Kailun Wang
Capstone Project Proposal

1. Domain Background
    - Essentially, this project is making a stock price indicator that takes historical stock trading data and outputs predicted future stock prices for given queries.

    - Broadly speaking, financial sector is one of the earliest domains which aggressively attempt to adopt new technologies with the incentive of higher profits. Since the financial data on stocks, commodities or interests rates are rich, labeled and well-documented, so with simple preprocessing on those data, they will be qualified fuel to drive artificial intelligence that serves as a necessarily complementary part of humans in the financial sector. Financial service companies like Goldman Sachs, Morgan Stanley have been using machine learning for years to better observe markets and organize inner resources. Hedge funds like Renaissance Technology, Citadel LLC or Bridgewater Associates have been intensively hiring software engineers, mathematicians and data scientists to improve their trading algorithms. In contrast to traditional floor trading or phone trading, this is a significant leap.

    - Individually speaking, if common people plan to invest in stock market in order to appreciate their cash assets, many of factors are needed to be considered. Questions like what stocks will be in the investment portfolio, what will be the predicted prices, how much risks can be taken and etc. will be raised. They would need an additional technical analysis tool for trading strategies. A machine learning algorithms backed stock price predictor will achieve this goal.

    - Personally speaking, this project is a combination of my past investing-related dots and a expansion of new financial technical analysis I have just learned. I like spending time on things value more tomorrow than today and investing is a good field to practice it. I followed insights of value investing masters Warren Buffet and Charlie Munger and purchased stocks, such as Amazon, Nvidia, Facebook, that I think that they truly have brought irreplaceable values to their customers or shareholders. And I was rewarded fairly. Yet, I am still curious if machine can be the assistant on investing. Later on, I learned two legendary founders of the world top hedge funds, James Simon and Ray Dalio. Unlike Buffet's value investing, those hedge fund founders work closely with machine learning and focus intensively on technical analysis like momentum of stocks, volume of trading, trends of stock prices, etc. Also, with the academic support from Udacity's Georgia Tech "Machine Learning on Trading", I have started this journey.

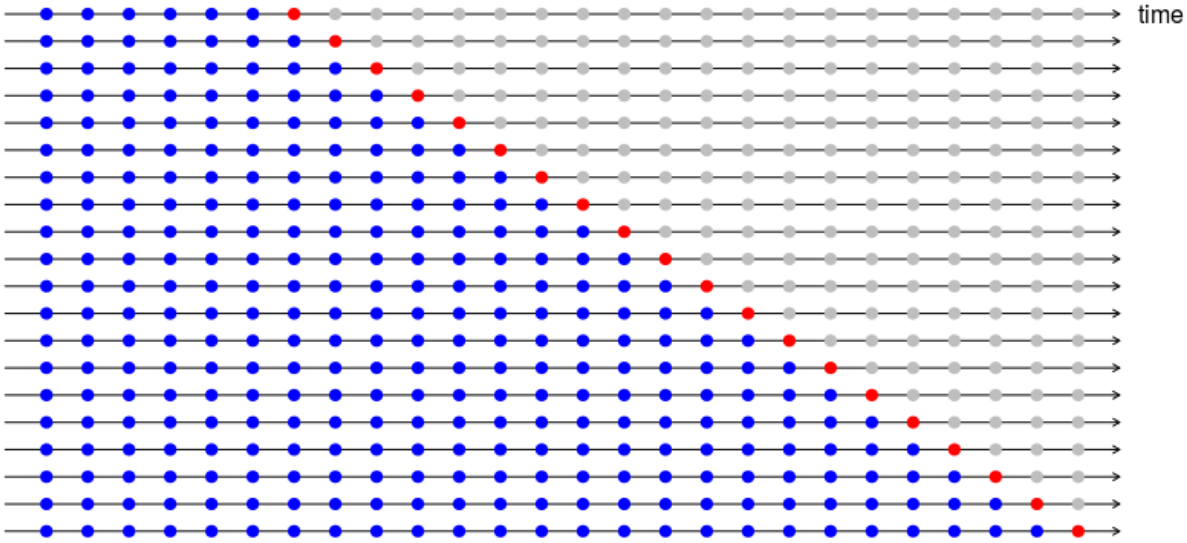
2. Problem Statement
    - Use machine learning models to better predict a next day adjusted close price (not close price, the future predicted value should be converted into present value) over last ten years of trading data for a given query stock. Therefore, comparing with buying stocks without technical analysis, using a stock indicator backed by machine learning can make better forecasts.

4. Dataset and inputs
    - for training algorithms, the dataset will be features of past ten-year stock trading data in CSV format from Yahoo! finance.
        - daily volume of the stock which is an integer
        - daily adjusted closed price of the stock (after adjusted inflation) which is a float
        - candidate datasets:
            1. SP500 (Market Average)
            2. Amazon (Retailing)
            3. Apple (Consumer Electronics)
            4. Berkshire Hathaway (Insurance & Investment)
            5. Google (Internet)
            6. Netflix (Internet)
            7. Disney (Media)
            8. Nvidia (IT)
            9. Tesla (Manufacturing)
            10. Facebook (Internet)
            11. Nike (Apparel)
            12. Microsoft (IT)
            13. Costco (Retailing)
            14. Wells Fargo (Finance)
            15. Starbucks (Catering)
    - inputs will be:
        - stock price momentum (the slope of the price line)
        - Bollinger value (the boundary of price volatility)
        - current stock price

5. Solution Statement

- Solution is to create models that can be used to predict future prices for stocks. First, clean the past 10 years of trading data and get only necessary features (dates, volume, adj close) into a new dataframe.

1. Next, split data in test and train set given a date. (e.g. Feb 14th 2018). Train data is from Jan 5th 2009 to Feb 13th 2018, test data is Feb 14th 2018
2. Split train set in maybe 10 consecutive time folds
3. Train algorithms (linear regression, KNN, SVM and Random Forests) with train data.
4. Train on fold 1 → Test on the first day of fold 2
5. Train on fold 1+2 → Test on the first day of fold 3
6. Train on fold 1+2+3 → Test on the first day of fold 4
7. Train on fold 1+2+3+4 → Test on the first day of fold 5
8. Train on fold 1+2+3+4+5 → Test on the first day of fold 6
9. Train on fold 1+2+3+4+5+6 → Test on the first day of fold 7
10. Train on fold 1+2+3+4+5+6+7→ Test on the first day of fold 8
11. Train on fold 1+2+3+4+5+6+7+8 → Test on the first day of fold 9
12. Train on fold 1+2+3+4+5+6+7+8+9 → Test on the first day of fold 10
13. Compute the average accuracies of the 9 test folds.



- Reference:
  - http://francescopochetti.com/pythonic-cross-validation-time-series-pandas-scikit-learn/
  - https://robjhyndman.com/hyndsight/tscv/

6. Benchmark Model

- Linear Regression model will be the benchmark model here, because it is simple and comparable.

7. Evaluation Metrics
    - Metrics: How well that fund is meeting those goals?
        - Cumulative return
            - in the array of values of our portfolio, then,
            - cumulative return = (val[-1]/val[0]) - 1
        - volatility:
            - how rapidly and aggressively the portfolio goes up and down in value, of course, lower volatility is better
            - volatility = daily_returns.std()
        - risk/reward
            - sharp ratio, also called risk adjusted reward
            - S.R. = Sqrt(252) * [mean*(daily_return - Rriskfree)/daily_return.std()]
        - Mean Squared Error
            - Stock price is a line, so MSE is a good metric to measure the difference between predicted values and true values
            - To use MSE, use command “from sklearn.metrics import mean_squared_error”
        - R2_score
            - Which is r square.
            - This can explain relation between y true and y predicted
            - To use score, use command “from sklearn.metrics import r2_score”

8. Project Design

This project will be divided into five major parts:

- Part1: Overview
- Part2: Analysis
- Part3: Methodology
- Part4: Results
- Part5: Conclusions

Below each of the five major parts, several sections will be added.

Part1: Overview:
Firstly it is domain background. This first section will briefly talk about what this project is all about, and then it will broadly state the relationship between the finance industry and machine learn. Then, it will demonstrate why individuals will need it and personal motivate of creating this project.

Secondly, it is problem statement. This part will elaborate the specific problem this project will solve.

Thirdly, it is solution statement in laymen's terms with supports of graphs.

Fourth, it is dataset and inputs. This part will provide necessary datasets and source where they come from. There should be about 15 datasets of last 10 years trading data from Yahoo! Finance.

Fifth: it is evaluation metrics. To test the project's results, several metrics are needed. 1) Cumulative return. In code: cumulative_return = (val[-1]/val[0]) – 1. 2) Volatility, this measures how rapidly and aggressively the stock goes up or down. In code: volatility = daily_returns.std(). 3) Risk reward ratio, also is the sharp ratio. In code: S.R. = Sqrt(252) * [mean*(daily_return - Rriskfree)/daily_return.std()]

Part2: Analysis,

Firstly, it is data exploration. Each dataset has 7 features: date, open price, high price, low price, close price, volume, adjusted close price. Next, use command dataframe.describe() to get statics. Not all of those features are needed, and only three of them will be fed into algorithms: date, volume, adjusted close price. Before feed data into algorithms, there will be several problems in dataset. The first common problem is the split in stock price due to extra stock offering, and the solution for it will be normalization. The second common problem is incomplete data due to no trading among some dates, and the solution for it will be command fillna() method.

Secondly, it is data visualization. Visualize the 15 dataset and extract characteristics of them. Since there are different techniques to analyze stock prices, there are different ways to visualize stock prices based on those techniques. Chart1 will be the most common way to show prices of those stocks over last 10 years. Chart2 will be daily returns of some stocks comparing with benchmark. Chart3 will be Bollinger Band of some stock to show lower band and upper bank of risks. Chart4 will be correlations between some stocks and benchmark. Chart5 will be the chosen portfolio performance comparing with benchmark performance. Chart6 will be the scatterplots between risk and return. Extra charts will be added.

Thirdly, it is algorithms and methodology. This project will use several algorithms: Linear Regression, KNN, Decision Trees and Ensemble Learners. A brief introduction of each algorithms and the reasons of applying them will be explained.

Fourthly, it is benchmark. The benchmark for this project Standard & Poor 500 (SPY). This benchmark is a reflection of the average market performance. Since the goal for this project is to help its users or investors to achieve a performance better than the

market, so this is a fit candidate benchmark to represent the average market performance.

Part3: Methodology

Firstly, it is data preprocessing, detailed steps of how data being preprocessed. Earlier mentioned, only 3 features out of 7 will be used, so put the 3 features into a empty dataframe. Also add more dimensions to the dataframe, because there are 15 stocks. Next, solve the two major problems earlier mentioned before: the split problem and the incomplete data problem.

Secondly, it is implementation. Step1, implement linear regression. Use the cleaned 10 year trading data to train this model. During this process, train_test_split method and cross_validatiion method will be used to improve the performance of the model. Since financial data does not allow to peekinto the future, training data is always before the testing data with multiple trails rolling forward when doing cross_validation. Step2, repeat step 1 with KNN algorithms. Step3, repeat step 1 with Random Forests. Step4, repeat step 1 with SVM. Step4, test those trained models to on testing data and get accuracy scores.

Thirdly, it is refinement. Change hyperparameters of those models to see if the accuracy will improve.

Part4: Results

Firstly, it is model evaluation and metrics. Compare results of those 4 models.

Secondly, it is justification. Compare results with the benchmark.

Part5: Conclusions

Firstly, it is about observations. Talk about what the model has found.

Secondly, it is about improvements. Talk about what potential improvements can be made. First thing might be adding reinforcement learning and related policies to teach the model the certain of stock price change and timing of exit. Also, the correlated stocks of a certain stock should be considered.