# Quantopian Algorithm

●●●

Edwin Ricardo Mahecha Parra

# Idea

Merge various factors, provided by the platform with the objective of maximize the profits.

- We use Longs and Shorts to keep the investments under control.
- The stocks are distributed over the different sectors (health, technology, etc.) in order to minimize loses if some sectors begin to fail.
- We trade daily.

# Performance

# Factors Used

- QTradableStocksUS: It is a list of the most liquid stocks in the US. This is a requirement to submit the algorithm to the platform.
- Returns: Provide the returns of the stocks in a window of time (in our case, 5 days).
- Value: returns the value of the stocks defined by a mask. The mask in this case is QTradableStocksUS.
- "feeling_returns" and "message_volume" are custom factors that use stocktwits. Stocktwits allows us to determine if a stock is bearish or bullish according to the tweets related to these stocks.
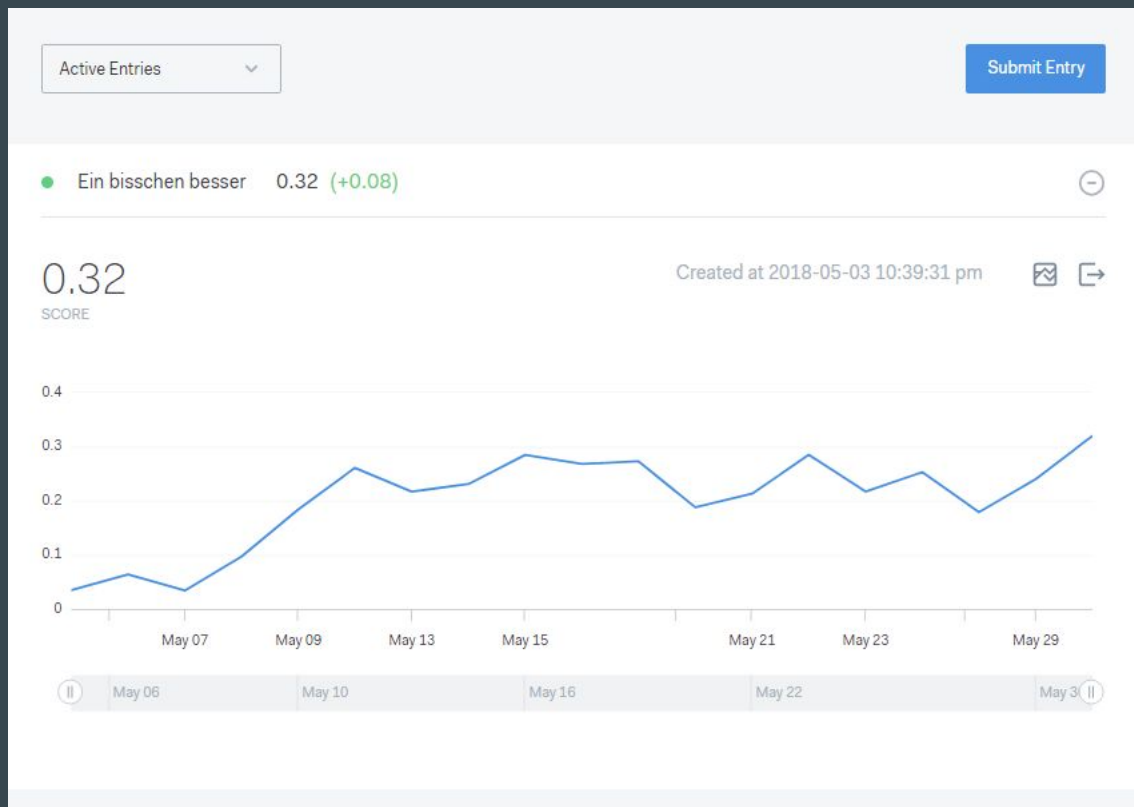
# Pipeline

- We use the feeling combined with the message volume (volume of tweets). The feeling_returns is the difference between bulls and bears.
- We combine the factors using zscores (data standardization) and we take the upper and lower percentiles. These are the stocks we are going to trade.

*Bulls: The market price tends upwards and there is optimism.

```python
def make_pipeline(context):
    universe = QTradableStocksUS()
    recent_returns = Returns(window_length=5,mask=universe)
    value_ret = Value(mask = universe)

    #feeling
    feeling_returns = Feeling(mask = universe)
    message_volume = MessageVolume(mask = universe)
    comb_feeling = feeling_returns*message_volume.zscore();


    unified_alpha = ((recent_returns.zscore() + value_ret.zscore())/2.0 +
comb_feeling.zscore())/2.0

    low_returns = unified_alpha.percentile_between(0,10)
    high_returns = unified_alpha.percentile_between(90,100)

    securities_to_trade = (low_returns | high_returns)

    pipe = Pipeline(
        columns={
            'alpha': unified_alpha
        },
        screen=securities_to_trade
    )

    return pipe
```

# Performance in contest.

End!