COMP4432 Individual Assignment

LAI Peiheng 22098149d   Sem 2, 2025

Task 1: Insurance Charge Analysis

Nowadays, insurance charges has been a focus of analysis, influenced by various factors of beneficiary. In **Task 1**, we will apply **multiple linear regression** model to find relations between insurance *charges* and other relevant issues based on existing data, and make further prediction. Also, we will apply a **unsupervised** model to find some hidden pattern of information.

First we look at the **features** in given dataset. There are 6 **regressor features**: *age*, *sex*, *bmi*, *children*, *smoker*, *region*, and a **response** *charges*. The *age*, *bmi*, *children* are **numerical features** who may directly contribute to the linear regression. While *sex*, *smoker*, *region* are **categorical features**, and we may apply them as **indicator variable** by assigning values (e.g. assign smoker as 1 and non-smoker as 0) and participate linear regression.

We can give a general look on relations of each feature and calculate their **correlation**. We may observe from the heatmap shown as Figure 1 that *charges* has highest correlation with *smoker* (value 0.79), and large correlation with *age* (value 0.30) and *bmi* (value 0.20). Therefore we may visualize the data according to *smoker* later and expect there is some patterns, shown as Figure 2.

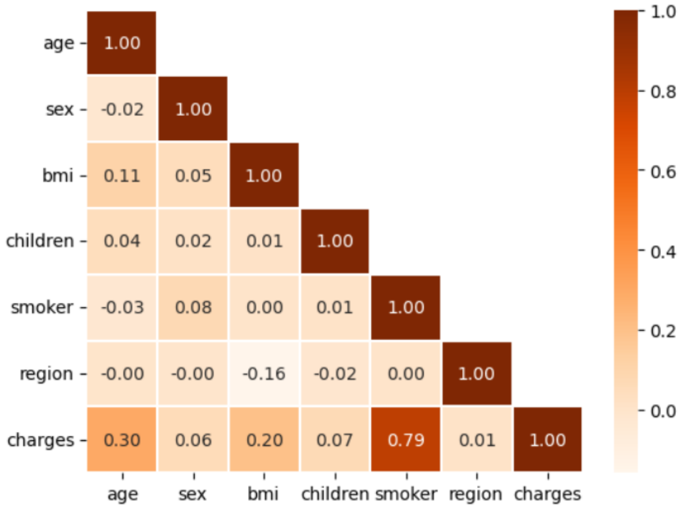

Figure 1: Correlation Heatmap of Features

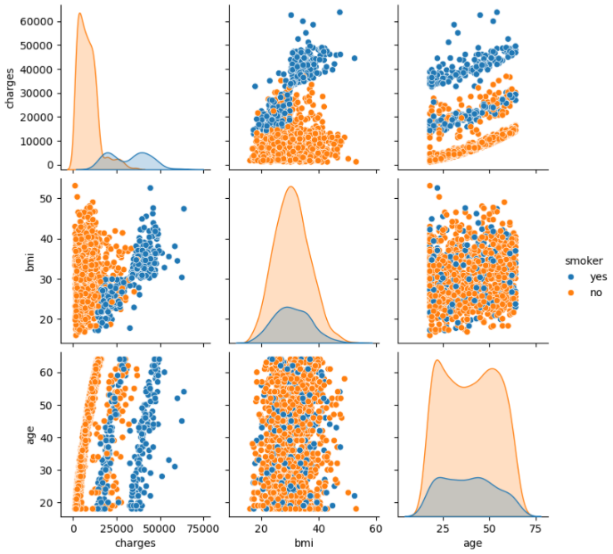

Figure 2: Pair Relation Class by *Smoker*

We can see prominent difference of data distribution identified by *smoker*, also we can make some detailed visualization on *region* difference, demonstrated as Figure 3,4,5:

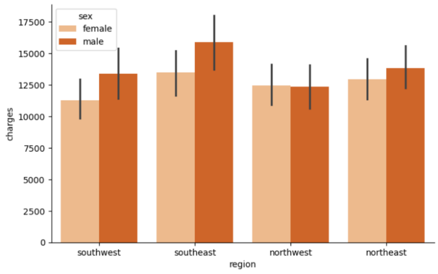

Figure 3: *Charges* Class by *Smoker*

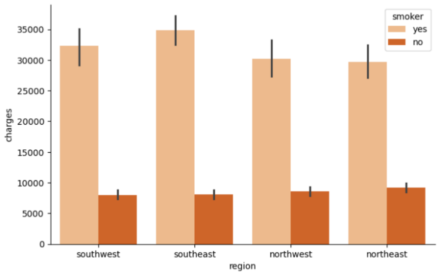

Figure 4: *Charges* Class by *Sex*

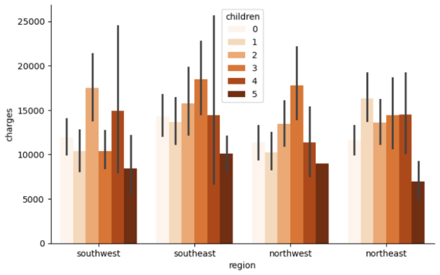

Figure 5: *Charges* Class by *Children*

Also, we can directly apply regression on numerical features categorized by *smoker* to observe its different influence on *age* and *bmi*, shown as Figure 6,7:

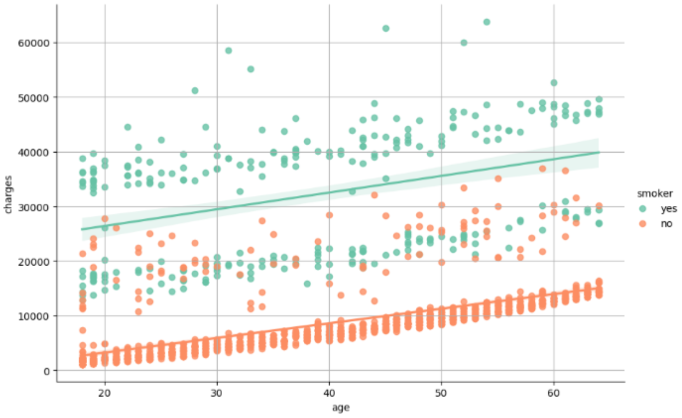

Figure 6: *Charges* vs *Age* Class by *Smoker*

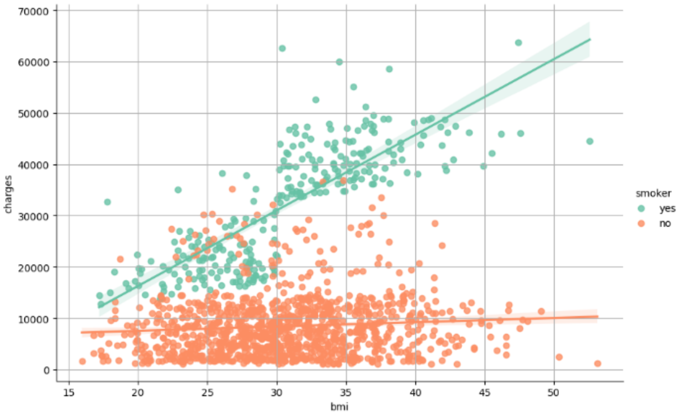

Figure 7: *Charges* vs *Bmi* Class by *Smoker*

We observe both smoker and non-smoker has almost the same **growing rate** of *charges* as their *age* increase, but smoker group has larger *charges* value. Meanwhile, smoker perform a more rapid growth on *charges* when *bmi* increases than non-smoker.

After assigning value to categorical feature, we fit our data to a **multiple linear model**. The fitted model is given by

$$\widehat{\text{charges}}_i = -12816 + 254\text{age}_i + 24\text{sex}_i + 328\text{bmi}_i + 444\text{children}_i + 23569\text{smoker}_i + 289\text{region}_i$$

with $R^2 = 0.799$, which is a relatively satisfying linear relation. We can see **whether smoking or not** contributes significantly to the insurance charges, while *sex* has little impact on the insurance charges. We also plot the predicted data against actual data for both train and test set to see general performance of our model. Shown as Figure 8, the prediction is pretty good at low *charges*, while higher estimated at middle *charges* and lower estimated at high *charges*.

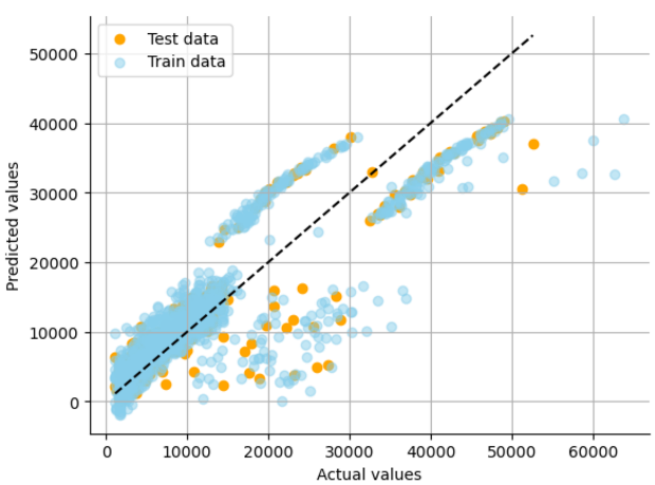

Figure 8: Prediction vs Actuality

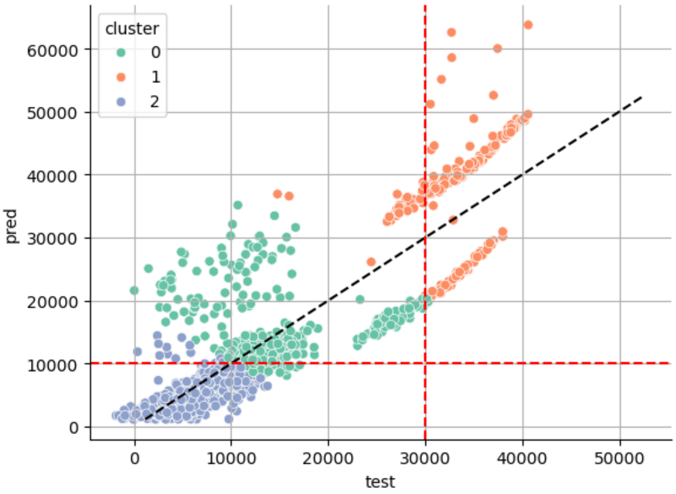

Figure 9: K-Mean Clustering of Prediction

To investigate more hidden pattern and give interpretation, we apply **K-mean clustering** to our prediction data point and divide them into **3 clusters**, illustrated in Figure 9. We calculate the mean for each feature in every cluster, giving

Cluster	***Age***	***Sex* Ratio**	***Bmi***	***Children***	***Smoker* Ratio**	***Region***	***Charges***
0	48.88	0.47	31.21	1.09	**0.15**	2.57	**14835**
1	42.07	0.60	32.69	1.16	**0.99**	2.45	**36720**
2	32.06	0.50	29.69	1.08	**0.00**	2.44	**5101**

Table 1: Mean and Ratio for each Cluster

It is spectacular to find that for people in cluster 1 with highest insurance charges, 99% **are smoker!** While for people in cluster 3 with lowest insurance charges, **non of them are smoker**! Others features are almost the same in each cluster.

It seems promising and prominent that smoking is significantly influencing people's health as well as their health insurance charge, showing the success of our machine learning work. Though, this model has several **limitations**. Standardization and scaling technique have not been used in this work, which may lead to some data bias or slow training speed. Also, the estimate bias in cluster 0 and 1 is not well-explained

Task 2: Stock Price Prediction

Stock price prediction using machine learning has been a great interesting of financial firms. In **Task 2**, we will implement several machine learning model to make prediction based on existing stock data from Google and Microsoft.

The data we use is essentially **time series data**, where feature values are **ordered by time**. We first plot the *Adj Close* value, which will be our only focus, and give a general look in Figure 10:

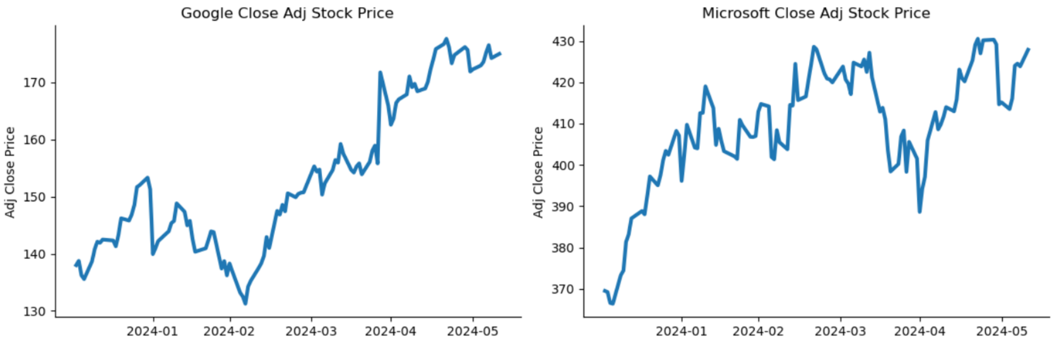

Figure 10: Stock Price of Two Firms

In Stock price analysis, we should also consider the **moving average (MA)**[1] of the price within a specific interval, shown in Figure 11:

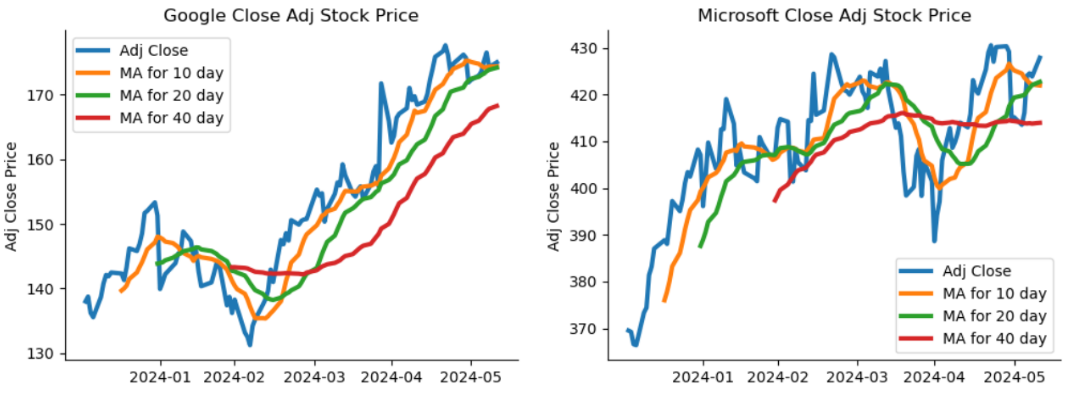

Figure 11: Stock Price Moving Average of Two Firms

We regard the **MA for 10 day** are most capable of showing tendency of origin data, while others may miss some small but important changes. The first model we want to use is the general **Linear Regression (LR)**[5]. Denoting the *Adj Close* data as P_t at specific time point t. We construct the linear regression model[2] to be

$$h_\theta(P) = \theta_0 + \theta_1 P_{t-1} + \theta_2 \mathrm{MA}_{t-1}$$

where

$$\mathrm{MA}_t = \frac{1}{10} \sum_{i=0}^{10} P_{t-i}$$

The fitted model and prediction result for Google and Microsoft stock are shown in Figure 12 and 13 respectively.

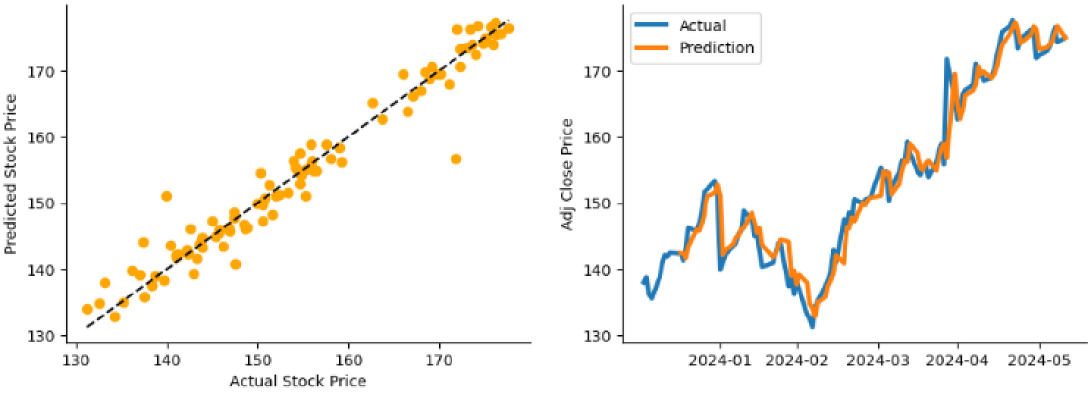

Figure 12: Google Stock Price Prediction by LR

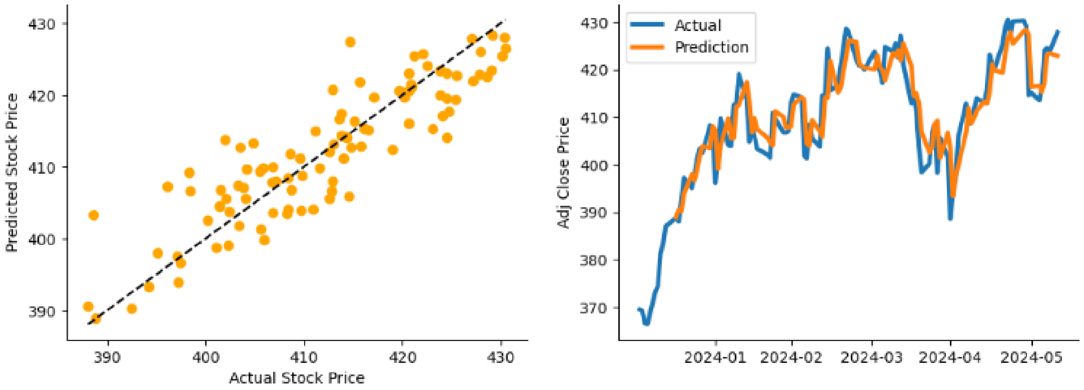

Figure 13: Microsoft Stock Price Prediction by LR

The Linear Regression performance is given by

Company	**Coefficients**	R^2	**MSE**
Google	$\theta_0 = 0.00,\ \theta_1 = 0.22,\ \theta_2 = 0.78$	0.96	7.68
Microsoft	$\theta_0 = 46.11,\ \theta_1 = 0.13,\ \theta_2 = 0.76$	0.79	23.43

Table 2: Performance of Linear Regression Model

where we use R^2 to evaluate the linear relation and **mean square error (MSE)** to examine the performance of prediction. We can see that data from Google fit a nearly perfect linear model, due to **small vibration** of its stock price. On the other hand, we have noticed that stock price of Microsoft have more **severe vibration** thus it indeed should be more unpredictable. Also, we see that MA has a larger contribution on the prediction, meaning stock in a certain past interval could also indicates the variation tendency.

The second machine learning model we would like to apply is the simple **neural network (NN)**[4]. The propagation from a_j^{l-1} (j^{th} neuron in $(l-1)^{\text{th}}$ layer) to a_j^l in NN can be expressed as

$$a_j^l = \sigma\left(\sum_k w_{jk}^l a_k^{l-1} + b_j^l\right)$$

where w is weight, b is bias and σ is the activation function. We build the NN model by setting **RELU** as our activation function and **50,30 neurons** in first and second hidden layer respectively. The prediction result is plotted in Figure 14:

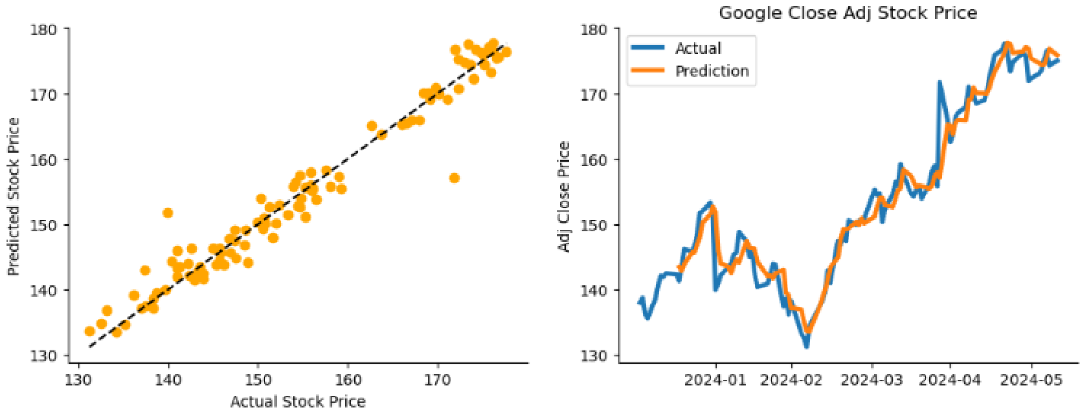

Figure 14: Google Stock Price Prediction by Simple NN

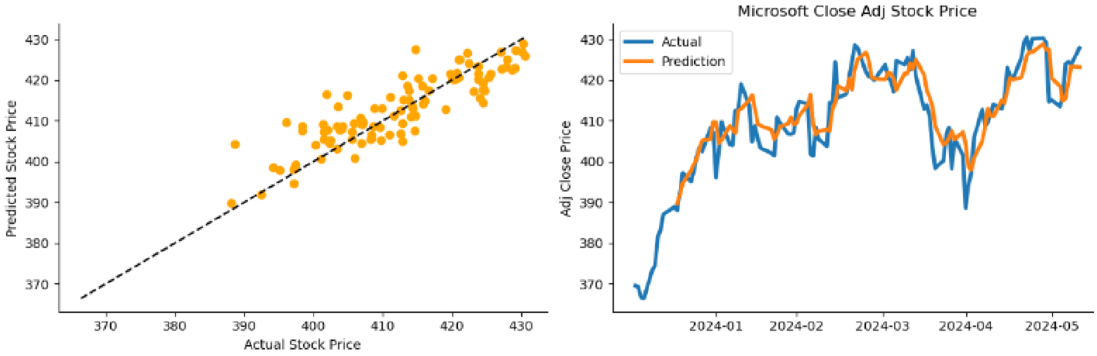

Figure 15: Microsoft Stock Price Prediction by Simple NN

The MSE for two firms are $\mathbf{MSE(Google)} = \mathbf{7.66}$ and $\mathbf{MSE(Microsoft)} = \mathbf{26.71}$

Since simple NN is a general model, and has no advantage on time series prediction. Thus we introduce **Long short-term memory (LSTM)**[3] as our third model, which is a frequent used machine learning model on time series prediction. A general framework of LSTM is demonstrated in Figure 16:

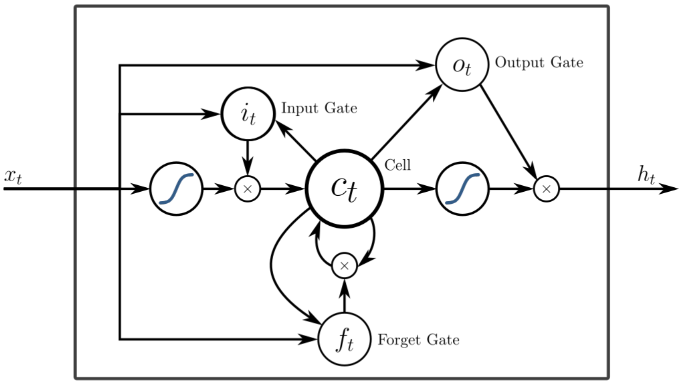

Figure 16: Network Framework of LSTM[6]

The core idea of LSTM modified from **recurrent neural network (RNN)** is the **forget gate** f_t:

$$f_t = \sigma(W_f \cdot x_t + U_f \cdot h_{t-1} + b_f)$$

where W_f, U_f are weight, b_f is bias, x_t is current input and h_{t-1} is the last hidden layer. Meanwhile, the **cell state** C_t is updated by

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C_t}$$

where $\tilde{C_t}$ is the candidate cell state. Implementing LSTM by **Pytorch**, we set the number of hidden layer to be 50, and plot the prediction results with the actual value as Figure 17,18:

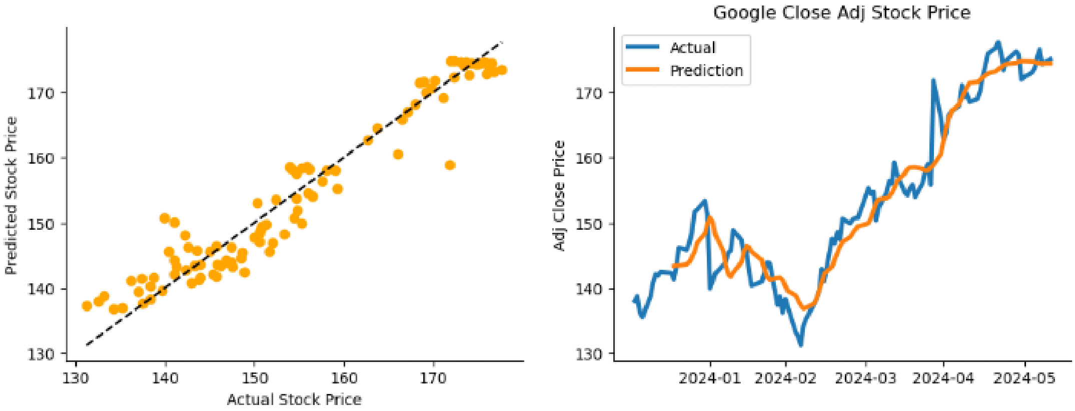

Figure 17: Google Stock Price Prediction by LSTM

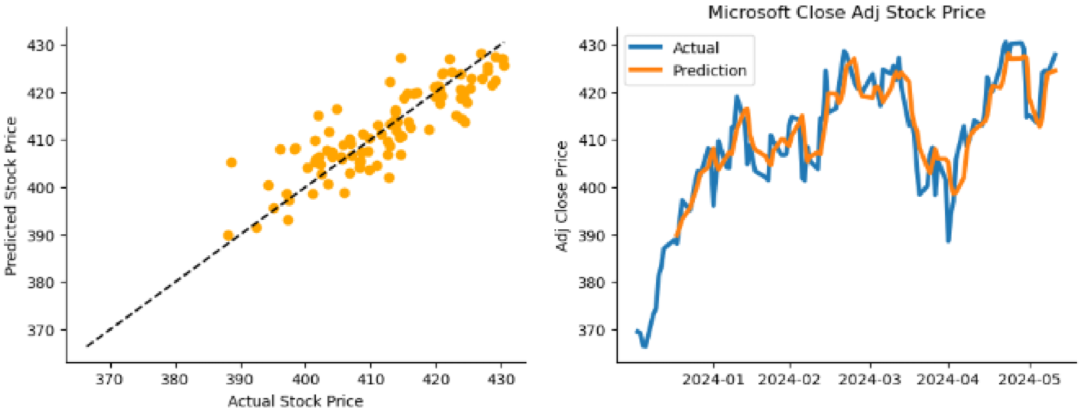

Figure 18: Microsoft Stock Price Prediction by LSTM

The MSE for two firms are $\mathbf{MSE(Google) = 11.17}$ and $\mathbf{MSE(Microsoft) = 43.89}$

Finally, we may compare the performance of three models applied according to MSE.

Company	LR	Simple NN	LSTM
Google	7.68	7.66	11.7
Microsoft	23.43	26.71	43.89

Table 3: Model MSE Summary

We consider better performances of LR and simple NN compared to LSTM are attributed to **heavy over-fitting**, due to not rigorous testing and training approach, and complex model (especially simple NN).

Though the current work and prediction seems satisfying and successful, there remain numerous limitations. Our prediction work use great proportion of data to only predict value of next time point, while accurate prediction of next **long time sequence** is difficult to achieve. This is due to the lack of **model complexity** and **feature completeness**. Since we only focus on *Adj Close* stock price in this work, we may consider **extra feature** such as *daily return*, *volume* to reach a more promising and long-period prediction in reality. Moreover, we also need to consider various external factors in real prediction, such as stock price of other firms and tremendous events.

(The implementation code can be found at **https://github.com/LPHQaq/COMP4432**)

THE END

References

[1] Md Masum Billah, Azmery Sultana, Farzana Bhuiyan, and Mohammed Golam Kaosar. Stock price prediction: comparison of different moving average techniques using deep learning model. *Neural Computing and Applications*, 36:5861–5871, 2024.

[2] Kevin Kam Fung Yuen. Towards regression model selection framework for stock price forecasting. In *TENCON 2024 - 2024 IEEE Region 10 Conference (TENCON)*, pages 1672–1675, 2024.

[3] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber. Learning precise timing with lstm recurrent networks. *Journal of Machine Learning Research*, 3(1):115–143, 2003.

[4] Alberto Prieto, Beatriz Prieto, Eva Martinez Ortigosa, Eduardo Ros, Francisco Pelayo, Julio Ortega, and Ignacio Rojas. Neural networks: An overview of early research, current frameworks and new challenges. *Neurocomputing*, 214:242–268, 2016.

[5] J. Margaret Sangeetha and K. Joy Alfia. Financial stock market forecast using evaluated linear regression based machine learning technique. *Measurement: Sensors*, 31:100950, 2024.

[6] Eddie Antonio Santos. Peephole long short-term memory, 2017. Accessed: 2025-03-12.