Project Proposal: Comparing Methods and Models used to predict Financial Market Crashes using Market Data

Ping-Chieh Tu, Adrien Bélanger, Inigo Torres

November 3rd 2024

Background, motivation and Project Thesis

There have been several crashes in the market history. They also come with severe consequences. In 1929, one of the biggest market crashes in history came before the largest worldwide economic crisis in history. When the market crashes, individual investors often lose money by panic selling which will cause more crashes. Nowadays, machine learning can help statisticians and economists predict market crashes and prevent them from happening. In this project, we will see different methods that are used to predict time series.

For this project, we will define a crash as when there is a decline of 20% or more in a major market index from its recent peak over a short period. We will use this as an indicator to predict market crashed. This is commonly used in litterature.

This project aims to compare several commonly used models to predict market crashes using market data.

Objectives

- Find and correctly implement commonly used models for Time Series Analysis in our context
  a. Linear Regression
  b. Logistic regression, Decision Tree and Neural Network
  c. Autoregressive Moving Integrated Average Model (ARIMA)

  Although Linear Regression might not do well in predicting time series, we wish to see how it will perform.
- Run and Analyze Results on the multiple scenarios across all models
- Compare the results of each model and discuss their strengths and weaknesses. We will compare the models based on their
  a. Accuracy
  b. Efficiency
- Conclude the most favorable approach indicated by our results and compare with literature

Success Criteria

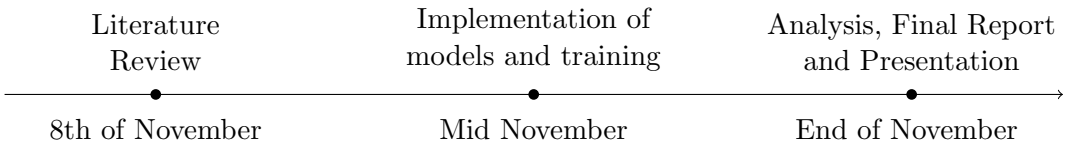
- Implement the models correctly so the empirical comparison between models depends on their strength and not their

- All models should achieve at least the expectation of them found in literature
- Comparison of the models should show their strengths and weaknesses in different scenarios
- Our final report should adequately compare our methods and results with the ones found in literature

Project Plan

- Literature Review: Find models and datasets with which we can work, gather their expected performance and their original implementation
- Implementation of models and training: Implement each model to at least the satisfaction outlined in our Success Criteria
- Analysis, Final Report and presentation

Timeline



Implementation

To implement our project, we will use a variety of historical financial datasets, including stock prices, trading volumes, and volatility indices. Sources such as **Yahoo Finance**, which provides stock market data and historical prices; **Quandl**, offering a wide range of financial, economic, and alternative datasets; and the **World Bank Open Data**, which supplies extensive macroeconomic time series and global financial statistics, provide extensive repositories of financial time series data suitable for our analysis. Additionally, we may go beyond numerical data by incorporating sentiment analysis on articles and social media metadata to enrich our dataset.

Our implementation will involve exploring and implementing a variety of machine learning models to find the best combination possible. Specifically, we will examine Linear Regression for basic trend analysis, Logistic Regression to classify periods as "Crash" or "No Crash," Decision Trees for decision-making pathways, Neural Networks with gradient descent for deep learning-based trend prediction, and ARIMA for large time series forecasting. However, we remain open to adapting and adding more machine learning models as needed. In fact, recent mathematical literature suggests that hidden Markov Chains are present in most time series data, which we intend to explore further. By identifying the strengths and weaknesses of each approach, we weill be able to select the most apropriate models for our final implementation.

We will use Python libraries such as `pandas` and `NumPy` for data manipulation and preprocessing, and machine learning frameworks like `TensorFlow` and `scikit-learn` for model development and training. Regarding computational resources, if needed, we will utilize cloud-based platforms like Google Colab or AWS EC2 instances for training complex models on extremely large datasets.