



پروژه پایانی زنجیره مارکوف

سید سروش مجد

۴۰۰۴۴۳۱۸۱

درس یادگیری ماشین | تیر ۱۴۰۱

استاد درس: جناب آقای دکتر احمدعلی آبین

۳مقدمه
۳ معرفی مجموعه داده
۴ مدل پنهان مارکوف
۵ پیشبینی با داده‌های تست و نتایج مارکوف
۷ شبکه عصبی
۸ نتیجه‌های شبکه عصبی
۹ LSTM
۱۰ نتیجه LSTM
۱۰ نتیجه‌گیری و جمع‌بندی
۱۱ منابع و مراجع

مقدمه

حوزه استفاده از مدل پنهان مارکوف یا Hidden Markov Model وسیع است و در حوزه‌های تشخیص الگوهای مالی، داده‌های به صورت دنباله سری زمانی و هوش مصنوعی بسیار کاربرد دارد. در این تمرین تلاش می‌کنیم با استفاده از مدل پنهان مارکوف عمل پیشبینی بازار سهام بر روی مجموعه‌داده Tata-steel را انجام دهیم و دقت آن را با یک شبکه عصبی پیشنهادی مقایسه کنیم.

معرفی مجموعه‌داده

قبل از اینکه مدلمان را تولید کنیم یک دید کلی از مجموعه‌داده به دست آوریم. مجموعه‌داده ما از ۱۲ ستون تشکیل شده است. که عبارتند از Date, Low Price, High Price, Open Price, Total Turnover, Quantity, Deliverable, (Rs.), No. of Trades, No. of Shares, WAP, Close Price, Spread Close-Open و Spread High-Low. Deli. Qty to Traded Qty روز قرار گرفته‌اند و ۱۴۸۵ سطر داریم که هر کدام مربوط به اطلاعات سهام در هر روز می‌باشند. برای آموزش و پیشبینی از چهار ستون Low Price, High Price, Open Price و Close Price استفاده خواهیم کرد. هدف ما پیشبینی Close Price می‌باشد. ویژگی‌هایی با آن‌ها کار می‌کنیم به صورت زیر می‌باشند:

1. The fractional change in opening and closing prices
2. The fractional change in high prices
3. The fractional change in low prices

یا:

$$\frac{\text{Close Price} - \text{Open Price}}{\text{Open Price}}$$

$$\frac{\text{High Price} - \text{Open Price}}{\text{Open Price}}$$

$$\frac{\text{Open Price} - \text{Low Price}}{\text{Open Price}}$$

این‌ها به صورت جداگانه در مجموعه داده‌های آموزش و پیشبینی به دست می‌آیند.

مدل پنهان مارکوف

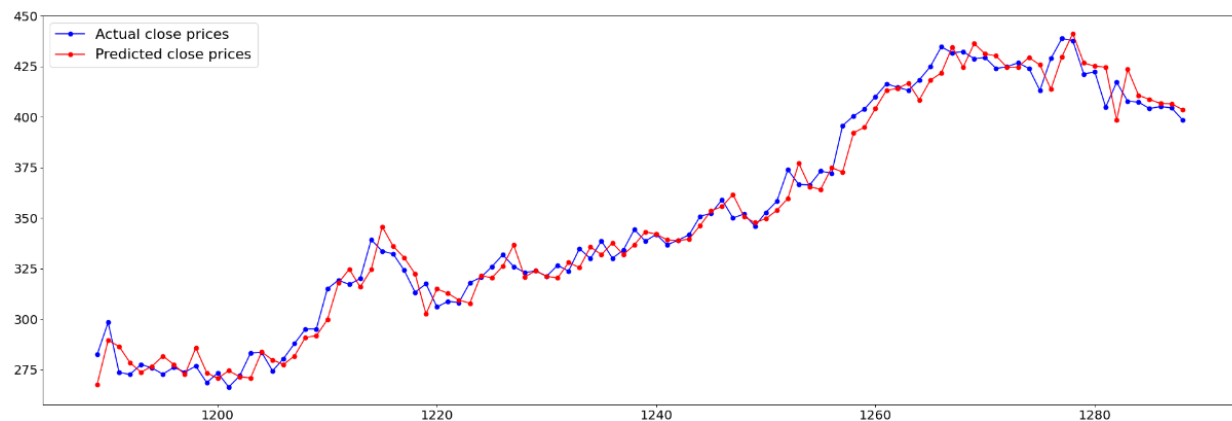
در مدل پنهان مارکوف یا HMM تعدادی state وجود دارد که احتمال اولیه قرارگیری در آن‌ها با π نمایش داده می‌شود. احتمال جابه‌جایی از حالت‌ها با ماتریس A و احتمال مشاهده‌ها در state با ماتریس B نشان داده می‌شوند.

برای پیاده‌سازی مدل از کتابخانه `hmmlearn` و مدل `Gaussian HMM` استفاده شد. فرض کرده ایم که دنباله‌های ما گوسی هستند یعنی ما برای مجموعه داده‌ها یک میانگین و واریانس گوسی چند متغیره داریم، ۱۰ حالت پنهان در نظر می‌گیریم (ممکن است بتوان جستجوی را در میان مجموعه‌ای از مقادیر ممکن برای تعداد حالت‌های پنهان انجام داد تا ببینیم کدام تعداد حالت بهتر است و هرچه تعداد state‌ها بیشتر باشد مدل قوی‌تر شده ولی احتمال بیش‌برازش بیشتر می‌شود) و به ازای هر کدام از آن‌ها بردار میانگین و ماتریس کوواریانس را ذخیره کردیم. برای آموزش از الگوریتم `Baum-Welch` که از `Expectation Maximization` بهره می‌برد استفاده و پارامترهای بهینه برای HMM محاسبه می‌شود.

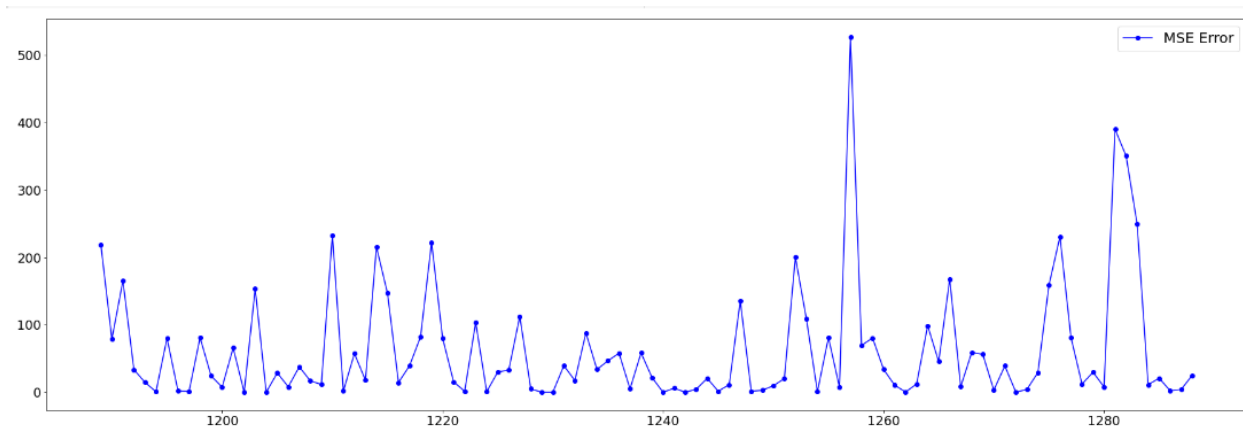
بعد از آموزش با استفاده از روش MAP یا Maximum A Posteriori عمل پیشبینی را انجام دادیم. با ورود d روز به مدل مارکوف مقدار Close Price را برای روز d+1 محاسبه می‌کنیم. (Fractional change in opening and closing price را تخمین می‌زنیم). مقادیر ممکن برای فیچرها به جای آنکه continuous در نظر گرفته شود برای فیچر اول (The fractional change in opening and closing prices) مقدار بین -۰.۱ و +۰.۱ و ۱۰ مقدار بین ۰ و ۰.۱ برای فیچر دوم و سوم (The fractional change in low prices) و (change in high prices) در نظر گرفته شده است. سپس داده تست را برای d روز متوالی می‌گیریم و score برای هر کدام از مقادیر ممکن برای خروجی را محاسبه می‌کنیم و می‌بینیم کدام امتیاز بیشتری می‌گیرد. خروجی که بیشترین امتیاز را بگیرد برای پیشبینی Closed Price استفاده می‌شود. مقدار d در کد ما عدد ۵۰ در نظر گرفته شده است.

پیشبینی با داده‌های تست و نتایج مارکوف

نمودار آبی نمایانگر قیمت واقعی close و نمودار قرمز قرمز نمایانگر قیمت پیشبینی شده می‌باشد با ۸۰ درصد آموزش و ۲۰ درصد تستو تعداد ۱۰ state:

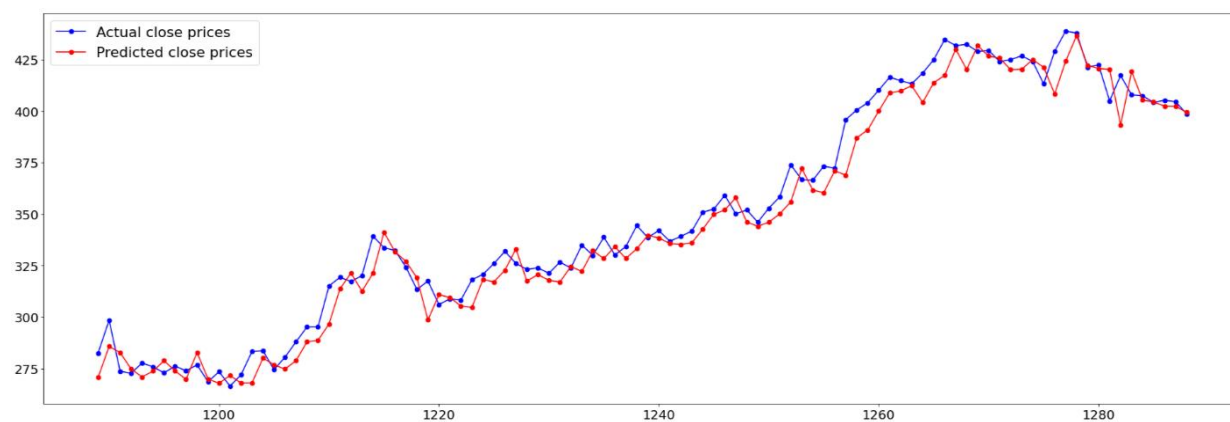


نمودار زیر توان دو خطا برای هر سطر داده را نمایش می‌دهد.

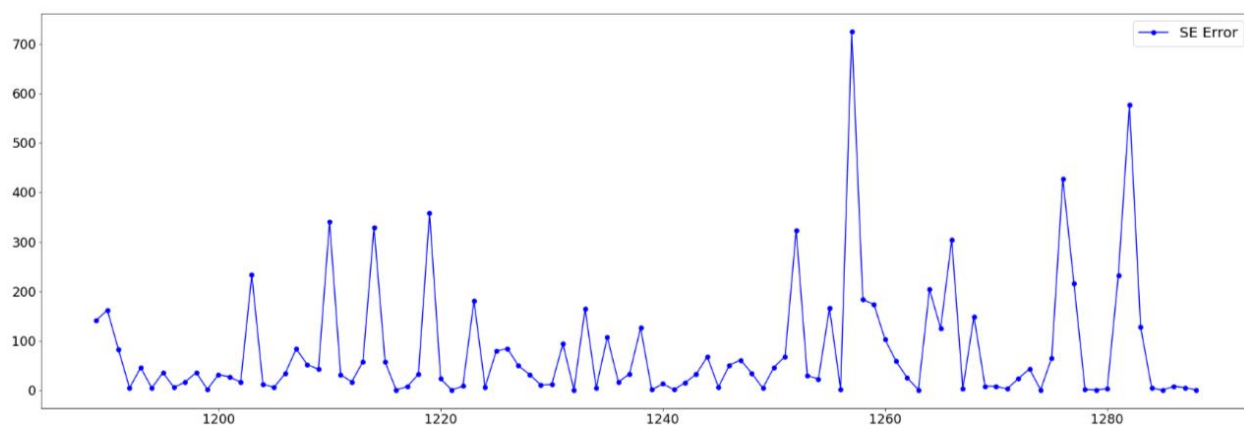


MSE یعنی میانگین نمودار بالا برای مدل مارکوف برابر با ۶۲.۴۴ به دست آمد.

نمودار آبی نمایانگر قیمت واقعی close و نمودار قرمز نمایانگر قیمت پیشبینی شده می باشد با ۸۰ درصد آموزش و ۲۰ درصد تستو تعداد ۲ state:

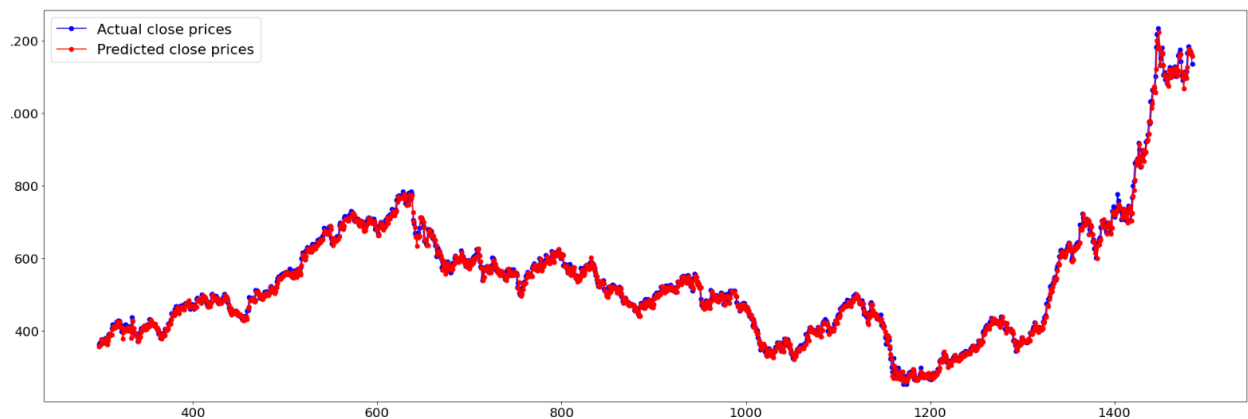


نمودار زیر توان دو خطا برای هر سطر داده را نمایش می دهد.

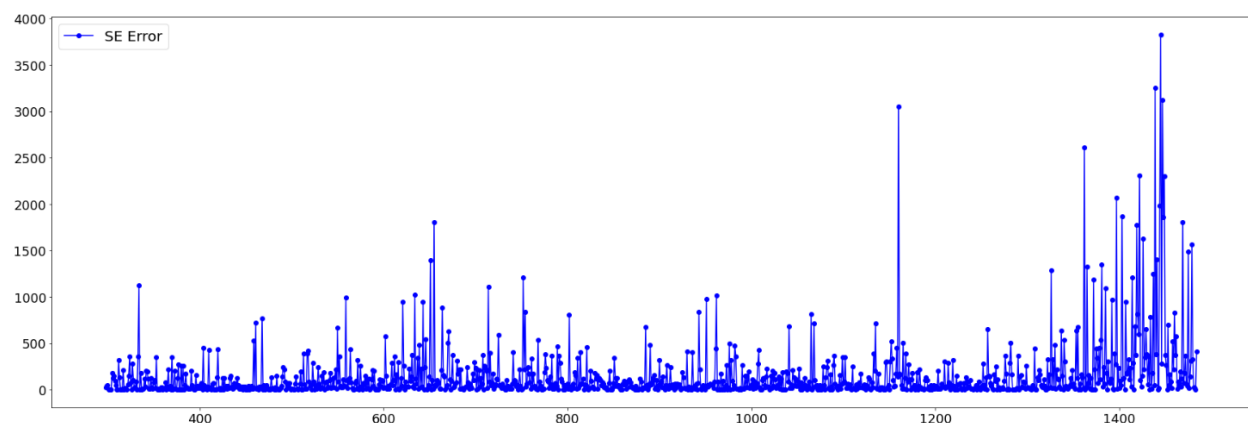


MSE یعنی میانگین نمودار بالا برای مدل مارکوف برابر با ۸۰.۷۹ به دست آمد.

سپس به جای آنکه با ۸۰ درصد آموزش دهیم با ۲۰ درصد آموزش را شروع کردیم و ۸۰ درصد را به عنوان داده تست در نظر گرفتیم و نتایج زیر بدست آمد. نمودار آبی نمایانگر قیمت واقعی `close` و نمودار قرمز نمایانگر قیمت پیشبینی شده می باشد.



نمودار زیر توان دو خطا برای هر سطر داده را نمایش می دهد.



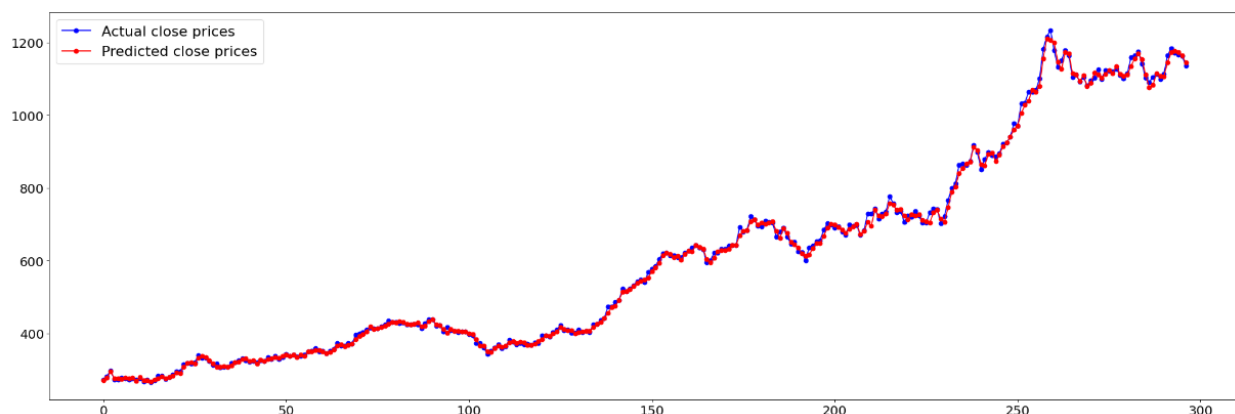
MSE یعنی میانگین نمودار بالا برای مدل مارکوف برابر با ۱۴۹.۸۳ به دست آمد.

شبکه عصبی

شبکه عصبی انتخاب شده متشکل از یک لایه مخفی ۴۰ نورونه با تابع فعالساز رلو، لایه آخر تک نورونه با تابع فعالساز خطی و می باشد و بهینه ساز مورد استفاده Adam می باشد. با استفاده از این شبکه در واقع می خواهیم مسئله رگرسیون حل کنیم. برای پیاده سازی این مدل از کتابخانه `sklearn` استفاده شده است.

نتیجه‌های شبکه عصبی

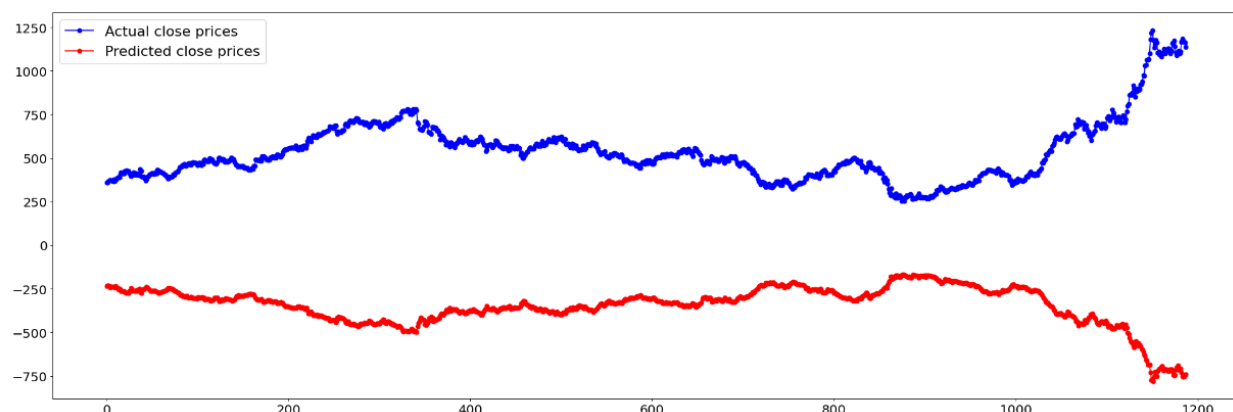
نمودار آبی نمایانگر قیمت واقعی $close$ و نمودار قرمز نمایانگر قیمت پیشبینی شده می‌باشد. ۲۰ درصد داده تست و ۸۰ درصد آموزش در نظر گرفته شده است.



همانطور که مشاهده می‌شود با استفاده از شبکه عصبی پیشنهادی داده‌های تست و پیشبینی شده در شکل بالا بسیار نزدیک به یکدیگر هستند.

MSE برابر با ۸۲.۲۵ به دست آمد.

سپس به جای آنکه با ۸۰ درصد آموزش دهیم با ۲۰ درصد آموزش را شروع کردیم و ۸۰ درصد را به عنوان داده تست در نظر گرفتیم و نتایج زیر بدست آمد. نمودار آبی نمایانگر قیمت واقعی $close$ و نمودار قرمز نمایانگر قیمت پیشبینی شده می‌باشد.



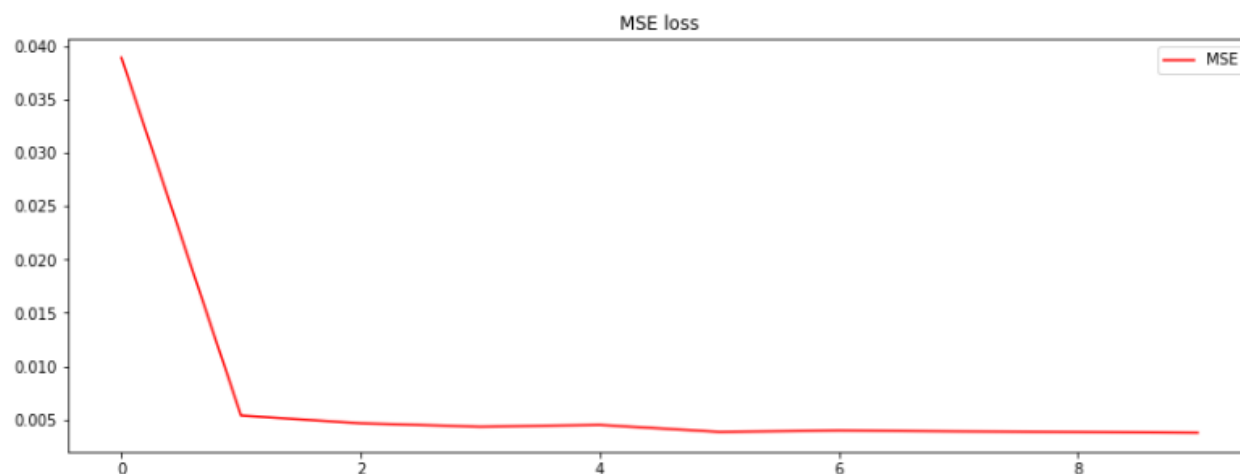
MSE برابر با ۲۶۸۱۳۲.۶۵ به دست آمد.

LSTM

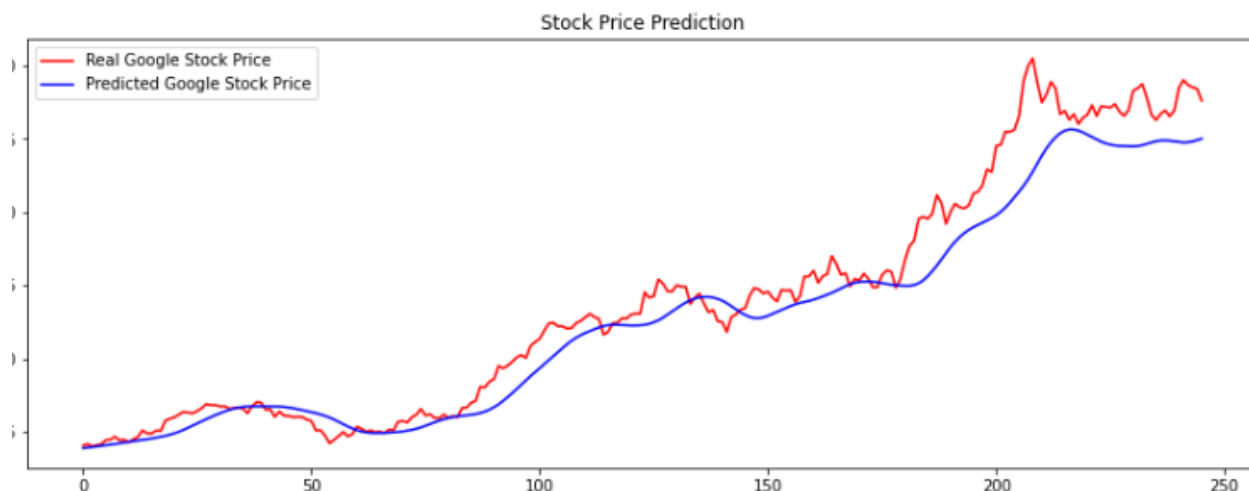
برای پیاده‌سازی از کتابخانه `sklearn` استفاده کردیم. و قبل از اینکه داده‌ها را به شبکه بدهیم با `minmax scaler` مقدار فیچرها را نرمالایز کردیم. در شبکه‌ها افزایش تعداد لایه‌ها و نورونها باعث میشود شبکه پیچیده‌تر شده و بتواند پترن‌های پیچیده‌تری یاد بگیرد و برای جلوگیری از بیش برآزش و فیت نشدن شبکه بر روی اطلاعات `outlier` از `drop out` استفاده کردیم تا تعمیم‌پذیری را نیز افزایش دهیم. در این بخش مدلی برای پیشبینی داده‌های بورس با استفاده از واحدهای LSTM به صورت پشته‌های برای بازه زمانی ۵۰ روزه پیاده‌سازی شد. الگوریتم بهینه‌سازی، تابع فعالیت و تابع زیان به ترتیب از الگوریتم `Adam`، تابع `tanh` و تابع `MSE` استفاده شد. معماری استفاده شده:

Layer (type)	Output Shape	Param #
lstm_27 (LSTM)	(None, 50, 60)	14880
dropout_27 (Dropout)	(None, 50, 60)	0
lstm_28 (LSTM)	(None, 50, 70)	36680
dropout_28 (Dropout)	(None, 50, 70)	0
lstm_29 (LSTM)	(None, 100)	68400
dropout_29 (Dropout)	(None, 100)	0
dense_12 (Dense)	(None, 1)	101
Total params: 120,061		
Trainable params: 120,061		
Non-trainable params: 0		

آموزش شبکه:



نتیجه LSTM



نتیجه گیری و جمع بندی

داده های بورسی داده هایی سری زمانی هستند و قیمت های روزهای قبلی بر روز های بعدی تاثیر گذار هستند. می دانیم که شبکه های عصبی عادی نمی توانند داده های سکونسی را دقیق پیش بینی کنند. در مرحله اول وقتی ۸۰ درصد داده ها را به عنوان داده آموزش و ۲۰ درصد آن ها را به عنوان تست در نظر گرفتیم مشاهده کردیم MSE برای مدل مارکوف برابر با ۶۲.۴۴ و برای مدل شبکه عصبی برابر ۸۲.۲۵ بود و به نظر می رسید شبکه عصبی خیلی خوب عمل کرده و به مدل مارکوف نزدیک شده است. ولی این به این دلیل بوده که شبکه عصبی ویژگی های خاصی را از داده ها یاد گرفته که به نظر من اصلا قدرت تعمیم پذیری ندارد. در مرحله بعد ۲۰ درصد از داده ها را به عنوان آموزش و ۸۰ درصد را به عنوان تست در نظر گرفتیم و مشاهده کردیم که شبکه عصبی جواب کاملا پرتی را نتیجه می داد ولی شبکه مارکوف بسیار بهتر از شبکه عصبی عمل کرد و با اینکه مقدار MSE برابر با ۱۴۹.۸۳ شد و بیشتر شد ولی مشخص است که مدل مارکوف با توجه به ماهیتش در این مجموعه داده قدرت تعمیم پذیری بسیار بیشتری نسبت به شبکه عصبی دارد. همچنین شبکه LSTM که ماهیت دنباله ای و بازگشتی دارد نیز پیاده سازی شد ولی مدل مارکوف از این شبکه هم بهتر عمل کرد. همچنین مشاهده کردیم با کاهش state ها در مدل مارکوف از دقت نتایجمان کاسته شد و مقدار loss کمی بیشتر شد در کل می توان گفت برای این مجموعه داده hmm بهترین نتیجه را داده است و حتی از شبکه بازگشتی قوی LSTM نیز بهتر عمل کرده است.

منابع و مراجع

https://users.cs.duke.edu/~bdhingra/papers/stock_hmm.pdf

لینک کدها:

مارکوف: در فایل ها موجود است.

شبکه عصبی:

https://colab.research.google.com/drive/1dW1lQOiV_C447cr4vU6B6VMI4K0rJubB#scrollTo=eBQQ5BqHXn9S

:LSTM

https://colab.research.google.com/drive/1_PTPK8RTleEKMf4Tv4fFU7hgbK3D-Fxh#scrollTo=tAWHEVYiu402&uniqifier=3