04/7/2020 - (8h a 12h) Link para aula <https://meet.google.com/oux-bihv-kfb>

**K-means Clustering**

O agrupamento K-means é um tipo de aprendizado não supervisionado, usado com o conjunto de dados não rotulado. O objetivo deste algoritmo é encontrar grupos K nos dados. O algoritmo trabalha iterativamente para atribuir cada ponto de dados a um dos grupos K com base nos recursos fornecidos. Os pontos de dados são agrupados com base na similaridade de recursos. Os resultados do algoritmo de agrupamento K-means são:

* Os centróides dos clusters K, que podem ser usados para rotular novos dados
* Etiquetas para os dados de treinamento (cada ponto de dados é atribuído a um único cluster)

K-means funciona definindo clusters esféricos que são separáveis de forma que o valor médio converja para o centro do cluster. Por esse motivo, o K-Means pode ter um desempenho inferior às vezes.

**Casos de Uso:**

* Classificação de documentos
* Otimização da loja de entrega
* Segmentação de Clientes
* Detecção de fraudes de seguros etc.

**Algoritmo :**

O algoritmo K-means divide um conjunto de n amostras X em k clusters disjuntos cᵢ, i = 1, 2,…, k, cada um deles descrito pela média (centróide) μᵢ das amostras no cluster. K-significa assume que todos os grupos k têm variação igual. Grosso modo, o algoritmo executa as seguintes etapas:

1. Escolha o número (k) de clusters;
2. Especifique as sementes do cluster (defina a posição inicial dos centróides);
3. Atribua cada ponto a um centróide com base em sua proximidade (se o centróide α for o centróide mais próximo do ponto p, atribua p a α, faça-o para todos os pontos;
4. Ajuste o centróide (recalcule a posição de cada centróide com base nos pontos atribuídos a eles);
5. Repita as etapas 3 e 4, até atingir um critério de parada;
6. Após atingir um critério de parada (geralmente o número máximo predefinido de iterações), finalize a execução.

Entradas do algoritmo de clustering são o número de clusters Κ e o dataset. O algoritmo começa com estimativas iniciais para os centróides, que podem ser gerados aleatoriamente ou selecionados aleatoriamente no conjunto de dados. O algoritmo itera entre duas etapas:

**1. Etapa de atribuição de dados:**

Cada centróide define um dos clusters. Nesta etapa, cada ponto de dados com base na distância euclidiana quadrada é atribuído ao centróide mais próximo. Se 𝑐𝑖ci é a coleta de centróides no conjunto C, então cada ponto de dados x é atribuído a um cluster com base em

min𝑐𝑖∈𝐶𝑑𝑖𝑠𝑡(𝑐𝑖,𝑥)2minci∈Cdist(ci,x)2

, onde dist( · ) é a distância euclidiana padrão (L2).

**2. Centroid etapa de atualização:**

Os centróides são recalculados calculando a média de todos os pontos de dados atribuídos ao cluster desse centróide.

O algoritmo itera entre as etapas um e dois até que um critério de parada seja atendido (nenhum ponto de dados altera os clusters, a soma das distâncias é minimizada ou é atingido um número máximo de iterações).

**Este algoritmo pode convergir para um ótimo local.**A avaliação de mais de uma execução do algoritmo com centróides iniciais aleatórios pode fornecer um resultado melhor.

**Escolhendo K**

Se o rótulo verdadeiro não for conhecido antecipadamente, o agrupamento K-Means pode ser avaliado usando **Critério de cotovelo**, **Coeficiente de silhueta**, validação cruzada, critérios de informação e salto teórico da informação método e o algoritmo G-means.

**Método do critério de cotovelo:**

A idéia por trás do método cotovelo é executar o agrupamento de médias k em um determinado conjunto de dados para um intervalo de valores de k (por exemplo, k = 1 a 10), para cada valor de k, calcular a soma dos erros ao quadrado (SSE).

Calcule a distância média entre os pontos de dados e o centróide do cluster. Aumentar o número de clusters (K) sempre reduzirá a distância para os pontos de dados, diminuindo assim essa métrica, chegando a zero quando K for igual ao número de pontos de dados. Portanto, o objetivo é escolher um pequeno valor de k que ainda tenha um SSE baixo.

Executamos o algoritmo para diferentes valores de K (digamos, K = 10 a 1) e plotamos os valores de K em relação ao SSE (soma dos erros ao quadrado). E selecione o valor de K para o ponto do cotovelo.

**Método do coeficiente de silhueta:**

Uma pontuação mais alta do coeficiente de silhueta refere-se a um modelo com clusters melhor definidos. O coeficiente da silhueta é definido para cada amostra e é composto por duas pontuações:

* A distância média entre uma amostra e todos os outros pontos da mesma classe.
* A distância média entre uma amostra e todos os outros pontos no próximo cluster mais próximo.

O coeficiente da silhueta é para uma única amostra e é dado como:

𝑠=𝑏−𝑎𝑚𝑎𝑥(𝑎,𝑏)s=b−amax(a,b)

Para encontrar o valor ideal de k para o KMeans, passe por 1..n para n\_clusters no KMeans e calcule o Coeficiente da silhueta para cada amostra.

Um coeficiente de silhueta mais alto indica que o objeto é bem correspondido ao seu próprio cluster e mal correspondido aos clusters vizinhos.

O algoritmo K-Means usa Distância Euclidiana, outras métricas de distância populares em Machine Learning são:

1. **Distância do cosseno**: Determina o cosseno do ângulo entre os vetores de pontos dos dois pontos no espaço dimensional n. Quanto mais próximos os vetores de pontos estiverem em ângulo, maior será a semelhança cosseno

cos𝜃=𝑎→.𝑏→∥𝑎→∥∥𝑏→∥=∑𝑛𝑖=1𝑎𝑖𝑏𝑖∑𝑛𝑖=1𝑎2𝑖∑𝑛𝑖=1𝑏2𝑖⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯√⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯√cos⁡θ=a→.b→∥a→∥∥b→∥=∑i=1naibi∑i=1nai2∑i=1nbi2

onde 𝑎→.𝑏→=∑𝑛𝑖=1𝑎𝑖𝑏𝑖=𝑎1𝑏1+𝑎2𝑏2+...+𝑎𝑛𝑏𝑛a→.b→=∑i=1naibi=a1b1+a2b2+...+anbn

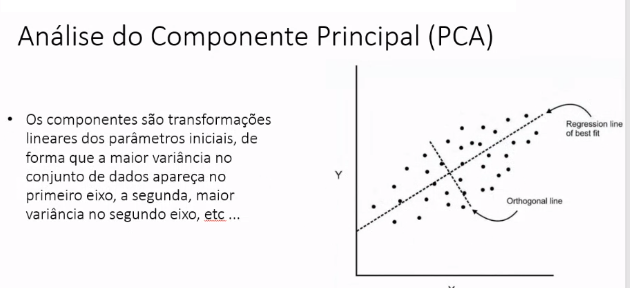
1. **Distância Manhattan**: é a soma total da diferença entre as coordenadas x e as coordenadas y.

𝑀𝑎𝑛ℎ𝑎𝑡𝑡𝑎𝑛𝐷𝑖𝑠𝑡𝑎𝑛𝑐𝑒=|𝑥1–𝑥2|+|𝑦1–𝑦2|ManhattanDistance=|x1–x2|+|y1–y2|

Tanto o RMSE quanto o MAE são maneiras de medir a distância entre dois vetores: o vetor de previsões e o vetor de valores-alvo. São possíveis várias medidas ou normas de distância:

* A computação da raiz de uma soma dos quadrados (RMSE) corresponde à norma euclidiana: é a noção de distância com a qual você está familiarizado. É também chamada de norma ℓ2(...)
* Computar a soma dos absolutos (MAE) corresponde à norma ℓ1, (...). Às vezes, é chamada de norma Manhattan, porque mede a distância entre dois pontos em uma cidade se você puder apenas viajar ao longo de quarteirões ortogonais.

PCA – Analise do componente Principal

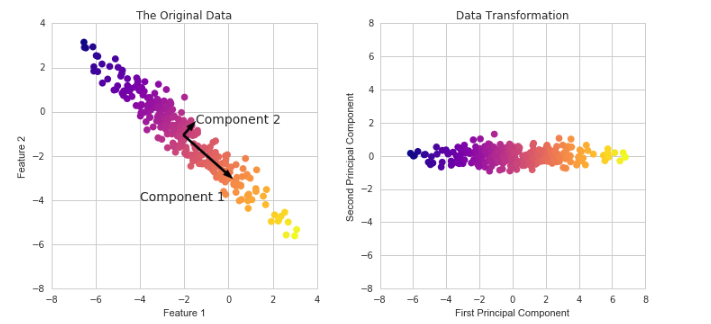


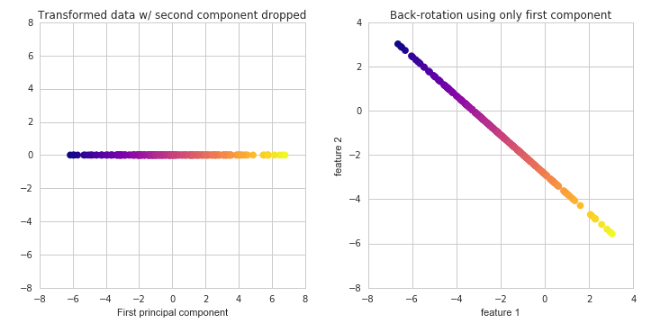
**Análise do componente principal (Principal Component Analysis - PCA)**[**¶**](http://localhost:8888/notebooks/_Pos_Uniceub/curso_introducao_aprendizagem_de_maquina-master/06_Analise_do_componente_principal_PCA/6.3_PCA.ipynb#An%C3%A1lise-do-componente-principal-(Principal-Component-Analysis---PCA))

Vamos discutir PCA! Uma vez que este não é exatamente um algoritmo de Machine Learning completo, mas apenas um algoritmo de aprendizagem não supervisionado.

**Revisão de PCA**

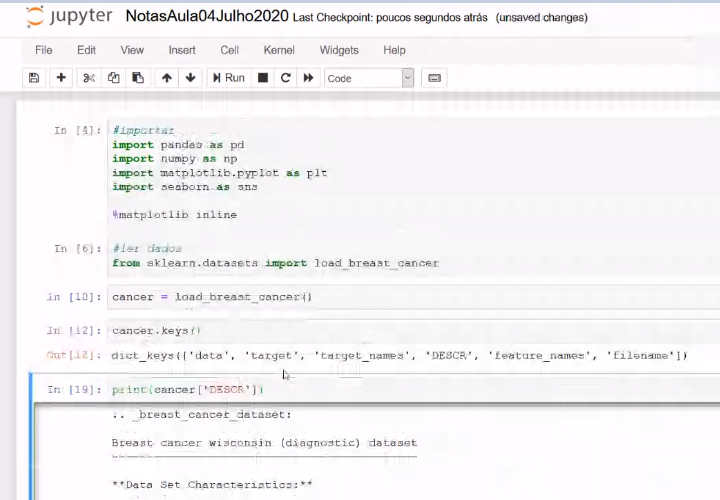
Lembre-se de que o PCA é apenas uma transformação dos seus dados e tenta descobrir quais recursos explicam a maior variação em seus dados. Por exemplo:

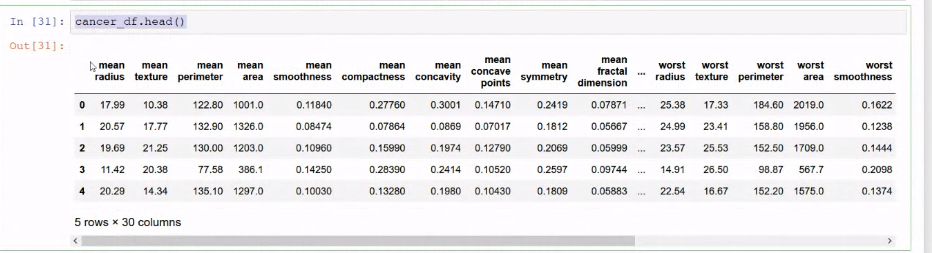


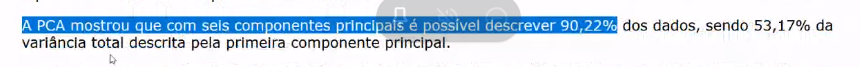


Exemplo: Tenho várias variáveis para avaliar um apt. PCA faz isso verifica a variáveis mais relevantes.

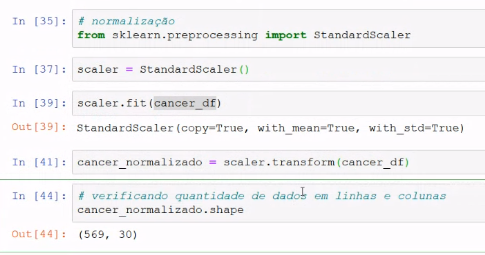
Exemplo



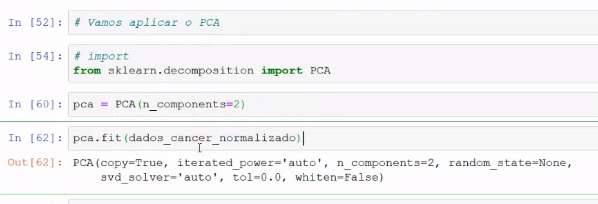


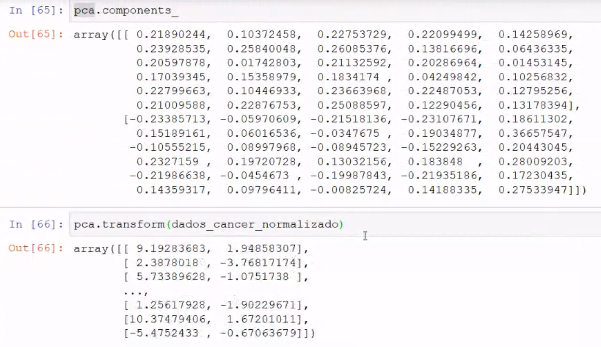


Normalizar



Usar o PCA





Passou para 2 colunas

