Name – Umar Momin

PRN -20210812016

DNN

Tensorflow implementation of logistic regression.

AIM- Write a program to implement a tensorflow of logistic regression

Theory-

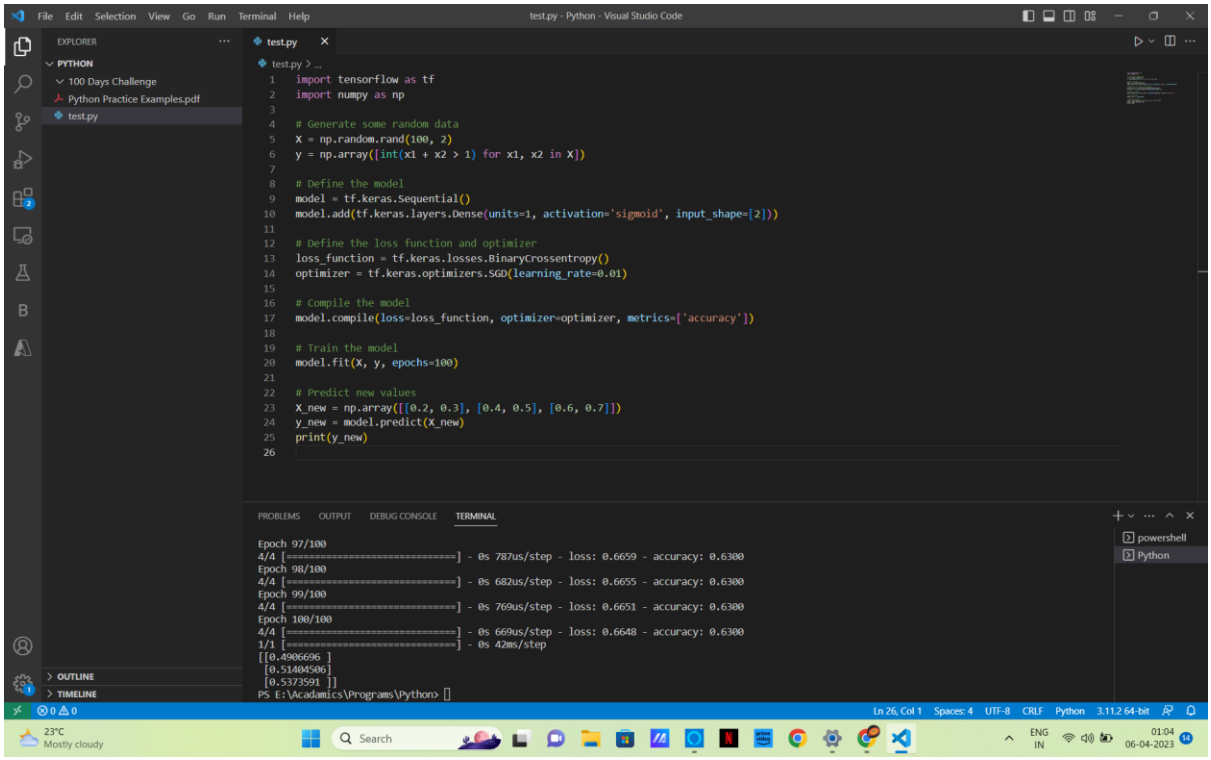
Tensorflow is a popular open-source library used for machine learning and deep learning applications. It has become an industry standard for building and deploying complex neural networks.

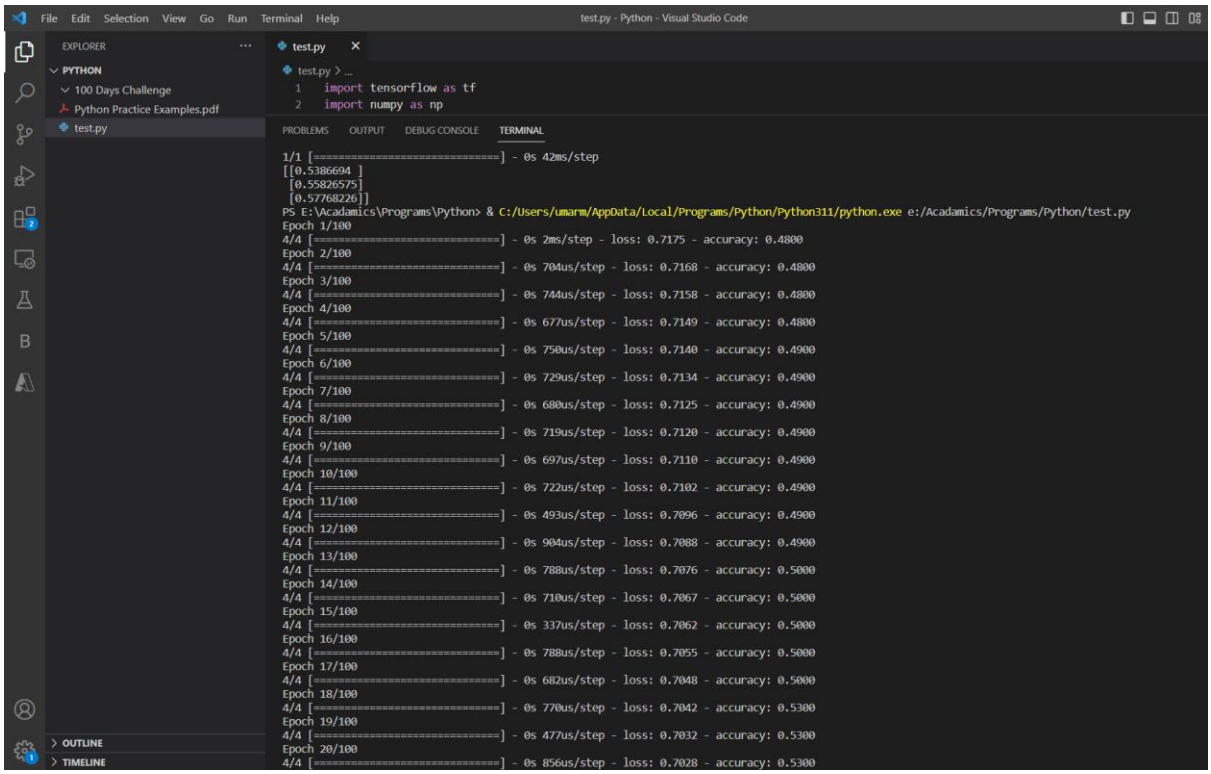
Logistic regression is a commonly used statistical method for binary classification. It involves predicting the probability of an outcome being true or false based on input variables. TensorFlow provides a straightforward implementation of logistic regression using its computational graph and loss function optimization capabilities.

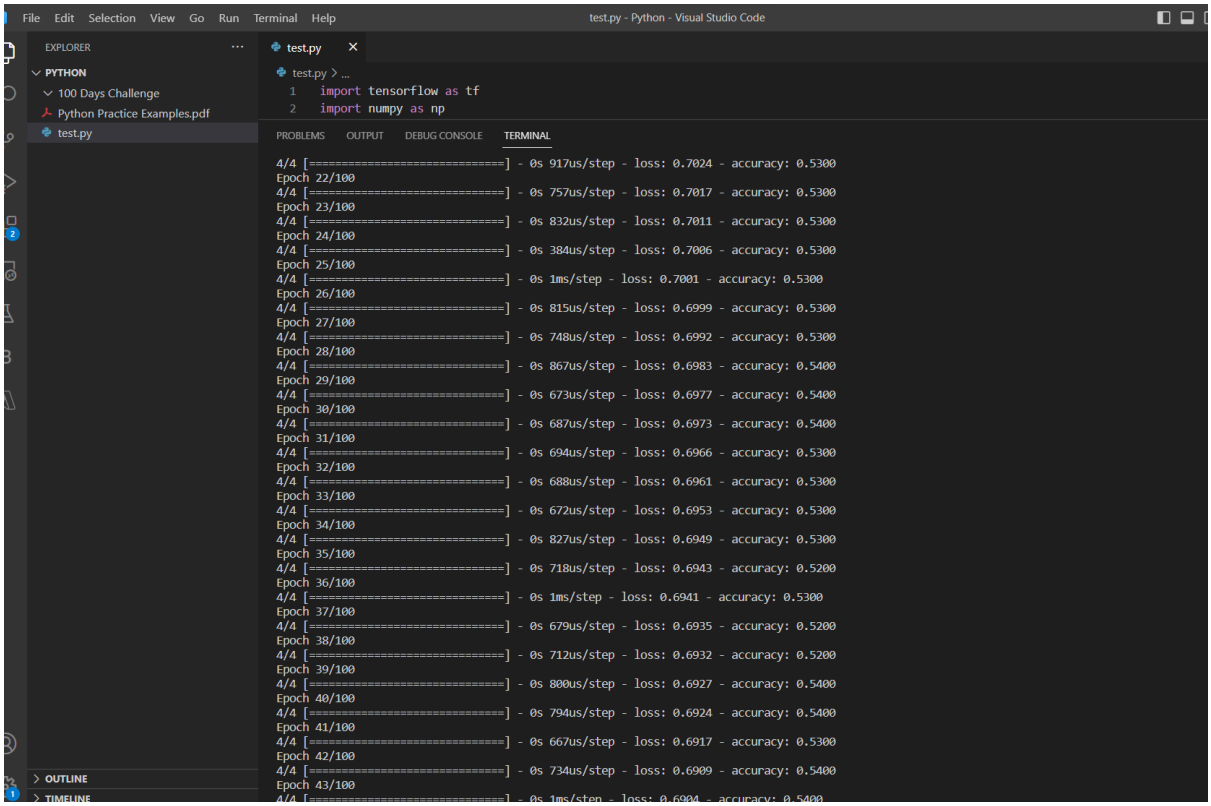
The basic steps involved in the TensorFlow implementation of logistic regression are to define the input features, create the model, set up the loss function, optimize the loss using gradient descent, and train the model on the dataset.

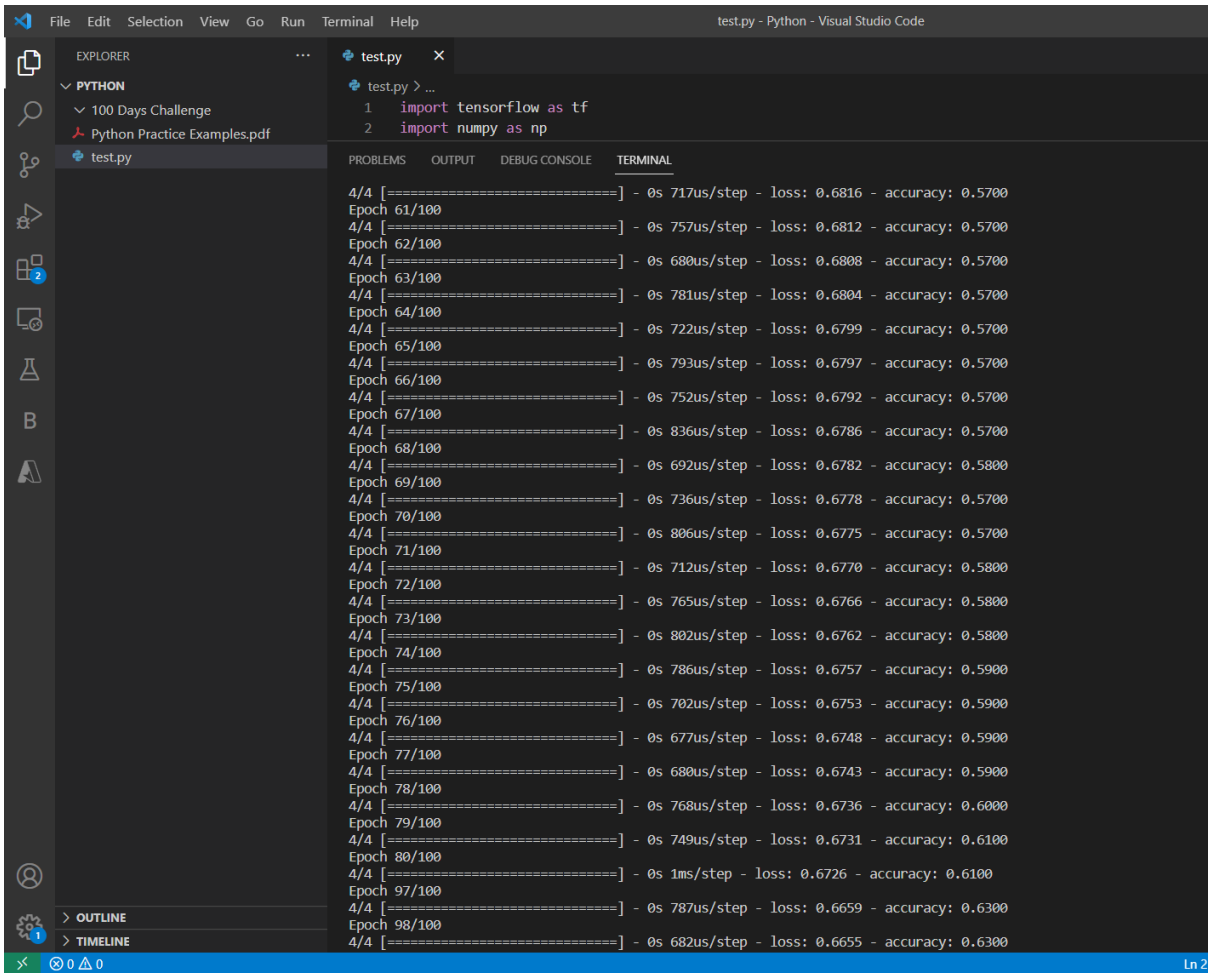
TensorFlow offers extensive support for building and training machine learning models, and its implementation of logistic regression is a prime example. The versatility and flexibility of TensorFlow make it an ideal tool for all kinds of machine learning and deep learning applications.

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
import tensorflow as tf
import numpy as np
# Generate some random data
X = np.random.rand(100, 2)
y = np.array([int(x1 + x2 > 1) for x1, x2 in X])
# Define the model
model = tf.keras.Sequential()
model.add(tf.keras.layers.Dense(units=1, activation='sigmoid', input_shape=[2]))
# Define the loss function and optimizer
loss_function = tf.keras.losses.BinaryCrossentropy()
optimizer = tf.keras.optimizers.SGD(learning_rate=0.01)
# Compile the model
model.compile(loss=loss_function, optimizer=optimizer, metrics=['accuracy'])
# Train the model
model.fit(X, y, epochs=100)
# Predict new values
X_new = np.array([[0.2, 0.3], [0.4, 0.5], [0.6, 0.7]])
y_new = model.predict(X_new)
print(y_new)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Epoch 97/100
4/4 [==============================] - 0s 787us/step - loss: 0.6659 - accuracy: 0.6300
Epoch 98/100
4/4 [==============================] - 0s 682us/step - loss: 0.6655 - accuracy: 0.6300
Epoch 99/100
4/4 [==============================] - 0s 769us/step - loss: 0.6651 - accuracy: 0.6300
Epoch 100/100
4/4 [==============================] - 0s 669us/step - loss: 0.6648 - accuracy: 0.6300
1/1 [==============================] - 0s 42ms/step
[[0.4906696]
[0.51404506]
[0.5373591]]
PS E:\Acadamics\Programs\Python>
powershell
Python
OUTLINE
TIMELINE
Ln 26, Col 1 Spaces: 4 UTF-8 CRLF Python 3.11.2 64-bit
23°C
Mostly cloudy
Search
ENG
IN
01:04
06-04-2023


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
import tensorflow as tf
import numpy as np
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1/1 [==============================] - 0s 42ms/step
[[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
OUTLINE
TIMELINE


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
test.py > ...
1 import tensorflow as tf
2 import numpy as np
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
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
Epoch 41/100
4/4 [==============================] - 0s 667us/step - loss: 0.6917 - accuracy: 0.5300
Epoch 42/100
4/4 [==============================] - 0s 734us/step - loss: 0.6909 - accuracy: 0.5400
Epoch 43/100
OUTLINE
TIMELINE


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
test.py > ...
1 import tensorflow as tf
2 import numpy as np
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
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
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
4/4 [==============================] - 0s 680us/step - loss: 0.6743 - accuracy: 0.5900
Epoch 78/100
4/4 [==============================] - 0s 768us/step - loss: 0.6736 - accuracy: 0.6000
Epoch 79/100
4/4 [==============================] - 0s 749us/step - loss: 0.6731 - accuracy: 0.6100
Epoch 80/100
4/4 [==============================] - 0s 1ms/step - loss: 0.6726 - accuracy: 0.6100
Epoch 97/100
4/4 [==============================] - 0s 787us/step - loss: 0.6659 - accuracy: 0.6300
Epoch 98/100
4/4 [==============================] - 0s 682us/step - loss: 0.6655 - accuracy: 0.6300
OUTLINE
TIMELINE
0 0
Ln 2

Conclusion-

Here in this practical we have successfully performed program on logistic regression using tensorflow