August 24th, 2023

C964 Capstone: Stock Prediction

Part A: Project Proposal for Business Executives …………......................................................... 2

Letter of Transmittal ........................................................................................................... 2 Project Recommendation .................................................................................................. 3 Problem Summary ............................................................................................................. 3 Application Benefits ............................................................................................................ 3 Application Description ....................................................................................................... 3 Data Description ................................................................................................................ 3 Objectives and Hypothesis ................................................................................................ 3 Methodology ....................................................................................................................... 4 Funding Requirements ...................................................................................................... 4 Data Precautions ............................................................................................................... 4 Developer’s Expertise ........................................................................................................ 4

Part B: Project Proposal ................................................................................................................ 5

Problem Statement ............................................................................................................ 5 Customer Summary .......................................................................................................... 5 Existing System Analysis ................................................................................................... 5 Data .................................................................................................................................... 5 Project Methodology ........................................................................................................... 6 Project Outcomes .............................................................................................................. 6 Implementation Plan ........................................................................................................... 6 Evaluation Plan ................................................................................................................... 7 Resources and Costs ........................................................................................................ 7 Timeline and Milestones ..................................................................................................... 7

Part C: Application .......................................................................................................................... 8

Part D: Post-implementation Report .............................................................................................. 9

A Business (or Organization) Vision .................................................................................. 9 Datasets ............................................................................................................................. 9 Data Product Code .............................................................................................................9 Objective (or Hypothesis) Verification .............................................................................. 10 Effective Visualization and Reporting ............................................................................... 10 Accuracy Analysis .............................................................................................................10 Application Testing ............................................................................................................ 11 Application Files ................................................................................................................ 11 User Guide ....................................................................................................................... 11 Summation of Learning Experience ..................................................................................12

# **Part A: Project Proposal for Business Executives**

## Letter of Transmittal

August 24th, 2023

Esteemed Directors & Executives,

Currently company savings are underutilized, the money is not being put to work to help us invest and grow. Hiring professional investors would be too much of a drain on these savings to be worthwhile until we have grown more. As such I am proposing an AI stock predictor that will give us predictions on stock prices for the following 7 days. The predictions can be very accurate when only predicting 1 day in advance, with decreasing accuracy the further out it goes. Having an accountant check the predictor daily to inform investment decisions could be a low-risk way to increase gains. Total cost would be fairly low for such a project, as we don’t need a highly sophisticated model, I and one or two other developers on the payroll could complete the project in under 2 weeks using existing Machine Learning libraries. This puts cost estimations around 7 grand, depending on the amount of hourly work put towards the project. I look forward to hearing your decision.  
  
Best Regards,  
Mike Fasnacht

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## Project Recommendation

### Problem Summary

The Project would include simplified code and instructions to be easily modifiable code for non-technical individuals, to keep developmental costs down, but may be expanded upon in the future if desired. It would allow non-technical employees to easily modify variables in the code to predict various stocks, with data from the Yahoo finance API, and allow users to create predictions and display them graphically with a graph, without needing to understand the underlying code. This project would increase gains on excess company savings, while still keeping these funds fairly liquid. Currently, company funds simply accrue interest in savings accounts, if this project is greenlit it would allow our employed accountants to easily make smart, low-risk investment decisions. The project can be finished in under 2 weeks, with a couple of developers, It will be barebones but easy to use with clear instructions. Should this meet expectations, it can be expanded upon and developed more fully.

### Application Benefits

This project is intended as a barebones trial to determine the value of pursuing AI-driven investment. The goals are to keep costs and deliverables to a minimum while showcasing the value of machine learning stock prediction. Ideally, it will also prove to be a valuable asset for increasing company growth by making use of company savings and providing more return than interest can, but the main goal is to prove AI stock prediction is a worthwhile pursuit and to incentivize additional projects, where if the initial project does not greatly increase return on savings/investment future projects will.

### Application Description

The application will provide more accurate stock predictions than could be reasonably expected of any human. This allows us to make more risk-free investments and substantially increase returns. It does this by pulling historical stock information from the Yahoo finance API, to train a linear regression algorithm and more current stock information to make predictions. A graph of actual and predicted prices for each day over the 7-day prediction will also generated for testing and evaluation purposes.

### Data Description

The raw data will come from the Yahoo Finance API. This data contains the date of each data entry, the open, high, low, close & adjusted closing price, as well as the volume traded for that day. From the date we grab integers representing the year, month, day of the month, as well as the day of the week so that the date can be used for training the algorithm. From the high and low prices, we get the daily price range, from the closing price of previous days we generate a moving average for both the last 5 and the last 20 days. Then we calculate the relative strength index (RSI) based on the closing prices. Finally, the target variables are filled in based on the closing price of dates ahead of it.

### Objectives and Hypothesis

The goal of the project aside from highlighting the value machine learning can offer, is to predict the closing price for the following day and each following day up to 7 days in the future, excluding weekends. While there is no explicit prediction accuracy target, the goal is for prediction accuracy to be very accurate for the first several days’ predictions, relative to the value of a stock. (ex. Being $10 dollars off on a stock worth $50 is a much bigger problem than on a stock worth $300)

### Methodology

For this project Rapid application development (RAD) will be used. This methodology was used due to the low computational complexity of this stock predictor, as more advanced versions would be more expensive and more difficult to convince stockholders of its worth without first seeing what a simpler implementation is capable of. First requirements are defined, in this case, a simplistic prediction model with visuals to make a judgment of the broader prediction accuracy more easily understood. Next, a Prototype is constructed, arguably in this case the finished product will be the prototype before proposals for more advanced projects are submitted. Next is the construction stage in which engineers and developers work to flesh out the prototype into a more cohesive program, bugs are fixed, feedback addressed, etc. Finally, in the deployment stage, the technical documentation is written, scale testing is conducted, final changes & customizations are made and final debugging tasks are carried out before the project is deployed, or in this case, results submitted to stakeholders.

### Funding Requirements

As this project is essentially a probe into the usefulness of AI as an investing tool, It is being kept minimal to keep costs down and make it an easier sell to stockholders. As such the only funding needed is the usual pay for myself and two other developers for 2 weeks and our usual development environments, putting costs around 7 grand.

### 

### Data Precautions

No sensitive data will be used as the stock information is publically available through the Yahoo Finance API and many other sources.

### Developer’s Expertise

My expertise comes mostly from my Computer Science degree from WGU, School coding projects as well as personal projects have given me the experience to write programs such as the one being proposed.

# **Part B: Project Proposal**

## Problem Statement

Excess company savings/funds sit unused, passively generating interest. It could be put to better use without needing professional investors through the use of machine learning stock predictions. Prediction algorithms are capable of much higher accuracy than any human and can be improved over time to provide predictions that can not only match but also far exceed the potential earnings of standard interest rates.

## Customer Summary

In this scenario, our company and its executives are the clients, should they choose to accept this proposal. The proposed application can predict stock prices for investment strategies with more accuracy than a human and can make for relatively safe investments. With these prediction algorithms company savings can be put to better use to help improve capital gains and company growth.

## Existing System Analysis

Currently, no application is used for this purpose, or fills a similar role, only the savings accounts in which funds are stored and generate interest, While simpler and cheaper this method is not maximizing potential earnings. An algorithm to predict stock prices and assist investment decisions could increase returns substantially by providing predictions based on historical prices trade volume and other factors that could affect prices, to not just determine directional changes. But also to estimate what the price may be as well.

## Data

The raw data sets are pulled from the Yahoo finance API, namely the stocks under the DOW, from 2013 to the end of 2022. For each day within that range of a particular stock, we collect the prices for the Open, High, Low, Close, and Adj. Closing price, as well as the volume traded that day. Once all the data is collected and cleaned, new values are extrapolated from the existing ones, including breaking the date column into integer values for year, month, day, and day of the week. a Daily Price Range from the high and low of each day. A Moving Average (both 5 and 20-day averages) from the closing price of the previous days, and finally the Relative Strength Index (RSI) to determine when a stock is being overbought or oversold, and help the algorithm detect patterns that are not evident from prices alone. It compares the closing prices in the date range and stores positive and negative values as well as average gain and loss over that window and uses them as a ratio to give us a sense of how much the price goes up compared to how much it goes down over a chosen period. Finally, this ratio is mapped on a scale of 0 to 100 with the formula RSI = 100 - (100 / (1 + RSI)) and helps suggest if a stock is being overbought or oversold.

## Project Methodology

While no official deployment is applicable to this project, in this case, deployment means a showcasing of prediction results to stockholders, to reach that point a Rapid Application Development methodology will be used and adapted for our purposes. Because we’re aiming for the simplest implementation of a Stock predictor as a proof of concept for executives, the low computational complexity makes RAD a good fit. First requirements are defined, in this case, a simplistic prediction model with visuals to make a judgment of the broader prediction accuracy more easily understood. Next, a Prototype is constructed, which in our case is most of our final product, which is why I believe RAD to be a good fit. Next is the construction stage in which engineers and developers work to flesh out the prototype into a more cohesive program, bugs are fixed, feedback addressed, etc., which in our case is minimal as it will only include some bug fixes and slight code restructuring for ease of use. Finally, in the deployment stage, the technical documentation is written, final changes are made, and a detailed and easy-to-follow user guide is written, since this product requires interacting with the code to make changes, lacking any traditional GUI for making user-end adjustments.

## Project Outcomes

The final application will be written in Jupyer Notebook with the option to predict any stock from the DOW by changing a variable that will be explicitly notated. It will generate CSV files containing all the data used and stored and will generate 7 simple line graphs to visualize the prediction accuracy of the selected stock, 1 for each target variable being predicted (Next day prediction, and the other 6 days following). The application and instructions will be kept as simple as possible limiting how much a user must interact with code they may not understand and making sure they understand no changes except those mentioned should be made. The instructions will be kept as simple as possible to avoid confusion and explain the installation of prerequisites, and what changes can be safely made to the application, such as the variable to determine a stock to predict. This final product along with its user guide are the final deliverables and are intended to be a proof of concept of what AI stock prediction can do.

## Implementation Plan

Our ‘implementation’ is a mock-up investment following the delivery of the product. We will perform an experiment where we ’pretend’ to invest money in stocks based on the predictions. We’ll take notes of our initial investment amount and where we invest it, based on what the predictor suggests will give us the biggest return in a week (the 7-day prediction) and check its more accurate 1-day prediction daily to adjust investments as needed, making notes of what our theoretical assets are at close each day, and how much our money has grown or lost overall. Ultimately we will not be making any independent decisions and will only adjust our faux-investments based on the output of the predictor. Once the experiment is over we will use the data to make charts and graphs for a follow-up presentation to hopefully help further sell executives on the idea of AI stock prediction. Ideally, if the executives were not already sold on the idea after the initial delivery, they would be after seeing the results of this experiment and would greenlight future projects.

## Evaluation Plan

Evaluation will be done throughout the project’s development. First, each new code block will be tested to ensure no runtime errors are introduced. Once data is pulled from the Yahoo Finance API, the resulting CSV files storing the data will be examined to ensure the data is in order. The Python library, Pandas, will be used to assist in examining the data ensuring all values are populated and there are no null or missing values. After the preprocessing code is written the updated CSV files will also be examined through pandas, to ensure the data was cleaned and is ready to be split into training and testing groups. Once the test/train split is done the CSVs storing these splits will be examined to ensure the data was split properly (e.g. the first 8 years is a training set, and the last 2 years is a testing set), and that the correct data values are in each split. Once that is done, the sets are fed to the algorithm, and graphs are generated from the results to provide visuals to examine prediction accuracy more easily. Should everything be in order, and the predictions are not deemed too inaccurate, then a user guide is written and the deliverables are presented to executives and stakeholders.

## Resources and Costs

* Windows 10 PC - Already Owned
* Jupyter Notebook - Free (non-hosted, runs on the win 10 pc)
* Project Head Developer, 2 Week salary ~$4000
* Asst Software Dev, 2 Week salary ~$3000

## Timeline and Milestones

All Milestones are carried out with both the Project head and assistant developer and are dependent on the milestones that come before them.

* Development Environment prepared …. 9/11/2023 - 9/12/2023 ~ 24 hours
  1. Includes installation of necessary software and libraries and configuration of the host machine
* Data gathered and cleaned ……………. 9/12/2023 - 9/13/2023 ~ 24 hours
* Feature engineering complete ………… 9/13/2023 - 9/14/2023 ~ 24 hours
  1. Extrapolation of useful metrics from existing data
* Train/Test split complete ……………... 9/14/2023 - 9/15/2023 ~ 24 hours
* Stock Linear Regression implemented .. 9/15/2023 - 9/16/2023 ~ 24 hours
* Visualization added to predictions ……. 9/16/2023 - 9/16/2023 ~ 24 hours
* Accuracy evaluations and tuning …….. 9/19/2023 - 9/21/2023 ~ 48 hours
  1. Testing and tuning prediction metrics or the algorithm if needed.
* User Guide Written …………………… 9/21/2023 - 9/22/2023 ~ 24 hours
* Final Adjustments ……………………. 9/22/2023 - 9/24/2023 ~ 72 hours
  1. Fixing, cleaning, & tuning code
* Project results showcased ……………. 9/25/2023 - 9/25/2023 ~ 2 hours
  1. Presentation of initial Results immediately following project completion

# **Part C: Application**

The only requirements are a python installation, Jupyer notebook, the submitted file ‘MF WGU C964 Stock Predictor.ipynb’ as well as the following libraries:

* Scikit-learn
* Pandas
* Pandas Datareader
* Numpy
* yFinance
* matplotlib

# **Part D: Post-implementation Report**

## A Business (or Organization) Vision

The business problem addressed is the underutilization of company assets, which my proposed application helps improve utilization. The predictor makes fairly accurate predictions when talking about high-value stocks, and even when wrong appears to at least predict the trends of the stock. Using the application a user can create predictions of any of the stocks under the DOW, or more if the stock symbols variable is modified. All a user has to do once the prerequisites are installed is run each code block in order until the final block, simply replace the ‘symbol’ variable with the symbol of the stock they wish to predict, and run the final code block. Once predictions are made of several stocks, a user can determine where the best place to invest might be, so long as the predictions are rerun daily, there is little chance for significant losses as the predictor is very accurate for 1-day predictions.

## Datasets

The raw datasets are CSVs of the data pulled directly from the Yahoo Finance application. The processed data has gone through cleaning and preprocessing. Specifically, the date column has been broken into separate integer values for year, month, day, and day of the week, a daily range column with the price difference between the low and high, a moving average for the last 5 and 20 days was also added, calculated from the closing price of the previous days, and finally an RSI (relative strength index) calculated from the closing price of all entries. Finally, target variables 1 through 7 (represents how many days are forecasted) are added as columns, and filled in with the following day's closing price. Once all of this is done, the dataset is split into a training and testing group, the training group contains all variables including the target variables, and the training set only has access to variables available for prediction. This data is saved as a CSV and fed to the algorithm so that it can predict one day at a time, the closing price for the following 7 days, and it does this for every date entry in the data set.

## Data Product Code

The code for data analysis:

*stock\_symbols = ['AAPL', 'MSFT', 'GOOGL', 'AMZN', 'META']*

*for symbol in stock\_symbols:*

*file = f"{symbol}\_stock\_data.csv"*

*data = pd.read\_csv(file)*

*print(f"\n\n\n{symbol}\n")*

*print(f"Head\n{data.head()}\n")*

*print(f"Info\n")*

*display(data.info())*

*print(f"\nDescribe\n{data.describe()}\n")*

The code above was used to ensure data was clean and prepared before and after each step in preprocessing to ensure clean workable data was given to the final algorithm. After the prediction algorithm runs, The prediction results from y\_test, are plotted on line graphs to provide a visual of how accurate or inaccurate the predictions were. The following code used (plt is matplotlib):

*for i in range (y\_test.shape[1]):*

*plt.figure(figsize=(10, 6))*

*plt.plot(y\_test.iloc[:, i], label=f'Actual Target {i+1}')*

*plt.plot(y\_pred[:, i], label=f'Predicted Target {i+1}')*

*plt.legend()*

*plt.title(f"Actual vs. Predicted Stock Prices for Target {i+1}")*

*plt.xlabel("Data Points")*

*plt.ylabel("Stock Price")*

*plt.show()*

This code displays 7 readable line graphs that compare the prediction with the actual amount, one for each target being predicted (1 day in advance up to 7 days in advance) making it easy to compare results.

## Objective (or Hypothesis) Verification

Ultimately I believe the project accomplishes its goals of proving what a reliable asset machine learning can be. So long as predictions were to be checked daily and investments adjusted accordingly, I believe you could make smart decisions to grow your investments faster than interest on a savings account could. With further development, the predictor could be even more reliable to make predictions further in advance with enough accuracy to avoid daily check-ins for the most accurate predictions.

## Effective Visualization and Reporting

My descriptive methods helped support my non-descriptive methods by informing me of the state of my data and helping me perform data analysis and exploration. The code provided earlier provided a summary of our data to perform the analysis, cleaning, and preprocessing of the data more effectively. Once the project was nearly done, the line graphs that were plotted helped summarize the results in a more readable format.

## Accuracy Analysis

We used RMSE (Root Mean Squared Error) in addition to line graphs for analyzing accuracy. In our context, it roughly equates to the average error in dollars, and we’re usually between 5 and 15, which for stocks worth 100s of dollars, is quite respectable for such a simplistic model. The majority of the errors are accounted for in the longer-term predictions between 4 and 7 days while days 3 and under are fairly accurate, with day 1 being the most accurate by a good margin.

## Application Testing

The application was tested in a standard way for stock predictors. The dataset was split into a training and testing set the first 8 years of data, was used to train the model, and the last 2 years to test its accuracy. I concluded that no specific improvements were necessary for this iteration, as this project was only meant to be a prototype to display the effectiveness of machine learning applications, to incentivize more advanced applications in the future.

## Application Files

* Jupyter Notebook
  + MF WGU C964 Stock Predictor.ipynb (provided)
    - Scikit-learn Python Library
    - Pandas & Pandas Datareader Python Library
    - Yahoo Finance API Python Library
    - Numpy Python Library
    - Matplotlib Python Library

All the source code is contained within the ipynb file and the libraries mentioned. The code in the file is organized in code blocks based on purpose. For example, imports or other code used often throughout the program, a code block to download the stock data from the yahoo finance API, a code block for data analysis and cleaning, another for feature engineering, one for a testing & training split, and finally a block initializing and running the model itself, and plotting the results in a readable format.

## 

## User Guide

1. Ensure Python is installed
   1. You can check if Python is installed by opening a command prompt and entering ‘python –version’ or ‘python3 –version’
   2. If not installed download the latest version from python.org
2. Install pip if needed
   1. Some versions of Python come with ‘pip’ pre-installed, check by typing ‘pip –version’ into the console.
   2. If not installed follow the instructions at <https://pip.pypa.io/en/stable/installation/> and install it
3. Install Jupyter Notebook
   1. Open a terminal/command prompt and run: ‘pip install jupyter’
   2. Detailed instructions can be found at https://scikit-learn.org/stable/install.html
4. Install Scikit-learn
   1. Run ‘pip install -U scikit-learn’ in the console
5. Install Pandas & Pandas Datareader
   1. Run ‘pip install pandas’ in the console
   2. If issues occur more detailed instructions can be found at <https://pandas.pydata.org/docs/getting_started/install.html>
   3. Run ‘pip install pandas-datareader’ in the console
   4. More details at https://pypi.org/project/pandas-datareader/
6. Install the Yahoo Finance API
   1. Run ‘pip install yahoo\_fin’ in the console
7. Install Numpy
   1. Run ‘pip install numpy’ in the console
   2. Details at <https://numpy.org/install/>
8. Install Matplotlib
   1. Run these two commands in console: ‘python -m pip install -U pip’ followed by   
      ‘python -m pip install -U matplotlib’
   2. If issues are encountered, more details can be found at https://matplotlib.org/stable/users/installing/index.html
9. Open MF WGU C964 Stock Predictor.ipynb in Jupyer Notebook
   1. Run Jupyer notebook by pressing the windows key and searching for it, click run when found
   2. Navigate to the directory where the file is stored and run it (recommended to keep it in its own folder as it will create many other files)
10. Do not change anything except when explicitly stated, or you may break the code
11. Navigate to the first code block, press Shift + Enter to run the block of code
12. Do this for the next 4 blocks of code until reach the last block of code
13. Locate the line of text that says:  
     *symbol = “AAPL”* The text following the ‘=’ may be different, but this is the correct variable.
    1. Change the variable (if desired) to any of the symbols provided in the comment line above it (AAPL', 'MSFT', 'GOOGL', 'AMZN', 'META') these are the stocks that may be predicted
14. Once you’ve selected what stock you wish to predict, run the code block by pressing shift + enter
15. Review the results

## Summation of Learning Experience

My only previous experience with Machine Learning has been the Intro to AI course, which was not very enlightening. This project required a lot more research and study and watching many courses on Pluralsight before I felt confident to actually create the project, but overall it ended up being much simpler than I expected it to be. I used a couple of additional learning resources for this project, including numerous courses on Pluralsight (I wouldn’t be able to recall them all, as I jumped around a lot), a few YouTube tutorials, and I also made extensive use of the official documentation on the library websites, such as the scikit-learn website, the pandas website, matplotlib, and the other libraries used. I had expected to have to create my own prediction algorithm instead of using one provided by a library like scikit-learn. In the future, I’d like to pursue my own personal projects to expand on this one potentially even creating my own prediction algorithm, or at least refining the ones provided by sklearn or other libraries to create more accurate predictions.