Rildo luiz da silva uchôa Júnior

Matrícula: 01425188

1ª Atividade

1. Quais os tipos de aprendizagem de máquina e suas definições?

2. Para cada tipo de aprendizagem de máquina apresente um exemplo que se encaixa com a sua definição.

O aprendizado de máquina é um campo de estudo da inteligência artificial (I.A.) e da ciência da computação que, através do uso de dados e algoritmos, projetam (imitam) a maneira como os humanos aprendem, e redefinem suas ações, possibilitando previsões e a descoberta de insights importantes em projetos de [mineração de dados](https://blog.betrybe.com/tecnologia/mineracao-de-dados/), inclusive. Tudo isso origina e possibilita a tomada de decisões em aplicativos e negócios. O que permite a essa tecnologia, ter um potencial ilimitado que pode ser direcionada ao uso em diversas áreas.

Assim, esse tipo de aprendizado pode ser feito de três formas (ou tipos) e são eles:

O aprendizado supervisionado ocorre quando as pessoas responsáveis pelo desenvolvimento supervisionam, e eninam a máquina a trabalhar com os dados ofertados e resultados, possibilitando que o algoritmo encontre uma forma de trabalhar com isso, para prever resultados de eventos parecidos.

Um bom exemplo deste tipo de aprendizado é para o cálculo de um frete onde uma série de fatores que podem variar, como o preço dos combustíveis, entre outros, e através deste aprendizado, um software pode analisar cada um desses fatores, assim como os custos de fretes anteriores, e definir a partir disto, possíveis valores de fretes de acordo com a realidade informada.

No aprendizado semi supervisionado, somente alguns dados são supervisionados e outros não. Assim temos um conjunto de dados rotulados com um conjunto grande de dados não rotulados e esses dados não supervisionados ou não rotulados possam ser coletados de forma automática, sem necessitar de um filtro de classificação. O que, para coletar um dado rotulado, é necessário fazer uma análise mais complexa, que pode ter um custo alto com pessoas profissionais especializadas.

Para a utilização de um software que compreenda padrões de compra dos clientes e otimizem a venda através de estratégias eficientes, p aprendizado semi supervisionado é uma boa escolha pois pode analisar os dados de interação entre a empresa e clientes para identificar e projetar padrões de compra desses clientes e fazer ofertas mais direcionadas e organizar processos de venda de um modo mais eficiente.

Agora, no aprendizado por reforço, algoritmo pode interagir com o ambiente e testar várias atitudes, possibilidades e decisões para encontrar as que entreguem os melhores resultados através do método da tentativa e erro, onde o software em questão se torna capaz de definir os meios mais inteligentes para realizar uma tarefa.

Como exemplo deste tipo de aprendizagem, temos elaboração de uma carteira de investimento em meio a um mercado dinâmico, em que diversas variáveis podem afetar o preço de ações e com o uso do aprendizado de reforço podem ser feitos investimentos mais eficiente. Através de dados relacionados a carteiras de ativos criadas anteriormente e, com base nos resultados, positivos e negativos, elaborar uma carteira de acordo com as escolhas que entregaram resultados mais positivos. Outro exemplo disso é o algoritmo do youtube por exemplo onde a plataforma te oferece títulos que acredita que são do seu interesse e caso você não os assiste até o fim, o método é reconfigurado para acertar ou melhor ofertar na próxima vez.

2ª Atividade

Por que devemos fazer testes automatizados nas aplicações que desenvolvemos?

Porque é uma importante ferramenta bugs, falhas e outros problemas para sua aplicação. A automação permite que o teste seja repetido várias vezes, sendo mais fácil encontrar novos erros através da repetição e da simulação de cenários específicos, minimizando os problemas das abordagem manual,como o tempo despendido e, consequentemente, o custo final.

O que são testes unitários?

É um método ou teste, muito comum para identificar a consistência de um sistema, o dividindo em partes isoladas para análise do código, testando o fluxo geral e funcionalidades do sistema, otimizando o processo de descoberta de erros.

O que são testes automatizados?

São testes de avaliação de código-fonte tem como base a verificação de várias características de uma aplicação. Tudo isso sem a interferência de quem está criando o sistema. Onde a partir desta análise(teste), é possível identificar, brechas, bugs com menor custo e tem, assim como mão de obra pois, pode se criar um número adequado de máquinas virtuais para o fazer.

Escolha uma pirâmide de testes e descreva com suas palavras cada secção da pirâmide.

Podemos dividir esta pirâmide de testes como a representação de sua base, meio e topo, onde a base é o teste de unidade, o meio são os testes de integração e o topo são os testes de ponta (E2E).

Com os testes de unidade – podemos testar pequenas partes como se fossem funções/componentes independente da sua interação com outras partes do código, pois é a linha de frente na estratégia dos testes, pois tem menos complexidade, cobrem o maior número de linhas do código, possuem tempos de execução rápidos e são fáceis de identificar os erros. Assim, caso algum erro for encontrado e/ou houver alguma falha no teste é possível saber o local exato do erro e não precisar inspecionar o código todo.

Os Testes de Integração – garantem que quando as partes do código foram unidas, o sistema funcionará como um todo. Assim, testa um conjunto de unidades, um bom exemplo são funções/componentes que precisam interagir uma com as outras e funcionar em conjunto. É certo que as unidades podem funcionar normalmente, mas quando unidas uma com as outras pode acontecer do código não funcionar como esperado. A comunicação da aplicação como o banco de dados e APIS que fazem comunicação externa com a aplicação é um bom caminho para utilização do teste de inegração.

O Testes de Ponta a Ponta (E2E) é também conhecido como End to end. Testa o fluxo da operação do início ao fim para validar se a aplicação corresponde aos requisitos do projeto. Permite que o usuário pode usar de todas as funções que ele pode utilizar nessa aplicação final como se cadastrar, ver se a experiência do usuário(UX) condiz com o esperado entre outros. Tudo isso para que seja possível mapear os cenários que são relevantes e que possuem o maior impacto dentro do sistema.