

# Mathematical Finance 2020

## Project

### Stock Price Prediction

Note:1 You should submit a notebook in which all cells have been run. Outputs should be visible.

Note:2 A Jupyter notebook “LSTM Stock Price Prediction” is available which does preprocessing using a different method to that which you will use, and setting up the model and training in Keras using the same method as you must use. You should also look up how the LSTM works and what stateful means in Keras on the internet.

1. Use Python module `yfinance` to download daily, historical prices of the JSE LIMITED (JSEJF) Stock Prices for the period January 01-2010 to December-31 2019. Clean the data if necessary.
2. Construct a pandas dataframe with columns labelled Open and Close, indexed by dates and containing the data from these two columns of the input, using the data downloaded.
3. Use the time series for Close in order to predict the following day's value of Close. Compare this with predicting the Close value as being the same as the Open value of the same day. Proceed as follows:
4. Use ScikitLearn's `StandardScaler()` function to normalise your data to be between 0 and 1.
5. Split your data into training and testing at a proportion of your choice. Justify why you used such a proportion.
6. Using a rolling window (look-back) of 10, create labelled training and testing sets, and reshape these sets to have the format the sizes of [samples, timesteps, features].
7. Train and test a Stateful LSTM model. State your final training and testing losses.
8. Super-imposed on each other, plot the training, the predictions on the training, the testing and the predictions on the testing data in one figure.
9. Plot in the same figure the prediction of today's value of Close on the test set as being the value of Open on that day, and the previous predictions of the test set by the neural network.
10. (Bonus points) Repeat 7 using the actual prices of the stock instead of the normalised values used in 7 above. (Hint: this involves using the inverse of `StandardScaler()`).