# Predicting a Stock’s Closing Price:

# Can You Beat a Trained Model?

# Introduction

## Coders

Sung-Min Chang (19-616-226)

Gian Berwert (19-606-441)

Adrien Renner (19-615-566)

André Krämer (19-608-389)

## Objective of the group project

This project aims to predict a stock's closing price based on its opening, low, and high price of a day. For this purpose, we train and test a model, which is based on a Long Short-Term Memory (LSTM) with Keras.

At first, the user can choose the stock whose closing price should be predicted. Before the forecasted price of the model is revealed, the user can predict the stock price himself/herself to compete with the machine learning model. To do so, the stock's opening, high, and low price is shown. Eventually, both predictions are compared to the actual closing price and the winner is determined. Importantly, the user and model are asked to predict the closing price of the last trading day as the actual closing price is required to compare both predictions.

With our code, we enable the user to predict closing prices from 5’106 US stocks from the three major US indices (S&P500, Nasdaq, Dow Jones Industrial Average) in total.

## Assignment Deliverable

The deliverable of this group project is the following file: Project.py

# Execution of the code

Below are all steps the user must follow to get the predicted closing price of the model:

**Prerequisites**

1. Open this [link](https://github.com/Eqira/HSG_Programming_Project) to get access to all documents in the GitHub branch and open “Project.py”
2. Copy the raw content by clicking this icon  and paste it into your IDE of choice such as Visual Studio Code
3. Install all necessary modules (quandl, yfinance, yahoo\_fin, keras, pytz, holidays, matplotlib, pandas, sklearn, numpy, tabulate) by running commands in the anaconda prompt in the following way:

pip install [module]

**Running the code**

1. When you run the code, you are first asked to enter the ticker of the stock whose closing price you want the model to predict. Please remember to only input valid tickers, for example, AAPL for Apple Inc.
2. Table

   Description automatically generatedGiven the validity of your input, you are then shown a summary of different numbers of your selected stock, which include prices, dividends, stock splits, and the traded volume. Please find a sample summary of the Apple stock below:
3. A picture containing text

   Description automatically generatedNext, a table will be displayed, which summarizes the opening, high, and low price of your selected stock for the most recent trading day. Below is a sample table of the Apple stock:
4. Given this table of different prices, you are then requested to give an estimate for the closing price (in USD) of the same trading day. Please enter a valid numerical value when the input field pops up; in this case, an expected price of 133$ was entered. Please note that it may take some time until you are shown the table and asked to enter your prediction. This is because in the background the model is simultaneously trained and tested for the next step when the model will predict the closing price.
5. Table

   Description automatically generatedApart from you, also the trained model will predict the closing price based on the opening, high, and low price, which were also provided to you. Subsequently, a table will appear, which summarizes your estimate as well as that of the model and adds the actual closing price of your selected stock. As such, you can compare the results and figure out if you or the trained model was better at predicting the closing price. A sample table for the Apple stock and a predicted price of 133$ is depicted below.
6. Lastly, the model’s predictions are extended from just the most recent trading day to the last five months. More specifically, the model’s predictions for the last five months are plotted against the true closing prices. Below, there is an example plot for three different stocks each. To allow you the evaluation of the model’s performance, the Root Mean Squared Error (RMSE) and the Mean Absolute Percentage Error (MAPE) are displayed as well. For example, the absolute average deviation from the true closing price (RMSE) was just 1.81$ for the Apple stock; put differently, the relative average deviation from the true closing price (MAPE) was approximately 1%. The blue line depicts the true closing prices whereas the orange line depicts the predicted closing prices.

|  |  |  |
| --- | --- | --- |
| Apple Inc. (AAPL)  Chart, line chart  Description automatically generated | Amazon.com Inc. (AMZN)  Chart, line chart  Description automatically generated | Tesla Inc. (TSLA)  Chart, line chart  Description automatically generated |

# Machine Learning Model

The model, which is based on a Long Short-Term Memory (LSTM) with Keras, is illustrated in three parts.

1. First, our model defines all valid stock tickers, which are later used to train the model. That is, stocks with tickers that have more than four characters and a specific suffix ("W", "R", "P", "Q") are removed. For all stocks with a valid ticker, five years of historical stock price data is retrieved using Yahoo Finance's API. This preselection results in 5'106 stocks from the major US indices, which are used to train the model.

A picture containing table

Description automatically generated

1. Second, for the user-selected stock, its past five-year prices are split into a training set (90% of past data) and a test set (10% of past data). Using the stocks' opening, low, and high price as feature variables, the model trains an LSTM with the stock’s closing prices being the target value. For increased efficiency, a batch size of four is introduced, as well as epochs limited to 100 to reduce overfitting and underfitting risks. Based on a stock's opening, low, and high price, the model attempts to predict the stock's closing price.
2. In the final step the model, which was trained on the training set, predicts the closing prices of the test set. Sample plots of the model's predictions can be found in chapter 2, step 9. Through the described model, we allow the user to predict a stock's closing price based on its opening, high, and low price for any trading day.