

Aula 03- Pré-processamento

1001524 – Aprendizado de Máquina I
2023/1 - Turmas A, B e C
Prof. Dr. Murilo Naldi

naldi@dc.ufscar.br

Agradecimentos

- Parte do material utilizado nesta aula foi cedido pelos professores André C.P.L.F de Carvalho e Ricardo J.G.B. Campello e, por esse motivo, o crédito deste material é deles
- Parte do material utilizado nesta aula foi disponibilizado por M. Kumar no endereço:
 - www-users.cs.umn.edu/~kumar/dmbook/index.php
- Agradecimentos a Intel Software e a Intel IA Academy pelo material disponibilizado e recursos didáticos

Copyright © 2017, Intel Corporation. All rights reserved.

Conteúdo

- Pré-processamento
- Problema da dimensionalidade
 - Redução de atributos
 - Seleção de atributos

Pré-processamento

- Primeira etapa para preparar os dados
- Consiste em preparar os dados para o processo de aprendizado
- Dentre as técnicas:
 - Agregação
 - Amostragem
 - Seleção de atributos
 - Redução de dimensionalidade

Agregação

- Consiste em reduzir o tamanho do conjunto de dados agregando dois ou mais objetos em um único
- Atributos agregados são mais robustos a variações na coleta dos dados
- Na agregação, atributos quantitativos podem ser somados, enquanto os qualitativos pode ser omitido ou resumido



Agregação

- Exemplos:
 - Temporal: dados de vendas de um determinado produto podem ser agregados por dia ou mês
 - De origem: dados de diferentes lojas de uma mesma empresa podem ser agregados
 - De categoria: dados de uma mesma categoria podem ser agregados

Amostragem

- Consiste em selecionar uma amostra representativa dos dados e utilizá-la
- Reduz consideravelmente o tamanho do conjunto
- Contudo, é preciso que ela possua as mesmas características do conjunto de dados

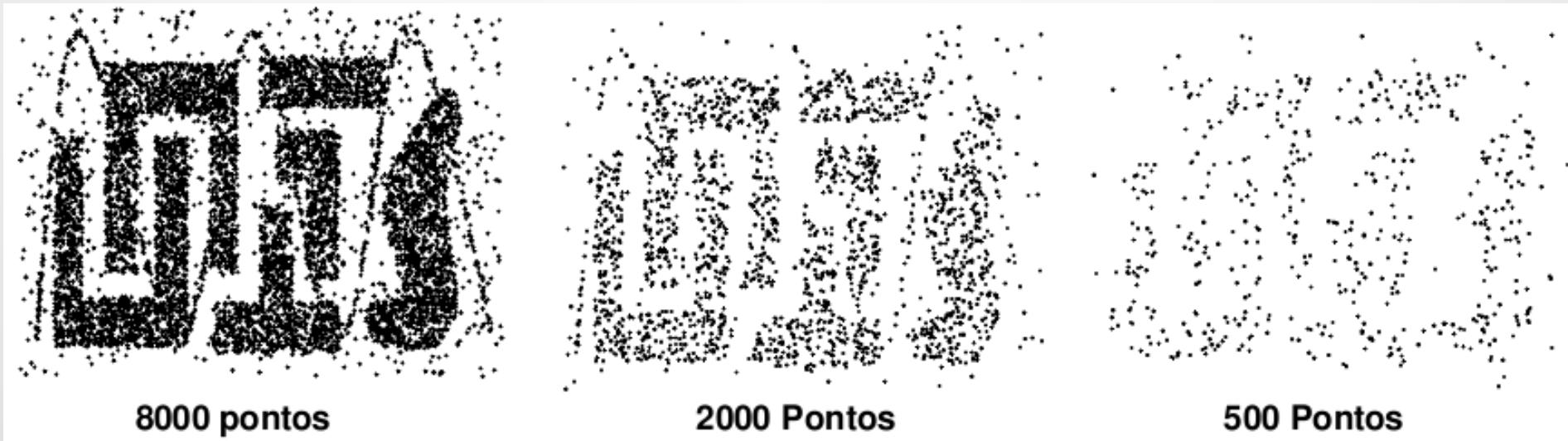


Tipos de amostragem

- Amostragem **aleatória simples**:
 - Com substituição
 - Sem substituição
- Amostragem **estratificada**:
 - Número de objetos igual por grupo
 - Número de objetos proporcional ao tamanho do grupo

Tamanho da amostra

- O tamanho da amostra influênci na quantidade de informação que a amostra possui em relação ao conjunto de dados



Amostragem

- Código

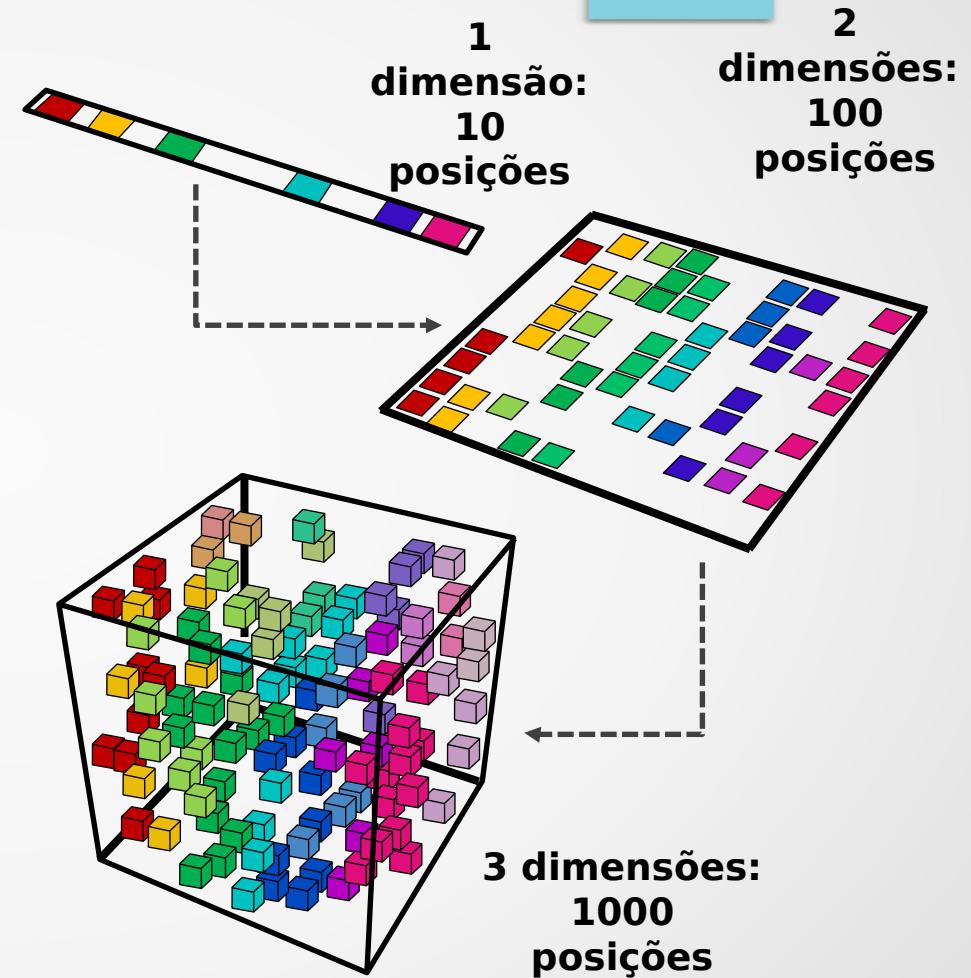
```
# Amostragem de 5 colunas sem reposição  
sample = (data.sample(n=5, replace=False, random_state=42))  
print(sample.iloc[:, -3:])
```

- Saída

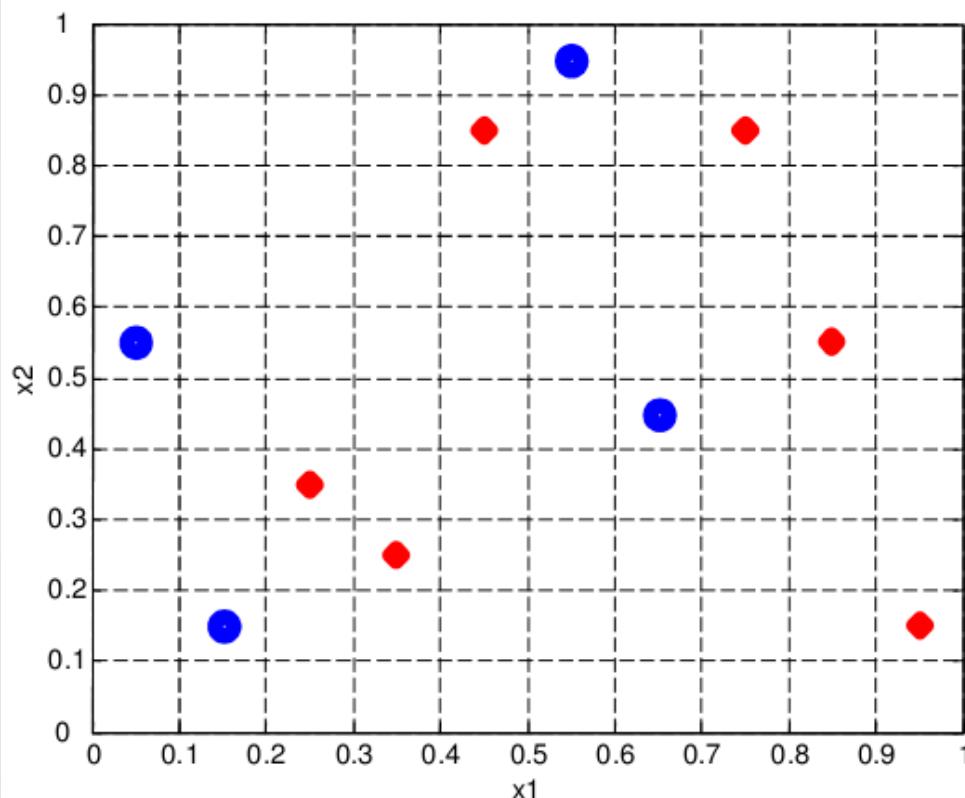
	petal_length	petal_width	species
73	4.7	1.2	Iris-versicolor
18	1.7	0.3	Iris-setosa
118	6.9	2.3	Iris-virginica
78	4.5	1.5	Iris-versicolor
76	4.8	1.4	Iris-versicolor

Maldição de dimensionalidade

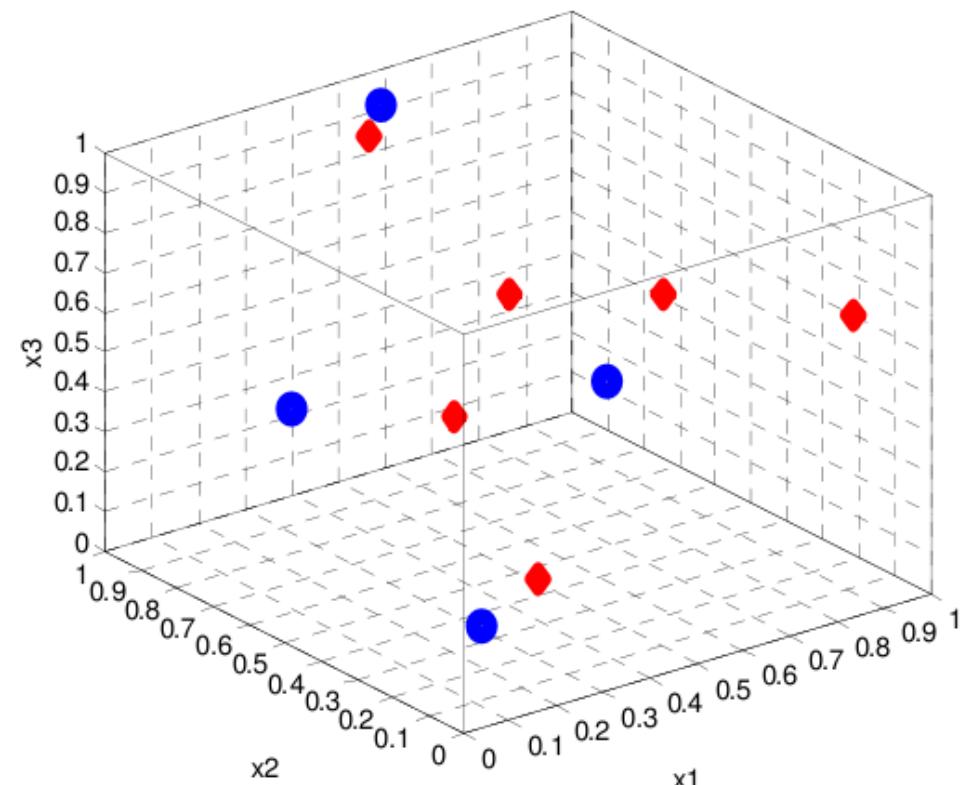
- Teoricamente, aumentar dimensões deveria melhorar a performance
- Mas não é o caso
- A medida que a quantidade de atributos aumenta, os dados se tornam mais dispersos



Exemplo aumento dimensões



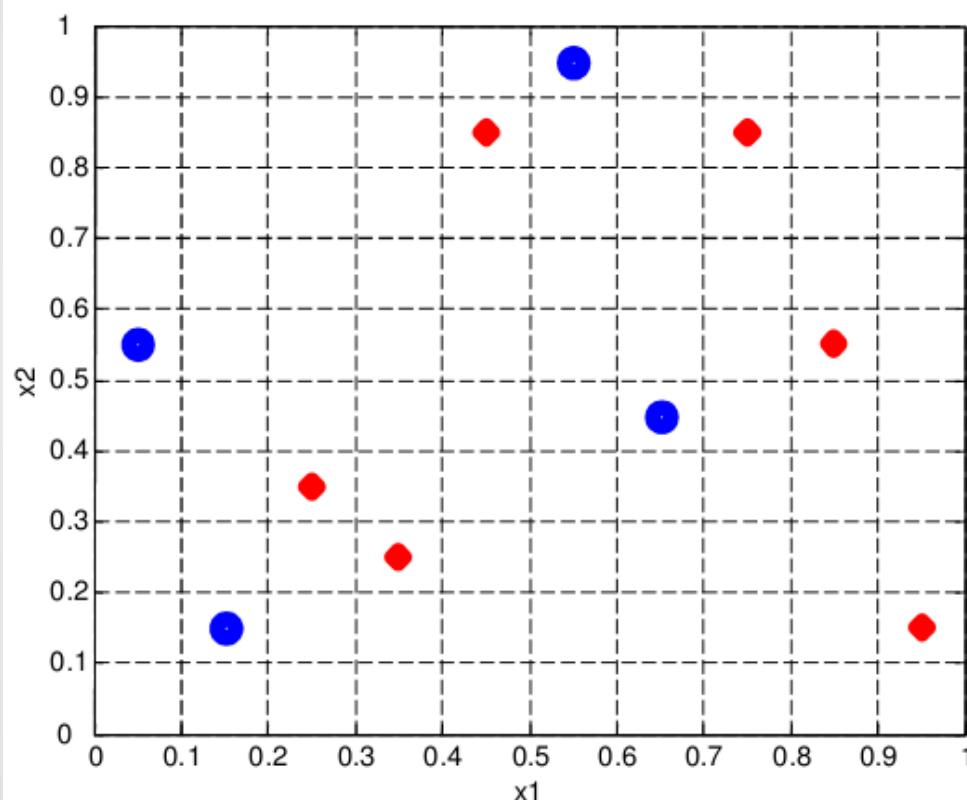
● Exemplos positivos ◆ Exemplos negativos



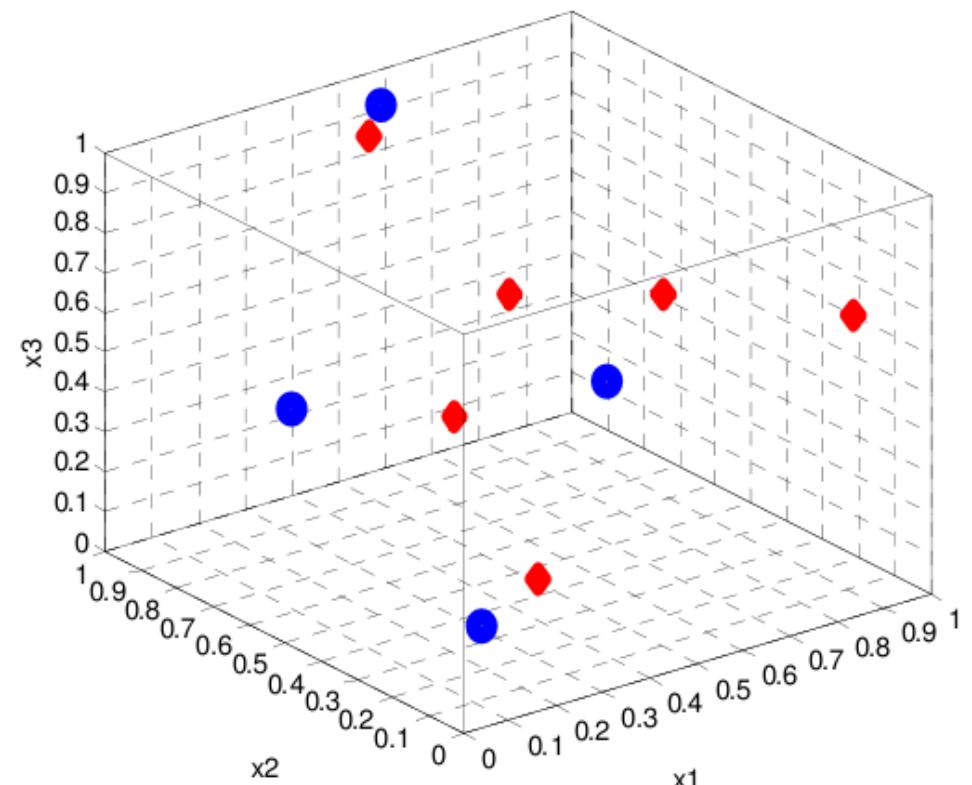
Maldição de dimensionalidade

- Para que a estrutura dos dados se mantenha, o número de exemplos necessários aumenta exponencialmente com a dimensionalidade
- **Para classificação:**
 - Impede a construção de um modelo que atribua de forma confiável uma classe a todos os objetos possível
- **Para agrupamento:**
 - A densidade e distância entre os objetos se tornam menos significativas

Exemplo



● Exemplos positivos ◆ Exemplos negativos



Redução de dimensionalidade

- Consiste em métodos que reduzem a **quantidade de atributos** do conjunto de dados
- Necessário, especialmente quando há milhares de atributos envolvidos
 - Termos de documentos
 - Séries temporais
 - Expressão de genes

Redução de dimensionalidade

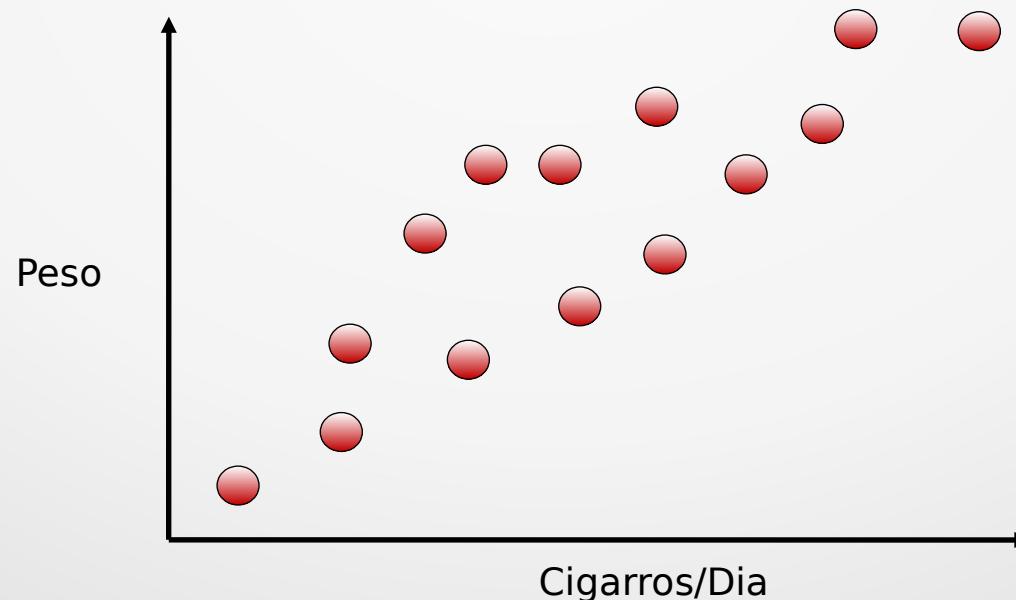
- Pode eliminar atributos irrelevantes e reduzir o ruído
- Cria novos atributos a partir da combinação de atributos antigos
 - Não confundir com seleção de atributos!
(descrito mais a diante...)

ID	Proprietário	E. Civil	Renda	Investe
1	Sim	Solteiro	1.500,00	Pouco
2	Não	Casado	812,00	Muito
3	Não	Solteiro	2.345,67	Não
4	Sim	Casado	4.768,00	Muito
5	Não	Divorciado	734,00	Não
6	Não	Casado	3.900,00	Pouco
7	Sim	Divorciado	2.100,00	Muito

ID	Estado	Renda	Investe
1	A	1.500,00	Pouco
2	B	812,00	Muito
3	C	2.345,67	Não
4	D	4.768,00	Muito
5	E	734,00	Não
6	B	3.900,00	Pouco
7	F	2.100,00	Muito

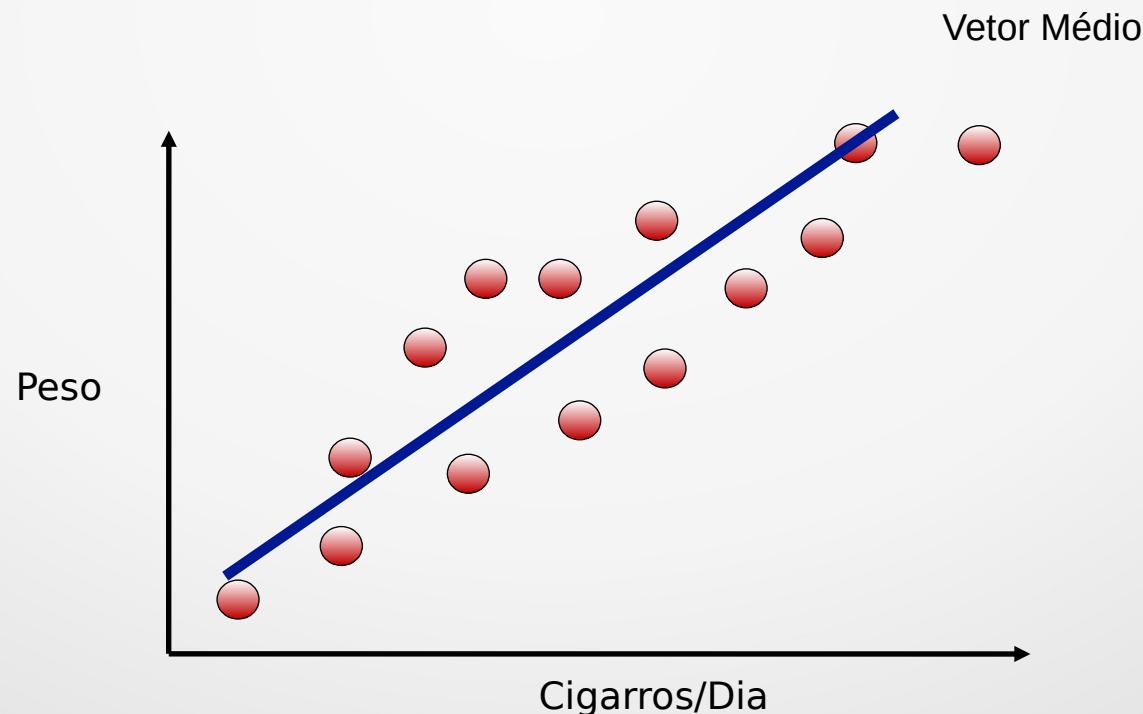
Redução e correlação

- Correlação exerce papel importante da seleção de atributos a serem reduzidos
 - Se atributos possuem alta correlação, podem ser reduzidos sem muita perda de informação
 - Observe o gráfico a seguir



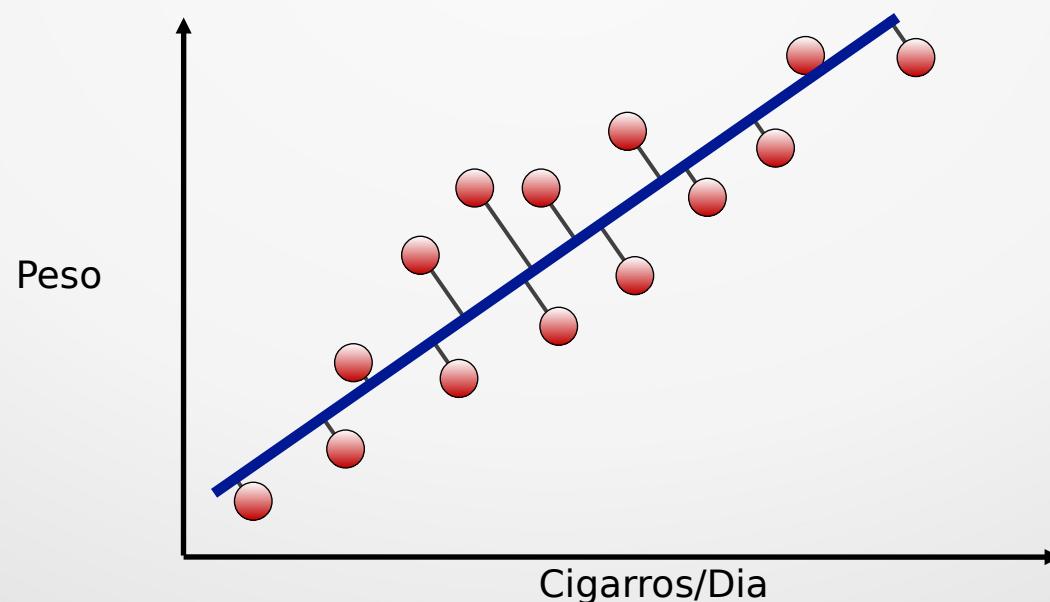
Redução e correlação

- A alta correlação entre peso e cigarros por dia pode ser reduzida em uma única componente
 - Traçar um vetor médio



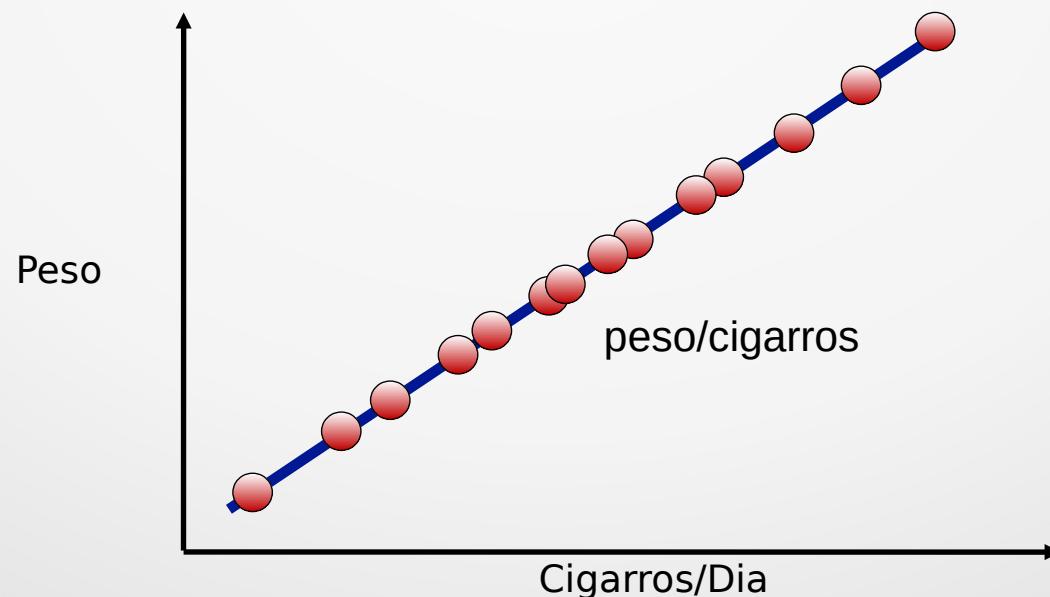
Redução e correlação

- A alta correlação entre peso e cigarros por dia pode ser reduzida em uma única componente
 - Traçar um vetor médio
 - Projetar os pontos sobre o vetor



Redução e correlação

- A alta correlação entre peso e cigarros por dia pode ser reduzida em uma única componente
 - Traçar um vetor médio
 - Projetar os pontos sobre o vetor
 - Gerar um novo atributo único: peso/cigarros



Técnicas redução de dimensionalidade

- Grande parte vem da álgebra linear
 - Análise de Componentes Principais (PCA)
 - Para atributos contínuos
 - Os atributos gerados devem ser combinações lineares, ortogonais e capturem o máximo de variações nos dados
 - Decomposição em Valores Singulares (SVD)
 - Fatoração dos dados em uma matriz positiva e normal

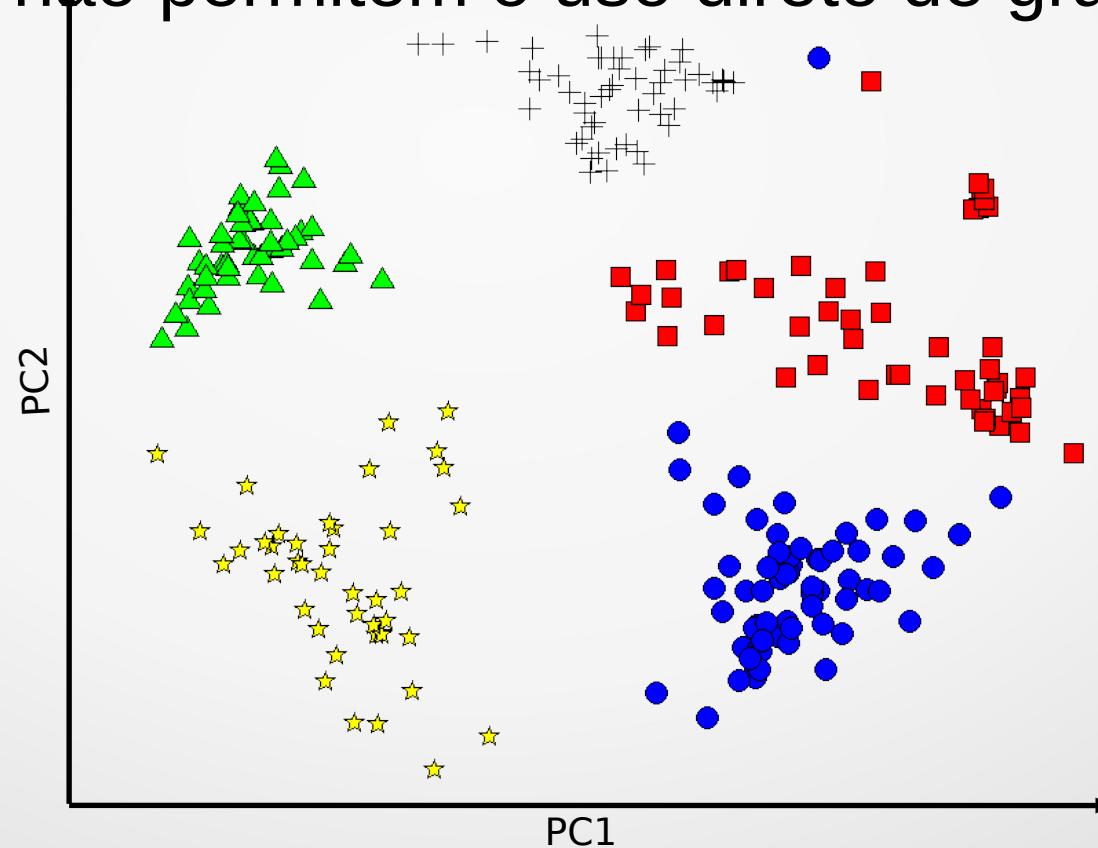
Análise de Componentes Principais

- Utiliza uma transformação ortogonal para converter um conjunto de observações (possivelmente correlacionadas) em um conjunto de variáveis não correlacionadas.
 - O número de componentes é menor que o de atributos
 - Se os dados tiverem distribuição normal, as componentes são garantidamente independentes

Exemplo de uso

- Visualização
 - Documentos com muitos atributos na forma *bag-of-words* não permitem o uso direto de gráficos

Conjunto de artigos de cinco revistas distintas:
American Political Science Review,
DNA Research,
Monthly Weather Review,
British Food Journal,
Transactions on Mobile Computing



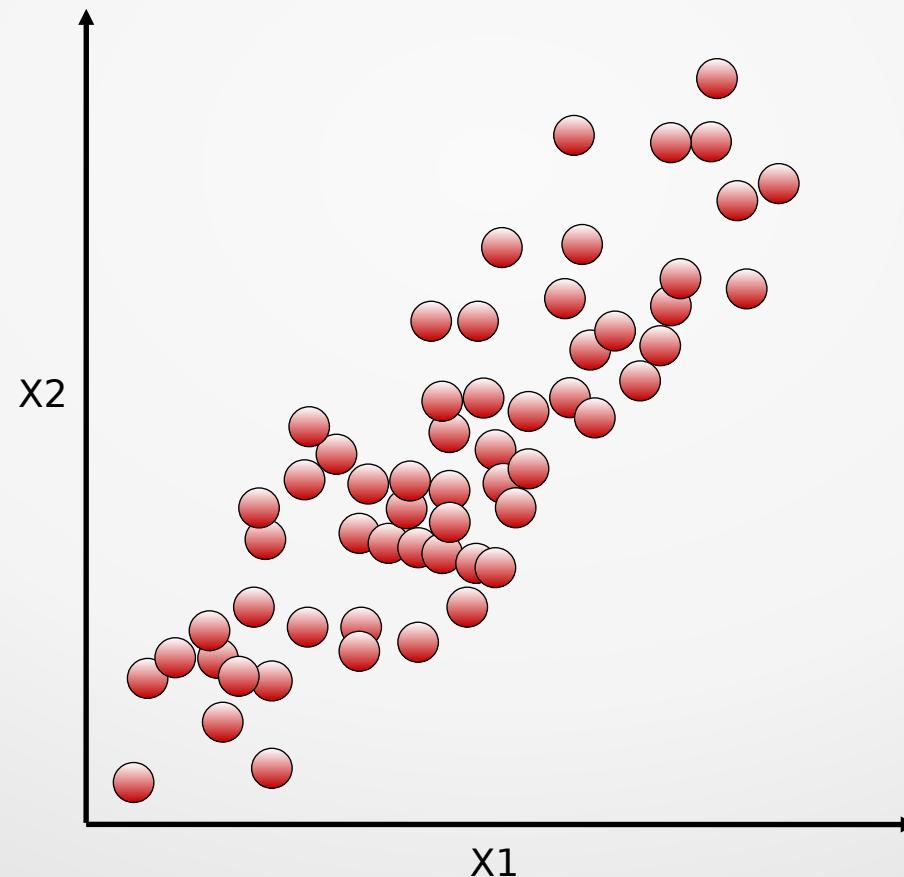
Exemplo de uso

- Mas como fazer para colocar eles em um gráfico de duas dimensões utilizando PCA?
 - Cada palavra é um atributo
 - Cada documento, uma tupla

Documento	Abacate	Avião	Beterraba	Casa	Dados
1	213	0	35	0	0
2	18	0	123	0	0
3	0	0	0	0	0
4	0	7	0	3	7
5	0	2	0	5	15
6	14	0	0	17	0
7	0	0	12	0	0

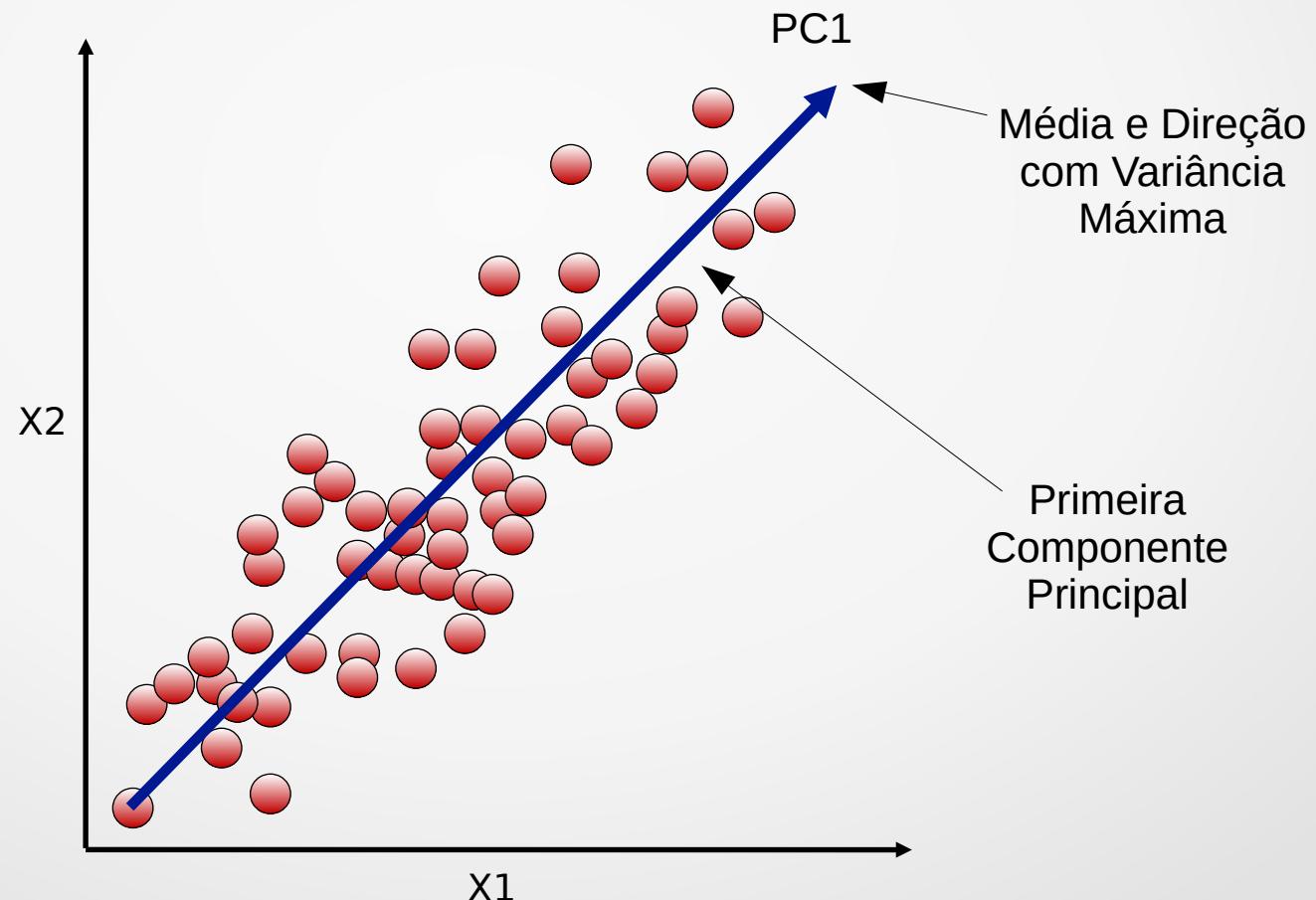
Análise de Componentes Principais

- Exemplo, considere com conjunto de dados abaixo com os atributos x_1 e x_2 .



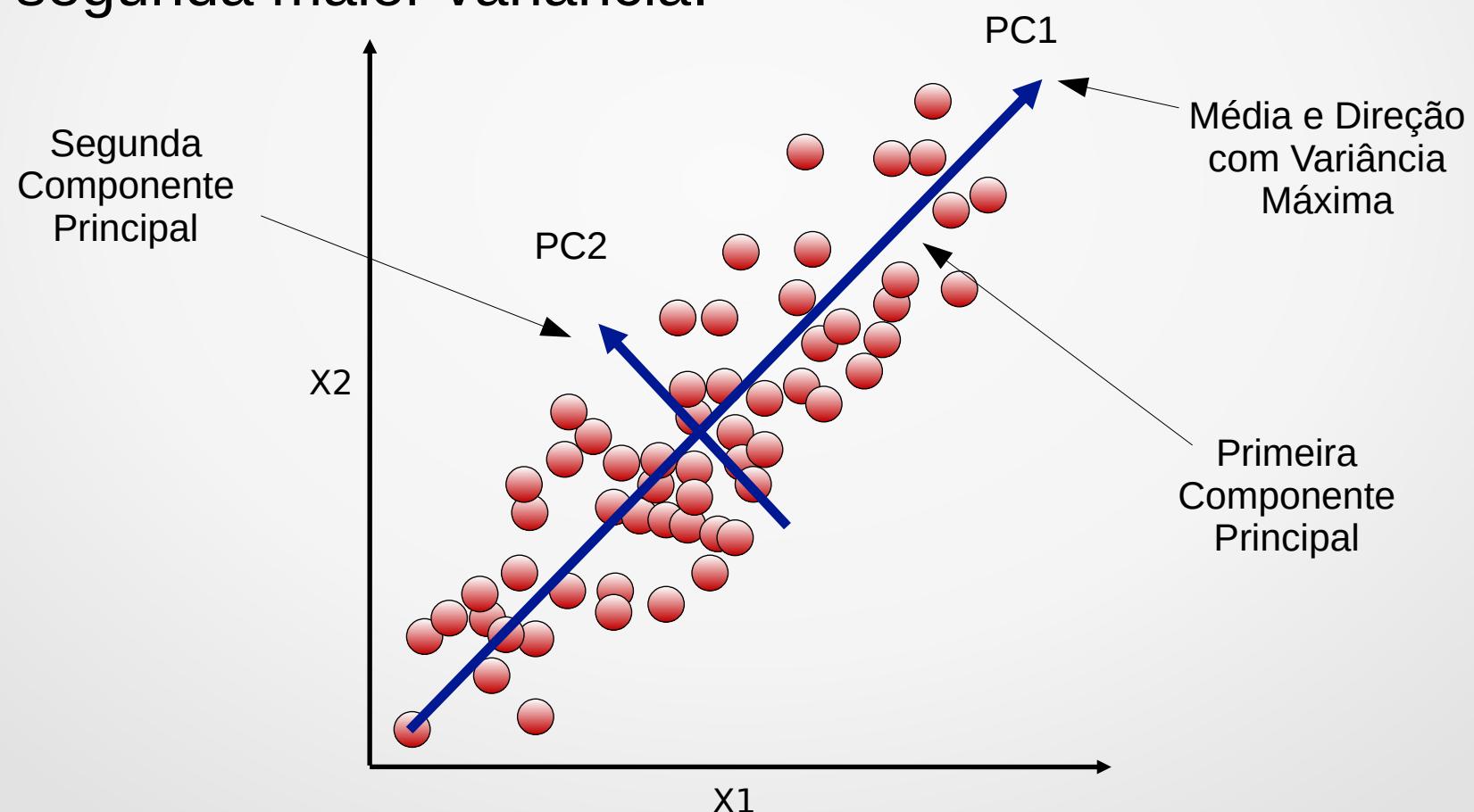
Análise de Componentes Principais

- Para encontrar a primeira componente principal, utilize-se o vetor com a média e direção de maior variância.



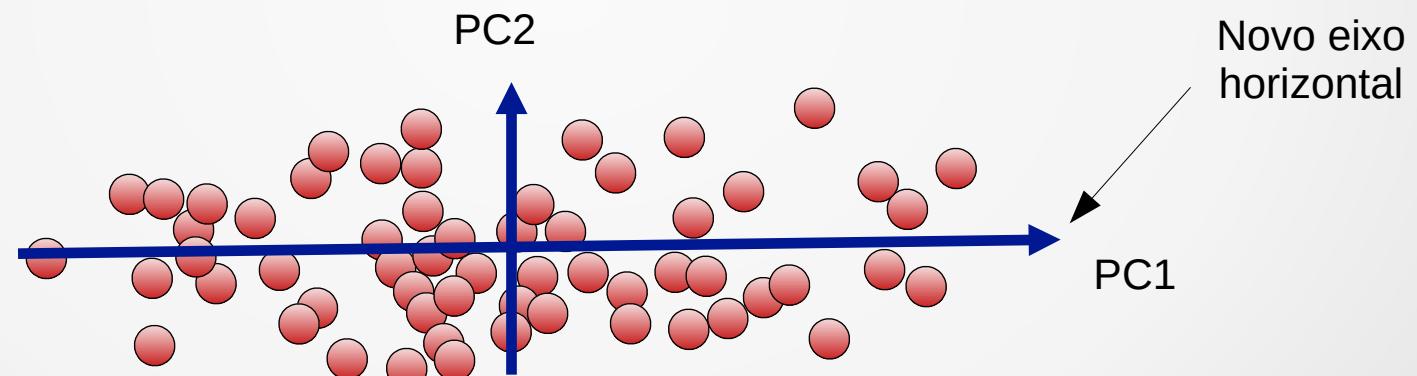
Análise de Componentes Principais

- A segunda componente principal é um vetor orthonormal com a primeira direção que possui média e direção com a segunda maior variância.



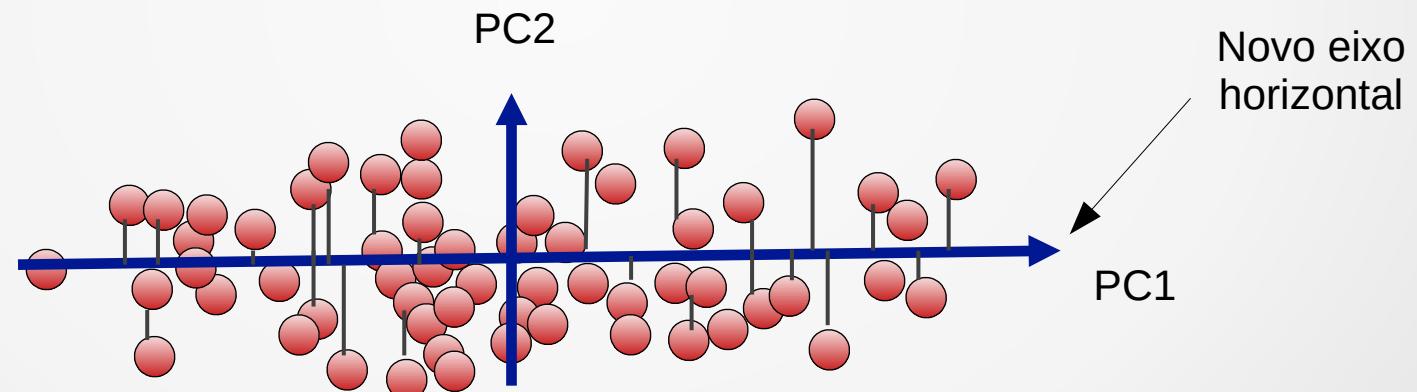
Análise de Componentes Principais

- Aplicando uma reorientação do gráfico dados com uma rotação, podemos ver que a componente principal se torna um eixo horizontal.



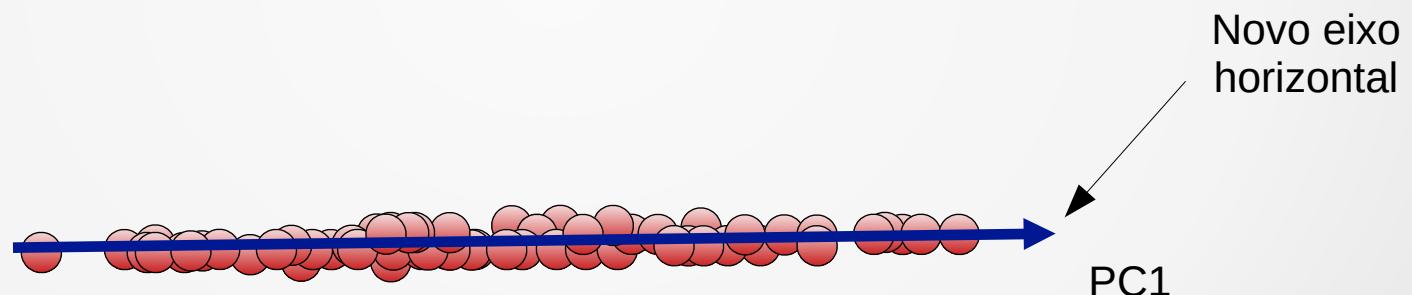
Análise de Componentes Principais

- Os objetos são projetados no eixo correspondente a componente principal



Análise de Componentes Principais

- Os objetos projetados são reduzidos de duas dimensões para uma, baseado na variância máxima.
- De forma similar, PCA é aplicado em conjuntos de dimensionalidade maior, fazendo com que atributos de menor variância sejam reduzidos.



Análise de Componentes Principais

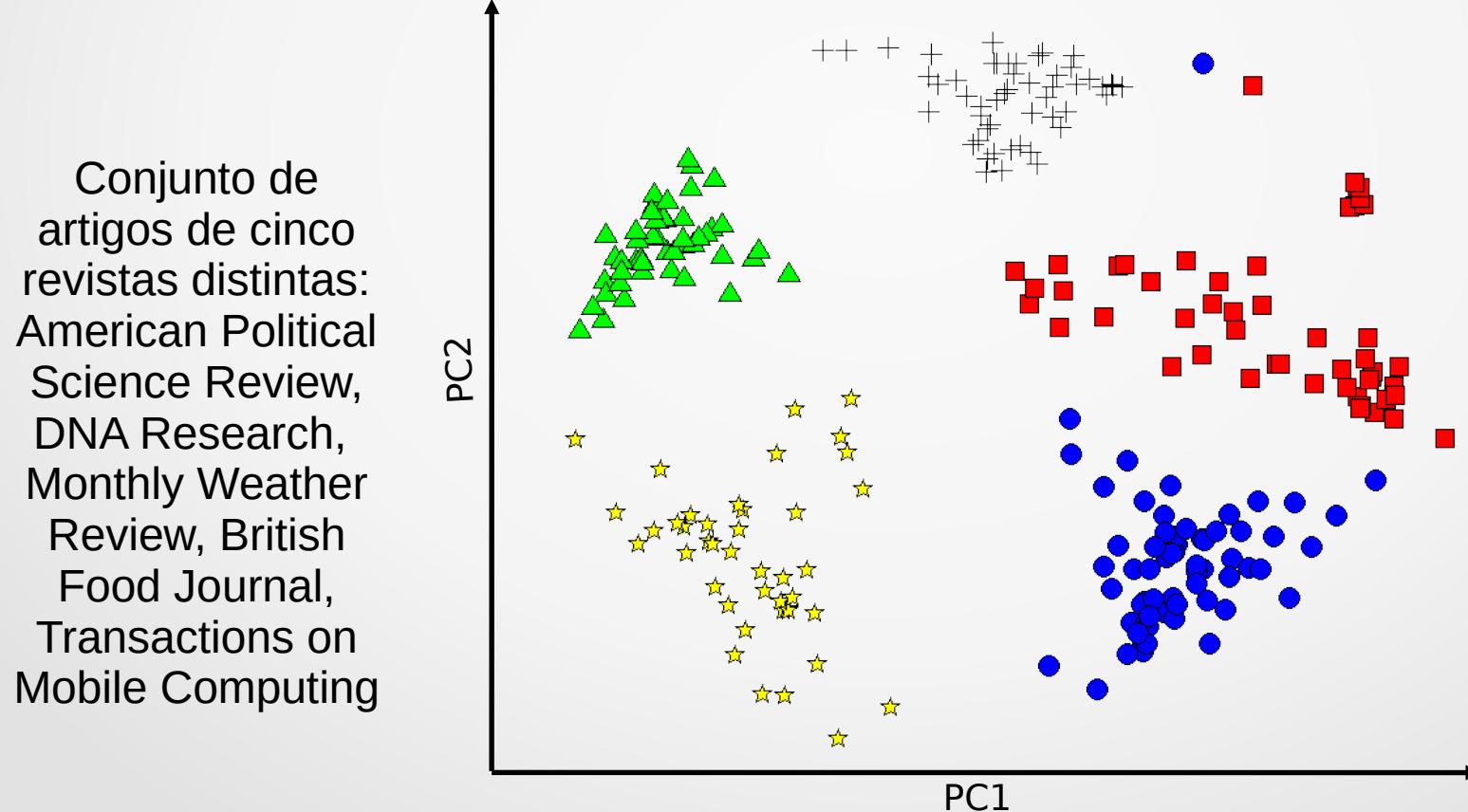
- Existe perda de informação na componente com menor variação
 - Contudo, é mínima possível
 - A melhoria do desempenho do classificador pode compensar
 - É como cinema
 - Muitas vezes, não se tem grande perda de um filme 3D para filmes 2D

Análise de Componentes Principais

- E se os dados tiverem mais de duas dimensões?
 - PC1 será criada na direção que possuir a maior variação
 - PC2 será criada na direção que possuir a segunda maior variação
 - PC3 será criada na direção que possuir a terceira maior variação
 - E assim sucessivamente....
- As componentes de menor variação serão reduzidas.

Exemplo de uso

- No caso dos documentos, a ideia é ir reduzindo as componentes principais até que sobre apenas duas!



Formalização

- Dado um conjunto de dados (\mathbf{X}) com n objetos de p dimensões, a transformação é definida por um conjunto de vetores de pesos ou coeficientes $\mathbf{w}_{(k)} = (w_1, \dots, w_p)_{(k)}$ que mapeiam cada objeto $\mathbf{x}_{(i)}$ de \mathbf{X} em um novo vetor de componentes principais $\mathbf{t}()=(t_1, \dots, t_l)_{(i)}$, sendo:

$$t_{k(i)} = \mathbf{x}_{(i)} \cdot \mathbf{w}_{(k)}$$

para todo $i=1, \dots, n$ e $k=1, \dots, l$, de forma que \mathbf{t} herda a variância máxima de \mathbf{x} e \mathbf{w} está restrito a ser um vetor unitário.

Primeira componente

- Para maximizar a variância, o primeiro vetor $\mathbf{w}_{(1)}$ de pesos deve satisfazer:

$$\mathbf{w}_{(1)} = \arg \max_{\|\mathbf{w}\|=1} \left\{ \sum_i (t_1)_{(i)}^2 \right\}$$

$$\mathbf{w}_{(1)} = \arg \max_{\|\mathbf{w}\|=1} \left\{ \sum_i (\mathbf{x}_{(i)} \cdot \mathbf{w})^2 \right\}$$

$$\mathbf{w}_{(1)} = \arg \max_{\|\mathbf{w}\|=1} \left\{ \|\mathbf{Xw}\|^2 \right\}$$

$$\mathbf{w}_{(1)} = \arg \max_{\|\mathbf{w}\|=1} \left\{ \mathbf{w}^T \mathbf{X}^T \mathbf{X} \mathbf{w} \right\}$$

Outras componentes

- A k -ésima componente pode ser encontrada subtraindo as primeiras $k-1$ componentes de \mathbf{X} :

$$\hat{\mathbf{X}}_k = \mathbf{X} - \sum_{s=1}^{k-1} \mathbf{X} \mathbf{w}_{(s)} \mathbf{w}_{(s)}^T$$

para depois encontrar o vetor que extrai a maior variância desta nova matriz

$$\mathbf{w}_{(k)} = \arg \max_{\|\mathbf{w}\|=1} \left\{ \|\hat{\mathbf{X}}_k \mathbf{w}\|^2 \right\}$$

- Os vetores resultantes são autovalores da matriz \mathbf{X}

Decomposição

- A k -ésima componente de $\mathbf{x}_{(i)}$ pode ser obtida por:

$$t_{k(i)} = \mathbf{x}_{(i)} \cdot \mathbf{w}(k)$$

- E a principal decomposição completa de X é dada por:

$$\mathbf{T} = \mathbf{XW}$$

onde W é a matriz com os auto-vetores de $\mathbf{X}^T \mathbf{X}$.

Redução da dimensionalidade

- A transformação $\mathbf{T} = \mathbf{X}\mathbf{W}$ mapeia um vetor $\mathbf{x}_{(i)}$ do espaço original de p dimensões para outro espaço de dimensões não correlacionadas sobre o conjunto de dados.
- Contudo, nem todas as componentes principais necessitam ser mantidas
 - Talvez seja interessante manter os L primeiros (e de maior variação) componentes
- Assim temos uma nova transformação

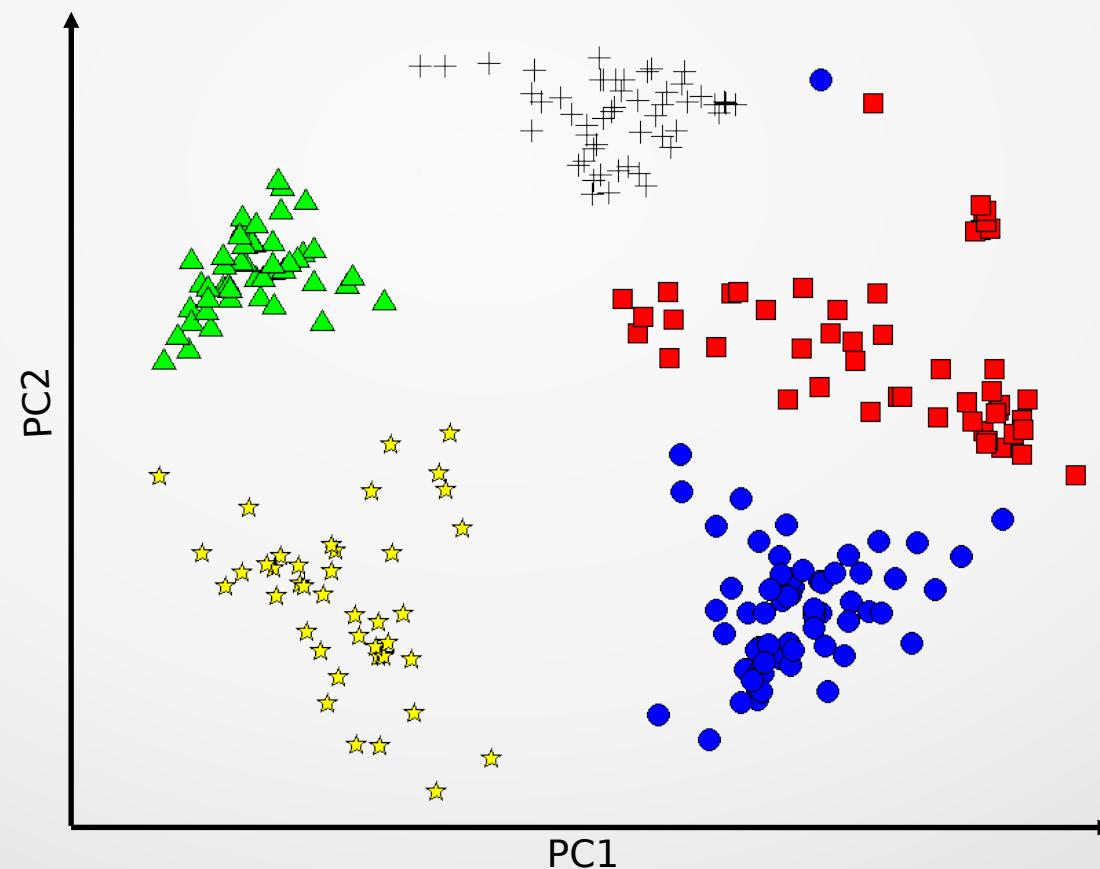
$$\mathbf{T}_L = \mathbf{X}\mathbf{W}_L$$

onde \mathbf{T}_L tem n linhas mas apenas L colunas.

Exemplo de uso

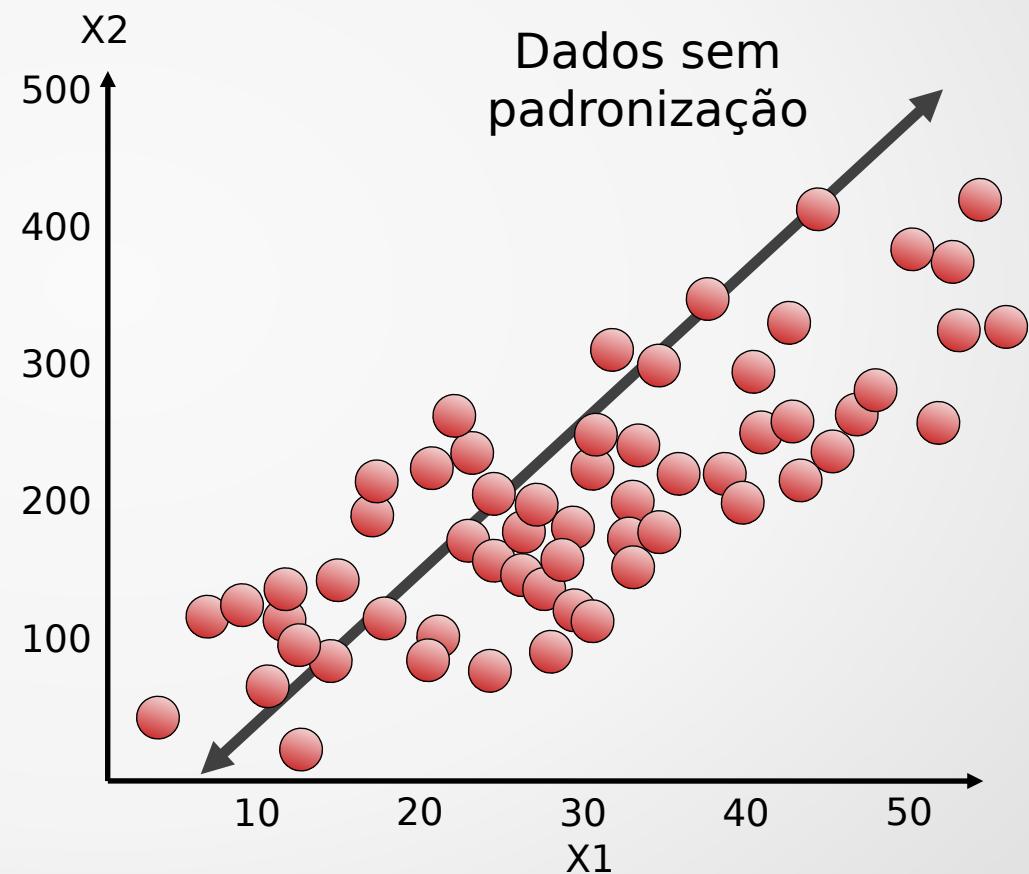
- Ou seja, os dados que estamos vendo neste gráfico são T_2 , ou seja, $T_2 = XW_2$

Conjunto de artigos de cinco revistas distintas:
American Political Science Review,
DNA Research,
Monthly Weather Review, British Food Journal,
Transactions on Mobile Computing



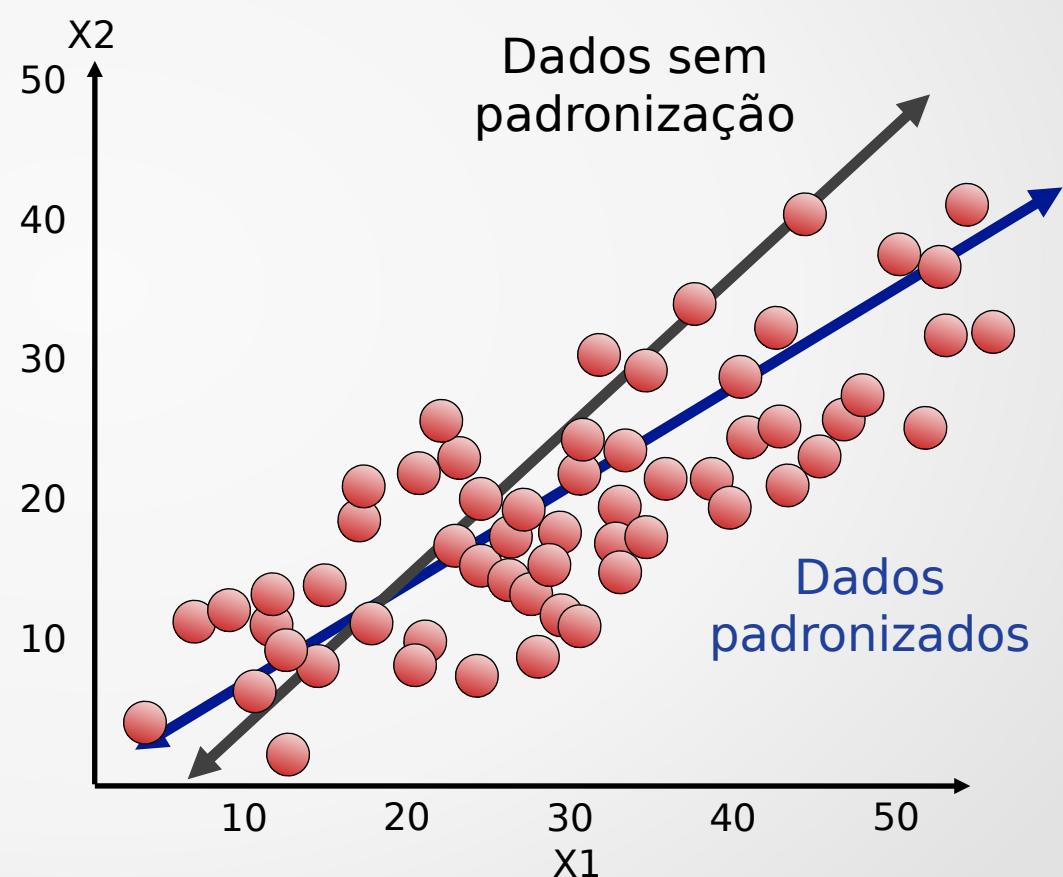
Importância de padronização

- Métodos **sensíveis** à escala relativa das variáveis originais!
 - Pois buscam a maior variância



Importância de padronização

- Métodos **sensíveis à escala** relativa das variáveis originais!
 - Pois buscam a maior variância
 - Uma padronização prévia pode melhorar seus resultados!



Aplicando PCA

- Exemplo de aplicação de PCA no conjunto Iris
- Código

```
from sklearn.decomposition import PCA  
#Cria uma instância da classe PCA
```

```
PCAinst = PCA(n_components=2, whiten=True)
```

```
#Ajusta a instância aos dados e o transforma  
trans = PCAinst.fit_transform(data.iloc[:, :4])
```

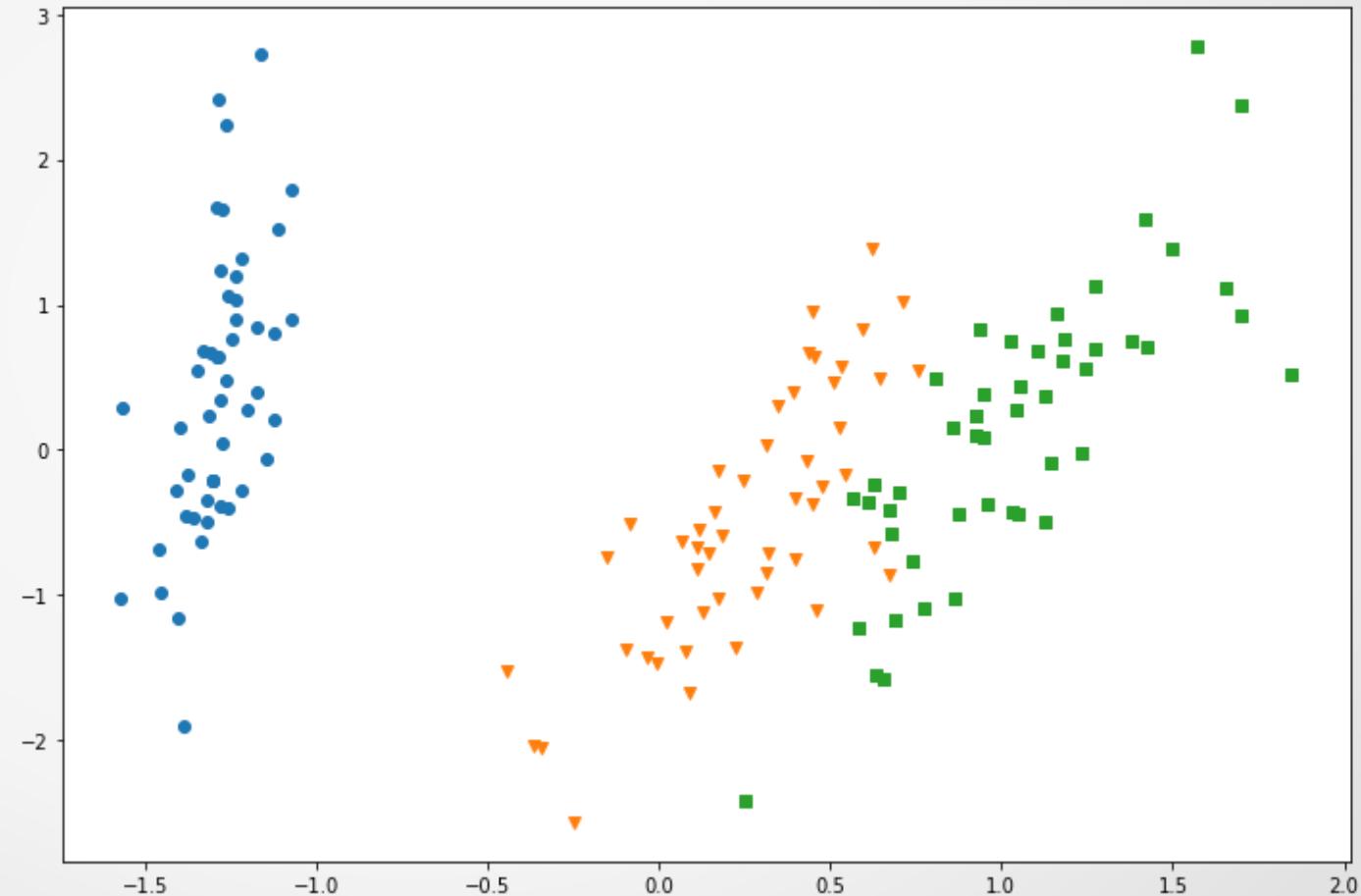
```
#Configura o dataframe resultante  
data_trans = pd.DataFrame(data=trans)  
data_trans.columns = ['PC1', 'PC2']  
data_trans['species'] = data['species']
```

whiten =
reescalar e
centralizar os
dados

Aplicando PCA

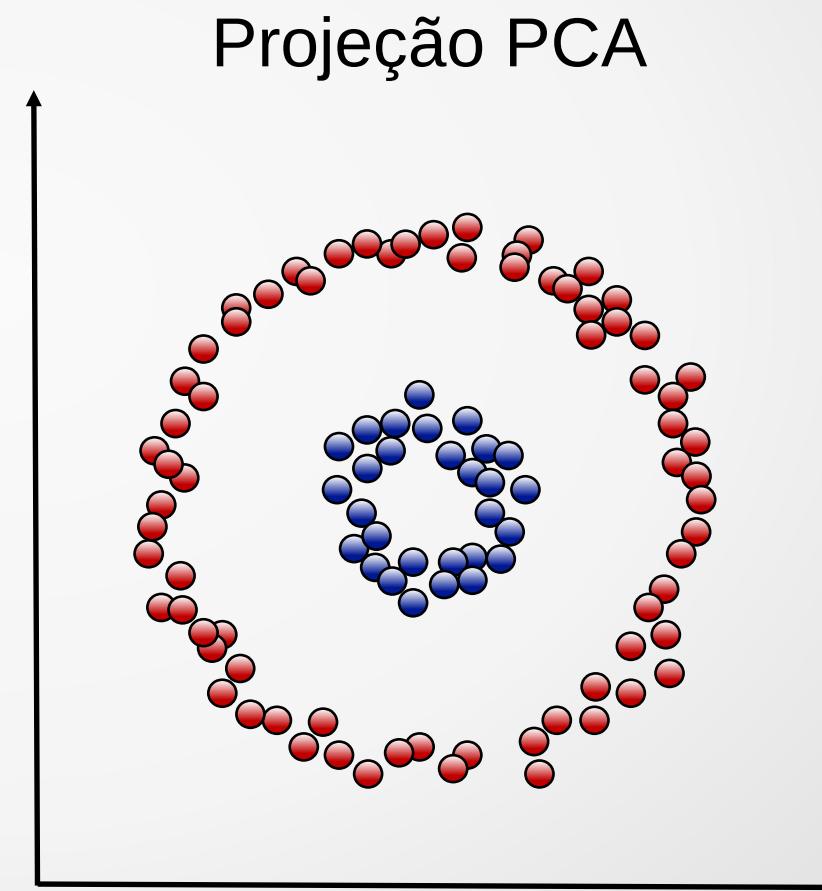
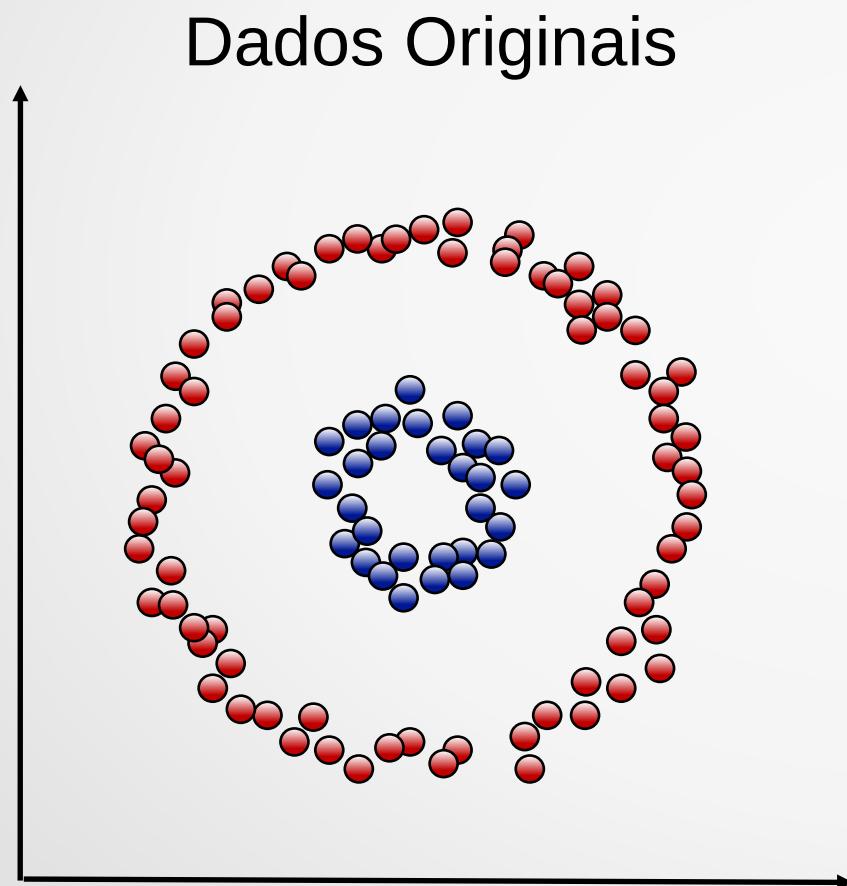
- Código

```
import  
matplotlib.pyplot  
as plt  
plt.plot(data_trans.PC1[0:50],  
data_trans.PC2[0:  
50] ,ls=' ',  
marker='o')  
plt.plot(data_trans.PC1[50:100],  
data_trans.PC2[50  
:100] ,ls=' ',  
marker='v')  
plt.plot(data_trans.PC1[100:150],  
data_trans.PC2[10  
0:150] ,ls=' ',  
marker='s')
```



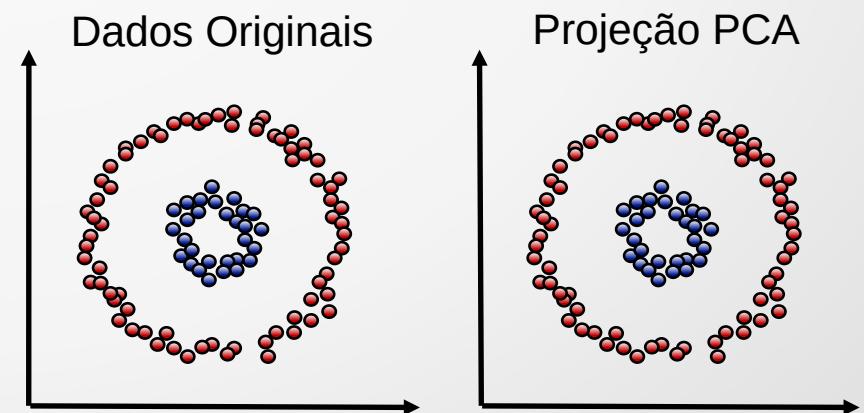
Tratando problemas não lineares

- Considere os dados a seguir



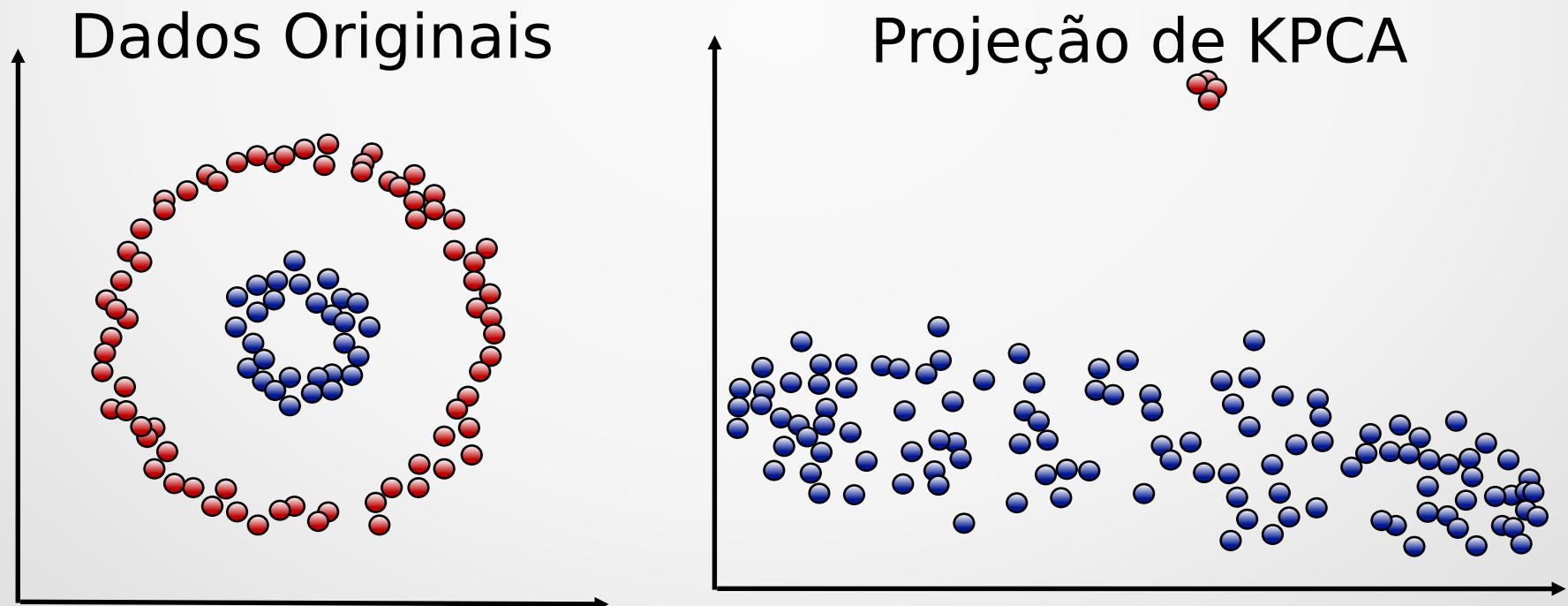
Tratando problemas não lineares

- Transformações feitas com PCA/SVD são lineares
 - Contudo, os dados podem ter atributos não lineares
 - Isso pode ser prejudicial a redução de dimensionalidade



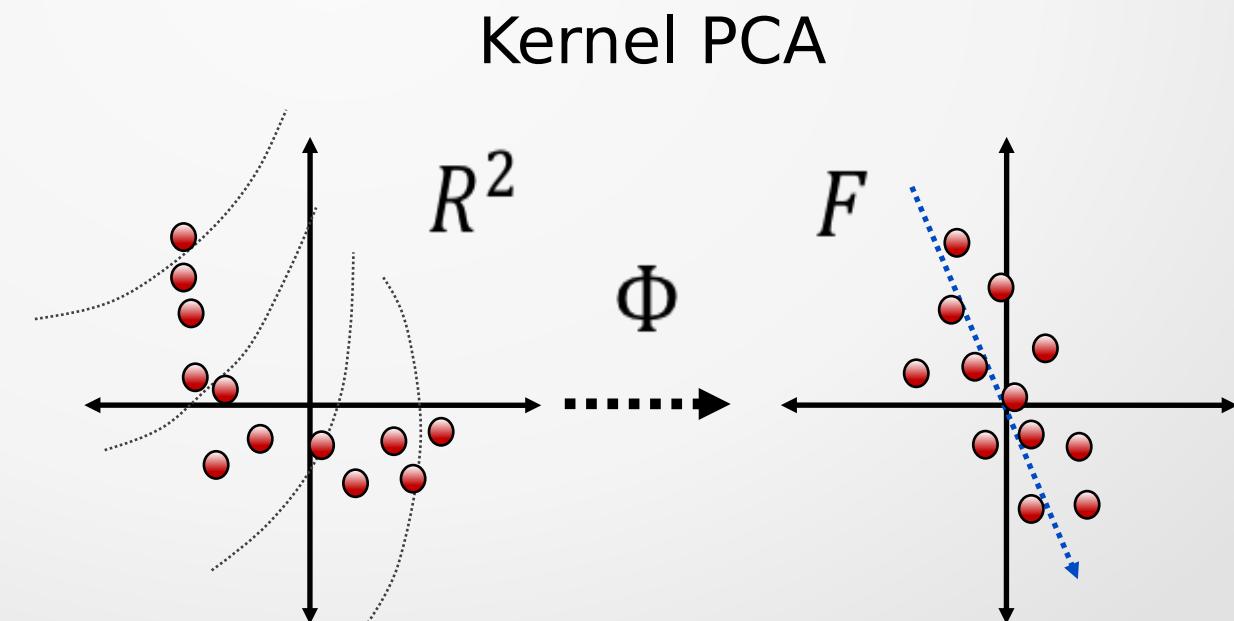
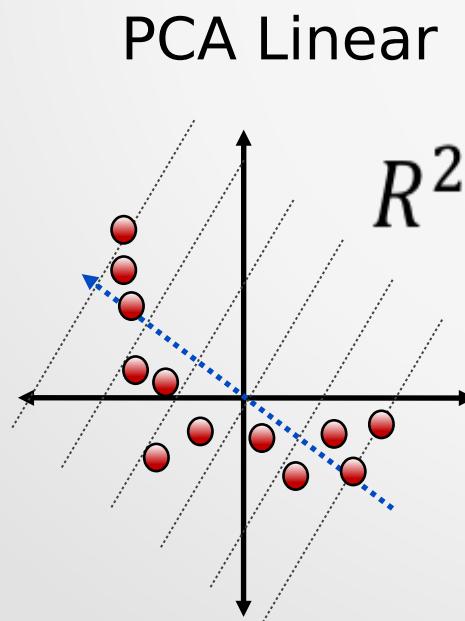
Kernel PCA

- Possível solução são *kernels* para PCA não linear
 - Funções *kernel* usam produto interno implícito, isso é, entre as imagens de todos os pares de dados no espaço de atributos (chamado truque de *kernel*).



Kernel PCA

- Com o truque de kernel é possível mapear os atributos para uma espaço de características mais adequado
 - O que permite uma melhor aplicação do PCA
 - Veremos mais detalhes nas aulas de SVM!



Aplicando Kernel PCA

- Exemplo de uso do Kernel PCA

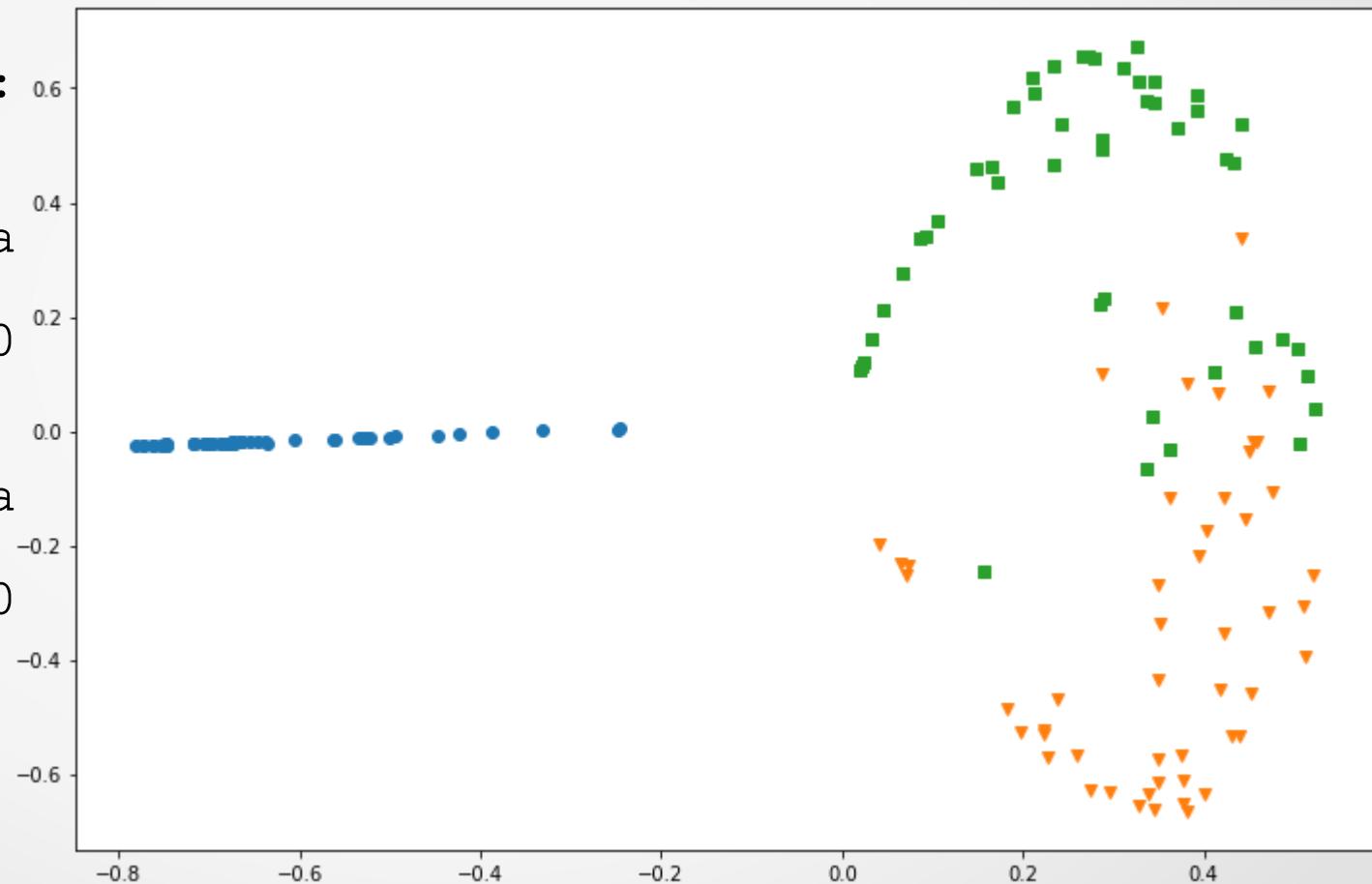
- Código

```
from sklearn.decomposition import KernelPCA  
#Cria uma instância da classe KernelPCA  
kPCA = KernelPCA(n_components=2, kernel='rbf',  
gamma=1.0)  
#Ajusta a instância aos dados e o transforma  
trans = kPCA.fit_transform(data.iloc[:, :4])  
#Configura o dataframe resultante  
data_trans = pd.DataFrame(data=trans)  
data_trans.columns = ['PC1', 'PC2']  
data_trans['species'] = data['species']
```

Aplicando Kernel PCA

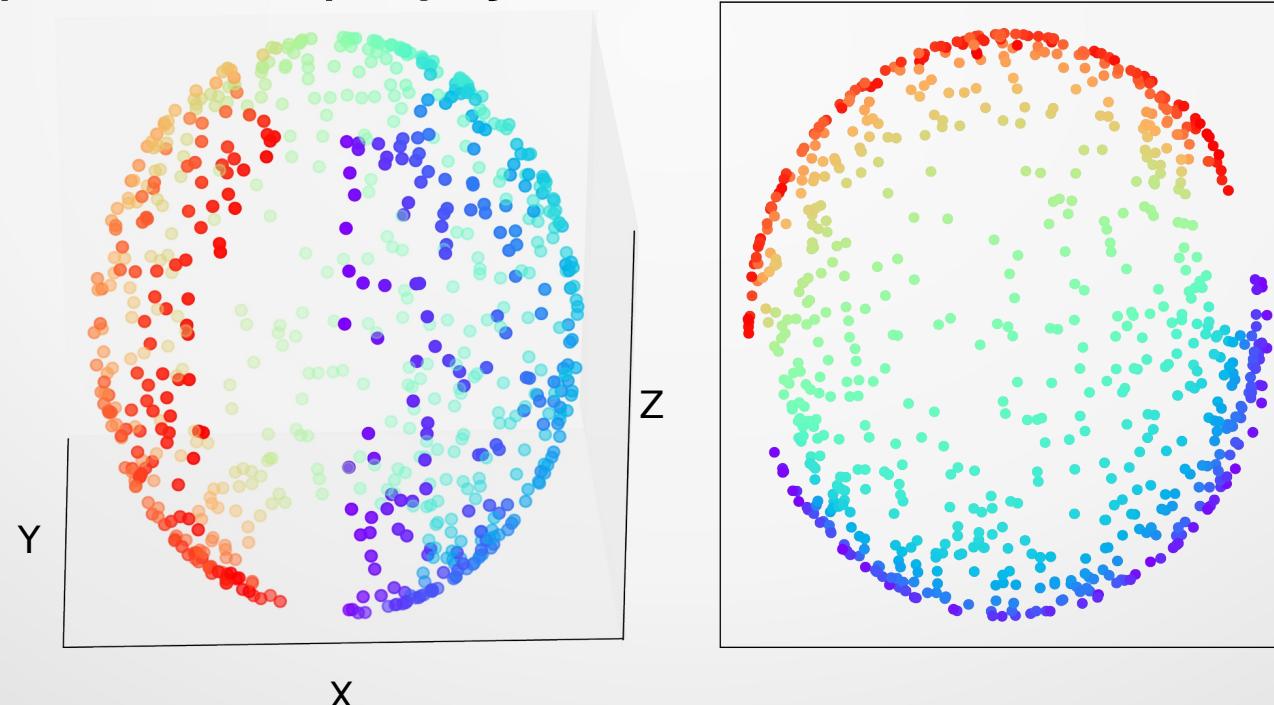
- Código

```
plt.plot(data_trans.PC1[0:50],  
         data_trans.PC2[0:50], ls='',  
         marker='o')  
plt.plot(data_trans.PC1[50:100],  
         data_trans.PC2[50:100], ls='',  
         marker='v')  
plt.plot(data_trans.PC1[100:150],  
         data_trans.PC2[100:150], ls='',  
         marker='s')
```



Multi-Dimensional Scaling (MDS)

- Consiste em uma transformação não linear
 - Que não foca na manutenção da variância média
 - Ao invés disso, mantém a distância geométrica entre os pontos na projeção



Aplicando MDS

- Exemplo de uso MDS

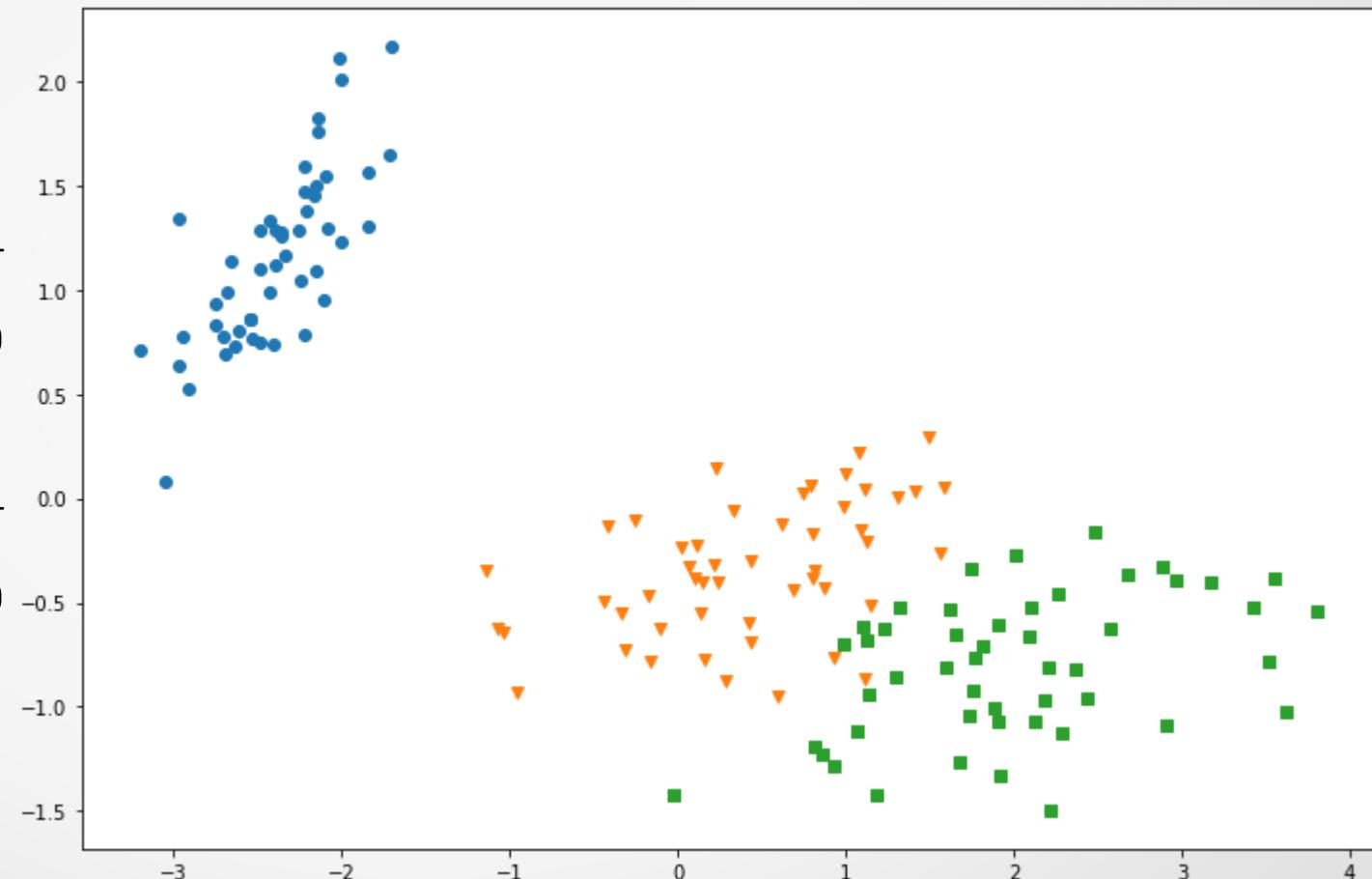
- Código

```
from sklearn.manifold import MDS
#Cria uma instância da classe MDS
mdsMod = MDS(n_components=2)
#Ajusta a instância aos dados e o transforma
trans = mdsMod.fit_transform(data.iloc[:, :4])
#Configura o dataframe resultante
data_trans = pd.DataFrame(data=trans)
data_trans.columns = ['PC1', 'PC2']
data_trans['species'] = data['species']
```

Aplicando MDS

- Código

```
plt.plot(data_trans.PC1[0:50],  
         data_trans.PC2[0:50], ls='',  
         marker='o')  
plt.plot(data_trans.PC1[50:100],  
         data_trans.PC2[50:100], ls='',  
         marker='v')  
plt.plot(data_trans.PC1[100:150],  
         data_trans.PC2[100:150], ls='',  
         marker='s')
```



Exemplos de aplicação

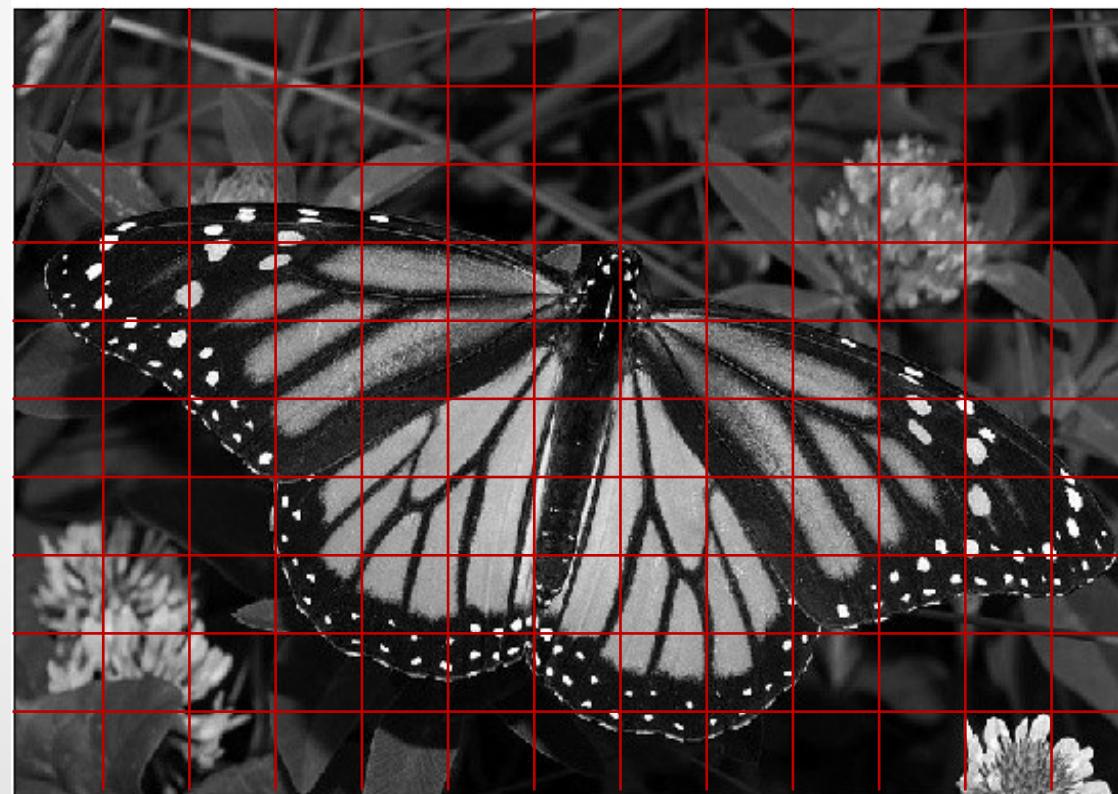
- Além de visualização, a redução do número de atributos também benéfica para tarefas de aprendizado de dados
 - Processamento de documentos
 - Conjuntos de dados de imagens



https://commons.wikimedia.org/wiki/File:Monarch_In_May.jpg

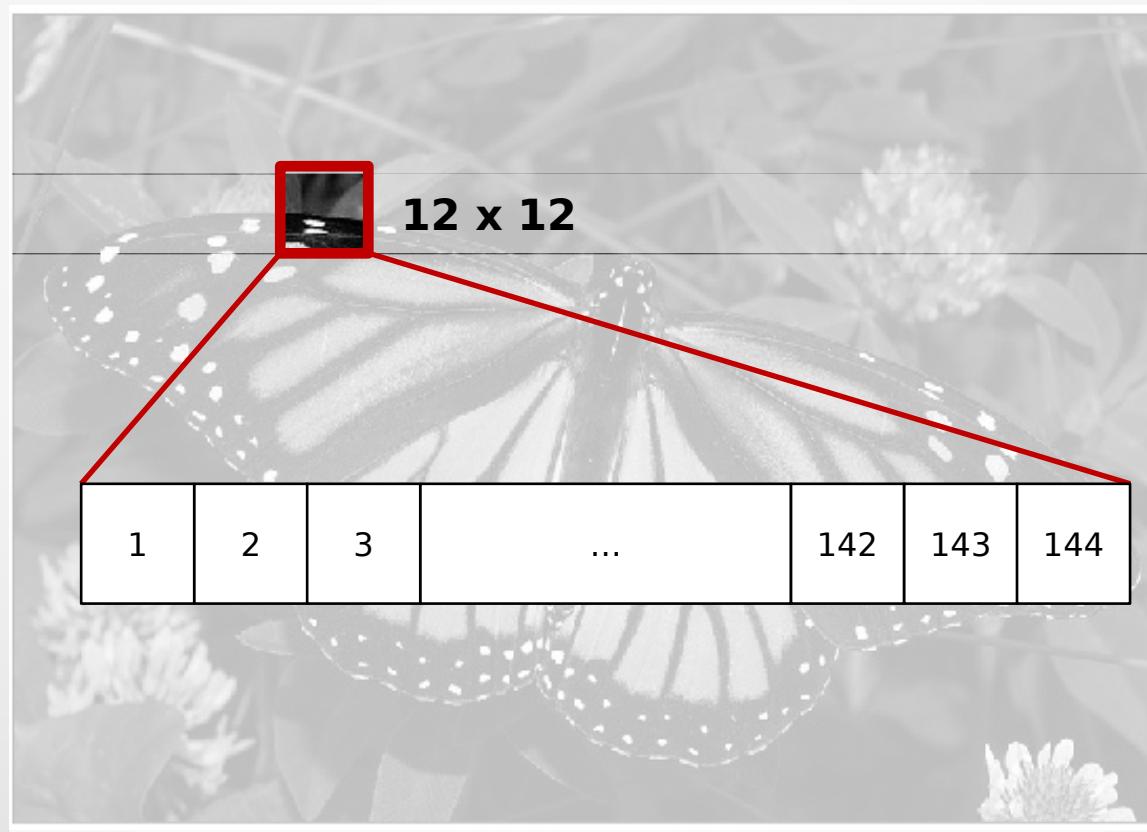
Exemplos de aplicação

- Considerando a imagem
 - Por exemplo: podemos dividir imagem em quadrados com 12x12 pixels cada



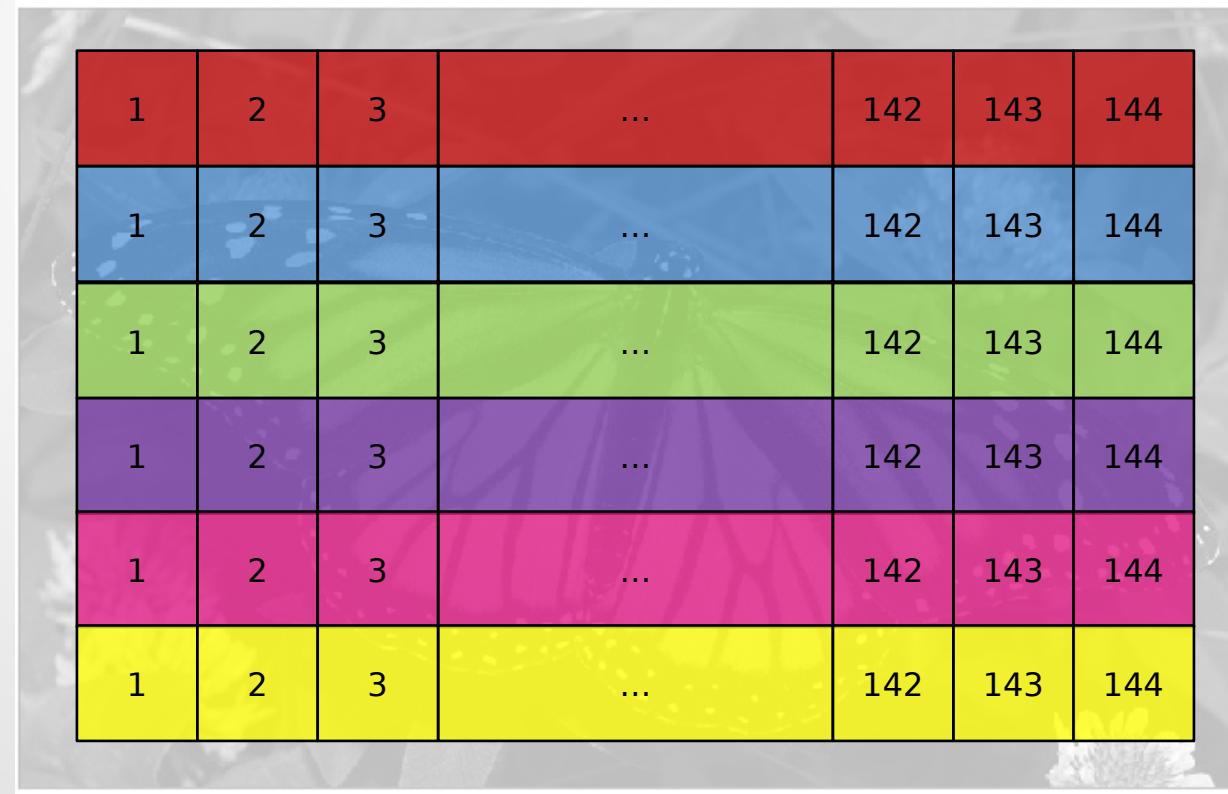
Exemplos de aplicação

- Dividir imagem em quadrados de 12x12 pixels
- Transforma em um vetor de 144 posições



Exemplos de aplicação

- Dividir imagem em quadrados de 12x12 pixels
- Transforma em um vetor de 144 posições
- Aplica PCA sobre os dados



1	2	3	...	142	143	144
1	2	3	...	142	143	144
1	2	3	...	142	143	144
1	2	3	...	142	143	144
1	2	3	...	142	143	144
1	2	3	...	142	143	144

Exemplos de aplicação

- Compressão de imagem de 144 para 60 dimensões

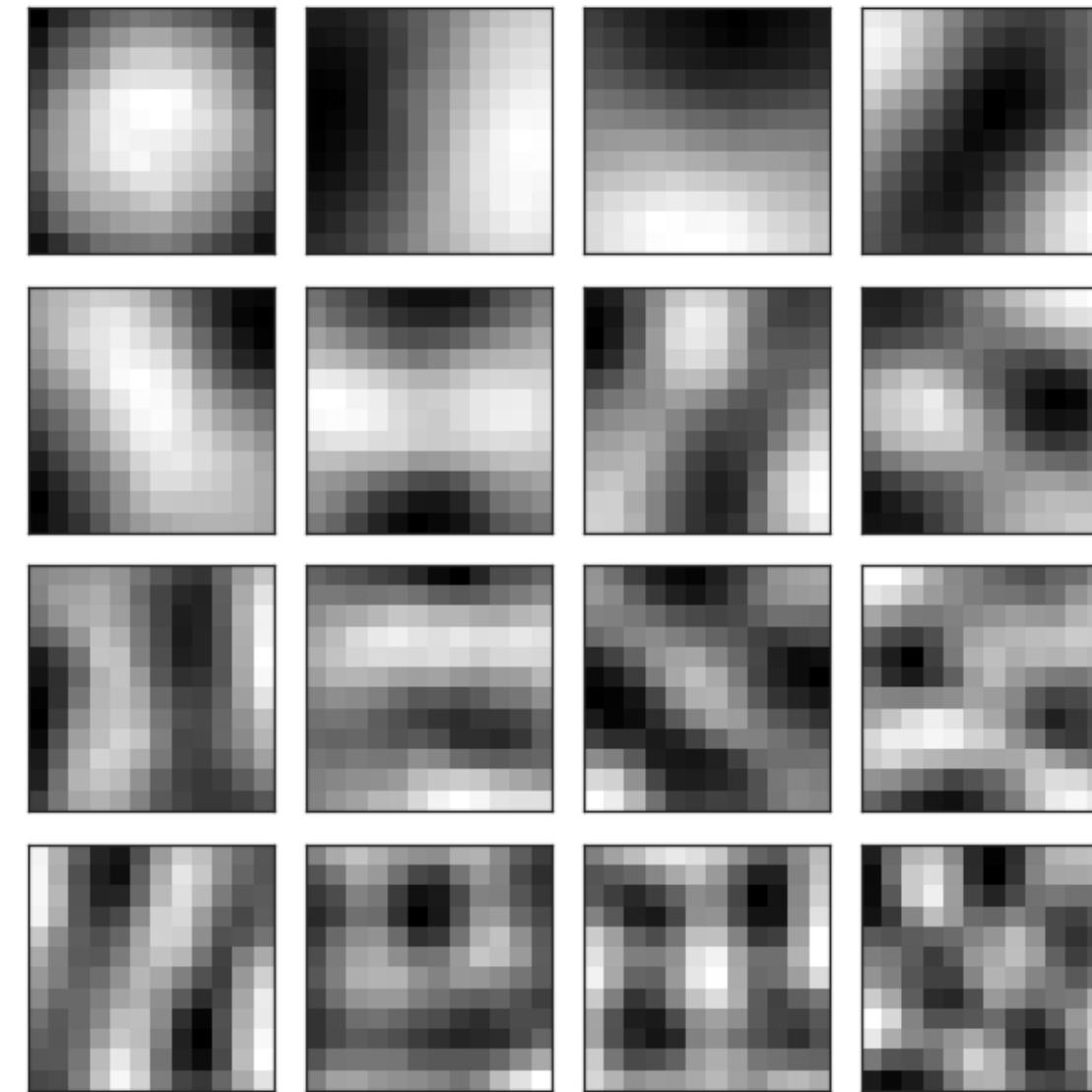


Exemplos de aplicação

- Compressão de imagem de 144 para 16 dimensões



16 Autovalores Principais



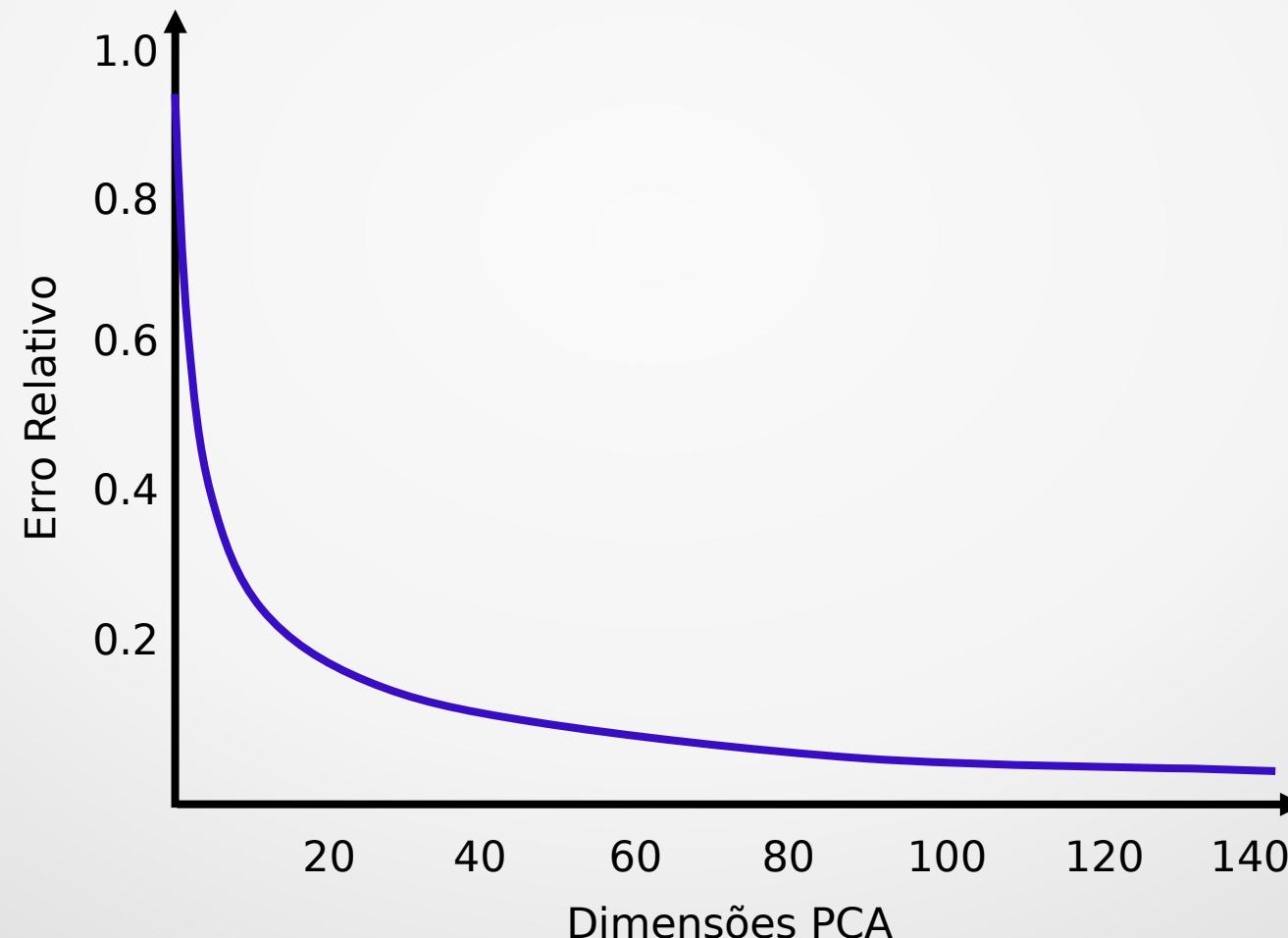
Exemplos de aplicação

- Compressão de imagem de 144 para 4 dimensões



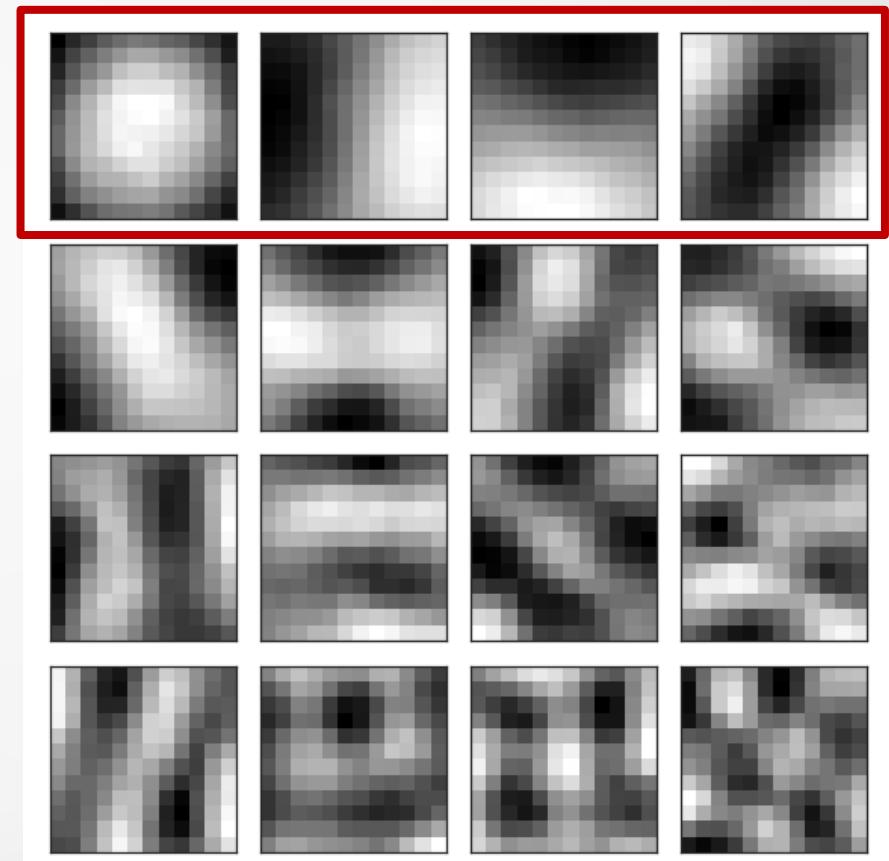
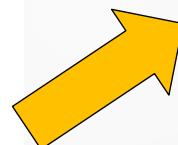
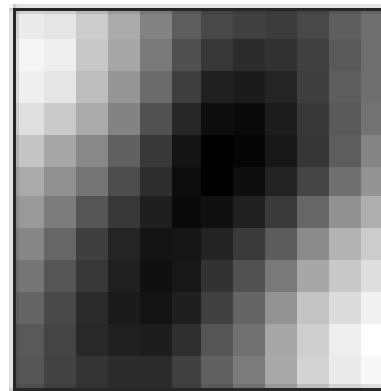
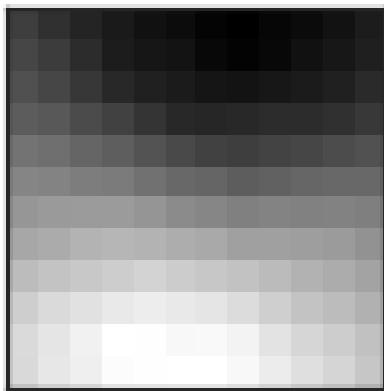
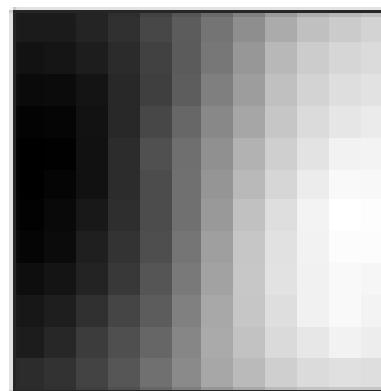
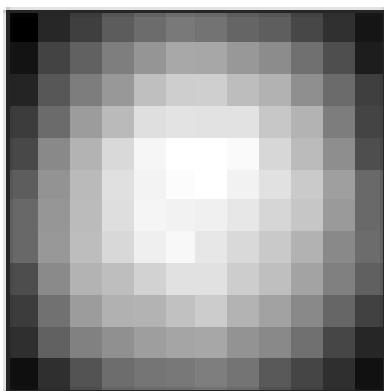
Deterioração do modelo

- Podemos ver como o erro L2 se comporta com a redução do número de dimensões



Análise dos autovalores

- Quatro autovalores principais



Exemplos de aplicação

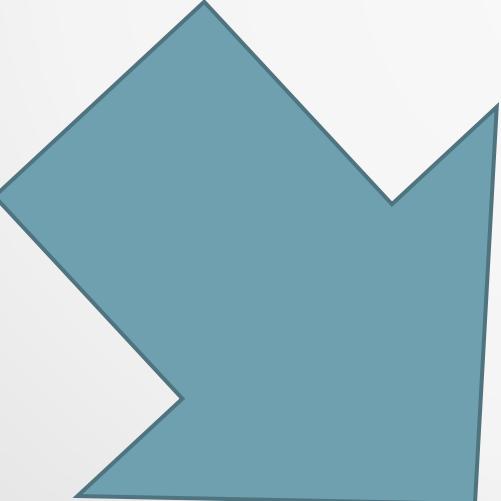
- Compressão de imagem de 144 para 4 dimensões



Técnicas de seleção de atributos

- Vimos que uma das alternativas para redução de dimensionalidade consiste na redução
 - Onde algumas dimensões são reduzidas em outras
- Outro tipo de solução para alta dimensionalidade é a seleção de atributos
 - Objetivo é eliminar atributos irrelevantes ou redundantes

ID	Proprietário	E. Civil	Renda	Investe
1	Sim	Solteiro	1.500,00	Pouco
2	Não	Casado	812,00	Muito
3	Não	Solteiro	2.345,67	Não
4	Sim	Casado	4.768,00	Muito
5	Não	Divorciado	734,00	Não
6	Não	Casado	3.900,00	Pouco
7	Sim	Divorciado	2.100,00	Muito



ID	Proprietário	Renda	Investe
1	Sim	1.500,00	Pouco
2	Não	812,00	Muito
3	Não	2.345,67	Não
4	Sim	4.768,00	Muito
5	Não	734,00	Não
6	Não	3.900,00	Pouco
7	Sim	2.100,00	Muito

Abordagens de seleção

- **Internas**: a seleção ocorre naturalmente como parte do algoritmo de aprendizado de dados
 - Alguns algoritmos selecionam automaticamente quais atributos são mais relevantes para a tarefa de AM
 - Exemplo: árvores de decisão e regras selecionam os atributos que com maior ganho de informação ou menor erro

Abordagens de seleção

- **Filtro**: atributos são selecionados antes da aplicação da técnica de aprendizado
 - Também conhecidos como serviços de análise
 - Por exemplo, para um determinado atributo binário, filtrar somente as tuplas que possem um dos valores e remover o atributo
 - Processo pode ser repetido para os valores complementares

Abordagens de seleção

- **Envoltório (wrapper)**: utilizam o algoritmo de aprendizado de máquina para avaliar quais atributos devem ser selecionados
 - Nesse caso, um conjunto de atributos é selecionado e a técnica de AM aplicada
 - O desempenho da técnica é estimado
 - Processo é repetido para outros conjuntos de atributos
 - Seleção é feita a partir dos desempenhos

Seleção de atributos recursiva

- Considerando os métodos discutidos, a seleção de atributos pode ocorrer de forma recursiva
 - O melhor/pior atributo pode ser escolhido utilizando alguma medida e retirado
 - O processo se repete até que seja selecionados/sobrem os L melhores atributos
 - Entre uma iteração e outra, um modelo dos dados pode ser gerado
 - Ou qualquer outro tipo de avaliação

Arquitetura de filtro com envoltório

