

Import Necessary libraries

```
In [1]: import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer

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
```

Import data

```
In [2]: data = pd.read_csv('forestfires.csv')
data
```

```
Out[2]:
```

	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	...	monthfeb	monthjan	mont
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



In [3]: data.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
```

In [4]: data.describe().T

Out[4]:

	count	mean	std	min	25%	50%	75%	max
FFMC	517.0	90.644681	5.520111	18.7	90.2	91.60	92.90	96.20
DMC	517.0	110.872340	64.046482	1.1	68.6	108.30	142.40	291.30
DC	517.0	547.940039	248.066192	7.9	437.7	664.20	713.90	860.60
ISI	517.0	9.021663	4.559477	0.0	6.5	8.40	10.80	56.10
temp	517.0	18.889168	5.806625	2.2	15.5	19.30	22.80	33.30
RH	517.0	44.288201	16.317469	15.0	33.0	42.00	53.00	100.00
wind	517.0	4.017602	1.791653	0.4	2.7	4.00	4.90	9.40
rain	517.0	0.021663	0.295959	0.0	0.0	0.00	0.00	6.40
area	517.0	12.847292	63.655818	0.0	0.0	0.52	6.57	1090.84
dayfri	517.0	0.164410	0.371006	0.0	0.0	0.00	0.00	1.00
daymon	517.0	0.143133	0.350548	0.0	0.0	0.00	0.00	1.00
daysat	517.0	0.162476	0.369244	0.0	0.0	0.00	0.00	1.00
daysun	517.0	0.183752	0.387657	0.0	0.0	0.00	0.00	1.00
daythu	517.0	0.117988	0.322907	0.0	0.0	0.00	0.00	1.00
daytue	517.0	0.123791	0.329662	0.0	0.0	0.00	0.00	1.00
daywed	517.0	0.104449	0.306138	0.0	0.0	0.00	0.00	1.00
monthapr	517.0	0.017408	0.130913	0.0	0.0	0.00	0.00	1.00
monthaug	517.0	0.355899	0.479249	0.0	0.0	0.00	1.00	1.00
monthdec	517.0	0.017408	0.130913	0.0	0.0	0.00	0.00	1.00
monthfeb	517.0	0.038685	0.193029	0.0	0.0	0.00	0.00	1.00
monthjan	517.0	0.003868	0.062137	0.0	0.0	0.00	0.00	1.00
monthjul	517.0	0.061896	0.241199	0.0	0.0	0.00	0.00	1.00
monthjun	517.0	0.032882	0.178500	0.0	0.0	0.00	0.00	1.00
monthmar	517.0	0.104449	0.306138	0.0	0.0	0.00	0.00	1.00
monthmay	517.0	0.003868	0.062137	0.0	0.0	0.00	0.00	1.00
monthnov	517.0	0.001934	0.043980	0.0	0.0	0.00	0.00	1.00
monthoct	517.0	0.029014	0.168007	0.0	0.0	0.00	0.00	1.00
monthsep	517.0	0.332689	0.471632	0.0	0.0	0.00	1.00	1.00

In [5]: *# Dropping column which are not required*

```
data = data.drop(['dayfri', 'daymon', 'daysat', 'daysun', 'daythu', 'daytue', 'day',
                  'monthfeb', 'monthjan', 'monthjul', 'monthjun', 'monthmar', 'month',
                  axis = 1)
data
```

Out[5]:

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

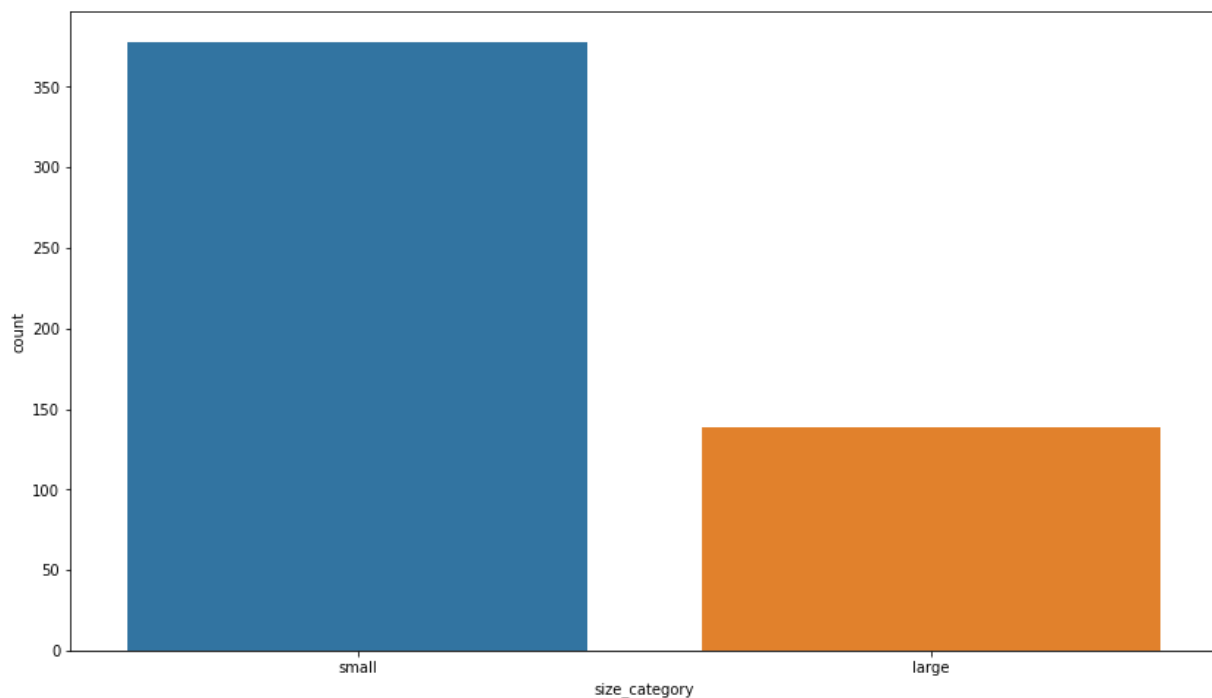
517 rows × 12 columns

In [6]: *#Checking how much data points have large and small area*

```
data.size_category.value_counts()
```

Out[6]: small 378
large 139
Name: size_category, dtype: int64

```
In [7]: import seaborn as sns
from matplotlib import pyplot as plt
plt.figure(figsize=(14,8))
sns.countplot(x= 'size_category',data= data)
plt.show()
```



```
In [8]: #Checking for which value of area is categorised into large and small by creating
pd.crosstab(data.area, data.size_category)
```

```
Out[8]: size_category large small
```

area		
	large	small
0.00	0	247
0.09	0	1
0.17	0	1
0.21	0	1
0.24	0	1
...
200.94	1	0
212.88	1	0
278.53	1	0
746.28	1	0
1090.84	1	0

251 rows × 2 columns

In [9]: `data.corr()`

Out[9]:

	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area
FFMC	1.000000	0.382619	0.330512	0.531805	0.431532	-0.300995	-0.028485	0.056702	0.040122
DMC	0.382619	1.000000	0.682192	0.305128	0.469594	0.073795	-0.105342	0.074790	0.072994
DC	0.330512	0.682192	1.000000	0.229154	0.496208	-0.039192	-0.203466	0.035861	0.049383
ISI	0.531805	0.305128	0.229154	1.000000	0.394287	-0.132517	0.106826	0.067668	0.008258
temp	0.431532	0.469594	0.496208	0.394287	1.000000	-0.527390	-0.227116	0.069491	0.097844
RH	-0.300995	0.073795	-0.039192	-0.132517	-0.527390	1.000000	0.069410	0.099751	-0.075519
wind	-0.028485	-0.105342	-0.203466	0.106826	-0.227116	0.069410	1.000000	0.061119	0.012317
rain	0.056702	0.074790	0.035861	0.067668	0.069491	0.099751	0.061119	1.000000	-0.007366
area	0.040122	0.072994	0.049383	0.008258	0.097844	-0.075519	0.012317	-0.007366	1.000000

In [10]: `# Encoding categorical data of month and day feature`
`data.month.replace(('jan', 'feb', 'mar', 'apr', 'may', 'jun', 'jul', 'aug', 'sep', 'oct', 'nov', 'dec'),`
`(1,2,3,4,5,6,7,8,9,10,11,12), inplace=True)`
`data.day.replace(('mon', 'tue', 'wed', 'thu', 'fri', 'sat', 'sun'),(1,2,3,4,5,6,7), inplace=True)`
`data.head()`

Out[10]:

	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area	size_category
0	3	5	86.2	26.2	94.3	5.1	8.2	51	6.7	0.0	0.0	small
1	10	2	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	0.0	small
2	10	6	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	0.0	small
3	3	5	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2	0.0	small
4	3	7	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0	0.0	small

In [11]: *# Encoding target variable 'size category'*

```
data.size_category.replace(('small', 'large'), (0, 1), inplace = True)
data
```

Out[11]:

	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area	size_category
0	3	5	86.2	26.2	94.3	5.1	8.2	51	6.7	0.0	0.00	0
1	10	2	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	0.00	0
2	10	6	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	0.00	0
3	3	5	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2	0.00	0
4	3	7	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0	0.00	0
...
512	8	7	81.6	56.7	665.6	1.9	27.8	32	2.7	0.0	6.44	1
513	8	7	81.6	56.7	665.6	1.9	21.9	71	5.8	0.0	54.29	1
514	8	7	81.6	56.7	665.6	1.9	21.2	70	6.7	0.0	11.16	1
515	8	6	94.4	146.0	614.7	11.3	25.6	42	4.0	0.0	0.00	0
516	11	2	79.5	3.0	106.7	1.1	11.8	31	4.5	0.0	0.00	0

517 rows × 12 columns

In [12]: *# Splitting data into test data and train data*

```
from sklearn.model_selection import train_test_split
```

```
x_train, x_test, y_train, y_test = train_test_split(data, data['size_category'], t
```

In [13]: x_train

Out[13]:

	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area	size_category
311	9	7	92.4	105.8	758.1	9.9	24.8	28	1.8	0.0	14.29	1
368	9	6	91.2	94.3	744.4	8.4	16.8	47	4.9	0.0	12.64	1
23	8	6	90.2	110.9	537.4	6.2	19.5	43	5.8	0.0	0.00	0
271	8	2	92.1	152.6	658.2	14.3	20.1	58	4.5	0.0	9.27	1
299	6	6	53.4	71.0	233.8	0.4	10.6	90	2.7	0.0	0.00	0
...
71	9	5	94.3	85.1	692.3	15.9	17.7	37	3.6	0.0	0.00	0
106	3	4	91.4	30.7	74.3	7.5	18.2	29	3.1	0.0	0.00	0
270	8	2	92.1	152.6	658.2	14.3	21.8	56	3.1	0.0	0.52	0
435	7	6	90.8	84.7	376.6	5.6	23.8	51	1.8	0.0	0.00	0
102	8	2	94.8	108.3	647.1	17.0	20.1	40	4.0	0.0	0.00	0

361 rows × 12 columns

In [14]: x_test

Out[14]:

	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area	size_category
304	5	6	85.1	28.0	113.8	3.5	11.3	94	4.9	0.0	0.00	0
501	8	2	96.1	181.1	671.2	14.3	21.6	65	4.9	0.8	0.00	0
441	8	1	92.1	207.0	672.6	8.2	25.5	29	1.8	0.0	1.23	0
153	9	5	94.3	85.1	692.3	15.9	20.1	47	4.9	0.0	1.46	0
503	8	3	94.5	139.4	689.1	20.0	29.2	30	4.9	0.0	1.95	0
...
192	8	2	94.8	108.3	647.1	17.0	24.6	22	4.5	0.0	8.71	1
66	9	5	92.4	117.9	668.0	12.2	23.0	37	4.5	0.0	0.00	0
286	7	3	91.2	183.1	437.7	12.5	12.6	90	7.6	0.2	0.00	0
422	7	7	88.9	263.1	795.9	5.2	29.3	27	3.6	0.0	6.30	1
94	8	1	91.1	103.2	638.8	5.8	23.4	22	2.7	0.0	0.00	0

156 rows × 12 columns

In [15]: y_train

Out[15]:

```

311    1
368    1
 23    0
271    1
299    0
...
 71    0
106    0
270    0
435    0
102    0
Name: size_category, Length: 361, dtype: int64

```

In [16]: y_test

Out[16]:

```

304    0
501    0
441    0
153    0
503    0
...
192    1
 66    0
286    0
422    1
 94    0
Name: size_category, Length: 156, dtype: int64

```

Building model with GridSearch CV


```
In [17]: clf = SVC()
param_grid = [{'kernel': ['rbf'], 'gamma': [50, 5, 10, 0.5], 'C': [15, 14, 13, 12, 11, 10, 0.1, 0.001]},
               {'kernel': ['rbf'], 'gamma': [50, 5, 10, 0.5], 'C': [15, 14, 13, 12, 11, 10, 0.1, 0.001]}]
gsv = GridSearchCV(clf, param_grid, cv=10)
gsv.fit(x_train, y_train)
```

```
Out[17]: GridSearchCV(cv=10, estimator=SVC(),
                      param_grid=[{'C': [15, 14, 13, 12, 11, 10, 0.1, 0.001],
                                    'gamma': [50, 5, 10, 0.5], 'kernel': ['rbf']}])
```

```
In [18]: gsv.best_params_ , gsv.best_score_
```

```
Out[18]: ({'C': 15, 'gamma': 50, 'kernel': 'rbf'}, 0.7340840840840841)
```

```
In [19]: 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 = 73.71794871794873

```
Out[19]: array([[115,  0],
                [ 41,  0]], dtype=int64)
```