

deepssharma / Capstone

Type  to search

>

+














<> Code


 Issues


 Pull requests

 Actions

 Projects


 Wiki


 Security


 I

 **Capstone**

Public

 master

 1 branch


 0 tags



















Go to file

Add file






About

Code



	d...	89450b0	6 minutes ago	 20 commits
	figures	Added pdf plot	6 minutes ago	
	logs	Final plots and fig...	16 minutes ago	
	logs_...	Added Predicted f...	5 days ago	
	models	Final plots and fig...	16 minutes ago	
	model...	Final plots and fig...	16 minutes ago	
	plots	Final plots and fig...	16 minutes ago	
	plots_...	Added Predicted f...	5 days ago	
	.DS_S...	Added figures	yesterday	
	.gitign...	Added Presentation	14 minutes ago	
	EDA.i...	Final plots and fig...	16 minutes ago	
	READ...	Update README....	36 minutes ago	
	Stock...	Added pdf plot	6 minutes ago	
	Stock...	Final plots and fig...	16 minutes ago	
	Stock...	Final plots and fig...	16 minutes ago	
	lstm_...	Cleaned up the ma...	5 days ago	
	predic...	Final plots and fig...	16 minutes ago	

No description, website, or topics provided.

-  Readme
-  Activity
-  0 stars
-  2 watching
-  0 forks


Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Languages

-  Jupyter Notebook 100.0%

prese... Added Presentation 14 minutes ago

☰ README.md



Modeling Stock Prices and Portfolios Prediction - Capstone Project

Business Problem

We would like to invest some money into stock market and have a portfolio that will maximize returns with as little risk as possible. We therefore want to minimize the risk involved while maximizing the profit.

Dataset Information

The data was obtained from Yahoo Stock Finance using the python inbuilt library *yfinance*. We looked at the top 29 companies by weight in the S&P index (<https://www.slickcharts.com/sp500>), and this study includes modeling those 29 stocks and predicts portfolios with them. The symbols and abbreviations can be found at the above link (<https://www.slickcharts.com/sp500>).

- The chosen order of symbols in the list is based on their weights in S&P index as described in the same link.
- We chose a period of 10 years to look at the historical data. Also this period is used so that all the listed companies have data for the selected period.

Analysis Approach

- This is a multi-step problem that can be divided as follows:
 - First step us to build models that can predict the stock prices. The idea is that even though the model can not get the stock prices right, it should be able to predict the general trend of ups and downs in the stocks movement. Ideally we should use a few models to find the once that describes the trend best. But for now We are going to use only the below listed models:
 - Stacked LSTM model
 - ARIMA/GARCH models: done here only as exploratory analysis and will be included later as keep improving the code quality.
 - Once we have a reliable model, we will generate predictions and then calculate returns on stocks and eventually build profitable portfolios.

- We will use **Shapre Ratio** and also check **Volatility** as the measures to predict portfolios.
- One needs to include sentinet analysis as well to understand the effect of news and other factors on stocks price movement. However, at this point, we have not included it due to lack of time. .
- Other thing to include is the information contained in the SEC filings of the companies and incorporate that into models.

Data Preparation

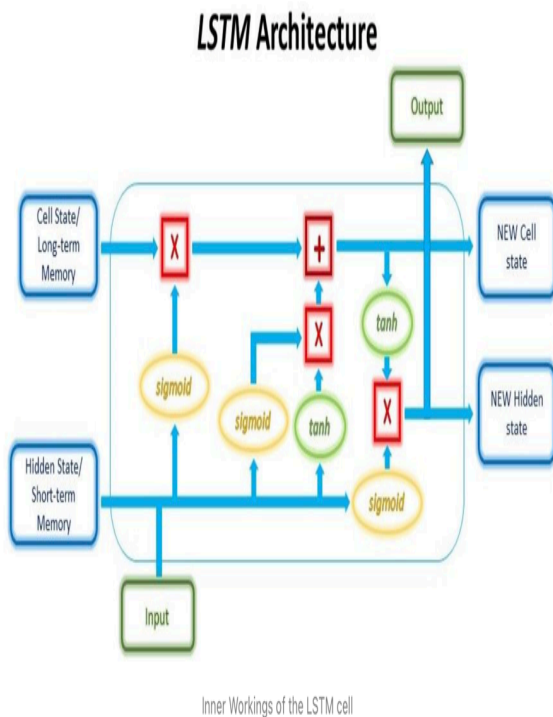
- The data contains "Open", "Close", "Low", "High", "Adjusted Close" and "Volume" entries for each stock.
Those values refer to the :
 - Open: stock opening price at the begining of the given day
 - Close: stock closing price at the end of the given day
 - Low: stock lowest price reached on the given day
 - High :Stock highest price reached on the given day
 - Adjusted Close: stock closing price on the given day after adjusting for various factors such as dividends distribution, applicable splits etc etc
 - Volume: Number of shares traded

- Checked for stationarity in the time series by looking at auto-correlations, qq-plots, and adfuller test with and without transformation techniques (this is important to know if one builds ARIMA/GARCH models).
- Added several **Financial Indicators** to the data by using the python Librray **Finta**. Details can be found at: (<https://pypi.org/project/finta/>).
- The data was split into train, test and validation datasets. Since this is a time-series data, we can not shuffle data randomly. We used **Keras TimeSeries Generator** to build the data samples that preserve the time-order information.
- Used TensorFlow **TensorBoard** utility to save models, plots and logs etc.

Stocks Prediction using LSTM Model

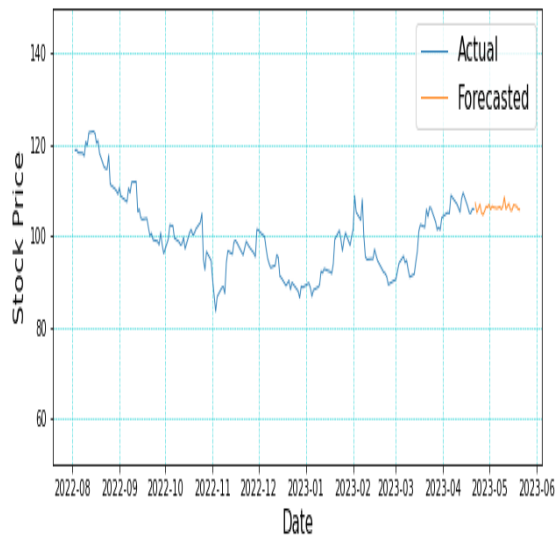
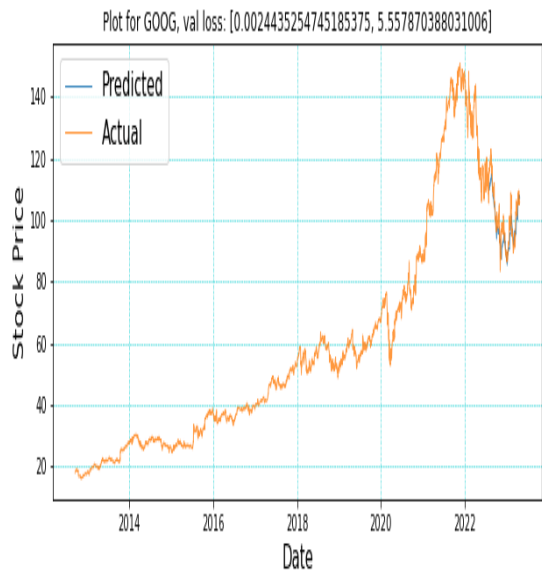
(https://github.com/MohammadFneish7/Keras_LSTM_Diagram)

- We will use the **Long Short-Term Memory (LSTM) model**, a common deep learning recurrent neural network (RNN) used in predicting time series data. The diagram credit goes to (<https://blog.floydhub.com/long-short-term-memory-from-zero-to-hero-with-pytorch/>)



- LSTM has logic gates (input, output and forget gates) which give inherent ability for it to retain information that is more relevant and forgo unnecessary information. This makes LSTM a good model for interpreting patterns over long periods.
- The important thing to note about LSTM is the input, which needs to be in the form of a 3D vector (samples, time-steps, features). Hence, the input has to be reshaped to fit this.

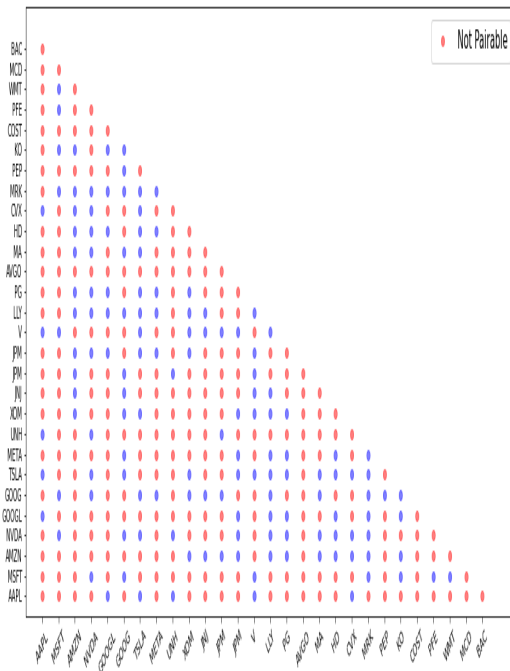
An Example of Model fitted to GOOGLE stocks Data and Prediction



* Model fits to rest of the stocks can be found under plots sub-directory.

Pairable Stocks

Only those stocks are paired to each other that have correlation below 0.5 and a covariance smaller than their individual values. If a pair of stock meets both the Covariance and correlation thresholds then get paired together. The plot below shows pairable stocks in blue and unpairable ones in red.



Risk-Adjusted Returns: (Sharpe Ratio and Volatility)

The **Sharpe ratio**—also known as the modified Sharpe ratio or the Sharpe index—is a way to measure the performance of an investment by taking risk into account. The Sharpe Ratio is calculated by determining an asset or a portfolio's "excess return" for a given period of time. This amount is divided by the portfolio's standard deviation, which is a measure of its volatility. To calculate the Sharpe Ratio, we use the below listed formula:

- Sharpe Ratio = $(R_p - R_f) / \text{Standard deviation}$ where:
 - **R_p**: return of portfolio/mean return,
 - **R_f**: risk-free rate of 3.7% ie current risk free rate of US market,
 - **Standard Deviation**: standard deviation of the portfolio's excess return,

Sharpe ratio > 1 is considered **Good**, > 2 is considered **Very Good**, and > 3 is considered **Excellent**

Top 3 Portfolios:

Below one can see all the stocks that can be paired to each other that have correlation below 0.5 and a covariance smaller than individual thresholds.

Using Shapre Ratio and Portfolio Volatility, this analysis yields the following three best portfolios:

	Portfolio	Weights	Sharpe Ratio	Portfolio_Returns	Portfolio_Volatility
0	[AMZN, META, JPM, LLV, MRK]	[0.06165540540540541, 0.1258442845945946, 0.10219594594594594, 0.4028715216216216, 0.30743243243243246]	2.582864	11.64	35.074324
1	[NVDA, MRK]	[0.31555555555555553, 0.6844444444444444]	2.999636	10.46	32.084729
2	[XOM, GOOGL, JPM, LLV]	[0.134029590948651, 0.13228894691035684, 0.35509136381201044, 0.3785900783288817]	2.089932	11.47	33.519018

Conclusions

- We have built a basic framework that:
 - fits LSTM models on all the stocks that are included in study.
 - builds Returns for each individual stocks.
 - predicts combinations of stocks to build profitable portfolios based on Shapre Ratio and Volatility.

Future Improvements

- Check other models such as GARCH, Random Forests etc to get prediction for stocks movement.
- Hyperparameter tuning of Models.
- Explore the models by including few variables versus all the variables that are included in this study.
- Include the sentienet analysis which includes web-scraping for news that affect stocks market.
- Also implement the information from SEC reports submitted by companies.

Repository Structure

└─ figures
Images folder, containing all
referenced image files

└─ plots*
Images folder, that contains
model fits for all the stocks

└─ models*
Folder, that contains model fits
parameters for all the stocks

└─ .gitignore
List of files and folders not to
be uploaded into repo

└─ Stocks_Analysis.ipynb
Main Jupyter notebook, contains
analysis

└─ EDA.ipynb
Jupyter notebook, that contains
exploratory analysis

└─ Notebook.pdf
PDF version of main Jupyter
notebook

└─ presentation.pdf
PDF Version of project
presentation

└─ README.md
The top-level README

Contact Info:

- Email: deeps.sharma@gmail.com
- GitHub: [@deepssharma](https://github.com/deepssharma)
- [LinkedIn](#)