Answer Book for Remote Assessment Candidates

Candidates can type or write their assessments by hand. Where answers are handwritten, these must be legible for the examiners. Candidates can scan handwritten documents using a scanner or their phone's camera or an app such as Microsoft Lens. This is also a useful way to draw and capture mathematical workings, drawings or graphs for insertion into the assessment document. Assessments can also be dictated using speech to text on your mobile phone. You can find further information on using your mobile device for online assessment here.

Student and Assessment Details

Student ID Number: 13199041

School: BEI

Department: Computer Science and Information Systems

Module Title: Applied Machine Learning

Module Code: BUCI077H7

Credit Value: 15

Start your answer here

1. Analyze the problem

The goal for this exam is to build a machine learning model that can predict stock price by analyzing the stock market data for a particular stock which can help the company's stock not to be undervalued.

This stock market data is a time-series problem (forecasting), we can use the recurrent neural network model (RNN) the LSTM version [1] for this forecasting problem. The LSTM used for this model is known as gated RNN.

The solution model is presented below (fig.1) consist of a pre-processing layer followed by a LSTM classifier which will then be tuned (hyperparameter) for obtaining the best accuracy. The tuned parameter will then be fitted and transformed on the model to obtain the accurate predictions. The goal for LSTM would be to obtain/generate a predicted time series which can be used to be verify with the future stock price trends.

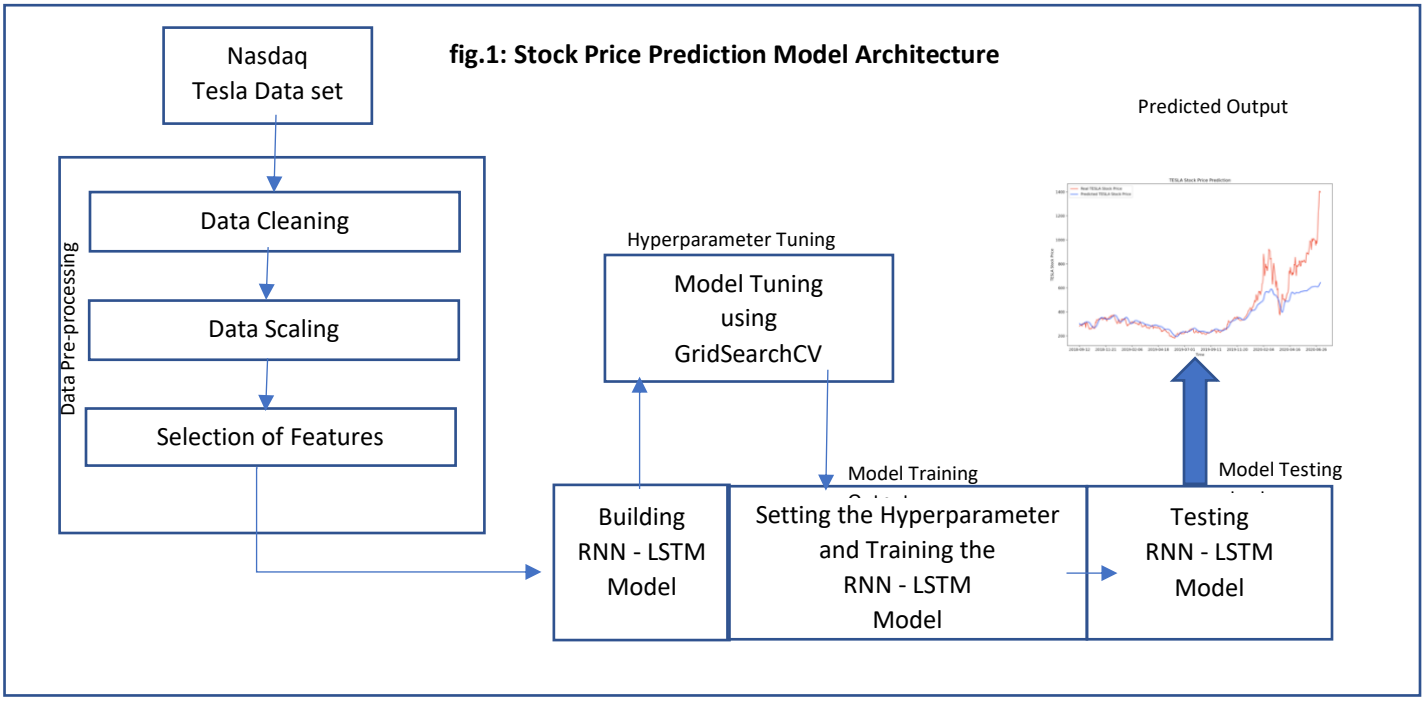


fig.1: Stock Price Prediction Model Architecture

LSTM are a complex feature [11] of recurrent neural network (RNN) which has a feedback mechanism that gives the feedback from the previous block to learn the patterns of the new input data.

2. Pre-processing of the data

This section will be divided into four parts

2.1 Source of data
2.2 Data cleaning techniques
2.3 Data scaling techniques
2.4 Selection of features

2.1 Source of data

To start with the process, we are given a link of 'Nasdaq historical data' [2] which will be the source of the datasets. This historical data consists of daily stock prices which can be dated from most recent up to 10 years in the past. These historical data are separated according to the most popular companies who are in trend for investment. Companies like Apple, Starbucks, Microsoft, CISCO, Amazon, Facebook, Tesla are few of the companies which are mentioned on this page. For our model, we will select any one company and download their historical data for the past 2 months. One of the important reasons to select the data of past 2 months is to identify the current trend of the stock price. This data consists of the following lables:

date (date of closing of stock)
closed/last (the closing price of stock on a particular given date)
volume (number of shares circulated in the market)
open (the stock price of the share at the beginning of the date)
high (the stock price at the highest peak on the given date)
low (the stock price at the lowest peak on the given date)

The target value from this data set will be '**closed/last**' as it will help us understand the predicted output from the model are accurate.

2.2 Data cleaning techniques

As the data is the stock market data, it is impossible to get any missing values, noise or inconsistency in the data. If there is any missing data in prices, we can fill-up with the mean of the same feature. This is highly subjective as the trend of the stock prices could be changed drastically. So, to avoid getting drastic values we would use the mean of past 7 days from the missing values.

2.3. Data scaling techniques

After the data cleaning step (if there is any necessity of this step) we will have to scale the data using 'MinMaxScaler' from the 'sklearn.pre-processing' library which helps to rescale the attributes of the data between the range 0-1. This scaling is also known as normalizing data as it helps the neural network (RNN-LSTM) [3] understand the values of the dataset.

2.4. Selection of features

In this section we will be considering all the features of the dataset as the stock prices for opening, closing, volume, high, low and date are important factors for predicting the accurate output. So, we will not be using any feature selection techniques in this section as we are considering all the features of the dataset.

We will convert the dataset into array (data frame) i.e. 'X' & 'Y'. Where 'X' will consist of features like volume, open, high, low and date. On the other hand, 'Y' will consist of 'closed/last' i.e. closing price of the stock data on the given date.

Now we split the data frame into 'train-test split', i.e. the training set will be 70% and the test 30% using the 'train_test_split' from the 'sklearn.model_selection' library [4] which divide the data for training and testing using 'seed' equal to 7 (i.e. random_state = seed). Seed is a random number assigned to the model to get the same result again and again, it also shows, the beginning of the count [5].

3. Model Building and Model Tunning:

3.1. Model Building:

To build a RNN - LSTM model we will first create four hidden layers with 50 neurons in each layer and use a single neuron in the output layer to make appropriate predictions on the data. Each layer will be appointed with a dropout which will help the neural network to prevent it from overfitting [6] we used the optimizer 'adam' and the loss function as 'mean_squared_error'. The adam optimizer helps in balancing the weights in neural network by adaptively estimating the first and second order of the movements in the neural network [7]. The mean squared error (MSE) is a commonly used loss function which takes the square of distance from the target class and the predicted output [8].

Now we will fit the ‘X_train’ and ‘Y_ train’ (from the train test split output) on this LSTM model with an epochs of 100 and badge size 30. We will now get average model accuracy along with the average loss.

3.2. Tuning the model

After we get the average accuracy and the average loss, we will introduce this model to ‘GridSearchCV’ from the ‘slkearn.model_selection’ library [9] which takes various parameters to tune this model and give us the best accuracy along with the best hyper parameter to achieve that accuracy. In this model we will assign a tuple of epochs = [50,100,150,200] and the badge_size = [20,30,40,50]. From this ‘GridSearchCV’ output we can understand the best hyperparameters required to tune this model. Now we will run the training dataset again on the LSTM model along with the new acquired hyperparameter’s (i.e epoch and badge size).

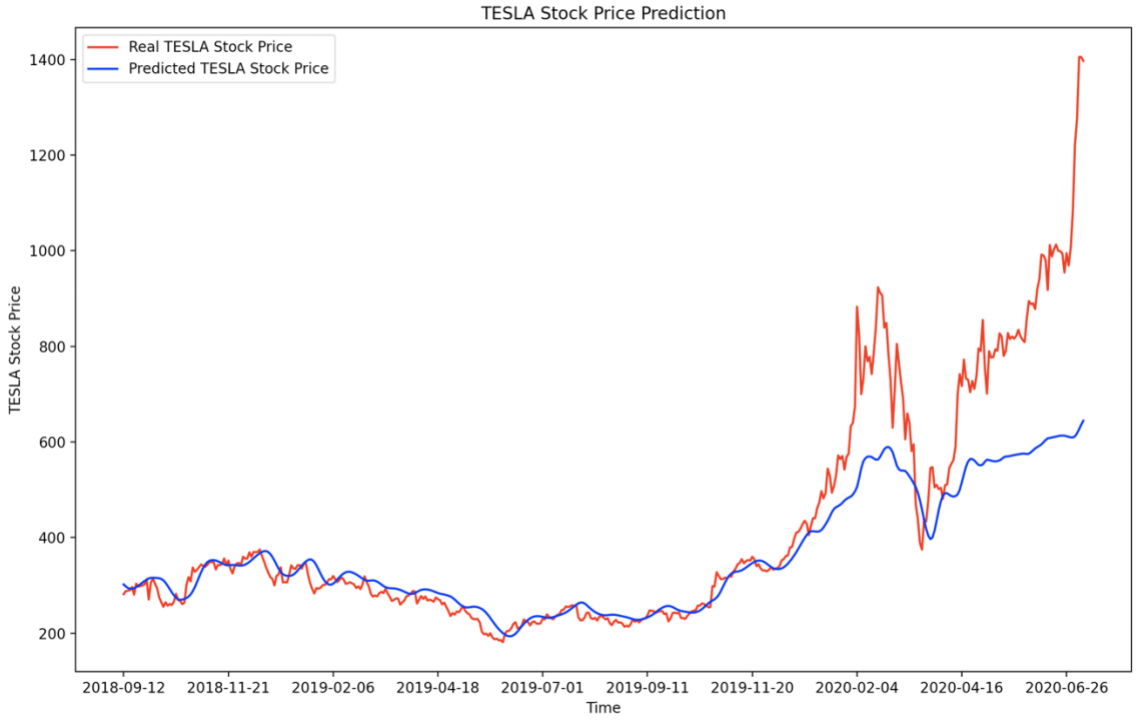
3.3. Testing the model

After tuning the model, we now fit the test dataset (X_test) on the model by using the predict function on the test dataset. We now must inverse the predicted values to its original representation because we had scaled the data using the ‘minmaxscaler’ in the data preprocessing stage. To inverse the values we can use ‘inverse_transform’ from ‘sklearn.preprocessing.minmaxscaler’ library [3].

4. Interpreting the predicted results.

This this section we cannot use the confusion matrix to read the correct prediction as this is a time-series data and it will always predict in approximate values. We can represent the data using a comparison plot (fig. 2) where we can project the real stock price and predicted stock price.

fig.2 : Sample - Stock Price Prediction Output



The image above is sourced from the website 'towardsdatascience.com' [10] which show the real data is red color and predicted data in blue color. We can note from the above image that the predicted output can be lower than the actual output as there are many circumstances that can affect the actual stock price, for an example the COVID-19 Lockdown impacted stock prices globally, which this model cannot take into consideration.

Conclusion

The impact of predicting stock prices compared to the real value of the stock could have unexpected hikes or drops depending on the circumstances that affect the company's business and stock prices. To predict better on this model, it would be best to only use data from past 3 days because it will learn minor fluctuation of the stock price and can give much accurate output compared to the real stock price.

References

[1] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2017.

[2] *Nasdaq.com*. [Online]. Available: https://www.nasdaq.com/market-activity/quotes/historical. [Accessed: 10-May-2021].

[3] "sklearn.preprocessing.MinMaxScaler — scikit-learn 0.24.2 documentation," *Scikit-learn.org*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html. [Accessed: 10-May-2021].

[4] "Sklearn.Model_selection.Train_test_split — scikit-learn 0.24.2 documentation," *Scikit-learn.org*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html. [Accessed: 10-May-2021].

[5] "Python random seed() method," *W3schools.com*. [Online]. Available: https://www.w3schools.com/python/ref_random_seed.asp. [Accessed: 10-May-2021].

[6] C. Maklin, "Dropout neural network layer in keras explained - towards data science," *Towards Data Science*, 02-Jun-2019. [Online]. Available: https://towardsdatascience.com/machine-learning-part-20-dropout-keras-layers-explained-8c9f6dc4c9ab. [Accessed: 10-May-2021].

[7] Keras Team, "Adam," *Keras.io*. [Online]. Available: https://keras.io/api/optimizers/adam/. [Accessed: 10-May-2021].

[8] P. Grover, "5 regression loss functions all machine learners should know," *Heartbeat*, 05-Jun-2018. [Online]. Available: https://heartbeat.fritz.ai/5-regression-loss-functions-all-machine-learners-should-know-4fb140e9d4b0. [Accessed: 10-May-2021].

[9] "sklearn.model_selection.GridSearchCV — scikit-learn 0.24.2 documentation," *Scikit-learn.org*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html. [Accessed: 10-May-2021].

[10] S. Loukas, "Time-series forecasting: Predicting stock prices using an LSTM model," *Towards Data Science*, 10-Jul-2020. [Online]. Available: https://towardsdatascience.com/lstm-time-series-forecasting-predicting-stock-prices-using-an-lstm-model-6223e9644a2f. [Accessed: 10-May-2021].

[11] "CS 230 - recurrent neural networks cheatsheet," Stanford.edu. [Online]. Available: https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks. [Accessed: 10-May-2021].