EN.705.603.82.FA22 Creating AI-Enabled Systems
Prof. John Hebeler

# System Project

## Stock Price Prediction

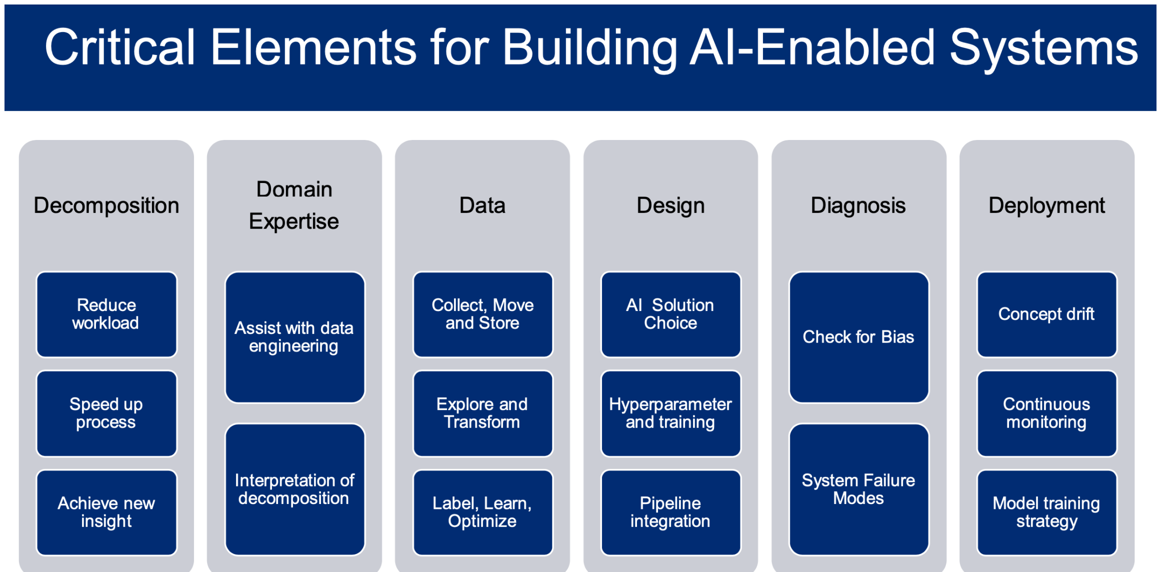
Peter Wang

Oct 29, 2022

## 1. Introduction

This project primarily focuses on the usage of 6 D's framework to create an AI enabled system. The AI enabled system for this project is to use a model long short-term memory-convolutional neural network to predict stock prices. Machine learning is a part of AI, and the developments in the field of machine learning allowed the use of computers in the task predicting financial markets. Machine learning is about building computer systems or programs that can learn from data. Traditional machine learning approaches to stock prediction have focused on improving their performance with different techniques for feature extraction to select the most promising features from a dataset. Deep learning networks are powerful machine learning algorithms that make use of many cascading layers to learn multiple levels of representations, which gain better performances, and can achieve high accuracy on stock price prediction tasks. More importantly, it is a big challenge to machine learning engineers to have an iterative pipeline to accomplish this task. To ease the stock prediction system development lifecycle, the 6D's framework will be followed. There are six key elements to create an AI Enabled system, which includes: Decomposition, Domain Expertise, Data, Design, Diagnosis, and Deployment. These six elements can be referenced as the 6 D's of

Creating AI Enabled systems and it takes a holistic view from the initial concept through deployment. The following sections will describe each component in detail for creating this system.




## 2. Decomposition

This decomposition component of the 6 D's framework mainly consists of refining the technology concept and identifying our main topic. In this step, the prediction system is reduced to one stock prediction, Apple's stock is chosen. To simplify the workload and gain new insight, a concept map is built as follows,

Concept Map

Concept Name: Stock price prediction using machine learning

Student Name: Peter Wang

How Might We: create a tool to predict stock price

Stakeholders: Primary: Stock traders

Visuals (e.g., pictures, flow charts, etc.):

What tool is:
predict stock price

What tools is not:
Replacement of human decision making

Explanation:

Predict stock price using machine learning to help stock traders make decision.

What Need Does it Address?

Help stock traders discover the future value of company stock and make decision.

How might this idea be quickly tested/prototyped?

Predict on unseen data

Why Might it Fail?

Prices moves in the opposite direction of the prediction
Too large error

## 3. Domain Expertise

Domain expertise plays an important role beyond decomposition, providing key support during the subsequent phases of the 6 D's framework. For example, domain experts bring credibility to community members who will be the ultimate users of AI-enabled systems and can assist with adoption of new technologies. Domain expertise helps on data engineering to let us pick proper features and engineering methods. Domain expertise also gives deep understanding of the data, the dynamic change of the data. According to the domain knowledge on stock market, models capable of predicting future values based on previously observed values are known as time-series forecasting models, and time-series data are those whose statistical features, such as mean and standard deviation, do not remain constant throughout time but instead change, so predicting the closing price of Apple stock for the following trading day will be implemented in this system.

## 4. Data

In AI projects, the data engineering phase is generally the most resource intensive, involving collecting, moving, storing, transforming, labeling, and optimizing data to facilitate the design of the system.

Data is the biggest and the most important aspect of any AI enabled system. There are several techniques to engineer data, one such data engineering perspective is the Extract, Transform, and Load (ETL) process. The Extraction step addresses the collection of raw data using a range of various methods and quality. The transformation is critical and generally is the most resource intensive step. This transformation typically consists of processing data, load and cleaning, feature normalization and extraction, stop words, etc. The pipeline for this transformation process can also be denoted as preprocessing. Lastly, the load step consists of the end-stage of data which it can be fed to the model.

### 4.1 Collection and Storage

This application requires data, like any AI systems, to build predictive models. The data are collected from Yahoo Finance for this system. Yahoo Finance is a media platform that provides financial news, data about stock quotes, press releases, and financial reports. And all the data provided by Yahoo Finance is free. Yahoo Finance API is the API that Yahoo provides to fetch financial information. The data include date, open price, high price, low price, close price, and volume. Storing in a CSV file is sufficient for this system, and there is no requirement to fetch the data in real-time, but saving the data into a time series database would be a good consideration.

### 4.2 Data Processing

To predict the closing price of Apple stock, all the data used are numerical values, no further preprocessing needed, but for fast training, normalization is applied on all the features.

## 5. Design

The AI field is characterized by several algorithmic approaches ranging from rule-based systems to machine learning algorithms. In practice, many AI-enabled solutions may include a combination of algorithm classes.

Once the dataset has been processed, the next step is to build the pipeline for training and evaluation. This pipeline starts builds the model, evaluates it on the evaluation dataset, and tunes the parameters to test the final performances of the test dataset.

### 5.1 AI solution choice

In this system, combined deep learning algorithms Convolutional Neural Network and Long Short-Term Memory are chosen.

An LSTM network is very powerful in sequence prediction problems because it can store past information. In this way, it allows the model to provide contextual information past a sequence of characters in the data. Convolutional Neural Network is also a powerful deep learning algorithm because it can detect features in an unsupervised fashion and learn distinctive features on its own.

**5.2 Hyper parameters and training**

Hyperparameters are used to control the learning process of the model. To optimize the model, the following hyper parameters are identified and tuned:

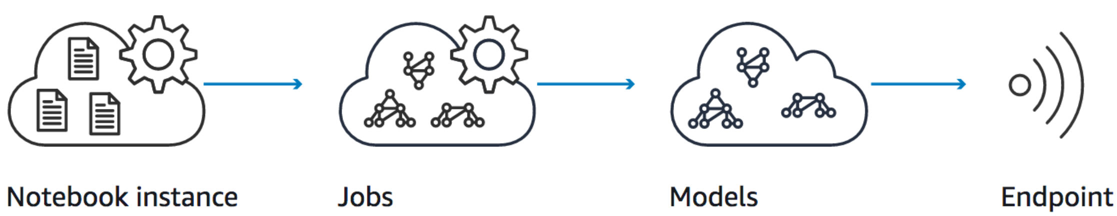
- Batch Size
- Number of Epochs
- Number of Layers
- Number of Nodes
- Activation Functions
- Learning rate and Decay

**5.3 Pipeline integration**

Cloud independence is a method of designing your technology solution to be cloud agnostic so that it's not locked to a single provider. The solution option of cloud independence is to use Kubernetes, but for this project cloud provider AWS is chosen. The product is named Amazon SageMaker which is the Machine Learning platform on AWS that provides infrastructure to run hosted Jupyter Notebooks. Amazon SageMaker is integrated with other storage and analytics services on AWS to make the essential data management tasks for a successful Machine Learning project secure, scalable and streamlined.

In this system, first Amazon SageMaker-hosted notebooks is used to fetch the data from Yahoo Finance, clean it, and aggregate it in Amazon S3 buckets.

In addition to hosted notebooks, Amazon SageMaker also provides managed training and hosting for machine learning models, using a variety of languages and libraries. Once the stock data are normalized and stored in Amazon S3, the next step is to containerize the machine learning training and prediction code, publish it on an Amazon Elastic Container Registry (Amazon ECR) repository, and host the model behind an Amazon SageMaker endpoint to generate predictions.




## 6. Diagnosis

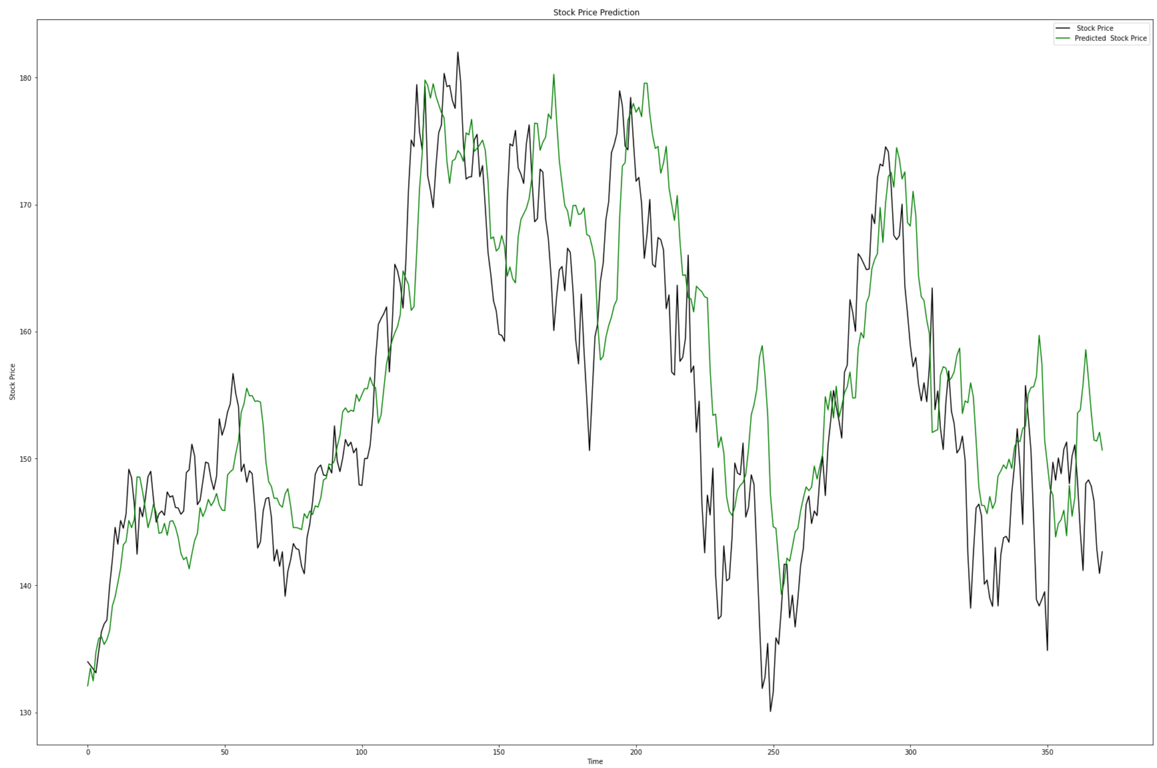
The diagnosis component addresses how AI-enabled systems are assessed and what metrics are used. The design section identified four algorithm classes that bring a different set of metrics. Typical metrics for supervised machine learning algorithms include accuracy, a confusion matrix, per class accuracy, log loss, precision, recall, mean average precision. Once the models have been trained and tested, the next step is to measure the performance of the models.

### 6.1 Evaluation Metrics

The main evaluation metrics used are identified below to evaluate the performance of the model on the test dataset.

MSE: 0.095

The MSE is for normalized differences between consecutive close prices.




## 6.2 Check for Bias

Data bias can be observed if the chosen dates cannot represent the recent trend of the stock price. To overcome this bias, normalized temporal difference is used to reduce this bias.

## 6.3 Ethical Issues

Regression estimation is prone to biases that exist within the data, it is wholly dependent on the dataset. It is trivial to the stock price prediction for ethical issues, because there is no race, gender, face or some other data that are related to ethics.

## 7. Deployment

The last D of the 6D framework is deployment. There is no one-size-fits-all approach for AI-enabled system deployment, but it is important to note that AI technologies are generally employed in a broader system context. This final phase is where the application is deployed and continuously monitored to ensure its performances. The strategy for this application is blue-green deployment. Blue-green deployment strategies involve two production environments instead of just models. The blue environment consists of the live model whereas the green environment consists of the new version of the model. The Blue environment which contains the original model is live and keeps servicing requests while the green environment acts as a staging environment for a new version of the model. It is subjected to deployment and final stages of testing against the real data to ensure that it performs well and is ready to deploy to production. Once the testing is successfully completed ensuring that all the bugs and issues are rectified the new model is made live. Once this model is made live, the traffic is diverted from the blue environment to the green environment. In most cases, the blue environment serves as a backup, in case something goes wrong the request can be rerouted to the blue model. Additionally, this approach provides the ideal amount of flexibility in which to deploy a more tuned system, depending on the model the user would choose.

Monitoring software requires good logging and alerting, but there are special considerations to be made for machine learning applications. All predictions generated by the model is logged in such a way that enables traceability back to the model training job. This application also is monitored for invalid predictions or data drift by human check, which requires the model to be retrained. By continually monitoring the deployment, the analytics of the application will be continually observed. As the data grow, we can introduce more data into the pipeline and further train the model to improve its performances.

## 8. Conclusion

The basic premise of iterative, incremental, and evolutionary project management is that a project is divided into early, frequent, and short duration delivery steps. While it is not difficult to envisage the steps of construction for a system, there is often difficulty in carrying out decomposition into steps when each step has to deliver. This paper gives some guidelines, policies, and principles for machine learning system application development based on a stock close price prediction project.

To build an effective application, the 6D framework is essential. Without properly understanding Decomposition, Domain Expertise, Data, Design, Diagnosis, and Deployment, the application could have suffered and may have been much harder to accomplish expected results. Not only the 6D allows for a smoother build, but it also makes the explanations easier to digest if you can trace the pipeline at each step of the build. With careful consideration of each "D" in the framework, we can modify the application in a modularized manner to appropriate the system for different applications in general.

References:

1. Ismail Fawaz, H., Forestier, G., Weber, J. et al. Deep learning for time series classification: a review. Data Min Knowl Disc 33, 917–963 (2019). https://doi.org/10.1007/s10618-019-00619-1

2. Livieris, I.E., Pintelas, E. & Pintelas, P. A CNN–LSTM model for gold price time-series forecasting. Neural Comput & Applic 32, 17351–17360 (2020). https://doi.org/10.1007/s00521-020-04867-x

3. Yusuf, S.A., Alshdadi, A.A., Alassafi, M.O. et al. Predicting catastrophic temperature changes based on past events via a CNN-LSTM regression mechanism. Neural Comput & Applic 33, 9775–9790 (2021). https://doi.org/10.1007/s00521-021-06033-3

4. Nourbakhsh, Z., Habibi, N. Combining LSTM and CNN methods and fundamental analysis for stock price trend prediction. Multimed Tools Appl (2022). https://doi.org/10.1007/s11042-022-13963-0

5. Lu, W., Li, J., Wang, J. et al. A CNN-BiLSTM-AM method for stock price prediction. Neural Comput & Applic 33, 4741–4753 (2021). https://doi.org/10.1007/s00521-020-05532-z