Universidade Federal do Paraná

Laboratório de Estatística e Geoinformação - LEG

Manipulando textos e imagens

Prof.: Eduardo Vargas Ferreira

Introdução

- Em machine learning são comuns as aplicações em que **x** representa objetos não convencionais, como imagens e documentos de texto.

- Entretanto, computadores não entendem tais documentos diretamente como nós;

- Ou seja, utilizamos conhecimentos "não lógicos" para reconhecer imagens e textos. Por exemplo:

  - ⋆ Quebramos automaticamente sentenças em unidades de significado;

  - ⋆ Reconhecemos padrões em imagens com certa facilidade.

Introdução

- Em machine learning são comuns as aplicações em que **x** representa objetos não convencionais, como imagens e documentos de texto.

- Entretanto, computadores não entendem tais documentos diretamente como nós;

- Ou seja, utilizamos conhecimentos "não lógicos" para reconhecer imagens e textos. Por exemplo:

  - ⋆ Quebramos automaticamente sentenças em unidades de significado;

  - ⋆ Reconhecemos padrões em imagens com certa facilidade.

Introdução

- Em machine learning são comuns as aplicações em que **x** representa objetos não convencionais, como imagens e documentos de texto.
- Entretanto, computadores não entendem tais documentos diretamente como nós;
- Ou seja, utilizamos conhecimentos "não lógicos" para reconhecer imagens e textos. Por exemplo:
  - ⋆ Quebramos automaticamente sentenças em unidades de significado;
  - ⋆ Reconhecemos padrões em imagens com certa facilidade.

Introdução

- Em machine learning são comuns as aplicações em que **x** representa objetos não convencionais, como imagens e documentos de texto.

- Entretanto, computadores não entendem tais documentos diretamente como nós;

- Ou seja, utilizamos conhecimentos "não lógicos" para reconhecer imagens e textos. Por exemplo:

⋆ Quebramos automaticamente
sentenças em unidades de significado;

⋆ Reconhecemos padrões em imagens
com certa facilidade.

Introdução

- Em machine learning são comuns as aplicações em que **x** representa objetos não convencionais, como imagens e documentos de texto.

- Entretanto, computadores não entendem tais documentos diretamente como nós;

- Ou seja, utilizamos conhecimentos "não lógicos" para reconhecer imagens e textos. Por exemplo:

  - ⋆ Quebramos automaticamente sentenças em unidades de significado;

  - ⋆ Reconhecemos padrões em imagens com certa facilidade.

Mineração de texto

- Vamos começar falando sobre mineração de texto (*text mining*);
- Sua apresentação (assim como as imagens) são **não estruturados**, ou seja, os dados são desorganizados e difíceis de trabalhar;
  - P. ex., artigos de jornais, *social media*, vídeo, e-mail etc.
- Os **estruturados** seriam dados organizados de forma gerenciável.
  - P. ex., OLAP (*Online Analytical Processing*), XML (*eXtensible Markup Language*) etc.
- Merrill Lynch projeta que em torno de 80-90% de toda informação potencialmente útil está na forma não estruturada;
- Em 2010, Computer World estimaram que a informação não estruturada representa 70-80% dos dados de uma empresa;

Mineração de texto

- Vamos começar falando sobre mineração de texto (*text mining*);
- Sua apresentação (assim como as imagens) são **não estruturados**, ou seja, os dados são desorganizados e difíceis de trabalhar;
  - ⋆ P. ex., artigos de jornais, *social media*, vídeo, e-mail etc.
- Os **estruturados** seriam dados organizados de forma gerenciável.
  - ⋆ P. ex., OLAP (*Online Analytical Processing*), XML (*eXtensible Markup Language*) etc.
- Merrill Lynch projeta que em torno de 80-90% de toda informação potencialmente útil está na forma não estruturada;
- Em 2010, Computer World estimaram que a informação não estruturada representa 70-80% dos dados de uma empresa;

Mineração de texto

- Vamos começar falando sobre mineração de texto (*text mining*);
- Sua apresentação (assim como as imagens) são **não estruturados**, ou seja, os dados são desorganizados e difíceis de trabalhar;
  - ⋆ P. ex., artigos de jornais, *social media*, vídeo, e-mail etc.
- Os **estruturados** seriam dados organizados de forma gerenciável.
  - ⋆ P. ex., OLAP (*Online Analytical Processing*), XML (*eXtensible Markup Language*) etc.
- Merrill Lynch projeta que em torno de 80-90% de toda informação potencialmente útil está na forma não estruturada;
- Em 2010, Computer World estimaram que a informação não estruturada representa 70-80% dos dados de uma empresa;

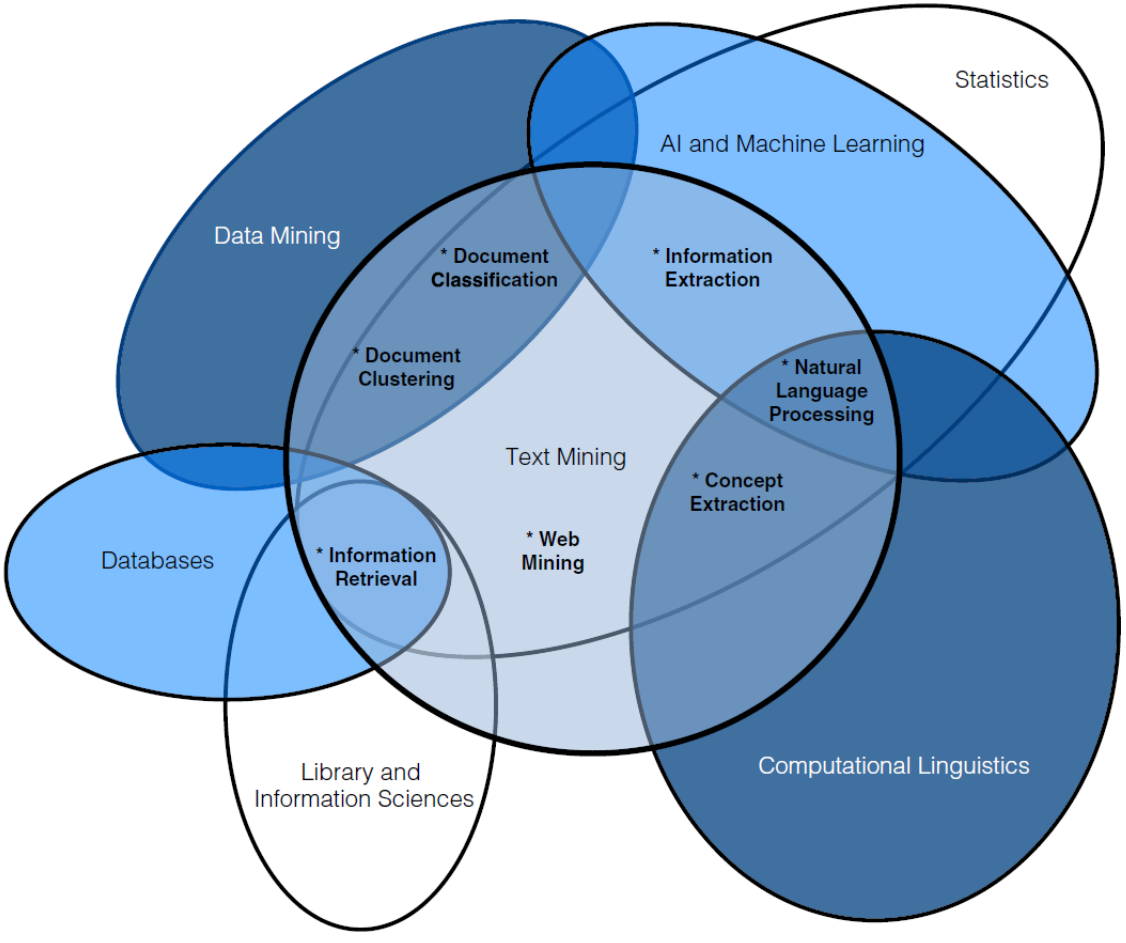
Mineração de texto

- Vamos começar falando sobre mineração de texto (*text mining*);
- Sua apresentação (assim como as imagens) são **não estruturados**, ou seja, os dados são desorganizados e difíceis de trabalhar;
  - ⋆ P. ex., artigos de jornais, *social media*, vídeo, e-mail etc.
- Os **estruturados** seriam dados organizados de forma gerenciável.
  - ⋆ P. ex., OLAP (*Online Analytical Processing*), XML (*eXtensible Markup Language*) etc.
- Merrill Lynch projeta que em torno de 80-90% de toda informação potencialmente útil está na forma não estruturada;
- Em 2010, Computer World estimaram que a informação não estruturada representa 70-80% dos dados de uma empresa;

Mineração de texto

- Vamos começar falando sobre mineração de texto (*text mining*);
- Sua apresentação (assim como as imagens) são **não estruturados**, ou seja, os dados são desorganizados e difíceis de trabalhar;
  - ⋆ P. ex., artigos de jornais, *social media*, vídeo, e-mail etc.
- Os **estruturados** seriam dados organizados de forma gerenciável.
  - ⋆ P. ex., OLAP (*Online Analytical Processing*), XML (*eXtensible Markup Language*) etc.
- Merrill Lynch projeta que em torno de 80-90% de toda informação potencialmente útil está na forma não estruturada;
- Em 2010, Computer World estimaram que a informação não estruturada representa 70-80% dos dados de uma empresa;

Mineração de texto no espaço da TI



Principais áreas na mineração de texto

- Existem pelo menos 5 principais áreas práticas na mineração de texto:
  - ⋆ **Extração de informação:** Identificação e extração de informação de fatores relevantes e relações entre textos não estruturados;
  - ⋆ **Clusterização de documentos:** agrupa e categoriza termos, fragmentos, parágrafos ou documentos utilizando métodos de agrupamentos;
  - ⋆ **Classificação de documentos:** classifica termos, fragmentos, parágrafos ou documentos a partir de exemplos já classificados (dados de treinamento);
  - ⋆ **Mineração web:** minera dados da internet com foco em interconexões da web;
  - ⋆ **Natural language processing (NLP):** se preocupa com a interação entre o computador e a linguagem (natural) humana. P ex., reconhecimento de voz, face etc.

Principais áreas na mineração de texto

- Existem pelo menos 5 principais áreas práticas na mineração de texto:
  - ⋆ **Extração de informação:** Identificação e extração de informação de fatores relevantes e relações entre textos não estruturados;
  - ⋆ **Clusterização de documentos:** agrupa e categoriza termos, fragmentos, parágrafos ou documentos utilizando métodos de agrupamentos;
  - ⋆ **Classificação de documentos:** classifica termos, fragmentos, parágrafos ou documentos a partir de exemplos já classificados (dados de treinamento);
  - ⋆ **Mineração web:** minera dados da internet com foco em interconexões da web;
  - ⋆ **Natural language processing (NLP):** se preocupa com a interação entre o computador e a linguagem (natural) humana. P ex., reconhecimento de voz, face etc.

Principais áreas na mineração de texto

- Existem pelo menos 5 principais áreas práticas na mineração de texto:
  - ⋆ **Extração de informação:** Identificação e extração de informação de fatores relevantes e relações entre textos não estruturados;
  - ⋆ **Clusterização de documentos:** agrupa e categoriza termos, fragmentos, parágrafos ou documentos utilizando métodos de agrupamentos;
  - ⋆ **Classificação de documentos:** classifica termos, fragmentos, parágrafos ou documentos a partir de exemplos já classificados (dados de treinamento);
  - ⋆ **Mineração web:** minera dados da internet com foco em interconexões da web;
  - ⋆ **Natural language processing (NLP):** se preocupa com a interação entre o computador e a linguagem (natural) humana. P ex., reconhecimento de voz, face etc.

Principais áreas na mineração de texto

- Existem pelo menos 5 principais áreas práticas na mineração de texto:
  - ⋆ **Extração de informação:** Identificação e extração de informação de fatores relevantes e relações entre textos não estruturados;
  - ⋆ **Clusterização de documentos:** agrupa e categoriza termos, fragmentos, parágrafos ou documentos utilizando métodos de agrupamentos;
  - ⋆ **Classificação de documentos:** classifica termos, fragmentos, parágrafos ou documentos a partir de exemplos já classificados (dados de treinamento);
  - ⋆ **Mineração web:** minera dados da internet com foco em interconexões da web;
  - ⋆ **Natural language processing (NLP):** se preocupa com a interação entre o computador e a linguagem (natural) humana. P ex., reconhecimento de voz, face etc.

Principais áreas na mineração de texto

- Existem pelo menos 5 principais áreas práticas na mineração de texto:
  - ⋆ **Extração de informação:** Identificação e extração de informação de fatores relevantes e relações entre textos não estruturados;
  - ⋆ **Clusterização de documentos:** agrupa e categoriza termos, fragmentos, parágrafos ou documentos utilizando métodos de agrupamentos;
  - ⋆ **Classificação de documentos:** classifica termos, fragmentos, parágrafos ou documentos a partir de exemplos já classificados (dados de treinamento);
  - ⋆ **Mineração web:** minera dados da internet com foco em interconexões da web;
  - ⋆ **Natural language processing (NLP):** se preocupa com a interação entre o computador e a linguagem (natural) humana. P ex., reconhecimento de voz, face etc.

Principais áreas na mineração de texto

- Existem pelo menos 5 principais áreas práticas na mineração de texto:
  - ⋆ **Extração de informação:** Identificação e extração de informação de fatores relevantes e relações entre textos não estruturados;
  - ⋆ **Clusterização de documentos:** agrupa e categoriza termos, fragmentos, parágrafos ou documentos utilizando métodos de agrupamentos;
  - ⋆ **Classificação de documentos:** classifica termos, fragmentos, parágrafos ou documentos a partir de exemplos já classificados (dados de treinamento);
  - ⋆ **Mineração web:** minera dados da internet com foco em interconexões da web;
  - ⋆ **Natural language processing (NLP):** se preocupa com a interação entre o computador e a linguagem (natural) humana. P ex., reconhecimento de voz, face etc.

Técnicas para estruturar os dados: tokenização

- Para transformar dados não estruturados em estruturados (na forma numérica) devemos empregar algumas técnicas;

- Este processo deve ser ao mesmo tempo

  (i) Rápido;

  (ii) Informativo.

- Primeiramente, separamos os elementos constituintes do texto, identificando cada palavra através de um processo chamado **tokenização**;

- No R, isto pode ser feito facilmente

```
string2 <- "Olá professor, sou aluna de Estatística"
strsplit(string2, " ")[[1]]
# [1] "Olá" "professor," "sou" "aluna" "de" "Estatística"
```

Técnicas para estruturar os dados: tokenização

- Para transformar dados não estruturados em estruturados (na forma numérica) devemos empregar algumas técnicas;

- Este processo deve ser ao mesmo tempo

  (i) Rápido;

  (ii) Informativo.

- Primeiramente, separamos os elementos constituintes do texto, identificando cada palavra através de um processo chamado **tokenização**;

- No R, isto pode ser feito facilmente

```
string2 <- "Olá professor, sou aluna de Estatística"
strsplit(string2, " ")[[1]]
# [1] "Olá" "professor," "sou" "aluna" "de" "Estatística"
```

Técnicas para estruturar os dados: tokenização

- Para transformar dados não estruturados em estruturados (na forma numérica) devemos empregar algumas técnicas;
- Este processo deve ser ao mesmo tempo
  - (i) Rápido;
  - (ii) Informativo.
- Primeiramente, separamos os elementos constituintes do texto, identificando cada palavra através de um processo chamado **tokenização**;
- No R, isto pode ser feito facilmente

```
string2 <- "Olá professor, sou aluna de Estatística"
strsplit(string2, " ")[[1]]
# [1] "Olá" "professor," "sou" "aluna" "de" "Estatística"
```

Técnicas para estruturar os dados: tokenização

- Para transformar dados não estruturados em estruturados (na forma numérica) devemos empregar algumas técnicas;

- Este processo deve ser ao mesmo tempo

  (i) Rápido;

  (ii) Informativo.

- Primeiramente, separamos os elementos constituintes do texto, identificando cada palavra através de um processo chamado **tokenização**;

- No R, isto pode ser feito facilmente

```
string2 <- "Olá professor, sou aluna de Estatística"
strsplit(string2, " ")[[1]]
# [1] "Olá" "professor," "sou" "aluna" "de" "Estatística"
```

Técnicas para estruturar os dados: stemming

- **Stemming** é a redução de um termo ao seu radical, removendo as desinências, afixos e vogais temáticas;
- Tal abordagem auxilia na filtragem e classificação do documento;
- Por exemplo, considere o conjunto de palavras: {prática, praticada, praticados, praticando, praticante, praticar, praticaram, praticidade};
- Apesar de terem características diferentes preservam o mesmo radical **PRATIC**;
- O processo ocorre em etapas, e em cada uma delas uma decisão.

Técnicas para estruturar os dados: stemming

- **Stemming** é a redução de um termo ao seu radical, removendo as desinências, afixos e vogais temáticas;
- Tal abordagem auxilia na filtragem e classificação do documento;
- Por exemplo, considere o conjunto de palavras: {prática, praticada, praticados, praticando, praticante, praticar, praticaram, praticidade};
- Apesar de terem características diferentes preservam o mesmo radical **PRATIC**;
- O processo ocorre em etapas, e em cada uma delas uma decisão.

Técnicas para estruturar os dados: stemming

- **Stemming** é a redução de um termo ao seu radical, removendo as desinências, afixos e vogais temáticas;
- Tal abordagem auxilia na filtragem e classificação do documento;
- Por exemplo, considere o conjunto de palavras: {prática, praticada, praticados, praticando, praticante, praticar, praticaram, praticidade};
- Apesar de terem características diferentes preservam o mesmo radical **PRATIC**;
- O processo ocorre em etapas, e em cada uma delas uma decisão.

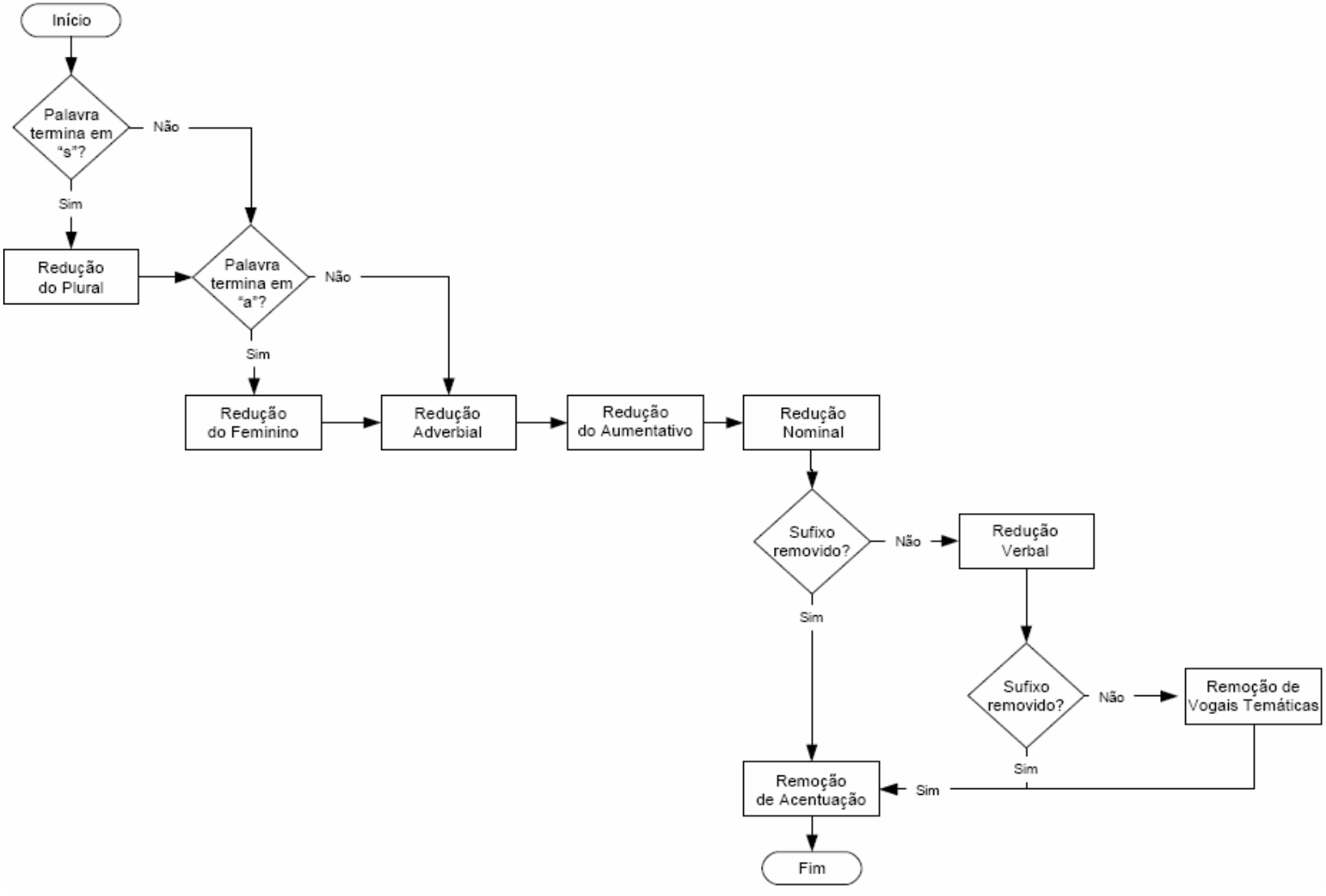
Técnicas para estruturar os dados: stemming

- **Stemming** é a redução de um termo ao seu radical, removendo as desinências, afixos e vogais temáticas;
- Tal abordagem auxilia na filtragem e classificação do documento;
- Por exemplo, considere o conjunto de palavras: {prática, praticada, praticados, praticando, praticante, praticar, praticaram, praticidade};
- Apesar de terem características diferentes preservam o mesmo radical **PRATIC**;
- O processo ocorre em etapas, e em cada uma delas uma decisão.

Técnicas para estruturar os dados: stemming

- **Stemming** é a redução de um termo ao seu radical, removendo as desinências, afixos e vogais temáticas;
- Tal abordagem auxilia na filtragem e classificação do documento;
- Por exemplo, considere o conjunto de palavras: {prática, praticada, praticados, praticando, praticante, praticar, praticaram, praticidade};
- Apesar de terem características diferentes preservam o mesmo radical **PRATIC**;
- O processo ocorre em etapas, e em cada uma delas uma decisão.

Técnicas para estruturar os dados: stemming



Técnicas para estruturar os dados: stopwords

- Outra tarefa na preparação dos dados é a identificação das palavras que podem ser desconsideradas nos passos posteriores da análise;
- Nesta fase, tenta-se retirar tudo que não constitui conhecimento no texto;
- Palavras que são muito comuns muitas vezes não são informativas (e.g., “a”, “esse”, . . .);
- O resultado é uma lista com as palavras a serem descartadas conhecido como **stopwords** ou **stoplist**.

Técnicas para estruturar os dados: stopwords

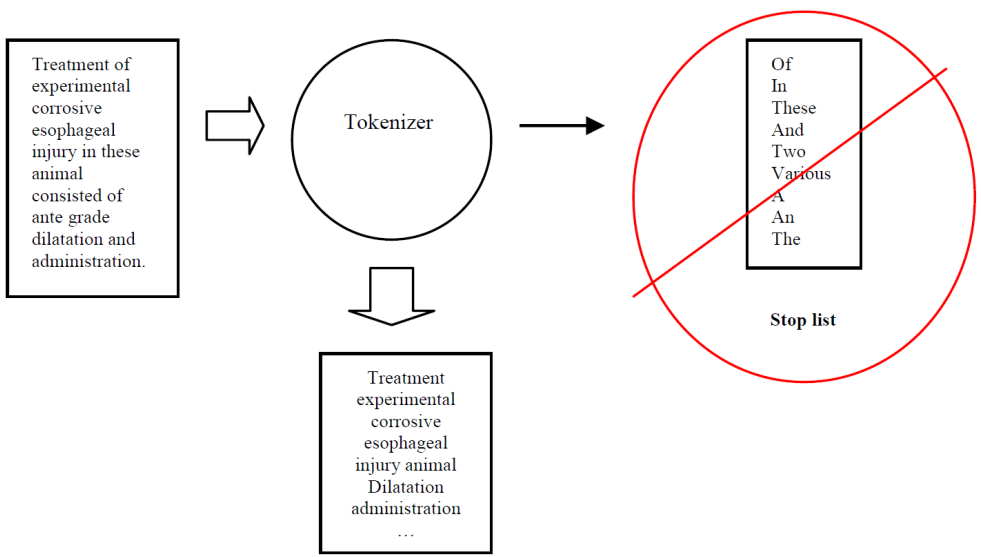
- Outra tarefa na preparação dos dados é a identificação das palavras que podem ser desconsideradas nos passos posteriores da análise;
- Nesta fase, tenta-se retirar tudo que não constitui conhecimento no texto;
- Palavras que são muito comuns muitas vezes não são informativas (e.g., "a", "esse", ...);
- O resultado é uma lista com as palavras a serem descartadas conhecido como **stopwords** ou **stoplist**.

Técnicas para estruturar os dados: stopwords

- Outra tarefa na preparação dos dados é a identificação das palavras que podem ser desconsideradas nos passos posteriores da análise;
- Nesta fase, tenta-se retirar tudo que não constitui conhecimento no texto;
- Palavras que são muito comuns muitas vezes não são informativas (e.g., “a”, “esse”, . . .);
- O resultado é uma lista com as palavras a serem descartadas conhecido como **stopwords** ou **stoplist**.

Técnicas para estruturar os dados: stopwords

- Outra tarefa na preparação dos dados é a identificação das palavras que podem ser desconsideradas nos passos posteriores da análise;
- Nesta fase, tenta-se retirar tudo que não constitui conhecimento no texto;
- Palavras que são muito comuns muitas vezes não são informativas (e.g., "a", "esse", ...);
- O resultado é uma lista com as palavras a serem descartadas conhecido como **stopwords** ou **stoplist**.

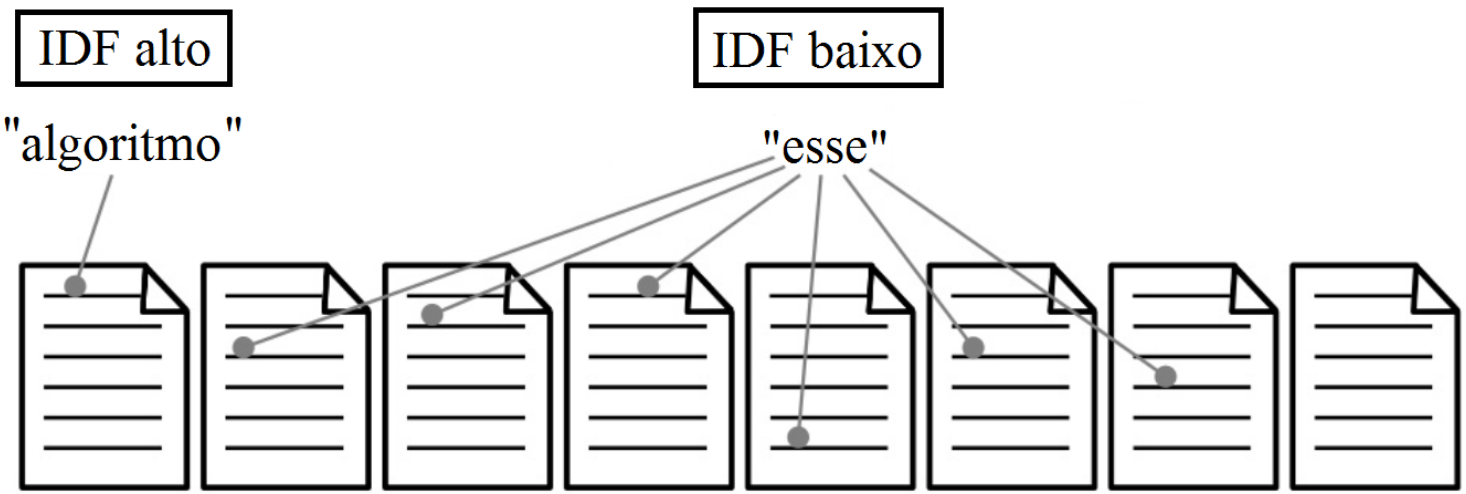


Inverse document frequency (IDF)

- Uma maneira mais eficiente de resolver a questão de palavras comuns é a chamada **Inverse document frequency (IDF)**;
- A ideia é simples, palavras muito comuns (mais frequentes) recebem menores pesos, pois discriminam menos os documentos;

Inverse document frequency (IDF)

- Uma maneira mais eficiente de resolver a questão de palavras comuns é a chamada **Inverse document frequency (IDF)**;
- A ideia é simples, palavras muito comuns (mais frequentes) recebem menores pesos, pois discriminam menos os documentos;



Inverse document frequency (IDF)

- Assim, para o i-ésimo termo do k-ésimo documento temos a seguinte formulação

$$a_{ik} = f_{ik} log \left(\frac{D}{n_i} \right),$$

em que

  - a_{ik} é o peso atribuído ao termo i no documento k;
  - f_{ik} a frequência do termo no documento;
  - D é o número total de documentos;
  - n_i o número de documentos que contém o termo i.

- Obs.: existem outras formas de se ponderar as frequências (e.g. pela raiz quadrada ao invés do logaritmo);

Minimizando distâncias entre vetores

- Queremos minimizar distâncias entre vetores com características similares;
- Considere o exemplo abaixo:

> "Eduardo quando crescer será Estatístico";
>
> "eduardo qundo crecser será estatístico".

- Gostaríamos que nosso algoritmo pontuasse as duas respostas da mesma forma;
- Entretanto, simples tokenização não captará tal semelhança;
- Assim, além de incluirmos apenas entradas minúsculas, devemos corrigir a ortografia (e.g. ▸ Peter Norvig's method).
- Atenção: sempre que fazemos isso perdemos informação!

Minimizando distâncias entre vetores

- Queremos minimizar distâncias entre vetores com características similares;
- Considere o exemplo abaixo:

> "Eduardo quando crescer será Estatístico";
>
> "eduardo qundo crecser será estatístico".

- Gostaríamos que nosso algoritmo pontuasse as duas respostas da mesma forma;
- Entretanto, simples tokenização não captará tal semelhança;
- Assim, além de incluirmos apenas entradas minúsculas, devemos corrigir a ortografia (e.g. ▸ Peter Norvig's method).
- Atenção: sempre que fazemos isso perdemos informação!

Minimizando distâncias entre vetores

- Queremos minimizar distâncias entre vetores com características similares;
- Considere o exemplo abaixo:

> "Eduardo quando crescer será Estatístico";
>
> "eduardo qundo crecser será estatístico".

- Gostaríamos que nosso algoritmo pontuasse as duas respostas da mesma forma;
- Entretanto, simples tokenização não captará tal semelhança;
- Assim, além de incluirmos apenas entradas minúsculas, devemos corrigir a ortografia (e.g. ▸ Peter Norvig's method).
- Atenção: sempre que fazemos isso perdemos informação!

Minimizando distâncias entre vetores

- Queremos minimizar distâncias entre vetores com características similares;
- Considere o exemplo abaixo:

> "Eduardo quando crescer será Estatístico";
>
> "eduardo qundo crecser será estatístico".

- Gostaríamos que nosso algoritmo pontuasse as duas respostas da mesma forma;
- Entretanto, simples tokenização não captará tal semelhança;
- Assim, além de incluirmos apenas entradas minúsculas, devemos corrigir a ortografia (e.g. ▸ Peter Norvig's method).
- Atenção: sempre que fazemos isso perdemos informação!

Minimizando distâncias entre vetores

- Queremos minimizar distâncias entre vetores com características similares;
- Considere o exemplo abaixo:

> "Eduardo quando crescer será Estatístico";
>
> "eduardo qundo crecser será estatístico".

- Gostaríamos que nosso algoritmo pontuasse as duas respostas da mesma forma;
- Entretanto, simples tokenização não captará tal semelhança;
- Assim, além de incluirmos apenas entradas minúsculas, devemos corrigir a ortografia (e.g. ▸ Peter Norvig's method).
- Atenção: sempre que fazemos isso perdemos informação!

Minimizando distâncias entre vetores

- Queremos minimizar distâncias entre vetores com características similares;
- Considere o exemplo abaixo:

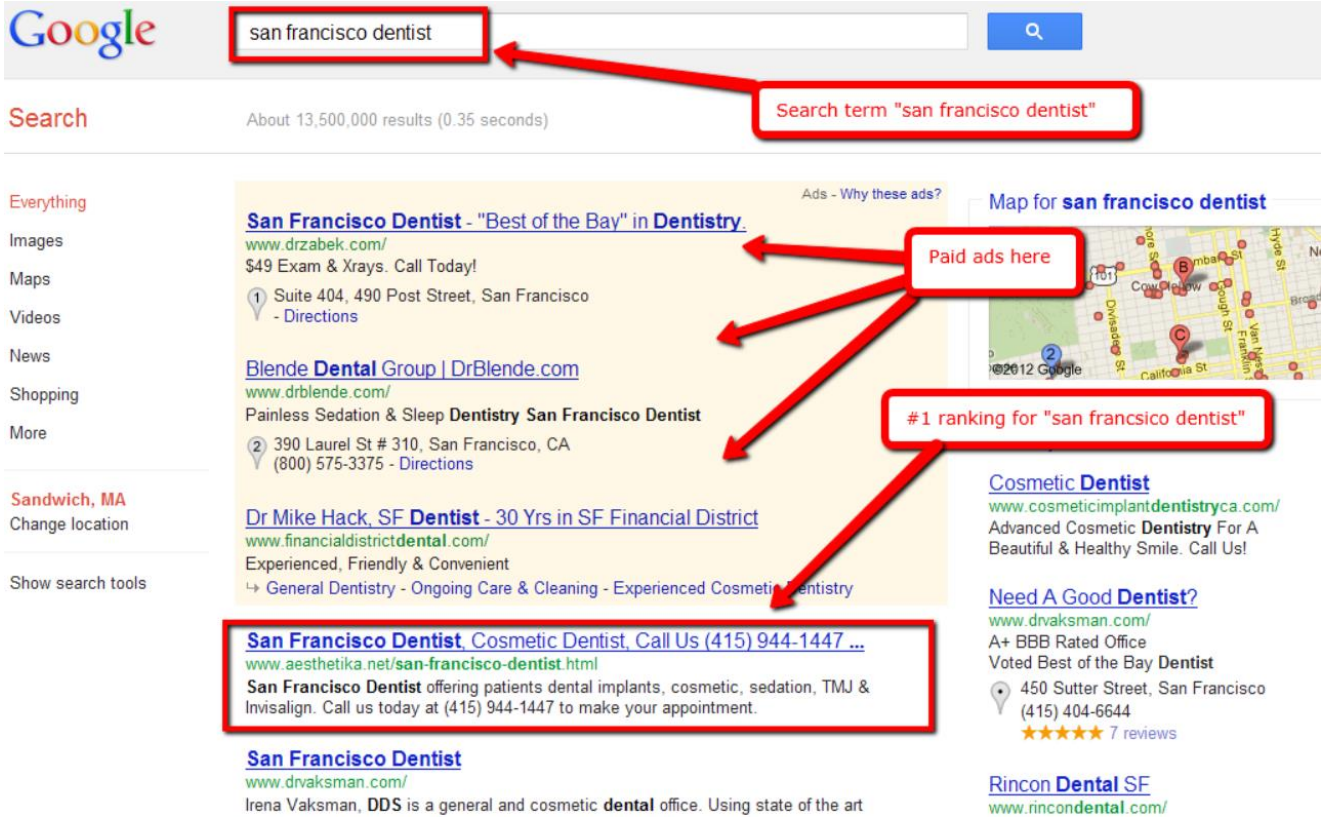
> "Eduardo quando crescer será Estatístico";
>
> "eduardo qundo crecser será estatístico".

- Gostaríamos que nosso algoritmo pontuasse as duas respostas da mesma forma;
- Entretanto, simples tokenização não captará tal semelhança;
- Assim, além de incluirmos apenas entradas minúsculas, devemos corrigir a ortografia (e.g. ▸ Peter Norvig's method).
- Atenção: sempre que fazemos isso perdemos informação!

Exemplo: Motores de busca (*Search Engine*)

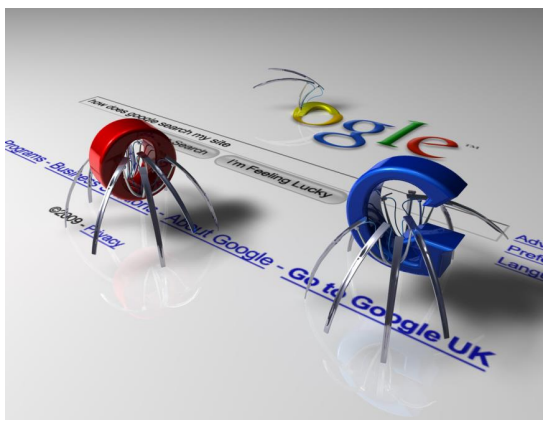
Motores de busca (Search Engine)

- Como encontramos informações que procuramos na internet?
- Como é determinado o ranking dos sites?



Motores de busca (Search Engine)

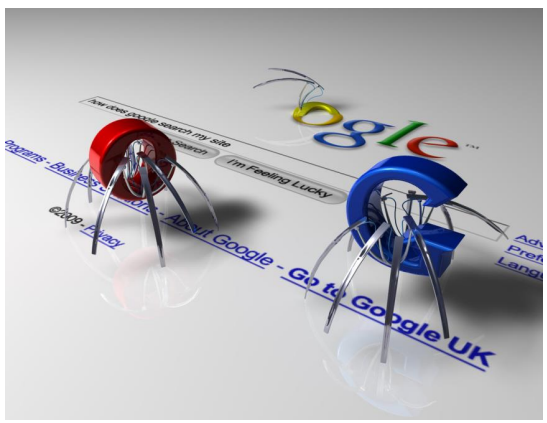
- Motores de busca são mecanismos para encontrar informações de texto a partir de palavras-chave indicadas pelo utilizador;
- Eles percorrem "toda" a internet em busca da informação que se pretende (documentos ou endereços de páginas web).
- A forma como a informação é indexada depende de cada motor de busca



  - ⋆ Por palavras, títulos e URL's (como é o caso do Google);
  - ⋆ Ou diretorias (como o Yahoo).

Motores de busca (Search Engine)

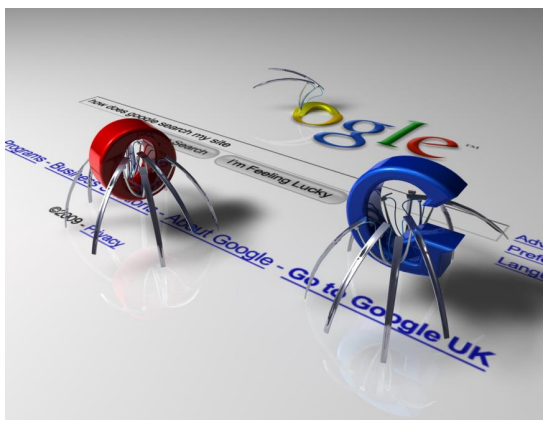
- Motores de busca são mecanismos para encontrar informações de texto a partir de palavras-chave indicadas pelo utilizador;
- Eles percorrem "toda" a internet em busca da informação que se pretende (documentos ou endereços de páginas web).
- A forma como a informação é indexada depende de cada motor de busca



  - ⋆ Por palavras, títulos e URL's (como é o caso do Google);
  - ⋆ Ou diretorias (como o Yahoo).

Motores de busca (Search Engine)

- Motores de busca são mecanismos para encontrar informações de texto a partir de palavras-chave indicadas pelo utilizador;
- Eles percorrem "toda" a internet em busca da informação que se pretende (documentos ou endereços de páginas web).
- A forma como a informação é indexada depende de cada motor de busca



- ⋆ Por palavras, títulos e URL's (como é o caso do Google);
- ⋆ Ou diretorias (como o Yahoo).

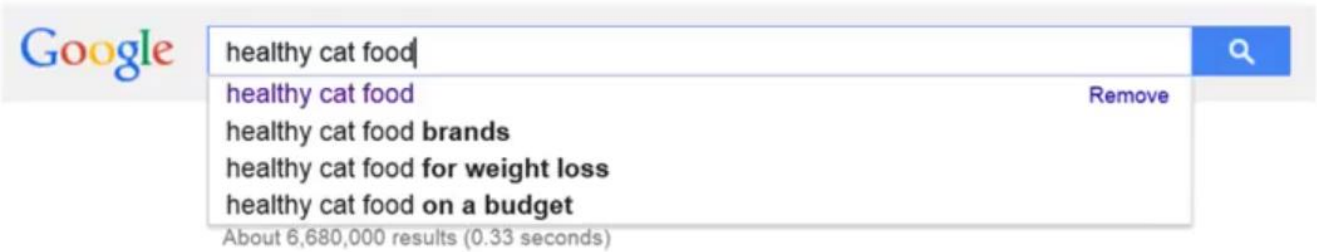
Motores de busca (Search Engine)

- Vamos construir nosso motor de busca para encontrar em um grupo de 7 websites (o Google utiliza mais de mil milhões) o melhor ranking;
- Nossa pesquisa será
- A visualização dos vetores do espaço fica então dessa forma.

Motores de busca (Search Engine)

- Vamos construir nosso motor de busca para encontrar em um grupo de 7 websites (o Google utiliza mais de mil milhões) o melhor ranking;

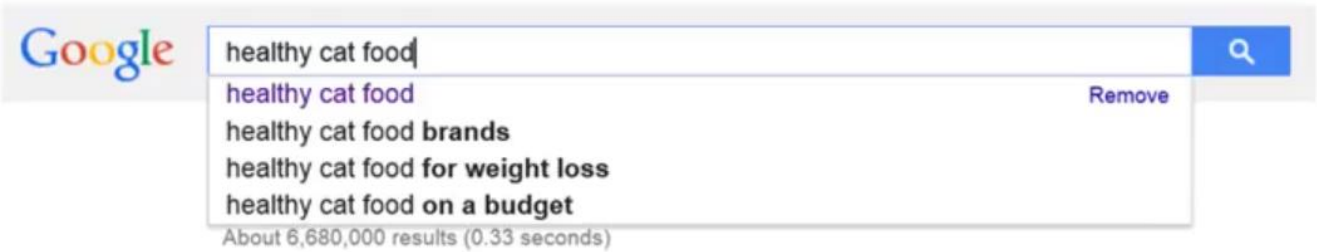
- Nossa pesquisa será



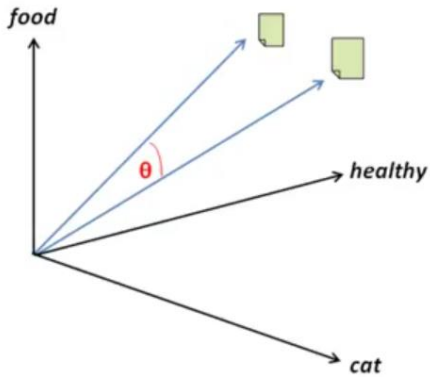
- A visualização dos vetores do espaço fica então dessa forma.

Motores de busca (Search Engine)

- Vamos construir nosso motor de busca para encontrar em um grupo de 7 websites (o Google utiliza mais de mil milhões) o melhor ranking;
- Nossa pesquisa será



- A visualização dos vetores do espaço fica então dessa forma.



Construindo o Corpus

- Primeiramente, devemos construir o **Corpus** (que é uma coleção de documentos de texto);
- Abaixo um caso de texto não estruturado para o nosso exemplo;
- Note que a maioria dos documentos contém alguma referência sobre **cat**, **healthy** ou **food**;

Construindo o Corpus

- Primeiramente, devemos construir o **Corpus** (que é uma coleção de documentos de texto);

- Abaixo um caso de texto não estruturado para o nosso exemplo;

Web Page	Text Field
1	"Stray cats are running all over the place. I see 10 a day!"
2	"Cats are killers. They kill billions of animals a year."
3	"The best food in Columbus, OH is the North Market."
4	"Brand A is the best tasting cat food around. Your cat will love it."
5	"Buy Brand C cat food for your cat. Brand C makes healthy and happy cats."
6	"The Arnold Classic came to town this weekend. It reminds us to be healthy."
7	"I have nothing to say. In summary, I have told you nothing."

- Note que a maioria dos documentos contém alguma referência sobre **cat**, **healthy** ou **food**;

Construindo o Corpus

- Primeiramente, devemos construir o **Corpus** (que é uma coleção de documentos de texto);

- Abaixo um caso de texto não estruturado para o nosso exemplo;

Web Page	Text Field
1	"Stray cats are running all over the place. I see 10 a day!"
2	"Cats are killers. They kill billions of animals a year."
3	"The best food in Columbus, OH is the North Market."
4	"Brand A is the best tasting cat food around. Your cat will love it."
5	"Buy Brand C cat food for your cat. Brand C makes healthy and happy cats."
6	"The Arnold Classic came to town this weekend. It reminds us to be healthy."
7	"I have nothing to say. In summary, I have told you nothing."

- Note que a maioria dos documentos contém alguma referência sobre **cat**, **healthy** ou **food**;

Preparando o Corpus para análise

- Como já dito, a fim de melhorar a qualidade da busca, precisamos preparar os dados antes da análise. Por exemplo em:

  Stray cats are running all over the place. I see 10 a day!

- Devemos realizar os seguintes passos:
  - ⋆ Remover pontuação:
    Stray cats are running all over the place. I see 10 a day!
  - ⋆ Stemming:
    Stray cats are running all over the place I see 10 a day
  - ⋆ Trocar os termos em letra maiúscula:
    Stray cat are run all over the place I see 10 a day
  - ⋆ Remover os números:
    stray cat are run all over the place I see 10 a day
  - ⋆ Eliminar os espaços desnecessários:
    stray cat are run all over the place I see a day!

Preparando o Corpus para análise

- Como já dito, a fim de melhorar a qualidade da busca, precisamos preparar os dados antes da análise. Por exemplo em:

  Stray cats are running all over the place. I see 10 a day!

- Devemos realizar os seguintes passos:
  - ⋆ Remover pontuação:
    Stray cats are running all over the place. I see 10 a day!
  - ⋆ Stemming:
    Stray cats are running all over the place I see 10 a day
  - ⋆ Trocar os termos em letra maiúscula:
    Stray cat are run all over the place I see 10 a day
  - ⋆ Remover os números:
    stray cat are run all over the place I see 10 a day
  - ⋆ Eliminar os espaços desnecessários:
    stray cat are run all over the place I see a day!

Preparando o Corpus para análise

- Como já dito, a fim de melhorar a qualidade da busca, precisamos preparar os dados antes da análise. Por exemplo em:

  Stray cats are running all over the place. I see 10 a day!

- Devemos realizar os seguintes passos:

  - ⋆ Remover pontuação:
    Stray cats are running all over the place. I see 10 a day!
  - ⋆ Stemming:
    Stray cats are running all over the place I see 10 a day
  - ⋆ Trocar os termos em letra maiúscula:
    Stray cat are run all over the place I see 10 a day
  - ⋆ Remover os números:
    stray cat are run all over the place I see 10 a day
  - ⋆ Eliminar os espaços desnecessários:
    stray cat are run all over the place I see a day!

Preparando o Corpus para análise

- Como já dito, a fim de melhorar a qualidade da busca, precisamos preparar os dados antes da análise. Por exemplo em:

  Stray cats are running all over the place. I see 10 a day!

- Devemos realizar os seguintes passos:

  - ⋆ Remover pontuação:
    Stray cats are running all over the place. I see 10 a day!
  - ⋆ Stemming:
    Stray cats are running all over the place I see 10 a day
  - ⋆ Trocar os termos em letra maiúscula:
    Stray cat are run all over the place I see 10 a day
  - ⋆ Remover os números:
    stray cat are run all over the place I see 10 a day
  - ⋆ Eliminar os espaços desnecessários:
    stray cat are run all over the place I see a day!

Preparando o Corpus para análise

- Como já dito, a fim de melhorar a qualidade da busca, precisamos preparar os dados antes da análise. Por exemplo em:

  Stray cats are running all over the place. I see 10 a day!

- Devemos realizar os seguintes passos:

  - ⋆ Remover pontuação:
    Stray cats are running all over the place. I see 10 a day!
  - ⋆ Stemming:
    Stray cats are running all over the place I see 10 a day
  - ⋆ Trocar os termos em letra maiúscula:
    Stray cat are run all over the place I see 10 a day
  - ⋆ Remover os números:
    stray cat are run all over the place I see 10 a day
  - ⋆ Eliminar os espaços desnecessários:
    stray cat are run all over the place I see a day!

Preparando o Corpus para análise

- Como já dito, a fim de melhorar a qualidade da busca, precisamos preparar os dados antes da análise. Por exemplo em:

  Stray cats are running all over the place. I see 10 a day!

- Devemos realizar os seguintes passos:

  - ⋆ Remover pontuação:
    Stray cats are running all over the place. I see 10 a day!
  - ⋆ Stemming:
    Stray cats are running all over the place I see 10 a day
  - ⋆ Trocar os termos em letra maiúscula:
    Stray cat are run all over the place I see 10 a day
  - ⋆ Remover os números:
    stray cat are run all over the place I see 10 a day
  - ⋆ Eliminar os espaços desnecessários:
    stray cat are run all over the place I see a day!

Preparando o Corpus para análise

- Como já dito, a fim de melhorar a qualidade da busca, precisamos preparar os dados antes da análise. Por exemplo em:

  Stray cats are running all over the place. I see 10 a day!

- Devemos realizar os seguintes passos:

  - ⋆ Remover pontuação:
    Stray cats are running all over the place. I see 10 a day!
  - ⋆ Stemming:
    Stray cats are running all over the place I see 10 a day
  - ⋆ Trocar os termos em letra maiúscula:
    Stray cat are run all over the place I see 10 a day
  - ⋆ Remover os números:
    stray cat are run all over the place I see 10 a day
  - ⋆ Eliminar os espaços desnecessários:
    stray cat are run all over the place I see a day!

Transformando o texto em matriz

- Neste caso, as linhas são os termos e as colunas são os documentos.

Terms	Web Page 1	Web Page 2	Web Page 3	Web Page 4	Web Page 5	Web Page 6	Web Page 7	Query
all	1	0	0	0	0	0	0	0
and	0	0	0	0	1	0	0	0
anim	0	1	0	0	0	0	0	0
are	1	1	0	0	0	0	0	0
arnold	0	0	0	0	0	1	0	0
around	0	0	0	1	0	0	0	0
best	0	0	1	1	0	0	0	0
billion	0	1	0	0	0	0	0	0
brand	0	0	0	1	2	0	0	0
buy	0	0	0	0	1	0	0	0
came	0	0	0	0	0	1	0	0
cat	1	1	0	2	3	0	0	1
classic	0	0	0	0	0	1	0	0
columbus	0	0	1	0	0	0	0	0
classic	0	0	0	0	0	1	0	0
columbus	0	0	1	0	0	0	0	0

Esta linha contém os valores do termo de consulta

- Na forma matricial fica

$$idf.matrix = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \dots & 0 \\ 1 & 1 & 0 & 2 & \dots & 1 \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \end{bmatrix}$$

Transformando o texto em matriz

- Neste caso, as linhas são os termos e as colunas são os documentos.

Terms	Web Page 1	Web Page 2	Web Page 3	Web Page 4	Web Page 5	Web Page 6	Web Page 7	Query
all	1	0	0	0	0	0	0	0
and	0	0	0	0	1	0	0	0
anim	0	1	0	0	0	0	0	0
are	1	1	0	0	0	0	0	0
arnold	0	0	0	0	0	1	0	0
around	0	0	0	1	0	0	0	0
best	0	0	1	1	0	0	0	0
billion	0	1	0	0	0	0	0	0
brand	0	0	0	1	2	0	0	0
buy	0	0	0	0	1	0	0	0
came	0	0	0	0	0	1	0	0
cat	1	1	0	2	3	0	0	1
classic	0	0	0	0	0	1	0	0
columbus	0	0	1	0	0	0	0	0
classic	0	0	0	0	0	1	0	0
columbus	0	0	1	0	0	0	0	0

Esta linha contém os valores do termo de consulta

- Na forma matricial fica

$$idf.matrix = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \dots & 0 \\ 1 & 1 & 0 & 2 & \dots & 1 \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \end{bmatrix}$$

Ponderando os termos nos documentos

- Note que os valores na nossa matriz são simples frequências observadas;
- Mas parece razoável supor que palavras raras pode impulsionar nosso algoritmo;
- P. ex., a palavra **healthy** aparece em somente um documento, enquanto **cat** aparece em 4;
- Então vamos ponderar as palavras pelo inverso da sua frequência (através do método IDF);

Ponderando os termos nos documentos

- Note que os valores na nossa matriz são simples frequências observadas;
- Mas parece razoável supor que palavras raras pode impulsionar nosso algoritmo;
- P. ex., a palavra **healthy** aparece em somente um documento, enquanto **cat** aparece em 4;
- Então vamos ponderar as palavras pelo inverso da sua frequência (através do método IDF);

Ponderando os termos nos documentos

- Note que os valores na nossa matriz são simples frequências observadas;
- Mas parece razoável supor que palavras raras pode impulsionar nosso algoritmo;
- P. ex., a palavra **healthy** aparece em somente um documento, enquanto **cat** aparece em 4;
- Então vamos ponderar as palavras pelo inverso da sua frequência (através do método IDF);

Ponderando os termos nos documentos

- Note que os valores na nossa matriz são simples frequências observadas;
- Mas parece razoável supor que palavras raras pode impulsionar nosso algoritmo;
- P. ex., a palavra **healthy** aparece em somente um documento, enquanto **cat** aparece em 4;
- Então vamos ponderar as palavras pelo inverso da sua frequência (através do método IDF);

Terms	Web Page 1	Web Page 2	Web Page 3	Web Page 4	Web Page 5	Web Page 6	Web Page 7	Query
cat	1	1	0	2	3	0	0	1

Terms	Weighting 1	Weighting 2	Weighting 3	Weighting 4	Weighting 5	Weighting 6	Weighting 7	Query
cat	0.8073549	0.8073549	0	1.61471	2.086982	0	0	0.807355

Calculando as similaridades

- Uma das vantagens de se trabalhar com vetores no espaço é poder calcular correlações através da geometria;
- Assim, a partir do produto interno entre os vetores (normalizados) temos uma medida sobre o grau de similaridade entre eles ($cos(\theta)$);
- Para tanto, considere `query.vector` como última coluna da matriz `idf.matrix`

```
query.vector <- idf.matrix[,(N.docs+1)]
idf.matrix <- idf.matrix[,1:N.docs]
```

- Então, temos

```
doc.scores <- t(query.vector) %*% idf.matrix
```

Calculando as similaridades

- Uma das vantagens de se trabalhar com vetores no espaço é poder calcular correlações através da geometria;
- Assim, a partir do produto interno entre os vetores (normalizados) temos uma medida sobre o grau de similaridade entre eles ($cos(\theta)$);
- Para tanto, considere `query.vector` como última coluna da matriz `idf.matrix`

```
query.vector <- idf.matrix[,(N.docs+1)]
idf.matrix <- idf.matrix[,1:N.docs]
```

- Então, temos

```
doc.scores <- t(query.vector) %*% idf.matrix
```

Calculando as similaridades

- Uma das vantagens de se trabalhar com vetores no espaço é poder calcular correlações através da geometria;
- Assim, a partir do produto interno entre os vetores (normalizados) temos uma medida sobre o grau de similaridade entre eles ($cos(\theta)$);
- Para tanto, considere `query.vector` como última coluna da matriz `idf.matrix`

```
query.vector <- idf.matrix[,(N.docs+1)]
idf.matrix <- idf.matrix[,1:N.docs]
```

- Então, temos

```
doc.scores <- t(query.vector) %*% idf.matrix
```

Calculando as similaridades

- Uma das vantagens de se trabalhar com vetores no espaço é poder calcular correlações através da geometria;
- Assim, a partir do produto interno entre os vetores (normalizados) temos uma medida sobre o grau de similaridade entre eles ($cos(\theta)$);
- Para tanto, considere `query.vector` como última coluna da matriz `idf.matrix`

```
query.vector <- idf.matrix[,(N.docs+1)]
idf.matrix <- idf.matrix[,1:N.docs]
```

- Então, temos

```
doc.scores <- t(query.vector) %*% idf.matrix
```

Calculando as similaridades

- O que fizemos foi (não com esses valores!)

$$\underset{t(query.vector)}{\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}} \underset{idf.matrix}{\begin{bmatrix} 2 & 1 & 3 \\ 3 & 3 & 2 \\ 4 & 1 & 2 \end{bmatrix}} = \underset{doc.scores}{\begin{bmatrix} 1 \times 2 + 2 \times 3 + 3 \times 4 \\ 1 \times 1 + 2 \times 3 + 3 \times 1 \\ 1 \times 3 + 2 \times 2 + 3 \times 2 \end{bmatrix}}$$

- Com os escores nas mãos, basta ordená-los e descobrir as melhores indicações

- Note que, devido a ponderação, a *web page* 6 ficou em segundo lugar, ainda que apresentou somente um termo, porém "raro".

Calculando as similaridades

- O que fizemos foi (não com esses valores!)

$$\underbrace{\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}}_{t(query.vector)} \begin{bmatrix} 2 & 1 & 3 \\ 3 & 3 & 2 \\ 4 & 1 & 2 \end{bmatrix}_{idf.matrix} = \underbrace{\begin{bmatrix} 1\times 2+2\times 3+3\times 4 \\ 1\times 1+2\times 3+3\times 1 \\ 1\times 3+2\times 2+3\times 2 \end{bmatrix}}_{doc.scores}$$

- Com os escores nas mãos, basta ordená-los e descobrir as melhores indicações

Web Page	Score	Text Field
5	0.344	Buy Brand C cat food for your cat. Brand C makes healthy and happy cats.
6	0.183	The Arnold Classic came to town this weekend. It reminds us to be healthy.
4	0.177	Brand A is the best tasting cat food around. Your cat will love it.
3	0.115	The best food in Columbus, OH is the North Market.
2	0.039	Cats are killers. They kill billions of animals a year.
1	0.036	Stray cats are running all over the place. I see 10 a day!
7	0.000	I have nothing to say. In summary, I have told you nothing.

- Note que, devido a ponderação, a *web page* 6 ficou em segundo lugar, ainda que apresentou somente um termo, porém "raro".

Calculando as similaridades

- O que fizemos foi (não com esses valores!)

$$\underset{t(query.vector)}{\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}} \underset{idf.matrix}{\begin{bmatrix} 2 & 1 & 3 \\ 3 & 3 & 2 \\ 4 & 1 & 2 \end{bmatrix}} = \underset{doc.scores}{\begin{bmatrix} 1\times 2+2\times 3+3\times 4 \\ 1\times 1+2\times 3+3\times 1 \\ 1\times 3+2\times 2+3\times 2 \end{bmatrix}}$$

- Com os escores nas mãos, basta ordená-los e descobrir as melhores indicações

Web Page	Score	Text Field
5	0.344	Buy Brand C cat food for your cat. Brand C makes healthy and happy cats.
6	0.183	The Arnold Classic came to town this weekend. It reminds us to be healthy.
4	0.177	Brand A is the best tasting cat food around. Your cat will love it.
3	0.115	The best food in Columbus, OH is the North Market.
2	0.039	Cats are killers. They kill billions of animals a year.
1	0.036	Stray cats are running all over the place. I see 10 a day!
7	0.000	I have nothing to say. In summary, I have told you nothing.

- Note que, devido a ponderação, a *web page* 6 ficou em segundo lugar, ainda que apresentou somente um termo, porém "raro".

Exemplo: Classificação de textos

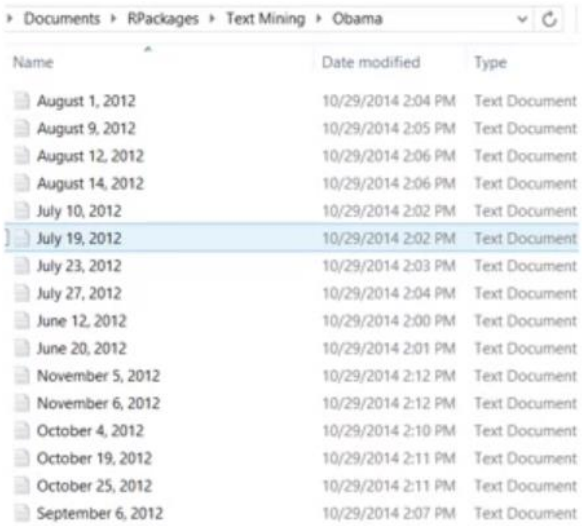
Campanha presidencial EUA-2012

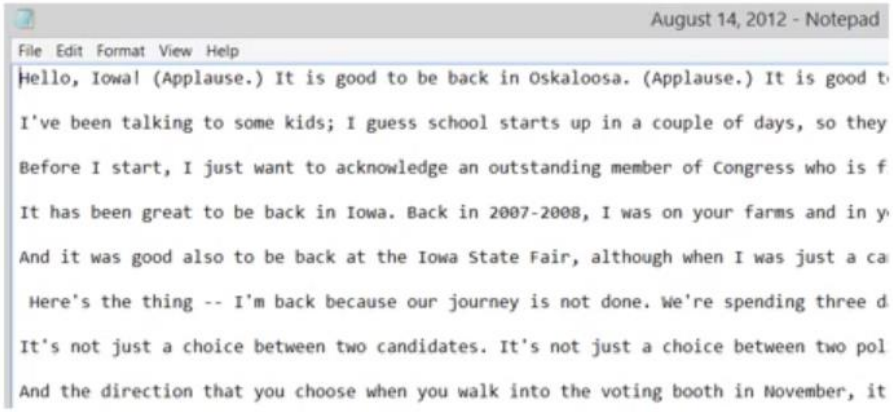
- Os discursos de Obama e Romney foram gravados e transcritos;
- A partir desses documentos, queremos encontrar um padrão no discurso;

- O algoritmo preditivo deve ser capaz de, a partir de um discurso desconhecido, determinar qual candidato o fez;
- Essa técnica de classificação tem aplicações em detecção de spam, fraude, dentre outras.

Campanha presidencial EUA-2012

- Todos os discursos foram arquivados em pastas para ambos os candidatos. Sem se fazer qualquer "limpeza" no texto.





Construindo o Corpus

- Inicialmente, precisamos construir o Corpus (coleção de textos);
- Para tanto, vamos aplicar um loop nos discursos removendo pontuações, espaço em branco, *stopwords* etc.;

```
cleanCorpus <- function(corpus){
  corpus.tmp <- tm_map(corpus, removePunctuation)
  corpus.tmp <- tm_map(corpus.tmp, stripWhitespace)
  corpus.tmp <- tm_map(corpus.tmp, tolower)
  corpus.tmp <- tm_map(corpus.tmp, removeWords, stopwords(''english''))
  return(corpus.tmp)
}
```

Construindo o Corpus

- Inicialmente, precisamos construir o Corpus (coleção de textos);
- Para tanto, vamos aplicar um loop nos discursos removendo pontuações, espaço em branco, *stopwords* etc.;

```
cleanCorpus <- function(corpus){
  corpus.tmp <- tm_map(corpus, removePunctuation)
  corpus.tmp <- tm_map(corpus.tmp, stripWhitespace)
  corpus.tmp <- tm_map(corpus.tmp, tolower)
  corpus.tmp <- tm_map(corpus.tmp, removeWords, stopwords(``english''))
  return(corpus.tmp)
}
```

Criando a matriz de documentos

- Neste exemplo temos 1330 termos e nas colunas as frequências de cada um nos discursos.

Document	Candidate	ability	achieve	across	act	advantage	afghanistan	agenda	always	america
August 1, 2012	Romney	1	1	0	1	0	0	1	0	7
August 12, 2012	Romney	0	1	2	0	1	0	3	0	19
August 14, 2012	Romney	3	1	2	1	0	1	0	0	11
August 9, 2012	Romney	0	0	0	0	1	0	0	8	53
July 10, 2012	Romney	0	0	2	0	2	0	1	2	7
July 19, 2012	Romney	2	0	1	0	2	3	0	2	17
July 23, 2012	Romney	1	1	0	0	0	1	0	1	8
July 27, 2012	Romney	0	1	2	1	0	2	0	0	5
June 12, 2012	Romney	1	0	3	0	0	0	1	1	29
June 20, 2012	Romney	0	0	1	1	0	0	0	1	18
November 5, 2012	Romney	0	1	4	0	0	0	0	0	12
November 6, 2012	Romney	0	1	6	5	1	0	1	0	9
October 19, 2012	Romney	0	0	2	0	0	0	0	0	5
August 1, 2012	Obama	0	0	1	1	0	0	0	2	12
August 12, 2012	Obama	0	0	3	3	0	1	0	5	20
August 14, 2012	Obama	0	0	9	0	0	1	0	1	27
August 9, 2012	Obama	0	0	4	2	0	1	0	0	12
July 10, 2012	Obama	0	0	5	1	0	1	0	4	15
July 19, 2012	Obama	0	0	7	3	0	1	0	4	16
July 23, 2012	Obama	0	0	3	2	0	6	0	3	20
July 27, 2012	Obama	0	0	0	2	0	1	0	3	17
June 12, 2012	Obama	0	0	7	0	0	1	0	3	12
June 20, 2012	Obama	0	0	1	0	0	1	0	5	12
November 5, 2012	Obama	0	0	3	1	0	1	0	9	8
November 6, 2012	Obama	0	0	0	0	0	0	0	4	15

Preparação dos dados para predição

- Separamos os dados em treino (70%) e teste (30%);

Dados de treino

Candidate	ability	achieve	across	act	advantage	afghanistan	agenda	always	america
Romney	1	1	0	1	0	0	1	0	7
Romney	0	1	2	0	1	0	3	0	19
Romney	3	1	2	1	0	1	0	0	11
Romney	0	0	0	0	1	0	0	8	53
Romney	0	0	2	0	2	0	1	2	7
Romney	2	0	1	0	2	3	0	2	17
Romney	1	1	0	0	0	1	0	1	8
Romney	0	1	2	1	0	2	0	0	5
Romney	1	0	3	0	0	0	1	1	29
Obama	0	0	1	1	0	0	0	2	12
Obama	0	0	3	3	0	1	0	5	20
Obama	0	0	9	0	0	1	0	1	27
Obama	0	0	4	2	0	1	0	0	12
Obama	0	0	5	1	0	1	0	4	15
Obama	0	0	7	3	0	1	0	4	16
Obama	0	0	3	2	0	6	0	3	20

Dados de teste

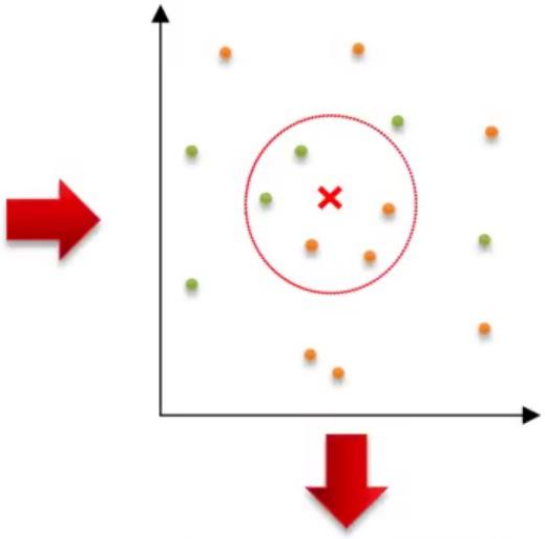
Speech	ability	achieve	across	act	advantage	afghanistan	agenda	always	america
Speech 1	0	0	1	1	0	0	0	1	18
Speech 2	0	1	4	0	0	0	0	0	12
Speech 3	0	1	6	5	1	0	1	0	9
Speech 4	0	0	2	0	0	0	0	0	5
Speech 5	0	0	0	2	0	1	0	3	17
Speech 6	0	0	7	0	0	1	0	3	12
Speech 7	0	0	1	0	0	1	0	5	12
Speech 8	0	0	3	1	0	1	0	9	8
Speech 9	0	0	0	0	0	0	0	4	15

- Note que a coluna dos candidatos foi removida dos dados de teste, pois será predita a partir dos discursos.

Predição dos candidatos

- Utilizando um algoritmo de classificação, chegamos no seguinte resultado

Speech	ability	achieve	across	act	advantage	afghanistan	agenda	always	america
Speech 1	0	0	1	1	0	0	0	1	18
Speech 2	0	1	4	0	0	0	0	0	12
Speech 3	0	1	6	5	1	0	1	0	9
Speech 4	0	0	2	0	0	0	0	0	5
Speech 5	0	0	0	2	0	1	0	3	17
Speech 6	0	0	7	0	0	1	0	3	12
Speech 7	0	0	1	0	0	1	0	5	12
Speech 8	0	0	3	1	0	1	0	9	8
Speech 9	0	0	0	0	0	0	0	4	15



Speech	Actual	Predicted
Speech 1	Romney	Romney
Speech 2	Obama	Obama
Speech 3	Romney	Romney
Speech 4	Romney	Obama
Speech 5	Obama	Obama
Speech 6	Romney	Romney
Speech 7	Obama	Obama
Speech 8	Obama	Obama
Speech 9	Obama	Obama

Mais informações e exemplos

- ▸ Introduction to the tm Package - Text Mining in R
- ▸ Text Mining Infrastructure in R
- ▸ Text Mining Handbook
- ▸ Distributed Text Mining in R
- ▸ Text mining with Twitter and R

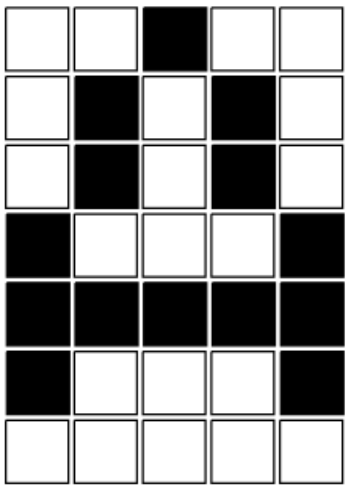
Manipulando Imagens

- Inicialmente, vamos entender como as imagens do tipo *raster* são representadas (e.g. JPEG, PNG, ...);
    - ⋆ *Raster* são imagens que contêm a descrição de cada pixel, em oposição aos gráficos vetoriais.
- Vamos começar com uma ideia simples, utilizando uma matriz binária

Manipulando Imagens

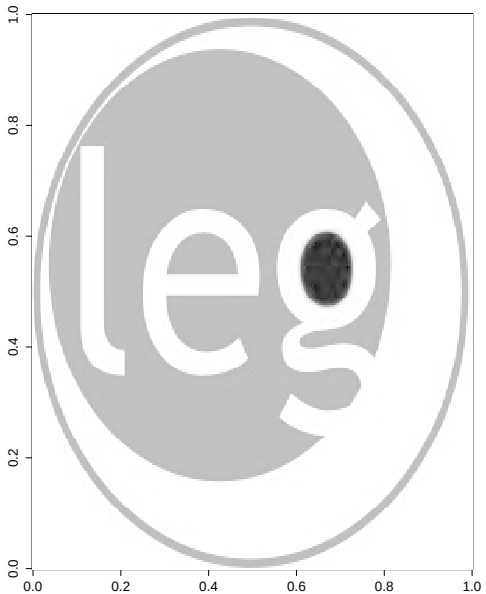
- Inicialmente, vamos entender como as imagens do tipo *raster* são representadas (e.g. JPEG, PNG, ...);
  - ⋆ *Raster* são imagens que contêm a descrição de cada pixel, em oposição aos gráficos vetoriais.
- Vamos começar com uma ideia simples, utilizando uma matriz binária

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



Manipulando Imagens

- Podemos ir um passo além



- Ao invés de usar apenas 0 (branco) e 1 (preto), usamos números entre 0 e 1 para denotar a intensidade de cinza.
- Quanto mais pixels, maior a resolução.

```
library(jpeg)
imagem=readJPEG(‘‘282px-leg.jpg")
image(t(imagem[282:1,,3]),col = grey.colors(1000,start = 0,end =1))
```

Manipulando Imagens

- Veja como fica o algarismo 7 escrito a mão.

```
> e<-matrix(c(dados[10,2:257]),16,,byrow=TRUE)
> e
      [,1]   [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9]  [,10]  [,11]  [,12]  [,13] [,14]  [,15]  [,16]
 [1,] -0.045 0.494  0.431 0.082 0.494 0.494 0.494 0.494 0.494 0.494  -0.098 0.375  0.184 -0.195 -0.372 -0.981
 [2,] 0.59   1      1     1     1     1     1     1     1     1      1      1      1     1      1      0.201
 [3,] -0.145 1      1     1     0.419 0.326 0.326 0.188 -0.362 -0.362 -0.292 0.042  0.54  1      1      0.432
 [4,] -0.996 -0.161 0.939 0.682 -0.781 -1    -1    -1    -1    -1     -1     -0.608 0.864 1      0.903  -0.752
 [5,] -1     -1     -0.932 -0.956 -1    -1    -1    -1    -1    -1     -0.833 0.832  1     1      0.544  -0.996
 [6,] -1     -1     -1    -1    -1    -1    -1    -1    -1    -1     0.262  1      1     0.772  -0.623 -1
 [7,] -1     -1     -1    -1    -1    -1    -1    -1    -1    -0.444 0.97   1      0.758 -0.871 -1     -1
 [8,] -1     -1     -1    -1    -1    -1    -1    -1    -0.917 0.232  1      0.951  0.028 -1     -1     -1
 [9,] -1     -1     -1    -1    -1    -1    -1    -1    -0.28  1      1      0.478  -1    -1     -1     -1
[10,] -1     -1     -1    -1    -1    -1    -1    -0.674 0.612  1      1      -0.12  -1    -1     -1     -1
[11,] -1     -1     -1    -1    -1    -1    -1    0.055  1      1      0.822  -0.826 -1    -1     -1     -1
[12,] -1     -1     -1    -1    -1    -1    -0.744 0.898  1      0.992  -0.261 -1     -1    -1     -1     -1
[13,] -1     -1     -1    -1    -1    -0.812 0.472  1      1      0.467  -0.998 -1     -1    -1     -1     -1
[14,] -1     -1     -1    -1    -0.832 0.439  1      1      0.522  -0.791 -1     -1     -1    -1     -1     -1
[15,] -1     -1     -1    -1    0.148  1      1      0.922  -0.692 -1     -1     -1     -1    -1     -1     -1
[16,] -1     -1     -1    -1    -0.633 0.471  0.061  -0.889 -1     -1     -1     -1     -1    -1     -1     -1
> e<-matrix(c(dados[16,2:257]),16,,byrow=TRUE)
> e
      [,1]   [,2]  [,3]   [,4]  [,5]  [,6]  [,7]  [,8]  [,9]  [,10]  [,11]  [,12]  [,13]  [,14]  [,15]  [,16]
 [1,] -1     -1    -1     -0.99 0.449 0.22  -0.83 -1    -1    -1     -1     -1     -1     -1     -0.972 -1
 [2,] -1     -1    -0.643 0.293 1     1     0.83  0.3   -0.068 -0.294 -0.294 -0.078 0.176  0.357  0.73   -0.869
 [3,] -1     -0.35 0.939  1     1     0.717 0.951 1     1      1      1      1      1      1      1      -0.517
 [4,] -0.466 0.969 1      0.83  -0.466 -0.974 -0.85 -0.412 -0.103 0.059  0.059  -0.222 -0.412 0.316  1      -0.161
 [5,] -0.105 1     0.514  -0.703 -1    -1    -1    -1    -1     -1     -1     -1     -0.354 1      0.997  -0.218
 [6,] -0.957 -0.315 -0.897 -1    -1    -1    -1    -1    -1     -1     -1     -0.66  0.833  1      0.252  -0.999
 [7,] -1     -1    -1     -1    -1    -1    -1    -1    -1     -1     -0.947 0.674  1      0.551  -0.935 -1
 [8,] -1     -1    -1     -1    -1    -1    -1    -1    -1     -1     0.198  1      0.78   -0.802 -1     -1
 [9,] -1     -1    -1     -1    -1    -1    -1    -1    -0.984 0.096  1      0.914  -0.601 -1     -1     -1
[10,] -1     -1    -1     -1    -1    -1    -1    -1    -0.029 1      0.981  -0.305 -1     -1     -1     -1
[11,] -1     -1    -1     -1    -1    -1    -1    -0.558 0.973  1      0.039  -1     -1     -1     -1     -1
[12,] -1     -1    -1     -1    -1    -1    -0.896 0.671  1      0.381  -0.98  -1     -1     -1     -1     -1
[13,] -1     -1    -1     -1    -1    -1    -0.058 1      0.883  -0.787 -1     -1     -1     -1     -1     -1
[14,] -1     -1    -1     -1    -1    -0.875 0.845  1      -0.098 -1     -1     -1     -1     -1     -1     -1
[15,] -1     -1    -1     -1    -1    -0.5  1      1      -0.213 -1     -1     -1     -1     -1     -1     -1
[16,] -1     -1    -1     -1    -1    -0.949 0.484  0.463  -0.926 -1     -1     -1     -1     -1     -1     -1
> e<-matrix(c(dados[23,2:257]),16,,byrow=TRUE)
> e
      [,1] [,2] [,3] [,4]  [,5]  [,6]  [,7]  [,8]  [,9]  [,10]  [,11]  [,12]  [,13]  [,14] [,15] [,16]
 [1,] -1   -1   -1   -0.45 0.897 0.655 0.407 -0.102 -0.623 -0.991 -1     -1     -1     -1    -1    -1
 [2,] -1   -1   -1   -1    -0.697 -0.354 0.102 0.403  1     0.502  -0.725 -1     -1     -1    -1    -1
 [3,] -1   -1   -1   -1    -1    -1    -1    -1    -0.84  0.162  0.924  -0.417 -1     -1    -1    -1
 [4,] -1   -1   -1   -1    -1    -1    -1    -1    -1     -1     -0.337 0.886  -0.865 -1    -1    -1
 [5,] -1   -1   -1   -1    -1    -1    -1    -1    -1     -1     -0.996 0.688  -0.356 -1    -1    -1
 [6,] -1   -1   -1   -1    -1    -1    -1    -1    -1     -1     0.378  -0.081 -1     -1    -1    -1
 [7,] -1   -1   -1   -1    -1    -1    -1    -1    -1     -1     0.518  -0.221 -1     -1    -1    -1
 [8,] -1   -1   -1   -1    -1    -1    -1    -1    -1     -1     -0.924 0.887  -0.718 -1    -1    -1
 [9,] -1   -1   -1   -1    -1    -1    -1    -1    -1     -1     -0.143 0.493  -1     -1    -1    -1
[10,] -1   -1   -1   -1    -1    -1    -1    -1    -1     -0.799 0.898  -0.64  -1     -1    -1    -1
[11,] -1   -1   -1   -1    -1    -1    -1    -1    -0.906 0.622  -0.091 -1     -1     -1    -1    -1
[12,] -1   -1   -1   -1    -1    -1    -1    -0.909 0.451  0.122  -1     -1     -1     -1    -1    -1
[13,] -1   -1   -1   -1    -1    -1    -0.998 0.192  0.271  -0.986 -1     -1     -1     -1    -1    -1
[14,] -1   -1   -1   -1    -1    -1    -0.284 0.685  -0.98  -1     -1     -1     -1     -1    -1    -1
[15,] -1   -1   -1   -1    -1    -0.57 0.857 -0.749 -1     -1     -1     -1     -1     -1    -1    -1
[16,] -1   -1   -1   -1    -1    0.45  -0.465 -1    -1     -1     -1     -1     -1     -1    -1    -1
> e<-matrix(c(dados[25,2:257]),16,,byrow=TRUE)
> e
      [,1]   [,2]  [,3]   [,4]  [,5]  [,6]  [,7]  [,8]  [,9]  [,10]  [,11]  [,12]  [,13]  [,14]  [,15]  [,16]
 [1,] -1     -1    -1     -1    -1    0.08  0.93  0.333 -0.101 -0.334 -0.334 -0.334 0.101  0.328  -0.672 -1
 [2,] -1     -1    -1     -1    0.028 1     1     1     1      1      1      1      1      1      0.947  -0.864
 [3,] -1     -1    -0.903 0.323 1     1     0.681 -0.288 -0.333 0.288  0.334  0.334  0.653  1      1      -0.421
 [4,] -1     -0.92 0.288  1     1     0.698 -0.733 -1    -1     -1     -1     -1     0.146  1      0.979  -0.792
 [5,] -1     0.093 1      1     0.715 -0.698 -1    -1    -1     -1     -1     -0.927 0.91   1      0.563  -0.99
 [6,] -0.605 0.979 1      0.382 -0.979 -1    -1    -1    -1     -1     -0.955 0.219  1      0.952  -0.882 -1
 [7,] -0.677 0.999 0.42   -0.965 -1    -1    -1    -1    -1     -1     -0.455 1      1      0.233  -1     -1
 [8,] -1     -0.608 -0.948 -1    -1    -1    -1    -1    -1     -0.984 0.588  1      0.767  -0.594 -1     -1
 [9,] -1     -1    -1     -1    -1    -1    -1    -1    -1     -0.115 1      0.92   -0.917 -1     -1     -1
[10,] -1     -1    -1     -1    -1    -1    -1    -1    -1     -0.998 0.72   1      0.167  -1     -1     -1
[11,] -1     -1    -1     -1    -1    -1    -1    -1    -0.441 1      0.781  -0.896 -1     -1     -1     -1
[12,] -1     -1    -1     -1    -1    -1    -1    -1    0.396  1      0.271  -1     -1     -1     -1     -1
[13,] -1     -1    -1     -1    -1    -1    -1    -0.905 0.915  0.843  -0.853 -1     -1     -1     -1     -1
[14,] -1     -1    -1     -1    -1    -1    -1    -0.598 1      0.375  -1     -1     -1     -1     -1     -1
[15,] -1     -1    -1     -1    -1    -1    -1    0.017  1      -0.017 -1     -1     -1     -1     -1     -1
[16,] -1     -1    -1     -1    -1    -1    -1    -0.573 0.871  -0.187 -1     -1     -1     -1     -1     -1
```

Manipulando Imagens

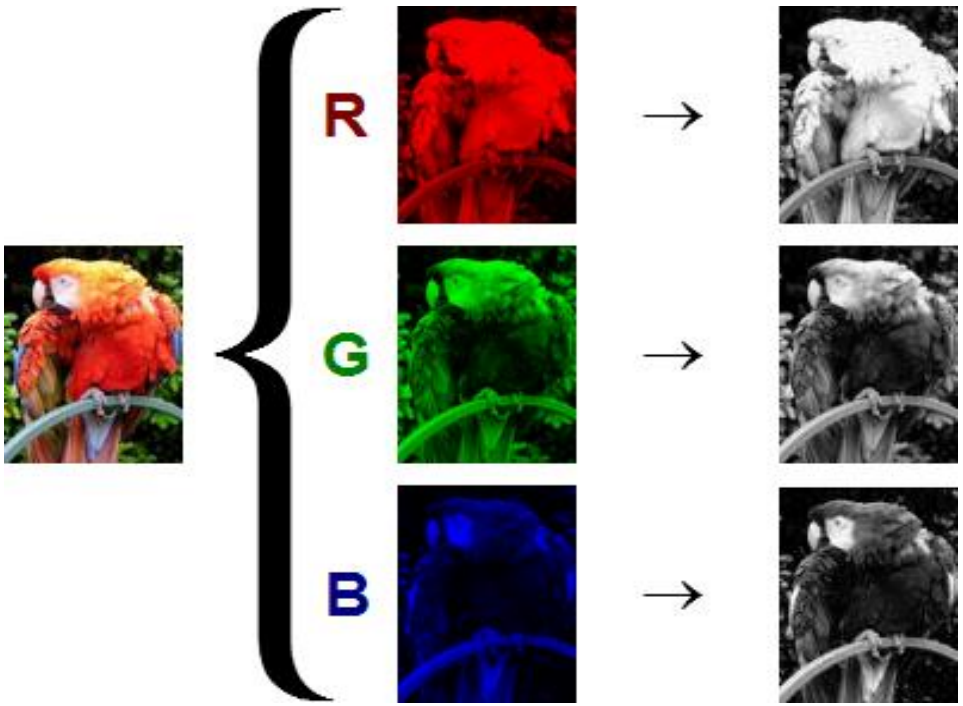
- Usando essa ideia, podemos representar uma imagem a partir de três matrizes simultaneamente (com as cores primárias. Cada elemento é um número entre 0 e 1)

- A primeira indica o quanto de azul em cada pixel;
- A segunda indica o quanto de amarelo;
- A terceira indica o quanto de vermelho.

```
library(jpeg)
imagem=readJPEG(‘‘282px-leg.jpg")
rasterImage(imagem, 0, 0, 1, 1)
```

Manipulando Imagens

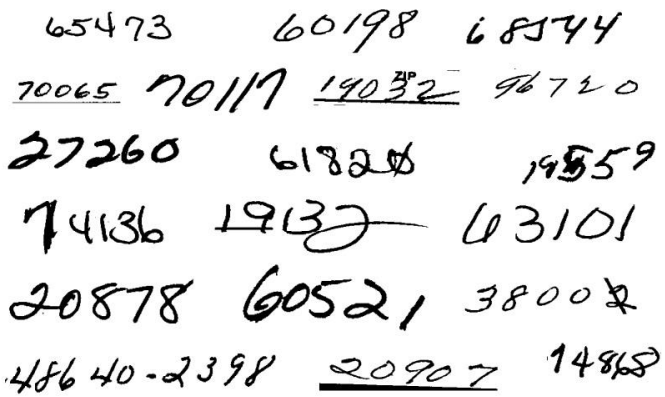
- Podemos, ao invés das cores primárias, utilizar o vermelho, verde e azul (*RGB channels*)



Exemplo: Classificação de dígitos escritos à mão

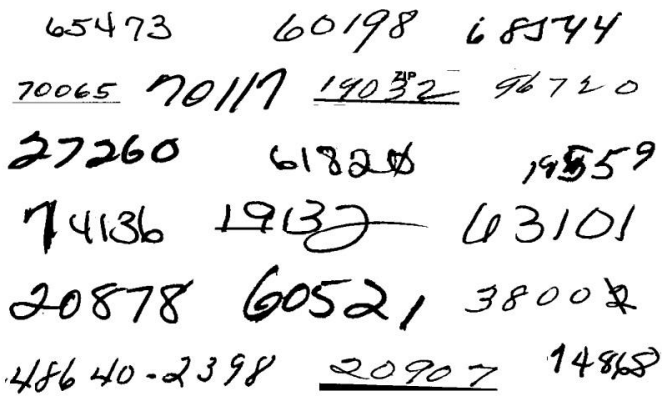
Classificação de dígitos escritos à mão

- O reconhecimento de imagens é um assunto muito estudado devido a sua variedade de aplicações;
- A classificação de dígitos escritos à mão é um dos assuntos mais discutidos nesta área, e muitos métodos foram desenvolvidos ao longo dos anos;
- A dificuldade deste reconhecimento é causada pela alta variabilidade das imagens;
- Neste exemplo vamos aplicar algumas técnicas vistas nas aulas anteriores para classificar os dígitos 1,2 e 7 escritos à mão.



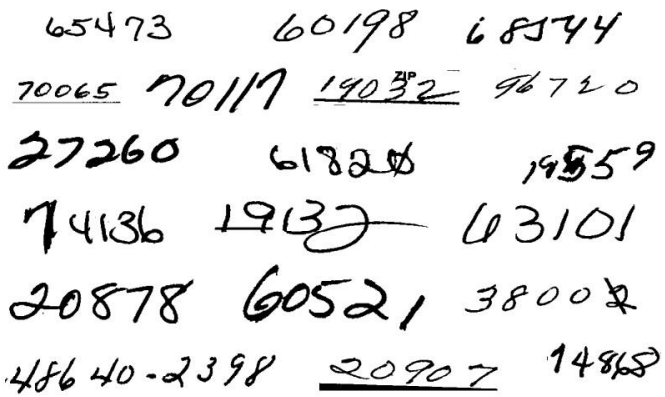
Classificação de dígitos escritos à mão

- O reconhecimento de imagens é um assunto muito estudado devido a sua variedade de aplicações;
- A classificação de dígitos escritos à mão é um dos assuntos mais discutidos nesta área, e muitos métodos foram desenvolvidos ao longo dos anos;
- A dificuldade deste reconhecimento é causada pela alta variabilidade das imagens;
- Neste exemplo vamos aplicar algumas técnicas vistas nas aulas anteriores para classificar os dígitos 1,2 e 7 escritos à mão.



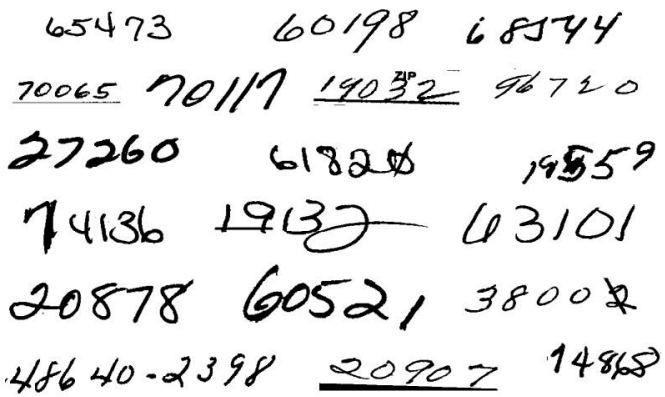
Classificação de dígitos escritos à mão

- O reconhecimento de imagens é um assunto muito estudado devido a sua variedade de aplicações;
- A classificação de dígitos escritos à mão é um dos assuntos mais discutidos nesta área, e muitos métodos foram desenvolvidos ao longo dos anos;
- A dificuldade deste reconhecimento é causada pela alta variabilidade das imagens;
- Neste exemplo vamos aplicar algumas técnicas vistas nas aulas anteriores para classificar os dígitos 1,2 e 7 escritos à mão.



Classificação de dígitos escritos à mão

- O reconhecimento de imagens é um assunto muito estudado devido a sua variedade de aplicações;
- A classificação de dígitos escritos à mão é um dos assuntos mais discutidos nesta área, e muitos métodos foram desenvolvidos ao longo dos anos;
- A dificuldade deste reconhecimento é causada pela alta variabilidade das imagens;
- Neste exemplo vamos aplicar algumas técnicas vistas nas aulas anteriores para classificar os dígitos 1,2 e 7 escritos à mão.



Preparação dos dados e métodos

- Os dígitos originais, de diferentes tamanhos e orientações, foram normalizados resultando em imagens 16×16 em escala de cinza;
- E, posteriormente, rearranjados em uma coluna de 256 colunas relativa às cores e uma referente ao dígito em questão;
- Note que a variável resposta é categórica (assume somente três valores: 1,2 ou 7);
- Os métodos em competição são:
  - *k-Nearest Neighbors* (K-NN);
  - *Linear discriminant analysis* (LDA);
  - *Quadratic discriminant analysis* (QDA);
  - *Support vector machine* (SVM);
  - Regressão logística.

Preparação dos dados e métodos

- Os dígitos originais, de diferentes tamanhos e orientações, foram normalizados resultando em imagens 16×16 em escala de cinza;
- E, posteriormente, rearranjados em uma coluna de 256 colunas relativa às cores e uma referente ao dígito em questão;
- Note que a variável resposta é categórica (assume somente três valores: 1,2 ou 7);
- Os métodos em competição são:
  - *k-Nearest Neighbors* (K-NN);
  - *Linear discriminant analysis* (LDA);
  - *Quadratic discriminant analysis* (QDA);
  - *Support vector machine* (SVM);
  - Regressão logística.

Preparação dos dados e métodos

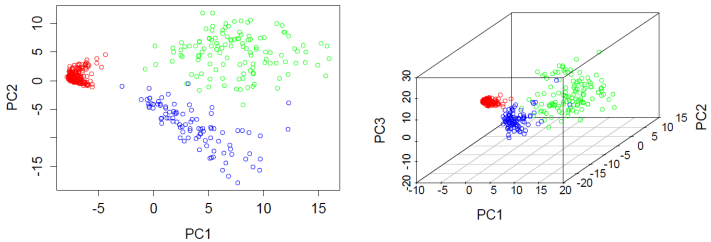
- Os dígitos originais, de diferentes tamanhos e orientações, foram normalizados resultando em imagens 16×16 em escala de cinza;
- E, posteriormente, rearranjados em uma coluna de 256 colunas relativa às cores e uma referente ao dígito em questão;
- Note que a variável resposta é categórica (assume somente três valores: 1,2 ou 7);
- Os métodos em competição são:
  - *k-Nearest Neighbors* (K-NN);
  - *Linear discriminant analysis* (LDA);
  - *Quadratic discriminant analysis* (QDA);
  - *Support vector machine* (SVM);
  - Regressão logística.

Preparação dos dados e métodos

- Os dígitos originais, de diferentes tamanhos e orientações, foram normalizados resultando em imagens 16×16 em escala de cinza;
- E, posteriormente, rearranjados em uma coluna de 256 colunas relativa às cores e uma referente ao dígito em questão;
- Note que a variável resposta é categórica (assume somente três valores: 1,2 ou 7);
- Os métodos em competição são:
  - ⋆ *k-Nearest Neighbors* (K-NN);
  - ⋆ *Linear discriminant analysis* (LDA);
  - ⋆ *Quadratic discriminant analysis* (QDA);
  - ⋆ *Support vector machine* (SVM);
  - ⋆ Regressão logística.

Forma de avaliar o desempenho

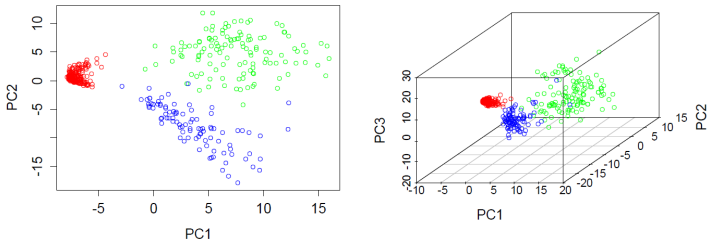
- Utilizamos análise das componentes principais a fim de reduzir a dimensão do espaço;



- Aplicamos o método de validação cruzada (k-dobras, com $k = 4$) para avaliar/comparar o desempenho dos modelos ajustados;
- Particionamos a amostra em k grupos de tamanhos iguais;
- Um grupo é separado para validação e modelo é ajustado para os $k - 1$ demais grupos.

Forma de avaliar o desempenho

- Utilizamos análise das componentes principais a fim de reduzir a dimensão do espaço;



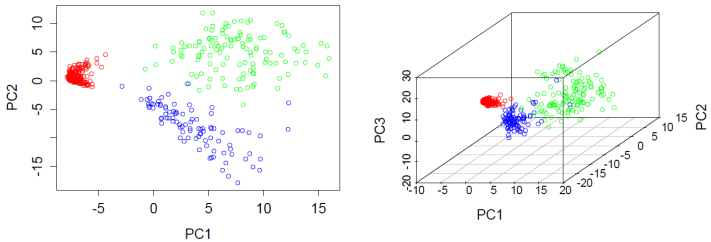
- Aplicamos o método de validação cruzada (k-dobras, com $k = 4$) para avaliar/comparar o desempenho dos modelos ajustados;

- Particionamos a amostra em k grupos de tamanhos iguais;

- Um grupo é separado para validação e modelo é ajustado para os $k - 1$ demais grupos.

Forma de avaliar o desempenho

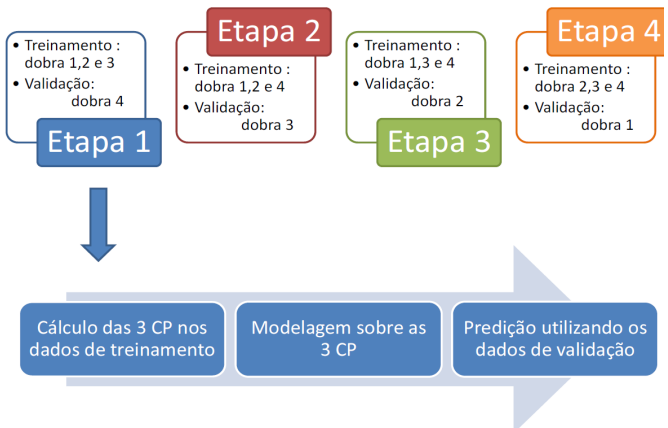
- Utilizamos análise das componentes principais a fim de reduzir a dimensão do espaço;



- Aplicamos o método de validação cruzada (k-dobras, com $k = 4$) para avaliar/comparar o desempenho dos modelos ajustados;

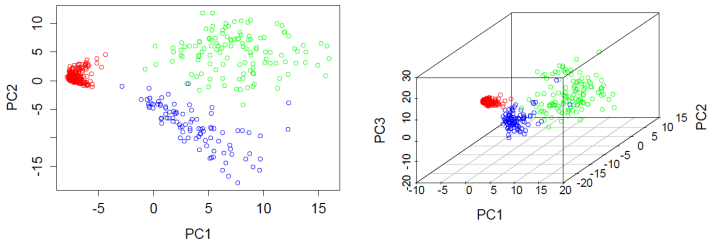
- Particionamos a amostra em k grupos de tamanhos iguais;

- Um grupo é separado para validação e modelo é ajustado para os $k - 1$ demais grupos.



Forma de avaliar o desempenho

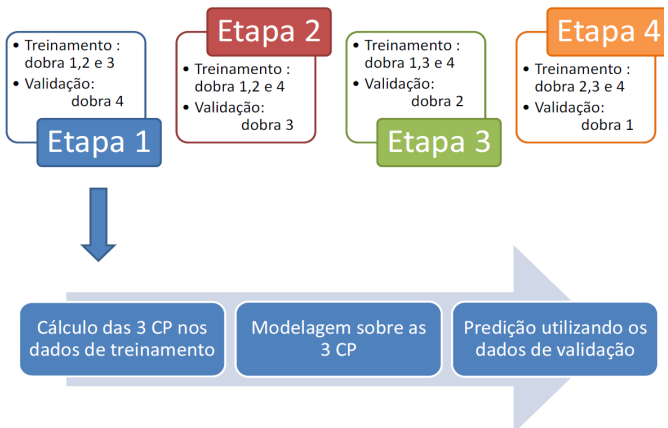
- Utilizamos análise das componentes principais a fim de reduzir a dimensão do espaço;



- Aplicamos o método de validação cruzada (k-dobras, com $k = 4$) para avaliar/comparar o desempenho dos modelos ajustados;

- Particionamos a amostra em k grupos de tamanhos iguais;

- Um grupo é separado para validação e modelo é ajustado para os $k - 1$ demais grupos.



Resultados

- Os modelos de classificação K-NN foram ajustados com valores de $K = 1, 3, 5, 10$ e 15;
- Os modelos de classificação SVM foram ajustados utilizando *kernel* linear, polinomial, sigmoide e radial;
- Para medir a qualidade do ajuste, utilizamos o percentual de **classificações incorretas**;

Resultados

- Os modelos de classificação K-NN foram ajustados com valores de $K = 1, 3, 5, 10$ e 15;
- Os modelos de classificação SVM foram ajustados utilizando *kernel* linear, polinomial, sigmoide e radial;
- Para medir a qualidade do ajuste, utilizamos o percentual de **classificações incorretas**;

Resultados

- Os modelos de classificação K-NN foram ajustados com valores de $K = 1, 3, 5, 10$ e 15;
- Os modelos de classificação SVM foram ajustados utilizando *kernel* linear, polinomial, sigmoide e radial;
- Para medir a qualidade do ajuste, utilizamos o percentual de **classificações incorretas**;

	Treinamento	*Rank*	Validação	*Rank*	Kaggle	*Rank*
Lógistica	0.003	2	0.013	8	0.04	10
LDA	0.023	10	0.023	10	0.04	10
QDA	0.013	9	0.02	9	0.04	10
K-NN (1)	0	1	0.01	7	0.03	7
K-NN (3)	0.005	5	0.008	5	0.01	1
K-NN(5)	0.007	7	0.005	3	0.02	5
K-NN (10)	0.007	7	0.005	3	0.02	5
K-NN(15)	0.009	8	0.008	5	0.03	7
SVM-Linear	0.004	4	0.005	3	0.02	5
SVM-Polinomial	0.048	12	0.055	12	0.06	12
SVM-Sigmoide	0.036	11	0.033	11	0.05	11
SVM-Radial	0.004	4	0.01	7	0.02	5