Capstone Proposal

Inspiration:

In the course Machine Learning Case studies, I learned how to use DeepAR to predict power usage. I would like to apply this technique to the finance world. Stock price data are also time series and can be analyzed using similar methods. As a data science student, I was very enthusiastic to try different machine learning algorithms and answer the question: can machine learning be used to predict stock market movement?

Domain Background：

Investment firms and hedge funds have been using financial models to better understand market behavior and make profitable investments and trades. A wealth of information is available in the form of historical stock prices data, suitable for machine learning algorithms to process. Machine learning has been used for identifying complex trading patterns across different markets in real-time. It takes advantage of its high speed and big data processing power to give real-time updates. That explains why it is used in wall street and the United States in general. AI has been used in many industries including trading. Electronic trades account for almost half of the total revenues from cash equity trading. Most companies such as hedge funds, use AI-powered analysis to get investment ideas and build portfolios. AI makes it easy to analyze data and execute trades at the best price. Through AI, analysts can be able to forecast markets with more accuracy. It is also easier for traders and exchange firms to efficiently mitigate risks for higher returns. In this project, I am going to use neural network to predict stock close prices.

Problem Statement：

Using Neural Network to predict daily stock price values. More specifically, we will train the model using data of 2014, 2015, 2016 and use the trained model to predict close price (represented by rate of change) in 2017.

Benchmark Model:

Time series forecasting can be solved by many algorithms. Classical forecasting methods, such as autoregressive integrated moving average (ARIMA) or exponential smoothing (ETS), fit a single model to each individual time series. They then use that model to extrapolate the time series into the future. Multiple linear regression can also be used to predict the trends. Multiple technical indicators (RSI, MACD, Bollinger Bands…) can be used as the input features. However, those models can only capture a general trend. In order to obtain more accurate prediction, deep learning and neural network should be good choices. Nowadays, with greater computational capacity and huge amount of trading data, deep learning seems to become more suitable in time series prediction. In this project, I am going to apply Recurrent Neural Network (RNN) in stock price prediction.

Datasets and Inputs:

Apple stock price data from 2013 to 2018 (Apple 2013-2018.csv). Inputs will be the rate of change (ROC) of close price.

Solution Statement:

Obtain the exact ROC prediction value for a given period of time in the future using DeepAR algorithm. ROC is calculated as the following, where r_t is ROC at time t, p_t is close price at time t and p_{t-1} is close price at time t-1.

$$r_t = \frac{p_t - p_{t-1}}{p_{t-1}}$$

We will use ROC as input data and the output prediction will also be ROC. Once we have ROC for every points, the actual close price can be recovered as following:

$$p_t = (1 + r_t)p_{t-1}$$

Evaluation Metrics:

Plots: An intuitive comparison between predictions and actual targets. From the plot, we can see if our prediction captures the general trend of price movement.

Test: RMSE: The root mean square error between the forecast and the actual target computed on the test set. Defined as:

$$\boldsymbol{RMSE = \sqrt{\frac{\sum_{t=1}^{T}(\hat{y}_t - y_t)^2}{T}}}$$

Test:mean_wQuantileLoss: The average overall quantile losses computed on the test set. To control which quantiles are used, set the test_quantiles hyperparameter.

Train:final_loss: The training negative log-likelihood loss averaged over the last training epoch for the model.

Project Workflow:

1. Loading and exploring the data. Plot the general trends of close price. Calculate ROC and plot the trends. Plot the histogram to see the distribution of ROC.
2. Creating training and test sets of time series. Training set is the data of total 252 trading days in one year. Test set is the data of last 20 trading days.
3. Formatting data as JSON files and uploading to S3. According to AWS Sagemaker DeepAR Documentation, the input training data should be a formatted JSON file.
4. Instantiating and training a DeepAR estimator. For DeepAR model, we need to specify a image “forecasting-deepar” and pass it to the base estimator constructor.
5. Deploying a model and creating a predictor.
6. Evaluating the predictor by the metrics mentioning above. Plot the predictions ang compare with the actual target values. RMSE, Quantile Loss…
7. Conclusion and future improvement.