

1. Go through the csv (Indiana-diabetes) file and find the prediction of first 15 records. Change the number of layers and then test the model. What is the change you note?

```
In [1]: import numpy as np
import pandas as pd
import tensorflow as tf
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix, accuracy_score
```

```
In [2]: df = pd.read_csv(r'C:\Users\Nirmalya Majhi\Desktop\Advanced IT Workshop\pima-indians-dia
x = df.iloc[:15, :].values
y = df.iloc[:15, -1].values
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.3,random_state=0)
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.fit_transform(x_test)
ann = tf.keras.models.Sequential()
ann.add(tf.keras.layers.Dense(units=6, activation='relu'))
ann.add(tf.keras.layers.Dense(units=6, activation='relu'))
ann.add(tf.keras.layers.Dense(units=1, activation='sigmoid'))
ann.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
ann.fit(x_train,y_train,batch_size=15)
y_pred = ann.predict(x_test)
y_pred = y_pred > 0.5
print("Predicting Test set results: \n",
      np.concatenate((y_pred.reshape(len(y_pred),1),y_test.reshape(len(y_test),1)),1))
cm = confusion_matrix(y_test, y_pred)
print("Confusion Matrix: \n",cm)
print("Accuracy (%) = ",(accuracy_score(y_test,y_pred))*100)
```

```
1/1 [=====] - 0s 434ms/step - loss: 0.8675 - accuracy: 0.5000
1/1 [=====] - 0s 48ms/step
```

Predicting Test set results:

```
[[1 1]
 [1 0]
 [1 1]
 [1 0]
 [1 1]]
```

Confusion Matrix:

```
[[0 2]
 [0 3]]
```

Accuracy (%) = 60.0

In []: