Uma imagem com texto, Tipo de letra, captura de ecrã, logótipo

Descrição gerada automaticamente

# **Otimização de Estratégias Orientadas por Dados**

***Algorithmic Trading with Reinforcement Learning - Microsoft***

Bruno Santos nº 125712

Eduardo Bagulho nº 126625

Gonçalo Rosa nº 125684

Mestrado em Ciência de Dados

DEZEMBRO, 2024

## Introdução

O objetivo deste relatório é sumarizar a nossa experiência de interação com algoritmos de aprendizagem, mais concretamente o *Reinforcement Learning* (Aprendizado por Reforço). Para tal, foi proposto, na unidade curricular de Otimização de Estratégias Orientada por Dados, o desenvolvimento de uma estratégia de investimento. Esta estratégia seria, posteriormente, aplicada num modelo de Reinforcement Learning, permitindo-nos analisar a eficiência do modelo ao realizar investimentos no mercado com base na aprendizagem obtida a partir da estratégia definida.

Neste contexto, o relatório abrange os seguintes tópicos principais:

A definição e construção da estratégia de investimento utilizada como base para o modelo, a aplicação do algoritmo de Reinforcement Learning para treinar o agente com a estratégia definida, incluindo a configuração dos Hiper parâmetros, como gamma e épsilon, etc. No final realizamos a análise dos resultados obtidos, avaliando o desempenho do modelo no contexto do mercado financeiro.

Reflexões sobre as vantagens, limitações e possíveis melhorias da abordagem. Em suma, este trabalho tem como objetivo explorar o potencial do Reinforcement Learning para resolver problemas complexos de decisão, como a alocação de recursos no mercado financeiro, permitindo-nos compreender melhor os desafios e benefícios desta abordagem no desenvolvimento de estratégias otimizadas e orientadas por dados.

## Reinforcement learning

O *reinforcement learning* é um modelo *value-based*, ondo o agente irá aprender a partir de uma função *state-action-function* (Q (s, a), com o objetivo de maximizar as *rewards* que são adquiridas por cada ação que execute. Podemos esperar ao longo deste relatório mencionar diversos conceitos como:

* Agente: Sujeito que irá tomar e efetuar as decisões.
* *Rewards*: Recompensa recebida por ação executada, não sendo necessariamente um número positivo.
* Ação: Opções que o agente possa escolher em cada estado.
* Estado: Representa a situação atual do ambiente.
* Política(π): A política funciona como um mapa, onde dita que ações em cada estado.
* Ambiente: Definição de quantos estados e ações iremos possuir para que o nosso agente possa transpor, no nosso caso como iremos definir priori os nossos estados e as ações possíveis de escolha (número discreto de ações e de estados) é considerado um ambiente simples. (Sutton & Barto, 2018)
* *State-value-function* (Q (s, a)): Função que nos permite calcular o valor para um certo estado e ação, com base na reward recebida e a futura ação. Permitindo ao agente perceber que ações são mais benéficas em certos estados.
* Q-table: Matriz constituída pelos estados possíveis (linhas) e ações (colunas) para cada combinação destes iremos ter valores que ditaram a qualidade de tomar certa ação em certo estado. Esta geralmente é iniciada a 0 e será sempre atualizada ao final de cada ação executada.
* Hiper parâmetros: Parâmetros que definem aspetos de aprendizagem do modelo e como este olha para certas decisões e retorno de *rewards*.
  + - Gamma: *Discount factor* ([0;1]), um fator que controla o peso das recompensas futuras no cálculo do valor atual. Ele ajuda a determinar a importância que o agente dá às recompensas imediatas em comparação com as recompensas de longo prazo, ajuda a evitar que valores futuros excessivamente altos causem decisões inadequadas.
    - Learning rate (α), determina o tamanho dos passos que o algoritmo dá ao ajustar os pesos (ou políticas, no caso do aprendizado por reforço) com base no erro ou nas recompensas recebidas, simplificando defini quão rápido o modelo aprende.
    - Épsilon: O épsilon define como o agente escolhe as ações que irá tomar, permitindo que explore o ambiente ao escolher ações aleatórias. Isso ajuda a preencher a *Q-table* com informações úteis, permitindo que o agente aprenda quais ações são benéficas em diferentes estados.
* Exploração: O agente escolhe ações aleatórias para explorar o ambiente e descobrir novas informações.
* Aproveitamento: O agente escolhe as ações que acredita serem mais vantajosas com base no conhecimento atual (ou seja, na Q-table).

O ϵ -greedy é a estratégia mais comum, onde:

* Com uma probabilidade ϵ, o agente explora (escolhe uma ação aleatória).
* Com uma probabilidade 1−ϵ, o agente explora (escolhe a melhor ação segundo a Q-table).

Também é possível como vamos ver introduzir um [decadência](https://www.linguee.pt/portugues-ingles/traducao/decad%C3%AAncia.html) , que permite ao agente explorar mais ao início e ao longo do tempo transitar para aproveitar o conteúdo na *Q-table*. (Sutton & Barto, 2018)

## Estratégia

O primeiro passo na construção de uma boa estratégia de investimento é sabermos naquilo que iremos investir, no nosso portfólio iremos apenas comprar ou vender ações da empresa Microsoft (MSFT), os dados de analise utilizados ao longo do trabalho são adquiridos pelo Yahoo Finance.

Iniciamos a estratégia olhando para os passados 9 anos e verificando como esta ação se comporta, na figura 1 podemos observar que a ação tem uma tendência a crescer, devido ao peso que a Microsoft tem em todo o mundo em diversos setores, juntamente com a ação possuímos dois indicadores chaves fundamentais na nossa estratégia, a *simple moving avarege* de 20 e de 100 dias.   
  
A *simple moving Average*, é um indicador que leva em conta um n número de dias, efetuando a média do preço de fecho da ação nesses n dias, obtendo uma linha menos ambígua do que da ação em si, tendo esta um sentido mais fiel ao desenvolvimento da ação.

Uma estratégica muito básica de utilização com *a moving Average*, quando esta excede a ação no gráfico podemos considerar como um sinal de venda e quando cruza abaixo um sinal de compra. Com base nesta logica que a nossa estratégia funciona.

Outro indicador que utilizamos é o RSI (*Relative Strength Index*), ou índice de força relativa, ajuda a indicar se existe uma sobrevalorização da ação ou desvalorização da ação. Varia entre 0 e 100 e temos como indicadores que abaixo de 30 a ação se encontra desvalorizada e a acima de 70 encontra-se sobrevalorizada.

Uma imagem com Gráfico, captura de ecrã, file, diagrama

Descrição gerada automaticamente

Figura 1- Ação Microsoft com indicadores

A nossa estratégia é muito simples, ao usar todos os indicadores mencionados, calculamos inicialmente o valor de diferença, em percentagem, entre a *moving Average* de 20 e 100, este valor ajuda-nos a entender se o mercado está num período de crescimento ou decrescente, ao qual iremos chamar de *IsTrade.* Com este valor calculado passamos para fase de decisão de compra, venda ou de manter a ação.

* Compra

Compramos a ação sempre que o preço de fecho seja superior à *moving average* de 100 e o RSI seja inferior a 30, ou o nosso *IsTrade* seja superior a 2.

* Venda

Vendemos sempre quando o preço de fecho é inferior à *moving average* de 100 e o RSI superior a 70, ou então quando o mercado tende para descer bastante, ou seja, o *IsTrade* abaixo dos -5.

* Manter

Mantemos sempre ação quando ambas as decisões de compra ou venda não se justificarem.

## Modelo estatístico

Agora que definimos os indicadores a utilizar e a estratégia onde os aplicamos, podemos então avançar com o nosso modelo estatístico, onde iremos todos os dias usar o preço de fecho, os indicadores mencionados e calculamos a decisão a realizar. No final de cada decisão calculamos o retorno financeiro da mesma para analise posterior. Repetindo este processo durante todos os dias que possuímos para estudo da ação, no nosso caso um total de 2222 dias.

Iniciamos o investimento com 10000 u.m (unidades monetárias), ao corrermos o nosso código, percorremos os dias para estudo efetuando uma das 3 decisões possíveis. No final podemos concluir quanto o nosso portfólio valorizou refletindo todas as decisões efetuadas neste período de 2222 dias.

Para comparamos as nossas decisões consoante o gráfico de desenvolvimento da ação, podemos observar a figura 2.

Uma imagem com texto, Gráfico, file, diagrama

Descrição gerada automaticamente

Figura 2 - Decisões de compra, venda, manter.

Observamos que a nossa estratégia se enquadrada bem com o desenvolvimento da ação, visto que as decisões de compra e venda assentam com o crescimento e decrescimento da ação.

Durante os nove 9, efetuamos 1536 compras, 171 vendas e 515 decisões de manter a ação, com investimento inicial de 10000 u.m, tivemos um retorno de 56276 u.m, ou seja, 560% de retorno.

Possivelmente a nossa estratégia poderá evoluir e alcançar um retorno maior, devido a ação por si cresceu 890% nestes mesmos 9 anos, mas visto que não possuímos um conhecimento alargada de investimentos decidimos que esta percentagem de retorno era satisfatória.

## Modelo reinforcement learning

Com a estratégia bem definida e uma baseline já estabelecida, podemos avançar para a integração do modelo de *reinforcement learning* com a estratégia previamente desenvolvida.

O primeiro passo é perceber a estrutura do modelo e os seus constituintes, uma wallet (com o dinheiro fora do mercado), um portfólio (com o nosso dinheiro no mercado) e por fim os nossos asset (com a soma do nosso dinheiro do portfólio e da wallet).

Ao comprar as ações, o nosso dinheiro é transferido para o portfólio onde podem ou não valorizar (consoante o mercado), ao vender as ações protegemos las da desvalorização do mercado e por fim ao manter as ações, seja tanto no portfólio ou na wallet, quando vender ou comprar não seja beneficiário.  
Nos casos em que estejamos fora do mercado e não possamos vender ações, nem possamos comprar ações enquanto estejamos dentro do mesmo.

Outro aspeto consiste em compreender os Hiper parâmetros que iremos utilizar no desenvolvimento do modelo e o impacto que terão no seu desempenho:

* Épsilon: utilizado para equilibrar a decisão entre explorar o ambiente (experimentar ações novas) e explorar a experiência adquirida (escolher a melhor ação com base no estado atual).
* Gamma (ou *discount factor*): aplicado para ajustar a relevância das recompensas futuras, evitando que o modelo tenda a favorecer cenários com recompensas acumuladas infinitas.
* StrategyProb: um parâmetro do nosso desenvolvimento, que utiliza os indicadores como base de decisão para a tomada de decisão (comprar, vender, manter).

De notar que implementamos um mecanismo de *decay* no épsilon e no StrategyProb, coagindo o agente a utilizar sempre a Q-table na tomada de decisões, não efetuando exploração desnecessária e utilizar oque aprendeu.

Um fator importante do reinforcement learning, são as *rewards*, num caso típico de *reinforcement learning* o cálculo das *rewards* seria sempre o mesmo, tendo por base os mesmos componentes, mas no nosso caso e com apenas dois estados possíveis no modelo (dentro e fora do mercado) a nossa ação de manter as ações irá produzir *rewards* com valor de 0, consequentemente a Q-table refletira nenhum benefício ao agente em manter as ações seja em que estado for.

Para resolver tal problema, decidimos então que teríamos de alterar a distribuição das *rewards* quando executamos a ação de manter as ações, para alem deste ajuste das *rewards* face à ação que tomamos também temos dois casos onde alteramos a forma como distribuímos a *reward*, quando o agente se encontra fora do mercado e pretende vender e quando se encontra dentro e pretende comprar, estas situações do ponto de vista do nosso modelo não são exequíveis visto que ao estar dentro ou fora do mercado o agente não consegue comprar ou vender mais ações . Logo recompensamos para tais estados e ações *rewards* extremamente negativas para que o agente nunca as efetuasse.

Com dois estados possíveis: dentro do mercado e fora do mercado, e três ações, resultando numa *Q-table* de dimensão 2x3. A atualização da *Q-table* é feita para cada estado com base nas *rewards* resultantes das ações tomadas.

A fórmula para atualizar a *Q-table* é a seguinte:



Onde:

Equação 1- formula de Q-table

* Q (s, a) é o valor atual para o estado s e ação a.
* α é a *learning rate*, que controla o quanto o novo valor influencia o antigo.
* r é a *reward* obtida após executar a ação no estado s.
* γ\gamma (discount factor) pondera a importância das recompensas futuras.
* Max Q (s′, a) é o maior valor da *Q-table* para o próximo estado s′, considerando todas as ações possíveis.

Este processo garante que a *Q-table* seja ajustada dinamicamente, permitindo que o modelo aprenda com as interações ao longo do tempo. Na figura 3 é possível visualizar com mais simplicidade o método de decisão que usamos no modelo.

Uma imagem com texto, diagrama, Esquema, Desenho técnico

Descrição gerada automaticamenteFigura 3 -Diagrama do processo de RL

A última componente necessária para a definição do nosso modelo de *reinforcement learning* será a fixação dos nossos hiperparâmetros. Esta atribuição de valores é crucial para uma aprendizagem mais eficiente e otimizada, resultando numa maior valorização do portfólio.

Para alcançar este objetivo, desenvolvemos um excerto código que executa o modelo 10 vezes, calculando a média do valor final do portfólio após cada iteração. Em cada 10 execuções, os híper parâmetros são reajustados e o processo reinicia, repetindo-se mais 10 vezes, sendo sempre calculada a média para análise posterior. No final da execução será exibida a seguinte mensagem:

Melhor Épsilon: 0.1

Com ε = 0.1, o agente explora aleatoriamente 10% do tempo e aproveita 90% das suas ações anteriores, equilibrando exploração e aproveitamento.

Melhor Probabilidade da Estratégia: 0.55

O agente tem uma probabilidade de 0.55 de4 seguir a estratégia estabelecida.

Melhor Alfa: 0.9

Alfa = 0.9 indica que o agente dá grande peso (90%) às recompensas ao ajustar os valores da Q-table, aprendendo rapidamente.

Melhor Gamma: 0.9

Gamma = 0.9 significa que o agente dá 90% de peso às recompensas futuras, equilibrando o curto e longo prazo.

Valor Médio Final dos Ativos: 64096.63

Com este *output* do código, conseguimos então definir os valores otimizados para os nossos Hiper parâmetros.

A tabela 1 representa a Q-table de uma iteração para os híper parâmetros acima devolvidos, servindo de exemplo, permite nos observar os valores que o agente tem ao seu dispor quando a consulta. Observamos quando este se encontra dentro do mercado a tendência será para manter as ações do que vender, sendo impossível comprar ações devido à impossibilidade deste ato como mencionamos previamente, o mesmo acontece quando estamos fora do mercado e pretendemos vender ações.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Estados | Ações | | | |
|  | Vender | Manter | Comprar |
| Sair do mercado | -8035 | 2575 | 2629 |
| Entrar no mercado | 2072 | 3518 | -7100 |

Tabela 1 – Q-Table

## Comparação de resultados

Concluindo a implementação e otimização do nosso modelo de *Reinforcement Learning e* com uma baseline definida pelo nosso modelo estatístico. Podemos então averiguar os resultados de ambos modelos e comparando os entre si, retirando conclusões se o nosso modelo de aprendizagem tem um melhor desempenho que o estatístico.

No modelo estatístico, usando a nossa estratégia conseguimos alcançar um retorno de 560%, elevando os nossos 10000 u.m para 56000 u.m, a ação da Microsoft só por si valorizou no mesmo intervalo de tempo em que investimos 890%, exibindo potencial para otimizarmos a nossa estratégia em 330%, mas como referimos anteriormente este trabalho visa para a aprendizagem de modelos de aprendizagem e não valorização financeira. Definimos então que a nossa baseline para o nosso modelo de reinforcement learning será os 560% de retorno.

Já no modelo de RL, como já indicado tivemos uma média de retorno de 640%, ou seja, uma superação do nosso modelo estatístico. Porem este valor é dúbio usando apenas 10 iterações para o alcançar, para tiramos conclusões mais concretas decidimos correr para os mesmos parâmetros mais otimizados, correr o modelo em 100 vezes.   
Com este número de iterações o nosso retorno médio já se encontra mais ajustado ao do nosso modelo estatístico, em torno de 604%, um ajuste de 107% face ao retorno do modelo estatístico.

## Conclusão

Este relatório tem como objetivo a construção e análise de dois modelos, um estatístico e outro baseado em técnicas de *Reinforcement Learning* (aprendizado por reforço), aplicados a uma estratégia de *trading* no mercado de ações, com foco na Microsoft. A abordagem inclui o desenvolvimento de estratégias para maximizar o retorno financeiro, explorando os princípios teóricos e práticos aprendidos ao longo do semestre.

No modelo estatístico, foram aplicadas técnicas tradicionais para identificar padrões de comportamento dos preços das ações, com ênfase em métricas como média móvel e RSI. Contudo, no modelo de Reinforcement Learning, foi utilizada uma abordagem centrada no Q-learning, complementada por iterações com diferentes configurações de hiperparâmetros, como taxa de aprendizagem (learning rate), fator de desconto (discount factor) e estratégias de aproveitamento-exploração (exploration vs. exploitation tradeoff).

Adicionalmente, o relatório avalia o impacto dessas escolhas na performance dos modelos, analisando indicadores como retorno acumulado, volatilidade e alavancagem do portfólio. Ao mesmo tempo que exploramos, um reajustamento nos híper parâmetros influenciam a convergência do aprendizado e a eficácia do modelo em ambientes de mercado voláteis.

Por fim, foi realizada uma comparação detalhada entre os dois modelos, destacando as vantagens e limitações de cada abordagem. O estudo reforça conceitos essenciais do semestre, proporcionando uma visão integrada sobre a relação entre teoria e prática, e abre espaço para futuras melhorias na aplicação de aprendizado por reforço no mercado financeiro.