```
In [1]: import numpy as np
        import pandas as pd
```

```
In [2]: data = pd.read_csv(r"D:\ML_Course\Works_on_python\Multi_Linear_Regression\Fish.csv")
```

```
In [3]: data
```

Out[3]:

	Species	Weight	Length1	Length2	Length3	Height	Width
0	Bream	242.0	23.2	25.4	30.0	11.5200	4.0200
1	Bream	290.0	24.0	26.3	31.2	12.4800	4.3056
2	Bream	340.0	23.9	26.5	31.1	12.3778	4.6961
3	Bream	363.0	26.3	29.0	33.5	12.7300	4.4555
4	Bream	430.0	26.5	29.0	34.0	12.4440	5.1340
...	...	...	...	...	...	...	...
154	Smelt	12.2	11.5	12.2	13.4	2.0904	1.3936
155	Smelt	13.4	11.7	12.4	13.5	2.4300	1.2690
156	Smelt	12.2	12.1	13.0	13.8	2.2770	1.2558
157	Smelt	19.7	13.2	14.3	15.2	2.8728	2.0672
158	Smelt	19.9	13.8	15.0	16.2	2.9322	1.8792

159 rows × 7 columns

```
In [4]: data['Species'].unique()
```

```
Out[4]: array(['Bream', 'Roach', 'Whitefish', 'Parkki', 'Perch', 'Pike', 'Smelt'],
              dtype=object)
```

In [5]: `data.isnull().any()`

Out[5]:
```
Species    False
Weight     False
Length1    False
Length2    False
Length3    False
Height     False
Width      False
dtype: bool
```

In [6]:
```python
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
data['Species'] = le.fit_transform(data['Species'])
```

In [7]: `data`

Out[7]:

	Species	Weight	Length1	Length2	Length3	Height	Width
0	0	242.0	23.2	25.4	30.0	11.5200	4.0200
1	0	290.0	24.0	26.3	31.2	12.4800	4.3056
2	0	340.0	23.9	26.5	31.1	12.3778	4.6961
3	0	363.0	26.3	29.0	33.5	12.7300	4.4555
4	0	430.0	26.5	29.0	34.0	12.4440	5.1340
...	...	...	...	...	...	...	...
154	5	12.2	11.5	12.2	13.4	2.0904	1.3936
155	5	13.4	11.7	12.4	13.5	2.4300	1.2690
156	5	12.2	12.1	13.0	13.8	2.2770	1.2558
157	5	19.7	13.2	14.3	15.2	2.8728	2.0672
158	5	19.9	13.8	15.0	16.2	2.9322	1.8792

159 rows × 7 columns

```
In [8]: data['Species'].unique()
```

```
Out[8]: array([0, 4, 6, 1, 2, 3, 5])
```

```
In [9]: x=data.iloc[:,1:7].values
        y=data.iloc[:,0:1].values
```

```
In [10]: x
```

```
Out[10]: array([[2.42000e+02, 2.32000e+01, 2.54000e+01, 3.00000e+01, 1.15200e+01,
                  4.02000e+00],
                 [2.90000e+02, 2.40000e+01, 2.63000e+01, 3.12000e+01, 1.24800e+01,
                  4.30560e+00],
                 [3.40000e+02, 2.39000e+01, 2.65000e+01, 3.11000e+01, 1.23778e+01,
                  4.69610e+00],
                 [3.63000e+02, 2.63000e+01, 2.90000e+01, 3.35000e+01, 1.27300e+01,
                  4.45550e+00],
                 [4.30000e+02, 2.65000e+01, 2.90000e+01, 3.40000e+01, 1.24440e+01,
                  5.13400e+00],
                 [4.50000e+02, 2.68000e+01, 2.97000e+01, 3.47000e+01, 1.36024e+01,
                  4.92740e+00],
                 [5.00000e+02, 2.68000e+01, 2.97000e+01, 3.45000e+01, 1.41795e+01,
                  5.27850e+00],
                 [3.90000e+02, 2.76000e+01, 3.00000e+01, 3.50000e+01, 1.26700e+01,
                  4.69000e+00],
                 [4.50000e+02, 2.76000e+01, 3.00000e+01, 3.51000e+01, 1.40049e+01,
                  4.84380e+00],
                 [5.00000e+02, 2.85000e+01, 3.07000e+01, 3.62000e+01, 1.42266e+01,
                  4.85040.001]
```

```
In [11]: from sklearn.preprocessing import OneHotEncoder
         one = OneHotEncoder()
         z=one.fit_transform(y[:,0:1]).toarray()
         y=np.delete(y,0,axis=1)
         y=np.concatenate((z,y),axis=1)
```

In [12]: `z`

Out[12]:
```
array([[1., 0., 0., ..., 0., 0., 0.],
       [1., 0., 0., ..., 0., 0., 0.],
       [1., 0., 0., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 0., 1., 0.],
       [0., 0., 0., ..., 0., 1., 0.],
       [0., 0., 0., ..., 0., 1., 0.]])
```

In [13]:
```python
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y, test_size = 0.2, random_state = 0)
```

In [14]: `y_train.shape`

Out[14]: (127, 7)

In [15]:
```python
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.fit_transform(x_test)
```

In [16]:
```python
import keras
from keras.models import Sequential
from keras.layers import Dense
```

```
Using TensorFlow backend.
C:\Users\anikp\Anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:516: FutureWarning: Passing (type, 1) or '1ty
pe' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint8 = np.dtype([("qint8", np.int8, 1)])
C:\Users\anikp\Anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:517: FutureWarning: Passing (type, 1) or '1ty
pe' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_quint8 = np.dtype([("quint8", np.uint8, 1)])
C:\Users\anikp\Anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:518: FutureWarning: Passing (type, 1) or '1ty
pe' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint16 = np.dtype([("qint16", np.int16, 1)])
C:\Users\anikp\Anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:519: FutureWarning: Passing (type, 1) or '1ty
pe' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_quint16 = np.dtype([("quint16", np.uint16, 1)])
C:\Users\anikp\Anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:520: FutureWarning: Passing (type, 1) or '1ty
pe' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint32 = np.dtype([("qint32", np.int32, 1)])
C:\Users\anikp\Anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:525: FutureWarning: Passing (type, 1) or '1ty
pe' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  np_resource = np.dtype([("resource", np.ubyte, 1)])
C:\Users\anikp\Anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:541: FutureWarning: Passing (type, 1)
or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)typ
e'.
  _np_qint8 = np.dtype([("qint8", np.int8, 1)])
C:\Users\anikp\Anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:542: FutureWarning: Passing (type, 1)
or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)typ
e'.
  _np_quint8 = np.dtype([("quint8", np.uint8, 1)])
C:\Users\anikp\Anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:543: FutureWarning: Passing (type, 1)
or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)typ
e'.
  _np_qint16 = np.dtype([("qint16", np.int16, 1)])
C:\Users\anikp\Anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:544: FutureWarning: Passing (type, 1)
or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)typ
e'.
  _np_quint16 = np.dtype([("quint16", np.uint16, 1)])
C:\Users\anikp\Anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:545: FutureWarning: Passing (type, 1)
or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)typ
e'.
  _np_qint32 = np.dtype([("qint32", np.int32, 1)])
```

```
C:\Users\anikp\Anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:550: FutureWarning: Passing (type, 1)
or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)typ
e'.
    np_resource = np.dtype([("resource", np.ubyte, 1)])
```

In [17]: `model = Sequential()`

```
WARNING:tensorflow:From C:\Users\anikp\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:74: The name tf.get_def
ault_graph is deprecated. Please use tf.compat.v1.get_default_graph instead.
```

In [18]: `x_train.shape`

Out[18]: `(127, 6)`

In [19]: `model.add(Dense(units = 6,init = 'random_uniform',activation = 'relu'))`

```
C:\Users\anikp\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: UserWarning: Update your `Dense` call to the Keras 2 API:
`Dense(units=6, activation="relu", kernel_initializer="random_uniform")`
    """Entry point for launching an IPython kernel.
```

In [20]: `model.add(Dense(units =9,init = 'random_uniform',activation = 'relu'))`

```
C:\Users\anikp\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: UserWarning: Update your `Dense` call to the Keras 2 API:
`Dense(units=9, activation="relu", kernel_initializer="random_uniform")`
    """Entry point for launching an IPython kernel.
```

In [21]: `model.add(Dense(units = 7 ,init = 'random_uniform',activation = 'softmax'))`

```
C:\Users\anikp\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: UserWarning: Update your `Dense` call to the Keras 2 API:
`Dense(units=7, activation="softmax", kernel_initializer="random_uniform")`
    """Entry point for launching an IPython kernel.
```

In [22]: `model.compile (optimizer = 'adam',loss = 'categorical_crossentropy',metrics = ['accuracy'])`

```
WARNING:tensorflow:From C:\Users\anikp\Anaconda3\lib\site-packages\keras\optimizers.py:790: The name tf.train.Optimizer is dep
recated. Please use tf.compat.v1.train.Optimizer instead.
```

In [23]:
```python
model.fit(x_train,y_train , batch_size = 10,epochs = 300)
```

WARNING:tensorflow:From C:\Users\anikp\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:517: The name tf.plac
eholder is deprecated. Please use tf.compat.v1.placeholder instead.

WARNING:tensorflow:From C:\Users\anikp\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:4138: The name tf.ran
dom_uniform is deprecated. Please use tf.random.uniform instead.

WARNING:tensorflow:From C:\Users\anikp\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:3295: The name tf.log
is deprecated. Please use tf.math.log instead.

WARNING:tensorflow:From C:\Users\anikp\Anaconda3\lib\site-packages\tensorflow\python\ops\math_grad.py:1250: add_dispatch_sup
port.<locals>.wrapper (from tensorflow.python.ops.array_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
WARNING:tensorflow:From C:\Users\anikp\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:986: The name tf.assi
gn_add is deprecated. Please use tf.compat.v1.assign_add instead.

Epoch 1/300
127/127 [==============================] - 0s 2ms/step - loss: 1.9436 - acc: 0.3307
Epoch 2/300

In [24]:
```python
y_pred = model.predict_classes(x_test)
```

In [25]:
```python
y_pred
```

Out[25]:
```
array([0, 4, 2, 4, 2, 2, 6, 3, 0, 2, 2, 3, 2, 2, 4, 2, 2, 3, 2, 0, 4, 6,
       2, 3, 4, 0, 0, 4, 2, 5, 0, 2], dtype=int64)
```

In [26]:
```python
data.head(1)
```

Out[26]:

	Species	Weight	Length1	Length2	Length3	Height	Width
0	0	242.0	23.2	25.4	30.0	11.52	4.02

In [27]:
```python
y_p = model.predict_classes(sc.transform([[1100,40.1,43,45.5,12.5125,7.4165]]))
```

In [28]:
```python
y_p[0]
```

Out[28]: 2

In [29]:
```python
index = ['Bream', 'Parkki','Roach', 'Whitefish',  'Perch', 'Pike', 'Smelt']
```

In [37]:
```python
class1 = index[y_p[0]]
```

In [38]:
```python
class1
```

Out[38]: 'Roach'

In [ ]: