Mineração de Dados e Aprendizado de Máquina

Prof. Edson Pedro Ferlin

Mineração de Dados

Introdução à Mineração de Dados

Objetivo

Encontrar padrões ocultos em grandes volumes

Técnicas

Clustering, classificação, regressão, associação

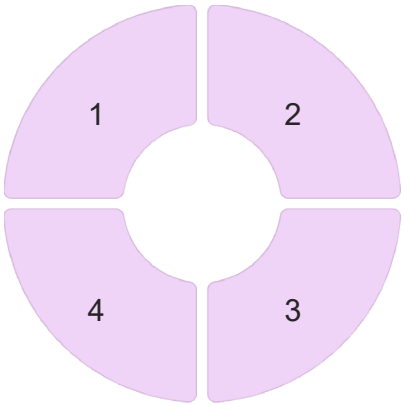
Aplicação

Análise de padrões de compra em e-commerce

Conceituação da Mineração de Dados

Classificação

Atribuição a categorias predefinidas



Agrupamento

Agrupar dados semelhantes sem rótulos

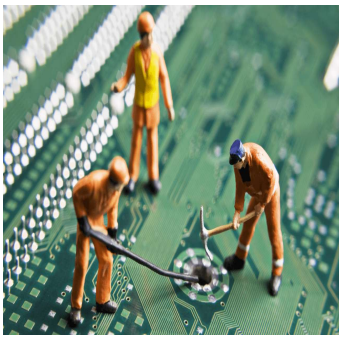
Detecção de Anomalias

Identificação de padrões incomuns

Associação

Descoberta de relações entre variáveis

Mineração de Dados



A mineração de dados utiliza como base para seus trabalhos experimentos de áreas como estatística, inteligência artificial, máquina de estado e banco de dados para construir seu modelo.

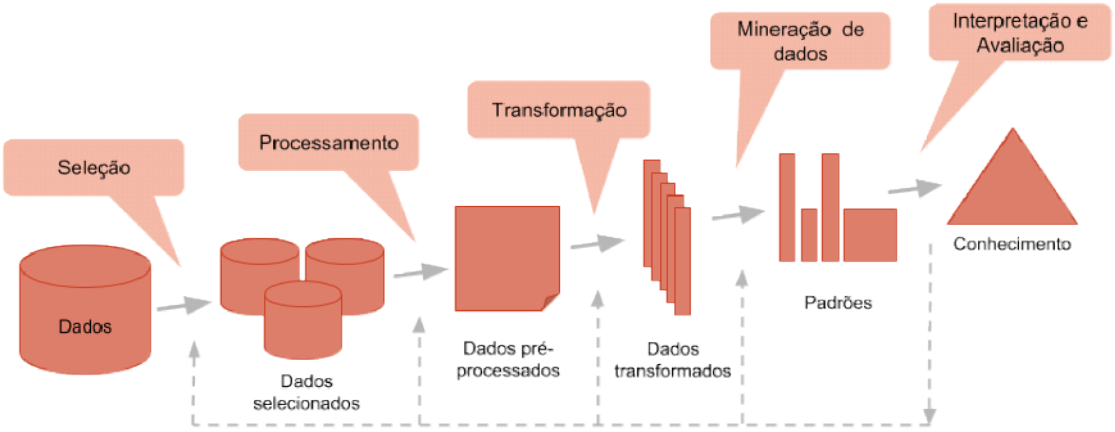
Com Data Mining (em português, mineração de dados), é possível descobrir informações de grande valor, principalmente para ajudar nas tomadas de decisões.

Mineração de Dados

A mineração de dados está relacionada, também, às áreas da inteligência artificial que são chamadas de descoberta de conhecimento e aprendizagem de máquina.

O termo “mineração de dados” está relacionado aos estágios de descoberta do processo de KDD (Knowledge Discovery in Databases), que é um processo, de várias etapas, não trivial, interativo e iterativo, para identificação de padrões compreensíveis, válidos, novos e potencialmente úteis a partir de grandes conjuntos de dados.

KDD (*Knowledge Discovery in Databases*)



Mineração por Grupo de Associação

A técnica de mineração por associação tem por objetivo identificar o relacionamento de itens que, em um específico conjunto de dados, sejam mais frequentes. Normalmente, o volume de dados que envolvem esse tipo de mineração é extenso e, diante dessa premissa, torna-se necessária a utilização de algoritmos que sejam mais rápidos e eficientes.

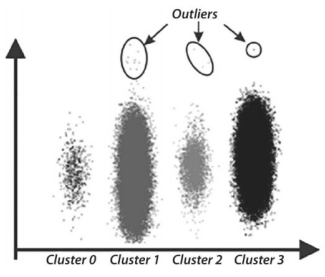
Regra 1: SE idade > 25 AND graduação completa = sim ENTÃO fazer mestrado = sim
Regra 2: SE idade <= 25 AND graduação completa = não ENTÃO fazer mestrado = não

Mineração de itens frequentes

Esta técnica, geralmente, é visualizada em duas etapas: na primeira delas, um conjunto de itens frequentes é desenvolvido e há um valor mínimo de frequência a ser respeitado.
Após essa etapa, regras de associação devem ser geradas pela mineração desse conjunto de itens.
A fim de que os resultados sejam válidos, para cada regra produzida, deverão ser utilizados conceitos de confiança e suporte.
O algoritmo mais utilizado para a estratégia da mineração de itens frequentes é o Apriori, no qual são envolvidas técnicas de hash, particionamento, redução de transações e segmentação.

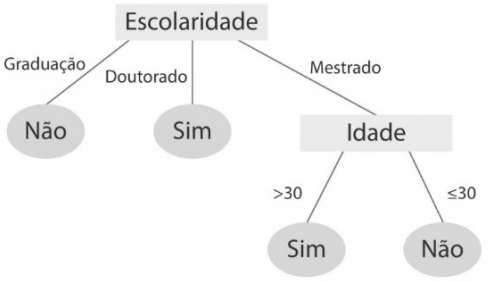
Mineração por Clustering

A técnica de clustering tem como objetivo identificar e aproximar dados semelhantes. Trata-se de uma coleção de registros semelhantes entre si, mas diferentes de registros em demais agrupamentos. Essa técnica não pretende classificar, estimar ou predizer o valor de qualquer variável, apenas pretende identificar grupos de dados similares.



Mineração por Árvores de Decisão

A técnica de mineração por árvores de decisão faz muito sucesso devido ao fato de não necessitar de parâmetros de configuração (o que a torna bastante simples) e por ter um alto grau de assertividade. Geralmente, é utilizada em categorizações ou previsões de dados. Árvores de decisão são formadas a partir de um conjunto de regras de classificação, em que cada caminho da raiz até uma folha representa uma dessas regras.



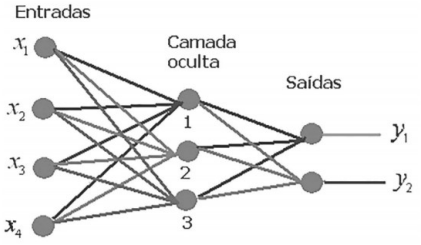
Mineração por Classificação Bayesiana

A técnica de mineração por classificação bayesiana é tida como uma técnica estatística e baseia-se no teorema de Thomas Bayes, segundo o qual é possível encontrar a probabilidade de um determinado evento ocorrer diante da probabilidade de outro evento já ter ocorrido:

Probabilidade (Y dado X) = Probabilidade (X e Y) / Probabilidade (X).

Mineração por Redes Neurais

Pode ser vista como um conjunto de entradas e saídas (assim como ocorre nos neurônios) que são conectadas por camadas intermediárias e na qual cada ligação tem um valor associado. É uma técnica que precisa de um grande período de treinamento, ajustes de parâmetros. É difícil de interpretar e também não é possível identificar de forma clara e precisa a relação entre a entrada e a saída.

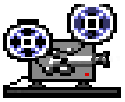


Aplicações de Mineração de Dados em Big Data

- **Segmentação de Mercado**
  Agrupamento de consumidores para marketing
- **Análise de Redes Sociais**
  Identificação de padrões e sentimentos
- **Análise de Cesta**
  Itens frequentemente comprados juntos
- **Detecção de Fraudes**
  Identificação de padrões incomuns

Deep Web e Dark Web

Assista o vídeo sobre Deep Web e Dark Web
(link: https://youtu.be/dFCzm6xHPE4).



Aprendizado de Máquina

Introdução ao Aprendizado de Máquina em Big Data

1 **Objetivo**

Treinar algoritmos para previsões com grandes volumes

2 **Aplicações**

Diagnóstico médico, previsão de demanda, análise de sentimentos

3 **Ferramentas**

Apache Spark MLlib, TensorFlow, Scikit-learn

Conceituação de Aprendizado de Máquina

Definição

Sistemas aprendem com experiência sem programação explícita

Supervisionado

Aprende de dados rotulados

Não Supervisionado

Identifica padrões sem rótulos

Reforço

Aprende por tentativa e erro

Terminologia

O termo aprendizado engloba alguma experiência ou prática sobre algum assunto. Neste caso, especificamente, o aprendizado de máquina remete à inserção desse conceito em máquinas computacionais. Porém, sempre há um propósito e, aqui, veremos a relação do aprendizado de máquina com os dados.

Definição

Aprendizado de máquina computacional (AM) é a aplicação de técnicas computacionais com o objetivo de encontrar padrões ocultos em dados.

Os padrões ocultos são aquelas características que não podem ser observadas tão claramente nos dados.

Além de estar relacionado à inteligência artificial, o aprendizado de máquina está interligado, também, com a estatística e, consequentemente, com a mineração de dados.

Elementos

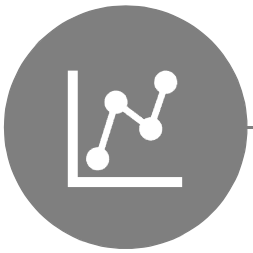
Treinamento	O treinamento faz parte do aprendizado de máquina, já que é devido ao uso de algoritmos e à inserção de dados que a máquina adquire os conhecimentos necessários para desempenhar as funções para as quais foi designada.
Indução	O processo de indução traz a procura de uma melhor hipótese, ou seja, de uma melhor resposta ou solução para determinada situação.
Regras	Limitam as possibilidades do algoritmo de aprendizado de máquina.
Hipóteses	São possíveis conclusões, ou seja, possíveis respostas predeterminadas e que são provadas, ou não, ao final.

Categorias de Aprendizado

Tipos de Aprendizado

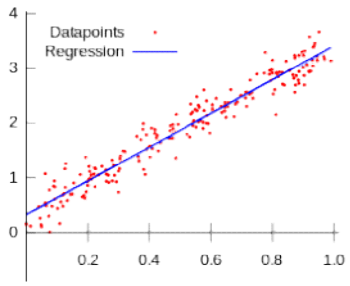
- Supervisionado: traz um objetivo estabelecido e pode ser dividido entre problemas de regressão e de classificação.
- Não supervisionado: quando o objetivo não está bem definido e temos o intuito de compreender melhor os dados para realizar o agrupamento.
- Por reforço: quando as saídas não estão bem definidas e as respostas só podem ser aferidas após algumas execuções.

Supervisionado



- Regressão: mapeiam um exemplo em um valor real. Um exemplo de regressão é prever o tempo de internação de um paciente em um hospital.
- Classificação: associa a descrição de um objeto a uma classe. Um exemplo de classificação é determinar a doença de um paciente pelos seus sintomas.

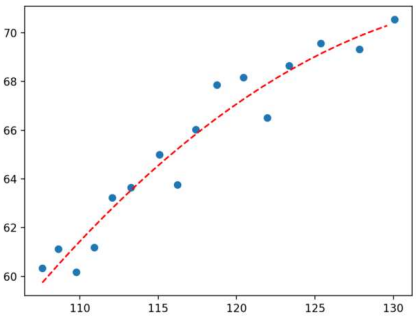
Regressão Linear



- Implementando o modelo de regressão linear

```
[ ] from sklearn.linear_model import LinearRegression
    from sklearn.metrics import mean_squared_error, r2_score

    reg_linear = LinearRegression().fit(X_train, y_train)

    reg_linear.score(X_train,y_train)

    0.3022780263167186 3

[ ] y_pred_reg_lin = reg_linear.predict(X_test)

    print('MSE = ',mean_squared_error(y_test, y_pred_reg_lin), 'R2 = ',r2_score(y_test,y_pred_reg_lin))

    MSE =  6320.446777983208 R2 =  0.33150317132991935
```


Regressão Não-Linear



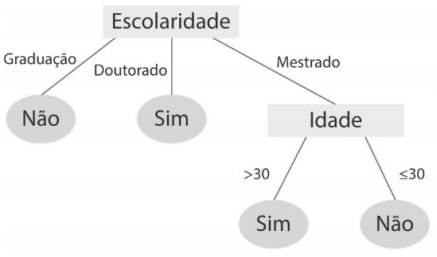
- Implementando o modelo de regressão não linear

```
[ ] from sklearn.preprocessing import PolynomialFeatures
    from sklearn.pipeline import make_pipeline
    from sklearn.metrics import mean_squared_error, r2_score

    degree=2

    reg_poli=make_pipeline(PolynomialFeatures(degree),LinearRegression())

    reg_poli.fit(X_train,y_train)

    reg_poli.score(X_train,y_train)

    0.5343740690841223

[ ] y_pred_reg_poli = reg_poli.predict(X_test)

    print('MSE = ',mean_squared_error(y_test, y_pred_reg_poli), 'R2 = ',r2_score(y_test,y_pred_reg_poli))

    MSE =  4575.520425461595 R2 =  0.5160593860878517
```

Não Supervisionado

- Agrupamento: os dados são agrupados de acordo com sua similaridade.
- Sumarização: busca encontrar uma descrição simples e compacta para um conjunto de dados.
- Associação: consiste em encontrar padrões frequentes de associações entre os atributos de um conjunto de dados

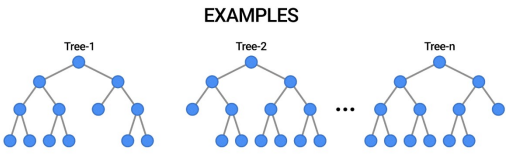
Árvore de Decisão



- Implementando o modelo de arvore de decisão

```
[ ] from sklearn.tree import DecisionTreeRegressor
    from sklearn import tree
    from sklearn.metrics import mean_squared_error, r2_score

    av_dec_reg = DecisionTreeRegressor(random_state=1212, max_depth=7)

    av_dec_reg.fit(X_train, y_train)

    av_dec_reg.score(X_train,y_train)

    0.925334831007542

[ ] y_pred_av_dec = av_dec_reg.predict(X_test)

    print('MSE = ',mean_squared_error(y_test, y_pred_av_dec), 'R2 = ',r2_score(y_test,y_pred_av_dec))

    MSE =  7290.426842506872 R2 =  0.2289109623003266
```

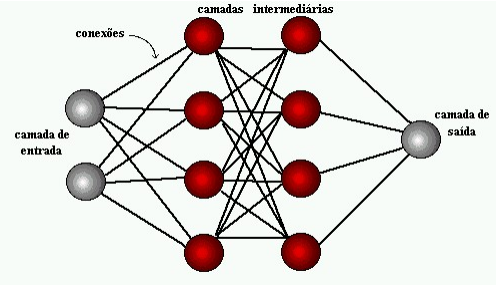
Mineração por Redes Neurais



- Implementando o modelo de random forest regression

```
[ ] from sklearn.ensemble import RandomForestRegressor
    ra_fr_re = RandomForestRegressor(random_state=1212, max_depth=11)
    ra_fr_re.fit(X_train, y_train)
    ra_fr_re.score(X_train, y_train)
    0.9329132561517521

[ ] y_pred_ra_fr_re = ra_fr_re.predict(X_test)
    print('MSE = ',mean_squared_error(y_test, y_pred_ra_fr_re), 'R2 = ',r2_score(y_test, y_pred_ra_fr_re))
    MSE =  3810.2485562019347 R2 =  0.5970001543026182
```

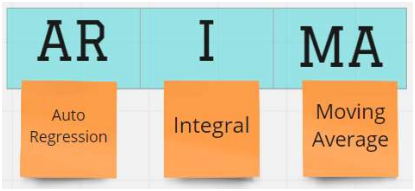

Redes Neurais

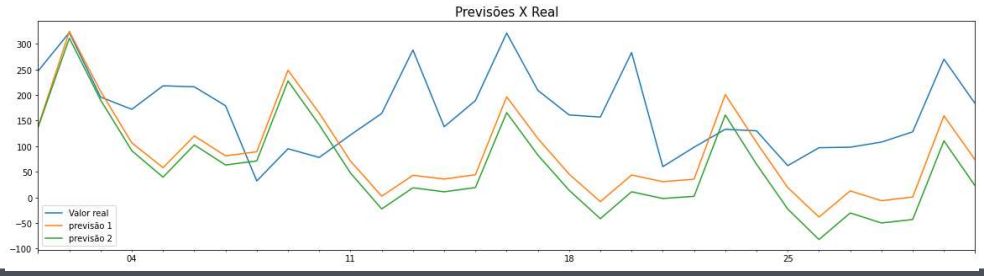


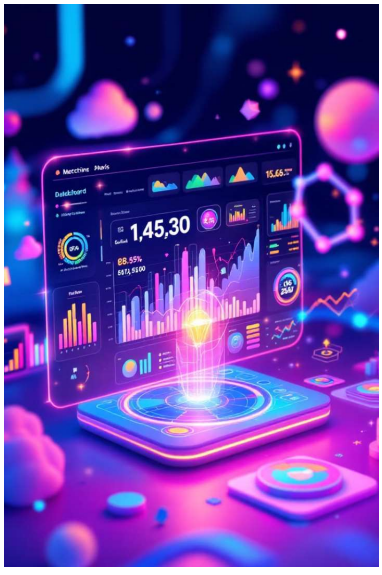
- Implementando o modelo de redes neurais

```
from sklearn.neural_network import MLPRegressor
MLP_reg = MLPRegressor(max_iter=1000, hidden_layer_sizes=(10000,))
MLP_reg.fit(X_train, y_train)
MLP_reg.score(X_train,y_train)
0.470945355850552

[ ] y_pred_MLP = MLP_reg.predict(X_test)
    print('MSE = ',mean_squared_error(y_test, y_pred_MLP), 'R2 = ',r2_score(y_test,y_pred_MLP))
    MSE =  5927.10145739977 R2 =  0.3731062586778283
```

Séries Temporais







Aplicações de Aprendizado de Máquina em Big Data

1. **Previsão de Tendências**
   Análise de dados históricos para prever comportamento
2. **Análise de Risco**
   Previsão de falhas, fraudes ou risco de crédito
3. **Personalização**
   Sistemas de recomendação baseados no comportamento
4. **Processamento de Imagens**
   Classificação e reconhecimento em dados multimídia

Aplicação: Machine Learning em E-commerce

1
Previsão de Vendas
Otimização de estoques e recursos.

2
Análise de Comportamento
Padrões de compra e navegação.

3
Processamento de Dados
Histórico de vendas e sazonalidade.

Exemplo de Aplicação

1
Coleta de Dados
Histórico de vendas e comportamentos

2
Processamento
Apache Spark para análise

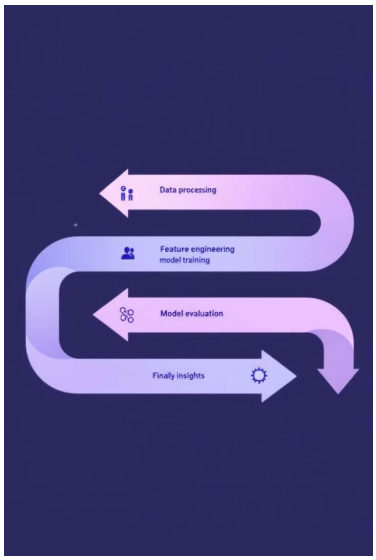
3
Modelagem
Algoritmos de regressão

4
Previsão
Demanda futura e otimização

Comparativo

Aprendizado de Máquina e Mineração de Dados

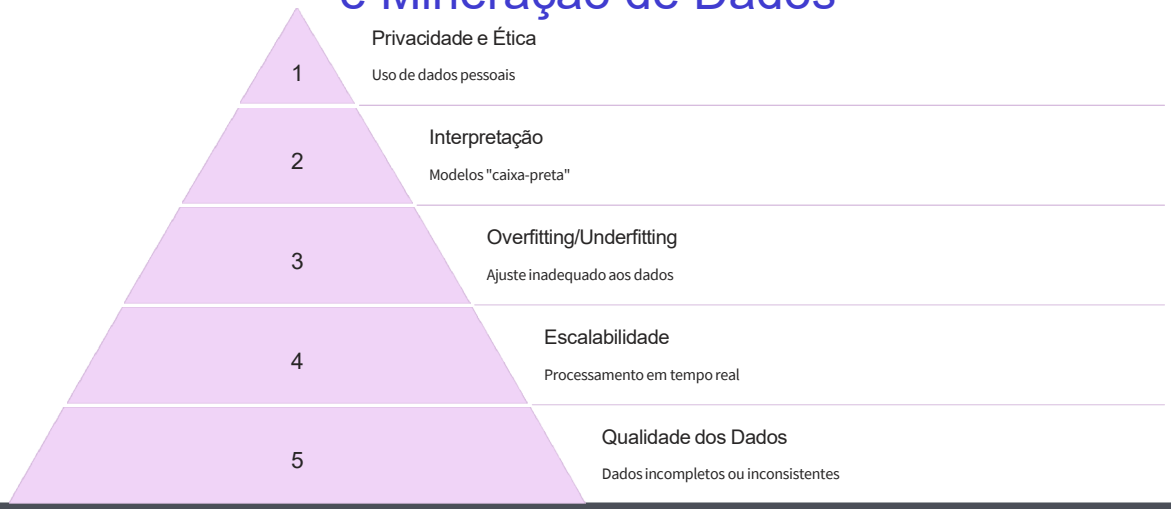
1. **Coleta e Preparação**
   Obtenção e limpeza dos dados brutos.
2. **Seleção de Algoritmos**
   Escolha de técnicas adequadas ao problema.
3. **Treinamento do Modelo**
   Ajuste e avaliação de performance.
4. **Interpretação**
   Extração de insights e aplicação prática.



Diferença entre Aprendizado de Máquina e Mineração de Dados

Critério	Aprendizado de Máquina	Mineração de Dados
Objetivo	Criar modelos preditivos	Identificar padrões e insights
Dados	Frequentemente rotulados	Rotulados ou não
Resultado	Predições e recomendações	Padrões e relacionamentos

Desafios no Uso de Aprendizado de Máquina e Mineração de Dados



Contato

eferlin@live.com

(BLOG) professorferlin.blogspot.com

(SITE) professorferlin.com.br

(YOUTUBE) ProfEdsonPedroFerlin