KNN as Classifier [As Regressor in another program]

```python
#Assigning Feature and local variables

#First Feature
weather = ['Sunny','Sunny',
'Overcast','Rainy','Rainy','Rainy','Overcast','Sunny','Sunny','Rainy',
'Sunny','Overcast','Overcast','Rainy']

#Second Feature
temp =
['Hot','Hot','Hot','Mild','Cool','Cool','Cool','Mild','Cool','Mild','M
ild','Mild','Hot','Mild']

# Label or Target Variable
play
=['No','No','Yes','Yes','Yes','No','Yes','No','Yes','Yes','Yes','Yes',
'Yes','No' ]

# import Label Encoder
from sklearn import preprocessing

#creating Label Encoder
le = preprocessing.LabelEncoder()

#converting Strigns Labels to numbers
weather_encoded = le.fit_transform(weather)

weather_encoded

array([2, 2, 0, 1, 1, 1, 0, 2, 2, 1, 2, 0, 0, 1], dtype=int64)

#converting Strigns Labels to numbers
temp_encoded = le.fit_transform(temp)
label_col =  le.fit_transform(play)

temp_encoded

array([1, 1, 1, 2, 0, 0, 0, 2, 0, 2, 2, 2, 1, 2], dtype=int64)

#combining weather and temp into single list of tuples
features= list(zip(weather_encoded,temp_encoded))

features

[(2, 1),
 (2, 1),
 (0, 1),
```

```
 (1, 2),
 (1, 0),
 (1, 0),
 (0, 0),
 (2, 2),
 (2, 0),
 (1, 2),
 (2, 2),
 (0, 2),
 (0, 1),
 (1, 2)]
```

```python
from sklearn.neighbors import KNeighborsClassifier
```

```python
model = KNeighborsClassifier (n_neighbors=3)
```

```python
model.fit(features,label_col)
```

```
KNeighborsClassifier(n_neighbors=3)
```

```python
#predict output
predicted = model.predict([[1,0]])
```

```python
predicted
```

```
array([1], dtype=int64)
```

```python
print(predicted)
```

```
[1]
```