[AAPL\_Analysis.ipynb](https://github.com/QuintessentialChicken98/Colab-Projects/blob/main/AAPL_Analysis.ipynb)

The quantitative analysis portion indicates AAPL’s consistent outperformance of SPY, which is accompanied by its greater volatility as represented by its Beta, standard deviation and returns distribution. The Sharpe ratio and excess returns facilitate one’s analysis of whether the excess risk on the benchmark is justified regarding the excess returns.

The Forecasts portion demonstrates Python’s capabilities applied machine learning models for the prediction of stock price movements (of course this, assumes inefficient markets), wherein the two machine learning models (Long Short Term Memory and Support Vector Regression) are compared against a traditional ARIMA, with the ML models providing more accurate and more flexible forecasting capabilities. Such forecasting may be useful in short-term forecasts of yield curves and interest rate movements, in an active fixed income strategy.

The Technical Analysis portion evidences the opportunities for TA with python using two libraries (pandas\_ta and vectorbt) with the former providing signals for TA strategies and the latter back testing them, whilst also facilitating advanced back testing methods such as Monte Carlo techniques to find the most effective strategy over a specified period, or here, a 3,23 day moving average crossover strategy affording a Sharpe ratio of 2.10 and a total return of 768%.

The Sentiment Analysis portion obtained a novel dataset of tweets referencing $AAPL, for which I utilised a Natural Language Processing library (TextBlob) to assess the tweets’ sentiment and polarity (strength of sentiment). Whereupon I aggregated the sentiment polarity across the three groups (positive, neutral and negative) to arrive at an aggregate sentiment which I plotted underneath the close price to assess whether the relationship between twitter sentiment and close price qualitatively. I then further assessed this relationship, focusing on the effect of each class of sentiment (positive, neutral and negative) upon the close price in an OLS regression, for which I found positive sentiment to be statistically significant in its effect upon the close price. This, however, was a pet-project, with little use-case in a strategy, one believes the VIX would provide a better indicator for aggregate macro-economic sentiment for informing any pre-emptive strategies.

The Fundamental Analysis portion obtains fundamental information on AAPL from yahoo finance, using the yahoo\_finance and yesg libraries, to intuitively compute and plot both fundamental and ESG analysis of AAPL stock. The findings of the ESG analysis were that whilst AAPL outperformed the S&P100, it underperformed relative to its sector peers. The findings of the fundamental analysis are that its profitability has increased steadily, hence, one concludes that the managers may well have been reinvesting more cash into the business, as the cash ratio has decreased, whilst ROTA has increased, and, due to this improved financial position, are preferring debt to equity in order to not dilute the stock price and retain ownership. Digressing, Pecking Order theory would indicate that this implies the managers are optimistic lest they would not have opted to issue more debt.

The Credit Risk Evaluation portion is more experimental than the previous sections, attempting to implement Merton’s Contingent Claims Analysis, however, in lie of a more refined recovery rate, the analysis of this portion is mitigated and, as such, it was omitted from the generalised version of this notebook. However, when analysing the credit risk of a company, contingent claims analysis may provide a unique insight when effectively implemented, which would bolster further qualitative and quantitative analysis. Similarly, the Implied Volatility Smile of a low maturity option chain provides a further intuitive assessment of AAPL’s inherent implied risk, with it being more likely that the stock increases in price in the short run (and greater strike calls go In The Money), than it decreases.

[Ticker\_Analysis.ipynb](https://github.com/QuintessentialChicken98/Colab-Projects/blob/main/Ticker_Analysis.ipynb)

This builds on the above analysis, instead facilitating analysis of any ticker, with the data starting from the day of activation to 5 years prior thereby future proofing the analysis. It opts instead for only the LSTM model because this was the machine learning time series analysis model I understood best at the time of publishing; and implements a 5 day forecast. Currently, this is in python notebook (.ipynb) format, however, it would be possible to convert this into an executable python (.py) file and build it into a fully fledged app utilising the tkinter library. I am currently inexperienced with this library but would happily dive into learning it and building apps to expedite analysis.

[Bonds.ipynb](https://github.com/QuintessentialChicken98/Colab-Projects/blob/main/Bonds.ipynb)

This was an exercise in practicing bond valuation methods and provides a salient example of the scraping capabilities of Python. One scraped a table of UK gilt yields at various maturities from <https://uk.investing.com/rates-bonds/uk-government-bonds> in order to plot the yield curve as it currently stands. Where one to be able to obtain historical yield changes for every maturity of gilt and potentially explore whether there was a significant correlation coefficient between yield and the FTSE100, for example, one may be able to forecast short term changes in yields by forecasting the FTSE100.

I hope this document goes some way to evidencing my interest in learning and applying further python and machine learning methods to, as well as Python’s suitability for, financial analysis. I have only been learning Python for one year and am majority self-taught since February wherein I had to teach myself Pandas to undertake my Dissertation. Python does have PDF scraping capabilities, which may also assist in expediting analysis: <https://www.databricks.com/notebooks/esg_notebooks/01_esg_report.html?TB_iframe=true&width=370.8&height=658.8>,

<https://geekyhumans.com/how-to-extract-text-and-images-from-pdf-using-python/#Step-3-Writing-the-code>,

<https://towardsdatascience.com/pdf-preprocessing-with-python-19829752af9f>.