Beykoz University

Department of “Computer Engineering”

“Engineering Project III- 60610PREOZ - CME0196”

Scope Definition - Final version

Stock Prediction Web Application
- Python & Dash Framework -

Lecturer: Mustafa Yıldız

Leyla Abdullayeva - 1904010038

Contents

1.

1.1. Project Name -

Stock Prediction Web App - Python

1.2. Project Developer

Leyla Abdullayeva - 1904010038

1.3 Project Description - (Briefly)

The working structure of this program is basically stock market prediction which is the act of trying to determine the future value of a company stock or other financial instrument traded on an exchange.
Application will be based on Machine Learning (ML) standart.
The practice of algorithmic trading has been gaining a strong footing in the industry for the past couple of years mainly because of the large scope to study and implement tons of data which would not have been humanly possible.
This project is work in progress that makes use of LSTM and RNN.
The main objective is to analyze large trends of data and maintain a high level of accuracy.

2.

2.1. Research

This application will investigate how different machine learning techniques can be used and will affect the accuracy of stock price predictions. Different models, from linear regression to dense and recurrent neural networks are tested. Different hyperparameters are also tuned for better performance.
The search space for all neural network architectures and hyperparameter combinations is huge, and with limited time in conducting this project, apart from manually trying different reasonable combinations, the team optimizes the models with evolution algorithm, replicating AutoML techniques from other researches with promising results in the financial context.

2.2 Research Design

Research Design contains problems that will occur while predicting the stock price.

2.3 Problem Framing

The problem of the project is set to predict the stock price for the next 10 business days. “10 days” is chosen as the timeframe as short term price movements tend to depend more on trend momentum and price pattern, while long term price movements depend on the fundamentals of a stock (e.g. company management capabilities, revenue model, market demand, macroeconomic factors, etc.).

2.4 Application

This project aims to provide stock price predictions based on the latest machine learning technologies to all retail investors. Web application must have been developed to provide predictions in an intuitive way. Different models’ performance and accuracy can also be compared.
The application also serves as another user interface (UI) in visualizing results from the research apart from Jupyter notebooks with lots of tables and graphs.

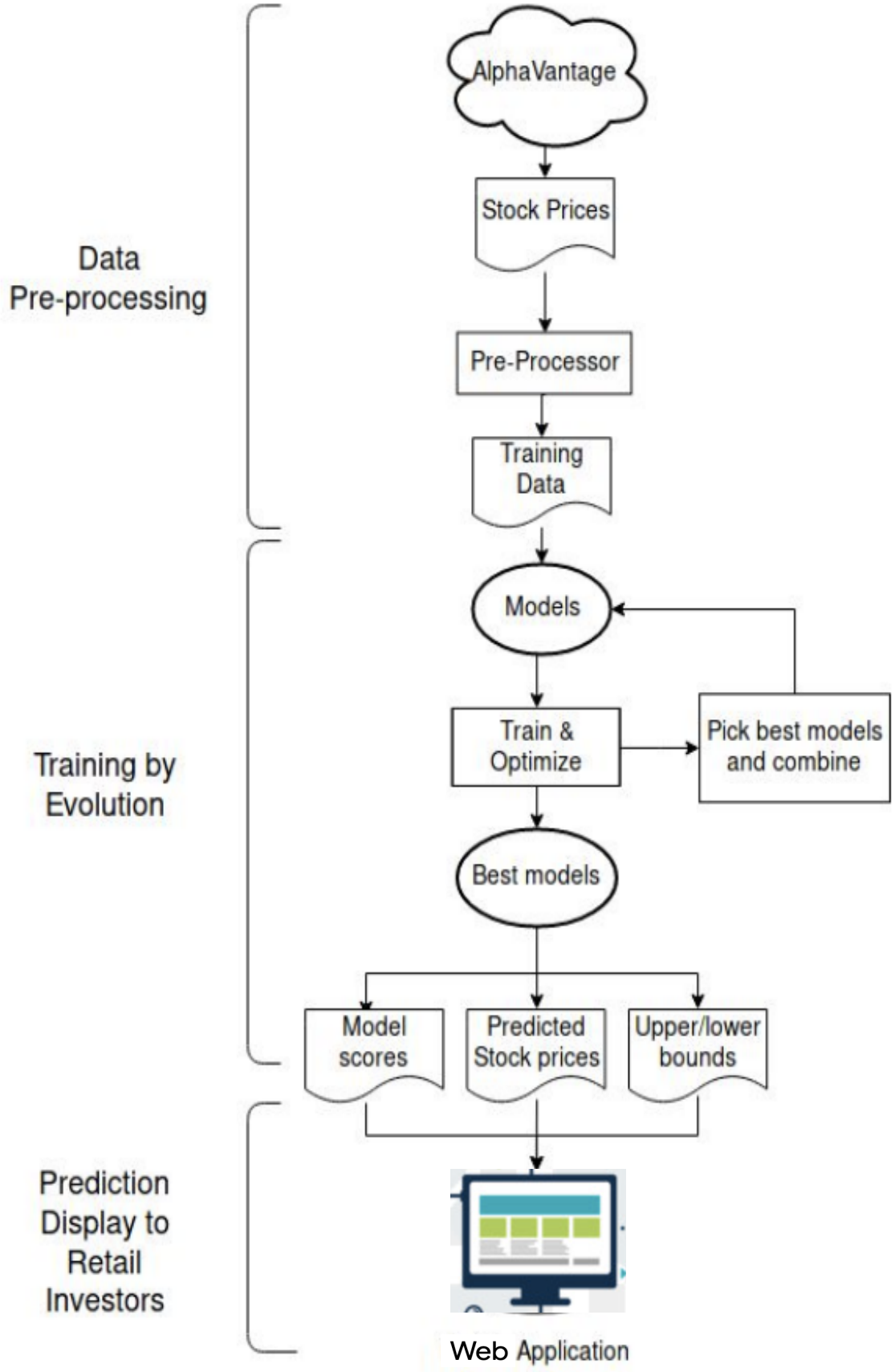
3.

3.1. System Architecture

The architecture of the system follows a client-server model, where the server and the client are loosely coupled.

3.2. Methodology - Design

The system design will be plotted in image format below. Upcoming application data processes will be like that


Data
Pre-processing
AlphaVantage
Stock Prices
Pre-Processor
Training
Data
Training by
Evolution
Models
Train &
Optimize
Pick best models
and combine
Best models
Model
scores
Predicted
Stock prices
Upper/lower
bounds
Prediction
Display to
Retail
Investors
Web Application

3.3. Data Pre-processing

Raw stock price data is pre-processed before inputting into machine learning models. Pre-processing includes transforming the raw data into a format that models can take from and operate on, most likely a feature matrix. It also attempts to extract some features, financial-domain-specific especially, manually to improve results, allowing the model to learn more abstractions.

3.4. Model

Different common neural network models should be tested, including dense neural networks, simple recurrent neural networks (RNNs), long short-term memory networks (LSTMs) and gated recurrent unit networks (GRUs).

4.

4.1. Progressive Web Application Motivation

The application will be written as a progressive web application.
The motivation behind this is that the application could be inherently adapted to desktop and mobile usage. It would be more costly to create native desktop and native mobile applications separately.
The web application will also allow the system to keep only one centralized instance, where information only has to be updated once without any duplicated effort.

4.2. Research Implementation

All machine learning-related code will be written in Python. Neural networks will be implemented with Keras while the linear regression model is implemented with scikit-learn.

4.3. Data Pre-processing (testing)

3 Python scripts will be written to transform the raw stock prices (.csv files) into feature vectors, for training, predicting and testing respectively.
The scripts will take the input options and the raw stock prices as inputs and produce the correct features by building the lookback arrays and the moving

averages. It concatenates the features into the final feature vectors, which will be passed to the model for training or testing.
NumPy and Pandas will be used to build the datasets.

4.4. Testing Model

A model base class is used as a common interface for all machine learning models. All models will have their own model class, specifying model-specific details like methods to build the model, train the model, use the model and save the model.

4.5. Data Training

In the training part, a randomized initial model is first generated from the model options definition.
A training set will be generated by the build training dataset script, which generates the training set features from the input options and the raw stock price data.
Then, the data is fed into the model for training.

4.6. Predicting

Stock Price When predicting stock price, the saved model will first be loaded.

5. Evolution Algorithm

5.1. Running Evolution Algorithm

All training will be done in the Jupyter notebook environment. With hardware limitations, apart from each team member's own computer, Google Colaboratory which provides an easy-to-use Jupyter notebook environment and free GPU service is also used to train models and run the evolution algorithm.

5.2. Server - Flask & Dash

For local development and testing, the Flask micro web framework is used to serve local saved data like raw stock price data and saved predictions file. It is written in Python and integrates well with the existing backend architecture.

6.

6.1. Project Requirement List:

1.1.1 Program will allow users to analyze the given stock data.

6.2. Required Hardware and Software

Hardware

● Windows / Linux / MAC laptops for development

Software

● Python 3.6 or latest version with machine learning libraries
(e.g. scikit-learn, Keras, Tensorflow)

● Visual Studio Code / Sublime Text for programming

● Google Chrome for debugging web applications Platforms

● Google Colaboratory for running evolution

● Firebase Hosting for app

● Firebase (Cloud Datastore, Cloud Storage, Authentication, Cloud Functions)

6.3. Prerequisites

A good understanding of the Python programming language.

A good understanding of neural networks.

6.4. Functional Requirements

1.1. System requires certain libraries and correct installation guidelines.

1.2. System requires dataset implementation in order to detect & predict information.

6.5. Modules and Technologies

UI AND OTHER REQUIRED FILES

1. Tkinter (UI) 2.Chrome Browser 3.Google Collab & Jupyter

7. Development

7.1. Future use

Program can be developed as a mobile application also rather than web application format.

It's really good practice software for those who wants to learn Machine Learning & Data structures in 2022.