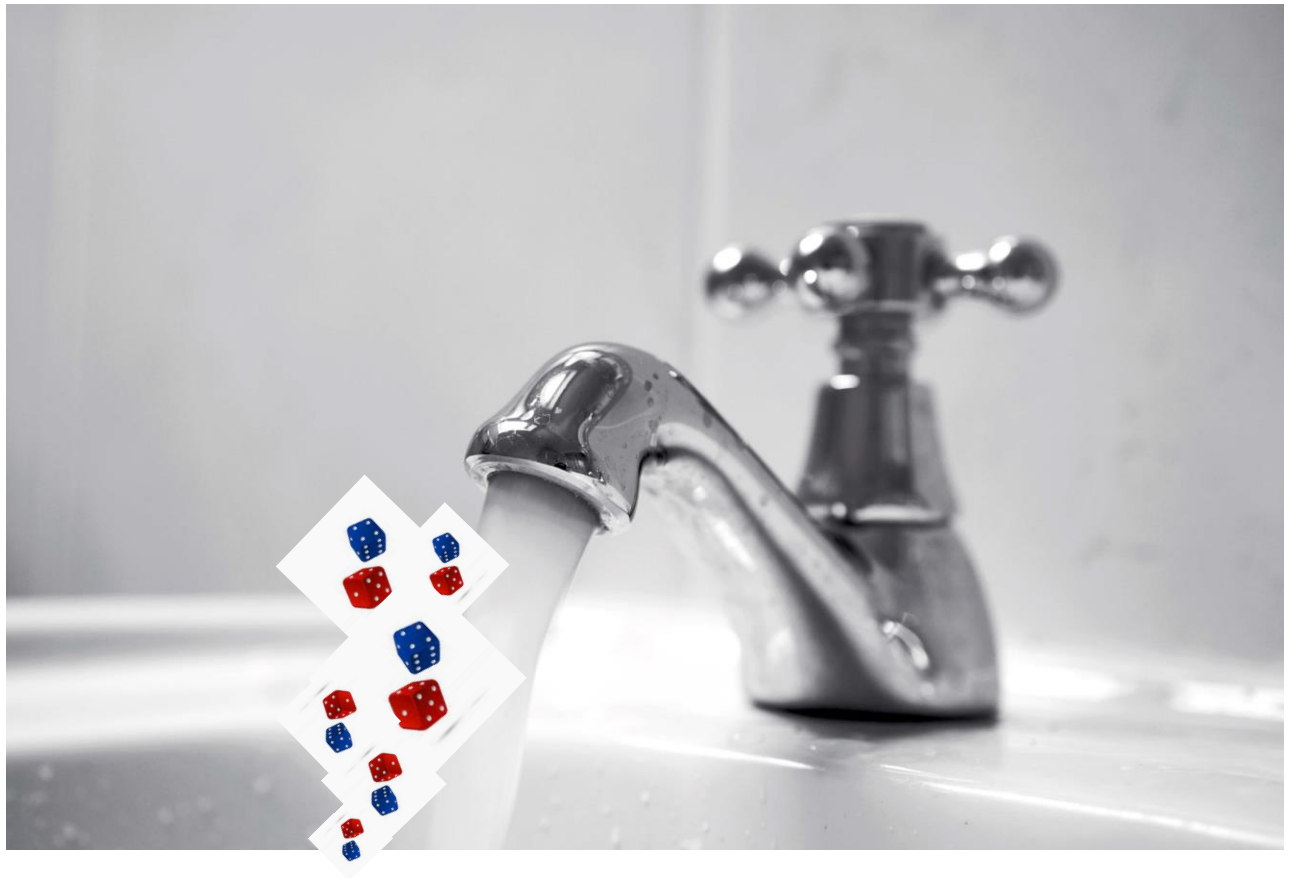
Aprendizado em Fluxo Contínuo de Dados



O que ocorreu ?

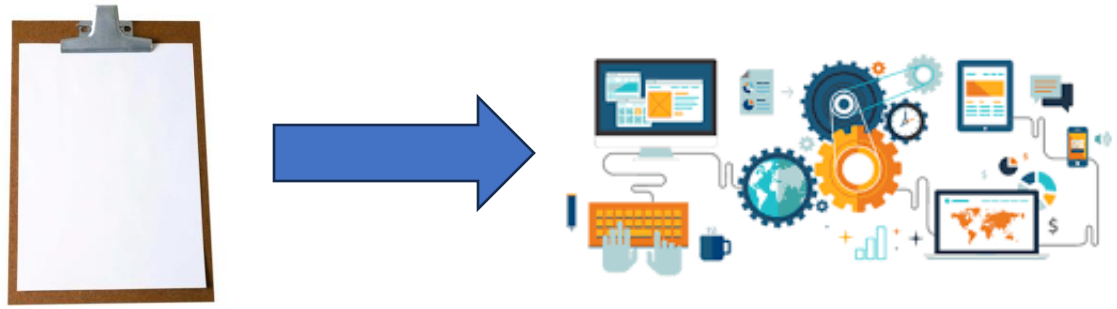
-Mudanças na forma de coletar dados (Manual/Automática)
-Alto fluxo contínuo de dados

Problema/Objetivo

-Retirar informação útil desse contínuo fluxo de dados

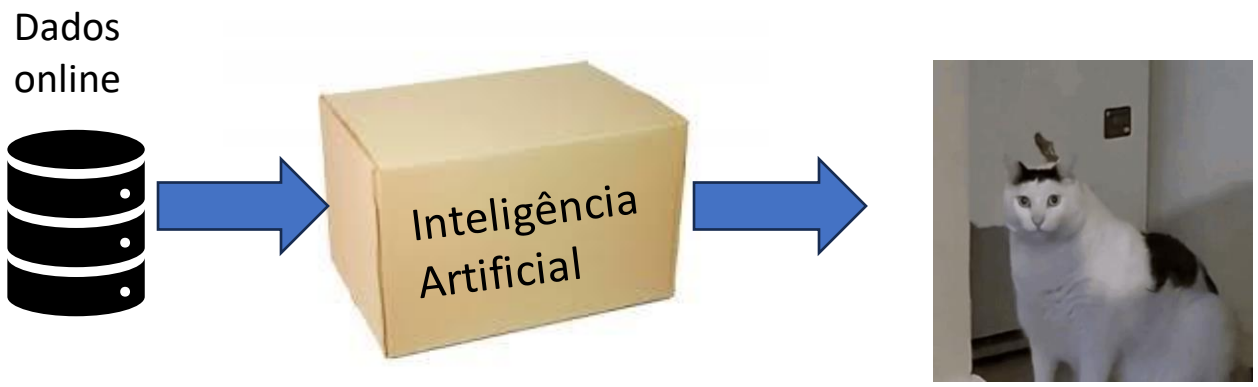
Possível solução

-Aplicar algoritmo de Aprendizado de Máquina



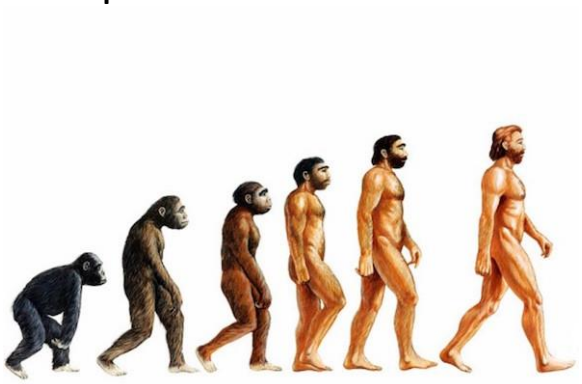
Desafios

-Não é possível aplicar técnicas de aprendizado de máquina tradicionais, pois os algoritmos de A.M. têm foco em lotes de dados offline.



Possível Solução

-Um algoritmo capaz de evoluir com o decorrer do tempo.



Desafios para implementar um algoritmo capaz de evoluir

1. Incrementabilidade
2. Aprendizado em tempo real
3. Capacidade de processar exemplos em tempo constante e com memória limitada
4. Acesso limitado a exemplos já processados
5. Capacidade de detectar e adaptar o modelo de decisão a mudanças de conceito *

Detecção de Mudança

- Pode ser feita de três formas:
  - De forma cega
  - Em algoritmos preditivos verificar a evolução do erro do classificador.
  - Em algoritmos descritivos pode ser feita através da medição da distância entre diferentes grupos.

Algoritmo 15.2 O Algoritmo SPC para detecção de mudança

Entrada: Um modelo de decisão atual $\hat{f}$
Uma sequencia de exemplos $\{(\mathbf{x}_j, y_j), j = 1, \ldots, n\}$

Seja $(\mathbf{x}_j, y_j)$ o exemplo atual
Computa a previsão do modelo: $\hat{y_j} \leftarrow \hat{f}(\mathbf{x}_j)$
Computa o erro: $erro_j$
Computa média dos erros: p_j e variância s_j
se $p_j + s_j < p_{min} + s_{min}$ **então**
    $p_{min} \leftarrow p_j$
    $s_{min} \leftarrow s_j$
fim
se $p_j + s_j < p_{min} + \beta \times s_{min}$ **então**
    $Aviso? \leftarrow Falso$
    Atualiza o modelo de decisão usando o exemplo atual: $\mathbf{x}_j, y_j$
fim
senão
    se $p_j + s_j < p_{min} + \alpha \times s_{min}$ **então**
        se $!\ Aviso?$ **então**
            $buffer \leftarrow \{(\mathbf{x}_j, y_j)\}$
            $Aviso? \leftarrow Verdadeiro$
        fim
        senão
            $buffer \leftarrow buffer \cup \{(\mathbf{x}_j, y_j)\}$
        fim
    fim
    senão
        Reaprende um novo modelo de decisão com os exemplos no *buffer*
        $Aviso? \leftarrow Falso$
        Reinicializa p_{min} e s_{min}
    fim
fim

Exemplo: **Algoritmo SPC (Preditivo)**

S- é a variância associada à estimativa p
P- é a probabilidade de erro
N- número de exemplos processados

$$s = \sqrt{p \times (1-p)/n},$$

Em geral
-Quanto maior o S e P pior.

1- Passo

Encontrar o S mínimo
Encontrar o P mínimo
Caso
Si + Pi < Smin + Pmin
Smin = Si
Pmin = Pi
Senão faça nada

2- Passo(Alerta)
Caso
Si + Pi >= Pmin + 2*Smin
Senão faça nada

3-Passo(Mudança)
Caso
Si + Pi >= Pmin + 3*Smin
Atualiza o algoritmo utilizando esses novos dados
Senão faça nada

Exemplos de Algoritmos Online

-Very Fast Decision Tree

-algoritmo de árvore de decisão que dinamicamente ajusta seu viés.

-os nós folhas são substituídos por nós de decisão que carregam estatísticas sobre os valores dos atributos.

-Limite de Hoeffding

Algoritmo 15.1 O algoritmo da árvore de Hoeffding

Entrada: Uma sequência de exemplos de treinamento $\mathbf{D} = \{(\mathbf{x}_i, y_i), i = 1, \ldots, \infty\}$
Uma função de avaliação de divisão $H(.)$
Número mínimo de exemplos N_{min}
δ: 1 menos a probabilidade desejada de escolher o atributo correto em qualquer nó.
τ: Constante usada para desempate.
Saída: HT: Árvore de Decisão

```
1  Seja HT ← Folha Vazia (Raiz) ;
2  para cada exemplo (x_i, y_i) ∈ D faça
3      Atravessar a árvore HT a partir da raiz até a folha l;
4      Atualizar as estatísticas suficientes em l;
5      se O número de exemplos em l é maior que N_min então
6          Calcular H(at_i) para todos os atributos ;
7          Seja at_a o atributo com maior H ;
8          Seja at_b o atributo com o segundo maior H ;
9          Calcular ε (limite de Hoeffding) ;
10         se (H(at_a) − H(at_b) > ε) então
11             Substituir l por um teste de divisão baseado no atributo at_a ;
12             Adicionar uma nova folha vazia para cada possível valor de at_a ;
13         fim
14         senão
15             se ε < τ então
16                 Substituir l por um teste de divisão baseado no atributo at_a ;
17                 Adicionar uma nova folha para cada possível valor de at_a ;
18             fim
19         fim
20     fim
21 fim
```

Exemplos de Algoritmos Online

-Analise de séries de Agrupamentos Temporais

-é um algoritmo de agrupamento e clustering de dados.

-é um dos primeiros exemplos de agrupamento em fluxo contínuo.

-possui duas operações: Divisão e Agregação

-Os nós folhas recebem informações atualizadas

-O diâmetro de um nó é determinado pela maior distância entre 2 variáveis incluídas no nó.

-Espera-se que o diâmetro diminua à medida que se percorre a árvore, mas, caso ele suba consideravelmente de um nó pai ao filho, deve-se agregar os filhos e reiniciar suas estatísticas suficientes.

-Os requisitos de armazenamento escalonam de forma quadrática conforme o número de variáveis, mas são constantes quanto à quantidade de exemplos.