Assignment-17-Support_Vector_Machines-01 Forest Fires

In [1]:

```
# SVM Classification
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
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
```

In [2]:

```
# load dataset
dataframe= pd.read_csv("C:/Users/LENOVO/Documents/assignment/forestfires.csv")
dataframe
```

Out[2]:

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

```
# Encode Data
dataframe.month.replace(('jan','feb','mar','apr','may','jun','jul','aug','sep','oct','nov',
dataframe.day.replace(('mon','tue','wed','thu','fri','sat','sun'),(1,2,3,4,5,6,7), inplace=

print("Head:", dataframe.head())
```

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

   monthjan  monthjul  monthjun  monthmar  monthmay  monthnov  monthoct  \
0         0         0         0         1         0         0         0
1         0         0         0         0         0         0         1
2         0         0         0         0         0         0         1
3         0         0         0         1         0         0         0
4         0         0         0         1         0         0         0

   monthsep  size_category
0         0          small
1         0          small
2         0          small
3         0          small
4         0          small

[5 rows x 31 columns]
```

In [4]:

```
#getting information of dataset
dataframe.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    int64  
 1   day            517 non-null    int64  
 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(22), object(1)
memory usage: 125.3+ KB
```

In [5]:

```
dataframe.drop('monthaug',axis='columns', inplace=True)
dataframe.drop('monthdec',axis='columns', inplace=True)
dataframe.drop('monthfeb',axis='columns', inplace=True)
dataframe.drop('monthjan',axis='columns', inplace=True)
dataframe.drop('monthjul',axis='columns', inplace=True)
dataframe.drop('monthjun',axis='columns', inplace=True)
dataframe.drop('monthmar',axis='columns', inplace=True)
dataframe.drop('monthmay',axis='columns', inplace=True)
dataframe.drop('monthnov',axis='columns', inplace=True)
dataframe.drop('monthoct',axis='columns', inplace=True)
dataframe.drop('monthsep',axis='columns', inplace=True)
```

In [6]:

```
#getting information of dataset
dataframe.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 517 entries, 0 to 516
Data columns (total 20 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   month          517 non-null    int64  
 1   day            517 non-null    int64  
 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  size_category  517 non-null    object 
dtypes: float64(8), int64(11), object(1)
memory usage: 80.9+ KB
```

In [7]:

```
dataframe.drop('daysat',axis='columns', inplace=True)
dataframe.drop('daysun',axis='columns', inplace=True)
dataframe.drop('daythu',axis='columns', inplace=True)
dataframe.drop('daytue',axis='columns', inplace=True)
dataframe.drop('daywed',axis='columns', inplace=True)
dataframe.drop('monthapr',axis='columns', inplace=True)
```

In [8]:

```
#getting information of dataset
dataframe.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 517 entries, 0 to 516
Data columns (total 14 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   month          517 non-null    int64  
 1   day            517 non-null    int64  
 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  size_category  517 non-null    object 
dtypes: float64(8), int64(5), object(1)
memory usage: 56.7+ KB
```

In [9]:

```
print("Head:", dataframe.head())
```

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

   daymon size_category
0       0         small
1       0         small
2       0         small
3       0         small
4       0         small
```

In [11]:

```
#Creating dummy vairables dropping first dummy variable
df=pd.get_dummies(dataframe,columns=['size_category'], drop_first=True)
```

In [12]:

```
print(df.head())
```

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

   daymon  size_category_small  
0       0                    1  
1       0                    1  
2       0                    1  
3       0                    1  
4       0                    1  
```

In [13]:

```
print("Shape:", dataframe.shape)
```

```
Shape: (517, 14)
```

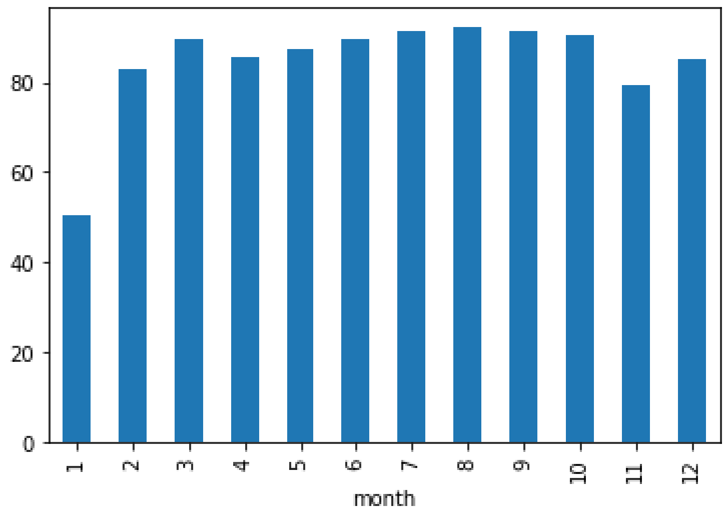
Visualizing the data for better understanding

In [14]:

```
dataframe.groupby('month').FFMC.mean().plot(kind='bar')
```

Out[14]:

```
<AxesSubplot:xlabel='month'>
```

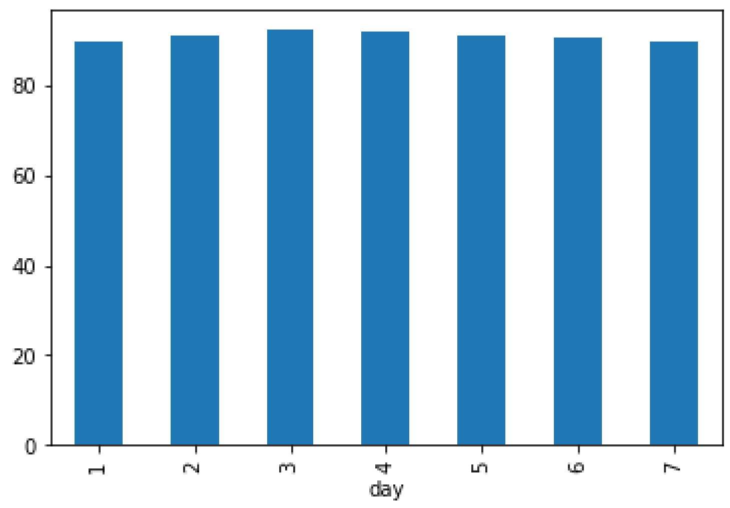


In [15]:

```
dataframe.groupby('day').FFMC.mean().plot(kind='bar')
```

Out[15]:

```
<AxesSubplot:xlabel='day'>
```

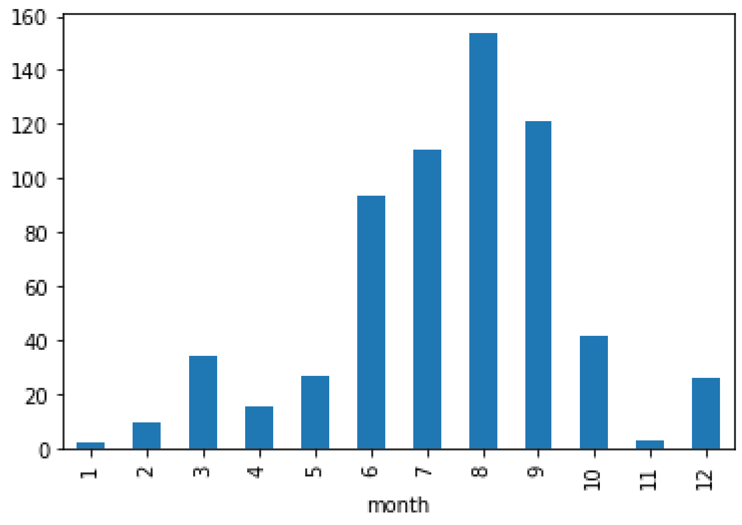


In [16]:

```
dataframe.groupby('month').DMC.mean().plot(kind='bar')
```

Out[16]:

```
<AxesSubplot:xlabel='month'>
```

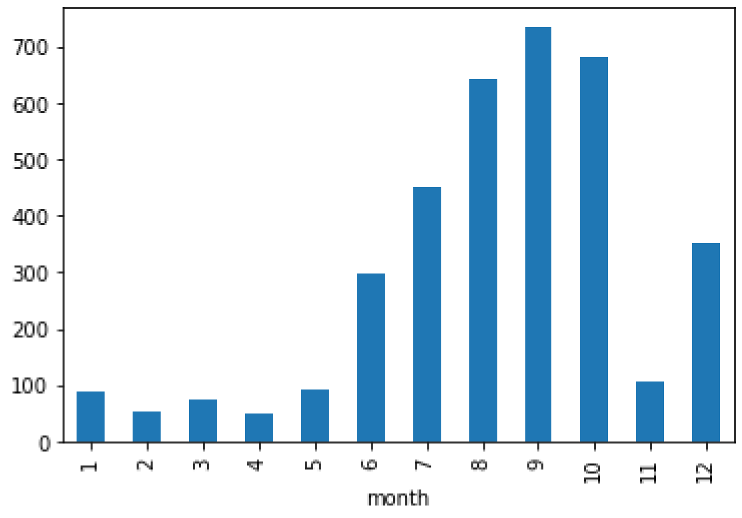


In [17]:

```
dataframe.groupby('month').DC.mean().plot(kind='bar')
```

Out[17]:

```
<AxesSubplot:xlabel='month'>
```

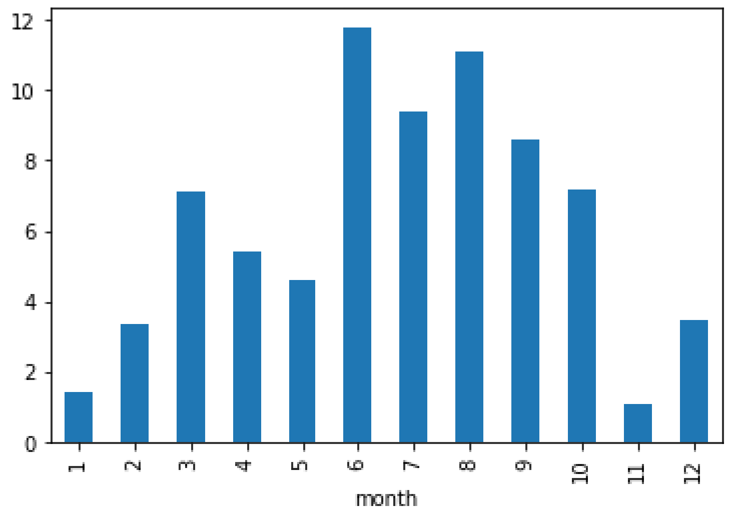


In [18]:

```
dataframe.groupby('month').ISI.mean().plot(kind='bar')
```

Out[18]:

```
<AxesSubplot:xlabel='month'>
```

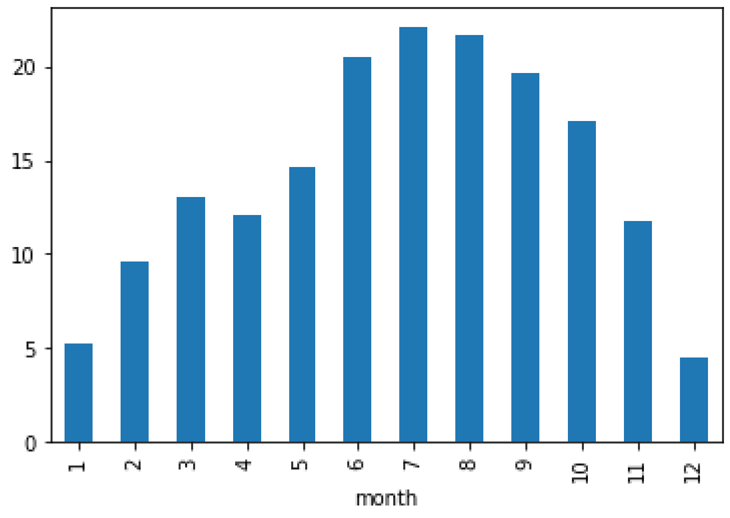


In [19]:

```
dataframe.groupby('month').temp.mean().plot(kind='bar')
```

Out[19]:

```
<AxesSubplot:xlabel='month'>
```

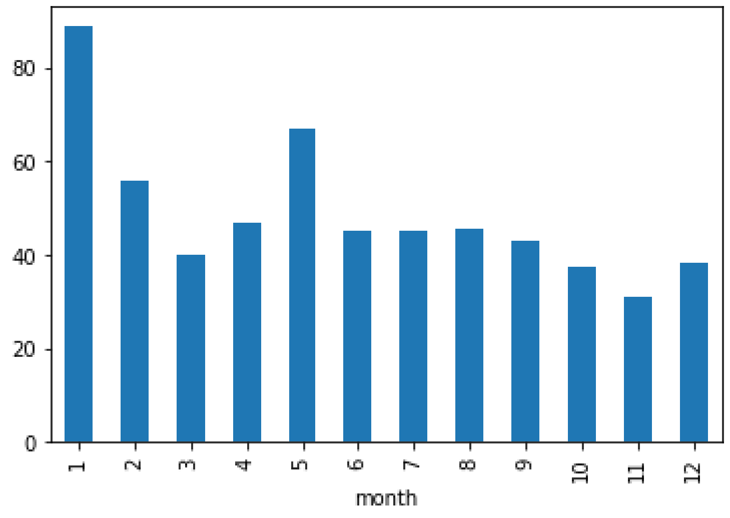


In [20]:

```
dataframe.groupby('month').RH.mean().plot(kind='bar')
```

Out[20]:

```
<AxesSubplot:xlabel='month'>
```

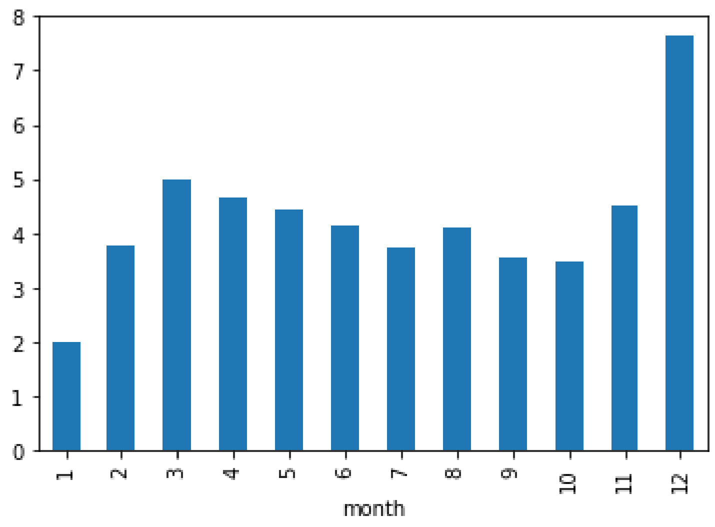


In [21]:

```
dataframe.groupby('month').wind.mean().plot(kind='bar')
```

Out[21]:

```
<AxesSubplot:xlabel='month'>
```

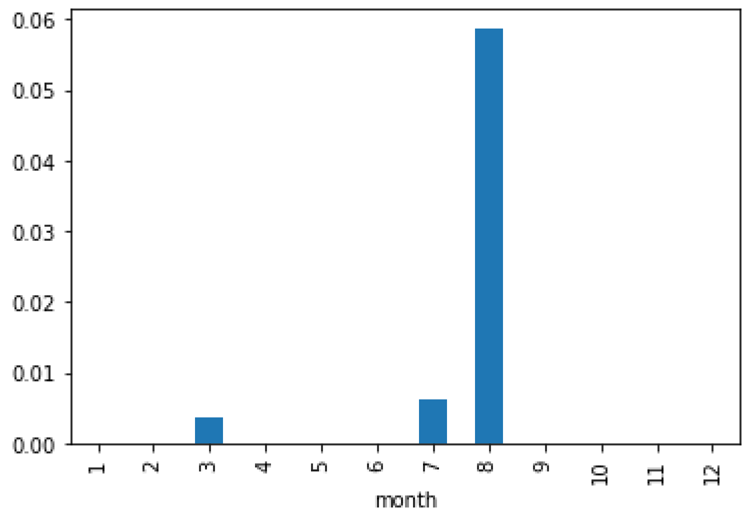


In [22]:

```
dataframe.groupby('month').rain.mean().plot(kind='bar')
```

Out[22]:

```
<AxesSubplot:xlabel='month'>
```

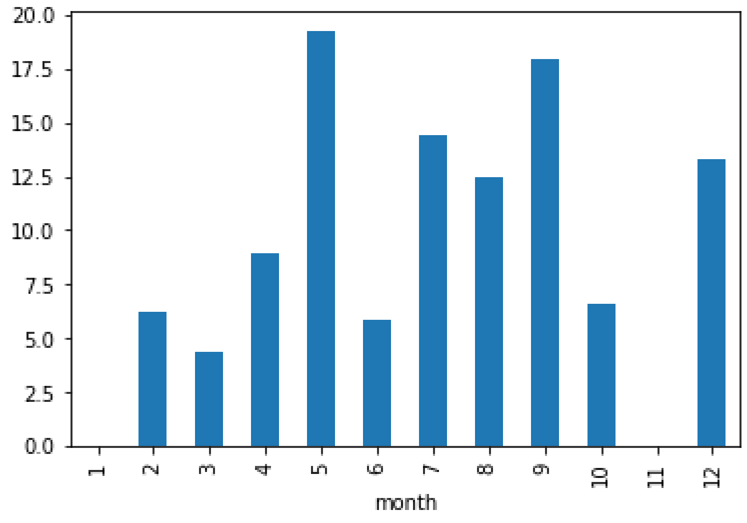


In [23]:

```
dataframe.groupby('month').area.mean().plot(kind='bar')
```

Out[23]:

```
<AxesSubplot:xlabel='month'>
```



Setting up a Support Vector Machine

In [24]:

```
from sklearn.model_selection import train_test_split

# Taking only the features that is important for now
X = dataframe[['FFMC', 'DMC']]

# Taking the labels (Income)
Y = dataframe['month']

# Spliting into 80% for training set and 20% for testing set so we can see our accuracy
X_train, x_test, Y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=0)
```

In [25]:

```
clf = SVC()
param_grid = [{'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)
```

```
C:\Users\LENOVO\anaconda3\lib\site-packages\sklearn\model_selection\_split.p
y:666: UserWarning: The least populated class in y has only 1 members, which
is less than n_splits=10.
  warnings.warn(("The least populated class in y has only %d"
```

Out[25]:

```
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 [26]:

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

Out[26]:

```
({'C': 15, 'gamma': 0.5, 'kernel': 'rbf'}, 0.7944250871080138)
```

In [30]:

```
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 = 85.57692307692307
```

Out[30]:

```
array([[ 1,  0,  0,  0,  0,  6,  0,  0,  0],
       [ 0,  7,  0,  0,  0,  1,  0,  0,  0],
       [ 0,  0,  0,  0,  0,  1,  0,  0,  0],
       [ 0,  0,  0,  2,  0,  1,  0,  0,  0],
       [ 0,  0,  0,  0,  2,  3,  0,  0,  0],
       [ 0,  0,  0,  0,  0, 35,  1,  0,  0],
       [ 0,  0,  0,  0,  0,  1, 39,  0,  0],
       [ 0,  0,  0,  0,  0,  1,  0,  2,  0],
       [ 0,  0,  0,  0,  0,  0,  0,  0,  1]], dtype=int64)
```

In [ ]:

```
```