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

In [3]:
```
a=pd.read_csv(r"C:\All Datasets\forestfires.csv")
a
```

Out[3]:

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

517 rows × 13 columns

In [4]:
```
a.isnull().sum()
```

Out[4]:
```
X        0
Y        0
month    0
day      0
FFMC     0
DMC      0
DC       0
ISI      0
temp     0
RH       0
wind     0
rain     0
area     0
dtype: int64
```

In [5]:
```
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 517 entries, 0 to 516
Data columns (total 13 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   X       517 non-null    int64  
 1   Y       517 non-null    int64  
 2   month   517 non-null    object 
 3   day     517 non-null    object 
 4   FFMC    517 non-null    float64
 5   DMC     517 non-null    float64
 6   DC      517 non-null    float64
 7   ISI     517 non-null    float64
 8   temp    517 non-null    float64
 9   RH      517 non-null    int64  
 10  wind    517 non-null    float64
 11  rain    517 non-null    float64
 12  area    517 non-null    float64
dtypes: float64(8), int64(3), object(2)
memory usage: 52.6+ KB
```

In [6]:
```
from sklearn.preprocessing import LabelEncoder
l=LabelEncoder()
a["month"]=l.fit_transform(a["month"])
a["day"]=l.fit_transform(a["day"])
```

In [7]:
```
a
```

Out[7]:

	X	Y	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area
0	7	5	7	0	86.2	26.2	94.3	5.1	8.2	51	6.7	0.0	0.00
1	7	4	10	5	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	0.00
2	7	4	10	2	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	0.00
3	8	6	7	0	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2	0.00
4	8	6	7	3	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0	0.00
...	...	...	...	...	...	...	...	...	...	...	...	...	...
512	4	3	1	3	81.6	56.7	665.6	1.9	27.8	32	2.7	0.0	6.44
513	2	4	1	3	81.6	56.7	665.6	1.9	21.9	71	5.8	0.0	54.29
514	7	4	1	3	81.6	56.7	665.6	1.9	21.2	70	6.7	0.0	11.16
515	1	4	1	2	94.4	146.0	614.7	11.3	25.6	42	4.0	0.0	0.00
516	6	3	9	5	79.5	3.0	106.7	1.1	11.8	31	4.5	0.0	0.00

517 rows × 13 columns

In [20]:
```
x=a.iloc[:,2:].values
y=a.iloc[:,0].values
```

In [21]:
```
from sklearn.preprocessing import StandardScaler
s=StandardScaler()
x1=s.fit_transform(x)
```

In [22]:
```
x1
```

Out[22]:
```
array([[ 0.28422225, -1.42312073, -0.80595947, ...,  1.49861442,
        -0.07326831, -0.20201979],
       [ 0.97087134,  1.17671466, -0.00810203, ..., -1.74175564,
        -0.07326831, -0.20201979],
       [ 0.97087134, -0.38318657, -0.00810203, ..., -1.51828184,
        -0.07326831, -0.20201979],
       ...,
       [-1.08907592,  0.13678051, -1.64008316, ...,  1.49861442,
        -0.07326831, -0.02653216],
       [-1.08907592, -0.38318657,  0.68095666, ..., -0.00983371,
        -0.07326831, -0.20201979],
       [ 0.74198831,  1.17671466, -2.02087875, ...,  0.26950853,
        -0.07326831, -0.20201979]])
```

In [14]:
```
a["X"].value_counts()
```

Out[14]:
```
4    91
6    86
2    73
8    61
7    60
3    55
1    48
5    30
9    13
Name: X, dtype: int64
```

In [23]:
```
from imblearn.over_sampling import RandomOverSampler
rs=RandomOverSampler()
X,Y=rs.fit_resample(x1,y)
```

In [24]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(X,Y,random_state=20,test_size=0.2)
```

In [34]:
```
from sklearn.neighbors import KNeighborsClassifier
kn=KNeighborsClassifier(n_neighbors=5,metric="minkowski",p=2)
kn.fit(x_train,y_train)
```

Out[34]:
```
KNeighborsClassifier()
```

In [35]:
```
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression()
lr.fit(x_train,y_train)
```

Out[35]:
```
LogisticRegression()
```

In [36]:
```
from sklearn.svm import SVC
sv=SVC(kernel="linear")
sv.fit(x_train,y_train)
```

Out[36]:
```
SVC(kernel='linear')
```

In [37]:
```
from sklearn.ensemble import VotingClassifier
vc=VotingClassifier(estimators=[("knn",kn),("logistic",lr),("SVM",sv)])
vc.fit(x_train,y_train)
```

Out[37]:
```
VotingClassifier(estimators=[('knn', KNeighborsClassifier()),
                             ('logistic', LogisticRegression()),
                             ('SVM', SVC(kernel='linear'))])
```

In [38]:
```
pred=vc.predict(x_test)
```

In [39]:
```
pred
```

Out[39]:
```
array([9, 9, 1, 2, 9, 5, 3, 9, 5, 2, 2, 4, 3, 3, 2, 8, 9, 1, 5, 2, 1, 9,
       1, 2, 3, 8, 2, 1, 1, 2, 2, 2, 9, 2, 3, 1, 9, 1, 3, 2, 5, 2, 1, 1,
       5, 7, 7, 3, 3, 8, 3, 7, 8, 5, 5, 1, 2, 1, 5, 2, 5, 4, 1, 2, 1, 1,
       9, 1, 3, 3, 2, 9, 9, 4, 2, 3, 1, 9, 1, 3, 9, 3, 6, 9, 9, 2, 7, 2,
       7, 1, 9, 5, 3, 9, 5, 9, 1, 9, 2, 8, 5, 2, 1, 1, 7, 3, 5, 1, 5, 8,
       3, 2, 2, 1, 7, 9, 6, 9, 6, 3, 4, 5, 7, 2, 1, 8, 2, 6, 9, 9, 2, 6,
       3, 2, 1, 4, 9, 9, 2, 5, 1, 4, 1, 1, 6, 6, 5, 7, 9, 5, 1, 3, 6, 7,
       2, 8, 9, 9, 2, 2, 2, 2, 6, 7], dtype=int64)
```

In [42]:
```
from sklearn.metrics import accuracy_score
accuracy_score(pred,y_test)*100
```

Out[42]:
```
21.951219512195124
```

Cross Validation

In [44]:
```
from sklearn.model_selection import StratifiedKFold
st=StratifiedKFold(n_splits=5,random_state=10,shuffle=True)
st.get_n_splits(x_train,y_train)
```

Out[44]:
```
5
```

In [45]:
```
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import cross_val_predict
from sklearn.metrics import accuracy_score
```

In [46]:
```
scores=cross_val_score(vc,x_train,y_train)
pred2=cross_val_predict(vc,x_test,y_test)
s2=accuracy_score(pred2,y_test)*100
print((scores)*100)
print(pred2)
print(s2)
```

```
[25.95419847 29.77099237 32.0610687  27.48091603 26.71755725]
[2 9 6 1 2 3 4 2 3 3 2 6 5 6 3 2 9 6 1 8 1 5 6 1 4 2 4 1 8 1 7 2 9 4 1 8 9
 6 1 3 4 2 4 2 4 1 7 6 7 1 1 4 8 6 3 6 1 6 6 3 3 3 8 1 8 6 8 6 6 3 7 9 8 6
 6 2 6 8 6 6 4 4 7 9 2 8 1 5 3 6 1 3 6 2 3 2 1 9 2 8 5 4 6 6 6 6 7 7 7 8 1
 1 3 6 7 9 5 2 4 6 4 6 3 5 8 8 2 4 9 4 2 7 6 8 6 3 9 9 9 3 1 6 6 6 2 7 3 7
 2 3 6 7 7 6 2 1 8 9 2 4 2 7 3 3]
21.341463414634145
```

In [ ]: