CSC 215-01 Artificial Intelligence (Spring 2023)

Project 2: Time Series Forecasting using NN, LSTM and CNN

Due at 3:00 pm, Monday, March 6, 2023

Demo: class time, Monday, March 6, 2023

1. Problem Formulation

Time series forecasting is an important area of AI. It is important because there are so many prediction problems that involve a time component. This time component makes time series problems more difficult to handle. In this project, you practice with time series data to predict stock price.

This project is threefold:

- Task 1: Predict [Close] of a day based on the last 7 days' data [Open, High, Low, Volume, Close] using a **full-connected neural network model**. In other words, we want to predict the price in the green cell using all the numbers in the red cell. Use the first 70% of the available records for training and the remaining 30% of the available records for test. Report the RMSE of the model. Show the "regression lift chart" of your test data.

  Hint: Each record has 7 * 5 = 35 input features and 1 output feature.

- Task 2: Do the same as Task 1 but use a **LSTM model**. Report the RMSE of the model. Show the "regression lift chart" of your test data.

  Hint: Each record can be viewed as a sequence of 7 vectors, each vector with 5 dimensions.

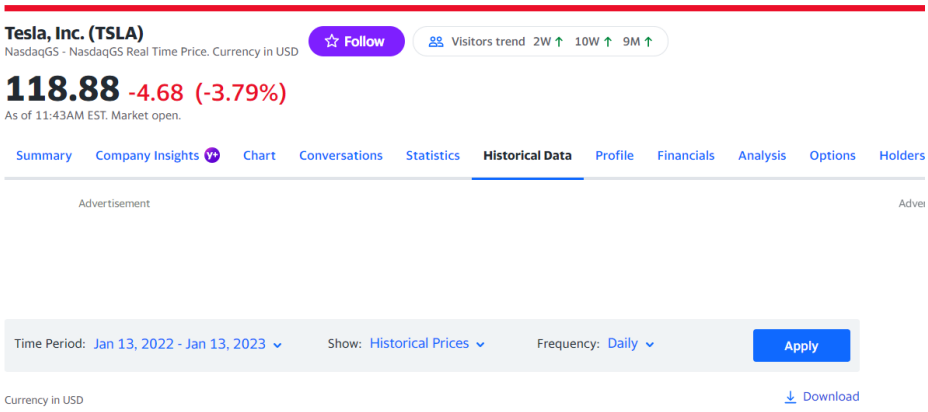
- Task 3: Do the same as Task 1 but use a **CNN model**. Report the RMSE of the model. Show the "regression lift chart" of your test data.

  Hint: The red cell can be viewed as a 1D image of 7 pixels, each pixel with 5 channels, or as a 2D image of 7 * 5 = 35 pixels, each pixel with 1 channel.

	A	B	C	D	E
1	# http://finance.yahoo.com/quote/GOOG/history?ltr=1				
2	# Open	High	Low	Volume	Close
3	828.66	833.45	828.35	1247700	831.66
4	823.02	828.07	821.655	1597800	828.07
5	819.93	824.4	818.98	1281700	824.16
6	819.36	823	818.47	1304000	818.98
7	819	823	816	1053600	820.45
8	816	820.959	815.49	1198100	819.24
9	811.7	815.25	809.78	1129100	813.67
0	809.51	810.66	804.54	989700	809.56
1	807	811.84	803.19	1155300	808.38
2	803.99	810.5	801.78	1235200	806.97

2. Dataset

1. Go to https://finance.yahoo.com/
2. Type any favorite company (such as Apple, Tesla, Alphabet) in the top search bar.
3. Click on the tab “Historical Data”



4. In “Time Period”, choose “Max”.
5. Click on “Apply” and Choose “Download”

Note that the downloaded excel file has seven columns as follows:

Date, Open, High, Low, Close, Adj_Close, Volume

Remove "Date" and "Adjusted_Close" columns since we do not need them.

3. Requirements

- Split data to training and testing. Use training data to train your models and evaluate the model quality using test data
- Do feature normalization. **Notice that you should never normalize the output feature when training any regression models.** Otherwise, the RMSE of the regression model will be also normalized.

  Hint: Create two separate copies of the "close" column and use one as output feature and the other as part of the input
- You must use EarlyStopping and ModelCheckpoint when training neural networks, LSTM, and CNN using Tensorflow.
- You must use TensorBoard to plot the training and test loss when using Tensorflow.
- Tuning the following hyperparameters to record how they affect performance in your report. Tabulate your findings.
  - **Activation:** relu, sigmoid, tanh
  - **Number of layers and neuron count in each layer**
  - **Optimizer:** adam and sgd
  - **Kernel number and kernel size** (for CNN only)
  - **Number of LSTM layers and neuron count in each layer** (for LSTM only)

4. Grading Breakdown

You may feel this project is described with some certain degree of vagueness, which is left on purpose. In other words, **creativity is strongly encouraged**. Your grade for this project will be based on the soundness of your design, the novelty of your work, and the effort you put into the project.

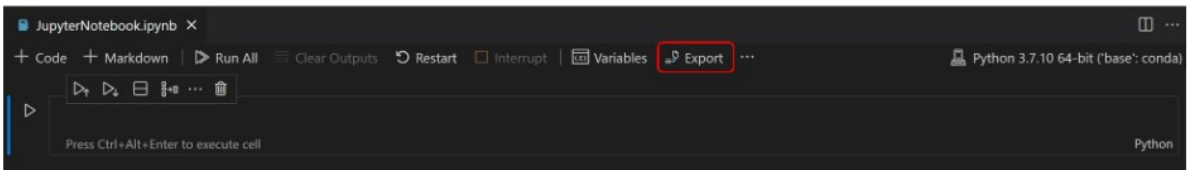
Use the evaluation form on Canvas as a checklist to make sure your work meets all the requirements.

5. Teaming

Students must work in teams of 2 people. Think clearly about who will do what on the project. Normally people in the same group will receive the same grade. However, the instructor reserve the right to assign different grades to team members depending on their contributions.

6. Deliverables

(1) The HTML version of your notebook that includes all your source code. In VS Code, you can export a Jupyter Notebook as an HTML file. To export, select the Export action on the main toolbar. You'll then be presented with a dropdown of file format options.



(2) Your report in PDF format, with your name, your id, course title, assignment id, and due date on the first page. As for length, I would expect a report with more than one page. Your report should include the following sections (but not limited to):

- Problem Statement
- Methodology
- Experimental Results and Analysis
- Task Division and Project Reflection
- Additional Features

In the section "Task Division and Project Reflection", describe the following:

- who is responsible for which part,
- challenges your group encountered and how you solved them
- and what you have learned from the project as a team.

In the section "Additional Features", you describe and claim credit for additional features.

To submit your notebook and report, go to Canvas "Assignments" and use "Project 2".

All the deliverables must be submitted **by team leader** on Canvas before

3:00 pm, Monday, March 6, 2023

NO late submissions will be accepted.

7. Possible Additional Features (5 pts per feature, 10 pts at most)

- In the project, you predict [Close] of a day based on the last 7 days' data. Can you find the best N value (number of the days we should consider in the past) that yields the most accurate model?

- Can you use LSTM to predict the stock prices for a continuous future time period (e.g., the prices in the next five days)? Show the true prices and predicted prices in the same chart.

  Hint: train a multi-output regression model.

- Can you build a mode using bidirectional LSTM and Attention layers to see if it can beat your baseline NN/CNN/LSTM models?

  Here you can find the implementation for bidirectional LSTM layer:

  https://keras.io/api/layers/recurrent_layers/

  Read this paper for a comparison of RNN, LSTM and Attention:

  https://medium.com/swlh/a-simple-overview-of-rnn-lstm-and-attention-mechanism-9e844763d07b

  Here you can find two APIs to quickly add Attention mechanism on top of your LSTM model:

  https://github.com/philipperemy/keras-attention-mechanism

  https://github.com/CyberZHG/keras-self-attention

- Can you build a transformer model (an extension of RNN) for this problem? How it will compare with your baseline NN/CNN/LSTM models?

  A nice introduction of Transformer: https://github.com/microsoft/AI-For-Beginners/blob/main/lessons/5-NLP/18-Transformers/TransformersTF.ipynb

  Code example: https://keras.io/examples/timeseries/timeseries_transformer_classification/

8. In-class Presentation.

On the due day, each team has 5 minutes to present your work in the class. Explain your solutions by referring to your notebook. You do not have to prepare the PowerPoint slides for your presentation.