```
import torch
import torch.nn as nn
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

######### Loading the dataset ############
df = pd.read_csv('Linear Regression - Sheet1.csv')

x = torch.tensor(df.X.to_numpy(), dtype=torch.float32).reshape(300,1)
y = torch.tensor(df.Y.to_numpy(), dtype=torch.float32).reshape(300,1)

######### Normalising data ################
x_train = (x - x.mean())/x.std()
y_train = (y - y.mean())/y.std()

######### Defining model ###############
model = nn.Linear(1,1)

######### Defining the optimizer and cost function ############
learning_r = 0.001
cost = nn.MSELoss()
optimizer = torch.optim.Adam(model.parameters(), lr = learning_r)

########## Training the model ##################
epochs = 500
for i in range(epochs):
    y_p = model(x_train)
    loss = cost(y_p, y_train)

    loss.backward()
    optimizer.step()
    optimizer.zero_grad()

    if i%10==0:
        print(f'epoch{i} --> loss = {loss}')

########### Make predictions #################
predictions = model(x).detach().numpy()

########### Plotting the results #############
fig, ax = plt.subplots(figsize=(9,6))
ax.plot(x, y, 'ro', label='Original data')
ax.plot(x, predictions, label='Fitted line')
ax.legend()
plt.show()
```