], dtype=object)
```

```
X_train, X_test, y_train, y_test = train_test_split(X,Y, test_size = 0.3)
```

```
X_train.shape, y_train.shape, X_test.shape, y_test.shape
```

```
((361, 28), (361,), (156, 28), (156,))
```

```
clf = SVC()
param_grid = [{'kernel':['rbf'],'gamma':[50,5,10,0.5],'C':[15,14,13,12,11,10,0.1,0.
gsv = GridSearchCV(clf,param_grid,cv=10)
gsv.fit(X_train,y_train)
```




```
gsv.best_params_ , gsv.best_score_
```

```
({'C': 15, 'gamma': 0.5, 'kernel': 'rbf'}, 0.7451951951951952)
```

```
clf = SVC(C= 15, gamma = 50)
clf.fit(X_train , y_train)
y_pred = clf.predict(X_test)
acc = accuracy_score(y_test, y_pred) * 100
print("Accuracy =", acc)
confusion_matrix(y_test, y_pred)
```

```
Accuracy = 71.7948717948718
array([[  1,  44],
       [  0, 111]])
```

```
clf = SVC(kernel= "linear")
clf.fit(X_train , y_train)
y_pred = clf.predict(X_test)
acc = accuracy_score(y_test, y_pred) * 100
print("Accuracy =", acc)
confusion_matrix(y_test, y_pred)
```

```
Accuracy = 98.71794871794873
array([[ 44,   1],
       [  1, 110]])
```

```
clf = SVC(kernel= "poly")
clf.fit(X_train , y_train)
y_pred = clf.predict(X_test)
acc = accuracy_score(y_test, y_pred) * 100
print("Accuracy =", acc)
confusion_matrix(y_test, y_pred)
```

```
Accuracy = 76.92307692307693
array([[  9,  36],
       [  0, 111]])
```