```
load('djia.mat')
p = 3;
NN = 520;

% Defining X of the current Decade
for i = 1:NN-p
    for j = 1:p
        X(i,j) = djia(4342+i+j-1);
    end
end

% Defining x
for i = p+1:NN
    x(i-p,1) = djia(i);
end
% Solve for coefficients using X and x
a = -X\x;
```

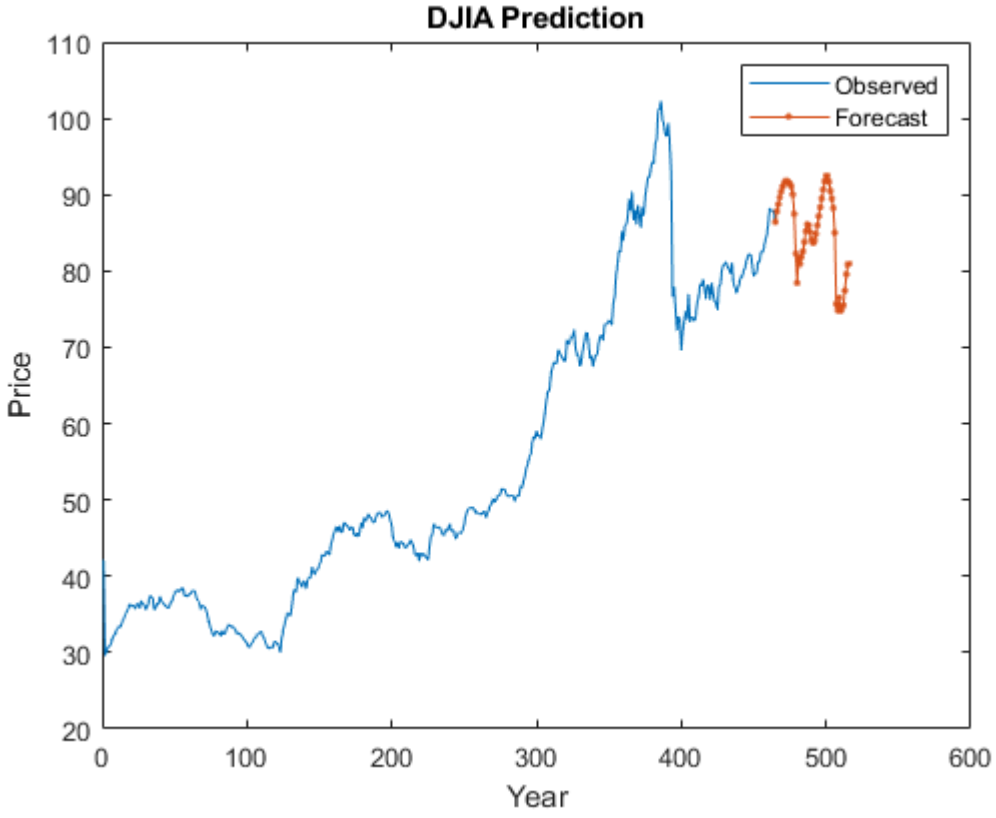
```
Warning: Rank deficient, rank = 4, tol =  4.316853e-09.
```

```
xhat1 = -X*a;
xhat2 = filter(-flip(a),1,djia(1:NN-p));

data = xhat1';

numTimeStepsTrain = floor(0.9*numel(data));

dataTrain = data(1:numTimeStepsTrain+1);
dataTest = data(numTimeStepsTrain+1:end);

mu = mean(dataTrain);
sig = std(dataTrain);

%Standardize the Data
dataTrainStandardized = (dataTrain - mu) / sig;

%Prepare Predictors and Responses
XTrain = dataTrainStandardized(1:end-1);
X = XTrain;
YTrain = dataTrainStandardized(2:end);
Y = YTrain;

numFeatures = 1;
numResponses = 1;
numHiddenUnits = 200;

layers = [ ...
    sequenceInputLayer(numFeatures)
    lstmLayer(numHiddenUnits)
    fullyConnectedLayer(numResponses)
    regressionLayer];

options = trainingOptions('adam', ...
    'MaxEpochs',500, ...
```

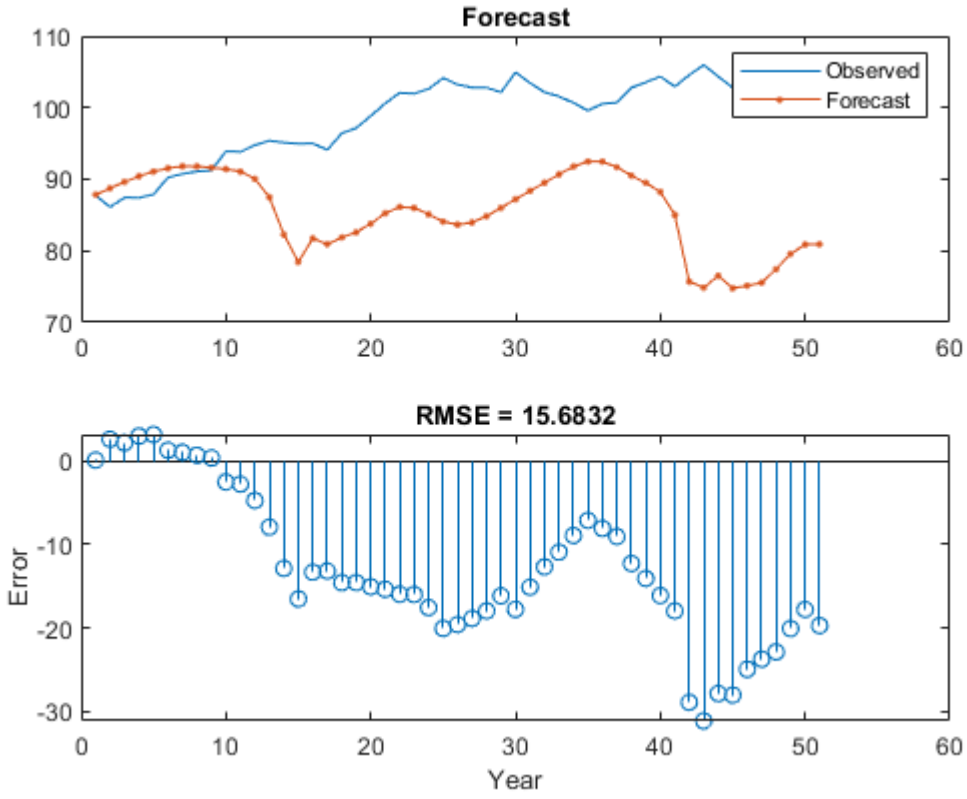
```
    'GradientThreshold',1, ...
    'InitialLearnRate',0.005, ...
    'LearnRateSchedule','piecewise', ...
    'LearnRateDropPeriod',125, ...
    'LearnRateDropFactor',0.2, ...
    'Verbose',0, ...
    'Plots','training-progress');

%Training LTSM
net = trainNetwork(X,Y,layers,options);

%Forecast Future Time Steps
dataTestStandardized = (dataTest - mu) / sig;
XTest = dataTestStandardized(1:end-1);

net = predictAndUpdateState(net,X);
[net,YPred] = predictAndUpdateState(net,Y(end));

numTimeStepsTest = numel(XTest);
for i = 2:numTimeStepsTest
    [net,YPred(:,i)] = predictAndUpdateState(net,YPred(:,i-1),'ExecutionEnvironment','cpu');
end

YPred = sig*YPred + mu;
YTest = dataTest(2:end);
rmse = sqrt(mean((YPred-YTest).^2))
```

```
rmse = single

    15.6832
```

```
figure
plot(dataTrain(1:end-1))
hold on
idx = numTimeStepsTrain:(numTimeStepsTrain+numTimeStepsTest);
plot(idx,[data(numTimeStepsTrain) YPred],'.-')
hold off
xlabel("Year")
ylabel("Price")
title("DJIA Prediction")
legend(["Observed" "Forecast"])
```




```
figure
subplot(2,1,1)
plot(YTest)
hold on
plot(YPred,'.-')
hold off
legend(["Observed" "Forecast"])
title("Forecast")

subplot(2,1,2)
stem(YPred - YTest)
xlabel("Year")
ylabel("Error")
title("RMSE = " + rmse(1))
```




```
xhat_u = max(YPred)/YPred(1);
upper_bound = 1000*xhat_u;
txt1 = ['Based on our prediction, in 520 weeks we will have ',...
    num2str(upper_bound)];
disp(txt1)
```

```
Based on our prediction, in 520 weeks we will have 1052.8615
```

```
lower_bound2 = 1000*YPred(end)/YPred(1);
txt2 = [' If we were to buy and hold, using $1000 we would yeild  ',...
    num2str(lower_bound2)];
disp(txt2)
```

```
 If we were to buy and hold, using $1000 we would yeild  921.3787
```

```
rr = ((YPred(end)/YPred(1))^(1/NN)-1)*52;
txt3 = ['To have the same, the APR rate should be ',num2str(rr*100),'%'];
disp(txt3)
```

```
To have the same, the APR rate should be -0.81887%
```

LSTM Analysis:

Observing the prediction results from the graph, we most likely did not have the most optimal prediction results, but it provided a decent prediction on the future results on DJIA. Based on the results we were given, we have to say the linear prediction had better results. It had a higher maximum total yield. Looking at the observed vs prediction, we were pretty off our marks in terms of predicting the actual target price. Looking at our Root

Mean Squared Error, towards the beginning of the predictions it was near accurate, but as the days went on, the prediction started swaying away and giving bad predictions.