Binary classification using Deep Neural Networks Example:

Classify movie reviews into positive" reviews and "negative" reviews, just based on the text content of the reviews. Use IMDB dataset

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

```
df1=pd.read_csv("/content/IMDB Dataset.csv")
#df1=df.head(1000)
df1
```

```
import tensorflow.keras as tk
from keras.preprocessing.text import Tokenizer
tokenizer1=Tokenizer(oov_token='<nothing>')
```

```
tokenizer1.fit_on_texts(df1['review'])
```

```
tokenizer1.word_index
```

```
tokenizer1.word_counts
```

```
tokenizer1.document_count
```

```
df1['review']=tokenizer1.texts_to_sequences(df1['review'])
```

```
df1
```

```
from keras.utils import pad_sequences
seq_df=pd.DataFrame(pad_sequences(df1['review'],padding="post"))
seq_df
```

```
df1 = df1.join(seq_df)
```

```
df1.drop(['review'],axis=1,inplace=True)
```

```
df1['sentiment'].value_counts()
```

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
df1['sentiment']=le.fit_transform(df1['sentiment'])
df1['sentiment']
```

```
df1.head(2)
```

```
from sklearn.model_selection import train_test_split
```

```
y = df1['sentiment']
x = df1.iloc[:,1:]
```

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=42)
```

```
x_train
```

```
import tensorflow.keras as tk
```

```
model = tk.Sequential()
model.add(tk.layers.Input(shape=(2493,)))
model.add(tk.layers.Dense(50, activation='relu',kernel_initializer="he_uniform"))
model.add(tk.layers.Dense(1, activation='sigmoid',kernel_initializer="he_uniform"))
model.summary()
```

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

```
obj1=model.fit(x=x_train,y=y_train,epochs=50,batch_size=64,validation_data=(x_test,y_test))
```

```
y_pred=model.predict(x_test)
```

```
from sklearn.metrics import accuracy_score
accuracy=accuracy_score(y_test,y_pred.round())
```

```
accuracy
```