

iris flower-categorical

In [1]:

```
import numpy as np
import pandas as pd
```

In [2]:

```
dataset = pd.read_csv(r"C:\Users\kotha\Downloads\iris flower classification.csv")
```

In [3]:

```
dataset
```

Out[3]:

	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
...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

In [4]:

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  -
0   sepal_length    150 non-null   float64
1   sepal_width     150 non-null   float64
2   petal_length    150 non-null   float64
3   petal_width     150 non-null   float64
4   species         150 non-null   object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

In [5]:

```
dataset["species"].unique()
```

Out[5]:

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

In [6]:

```
dataset.isnull().sum()
```

Out[6]:

```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

In [7]:

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
dataset["species"] = le.fit_transform(dataset["species"])
```

In [8]:

```
dataset.head(1)
```

Out[8]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	0

In [9]:

```
x = dataset.iloc[:,0:4].values
y = dataset.iloc[:,4:5].values
```

In [10]:

X

Out[10]:

```
array([[5.1, 3.5, 1.4, 0.2],
       [4.9, 3. , 1.4, 0.2],
       [4.7, 3.2, 1.3, 0.2],
       [4.6, 3.1, 1.5, 0.2],
       [5. , 3.6, 1.4, 0.2],
       [5.4, 3.9, 1.7, 0.4],
       [4.6, 3.4, 1.4, 0.3],
       [5. , 3.4, 1.5, 0.2],
       [4.4, 2.9, 1.4, 0.2],
       [4.9, 3.1, 1.5, 0.1],
       [5.4, 3.7, 1.5, 0.2],
       [4.8, 3.4, 1.6, 0.2],
       [4.8, 3. , 1.4, 0.1],
       [4.3, 3. , 1.1, 0.1],
       [5.8, 4. , 1.2, 0.2],
       [5.7, 4.4, 1.5, 0.4],
       [5.4, 3.9, 1.3, 0.4],
       [5.1, 3.5, 1.4, 0.3],
```

In [11]:

y

Out[11]:

[illegible]

In [12]:

```
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 [13]:

```
y.shape
```

Out[13]:

```
(150, 3)
```

In [14]:

```
x.shape
```

Out[14]:

```
(150, 4)
```

In [15]:

```
y
```

Out[15]:

```
array([[1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [1., 0., 0.]])
```

In [16]:

```
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 [17]:

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
```

In [18]:

```
from keras.models import Sequential
from keras.layers import Dense
```

Using TensorFlow backend.

C:\Users\kotha\anaconda3\lib\site-packages\tensorflow\python\framework\dtype
s.py:516: FutureWarning: Passing (type, 1) or '1type' as a synonym of type i
s 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\kotha\anaconda3\lib\site-packages\tensorflow\python\framework\dtype
s.py:517: FutureWarning: Passing (type, 1) or '1type' as a synonym of type i
s 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\kotha\anaconda3\lib\site-packages\tensorflow\python\framework\dtype
s.py:518: FutureWarning: Passing (type, 1) or '1type' as a synonym of type i
s 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\kotha\anaconda3\lib\site-packages\tensorflow\python\framework\dtype
s.py:519: FutureWarning: Passing (type, 1) or '1type' as a synonym of type i
s 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\kotha\anaconda3\lib\site-packages\tensorflow\python\framework\dtype
s.py:520: FutureWarning: Passing (type, 1) or '1type' as a synonym of type i
s 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\kotha\anaconda3\lib\site-packages\tensorflow\python\framework\dtype
s.py:525: FutureWarning: Passing (type, 1) or '1type' as a synonym of type i
s 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\kotha\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stu
b\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,)type'.

```
_np_qint8 = np.dtype [("qint8", np.int8, 1)]
```

C:\Users\kotha\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stu
b\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,)type'.

```
_np_quint8 = np.dtype [("quint8", np.uint8, 1)]
```

C:\Users\kotha\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stu
b\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,)type'.

```
_np_qint16 = np.dtype [("qint16", np.int16, 1)]
```

C:\Users\kotha\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stu
b\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,)type'.

```
_np_quint16 = np.dtype [("quint16", np.uint16, 1)]
```

C:\Users\kotha\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stu
b\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,)type'.

```
_np_qint32 = np.dtype [("qint32", np.int32, 1)]
```

```
C:\Users\kotha\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,)type'.
  np_resource = np.dtype [("resource", np.ubyte, 1)])
```

In [19]:

```
model = Sequential()
```

```
WARNING:tensorflow:From C:\Users\kotha\anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:74: The name tf.get_default_graph is deprecated. Please use tf.compat.v1.get_default_graph instead.
```

In [20]:

```
x.shape
```

Out[20]:

```
(150, 4)
```

In [21]:

```
model.add(Dense(units = 4 ,activation = "relu", init = "random_uniform"))
```

```
C:\Users\kotha\anaconda3\lib\site-packages\ipykernel_launcher.py:1: UserWarning: Update your `Dense` call to the Keras 2 API: `Dense(units=4, activation="relu", kernel_initializer="random_uniform")`
  """Entry point for launching an IPython kernel.
```

In [22]:

```
model.add(Dense(units = 8 ,activation = "relu", init = "random_uniform"))
```

```
C:\Users\kotha\anaconda3\lib\site-packages\ipykernel_launcher.py:1: UserWarning: Update your `Dense` call to the Keras 2 API: `Dense(units=8, activation="relu", kernel_initializer="random_uniform")`
  """Entry point for launching an IPython kernel.
```

In [23]:

```
model.add(Dense(units = 3 ,activation = "softmax", init = "random_uniform"))
```

```
C:\Users\kotha\anaconda3\lib\site-packages\ipykernel_launcher.py:1: UserWarning: Update your `Dense` call to the Keras 2 API: `Dense(units=3, activation="softmax", kernel_initializer="random_uniform")`
  """Entry point for launching an IPython kernel.
```

In [24]:

```
model.compile(optimizer = "adam",loss = "categorical_crossentropy",metrics = ["accuracy"])
```

```
WARNING:tensorflow:From C:\Users\kotha\anaconda3\lib\site-packages\keras\optimizers.py:790: The name tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.
```

In [25]:

```
model.fit(x_train,y_train, epochs = 500,batch_size = 32)
```

WARNING:tensorflow:From C:\Users\kotha\anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:517: The name tf.placeholder is deprecated. Please use tf.compat.v1.placeholder instead.

WARNING:tensorflow:From C:\Users\kotha\anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:4138: The name tf.random_uniform is deprecated. Please use tf.random.uniform instead.

WARNING:tensorflow:From C:\Users\kotha\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\kotha\anaconda3\lib\site-packages\tensorflow\python\ops\math_grad.py:1250: add_dispatch_support.<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\kotha\anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:3295: The name tf.log is deprecated. Please use tf.math.log instead.

In [26]:

```
y_pred = model.predict(x_test)
```

In [27]:

y_pred

Out[27]:

```
array([[5.6813789e-09, 2.5042382e-03, 9.9749571e-01],
       [7.5164973e-03, 9.7331953e-01, 1.9163951e-02],
       [9.9999428e-01, 5.7210200e-06, 5.2073108e-13],
       [2.1411972e-07, 1.3739519e-02, 9.8626024e-01],
       [9.9971122e-01, 2.8876498e-04, 6.3508382e-10],
       [1.4023108e-10, 3.2671474e-04, 9.9967325e-01],
       [9.9980730e-01, 1.9272430e-04, 3.0720632e-10],
       [8.8516047e-04, 9.7380573e-01, 2.5309136e-02],
       [9.8769856e-04, 8.7056470e-01, 1.2844771e-01],
       [3.5887659e-03, 9.8973316e-01, 6.6780834e-03],
       [6.8261470e-05, 3.1117794e-01, 6.8875378e-01],
       [9.9407404e-04, 9.9236965e-01, 6.6362391e-03],
       [2.6433303e-03, 9.8619360e-01, 1.1163128e-02],
       [6.7154784e-04, 8.1548572e-01, 1.8384278e-01],
       [1.0589699e-03, 9.6367413e-01, 3.5266899e-02],
       [9.9907303e-01, 9.2691806e-04, 5.1607190e-09],
       [8.1964763e-04, 9.4810563e-01, 5.1074762e-02],
       [3.2754710e-03, 9.8387873e-01, 1.2845752e-02],
       [9.9753076e-01, 2.4693217e-03, 3.1049893e-08],
       [9.9995744e-01, 4.2525793e-05, 1.9859997e-11],
       [9.1340161e-07, 3.6042683e-02, 9.6395642e-01],
       [8.1679720e-04, 9.8582119e-01, 1.3362050e-02],
       [9.9961013e-01, 3.8994351e-04, 1.1332201e-09],
       [9.9759299e-01, 2.4069943e-03, 3.0169229e-08],
       [1.6448301e-05, 1.4544788e-01, 8.5453570e-01],
       [9.9995053e-01, 4.9493534e-05, 2.6564634e-11],
       [9.9988854e-01, 1.1139530e-04, 1.1910246e-10],
       [2.6192681e-03, 9.9045396e-01, 6.9267717e-03],
       [6.8092109e-03, 9.8544425e-01, 7.7465181e-03],
       [9.9940026e-01, 5.9977197e-04, 2.4472580e-09]], dtype=float32)
```


In [28]:

```
y_test
```

Out[28]:

```
array([[0., 0., 1.],
       [0., 1., 0.],
       [1., 0., 0.],
       [0., 0., 1.],
       [1., 0., 0.],
       [0., 0., 1.],
       [1., 0., 0.],
       [0., 1., 0.],
       [0., 1., 0.],
       [0., 1., 0.],
       [0., 0., 1.],
       [0., 1., 0.],
       [0., 1., 0.],
       [0., 1., 0.],
       [0., 1., 0.],
       [1., 0., 0.],
       [0., 1., 0.],
       [0., 1., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [0., 0., 1.],
       [0., 1., 0.],
       [1., 0., 0.],
       [1., 0., 0.],
       [0., 0., 1.],
       [1., 0., 0.],
       [1., 0., 0.],
       [0., 1., 0.],
       [0., 1., 0.],
       [1., 0., 0.]])
```

In [29]:

```
yp = model.predict(sc.transform(np.array([[4.4, 2.9, 1.4, 0.2]])))
```

In [30]:

```
yp
```

Out[30]:

```
array([[9.9759299e-01, 2.4069953e-03, 3.0169286e-08]], dtype=float32)
```

In [31]:

```
ypc = model.predict_classes(sc.transform(np.array([[4.4, 2.9, 1.4, 0.2]])))
```

In [32]:

```
ypc
```

Out[32]:

```
array([0], dtype=int64)
```

In [33]:

```
p = ypc[0]
```

In [34]:

```
p
```

Out[34]:

```
0
```

In [35]:

```
index = ["Iris-setosa", "Iris-versicolor", "Iris-virginica"]
```

In [36]:

```
print(index[p])
```

```
Iris-setosa
```