EXPERIMENT NO 2

Name – Kaivalya Deshpande

Roll No – 32

```
import numpy as np
import pandas as pd
import tensorflow as ts
from tensorflow import keras as ks
import matplotlib.pyplot as plt
import statsmodels.api as sm
from sklearn.model_selection import train_test_split  #for spliting the
dataset
import keras as ks
from keras.layers import Dense
from keras.activations import sigmoid
from keras.losses import binary_crossentropy
from keras.optimizers import Adam
```

```
df=pd.read_csv('/content/insurance_data.csv')
df.head(10)
```

	age	affordibility	bought_insurance
0	22	1	0
1	25	0	0
2	47	1	1
3	52	0	0
4	46	1	1
5	56	1	1
6	55	0	0
7	60	0	1
8	62	1	1
9	61	1	1

```
x = df[['age', 'affordibility']]  # inputs
y = df['bought_insurance']        # outputs



# Split the data into training and testing sets
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2,
random_state=10)
x_train_scale=x_train.copy()
x_train_scale['age']=x_train_scale['age']/100


x_test_scale=x_test.copy()
x_test_scale['age']=x_test_scale['age']/100




import keras
from keras.models import Sequential
from keras.layers import Dense
model = keras.Sequential([
    keras.layers.Dense(1, input_shape=(2,), activation='sigmoid',
kernel_initializer='ones', bias_initializer='zeros')
])
model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])
model.fit(x_train_scale, y_train, epochs=5000)
```

```
Streaming output truncated to the last 5000 lines.
1/1 [==============================] - 0s 11ms/step - loss: 0.5761 - accuracy: 0.6818
Epoch 1669/5000
1/1 [==============================] - 0s 10ms/step - loss: 0.5761 - accuracy: 0.6818
Epoch 1670/5000
1/1 [==============================] - 0s 10ms/step - loss: 0.5760 - accuracy: 0.6818
Epoch 1671/5000
1/1 [==============================] - 0s 10ms/step - loss: 0.5760 - accuracy: 0.6818
Epoch 1672/5000
1/1 [==============================] - 0s 10ms/step - loss: 0.5759 - accuracy: 0.6818
Epoch 1673/5000
1/1 [==============================] - 0s 11ms/step - loss: 0.5759 - accuracy: 0.6818
Epoch 1674/5000
1/1 [==============================] - 0s 11ms/step - loss: 0.5759 - accuracy: 0.6818
Epoch 1675/5000
1/1 [==============================] - 0s 9ms/step - loss: 0.5758 - accuracy: 0.6818
Epoch 1676/5000
1/1 [==============================] - 0s 11ms/step - loss: 0.5758 - accuracy: 0.6818
Epoch 1677/5000
1/1 [==============================] - 0s 11ms/step - loss: 0.5757 - accuracy: 0.6818
Epoch 1678/5000
1/1 [==============================] - 0s 13ms/step - loss: 0.5757 - accuracy: 0.6818
Epoch 1679/5000
1/1 [==============================] - 0s 10ms/step - loss: 0.5756 - accuracy: 0.6818
Epoch 1680/5000
1/1 [==============================] - 0s 10ms/step - loss: 0.5756 - accuracy: 0.6818
Epoch 1681/5000
```