Project B3: STOCK MARKET. Using neural networks for stock market prediction

Project repository: https://github.com/allanalikas/ITDS_Stock_Prediction

Business understanding

1. **Identifying our business goals**
    - **Background:**
      In our project we will focus on prediction of the stock market data - the highly volatile and complex time series. Traditionally such machine learning approaches as SVM and Regressions were used to predict possible price movements. However, in recent research it is shown that neural networks are much better at handling non-linear models. In this project we will see for ourselves, if it is true.
      Our team will benefit the most from the completion of this project: (1) we pass the course "Introduction to Data Science"; (2) we get to practice the material on the real world data; (3) we get to implement neural networks not covered in this course; (4) while working on this project we could receive feedback from our instructors.
    - **Business goals:** (1) Implement the material learned during the course "Introduction to Data Science"; (2) Study in depth neural networks and their implementation on financial data.
    - **Business success criteria:** (1) correct implementation of machine learning techniques; (2) gain insight in neural networks; (3) achieving the performance not worse than the benchmark.

2. **Assessing your situation**
    - **Inventory of resources:**
      People: While working on this project we have access to the following human resources: (1) our team (three students from the Institute of Computer Science); (2) our instructors (we can ask questions during the consultation and practice sessions), (3) our classmates (through Piazza); (4) professional community, etc.(https://stackoverflow.com/ and other forums)
      Data:
      Huge Stock Market Dataset: Historical daily prices and volumes of all U.S. stocks and ETFs. https://www.kaggle.com/borismarjanovic/price-volume-data-for-all-us-stocks-etfs
      Software: (1) Google Colaboratory; (2) Git; (3) Jupyter Notebook.

Hardware: (1) Personal laptops, provided by the University's IT section; (2) Google hardware, including GPUs and TPUs, through Google Colaboratory

- **Requirements, assumptions, and constraints:**
  Schedule for completion: The deadline for submission of the project is the noon (12:00) of Monday, Dec 14.
  Requirements for acceptable finished work: (1) the poster slide; (2) the 3-min video; (3) program source code for our project
  Legal and security obligations: Any legal and security obligations do not apply to this project.
  Constraints: (1) time constraints – 30 hours per person (90 in total); (2) computational constraints – difficulty of implementing neural networks.
- **Risks and contingencies:** As a deadline for this project is a strict one, we need to make sure that the project is submitted on time. The only feasible risk is the malfunction of equipment or problems with internet connection. As a contingency, we will use Google Colaboratory that runs entirely in the cloud.
- **Terminology:** In this project we use the terminology as covered in the course "Introduction to Data Science". As we will try to implement several neural networks, we added some definitions specific for deep learning.
  Convolutional Neural Network (CNN): a deep learning neural network designed for processing structured arrays of data. The basic components of a Convolutional Layers are: (1) Convolutional Layer; (2) Pooling Layers; (3) Fully connected layer with output layer. A convolutional neural network is a feed-forward neural network.
  Feed-Forward Neural Network (FFNN): an artificial neural network in which the connections between nodes does not form a cycle. The drawback of FFNN is the fact that the consideration of current inputs and notion of order of time is absent.
  Recurrent Neural Network (RNN): a type of neural network that contains loops, allowing information to be stored within the network. RNN use their reasoning from previous experiences to inform the upcoming events. The drawback of the RNN is its vanishing gradient problem.
  Vanishing gradient problem: an issue that sometimes arises when training machine learning algorithms through gradient descent. Since the gradients control how much the network learns during training, if the gradients are very small or zero, then little to no training can take place, leading to poor predictive performance.
  Long Short-Term Memory Network (LSTM): a form of recurrent neural network widely used for image, sound and time series analysis, because they help solve the vanishing gradient problem by using memory gates.
- **Costs and benefits:** This aspect is not relevant for the project

3. Defining your machine learning goals

- **Machine learning goals:**

Models:
Traditional Machine Learning: (1) Regression; (2) SVM
Neural networks: (1) CNN; (2) RNN; (3) LSTM
Report: (1) submit program source code for our project; (2) modify the report if necessary (expand Glossary, change the list of models, etc.)
Presentation: We need to prepare for the final presentation and submit the following materials: (1) the poster slide; (2) the 3-min video

- **Machine learning success criteria:** in the framework of the course "Introduction to Data Science" this project will be successful if it fulfils the criteria set out by our instructors. In general, we would like to achieve the performance not worse than the benchmark.

Data understanding

1. **Gathering data**
   - **Data requirements:** Every entry of our data should have the following parameters: Date (Date object), when the trade was made, Open (a float value), the value the stock was at when the stock market opened on that specific date, High (float value), the max value of the stock during that specified date, Low (float value), the lowest value of the stock during that specific date, Close(float value), the value the stock ended at the end of that date, Volume (Integer value), the amount of stock that was traded during that date.
   - **Data availability:** We have a few thousand files with 5 thousand to 10 thousand entries per file. Each file is about a different company so in case our scope is too broad, we can filter the datasets with a specific criterion like IT companies or predict the stocks for a specific company. And as the data is in a chronological format then we could narrow the scope to a certain period.
   - **Selection criteria:** Our data is in text files that we downloaded from Kaggle. The relevant fields in the dataset are the previously mentioned Date, Open, High, Low, Close and Volume fields that are in every file.

2. **Describing data**
   Our data's brief explanation is "US company stock price changes during several years". Our dataset is from Kaggle and the dataset is meant for machine learning activities, which reduces the need for exorbitant data preparation to get the data into a state that we can use machine learning techniques on. The fields in the dataset are already data types that we can use straight away. The fields in this dataset are: Date, Open, High, Low, Close, OpenInt. The OpenInt field is the only field that has no real meaning for us. The dataset is made up of over seven thousand files each describing the stock price changes by days. Averaging multiple thousand entries per file means that we have plenty of data to work with and the dataset is described as a Time-Series dataset meaning that the entries are formatted chronologically. This gives us quite a deal of freedom to analyse periods of time for the different companies and deduce what kind of events in the world may have influenced the stock prices of those companies and find trends in the prices of the stock by date. This also allows us to

gather more data for neural networks to deduce whether certain events in the world had any effect on the stock prices in our dataset.

3. **Exploring data**
   Regarding the available data that we currently have, it seems that the earliest entry was in the 1970s, and the latest in the 2010s, 2017 to be correct. But, since our dataset is massive, more than a thousand files of content, we should have a really good chance of applying machine learning correctly to predict the stock market in the future. Each company has its own text file and in it all the data we need. From the amount of stock traded to the highest and lowest price of that particular date. Because every change in the stock market is shown, there really are not problems with data quality.
   Because our dataset is taken from Kaggle archives as stated before, the data is already in the format we need, so it saves us the time needed to set up the info in the wanted form and the stage for data preparation is set.

4. **Verifying data quality**
   For our dataset the quality is good, there does not seem to be any problems regarding the given content, only 1 field, OpenInt, we will not be using, since it has no actual use for us. Better yet, every company is in its own text file and in chronological order, with every change in the given date recorded. Because of this, we do not need to look for alternatives to get data, more likely if the project's scope is too large, we may need to cut down on the companies we look through, for example filter out non-IT companies and the like. And so we have verified that the given data is enough to complete our goals.

Planning the project

Plan:

- **Data preparation:** This part is done for us, since we have taken a premade dataset and checked if we are able to use it in the given format.
- **Data visualization (1.5 hour for each member)**
- **Modelling (25 hours for each team member)**
  Select modelling techniques:
    Traditional Machine Learning: (1) Regression; (2) SVM
    Neural networks: (1) CNN; (2) RNN; (3) LSTM
  Generate a test design: training, test and validation datasets
  Build the models:
    Parameter setting: chose the initial values of model's hyperparameters
    Model description: (1) Model description; (2) Necessary variables; (3) Model interpretation
    Models: implement the chosen modelling techniques
  Assess the models:
    Model assessment: (1) RMSE; (2) MAE; (3) MAPE

Revised parameters settings: tuning the model's hyperparameters

- **Evaluation (1 hour for each team member)**
  (1) summarize the results; (2) check if the business goals have been met
- **Final submission (1 hour for each team member)**
  Prepare the poster slide
  Prepare the 3-min video
  Prepare the final program source code
  Modify the project report
- **Final presentation (1.5 hours for each team member)**
  Participate in the poster session

Methods: Regression, SVM, Neural Networks

Tools: Google Colaboratory, Git