Introdução

Olá pessoas! Sejam bem-vindas aos meus apontamentos de algoritmos de Machine Learning.

Para quem não me conhece chamo-me Tiago, tenho 20 anos e sou português. Espero que a minha escrita não atrapalhe a aprendizagem de ninguém, no caso de terem alguma questão não hesitem em mandar uma mensagem nas minhas redes sociais (ver no fim do documento) ou abram uma issue aqui no github.

O porquê de eu estar aqui a mostrar os meus apontamentos, é pelo simples motivo de alguém poder aprender algo novo e/ou entender algo que até agora não conseguiu! Eu mesmo sou um total iniciante na área de ciência de dados, o que vou escrever é um resumo da leitura de 2 ótimos livros e de artigos que vi na Internet. Cito abaixo os 2 livros que li:

- An Introduction to Statistical Learning: With Applications in R
- Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow, 2nd Edition

Ambos estão em inglês, contudo os meus resumos/apontamentos estão completamente em português! Nas explicações dos algoritmos tentarei explicar a matemática por trás deles, lembrando que não será algo super técnico! Como disse eu sou leigo na matéria.

Nota:

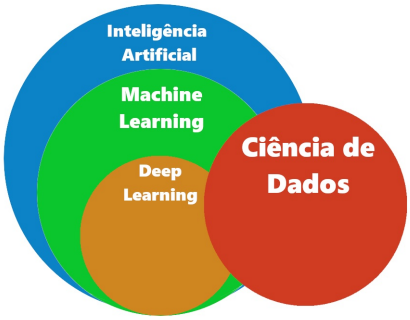
O segundo livro tem uma parte de Machine Learning e outra de Deep Learning. Estarei somente a usar como base a parte do Machine Learning, quem sabe no futuro inclua resumos de Deep Learning também...

O que é Ciência de Dados?

Esta é uma pergunta que muitos fazem e poucos sabem responder (incluindo eu). Pode-se dizer que ciência de dados é a área que **estuda dados e retira informações relevantes deles**, e que posteriormente usará essas informações para prever outras informações.

Com certeza não é apenas criar modelos preditivos, não é apenas olhar para um gráfico ou tabela, não é só tratar dados ... Na realidade é todo esse conjunto e mais algumas coisas que tornam esta área o que ela é.

Machine Learning é apenas uma sub-área de Ciência de Dados, assim como Deep Learning é uma sub-área de Machine Learning e todos elas encaixam-se na grande área de Inteligência Artificial.
No fim, não passa tudo de um conjunto de ferramentas e técnicas para estudar dados.



Nota:

Para quem quiser saber como iniciar na área de ciência de dados ou mesmo entender melhor o que é, têm este artigo super completo da Mikaeri Ohana: Medium (https://medium.com/@mikaeriohana/como-iniciar-na-carreira-de-ci%C3%AAncia-de-dados-9b37aa525181)

Para quem é direcionado estes apontamentos?

Para qualquer um que tenha interesse na área de ciência de dados e queira saber como os algoritmos realmente funcionam! Não precisam de saber matemática avançada nem de ter experiência com ciência de dados. O único requisito é **saber Python**, porque será a linguagem usada nos exemplos.

Como serão separados os apontamentos?

Bem a ideia é separar os documentos por pastas, algo deste género:

- **00. Introdução**
  - Introdução.pdf
  - Introdução.ipynb
- **01. Numpy**
  - Numpy.pdf
  - Numpy.ipynb
  - Numpy.py
- **02. Pandas**
  - Pandas.pdf
  - Pandas.ipynb
  - Pandas.py
- **03. Matplotlib**
  - Matplotlib.pdf
  - Matplotlib.ipynb
  - Matplotlib.py
- **04. Análise de Dados**
  - Análise de Dados.pdf
  - Análise de Dados.ipynb
  - Análise de Dados.py
- **05. Machine Learning**
  - **Regressões**
    - **Regressões Lineares**
      - Regressões Lineares.pdf
      - Regressões Lineares.ipynb
      - Regressões Lineares.py
    - **Árvore de Decisão(Regressão)**
      - Árvore de Decisão(Regressão).pdf
      - Árvore de Decisão(Regressão).ipynb
      - Árvore de Decisão(Regressão).py
    - **Random Forest(Regressão)**
      - Random Forest(Regressão).pdf
      - Random Forest(Regressão).ipynb
      - Random Forest(Regressão).py
  - **Classificadores**
    - **Perceptron**
      - Perceptron.pdf
      - Perceptron.ipynb
      - Perceptron.py
    - **Adaline**
      - Adaline.pdf
      - Adaline.ipynb
      - Adaline.py
    - **Regressão Logística**
      - Regressão Logística.pdf
      - Regressão Logística.ipynb
      - Regressão Logística.py
    - **Árvore de Decisão**
      - Árvore de Decisão.pdf
      - Árvore de Decisão.ipynb
      - Árvore de Decisão.py
    - **Random Forest**
      - Random Forest.pdf
      - Random Forest.ipynb
      - Random Forest.py
    - **Boosting**
      - Boosting.pdf
      - Boosting.ipynb
      - Boosting.py

    - **Bagging**
        - Bagging.pdf
        - Bagging.ipynb
        - Bagging.py
    - **AdaBoost**
        - AdaBoost.pdf
        - AdaBoost.ipynb
        - AdaBoost.py
    - **SVM**
        - SVM.pdf
        - SVM.ipynb
        - SVM.py
    - **KNN**
        - KNN.pdf
        - KNN.ipynb
        - KNN.py
- **Feature Selection**
    - **Sequential Backward Selection**
        - Sequential Backward Selection.pdf
        - Sequential Backward Selection.ipynb
        - Sequential Backward Selection.py
- **Feature Extraction**
    - **Linear Discriminant Analysis**
        - LDA.pdf
        - LDA.ipynb
        - LDA.py
    - **Principal Component Analysis**
        - PCA.pdf
        - PCA.ipynb
        - PCA.py
- **Avaliação do Modelo**
    - **K-Fold**
        - K-Fold.pdf
        - K-Fold.ipynb
        - K-Fold.py
    - **GridSearch**
        - GridSearch.pdf
        - GridSearch.ipynb
        - GridSearch.py
    - **Métricas**
        - Métricas.pdf
        - Métricas.ipynb
        - Métricas.py

Bem longa a lista de coisas que vou mostrar, mas acho que deu para entender como as coisas serão organizadas.

Nota:

Serão sempre providenciados os ficheiros **.pdf** e **.ipynb**, os ficheiros .py só serão disponibilizados caso as explicações contenham código.

Deixem-me explicar o que são essas tais de features, X e y

É meio estranho explicar já algo numa introdução, eu sei, mas é importante já terem uma noção do que estas 3 coisas são, porque vão encontrar em artigos e código algumas dessas nomenclaturas.

Vamos pensar no seguinte, vocês recebem um ficheiro excel que contém uma tabela idêntica a esta:

Out[2]:

	Área do lote(m²)	Classificação Qualidade	Preço
1	8450	7	208500
2	9600	6	181500
3	11250	7	223500
4	9550	7	140000
5	14260	8	250000
6	14115	5	143000
7	10084	8	307000
8	10382	7	200000
9	6120	7	129900
10	7420	5	118000

Bem esta tabela, como todas as outras, tem linhas e colunas. Neste caso, estamos a falar de preços de casas/habitações, logo esta é uma tabela que mostra uma dada casa por linha. Podemos ver que temos 10 linhas, logo temos informações sobre 10 casas.

As linhas nós chamamos de ***amostras*** ou, em inglês, ***samples***. Vejam elas como objetos, como eu disse ainda agora, cada linha é uma habitação/casa.

Muito bem, agora nas colunas podemos ver que temos:

- Área do lote
- Classificação Qualidade

As colunas parecem tentar descrever cada casa. Por exemplo sabemos que a casa da linha 5 tem 14 260 m^2 e, de 0 a 10, tem uma qualidade de 8! A estas colunas nós chamamos de ***features/variáveis explicativas/X*** e há mais nomes, mas ficamos com estes para já. Faz sentido pois elas descrevem/"explicam" como são as casas.

Nota:

A nomenclatura *X* é utilizada na programação, não em artigos e/ou livros teóricos. Vocês verão mais à frente quando eu fizer exemplos práticos.

Expliquei tudo, exceto a coluna preço ... Ela também não deveria ser uma variável explicativa??
Eu não vos disse qual era o nosso objetivo tendo estes dados, mas podemos supor que queremos no futuro prever o preço de uma casa tendo:

- Área do lote
- Classificação Qualidade

Logo o preço é aquilo que nós queremos como resultado, após a análise das nossas features! Esses resultados são colocados numa variável, convencionalmente, chamada de ***y*** (claro no caso de estarmos a programar), caso contrário podemos chamar-lhe, neste caso, de ***resultado/outcome***.

De certeza que alguns de vocês estão a perguntar?: *Como assim prever resultados?! Os resultados já estão lá!!*

Verdade, os resultados estão lá, mas apenas para os dados fornecidos! Imaginemos que agora chegava uma nova casa, e tinha:

- Área do lote = 7950 m²
- Classificação Qualidade = 5

Nós ainda não sabemos o preço desta nova casa, mas podemos usar os nossos X e y que já conhecemos para treinar um algoritmo que posteriormente será capaz de prever novos y com base nos X fornecidos.

Por exemplo, se já tivessemos um modelo preditivo treinado, agora podíamos passar estas features e obteriamos um **resultado estimado**, ou seja, uma estimativa do possível preço para a casa.

Vamos colocar features em outra perspetiva

Agora que vocês já têm uma ideia do que são features, vamos ver isto de outra forma.

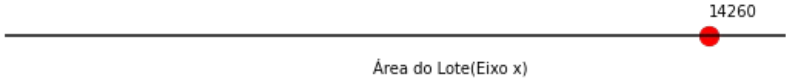
Primeiramente quero deixar claro que ***X*** e ***y*** não são coordenadas cartesianas ... Isto é, **eles NÃO são valores de x e y para vocês colocarem num gráfico como um ponto!** São apenas nomenclaturas regularmente utilizadas enquanto programamos.

Para tirar qualquer confusão, vamos visualizar os ***X***, ou melhor dizendo, features(variáveis explicativas) de outra forma.

Vamos observar o eixo abaixo:

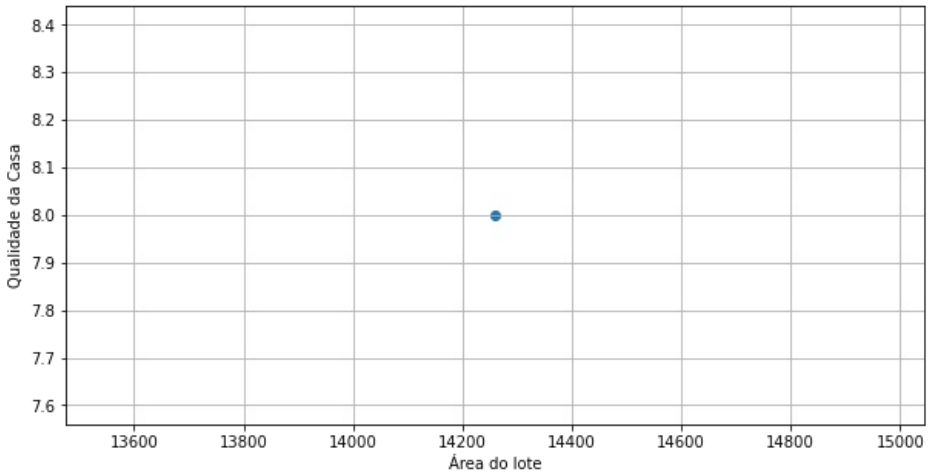
Out[37]:

```
Text(11000, 0.99, 'Área do Lote(Eixo x)')
```



Como podemos observar acima a área do lote da casa 5 está representado por um ponto, no eixo *x*.
Isto quer dizer que uma coluna pode ser representada em uma dimensão, ou seja, 1 eixo.

Vamos observar outro gráfico:



Agora nós usamos a coluna **Área do lote** e a coluna **Qualidade da Casa** para representar um ponto num gráfico cartesiano! Neste caso a área do lote é o valor no eixo *x* e a qualidade da casa é o valor no eixo *y*, portanto temos que a qualidade da casa é 8 quando o a área é de 14600 (estes são valores da casa 5).

Agora eu poderia representar, visualmente, em 3D utilizando outra coluna (no caso deste exemplo só temos 2 colunas/features) e dai para a frente já seria difícil representar visualmente ... Por isso que a matemática é importante.

Com isto podemos concluir que as features/X são **dimensões**! Cada variável explicativa é uma dimensão, quantas mais dimensões mais trabalho daremos ao nosso modelo para assim treinar e prever resultados.

Resumo

- As colunas de uma tabela são consideradas variáveis explicativas/features
- As linhas de uma tabela são as amostras
- A coluna que tem o resultado que pretendemos obter, não é considerada feature e sim a coluna dos resultados/outcomes
- X é a forma de nomear a variável que carrega as features (em programação)
- y é a forma de nomear a variável que carrega os resultados (em programação)
- Features são dimensões, cada feature é uma dimensão (1D, 2D, 3D, 4D ...)

Acredito que agora devem estar confusos e com uma dor de cabeça enorme! E isso é algo **101% normal**! Durante a leitura destes apontamentos, leitura de livros e/ou artigos de ciência de dados, vocês vão ficar assim várias vezes. E sabem que mais? Não é errado não aprender à 1ª, à 2ª, à 3ª, à 100000ª vez! Deêm tempo a vocês próprios para processar tudo aquilo leêm, leiam aos poucos, expliquem para vocês mesmos, tentem visualizar na vossa cabeça o que aprenderam. Todos nós demoramos o nosso tempo, não se apressem, e claro qualquer dúvida que tenham estou à disposição para vos ajudar!!!

Antes de partirmos para as explicações de algoritmos, vou dar umas explicações básicas acerca das 3 principais bibliotecas que vão utilizar, e muito, em ciência de dados! No entanto isso fica para a próxima leitura.

Contactos

Twitter (https://twitter.com/iN127pkt)
Instagram (https://www.instagram.com/t_1g4_x/)
Email (tiagodeha@protonmail.com)