

Tópicos Especiais em Sistemas de Informação

**Machine Learning:
Conceitos iniciais e introdução à classificação**

Ely – elydasilvamiranda@gmail.com

Fonte

- Slides baseados em:
 - Curso da Alura:
<https://cursos.alura.com.br/course/introducao-a-machine-learning-com-classificacao/>
 - Capítulo 1 dos livros:
 - Hands-On Data Science and Python Machine Learning
 - Python Machine Learning By Example
 - Python Machine Learning - Second Edition

Biblioteca necessária

- Scikit Learn;
 - > pip install scikit-learn
- Numpy:
 - > pip install numpy
- Scipy
 - > pip install scipy
- Documentação:
<http://scikit-learn.org/stable/documentation.html>

Machine Learning

- É um termo cunhado em torno de 1960 composto de duas palavras:
 - **Machine:** correspondente a um computador, robô ou outro dispositivo;
 - **Learning:** um processo de aprendizado a partir:
 - dados de uma atividade, ou de padrões de eventos, nos quais os seres humanos são bons.
- Usa algoritmos que podem aprender a partir da observação e análise de dados.
- A partir desse processo, os algoritmos podem fazer previsões baseadas nesses dados.

Por que ML?

- Há muitos problemas envolvendo conjuntos de dados enormes, ou cálculos complexos;
- Transformar (abundância de) dados (áudio, vídeo, imagem, texto..) em informação...
 - ... Preferencialmente, minimizando a intervenção humana em tarefas maçantes, tediosas e braçais;
 - Computadores não se cansam e são mais baratos;
- Os humanos ficariam (até quando?) com tarefas criativas.

Algumas aplicações

- Classificação e filtragem de spams em e-mails:
 - Padrões são alimentados e predições são feitas baseadas nisso; em caixas de e-mail;
- Publicidade on-line:
 - De forma automática são gerados anúncios com base em informações em dados coletados;
- Search engines:
 - Há análise e classificação contextual, classificação por relevância;
- Sistemas de recomendação:
 - Clientes são auxiliados a escolherem produtos e serviços.

Algumas aplicações

- A aplicação do aprendizado de máquina é ilimitada;
- Novos exemplos surgem todos os dias:
 - Detecção de fraudes de cartão de crédito;
 - Diagnóstico de doenças;
 - Previsão de eleição presidencial;
 - Tradução instantânea;
 - Marketing direto;
 - Risco de crédito;
 - Filtros de spam;
 - Previsão de doenças;
 - Separação de notícias
 - Reconhecimento de voz e de face;
 - Sistemas de recomendação/satisfação de clientes
 - Previsão de vendas;
 - Bots;
 - Etc...

Tipos de aprendizado

Supervised Learning

- > Labeled data
- > Direct feedback
- > Predict outcome/future

Unsupervised Learning

- > No labels
- > No feedback
- > Find hidden structure in data

Reinforcement Learning

- > Decision process
- > Reward system
- > Learn series of actions

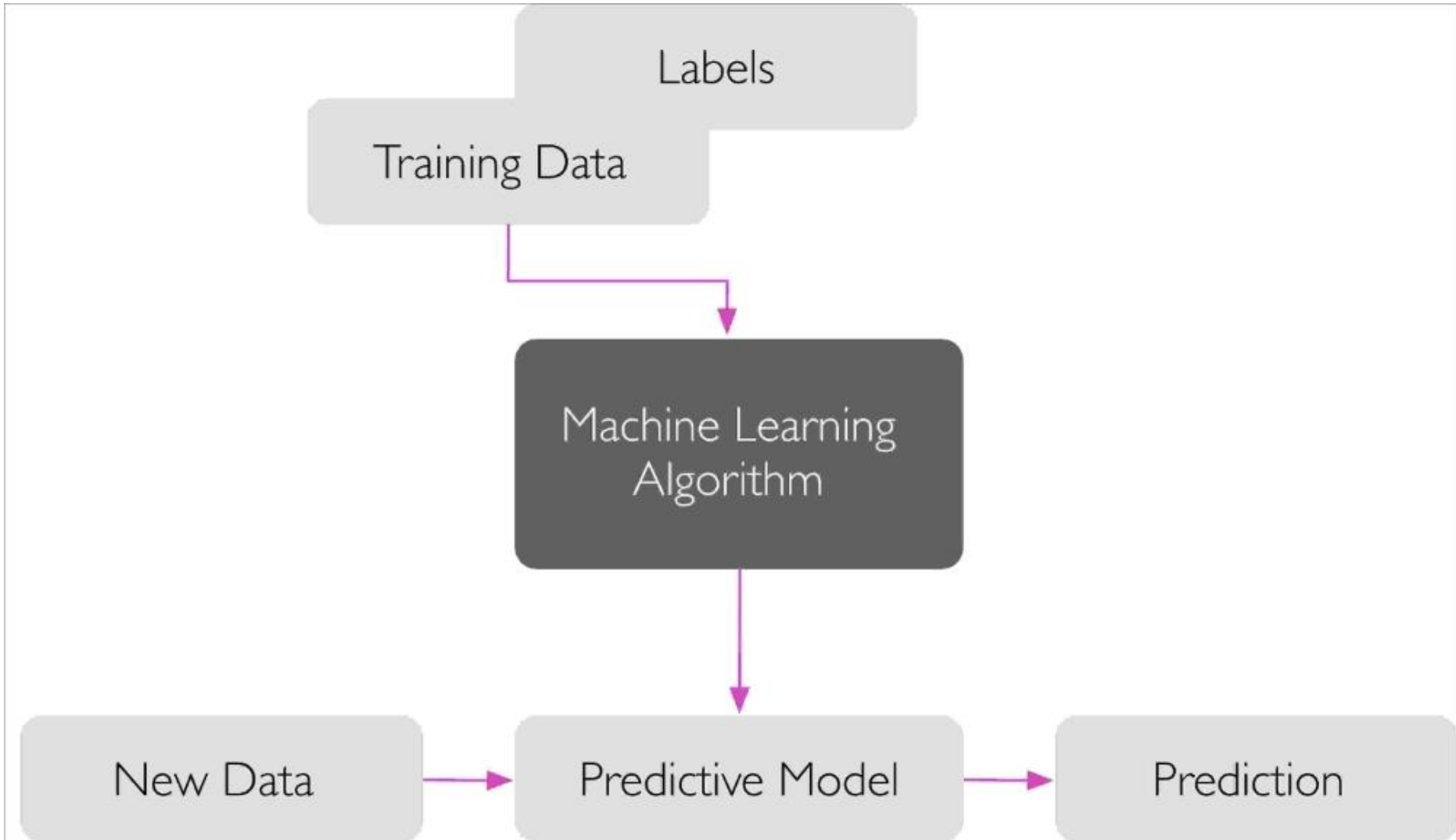
Aprendizagem supervisionada

- Esse processo de aprendizagem de dados possui:
 - Descrição;
 - Alvos ou saídas desejadas
 - Além de sinais indicativos;
- O objetivo é uma regra geral que mapeie entradas para saídas.

Aprendizagem supervisionada

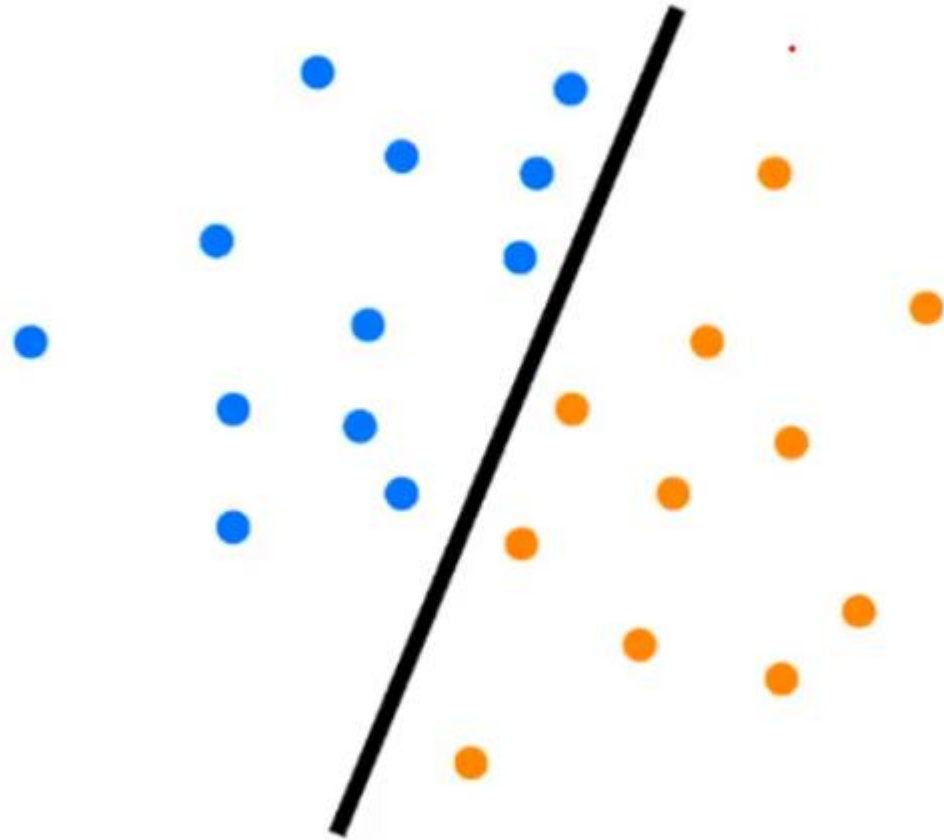
- O tipo de dados dessa aprendizagem é chamado de dados rotulados;
- Regra são definidas para para rotular novos dados com saídas desconhecidas;
- Subclassificações: classificação e regressão;

Aprendizagem supervisionada



Classificação

- Dado um problema como é spam ou não, podemos classificar a resposta como binária:
 - 1 é spam;
 - -1 não spam;
- Outro problema simples que é diferenciar um cachorro de um porco:
 - 1 é porco;
 - -1 é cachorro;
- Mas para classificar precisamos responder a algumas perguntas.



Porco ou cachorro?

- Algumas perguntas podem ser feitas para diferenciar cachorros de porcos:
 - Será que ele tem perna curta que nem o porco?. **Não;**
 - Será que ele é gordinho que nem o porco? **Alguns sim e outros não;**
 - Ele late que como o cachorro? **Sim. Eles faziam *au au*;**
 - Então ele não é um porquinho, ele é um cachorrinho.
- Nota: as suas perguntas podem ser diferentes das minhas.

É Spam ou não spam?

- Algumas características poderiam ser usadas:
 - Tamanho do e-mail;
 - As palavras que aparecem;
 - O horário que foi enviado
 - Se é um contato ou não;
 - Se é a primeira vez que esse e-mail vem para um destinatário;

Porco ou cachorro?

- Algumas características não seriam interessantes, como exemplo a cor;
 - "Ele é rosa?"
 - "Ele é preto?"
 - "Ele é branco?"
 - "Ele é cinza?"
- Essa característica pouco ajudaria para diferenciar os animais.

Implementando uma solução

- Definiremos cada animal como uma variável:
 - 3 porcos e 3 cachorros;
- Cada variável será uma resposta às 3 perguntas:
 - Ele é gordinho?
 - Ele tem pernas curtas?
 - Ele late?
- Essas variáveis serão nossos **dados**.

Definição das variáveis:

```
# classificacao.py
```

```
#[é gordinho?, tem pernas curtas?, ele late?]
```

```
porco1 = [1, 1, 0]
```

```
porco2 = [1, 1, 0]
```

```
porco3 = [1, 1, 0]
```

```
cachorro4 = [1, 1, 1]
```

```
cachorro5 = [0, 1, 1]
```

```
cachorro6 = [0, 1, 1]
```

```
dados = [porco1, porco2, porco3,  
cachorro1, cachorro2, cachorro3]
```

Definindo as respostas

- As respostas aqui serão chamadas de **marcações**;
- As marcações devem ter valores distintos para elementos diferentes:
 - 1 : porco;
 - -1 : cachorro.

```
# classificacao.py
```

```
#Código anterior omitido
```

```
marcacoes = [1, 1, 1, -1, -1, -1]
```

E agora?

- Queremos treinar um algoritmo para solucionar o caso abaixo:

```
# classificacao.py
```

```
# Código anterior omitido
```

```
misterioso = [1, 1, 1]
```

- Porco ou cachorro?

Usando um algoritmo de ML

- "Treinar um modelo": processo de aplicar um algoritmo de ML a um conjunto de dados;
- Usaremos o algoritmo MultinomialNB:
 - É uma implementação de um algoritmo bayesiano, chamado Naive Bayes;
 - Disponível no pacote `sklearn.naive_bayes`.

```
# classificacao.py
```

```
# ...
```

```
from sklearn.naive_bayes import  
MultinomialNB  
modelo = MultinomialNB()
```

Treinando e prevendo

- Há dois métodos disponíveis no modelo:
 - `fit(dados, marcacoes)`: treina um modelo;
 - `predict([dados])`: prevê os resultados para novos elementos;

```
modelo = MultinomialNB()  
modelo.fit(dados, marcacoes)  
print(modelo.predict([misterioso1]))
```

Implementação

- *# [é gordinho?, tem pernas curtas?, ele late?]*

```
porco1 = [1, 1, 0]
```

```
porco2 = [1, 1, 0]
```

```
porco3 = [1, 1, 0]
```

```
cachorro1 = [1, 1, 1]
```

```
cachorro2 = [0, 1, 1]
```

```
cachorro3 = [0, 1, 1]
```

```
dados_treino = [porco1, porco2, porco3, cachorro1,  
cachorro2, cachorro3]
```

```
marcacoes = [1, 1, 1, -1, -1, -1]
```

```
misterioso1 = [1, 1, 1]
```

```
from sklearn.naive_bayes import MultinomialNB
```

```
modelo = MultinomialNB()
```

```
modelo.fit(dados_treino, marcacoes)
```

```
print(modelo.predict([misterioso1]))
```

Resultado

- O resultado do Algoritmo deu -1 (porco)
- Está correto, entretanto, podemos dizer que é um resultado que o Algoritmo não poderia errar;
- Trata-se de um resultado que ele já conhece.

Exercício

- Considere os seguintes animais:
 - misterioso2 = [0, 0, 0]
 - misterioso3 = [0, 1, 0]
 - misterioso4 = [1, 0, 0]
 - misterioso5 = [0, 0, 1]
 - misterioso6 = [1, 0, 1]
- Eles são resultados já conhecidos pelo Algoritmo?
- Adicione-os à lista, separando por vírgulas e aplique o método predict novamente.
- Qual o resultado?

Exercício

- O resultado foi:
 - $[-1 \ 1 \ 1 \ -1 \ -1]$
- Podemos concluir que o algoritmo conseguiu acertar 80% dos resultados.

Exercício

- O resultado foi:
 - $[-1 \ 1 \ 1 \ -1 \ -1]$
- Podemos concluir que o algoritmo conseguiu acertar 80% dos resultados.

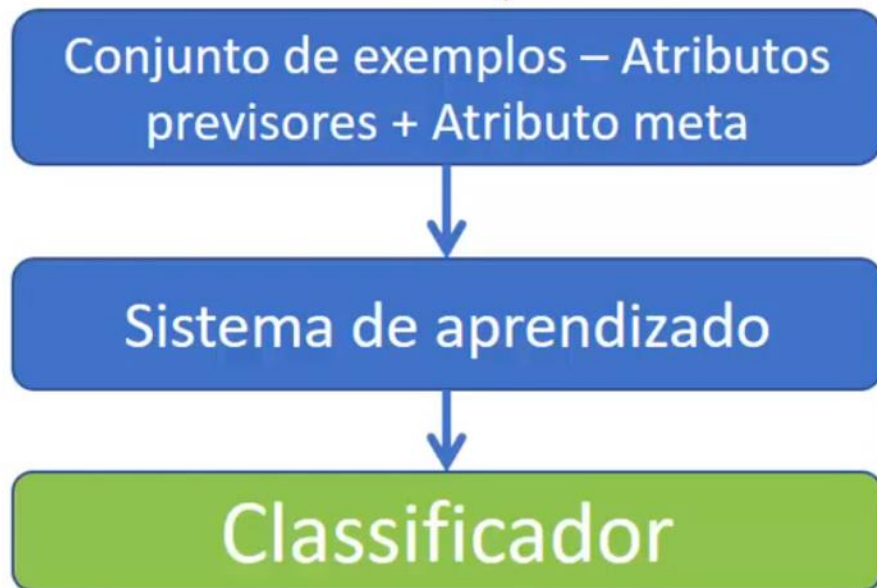
Tópicos Especiais em Sistemas de Informação

**Machine Learning:
Conceitos iniciais e introdução à classificação**

Ely – elydasilvamiranda@gmail.com

Resumo sobre Classificação

Fase 1



Fase 2

