In [124]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [125]:

```
df=pd.read_csv("D:\\Dataset\\IRIS.csv")
df
```

Out[125]:

	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 [126]:

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
df['species']=le.fit_transform(df['species'])
df
```

Out[126]:

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

150 rows × 5 columns

In [127]:
```
x=df.drop('species',axis=1)
x
```

Out[127]:

	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

150 rows × 4 columns

In [128]:
```
y1=df['species']
y1
```

Out[128]:
```
0      0
1      0
2      0
3      0
4      0
      ..
145    2
146    2
147    2
148    2
149    2
Name: species, Length: 150, dtype: int32
```

In [129]:
```
y2=np.array(y1)
y2
```

Out[129]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

In [130]:
```
y3=y2.reshape(-1,1)
y3
```

Out[130]:
```
array([[0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
```

In [131]:
```
from sklearn.preprocessing import OneHotEncoder
encoder=OneHotEncoder(sparse=False)
y=encoder.fit_transform(y3)
y
```

```
C:\Users\Hp\anaconda3\Lib\site-packages\sklearn\preprocessing\_encoder
s.py:868: FutureWarning: `sparse` was renamed to `sparse_output` in ver
sion 1.2 and will be removed in 1.4. `sparse_output` is ignored unless
you leave `sparse` to its default value.
  warnings.warn(
```

Out[131]:
```
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.],
```

In [200]:

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

```
(120, 4)
(120, 3)
(30, 4)
(30, 3)
```

In [201]:

```
import tensorflow as tf
from keras.models import Sequential
from keras.layers import Dense,Dropout
from keras.optimizers import Adam
```

In [202]:

```
model=Sequential()
model.add(Dense(10,activation='relu',input_shape=(4,)))
model.add(Dense(10,activation='relu'))
model.add(Dense(3,activation='softmax'))
```

In [203]:

```
model.compile(optimizer='Adam',loss='categorical_crossentropy',metrics=['ac
```

In [204]:

```
tf_callbacks=tf.keras.callbacks.TensorBoard(log_dir='logs/fit',histogram_fr
```

In [205]:

```
history=model.fit(x_train,y_train,validation_data=(x_test,y_test),epochs=50
Epoch 1/50
4/4 [==============================] - 1s 95ms/step - loss: 1.3466 - accur
acy: 0.3500 - val_loss: 1.5129 - val_accuracy: 0.2667
Epoch 2/50
4/4 [==============================] - 0s 33ms/step - loss: 1.2898 - accur
acy: 0.3500 - val_loss: 1.4444 - val_accuracy: 0.2667
Epoch 3/50
4/4 [==============================] - 0s 38ms/step - loss: 1.2386 - accur
acy: 0.3500 - val_loss: 1.3789 - val_accuracy: 0.2667
Epoch 4/50
4/4 [==============================] - 0s 47ms/step - loss: 1.1873 - accur
acy: 0.3500 - val_loss: 1.3183 - val_accuracy: 0.2667
Epoch 5/50
4/4 [==============================] - 0s 32ms/step - loss: 1.1458 - accur
acy: 0.3500 - val_loss: 1.2612 - val_accuracy: 0.2667
Epoch 6/50
4/4 [==============================] - 0s 37ms/step - loss: 1.1026 - accur
acy: 0.3500 - val_loss: 1.2104 - val_accuracy: 0.2667
Epoch 7/50
4/4 [==============================] - 0s 42ms/step - loss: 1.0683 - accur
acy: 0.3667 - val_loss: 1.1628 - val_accuracy: 0.2667
Epoch 8/50
4/4 [==============================] - 0s 44ms/step - loss: 1.0353 - accur
acy: 0.4417 - val_loss: 1.1199 - val_accuracy: 0.4000
Epoch 9/50
4/4 [==============================] - 0s 45ms/step - loss: 1.0055 - accur
acy: 0.5667 - val_loss: 1.0814 - val_accuracy: 0.5333
Epoch 10/50
4/4 [==============================] - 0s 37ms/step - loss: 0.9809 - accur
acy: 0.6833 - val_loss: 1.0501 - val_accuracy: 0.5667
Epoch 11/50
4/4 [==============================] - 0s 39ms/step - loss: 0.9603 - accur
acy: 0.6917 - val_loss: 1.0251 - val_accuracy: 0.5667
Epoch 12/50
4/4 [==============================] - 0s 46ms/step - loss: 0.9427 - accur
acy: 0.6917 - val_loss: 1.0022 - val_accuracy: 0.5667
Epoch 13/50
4/4 [==============================] - 0s 45ms/step - loss: 0.9268 - accur
acy: 0.6917 - val_loss: 0.9804 - val_accuracy: 0.5667
Epoch 14/50
4/4 [==============================] - 0s 38ms/step - loss: 0.9122 - accur
acy: 0.6917 - val_loss: 0.9597 - val_accuracy: 0.5667
Epoch 15/50
4/4 [==============================] - 0s 43ms/step - loss: 0.8963 - accur
acy: 0.6917 - val_loss: 0.9408 - val_accuracy: 0.5667
Epoch 16/50
4/4 [==============================] - 0s 41ms/step - loss: 0.8812 - accur
acy: 0.6917 - val_loss: 0.9223 - val_accuracy: 0.5667
Epoch 17/50
4/4 [==============================] - 0s 48ms/step - loss: 0.8673 - accur
acy: 0.6917 - val_loss: 0.9033 - val_accuracy: 0.5667
Epoch 18/50
4/4 [==============================] - 0s 41ms/step - loss: 0.8509 - accur
acy: 0.6917 - val_loss: 0.8856 - val_accuracy: 0.5667
Epoch 19/50
4/4 [==============================] - 0s 38ms/step - loss: 0.8346 - accur
acy: 0.6917 - val_loss: 0.8690 - val_accuracy: 0.5667
Epoch 20/50
4/4 [==============================] - 0s 39ms/step - loss: 0.8191 - accur
acy: 0.6917 - val_loss: 0.8521 - val_accuracy: 0.5667
Epoch 21/50
```

```
4/4 [==============================] - 0s 39ms/step - loss: 0.8037 - accur
acy: 0.6917 - val_loss: 0.8363 - val_accuracy: 0.5667
Epoch 22/50
4/4 [==============================] - 0s 36ms/step - loss: 0.7893 - accur
acy: 0.6917 - val_loss: 0.8207 - val_accuracy: 0.5667
Epoch 23/50
4/4 [==============================] - 0s 39ms/step - loss: 0.7765 - accur
acy: 0.6917 - val_loss: 0.8060 - val_accuracy: 0.5667
Epoch 24/50
4/4 [==============================] - 0s 47ms/step - loss: 0.7644 - accur
acy: 0.6917 - val_loss: 0.7925 - val_accuracy: 0.5667
Epoch 25/50
4/4 [==============================] - 0s 42ms/step - loss: 0.7524 - accur
acy: 0.6917 - val_loss: 0.7775 - val_accuracy: 0.5667
Epoch 26/50
4/4 [==============================] - 0s 36ms/step - loss: 0.7411 - accur
acy: 0.6917 - val_loss: 0.7625 - val_accuracy: 0.5667
Epoch 27/50
4/4 [==============================] - 0s 30ms/step - loss: 0.7296 - accur
acy: 0.6917 - val_loss: 0.7481 - val_accuracy: 0.5667
Epoch 28/50
4/4 [==============================] - 0s 38ms/step - loss: 0.7183 - accur
acy: 0.6917 - val_loss: 0.7345 - val_accuracy: 0.6000
Epoch 29/50
4/4 [==============================] - 0s 38ms/step - loss: 0.7077 - accur
acy: 0.6917 - val_loss: 0.7211 - val_accuracy: 0.6667
Epoch 30/50
4/4 [==============================] - 0s 33ms/step - loss: 0.6986 - accur
acy: 0.7250 - val_loss: 0.7068 - val_accuracy: 0.8000
Epoch 31/50
4/4 [==============================] - 0s 33ms/step - loss: 0.6878 - accur
acy: 0.7833 - val_loss: 0.6944 - val_accuracy: 0.8333
Epoch 32/50
4/4 [==============================] - 0s 43ms/step - loss: 0.6781 - accur
acy: 0.8083 - val_loss: 0.6836 - val_accuracy: 0.8333
Epoch 33/50
4/4 [==============================] - 0s 35ms/step - loss: 0.6689 - accur
acy: 0.8333 - val_loss: 0.6726 - val_accuracy: 0.9000
Epoch 34/50
4/4 [==============================] - 0s 34ms/step - loss: 0.6593 - accur
acy: 0.8667 - val_loss: 0.6617 - val_accuracy: 0.8667
Epoch 35/50
4/4 [==============================] - 0s 40ms/step - loss: 0.6506 - accur
acy: 0.9000 - val_loss: 0.6508 - val_accuracy: 0.9333
Epoch 36/50
4/4 [==============================] - 0s 38ms/step - loss: 0.6414 - accur
acy: 0.9000 - val_loss: 0.6415 - val_accuracy: 0.9333
Epoch 37/50
4/4 [==============================] - 0s 43ms/step - loss: 0.6333 - accur
acy: 0.8917 - val_loss: 0.6314 - val_accuracy: 0.9333
Epoch 38/50
4/4 [==============================] - 0s 32ms/step - loss: 0.6246 - accur
acy: 0.9000 - val_loss: 0.6219 - val_accuracy: 0.9333
Epoch 39/50
4/4 [==============================] - 0s 36ms/step - loss: 0.6166 - accur
acy: 0.9000 - val_loss: 0.6142 - val_accuracy: 0.9333
Epoch 40/50
4/4 [==============================] - 0s 39ms/step - loss: 0.6086 - accur
acy: 0.9000 - val_loss: 0.6050 - val_accuracy: 0.9333
Epoch 41/50
4/4 [==============================] - 0s 38ms/step - loss: 0.6007 - accur
```

```
acy: 0.9083 - val_loss: 0.5969 - val_accuracy: 0.9333
Epoch 42/50
4/4 [==============================] - 0s 45ms/step - loss: 0.5931 - accur
acy: 0.9083 - val_loss: 0.5895 - val_accuracy: 0.9333
Epoch 43/50
4/4 [==============================] - 0s 31ms/step - loss: 0.5856 - accur
acy: 0.9083 - val_loss: 0.5821 - val_accuracy: 0.9333
Epoch 44/50
4/4 [==============================] - 0s 34ms/step - loss: 0.5784 - accur
acy: 0.9167 - val_loss: 0.5750 - val_accuracy: 0.9333
Epoch 45/50
4/4 [==============================] - 0s 47ms/step - loss: 0.5712 - accur
acy: 0.9167 - val_loss: 0.5676 - val_accuracy: 0.9333
Epoch 46/50
4/4 [==============================] - 0s 42ms/step - loss: 0.5643 - accur
acy: 0.9167 - val_loss: 0.5598 - val_accuracy: 0.9333
Epoch 47/50
4/4 [==============================] - 0s 49ms/step - loss: 0.5575 - accur
acy: 0.9083 - val_loss: 0.5517 - val_accuracy: 0.9333
Epoch 48/50
4/4 [==============================] - 0s 43ms/step - loss: 0.5507 - accur
acy: 0.9083 - val_loss: 0.5450 - val_accuracy: 0.9333
Epoch 49/50
4/4 [==============================] - 0s 44ms/step - loss: 0.5443 - accur
acy: 0.9083 - val_loss: 0.5384 - val_accuracy: 0.9333
Epoch 50/50
4/4 [==============================] - 0s 32ms/step - loss: 0.5379 - accur
acy: 0.9167 - val_loss: 0.5325 - val_accuracy: 0.9667
```

In [123]:

```
%reload_ext tensorboard
```

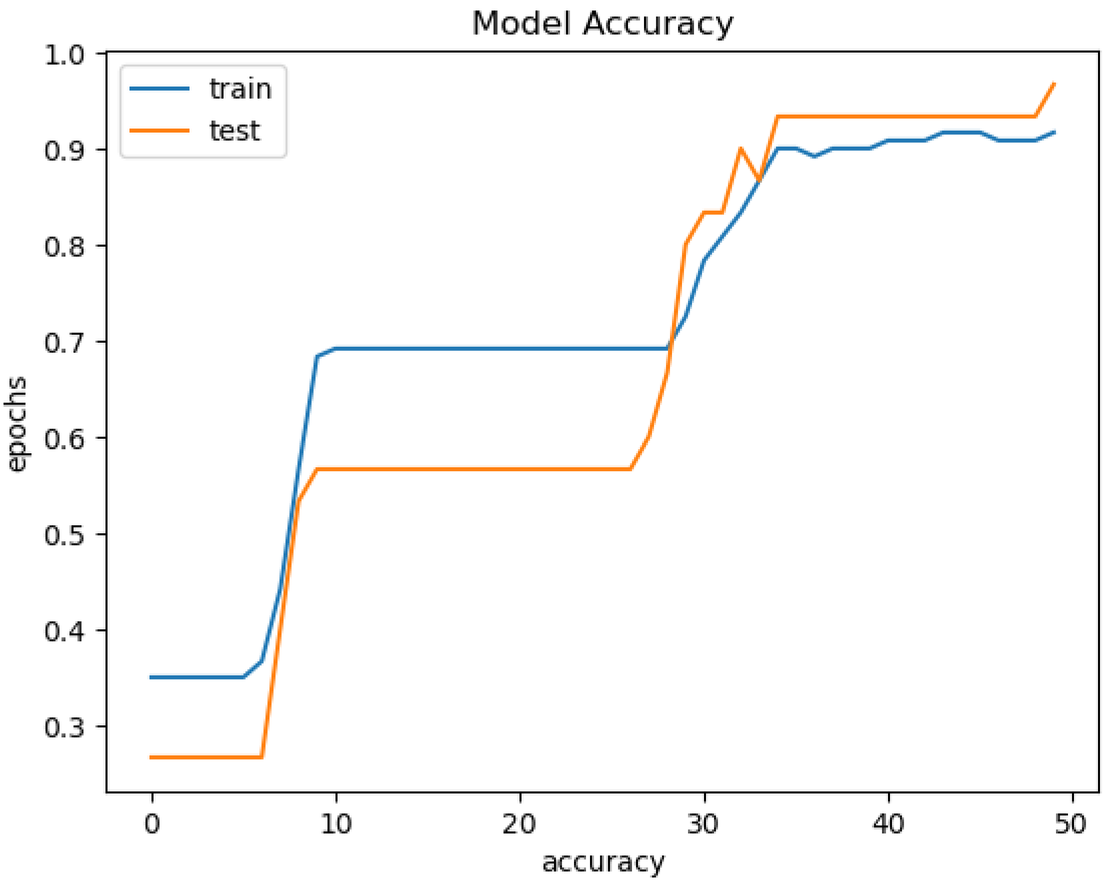
In [37]:

```
tensorboard --logdir logs/fit
```

```
Reusing TensorBoard on port 6006 (pid 13432), started 21 days, 21:13:43 ag
o. (Use '!kill 13432' to kill it.)
```

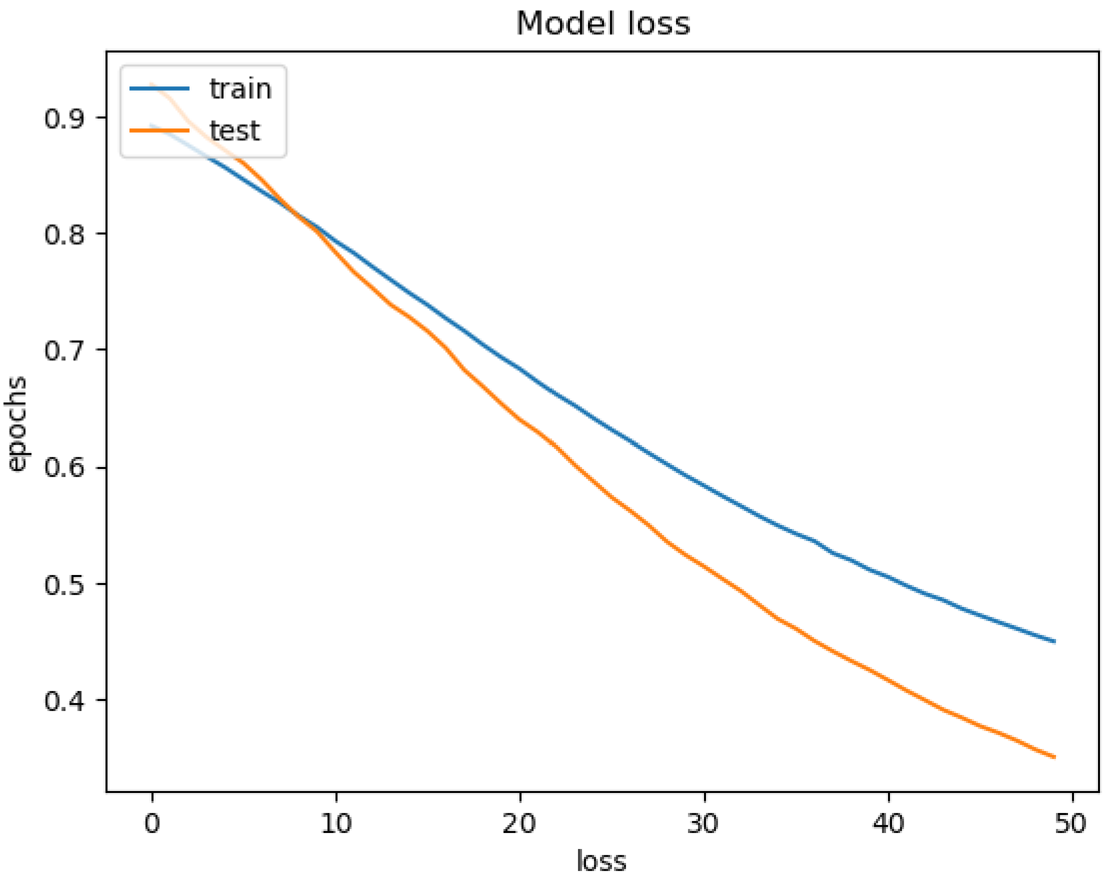
In [206]:

```
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model Accuracy')
plt.xlabel('accuracy')
plt.ylabel('epochs')
plt.legend(['train','test'],loc='upper left')
plt.show()
```



In [199]:

```
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.xlabel('loss')
plt.ylabel('epochs')
plt.legend(['train','test'],loc='upper left')
plt.show()
```



In [ ]: