

```

# Import libraries and data

import tensorflow as tf
from tensorflow import keras
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
import pandas as pd
from sklearn.metrics import accuracy_score

data = pd.read_csv('C:/Users/PandaBas/Desktop/Personal/Teaching/5thSem_PartA
/bank_marketing.csv')
data.head()

# Scale the numeric columns

col_to_scale = ['age']

data[col_to_scale] = scaler.fit_transform(data[col_to_scale])
data.head()

# Choose the independent and dependent variable

X = data.drop('y',axis = 1)
Y = data['y']

# Train, test/validation split
X_train, X_test, y_train, y_test = train_test_split(X,Y, test_size = 0.20,
random_state = 42)
test_df = pd.concat ([X_test,y_test],axis =1 )

# Create the network

model = keras.Sequential([
    keras.layers.Dense(3, input_shape = (5,),activation = 'relu'),
    keras.layers.Dense(2, input_shape = (3,),activation = 'relu'),
    keras.layers.Dense(1,activation = 'sigmoid'),
])

# Compile the network

model.compile(optimizer = 'adam',
              loss = 'binary_crossentropy',
              metrics = ['accuracy']
              )

```

```
# train the model
model.fit(X_train,y_train,epochs = 5)

# Evaluate the model

model.evaluate(X_test,y_test)

# Predict based on ANN model
yp = model.predict(X_test)

y_pred = []
for element in yp:
    if element > 0.5:
        y_pred.append(1)
    else:
        y_pred.append(0)

# Check accuracy of the model

accuracy_score(y_test, y_pred)
```