In [1]:
```
import pandas as pd     # Loading data
df=pd.read_csv(r'https://raw.githubusercontent.com/iiscleap/Coswara-Data/master/combined_data.csv')
```

In [2]:
```
df
```

Out[2]:

	id	a	covid_status	ep	g	l_c	l_l	l_s	rU	smoker	...	diarrhoea	pneumoni
0	iV3Db6t1T8b7c5HQY2TwxIhjbzD3	28	healthy	y	male	India	Anantapur	Andhra Pradesh	n	NaN	...	NaN	Nal
1	AxuYWBN0jFVLINCBqIW5aZmGCdu1	25	healthy	y	male	India	BENGALURU URBAN	Karnataka	n	True	...	NaN	Nal
2	C5eIsssb9GSkaAgIfsHMHeR6fSh1	28	healthy	y	female	United States	Pittsburgh	Pennsylvania	n	NaN	...	NaN	Nal
3	YjbEAECMBIaZKyfqOvWy5DDImUb2	26	healthy	y	male	India	Bangalore	Karnataka	n	NaN	...	NaN	Nal
4	aGOvk4ji0cVqIzCs1jHnzlw2UEy2	32	healthy	y	male	India	Nalanda	Bihar	n	NaN	...	NaN	Nal
...	...	...	...	...	...	...	...	...	...	...	...	...	.
1890	aH6o5lYN6vh40RCFhfyBYjkQVH13	33	healthy	y	male	India	24pgs	West Bengal	n	True	...	NaN	Nal
1891	hRMSJF9xD6h9CmQJzB3phXKNnTI2	42	recovered_full	y	male	India	Bangalore	Karnataka	n	NaN	...	NaN	Nal
1892	ZNTuppbDaCNybeDYGmyjJeYr93s1	27	healthy	y	female	India	NaN	Karnataka	n	NaN	...	NaN	Nal
1893	J7R2UkId7jU0aOSw5amaS8zJVIm1	37	healthy	y	male	India	NaN	Karnataka	n	NaN	...	NaN	Nal
1894	8MdsvTBMWAUInavE1pdmFQAZPF13	38	recovered_full	y	female	India	Bengaluru	Karnataka	n	NaN	...	NaN	Nal

1895 rows × 35 columns

In [3]:
```
df.isnull().sum()     #checking for null
```

Out[3]:
```
id                    0
a                     0
covid_status          0
ep                    0
g                     0
l_c                   0
l_l                 229
l_s                   0
rU                  680
smoker             1660
cold               1753
ht                 1770
diabetes           1784
fever              1768
asthma             1822
um                 1038
cough              1679
ihd                1884
bd                 1825
st                 1810
ftg                1803
mp                 1815
loss_of_smell      1837
cld                1887
test_status        1413
diarrhoea          1886
pneumonia          1885
ctScan             1783
testType           1808
test_date          1808
vacc               1783
ctDate             1884
ctScore            1884
others_resp        1894
others_preexist    1887
dtype: int64
```

In [5]:
```
df["rU"]=df["rU"].fillna("n")       #data - preprocessing
df["um"]=df["um"].fillna("n")
df=df.fillna(False)
```

In [6]:
```
df
```

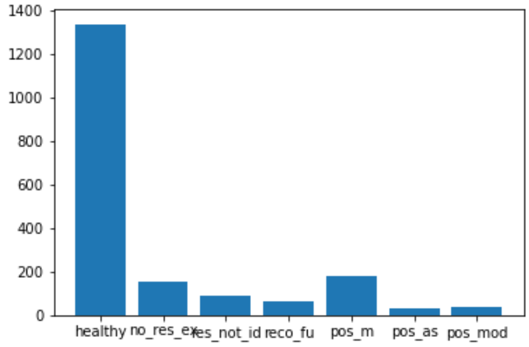
Out[6]:

	id	a	covid_status	ep	g	l_c	l_l	l_s	rU	smoker	...	diarrhoea	pneumoni
0	iV3Db6t1T8b7c5HQY2TwxIhjbzD3	28	healthy	y	male	India	Anantapur	Andhra Pradesh	n	False	...	False	Fals
1	AxuYWBN0jFVLINCBqIW5aZmGCdu1	25	healthy	y	male	India	BENGALURU URBAN	Karnataka	n	True	...	False	Fals
2	C5eIsssb9GSkaAgIfsHMHeR6fSh1	28	healthy	y	female	United States	Pittsburgh	Pennsylvania	n	False	...	False	Fals
3	YjbEAECMBIaZKyfqOvWy5DDImUb2	26	healthy	y	male	India	Bangalore	Karnataka	n	False	...	False	Fals
4	aGOvk4ji0cVqIzCs1jHnzlw2UEy2	32	healthy	y	male	India	Nalanda	Bihar	n	False	...	False	Fals
...	...	...	...	...	...	...	...	...	...	...	...	...	.
1890	aH6o5lYN6vh40RCFhfyBYjkQVH13	33	healthy	y	male	India	24pgs	West Bengal	n	True	...	False	Fals
1891	hRMSJF9xD6h9CmQJzB3phXKNnTI2	42	recovered_full	y	male	India	Bangalore	Karnataka	n	False	...	False	Fals
1892	ZNTuppbDaCNybeDYGmyjJeYr93s1	27	healthy	y	female	India	False	Karnataka	n	False	...	False	Fals
1893	J7R2UkId7jU0aOSw5amaS8zJVIm1	37	healthy	y	male	India	False	Karnataka	n	False	...	False	Fals
1894	8MdsvTBMWAUInavE1pdmFQAZPF13	38	recovered_full	y	female	India	Bengaluru	Karnataka	n	False	...	False	Fals

1895 rows × 35 columns

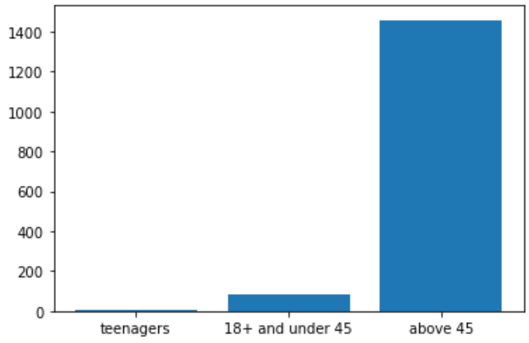
In [7]:
```
identified"]),len(a[a=="recovered_full"]),len(a[a=="positive_mild"]),len(a[a=="positive_asymp"]),len(a[a=="positive_mo
```

Out[7]:
```
<BarContainer object of 7 artists>
```



In [8]:
```
d=df["a"]                #age under covid-status
t1=0
t2=0
t3=0
for i in d:
    if (i<=11 and i<=18):
        t1+=1
    elif (i<=19 and i<=45):
        t2+=1
    elif (i<=46 and i<=100):
        t3+=1
plt.bar(["teenagers","18+ and under 45","above 45"],[t1,t2,t3])
```

Out[8]:
```
<BarContainer object of 3 artists>
```



In [9]:
```
from sklearn import preprocessing
l=preprocessing.LabelEncoder()
#a=df["covid_status"]
df["covid_status"]=l.fit_transform(df["covid_status"])
```

In [10]:
```
df["rU"]=l.fit_transform(df["rU"])
df["um"]=l.fit_transform(df["um"])
df["g"]=l.fit_transform(df["g"])
```

In [11]:
```
df["smoker"]=df["smoker"]*1
```

In [12]:
```
df["cold"]=df["cold"].astype(int)
df
```

Out[12]:

	id	a	covid_status	ep	g	l_c	l_l	l_s	rU	smoker	...	diarrhoea	pneumonia	ct
0	iV3Db6t1T8b7c5HQY2TwxIhjbzD3	28	0	y	1	India	Anantapur	Andhra Pradesh	0	0	...	False	False	
1	AxuYWBN0jFVLINCBqIW5aZmGCdu1	25	0	y	1	India	BENGALURU URBAN	Karnataka	0	True	...	False	False	
2	C5eIsssb9GSkaAgIfsHMHeR6fSh1	28	0	y	0	United States	Pittsburgh	Pennsylvania	0	0	...	False	False	
3	YjbEAECMBIaZKyfqOvWy5DDImUb2	26	0	y	1	India	Bangalore	Karnataka	0	0	...	False	False	
4	aGOvk4ji0cVqIzCs1jHnzlw2UEy2	32	0	y	1	India	Nalanda	Bihar	0	0	...	False	False	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1890	aH6o5lYN6vh40RCFhfyBYjkQVH13	33	0	y	1	India	24pgs	West Bengal	0	True	...	False	False	
1891	hRMSJF9xD6h9CmQJzB3phXKNnTI2	42	5	y	1	India	Bangalore	Karnataka	0	0	...	False	False	
1892	ZNTuppbDaCNybeDYGmyjJeYr93s1	27	0	y	0	India	False	Karnataka	0	0	...	False	False	
1893	J7R2UkId7jU0aOSw5amaS8zJVIm1	37	0	y	1	India	False	Karnataka	0	0	...	False	False	
1894	8MdsvTBMWAUInavE1pdmFQAZPF13	38	5	y	0	India	Bengaluru	Karnataka	0	0	...	False	False	

1895 rows × 35 columns

In [13]:
```
#df["cold"]=df["cold"].astype(int)
df["ht"]=df["ht"].astype(int)
df["diabetes"]=df["diabetes"].astype(int)
df["fever"]=df["fever"].astype(int)
df["asthma"]=df["asthma"].astype(int)
df["cough"]=df["cough"].astype(int)
df["ihd"]=df["ihd"].astype(int)
df["bd"]=df["bd"].astype(int)
df["st"]=df["st"].astype(int)
df["ftg"]=df["ftg"].astype(int)
df["mp"]=l.fit_transform(df["mp"])
df["loss_of_smell"]=df["loss_of_smell"].astype(int)
df["cld"]=df["cld"].astype(int)
#df["test_status"]=df["test_status"].fillna("False")
df["diarrhoea"]=df["diarrhoea"].astype(int)
df["pneumonia"]=df["pneumonia"].astype(int)
```

In [14]:
```
df["smoker"]=df["smoker"].replace("True",1)
#df["smoker"]=df["smoker"].astype(int)
```

In [15]:
```
df
```

Out[15]:

	id	a	covid_status	ep	g	l_c	l_l	l_s	rU	smoker	...	diarrhoea	pneumonia	ct
0	iV3Db6t1T8b7c5HQY2TwxIhjbzD3	28	0	y	1	India	Anantapur	Andhra Pradesh	0	0	...	0	0	
1	AxuYWBN0jFVLINCBqIW5aZmGCdu1	25	0	y	1	India	BENGALURU URBAN	Karnataka	0	1	...	0	0	
2	C5eIsssb9GSkaAgIfsHMHeR6fSh1	28	0	y	0	United States	Pittsburgh	Pennsylvania	0	0	...	0	0	
3	YjbEAECMBIaZKyfqOvWy5DDImUb2	26	0	y	1	India	Bangalore	Karnataka	0	0	...	0	0	
4	aGOvk4ji0cVqIzCs1jHnzlw2UEy2	32	0	y	1	India	Nalanda	Bihar	0	0	...	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1890	aH6o5lYN6vh40RCFhfyBYjkQVH13	33	0	y	1	India	24pgs	West Bengal	0	1	...	0	0	
1891	hRMSJF9xD6h9CmQJzB3phXKNnTI2	42	5	y	1	India	Bangalore	Karnataka	0	0	...	0	0	
1892	ZNTuppbDaCNybeDYGmyjJeYr93s1	27	0	y	0	India	False	Karnataka	0	0	...	0	0	
1893	J7R2UkId7jU0aOSw5amaS8zJVIm1	37	0	y	1	India	False	Karnataka	0	0	...	0	0	
1894	8MdsvTBMWAUInavE1pdmFQAZPF13	38	5	y	0	India	Bengaluru	Karnataka	0	0	...	0	0	

1895 rows × 35 columns

In [16]:
```
df.isnull().sum()
```

Out[16]:
```
id                 0
a                  0
covid_status       0
ep                 0
g                  0
l_c                0
l_l                0
l_s                0
rU                 0
smoker             0
cold               0
ht                 0
diabetes           0
fever              0
asthma             0
um                 0
cough              0
ihd                0
bd                 0
st                 0
ftg                0
mp                 0
loss_of_smell      0
cld                0
test_status        0
diarrhoea          0
pneumonia          0
ctScan             0
testType           0
test_date          0
vacc               0
ctDate             0
ctScore            0
others_resp        0
others_preexist    0
dtype: int64
```

In [17]:
```
df
```

Out[17]:

	id	a	covid_status	ep	g	l_c	l_l	l_s	rU	smoker	...	diarrhoea	pneumonia	ct
0	iV3Db6t1T8b7c5HQY2TwxIhjbzD3	28	0	y	1	India	Anantapur	Andhra Pradesh	0	0	...	0	0	
1	AxuYWBN0jFVLINCBqIW5aZmGCdu1	25	0	y	1	India	BENGALURU URBAN	Karnataka	0	1	...	0	0	
2	C5eIsssb9GSkaAgIfsHMHeR6fSh1	28	0	y	0	United States	Pittsburgh	Pennsylvania	0	0	...	0	0	
3	YjbEAECMBIaZKyfqOvWy5DDImUb2	26	0	y	1	India	Bangalore	Karnataka	0	0	...	0	0	
4	aGOvk4ji0cVqIzCs1jHnzlw2UEy2	32	0	y	1	India	Nalanda	Bihar	0	0	...	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1890	aH6o5lYN6vh40RCFhfyBYjkQVH13	33	0	y	1	India	24pgs	West Bengal	0	1	...	0	0	
1891	hRMSJF9xD6h9CmQJzB3phXKNnTI2	42	5	y	1	India	Bangalore	Karnataka	0	0	...	0	0	
1892	ZNTuppbDaCNybeDYGmyjJeYr93s1	27	0	y	0	India	False	Karnataka	0	0	...	0	0	
1893	J7R2UkId7jU0aOSw5amaS8zJVIm1	37	0	y	1	India	False	Karnataka	0	0	...	0	0	
1894	8MdsvTBMWAUInavE1pdmFQAZPF13	38	5	y	0	India	Bengaluru	Karnataka	0	0	...	0	0	

1895 rows × 35 columns

In [18]:
```
df["covid_status"]
```

Out[18]:
```
0       0
1       0
2       0
3       0
4       0
       ..
1890    0
1891    5
1892    0
1893    0
1894    5
Name: covid_status, Length: 1895, dtype: int32
```

In [19]:
```
df.corr()                               #finding correlation for target column to extraxt the features
```

Out[19]:

	a	covid_status	g	rU	cold	ht	diabetes	fever	asthma	um	...	bd	
a	1.000000	0.070818	-0.004444	0.007648	-0.004834	0.329970	0.323958	0.070876	-0.002322	0.075909	...	0.141344	-0.0
covid_status	0.070818	1.000000	-0.074612	0.037475	0.234706	0.106719	0.089614	0.310393	0.366153	0.247199	...	0.260441	0.1
g	-0.004444	-0.074612	1.000000	0.024445	-0.004895	0.011394	0.048830	0.004336	-0.003138	-0.091514	...	0.016974	-0.0
rU	0.007648	0.037475	0.024445	1.000000	0.059672	-0.019958	0.011304	0.042883	-0.017802	0.098737	...	0.033719	0.0
cold	-0.004834	0.234706	-0.004895	0.059672	1.000000	0.021263	0.031429	0.364572	-0.025726	0.123828	...	0.156791	0.2
ht	0.329970	0.106719	0.011394	-0.019958	0.021263	1.000000	0.268686	0.056307	0.035180	0.010695	...	0.094485	-0.0
diabetes	0.323958	0.089614	0.048830	0.011304	0.031429	0.268686	1.000000	0.067949	0.043484	0.029102	...	0.141777	0.0
fever	0.070876	0.310393	0.004336	0.042883	0.364572	0.056307	0.067949	1.000000	-0.009785	0.191502	...	0.361498	0.2
asthma	-0.002322	0.366153	-0.003138	-0.017802	-0.025726	0.035180	0.043484	-0.009785	1.000000	-0.016493	...	0.033486	-0.0
um	0.075909	0.247199	-0.091514	0.098737	0.123828	0.010695	0.029102	0.191502	-0.016493	1.000000	...	0.159796	0.0
cough	0.101326	0.326308	-0.023673	0.098252	0.396169	0.038480	0.037815	0.448407	-0.020025	0.201046	...	0.273108	0.2
ihd	0.131344	0.058897	0.014676	0.028663	0.031020	0.119621	0.188007	0.062860	0.056894	-0.000380	...	0.169181	0.0
bd	0.141344	0.260441	0.016974	0.033719	0.156791	0.094485	0.141777	0.361498	0.033486	0.159796	...	1.000000	0.1
st	-0.011312	0.194681	-0.001744	0.054822	0.228823	-0.037047	0.011086	0.247791	-0.003635	0.089384	...	0.160314	1.0
ftg	0.093844	0.271610	-0.041149	0.064611	0.271435	0.078456	0.027301	0.391139	-0.019699	0.165408	...	0.294222	0.2
mp	0.035811	0.234686	-0.057022	0.027134	0.259210	0.018216	0.037035	0.332047	-0.014751	0.168709	...	0.237159	0.2
loss_of_smell	-0.001018	0.209661	-0.024261	0.080044	0.205421	0.002149	0.033954	0.283174	-0.019648	0.059721	...	0.176356	0.1
cld	0.072513	0.100924	0.057541	-0.011151	0.012382	0.015486	0.053077	0.015098	0.071548	-0.020769	...	0.073551	-0.0
diarrhoea	0.019490	0.055659	0.024279	-0.011831	0.096951	-0.018358	0.048140	0.165658	0.065938	0.057500	...	0.108555	0.0
pneumonia	0.071803	0.120242	-0.005415	0.031301	0.034599	0.039331	0.074882	0.155247	0.098958	0.077398	...	0.178818	0.0
others_resp	0.039826	0.027341	0.013831	-0.003935	-0.006540	-0.006106	-0.005732	0.085733	-0.004599	-0.007329	...	0.117326	-0.0
others_preexist	0.105169	0.100924	-0.052545	-0.011151	0.074208	0.146645	0.122396	0.047647	0.029257	0.091680	...	0.030400	0.0

22 rows × 22 columns

In [20]:
```
test=df["covid_status"]                                     #train and test data
train=df[["fever","asthma","cough","ftg","bd","cold","um","mp","loss_of_smell","pneumonia"]]
```

In [21]:
```
from sklearn.model_selection import train_test_split                  #train and test data split
X_train, X_test, y_train, y_test = train_test_split(train, test, test_size = .3, random_state = 0)
```

In [22]:
```
import tensorflow                                 #importing libraries
from tensorflow import keras
from keras.models import Sequential
from keras.layers import Dense
```

In [23]:
```
df
```

Out[23]:

	id	a	covid_status	ep	g	l_c	l_l	l_s	rU	smoker	...	diarrhoea	pneumonia	c
0	iV3Db6t1T8b7c5HQY2TwxIhjbzD3	28	0	y	1	India	Anantapur	Andhra Pradesh	0	0	...	0	0	
1	AxuYWBN0jFVLINCBqIW5aZmGCdu1	25	0	y	1	India	BENGALURU URBAN	Karnataka	0	1	...	0	0	
2	C5eIsssb9GSkaAgIfsHMHeR6fSh1	28	0	y	0	United States	Pittsburgh	Pennsylvania	0	0	...	0	0	
3	YjbEAECMBIaZKyfqOvWy5DDImUb2	26	0	y	1	India	Bangalore	Karnataka	0	0	...	0	0	
4	aGOvk4ji0cVqIzCs1jHnzlw2UEy2	32	0	y	1	India	Nalanda	Bihar	0	0	...	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1890	aH6o5lYN6vh40RCFhfyBYjkQVH13	33	0	y	1	India	24pgs	West Bengal	0	1	...	0	0	
1891	hRMSJF9xD6h9CmQJzB3phXKNnTI2	42	5	y	1	India	Bangalore	Karnataka	0	0	...	0	0	
1892	ZNTuppbDaCNybeDYGmyjJeYr93s1	27	0	y	0	India	False	Karnataka	0	0	...	0	0	
1893	J7R2UkId7jU0aOSw5amaS8zJVIm1	37	0	y	1	India	False	Karnataka	0	0	...	0	0	
1894	8MdsvTBMWAUInavE1pdmFQAZPF13	38	5	y	0	India	Bengaluru	Karnataka	0	0	...	0	0	

1895 rows × 35 columns

In [24]:
```
from sklearn.preprocessing import StandardScaler                       #standardizing data
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

In [44]:
```
from sklearn.neural_network import MLPClassifier                       #training MLP classifier
clf = MLPClassifier(solver='sgd', alpha=1e-5,hidden_layer_sizes=(100, 100,10), random_state=1)
clf.fit(X_train, y_train)
```

```
C:\Users\sahit\anaconda3\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:582: ConvergenceWarning:
Stochastic Optimizer: Maximum iterations (200) reached and the optimization hasn't converged yet.
  warnings.warn(
```

Out[44]:
```
MLPClassifier(alpha=1e-05, hidden_layer_sizes=(100, 100, 10), random_state=1,
              solver='sgd')
```

In [45]:
```
a=clf.predict(X_test)
```

In [46]:
```
from sklearn.metrics import accuracy_score,confusion_matrix              #accuracy of MLP classifier
print(accuracy_score(y_test,a))
```

```
0.7557117750439367
```

In [25]:
```
from keras.models import Sequential                                  #importing libraries for cnn and lstm
import keras
from keras.models import Sequential
from keras.layers import Dense, Flatten, Conv1D, Dropout
from keras import utils
from keras.layers.convolutional import MaxPooling1D
from keras.utils import np_utils

from keras.utils.np_utils import to_categorical
```

In [26]:
```
X_train.shape
```

Out[26]:
```
(1326, 10)
```

In [27]:
```
X_test.shape
```

Out[27]:
```
(569, 10)
```

In [28]:
```
_train=y_train-1                                                       #reshapes of X_train and y_train, y_test and x_
_test=y_test-1
_train=keras.utils.np_utils.to_categorical(y_train,7)
_test=keras.utils.np_utils.to_categorical(y_test,7)
_train=X_train.reshape(1326,10,1)
_test=X_test.reshape(569,10,1)
```

In [29]:

```
model = Sequential()                                             #cnn model
model.add(Conv1D(64,3,input_shape=(10,1), activation='relu'))
#model.add(Dense(128, activation='relu'))
model.add(Dropout(0.1))
model.add(MaxPooling1D(pool_size=1))
model.add(Flatten())
model.add(Dense(64, activation='relu'))
model.add(Dense(7, activation='softmax'))

model.compile(loss=keras.losses.categorical_crossentropy,
              optimizer=keras.optimizers.Adam(),
              metrics=['accuracy'])
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv1d (Conv1D)              (None, 8, 64)             256       
_________________________________________________________________
dropout (Dropout)            (None, 8, 64)             0         
_________________________________________________________________
max_pooling1d (MaxPooling1D) (None, 8, 64)             0         
_________________________________________________________________
flatten (Flatten)            (None, 512)               0         
_________________________________________________________________
dense (Dense)                (None, 64)                32832     
_________________________________________________________________
dense_1 (Dense)              (None, 7)                 455       
=================================================================
Total params: 33,543
Trainable params: 33,543
Non-trainable params: 0
_________________________________________________________________
```

In [30]:

```
batch_size = 240                                           #training and evaluating cnn model
epochs = 200
history= model.fit(X_train, y_train,
          batch_size=batch_size,
          epochs=epochs,
          verbose=1,validation_split=0.3,shuffle=False)
[loss,score]=model.evaluate(X_test,y_test,verbose=0)
```

```
4/4 [==============================] - 0s 12ms/step - loss: 0.7094 - accuracy: 0.7993 - val_loss: 0.8318 - val_acc
uracy: 0.7638
Epoch 35/200
4/4 [==============================] - 0s 12ms/step - loss: 0.7093 - accuracy: 0.8017 - val_loss: 0.8335 - val_acc
uracy: 0.7638
Epoch 36/200
4/4 [==============================] - 0s 13ms/step - loss: 0.7068 - accuracy: 0.7987 - val_loss: 0.8338 - val_acc
uracy: 0.7613
Epoch 37/200
4/4 [==============================] - 0s 12ms/step - loss: 0.7096 - accuracy: 0.8004 - val_loss: 0.8326 - val_acc
uracy: 0.7663
Epoch 38/200
4/4 [==============================] - 0s 12ms/step - loss: 0.7058 - accuracy: 0.8027 - val_loss: 0.8339 - val_acc
uracy: 0.7663
Epoch 39/200
4/4 [==============================] - 0s 12ms/step - loss: 0.7019 - accuracy: 0.8036 - val_loss: 0.8383 - val_acc
uracy: 0.7613
Epoch 40/200
4/4 [==============================] - 0s 12ms/step - loss: 0.7029 - accuracy: 0.8050 - val_loss: 0.8417 - val_acc
uracy: 0.7588
```

In [32]:

```
print(score*100)                             #accuracy of cnn model
```

```
75.92267394065857
```

In [33]:

```
from keras.layers import LSTM                  #defining LSTM, training and evaluationg lstm model
model1 = Sequential()
model1.add(LSTM(64, input_shape=(10,1)))
model1.add(Dropout(0.1))
model1.add(Dense(7))
model1.compile(loss=keras.losses.categorical_crossentropy, optimizer='adam',metrics=['accuracy'])
lstm=model1.fit(X_train, y_train, epochs=200, batch_size=240, verbose=1,validation_split=0.3,shuffle=False)
l1,acc1=model1.evaluate(X_test,y_test,verbose=0)
```

```
Epoch 1/200
4/4 [==============================] - 5s 185ms/step - loss: 4.3221 - accuracy: 0.1304 - val_loss: 10.7632 - val_a
ccuracy: 0.6256
Epoch 2/200
4/4 [==============================] - 0s 19ms/step - loss: 10.9397 - accuracy: 0.5949 - val_loss: 10.8766 - val_a
ccuracy: 0.6256
Epoch 3/200
4/4 [==============================] - 0s 19ms/step - loss: 11.0258 - accuracy: 0.6259 - val_loss: 10.8325 - val_a
ccuracy: 0.6256
Epoch 4/200
4/4 [==============================] - 0s 18ms/step - loss: 11.0166 - accuracy: 0.6271 - val_loss: 10.8259 - val_a
ccuracy: 0.6256
Epoch 5/200
4/4 [==============================] - 0s 22ms/step - loss: 10.9539 - accuracy: 0.6267 - val_loss: 10.6906 - val_a
ccuracy: 0.6256
Epoch 6/200
4/4 [==============================] - 0s 21ms/step - loss: 10.9548 - accuracy: 0.6209 - val_loss: 10.6733 - val_a
ccuracy: 0.6256
Epoch 7/200
```

In [34]:

```
print(acc1*100)               # accuracy of lstm model
```

```
73.6379623413086
```

In [ ]: