Solution Design Document for the project titled "Deep Learning based Forecasting of Credit Derivatives Indices"

In this project we apply Deep Learning architecture to predict the spread of credit default swap (CDS) on the North America Investment Grade index (CDX.NA.IG). The implemented Long-Short Term Memory (LSTM) model is compared with a baseline Support Vector Machine (SVM) model through the root mean squared error. We also test the hypothesis that enhancing the feature space of the LSTM model with a rolling time series data of Hurst Exponent improves the training data fit and test data forecast accuracy.

An online repository for the current project is publicly available at the below GitHub address. Python is the programming language for implementation.

https://github.com/rks972633/MScFE_Capstone

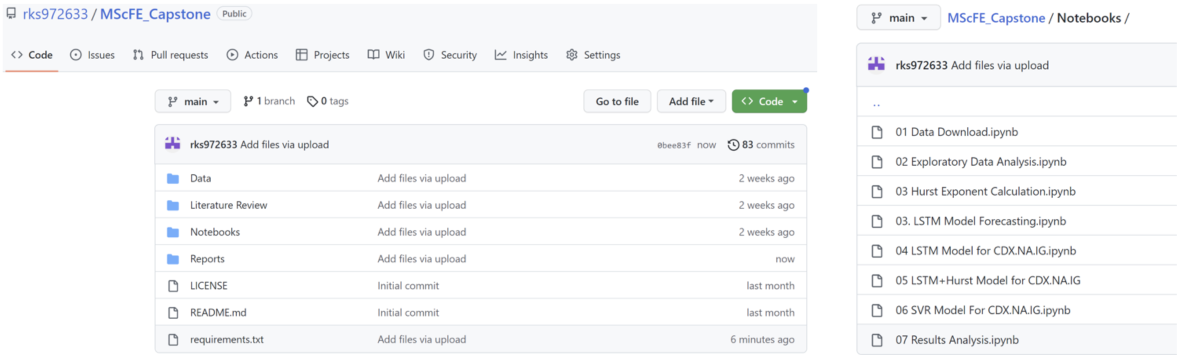

Figure 1. GitHub repository architecture

Figure 1 above shows the GitHub architecture for the project under study. We briefly go over the steps we took over the course of the project.

We start with downloading the time-series data of the CDX.NA.IG spreads using a Bloomberg terminal and save the csv file under the folder named Data and this file is used for the rest of the project via the various notebooks. As we can see the various python notebooks are segregated to perform specific part of the project and most of them are self-explanatory in terms of what they are supposed to do.

The key notebook in the project is the one titled "LSTM Model for CDX.NA.IG". It is here that we implement our Deep Learning based LSTM model to train and then predict the spread. *Keras* package has built-in routines to implement the LSTM model. We make use of another package named *sklearn* to scale our data and split data into training and testing set. Root mean squared error metric is used to evaluate the training fit and accuracy of the forecast. The various parameters are optimized through hyper-parameter tuning in this notebook. The approach is more like a brute force way.

Our model is compared with a baseline model based on Support Vector Regression. The implementation of the baseline model is done on similar lines in a separate notebook and the results are compared and that's where we test our first hypothesis that LSTM performs better than SVR model.

To test our second hypothesis we enhance our LSTM model by adding an additional feature. This is a rolling time series of Hurst Exponent and we perform similar training and testing to test our hypothesis that Hurst Exponent improves the prediction quality.

We report the results and conclusions after we have arrived at all the results from various models. Discussion around the two hypothesis testing done is also presented.