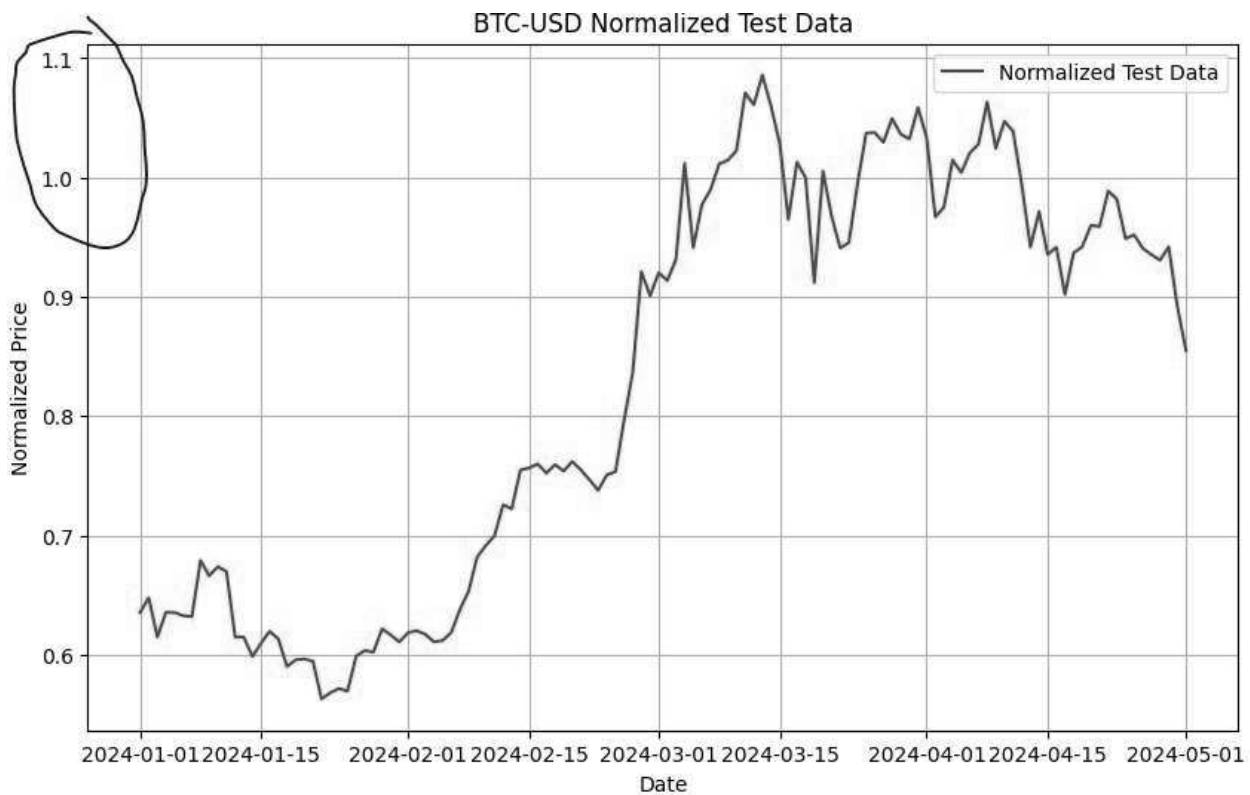


توضیح کد (قسمت a تا h)

ابتدا کتابخانه yahoo finance را نصب کردیم و تمام کتابخانه های مورد استفاده را ایمپورت کردیم. در قسمت بعدی دیتای train , test را به صورت جداگانه دانلود کردیم و نمایش دادیم. و در مرحله بعد plot مربوط به هر دو را نمایش دادیم.

در مرحله بعد هر دو داده را نورمالایز کردیم. به این ترتیب که ابتدا آبجکتی از MinMaxScaler ساختیم و با استفاده از آن آبجکت یا scaler ابتدا با تابع fit_transform دیتای train را نورمالایز کردیم و بعد بر اساس آن و با استفاده از تابع transform دیتای تست را نورمالایز کردیم. و پلات هر دو را نمایش دادیم.



با توجه به اینکه مقدار ماکسیموم در نورمالایز کردن ۱ می باشد علت اینکه در اینجا 1.1 داریم این است که ما بر اساس دیتای train عملیات scale کردن را انجام دادیم و معیار ما نورمالایز شده دیتای train بوده است که در برخی مقادیر تست از حداکثر آن عبور کرده ایم.

در مرحله بعد باید دیتای خود برای `train, test` را آماده کنیم. این کار را با استفاده از تابع `create dataset` انجام دادیم. اساس کار تهیه دیتا به این صورت است که ابتدا دیتای نورمالایز شده و سپس مقدار `look back` را به آن پاس میدهم. منظور از `look back` پنجره زمانی است که ما در آن گذشته را در نظر میگیریم.

مقدار خروجی نیز همانند خروجی ای است که در داک سوالات آورده شده است. به این صورت است که پنجره را به صورت مثلا 1 تا 60 بعد 2 تا 61 و به همین صورت تا 1766 تا 1826 بوده). (تعداد دیتایی که برای `train` داشتیم 1826 بوده).

پس از تهیه دیتا نوبت به این مدل مشابه با شکل داده شده را طراحی کنیم. ابتدا به صورت `sequential` مدل را طراحی میکنیم سپس `lstm unit` ها و مقادیر `dropout` را در هر مرحله به آن اضافه میکنیم. همچنین مدل را با `Adam optimizer` و با تابع خطای ذکر شده در سوال کامپایل میکنیم. و در نهایت برای `fit` کردن مدل تعداد `epoch, batch` را نیز برای آموزش در نظر میگیریم. در این مرحله تعداد لایه ها را 4 در نظر میگیریم. ابتدا لایه اول را با مقادیر ورودی میسازیم و بعد دو لایه میانی یعنی لایه دوم و سوم را طراحی میکنیم و طبق شکل پیش میرویم و در نهایت لایه نهایی را میسازیم.

در قسمت بعدی دیتاست مربوط به دیتای تست را میسازیم سپس عملیات `predict` را انجام میدهم سپس برای اینکه به طور دقیق ببینیم چه مقادیری از قیمت واقعی را پیش بینی کرده ایم مقادیر `scale` شده را برعکس میکنیم تا به مقدار واقعی قیمت برسیم و سپس در انتها پلات مربوط به پیش بینی های خود و مقادیر واقعی قیمت را نمایش میدهم.

(i)

افزایش تعداد روز ها باعث میشود که تعداد بازه هایی که داریم کمتر شود ولی طول بازه ها طولانی شود به این ترتیب در مرحله آموزش مدل ما تعداد بازه های کمتری دارد که طبق آن آموزش ببیند و خود را بهبود بخشد اما اگر کاهش تعداد روزها را داشته باشیم باعث میشود طول

بازه ها کوتاه تر اما تعداد بازه ها افزایش یابد و مدل نمونه های بیشتری برای یادگیری داشته باشد.

به صورت کلی کاهش و کمبود دیتا میتواند مزایا و معایب زیر را داشته باشد:

- آموزش سریعتر ، کاهش محاسبات (مزایا)
- ریسک **overfitting** بالاتر میرود ، مدل در برابر دیتای پرت حساس تر است ، تعمیم پذیری کاهش میابد.(معایب)

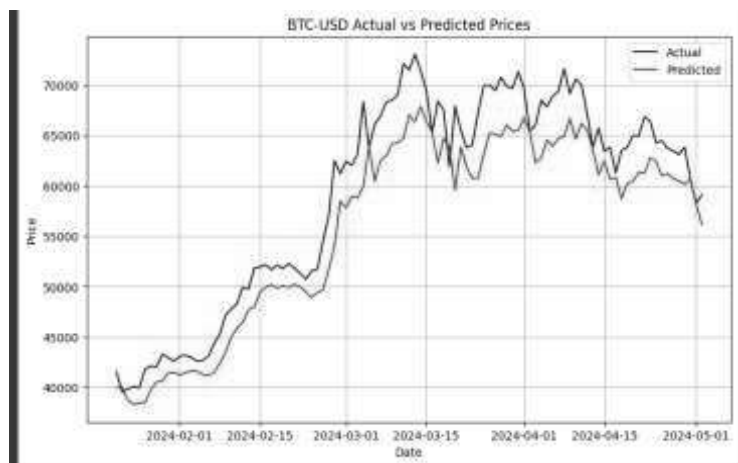
همچنین افزایش دیتا میتواند مزایا و معایب زیر را به همراه داشته باشد :

- بهبود تعمیم پذیری مدل ، **stability** بیشتر ، پیش بینی های دقیق تر به دلیل وجود دیتای بیشتر برای یادگیری (مزایا)
- هزینه محاسبات بالاتر میرود ، جمع آوری دیتا سخت تر میشود(در این مثال به دلیل وجود دیتا ممکن است مشکل نخوریم) ، مدل به حدی میرسد که در برابر دیتای جدید خیلی **sensitive** نباشد و اصلا به دنبال تغییر نباشد که اصلا مناسب نیست.(معایب)

تصاویر زیر نشان دهنده عملکرد بهتر با طول روز کمتر هستند :



به ازای **look back = 90**



به ازای $\text{look back} = 20$

که نشان میدهد مقدار پیش بینی ما به مقدار واقعی نزدیک تر است و فاصله کمتر میباشد.