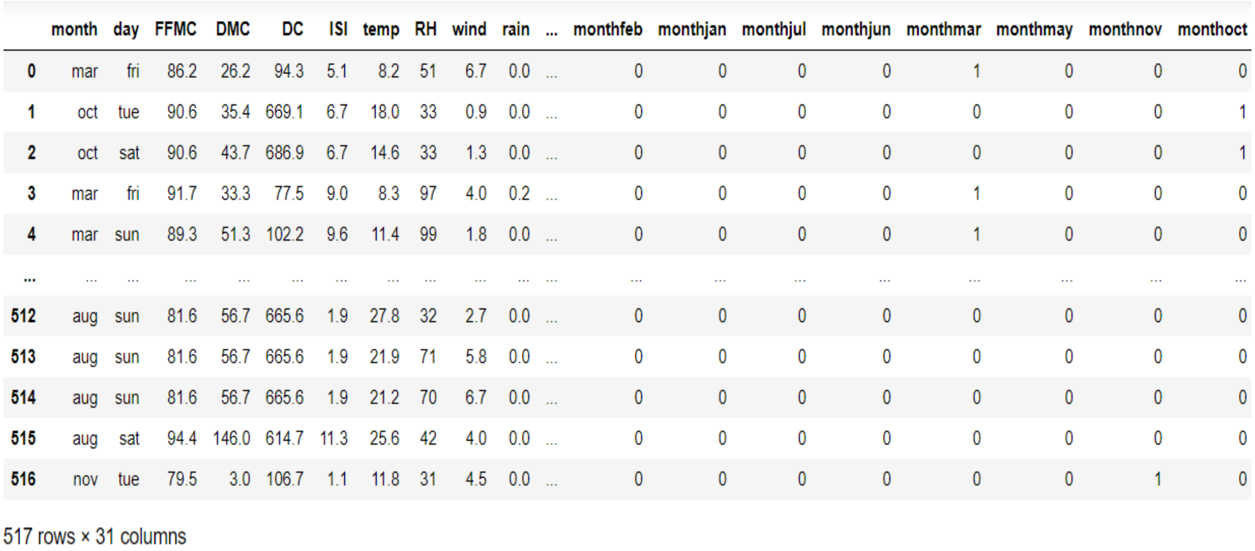
—-------------Important Libraries—--------------------

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn import svm
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split,cross_val_score
from sklearn.metrics import confusion_matrix,accuracy_score
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import classification_report
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
```

—-----------Read Dataset—-------------------

```
data = pd.read_csv('Downloads/forestfires.csv')
data
```

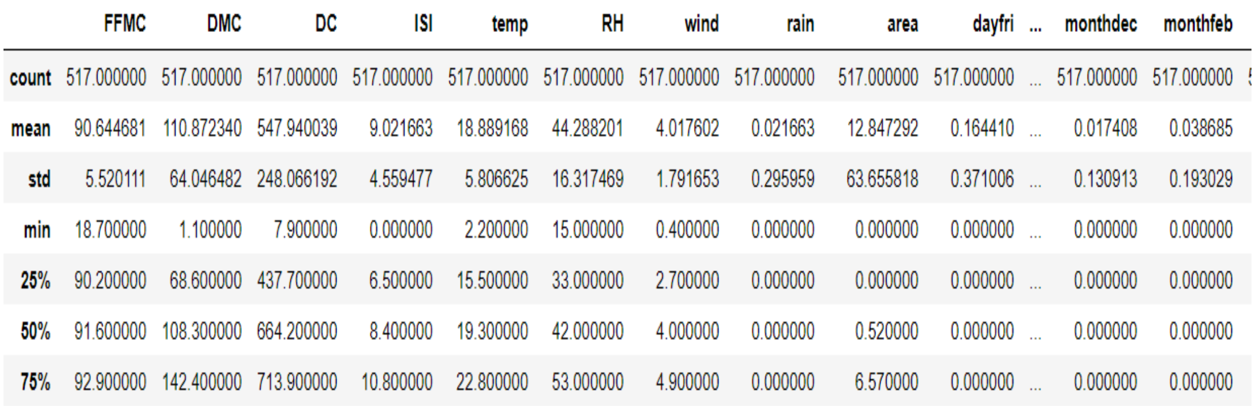


	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	...	monthfeb	monthjan	monthjul	monthjun	monthmar	monthmay	monthnov	monthoct
0	mar	fri	86.2	26.2	94.3	5.1	8.2	51	6.7	0.0	...	0	0	0	0	1	0	0	0
1	oct	tue	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	...	0	0	0	0	0	0	0	1
2	oct	sat	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	...	0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

517 rows × 31 columns

—---------Data Description—-------------

```
data.describe()
```

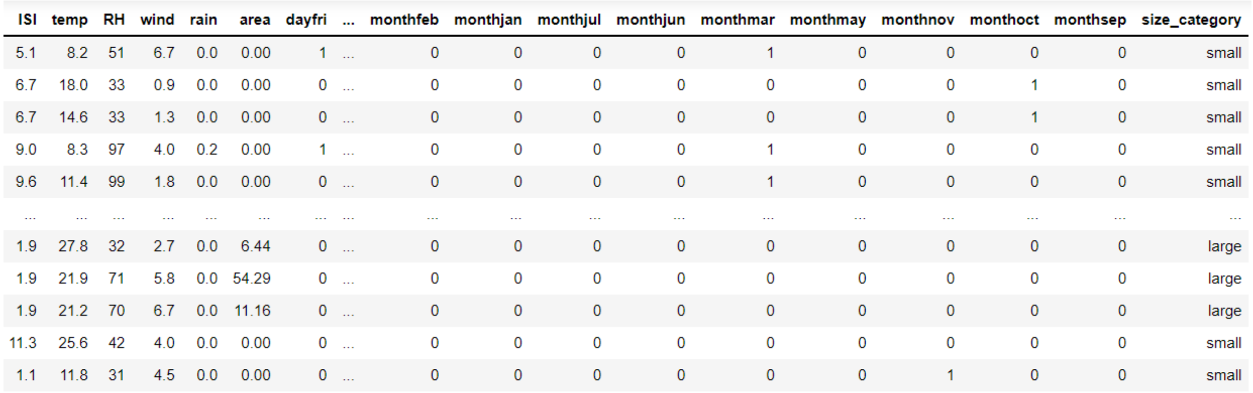


	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area	dayfri	...	monthdec	monthfeb
count	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	...	517.000000	517.000000
mean	90.644681	110.872340	547.940039	9.021663	18.889168	44.288201	4.017602	0.021663	12.847292	0.164410	...	0.017408	0.038685
std	5.520111	64.046482	248.066192	4.559477	5.806625	16.317469	1.791653	0.295959	63.655818	0.371006	...	0.130913	0.193029
min	18.700000	1.100000	7.900000	0.000000	2.200000	15.000000	0.400000	0.000000	0.000000	0.000000	...	0.000000	0.000000
25%	90.200000	68.600000	437.700000	6.500000	15.500000	33.000000	2.700000	0.000000	0.000000	0.000000	...	0.000000	0.000000
50%	91.600000	108.300000	664.200000	8.400000	19.300000	42.000000	4.000000	0.000000	0.520000	0.000000	...	0.000000	0.000000
75%	92.900000	142.400000	713.900000	10.800000	22.800000	53.000000	4.900000	0.000000	6.570000	0.000000	...	0.000000	0.000000

—----------Drop columns—--------------------

data = data.drop(['month','day'],axis=1)

data



	ISI	temp	RH	wind	rain	area	dayfri	...	monthfeb	monthjan	monthjul	monthjun	monthmar	monthmay	monthnov	monthoct	monthsep	size_category
	5.1	8.2	51	6.7	0.0	0.00	1	...	0	0	0	0	1	0	0	0	0	small
	6.7	18.0	33	0.9	0.0	0.00	0	...	0	0	0	0	0	0	0	1	0	small
	6.7	14.6	33	1.3	0.0	0.00	0	...	0	0	0	0	0	0	0	1	0	small
	9.0	8.3	97	4.0	0.2	0.00	1	...	0	0	0	0	1	0	0	0	0	small
	9.6	11.4	99	1.8	0.0	0.00	0	...	0	0	0	0	1	0	0	0	0	small
	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
	1.9	27.8	32	2.7	0.0	6.44	0	...	0	0	0	0	0	0	0	0	0	large
	1.9	21.9	71	5.8	0.0	54.29	0	...	0	0	0	0	0	0	0	0	0	large
	1.9	21.2	70	6.7	0.0	11.16	0	...	0	0	0	0	0	0	0	0	0	large
	11.3	25.6	42	4.0	0.0	0.00	0	...	0	0	0	0	0	0	0	0	0	small
	1.1	11.8	31	4.5	0.0	0.00	0	...	0	0	0	0	0	0	1	0	0	small

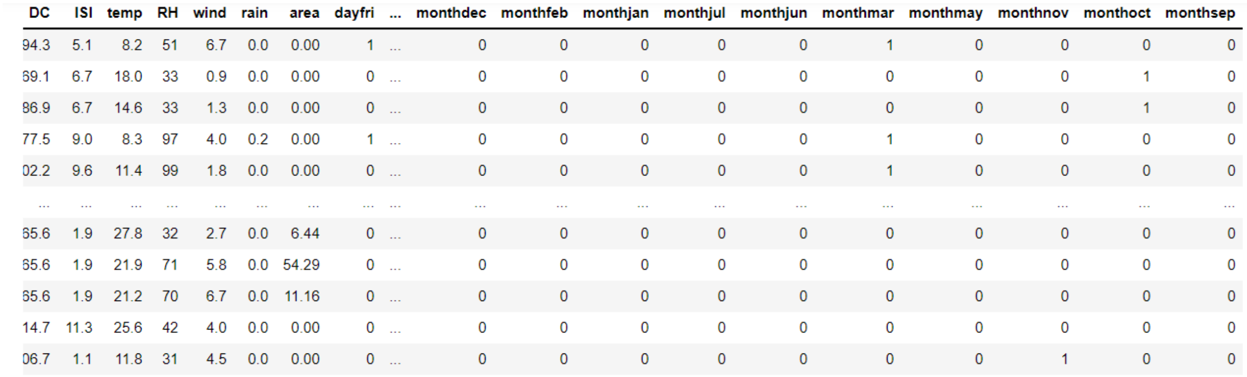
—----------Divide Data—---------------------

x = data.iloc[:,0:28]

y = data.iloc[:,28]

—----------Read x—------------------------

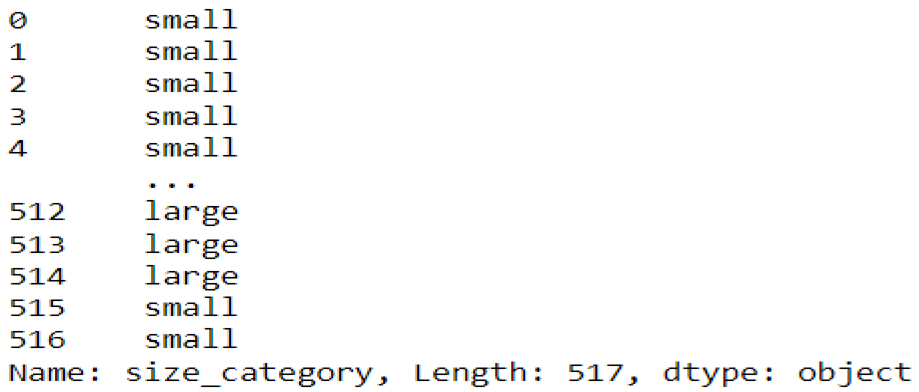
x



DC	ISI	temp	RH	wind	rain	area	dayfri	...	monthdec	monthfeb	monthjan	monthjul	monthjun	monthmar	monthmay	monthnov	monthoct	monthsep
94.3	5.1	8.2	51	6.7	0.0	0.00	1	...	0	0	0	0	0	1	0	0	0	0
69.1	6.7	18.0	33	0.9	0.0	0.00	0	...	0	0	0	0	0	0	0	0	1	0
86.9	6.7	14.6	33	1.3	0.0	0.00	0	...	0	0	0	0	0	0	0	0	1	0
77.5	9.0	8.3	97	4.0	0.2	0.00	1	...	0	0	0	0	0	1	0	0	0	0
02.2	9.6	11.4	99	1.8	0.0	0.00	0	...	0	0	0	0	0	1	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
65.6	1.9	27.8	32	2.7	0.0	6.44	0	...	0	0	0	0	0	0	0	0	0	0
65.6	1.9	21.9	71	5.8	0.0	54.29	0	...	0	0	0	0	0	0	0	0	0	0
65.6	1.9	21.2	70	6.7	0.0	11.16	0	...	0	0	0	0	0	0	0	0	0	0
14.7	11.3	25.6	42	4.0	0.0	0.00	0	...	0	0	0	0	0	0	0	0	0	0
06.7	1.1	11.8	31	4.5	0.0	0.00	0	...	0	0	0	0	0	0	0	1	0	0

—--------Read y—------------------

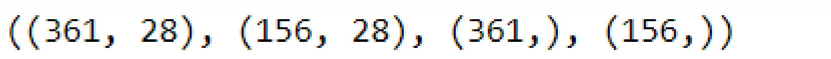
y



```
0      small
1      small
2      small
3      small
4      small
       ...
512    large
513    large
514    large
515    small
516    small
Name: size_category, Length: 517, dtype: object
```

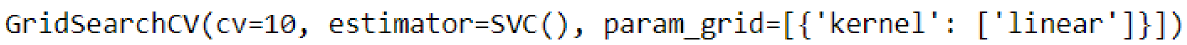
```
—-----------Train test—---------------------
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3)
```

```
—-----------Read shape—---------------------
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

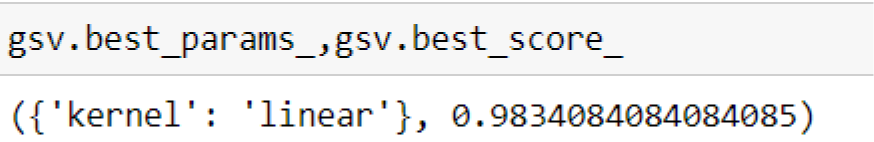


```
((361, 28), (156, 28), (361,), (156,))
```

```
—-----------Define SVM—-----------------
clf = SVC()
param_grid = [{'kernel':['linear']}]
gsv = GridSearchCV (clf,param_grid,cv=10)
gsv.fit(x_train,y_train)
```



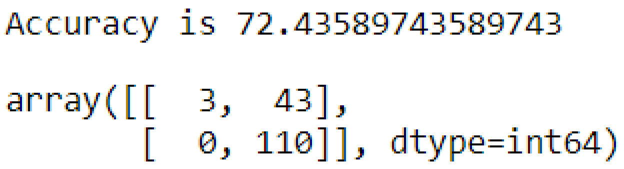
```
GridSearchCV(cv=10, estimator=SVC(), param_grid=[{'kernel': ['linear']}])
```



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

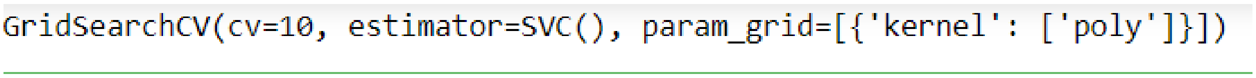
```
({'kernel': 'linear'}, 0.9834084084084085)
```

```
—------------Find Accuracy—----------
clf = SVC()
clf.fit(x_train,y_train)
ypredict = clf.predict(x_test)
accuracy = accuracy_score(y_test,ypredict)*100
print('Accuracy is',accuracy)
confusion_matrix(y_test,ypredict)
```

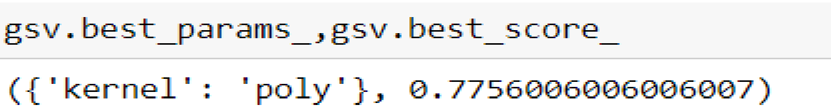


```
Accuracy is 72.43589743589743

array([[  3,  43],
       [  0, 110]], dtype=int64)
```

```
—----------Define SVM—-----------------
clf = SVC()
param_grid = [{'kernel':['poly']}]
gsv = GridSearchCV(clf,param_grid,cv=10)
gsv.fit(x_train,y_train)
```



```
GridSearchCV(cv=10, estimator=SVC(), param_grid=[{'kernel': ['poly']}])
```

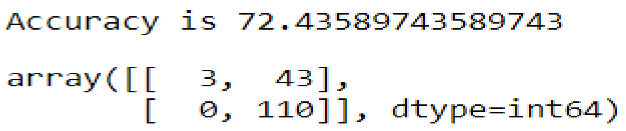


```
gsv.best_params_,gsv.best_score_
```

```
({'kernel': 'poly'}, 0.7756006006006007)
```

—----------Find Accuracy—------------------

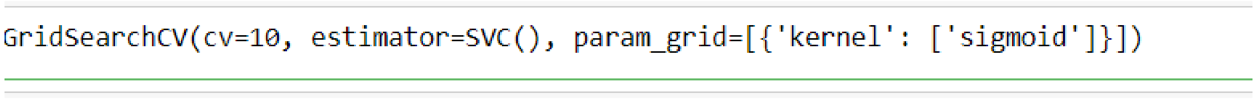
```
clf = SVC()
clf.fit(x_train,y_train)
ypredict = clf.predict(x_test)
accuracy = accuracy_score(y_test,ypredict)*100
print('Accuracy is',accuracy)
confusion_matrix(y_test,ypredict)
```



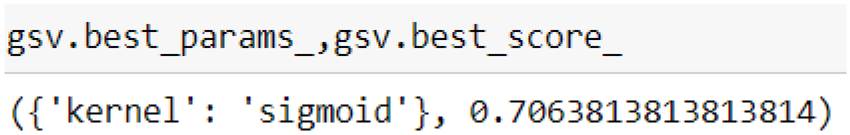
```
Accuracy is 72.43589743589743

array([[  3,  43],
       [  0, 110]], dtype=int64)
```

—----------Define SVM—----------------

```
clf = SVC()
param_grid = [{'kernel':['sigmoid']}]
gsv = GridSearchCV(clf,param_grid,cv=10)
gsv.fit(x_train,y_train)
```



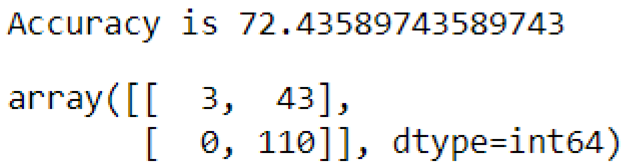
```
GridSearchCV(cv=10, estimator=SVC(), param_grid=[{'kernel': ['sigmoid']}])
```



```
gsv.best_params_,gsv.best_score_

({'kernel': 'sigmoid'}, 0.7063813813813814)
```

—---------Find accuracy—-------------------

```
clf = SVC()
clf.fit(x_train,y_train)
ypredict = clf.predict(x_test)
accuracy = accuracy_score(y_test,ypredict)*100
print('Accuracy is',accuracy)
confusion_matrix(y_test,ypredict)
```



```
Accuracy is 72.43589743589743

array([[  3,  43],
       [  0, 110]], dtype=int64)
```