Name – Umar Momin

PRN -20210812016

DNN

# Building a NN model with tensorflow

**AIM-** Write a program to implement a tensorflow of NN model

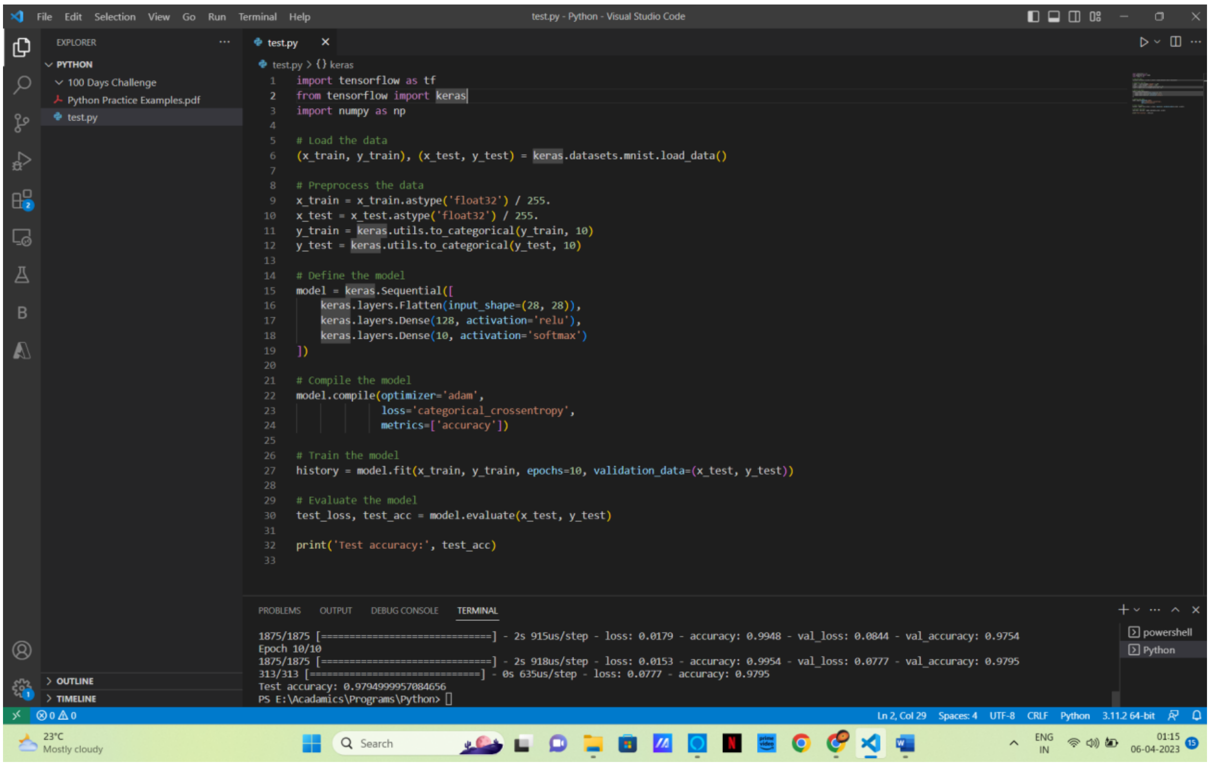
**Theory-**

TensorFlow is a popular open-source framework for building and deploying machine learning models. When building a neural network model in TensorFlow, there are several steps that need to be followed. First, the input data needs to be preprocessed and prepared for training. This can involve transforming the data into a format suitable for the model, scaling the features, and splitting the data into training and testing sets.

Next, the architecture of the neural network needs to be defined. This involves specifying the number of layers, the number of neurons in each layer, and the activation functions used. The parameters of the model are then initialized randomly, and the model is trained using an optimization algorithm such as stochastic gradient descent.

Once training is complete, the model can be evaluated on the test set to assess its performance. In addition, the model can be used to make predictions on new data. TensorFlow provides a range of tools and APIs to aid in these tasks, as well as to deploy the models in production environments.

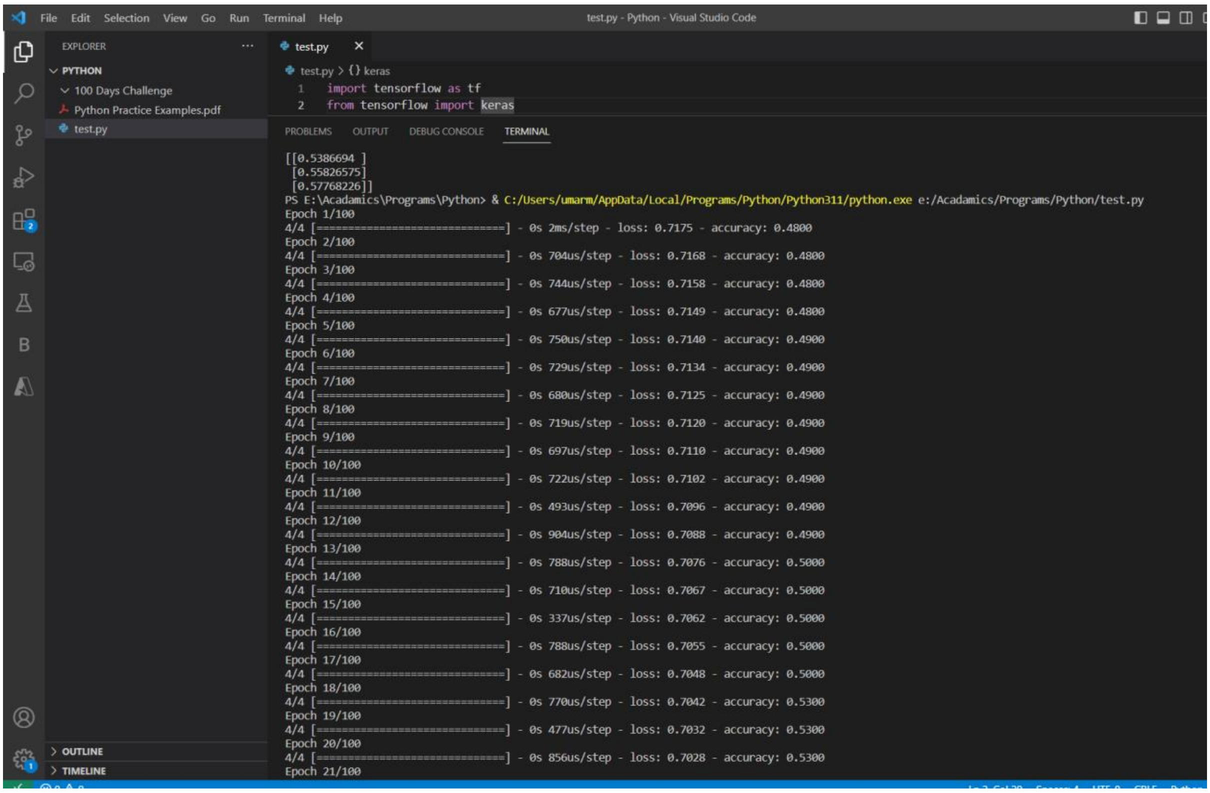
**Program-**



File Edit Selection View Go Run Terminal Help
test.py - Python - Visual Studio Code
EXPLORER
PYTHON
100 Days Challenge
Python Practice Examples.pdf
test.py
test.py > {} keras
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
powershell
Python
Ln 2, Col 29 Spaces: 4 UTF-8 CRLF Python 3.11.2 64-bit
23°C
Mostly cloudy
Search
ENG
IN
01:15
06-04-2023

```python
import tensorflow as tf
from tensorflow import keras
import numpy as np

# Load the data
(x_train, y_train), (x_test, y_test) = keras.datasets.mnist.load_data()

# Preprocess the data
x_train = x_train.astype('float32') / 255.
x_test = x_test.astype('float32') / 255.
y_train = keras.utils.to_categorical(y_train, 10)
y_test = keras.utils.to_categorical(y_test, 10)

# Define the model
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(28, 28)),
    keras.layers.Dense(128, activation='relu'),
    keras.layers.Dense(10, activation='softmax')
])

# Compile the model
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])

# Train the model
history = model.fit(x_train, y_train, epochs=10, validation_data=(x_test, y_test))

# Evaluate the model
test_loss, test_acc = model.evaluate(x_test, y_test)

print('Test accuracy:', test_acc)
```

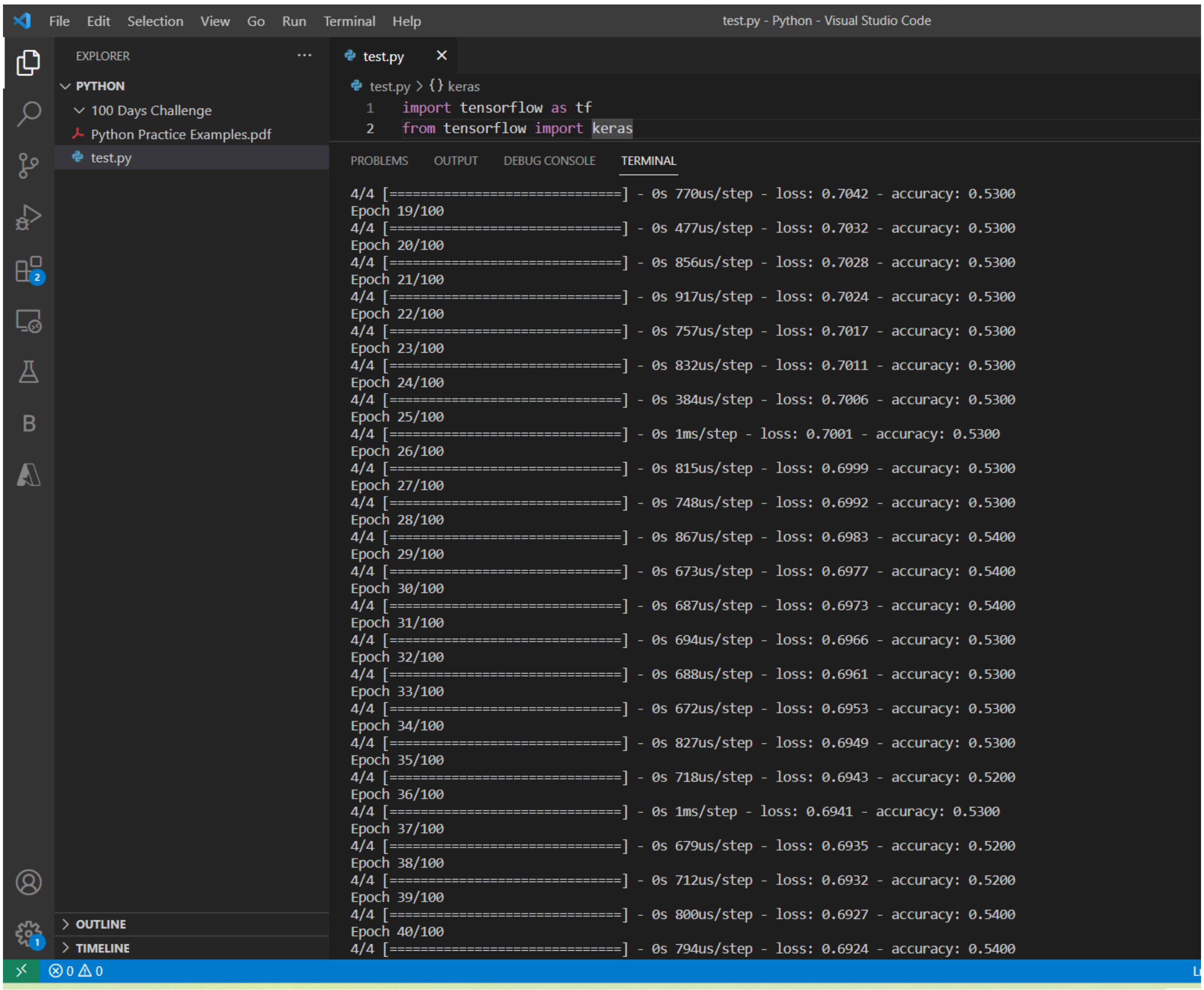
```
1875/1875 [==============================] - 2s 915us/step - loss: 0.0179 - accuracy: 0.9948 - val_loss: 0.0844 - val_accuracy: 0.9754
Epoch 10/10
1875/1875 [==============================] - 2s 918us/step - loss: 0.0153 - accuracy: 0.9954 - val_loss: 0.0777 - val_accuracy: 0.9795
313/313 [==============================] - 0s 635us/step - loss: 0.0777 - accuracy: 0.9795
Test accuracy: 0.9794999957084656
PS E:\Acadamics\Programs\Python> 
```



File Edit Selection View Go Run Terminal Help
test.py - Python - Visual Studio Code
EXPLORER
PYTHON
100 Days Challenge
Python Practice Examples.pdf
test.py
test.py > {} keras
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

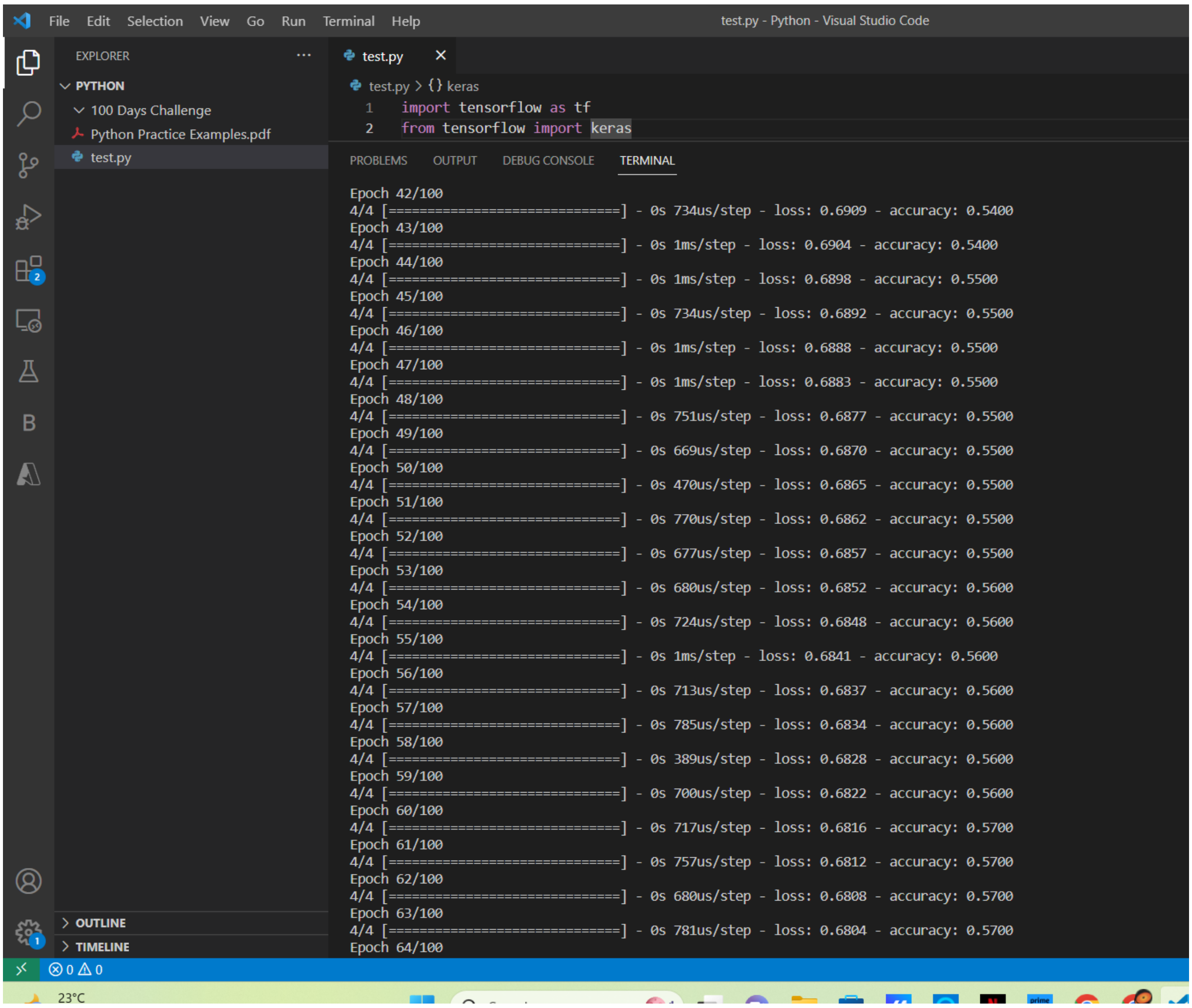
```python
import tensorflow as tf
from tensorflow import keras
```

```
[[0.5386694 ]
 [0.55826575]
 [0.57768226]]
PS E:\Acadamics\Programs\Python> & C:/Users/umarm/AppData/Local/Programs/Python/Python311/python.exe e:/Acadamics/Programs/Python/test.py
Epoch 1/100
4/4 [==============================] - 0s 2ms/step - loss: 0.7175 - accuracy: 0.4800
Epoch 2/100
4/4 [==============================] - 0s 704us/step - loss: 0.7168 - accuracy: 0.4800
Epoch 3/100
4/4 [==============================] - 0s 744us/step - loss: 0.7158 - accuracy: 0.4800
Epoch 4/100
4/4 [==============================] - 0s 677us/step - loss: 0.7149 - accuracy: 0.4800
Epoch 5/100
4/4 [==============================] - 0s 750us/step - loss: 0.7140 - accuracy: 0.4900
Epoch 6/100
4/4 [==============================] - 0s 729us/step - loss: 0.7134 - accuracy: 0.4900
Epoch 7/100
4/4 [==============================] - 0s 680us/step - loss: 0.7125 - accuracy: 0.4900
Epoch 8/100
4/4 [==============================] - 0s 719us/step - loss: 0.7120 - accuracy: 0.4900
Epoch 9/100
4/4 [==============================] - 0s 697us/step - loss: 0.7110 - accuracy: 0.4900
Epoch 10/100
4/4 [==============================] - 0s 722us/step - loss: 0.7102 - accuracy: 0.4900
Epoch 11/100
4/4 [==============================] - 0s 493us/step - loss: 0.7096 - accuracy: 0.4900
Epoch 12/100
4/4 [==============================] - 0s 904us/step - loss: 0.7088 - accuracy: 0.4900
Epoch 13/100
4/4 [==============================] - 0s 788us/step - loss: 0.7076 - accuracy: 0.5000
Epoch 14/100
4/4 [==============================] - 0s 710us/step - loss: 0.7067 - accuracy: 0.5000
Epoch 15/100
4/4 [==============================] - 0s 337us/step - loss: 0.7062 - accuracy: 0.5000
Epoch 16/100
4/4 [==============================] - 0s 788us/step - loss: 0.7055 - accuracy: 0.5000
Epoch 17/100
4/4 [==============================] - 0s 682us/step - loss: 0.7048 - accuracy: 0.5000
Epoch 18/100
4/4 [==============================] - 0s 770us/step - loss: 0.7042 - accuracy: 0.5300
Epoch 19/100
4/4 [==============================] - 0s 477us/step - loss: 0.7032 - accuracy: 0.5300
Epoch 20/100
4/4 [==============================] - 0s 856us/step - loss: 0.7028 - accuracy: 0.5300
Epoch 21/100
```



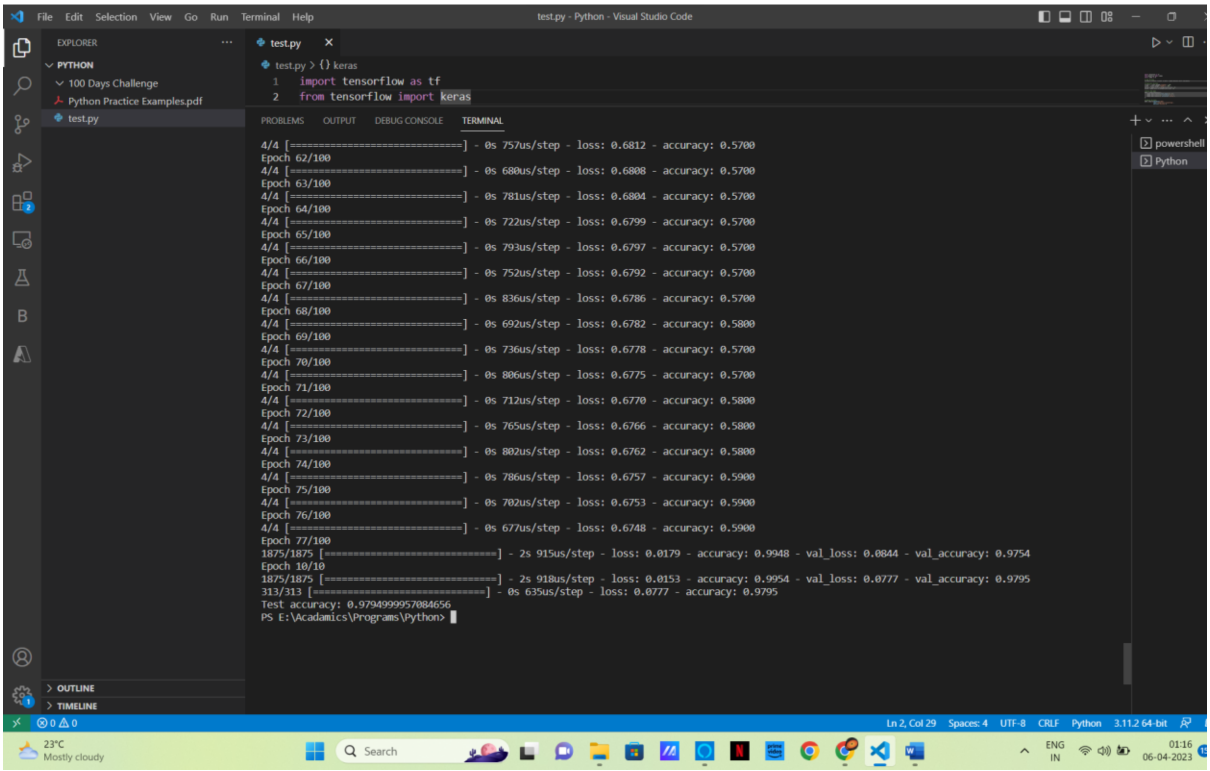
```python
import tensorflow as tf
from tensorflow import keras
```

```
4/4 [==============================] - 0s 770us/step - loss: 0.7042 - accuracy: 0.5300
Epoch 19/100
4/4 [==============================] - 0s 477us/step - loss: 0.7032 - accuracy: 0.5300
Epoch 20/100
4/4 [==============================] - 0s 856us/step - loss: 0.7028 - accuracy: 0.5300
Epoch 21/100
4/4 [==============================] - 0s 917us/step - loss: 0.7024 - accuracy: 0.5300
Epoch 22/100
4/4 [==============================] - 0s 757us/step - loss: 0.7017 - accuracy: 0.5300
Epoch 23/100
4/4 [==============================] - 0s 832us/step - loss: 0.7011 - accuracy: 0.5300
Epoch 24/100
4/4 [==============================] - 0s 384us/step - loss: 0.7006 - accuracy: 0.5300
Epoch 25/100
4/4 [==============================] - 0s 1ms/step - loss: 0.7001 - accuracy: 0.5300
Epoch 26/100
4/4 [==============================] - 0s 815us/step - loss: 0.6999 - accuracy: 0.5300
Epoch 27/100
4/4 [==============================] - 0s 748us/step - loss: 0.6992 - accuracy: 0.5300
Epoch 28/100
4/4 [==============================] - 0s 867us/step - loss: 0.6983 - accuracy: 0.5400
Epoch 29/100
4/4 [==============================] - 0s 673us/step - loss: 0.6977 - accuracy: 0.5400
Epoch 30/100
4/4 [==============================] - 0s 687us/step - loss: 0.6973 - accuracy: 0.5400
Epoch 31/100
4/4 [==============================] - 0s 694us/step - loss: 0.6966 - accuracy: 0.5300
Epoch 32/100
4/4 [==============================] - 0s 688us/step - loss: 0.6961 - accuracy: 0.5300
Epoch 33/100
4/4 [==============================] - 0s 672us/step - loss: 0.6953 - accuracy: 0.5300
Epoch 34/100
4/4 [==============================] - 0s 827us/step - loss: 0.6949 - accuracy: 0.5300
Epoch 35/100
4/4 [==============================] - 0s 718us/step - loss: 0.6943 - accuracy: 0.5200
Epoch 36/100
4/4 [==============================] - 0s 1ms/step - loss: 0.6941 - accuracy: 0.5300
Epoch 37/100
4/4 [==============================] - 0s 679us/step - loss: 0.6935 - accuracy: 0.5200
Epoch 38/100
4/4 [==============================] - 0s 712us/step - loss: 0.6932 - accuracy: 0.5200
Epoch 39/100
4/4 [==============================] - 0s 800us/step - loss: 0.6927 - accuracy: 0.5400
Epoch 40/100
4/4 [==============================] - 0s 794us/step - loss: 0.6924 - accuracy: 0.5400
```


File Edit Selection View Go Run Terminal Help
test.py - Python - Visual Studio Code
EXPLORER
PYTHON
100 Days Challenge
Python Practice Examples.pdf
test.py
test.py > {} keras
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
OUTLINE
TIMELINE
23°C

```python
import tensorflow as tf
from tensorflow import keras
```

```
Epoch 42/100
4/4 [==============================] - 0s 734us/step - loss: 0.6909 - accuracy: 0.5400
Epoch 43/100
4/4 [==============================] - 0s 1ms/step - loss: 0.6904 - accuracy: 0.5400
Epoch 44/100
4/4 [==============================] - 0s 1ms/step - loss: 0.6898 - accuracy: 0.5500
Epoch 45/100
4/4 [==============================] - 0s 734us/step - loss: 0.6892 - accuracy: 0.5500
Epoch 46/100
4/4 [==============================] - 0s 1ms/step - loss: 0.6888 - accuracy: 0.5500
Epoch 47/100
4/4 [==============================] - 0s 1ms/step - loss: 0.6883 - accuracy: 0.5500
Epoch 48/100
4/4 [==============================] - 0s 751us/step - loss: 0.6877 - accuracy: 0.5500
Epoch 49/100
4/4 [==============================] - 0s 669us/step - loss: 0.6870 - accuracy: 0.5500
Epoch 50/100
4/4 [==============================] - 0s 470us/step - loss: 0.6865 - accuracy: 0.5500
Epoch 51/100
4/4 [==============================] - 0s 770us/step - loss: 0.6862 - accuracy: 0.5500
Epoch 52/100
4/4 [==============================] - 0s 677us/step - loss: 0.6857 - accuracy: 0.5500
Epoch 53/100
4/4 [==============================] - 0s 680us/step - loss: 0.6852 - accuracy: 0.5600
Epoch 54/100
4/4 [==============================] - 0s 724us/step - loss: 0.6848 - accuracy: 0.5600
Epoch 55/100
4/4 [==============================] - 0s 1ms/step - loss: 0.6841 - accuracy: 0.5600
Epoch 56/100
4/4 [==============================] - 0s 713us/step - loss: 0.6837 - accuracy: 0.5600
Epoch 57/100
4/4 [==============================] - 0s 785us/step - loss: 0.6834 - accuracy: 0.5600
Epoch 58/100
4/4 [==============================] - 0s 389us/step - loss: 0.6828 - accuracy: 0.5600
Epoch 59/100
4/4 [==============================] - 0s 700us/step - loss: 0.6822 - accuracy: 0.5600
Epoch 60/100
4/4 [==============================] - 0s 717us/step - loss: 0.6816 - accuracy: 0.5700
Epoch 61/100
4/4 [==============================] - 0s 757us/step - loss: 0.6812 - accuracy: 0.5700
Epoch 62/100
4/4 [==============================] - 0s 680us/step - loss: 0.6808 - accuracy: 0.5700
Epoch 63/100
4/4 [==============================] - 0s 781us/step - loss: 0.6804 - accuracy: 0.5700
Epoch 64/100
```


File Edit Selection View Go Run Terminal Help
test.py - Python - Visual Studio Code
EXPLORER
PYTHON
100 Days Challenge
Python Practice Examples.pdf
test.py
test.py > {} keras
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
powershell
Python
OUTLINE
TIMELINE
Ln 2, Col 29 Spaces: 4 UTF-8 CRLF Python 3.11.2 64-bit
23°C
Mostly cloudy
Search
ENG
IN
01:16
06-04-2023

```python
import tensorflow as tf
from tensorflow import keras
```

```
4/4 [==============================] - 0s 757us/step - loss: 0.6812 - accuracy: 0.5700
Epoch 62/100
4/4 [==============================] - 0s 680us/step - loss: 0.6808 - accuracy: 0.5700
Epoch 63/100
4/4 [==============================] - 0s 781us/step - loss: 0.6804 - accuracy: 0.5700
Epoch 64/100
4/4 [==============================] - 0s 722us/step - loss: 0.6799 - accuracy: 0.5700
Epoch 65/100
4/4 [==============================] - 0s 793us/step - loss: 0.6797 - accuracy: 0.5700
Epoch 66/100
4/4 [==============================] - 0s 752us/step - loss: 0.6792 - accuracy: 0.5700
Epoch 67/100
4/4 [==============================] - 0s 836us/step - loss: 0.6786 - accuracy: 0.5700
Epoch 68/100
4/4 [==============================] - 0s 692us/step - loss: 0.6782 - accuracy: 0.5800
Epoch 69/100
4/4 [==============================] - 0s 736us/step - loss: 0.6778 - accuracy: 0.5700
Epoch 70/100
4/4 [==============================] - 0s 806us/step - loss: 0.6775 - accuracy: 0.5700
Epoch 71/100
4/4 [==============================] - 0s 712us/step - loss: 0.6770 - accuracy: 0.5800
Epoch 72/100
4/4 [==============================] - 0s 765us/step - loss: 0.6766 - accuracy: 0.5800
Epoch 73/100
4/4 [==============================] - 0s 802us/step - loss: 0.6762 - accuracy: 0.5800
Epoch 74/100
4/4 [==============================] - 0s 786us/step - loss: 0.6757 - accuracy: 0.5900
Epoch 75/100
4/4 [==============================] - 0s 702us/step - loss: 0.6753 - accuracy: 0.5900
Epoch 76/100
4/4 [==============================] - 0s 677us/step - loss: 0.6748 - accuracy: 0.5900
Epoch 77/100
1875/1875 [==============================] - 2s 915us/step - loss: 0.0179 - accuracy: 0.9948 - val_loss: 0.0844 - val_accuracy: 0.9754
Epoch 10/10
1875/1875 [==============================] - 2s 918us/step - loss: 0.0153 - accuracy: 0.9954 - val_loss: 0.0777 - val_accuracy: 0.9795
313/313 [==============================] - 0s 635us/step - loss: 0.0777 - accuracy: 0.9795
Test accuracy: 0.9794999957084656
PS E:\Acadamics\Programs\Python> 
```

**Conclusion –**

**Here in this practical we have successfully performed program on NN model using tensorflow**