```
import tensorflow as tf
```

```
a=tf.constant([[1,2],[3,4]])
b=tf.constant([[1,1],[3,4]])
```

```
print(a)
```

```
tf.Tensor(
[[1 2]
 [3 4]], shape=(2, 2), dtype=int32)
```

```
c=tf.add(a,b)
c
```

```
<tf.Tensor: shape=(2, 2), dtype=int32, numpy=
array([[2, 3],
       [6, 8]], dtype=int32)>
```

```
d=tf.subtract(a,b)
d
```

```
<tf.Tensor: shape=(2, 2), dtype=int32, numpy=
array([[0, 1],
       [0, 0]], dtype=int32)>
```

```
e=tf.matmul(a,b)
e
```

```
<tf.Tensor: shape=(2, 2), dtype=int32, numpy=
array([[ 7,  9],
       [15, 19]], dtype=int32)>
```

```
# element wise multiplication
f=tf.multiply(a,b)
```

```
f
```

```
<tf.Tensor: shape=(2, 2), dtype=int32, numpy=
array([[ 1,  2],
       [ 9, 16]], dtype=int32)>
```

```
g=tf.divide(a,b)
g
```

```
<tf.Tensor: shape=(2, 2), dtype=float64, numpy=
array([[1., 2.],
       [1., 1.]])>
```

```
import pandas as pd
```

```
df=pd.read_csv("/content/IRIS.csv")
df.head()
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

```
df['species'].unique()
```

```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

```
x=df.drop(columns=['species'])
x
```

	sepal_length	sepal_width	petal_length	petal_width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

```
y=df['species']
y
```

```
0         Iris-setosa
1         Iris-setosa
2         Iris-setosa
3         Iris-setosa
4         Iris-setosa
             ...
145    Iris-virginica
146    Iris-virginica
147    Iris-virginica
148    Iris-virginica
149    Iris-virginica
Name: species, Length: 150, dtype: object
```

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y)
```

```
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(112, 4)
(38, 4)
(112,)
(38,)
```

```
import numpy as np
x_train=np.array(x_train)
x_test=np.array(x_test)
y_train=np.array(y_train)
y_test=np.array(y_test)
```

```
def train(train,test):
  l1_distance=tf.abs(tf.add(train,tf.negative(test)))
  distance=tf.reduce_sum(l1_distance,axis=1)
  return np.array(tf.argsort(distance))
```

```
# single test
index=train(x_train,x_test[0])
print(index)
```

```
[  0 107  98  71  75  73   7  65  49  84  38   2   5  29  59  45  46  20
  28  30 109  39  78  91  92  61  87  62  33  15  17 111   9  54  27  42
  68  67  23  12  96  40  18  36  76  70 110  77  88 104 108  66  13  80
  14  25 100  83   6  57   3  41  89  11  43  93 103  34  10  21  22  60
  85  24  51  52  32  31  48 105   1   4  69  79  16  53  86 106  56  74
  55  99  35   8  95  58  63  81 102  37  72  47  19  26  44  50  82  64
  94 101  97  90]
```

```
k=3
```

```
k=3
```

```
classes=['Iris-setosa', 'Iris-versicolor', 'Iris-virginica']
```

```
correct_count=0
for i in range(len(x_test)):
  indd=[0,0,0]
  labels=[]
  index=train(x_train,x_test[i,:])
  for j in range(k):
    labels.append(y_train[index[j]])
  for l in labels:
    if(l=='Iris-setosa'):
      indd[0]=indd[0]+1
    elif(l=='Iris-versicolor'):
      indd[1]=indd[1]+1
    else:
      indd[2]=indd[2]+1
  print("epochs ",i,'predicted ',classes[np.argmax(y_train[indd])],"actual ",y_test[i])
  if(classes[np.argmax(y_train[indd])]==y_test[i]):
    correct_count+=1
print("correctly predicted: ",correct)
```

```
     epochs  0 predicted  Iris-versicolor actual  Iris-versicolor
     epochs  1 predicted  Iris-setosa actual  Iris-virginica
     epochs  2 predicted  Iris-setosa actual  Iris-setosa
     epochs  3 predicted  Iris-versicolor actual  Iris-versicolor
     epochs  4 predicted  Iris-versicolor actual  Iris-versicolor
     epochs  5 predicted  Iris-versicolor actual  Iris-virginica
     epochs  6 predicted  Iris-virginica actual  Iris-virginica
     epochs  7 predicted  Iris-setosa actual  Iris-setosa
     epochs  8 predicted  Iris-virginica actual  Iris-virginica
     epochs  9 predicted  Iris-setosa actual  Iris-virginica
     epochs  10 predicted  Iris-virginica actual  Iris-virginica
     epochs  11 predicted  Iris-setosa actual  Iris-setosa
     epochs  12 predicted  Iris-setosa actual  Iris-setosa
     epochs  13 predicted  Iris-setosa actual  Iris-virginica
     epochs  14 predicted  Iris-setosa actual  Iris-virginica
```

```
epochs  15 predicted  Iris-setosa actual  Iris-virginica
epochs  16 predicted  Iris-virginica actual  Iris-virginica
epochs  17 predicted  Iris-versicolor actual  Iris-versicolor
epochs  18 predicted  Iris-setosa actual  Iris-versicolor
epochs  19 predicted  Iris-virginica actual  Iris-virginica
epochs  20 predicted  Iris-setosa actual  Iris-setosa
epochs  21 predicted  Iris-versicolor actual  Iris-versicolor
epochs  22 predicted  Iris-versicolor actual  Iris-versicolor
epochs  23 predicted  Iris-versicolor actual  Iris-versicolor
epochs  24 predicted  Iris-virginica actual  Iris-virginica
epochs  25 predicted  Iris-setosa actual  Iris-setosa
epochs  26 predicted  Iris-setosa actual  Iris-setosa
epochs  27 predicted  Iris-setosa actual  Iris-setosa
epochs  28 predicted  Iris-virginica actual  Iris-virginica
epochs  29 predicted  Iris-virginica actual  Iris-virginica
epochs  30 predicted  Iris-setosa actual  Iris-virginica
epochs  31 predicted  Iris-virginica actual  Iris-virginica
epochs  32 predicted  Iris-setosa actual  Iris-setosa
epochs  33 predicted  Iris-virginica actual  Iris-virginica
epochs  34 predicted  Iris-setosa actual  Iris-setosa
epochs  35 predicted  Iris-setosa actual  Iris-setosa
epochs  36 predicted  Iris-versicolor actual  Iris-versicolor
epochs  37 predicted  Iris-virginica actual  Iris-virginica
correctly predicted:  0
```