```
import numpy as np
import torch
from torch.utils.data import Dataset
from torch.utils.data import DataLoader
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import sklearn.model_selection
```

```
url1 = "https://raw.githubusercontent.com/gyandevgupta/DS203_Assignment/master/test-Assign8
url2 = "https://raw.githubusercontent.com/gyandevgupta/DS203_Assignment/master/train-Assign
df1 = pd.read_csv(url1, encoding='utf8')
df2 = pd.read_csv(url2, encoding='utf8')
df2
```

	Reviews	Sentiment
0	When I first tuned in on this morning news, I ...	neg
1	Mere thoughts of "Going Overboard" (aka "Babes...	neg
2	Why does this movie fall WELL below standards?...	neg
3	Wow and I thought that any Steven Segal movie ...	neg
4	The story is seen before, but that does'n matt...	neg
...	...	...
24995	Everyone plays their part pretty well in this ...	pos
24996	It happened with Assault on Prescient 13 in 20...	neg
24997	My God. This movie was awful. I can't complain...	neg
24998	When I first popped in Happy Birthday to Me, I...	neg
24999	So why does this show suck? Unfortunately, tha...	neg

25000 rows × 2 columns

```
df = pd.concat([df1,df2], ignore_index=True)
df
```

	Reviews	Sentiment
0	Who would have thought that a movie about a ma...	pos
1	After realizing what is going on around us	pos
2	I grew up watching the original Disney Cindere...	neg
3	David Mamet wrote the screenplay and made his ...	pos
4	Admittedly, I didn't have high expectations of...	neg

```
print("Before PreProcessing : \n")
print(df.iloc[1]["Reviews"])
import re
REPLACE_NO_SPACE = re.compile("(\.)|(\;)|(\:)|(\!)|(\')|(\?)|(\,)|(\")|(\()|(\))|(\[)|(\])"
REPLACE_WITH_SPACE = re.compile("(<br\s*/><br\s*/>)|(\-)|(\/)")
df["Reviews"] = [REPLACE_NO_SPACE.sub("", line.lower()) for line in df["Reviews"]]
df["Reviews"] = [REPLACE_WITH_SPACE.sub(" ", line) for line in df["Reviews"]]
print("After PreProcessing: \n")
df.iloc[1]["Reviews"]
```

```
Before PreProcessing :

After realizing what is going on around us ... in the news .. in our homes .. the whol
After PreProcessing:

'after realizing what is going on around us  in the news  in our homes  the whole new
i remembered this show and how obsessed i was watching it every week in my town  i sta
looking for this series  3 days ago  didnt have luck till this moment  and i was shock
en i read about it and about cbs  people i believe they stopped the show because its t
g about something way ahead of our understanding of the new world  it was trying to de
a hidden message about something terrifying  the people who stopped it are the same wh
```

```
df.iloc[1]["Reviews"]
```

```
'after realizing what is going on around us  in the news  in our homes  the whole new
i remembered this show and how obsessed i was watching it every week in my town  i sta
looking for this series  3 days ago  didnt have luck till this moment  and i was shock
en i read about it and about cbs  people i believe they stopped the show because its t
g about something way ahead of our understanding of the new world  it was trying to de
```

2. Split it into train and validation in the ratio 80:20

```
X = df.drop(["Sentiment"], axis=1)
Y = df["Reviews"]
X = np.array(X)
Y = np.array(Y)
X_train, X_val, y_train, y_val = sklearn.model_selection.train_test_split(X, Y, test_size=0
# X_validate, X_test, y_validate, y_test = sklearn.model_selection.train_test_split(X_rem,
X_train.astype(str)
print(X_train.dtype)
```

```
object
```

3. Install transformers library of huggingface

```
pip install transformers
```

```
    Requirement already satisfied: transformers in /usr/local/lib/python3.7/dist-packages
    Requirement already satisfied: filelock in /usr/local/lib/python3.7/dist-packages (fro
    Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.7/dist-packag
    Requirement already satisfied: sacremoses in /usr/local/lib/python3.7/dist-packages (f
    Requirement already satisfied: huggingface-hub<1.0,>=0.1.0 in /usr/local/lib/python3.7
    Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.7/dist-packages (f
    Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.7/dist-pac
    Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.7/dist-pack
    Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.7/dist-packages (
    Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (fro
    Requirement already satisfied: tokenizers<0.11,>=0.10.1 in /usr/local/lib/python3.7/di
    Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.7/dist-packages (
    Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-pack
    Requirement already satisfied: pyparsing>=2.0.2 in /usr/local/lib/python3.7/dist-packa
    Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (fr
    Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages
    Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-pac
    Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/l
    Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-pack
    Requirement already satisfied: joblib in /usr/local/lib/python3.7/dist-packages (from
    Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from sac
    Requirement already satisfied: click in /usr/local/lib/python3.7/dist-packages (from s
```

```
X_train_list = X_train.tolist()
X_val_list = X_val.tolist()
```

4. From transformers import DistilBertTokenizerFast tokenizer of pretrained model “distilbert-base-uncased”

```
from transformers import DistilBertTokenizerFast, DistilBertModel

tokenizer = DistilBertTokenizerFast.from_pretrained('distilbert-base-uncased')
X_train_encoded = []
X_val_encoded = []
# tokens = tokenizer.basic_tokenizer.tokenize(text)
# print("Tokens: ", tokens)
# X_train_list = X_train.tolist()
# X_val_list = X_val.tolist()
# text = "Replace me by any text you'd like."
# X_train_list = X_train.tolist()
# encoded_input_list = []
# output_list = []

# for i in range(0,40000):
#   encoded_input = tokenizer(X_train_list[i], return_tensors='pt', max_length=512, truncat
#   encoded_input_list.append(encoded_input)
#   output = model(**encoded_input)
#   output_list.append(output)
for i in range(0,40000):
  encoded_input = tokenizer(X_train_list[i], truncation=True, padding = True, return_tensor
  X_train_encoded.append(encoded_input)
for i in range(0,10000):
  encoded_input = tokenizer(X_val_list[i], truncation=True, padding = True, return_tensors
  X_val_encoded.append(encoded_input)
# X_train_encoded = tokenizer(X_train_list, truncation=True, padding = True)
# X_val_encoded = tokenizer(X_val_list, truncation=True, padding = True)
```

```
class MyDistilbert(Dataset):

  def __init__(self,data,labels) :
    self.data = data
    self.labels = labels

  def __getitem__(self,index) :
    X = self.data[index]
    Y = self.labels[index]
    return X,Y
    # item = {key: torch.tensor(val[index]) for key, val in self.data.items()}
    # item['labels'] = torch.tensor(self.labels[index])
    # return item

  def __len__(self):
    return len(self.labels)
```

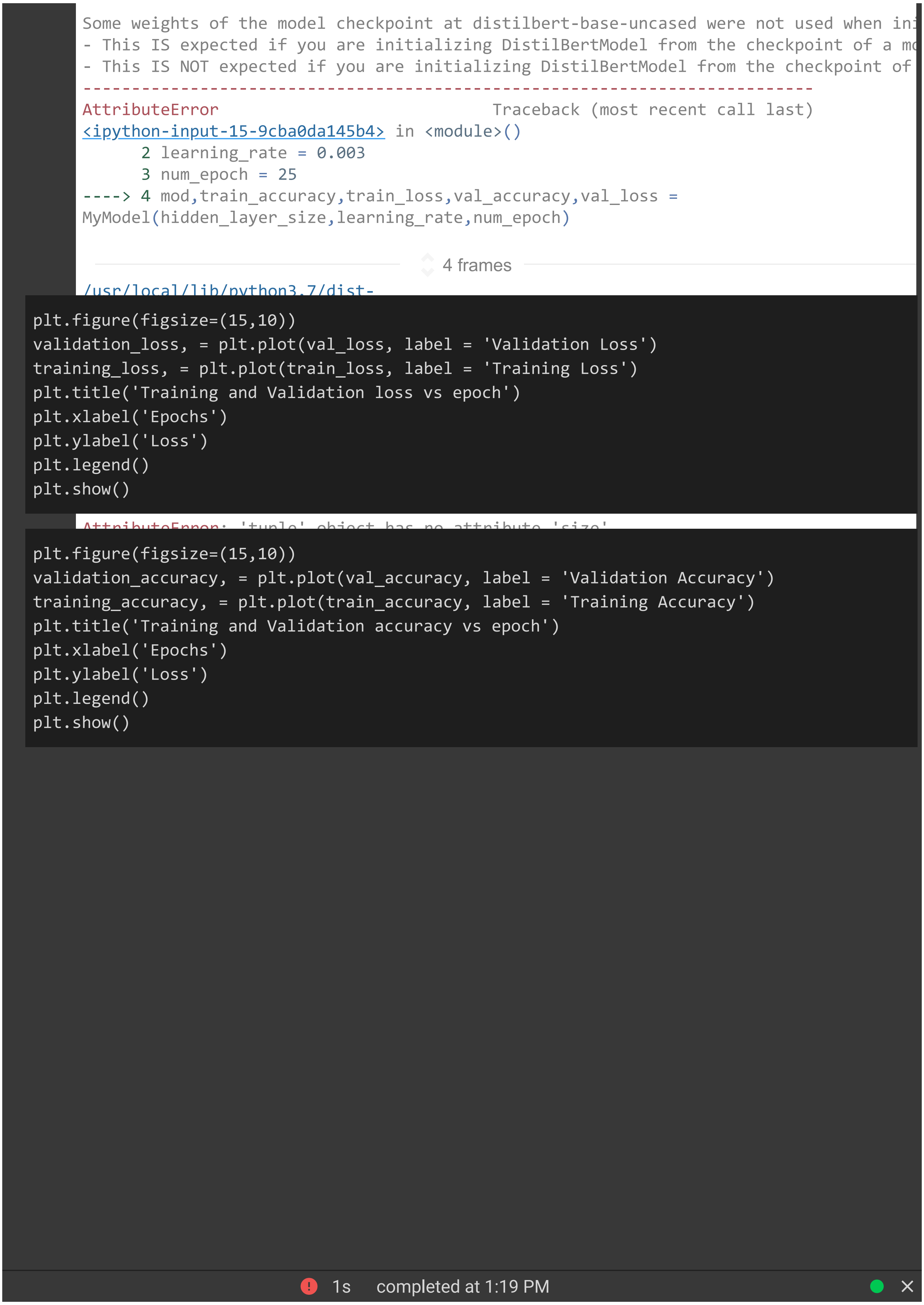
```
train_dataset = MyDistilbert(X_train_encoded,y_train)
val_dataset = MyDistilbert(X_val_encoded,y_val)
```

```
train_load = DataLoader(train_dataset, drop_last=False ,batch_size=128, shuffle=True , coll
val_load = DataLoader(val_dataset, drop_last=False, batch_size=128, shuffle=True, collate_f
```

```
class Network(torch.nn.Module):
  def __init__(self,hidden_layer_size = 4):
    super(Network,self).__init__()
    self.layer1 = DistilBertModel.from_pretrained("distilbert-base-uncased")
    self.dropout = torch.nn.Dropout(p=0.25)
    self.linear1 = torch.nn.Linear(2, hidden_layer_size)
    self.linear2 = torch.nn.Linear(hidden_layer_size, 1)

  def forward(self, x):
    x = self.layer1(x)
    x = self.dropout(x)
    x = torch.nn.functional.relu(self.linear1(x))
    out = torch.sigmoid(self.linear2(x))
    return out
```

```
def MyModel(hidden_layer_size=4, learning_rate=0.003, num_epoch = 25):
  model = Network(hidden_layer_size=4)
  loss_function = torch.nn.BCELoss(reduction='sum') # we would set mean for the loss
  optimizer = torch.optim.SGD(model.parameters(), learning_rate)

  train_loss = []
  val_loss = []
  train_accuracy = []
  train_accuracy = []
  val_accuracy = []

  for ind in range(num_epoch):
    tot_train_loss = 0
    tot_train_accuracy = 0
    tot_val_loss = 0
    tot_val_accuracy = 0

    for batch in train_load:
      batch_x = batch[0]
      batch_y = batch[1]
      model.train()
      y_pred = model(batch_x)
      loss = loss_function(np.array(y_pred),np.array(batch_y))/16
      tot_train_loss+=16*loss
      tot_train_accuracy+= ((y_pred>0.5)== batch_y).sum()
      optimizer.zero_grad()
      loss.backward()
      optimizer.step()

    train_accuracy.append(tot_train_accuracy.detach().numpy()/400)
    train_loss.append(tot_train_loss.detach().numpy()/40000)

    with torch.no_grad():
      model.eval()

      for batch in val_load:
        batch_x = batch[0]
        batch_y = batch[1]
        y_pred_val = model(batch_x.float())
        loss = loss_function(y_pred_val.float(),batch_y.float())
        tot_val_loss+=loss
        tot_val_accuracy+= ((y_pred_val>0.5)== batch_y).sum()

      val_accuracy.append(tot_val_accuracy.detach().numpy()/100)
      val_loss.append(tot_val_loss.detach().numpy()/10000)
  return model,train_accuracy,train_loss,val_accuracy,val_loss
```

```
hidden_layer_size = 4
learning_rate = 0.003
num_epoch = 25
mod,train_accuracy,train_loss,val_accuracy,val_loss = MyModel(hidden_layer_size,learning_ra
```



```
Some weights of the model checkpoint at distilbert-base-uncased were not used when ini
- This IS expected if you are initializing DistilBertModel from the checkpoint of a mo
- This IS NOT expected if you are initializing DistilBertModel from the checkpoint of
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-15-9cba0da145b4> in <module>()
      2 learning_rate = 0.003
      3 num_epoch = 25
----> 4 mod,train_accuracy,train_loss,val_accuracy,val_loss =
MyModel(hidden_layer_size,learning_rate,num_epoch)

4 frames

/usr/local/lib/python3.7/dist-
```

```
plt.figure(figsize=(15,10))
validation_loss, = plt.plot(val_loss, label = 'Validation Loss')
training_loss, = plt.plot(train_loss, label = 'Training Loss')
plt.title('Training and Validation loss vs epoch')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
```

```
AttributeError: 'tuple' object has no attribute 'size'
```

```
plt.figure(figsize=(15,10))
validation_accuracy, = plt.plot(val_accuracy, label = 'Validation Accuracy')
training_accuracy, = plt.plot(train_accuracy, label = 'Training Accuracy')
plt.title('Training and Validation accuracy vs epoch')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
```

1s completed at 1:19 PM