

Task 2.1

The strategy that was provided through <https://quantpedia.com/strategies/value-book-to-market-factor/> link, was built and backtested through Quantconnect website. The resulted backtest was Extracted, Transformed, and Loaded into PostgreSQL database for future reference, data upkeep, and ability to analyse and plot via 3rd party Software. Additionally it could be useful as a part of larger pipeline, where other data streams into the database such as: sentiment/news analysis, other backtests that were done on different datasets, markets, and platforms. Overall, storing backtest data into a relational database might be beneficial. Code for ETL process can be found in ETL.ipynb file. Backtest data for the base algorithm can be found in Base.json. The code for algorithm can be found under Base.py

When it comes to assessing the algorithm, regular “front page” metrics found on Quantconnect do a great job, lets focus on several and figure out what it means for our model. Its important to note that backtest results stop at ~ beginning of 2011 due to free-tier limitation of 10000 orders.

Net Profit: For our model Net Profit totalled 552.172%, this value tells us the profit earned by an investment after deducting all expenses and losses. So, 552% would mean that we earned 552% of the initial investment amount over the course of backtest. This metric is probably the most focused and beloved by all as it shows how profitable the model was, taking into account all losses incurred and can serve as an important metric. Although, it does not take into account risk which despite great net profit might be too risky for some investors. Following metrics will help us understand how risky our model is:

Net Profit
552.172%

Sharpe Ratio: for our model Sharpe ratio equalled 1.097. This means that model has generated 1.097 units of excess return for each unit of risk taken, thus, model has returned slightly more returns for the level of risk taken. 1.097 is not a bad value but could use some improvement. This metric is important for us to use since it gives us an idea of returns that are adjusted for the risk taken. Important to note, that Sharpe ratio assumes that returns follow a normal distribution, which may not always be the case.

Sharpe Ratio
1.097

Alpha: for our model Alpha was equal to 0.134. This means that our portfolio has outperformed benchmark by 0.134% over the course of a backtest, after adjusting for the risk taken. This is a good metric for us to use as it gives us an idea of our returns when adjusted for the risk.

Alpha
0.134

Beta: for our model Beta equals to -0.104. This means that our portfolio has negative correlation with the market, thus, moving in opposite direction of the market. This is great as we can expect portfolio to perform well when market is in a downturn.

Beta
-0.104

Drawdown: in our model drawdown was equal to 30.1%. This metric means that portfolio experienced a maximum decline of 30.1% from its peak value during the backtesting period. This is an important metric as it provides us with the idea of “worst-case” scenario that portfolio has experienced.

Drawdown
30.100%

The Treynor ratio: for our model we have -1.267. This metric means that portfolio returned -1.267 times the risk-free rate adjusted for its beta. Portfolio has severely underperformed the risk-free rate when adjusted for portfolio’s systemic risk. Negative value suggests that investment performed worse than risk free instrument. Following

Treynor Ratio
-1.267

metrics will help us understand the relationship of wins/losses in our model which will help us understand how our strategy performed.

Profit-Loss ratio: our model acquired profit-loss ratio of 1.86. This means that portfolio won \$1.86 for every dollar lost in losing trades. This is great record for our model.

Profit-Loss Ratio

Average Win / Average Loss: for our model its 0.14 and -0.07 respectively, thus, on our average win we earn a profit of 0.14%, and on a loss trade we lose - 0.07%. This metric highlights that we have a favourable risk-reward ratio and potential gains are larger than potential losses.

Average Win	Average Loss
-------------	--------------

0.14%	-0.07%
-------	--------

Loss Rate / Win Rate: our model has 42% and 58% respectively, mean that 42% and 58% of our trades were losing and winning ones respectively, this is a good value and useful metric for us, but its important to keep in mind that it doesn't take into account the size and frequency of trades, so theoretically one can have high win rate but still lose money. Gladly, so far model that we have shows good win rate and good profitability on wins.

Loss Rate	Win Rate
-----------	----------

42%	58%
-----	-----

Expectancy: it is the overall profitability of a trading system that factors in both the win rate and the average win size of the wins and losses. Our model has expectancy of 0.648, which means that each trade has a positive expected outcome of 0.648% of the initial investment. Important to note that it doesn't take into account risk and transaction costs.

Expectancy

0.648

Portfolio turnover: it refers to the amount of buying and selling of assets within a portfolio over a duration of backtest, in our case it is 0.48%, thus, the model replaced 0.48% of its holdings during the test. This is a relatively low value, which means that portfolio is not actively managed and despite the fact that we avoid high transaction costs, we may miss out on market opportunities. This can be increased via more frequent rebalancing periods.

Portfolio Turnover

0.48%

Task 2.2

Task 2.2 was approached the following way: 4 algorithms per stop-loss were created, one with each rebalancing period (Yearly, Quarters, Monthly, Weekly). Totalling 12 algorithms that were each passed through an ETL process, where they were loaded into a database, and appended into a singular dataframe for ease of comparison and further use such as machine learning, part of an automated backtesting report, and general clarity. File with a dataframe and ETL process for them is: "Task 2.2.ipynb", file with the ETL prototyping can be found under "ETLBase", file with the ETL function is "ETLfunction", file with a database password was omitted, Initial provided code can be found under "Base.py". Each json file was saved under corresponding name and code for each one

of them was saved under the same name in a Python file.

	Total Trades	Average Win	Average Loss	Compounding Annual Return	Drawdown	Expectancy	Net Profit	Sharpe Ratio	Probabilistic Sharpe Ratio	Loss Rate	Win Rate	Profit-Loss Ratio	Alpha	Beta	Annual Standard Deviation
Task2	9655	0.14%	-0.07%	18.418%	30.100%	0.648	552.172%	1.097	54.837%	42%	58%	1.86	0.134	-0.104	0.11
15PercWeek	9703	0.02%	-0.02%	-26.807%	25.500%	-0.061	-12.558%	-0.808	6.871%	51%	49%	0.92	-0.157	-0.607	0.21
15PercMonth	9510	0.04%	-0.04%	14.591%	26.800%	0.108	19.881%	0.703	32.136%	48%	52%	1.12	0.083	-0.528	0.18
15PercQuart	9443	0.07%	-0.05%	21.282%	30.100%	0.298	90.180%	0.958	40.322%	45%	55%	1.34	0.134	-0.384	0.16
15PercYear	9023	0.14%	-0.09%	20.563%	31.700%	0.336	559.545%	1.139	59.409%	46%	54%	1.49	0.148	-0.113	0.12
SD60dWeek	8312	0.04%	-0.06%	-83.828%	43.600%	-0.203	-35.226%	-2.14	0.230%	53%	47%	0.71	-0.649	-0.374	0.32
SD60dMonth	8166	0.07%	-0.06%	5.602%	32.200%	0.019	4.660%	0.299	21.392%	51%	49%	1.08	0.111	-1.066	0.36
SD60dQuart	9101	0.13%	-0.09%	91.901%	30.100%	0.360	357.048%	2.476	95.688%	45%	55%	1.45	0.562	-0.865	0.2
SD60dYear	8719	0.19%	-0.10%	29.644%	30.400%	0.733	716.978%	1.293	71.989%	42%	58%	1.98	0.22	-0.491	0.16
2SD120dWeek	8078	0.03%	-0.05%	-88.703%	43.700%	-0.256	-40.534%	-2.413	0.019%	54%	46%	0.62	-0.698	-0.49	0.31
2SD120dMonth	8142	0.07%	-0.07%	5.454%	33.200%	0.019	4.538%	0.296	21.289%	51%	49%	1.09	0.109	-1.039	0.36
2SD120dQuart	9081	0.13%	-0.09%	92.639%	30.100%	0.364	361.161%	2.52	96.224%	45%	55%	1.46	0.567	-0.822	0.24
2SD120dYear	8645	0.19%	-0.10%	29.644%	30.100%	0.723	717.004%	1.295	72.182%	42%	58%	1.95	0.22	-0.489	0.16

It is important to note that due to limitation of a free-tier account, only up 10000 orders were allowed to pass through, thus, limiting the historical data severely for algorithms that relied on more active rebalancing such as Weekly and Monthly trading, thus, rendering data hardly comparable with each other especially for metric such as Net Profit, which is evident by considerably higher Net Profit for Yearly algorithms. Additionally, Yearly and Quarters algorithms produce generally better results in Average Win/Loss, Loss/Win rate. SD60d and 2SD120d Quarters algorithms have a higher Alpha and lower Beta, higher PSR, and considerably higher Compounding Annual Return, higher Sharpe Ration, despite lower Treynor ratio it is probable to conclude that they are safer and produce a better return on investment, despite equal Drawdown. Although, it is important to keep in mind that if both Quarters and Yearly algorithms are compared over the same time period then results could be different, but with the data available they show better performance.