

Aplicação de Técnicas de Aprendizagem de Máquina utilizando R

Prof. Mário de Noronha Neto

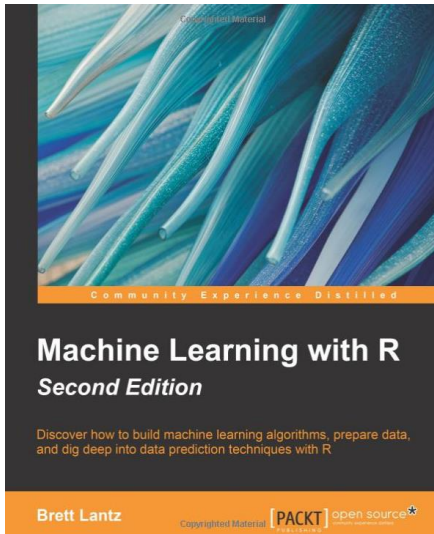
O material utilizado neste curso foi elaborado pelos professores Mario de Noronha Neto (IFSC) e Richard Demos Souza (UFSC)

Tópicos



- Introdução ao aprendizado de máquina
- Gerenciamento e interpretação de dados utilizando o R
- Classificação utilizando a técnica k-NN
- Regressão linear
- Identificação de agrupamentos de dados utilizando a técnica k-means

Material e software utilizados



Definições



Artificial Intelligence

The science behind intelligent machines



Machine Learning

How machines learn - a subset of AI



Deep Learning

Advanced machine learning, using neural networks



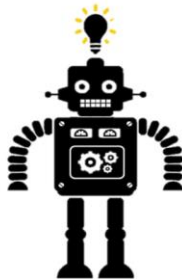
upstartAI.com

Definições



What is Machine Learning?

Machine learning allows computers to learn and infer from data.



*"**Machine learning** is functionality that helps software perform a task without explicit programming or rules." - Google Cloud*

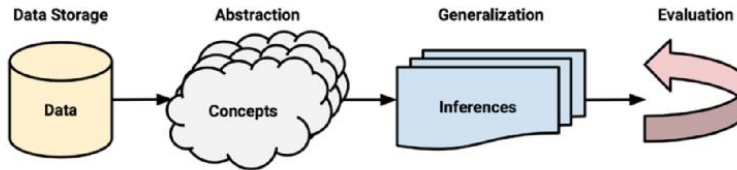
*"The field of study interested in the development of computer algorithms to transform data into intelligent action is known as **machine learning**." Brett Lantz – Machine Learning with R.*

Aplicações



- Reconhecimento de voz (Google...)
- Reconhecimento facial (Facebook...)
- Diagnósticos médicos (Pixeon...)
- Recomendações de filmes (Netflix...)
- Detecção de falhas, fraudes...
- Filtros de spam
- Veículos autônomos
- ...

Como funciona?



Fonte: Brett Lantz - Machine Learning with R – Second Edition

- Armazenamento de dados: Utiliza observação e memórias curtas e longas como base para o raciocínio
- Abstração: Envolve a tradução de dados armazenados em representações e conceitos mais amplos (modelos), como equações matemáticas, diagramas relacionais, etc...
- Generalização: Utiliza os dados abstraídos para gerar conhecimento e inferências que direcionam ações em novos contextos.
- Avaliação: Fornece um mecanismo de retorno para medir a utilidade do conhecimento aprendido e indicar possíveis caminhos para melhorias.

E na prática, como funciona?



1. **Coleta de dados:** Envolve a coleta do material de aprendizagem que um algoritmo utilizará para gerar conhecimento. Na maioria dos casos os dados são combinados em uma única fonte com um arquivo de texto, planilha ou base de dados.
2. **Análise e preparação dos dados:** A qualidade de qualquer projeto de aprendizado de máquina baseia-se amplamente na qualidade de seus dados de entrada. Assim, é importante aprender mais sobre os dados e suas nuances durante uma prática chamada exploração de dados. É necessário trabalho adicional para preparar os dados para o processo de aprendizado.
3. **Treinamento do modelo:** A tarefa específica de aprendizado de máquina escolhida informará a seleção de um algoritmo apropriado, e o algoritmo representará os dados na forma de um modelo.
4. **Avaliação do modelo:** Como cada modelo de aprendizado de máquina resulta em uma solução tendenciosa para o problema de aprendizado, é importante avaliar o quanto o algoritmo aprende com sua experiência.
5. **Melhorias no modelo:** Pode ser necessário mudar completamente o modelo atual para um tipo diferente de modelo. Pode ser necessário complementar seus dados com dados adicionais ou executar trabalhos preparatórios adicionais.

Tipos de dados de entrada

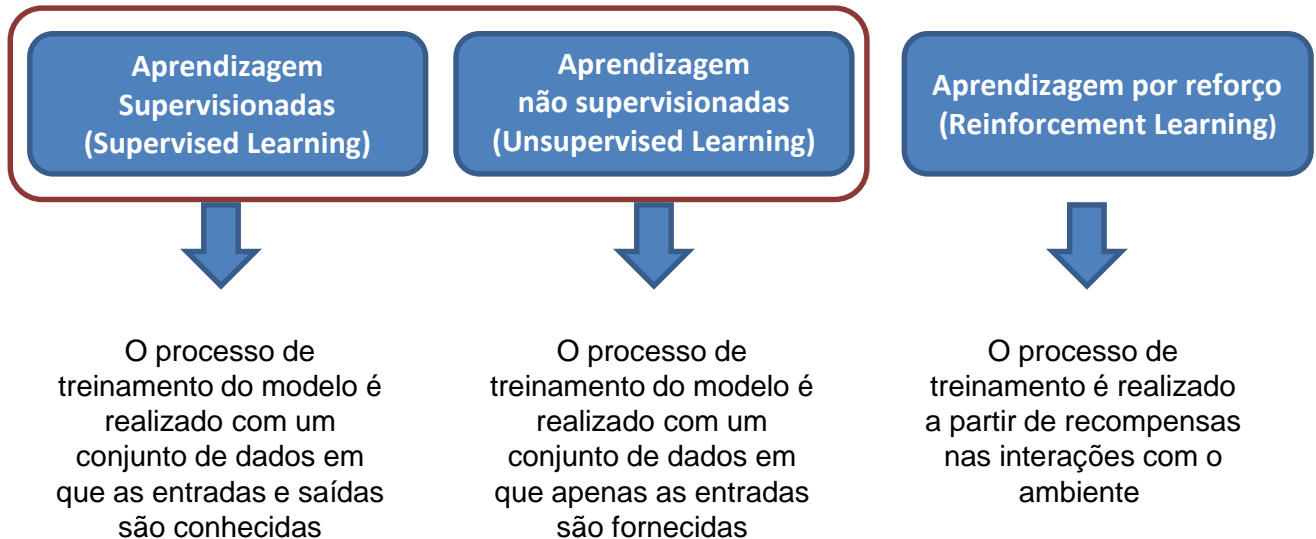
features

year	model	price	mileage	color	transmission
2011	SEL	21992	7413	Yellow	AUTO
2011	SEL	20995	10926	Gray	AUTO
2011	SEL	19995	7351	Silver	AUTO
2011	SEL	17809	11613	Gray	AUTO
2012	SE	17500	8367	White	MANUAL
2010	SEL	17495	25125	Silver	AUTO
2011	SEL	17000	27393	Blue	AUTO
2010	SEL	16995	21026	Silver	AUTO
2011	SES	16995	32655	Silver	AUTO

examples

As variáveis podem ser numéricas, categóricas ou categóricas ordenadas

Alguns tipos de aprendizado de máquina



Técnicas de Aprendizagem Supervisionada abordadas neste curso:

Regressão



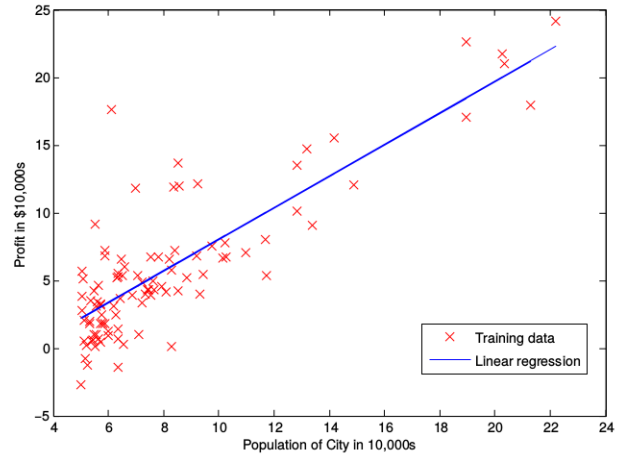
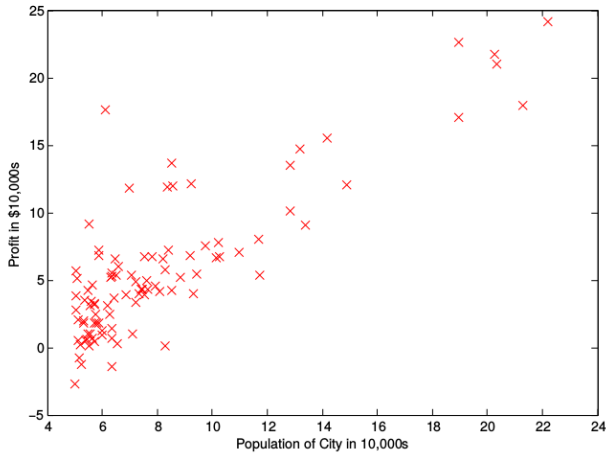
Utilizadas para prever
dados numéricos

Classificação



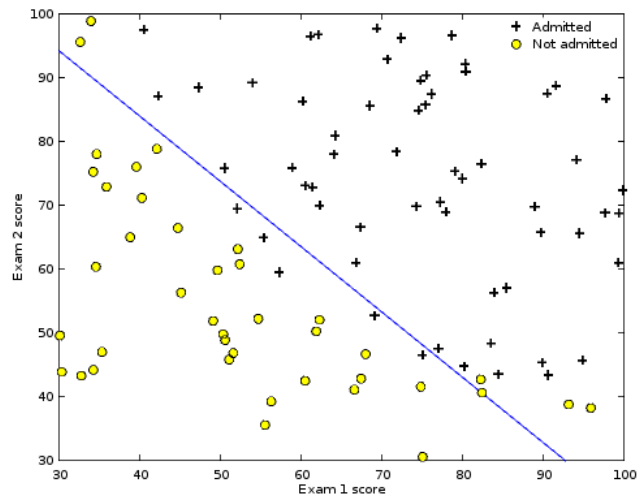
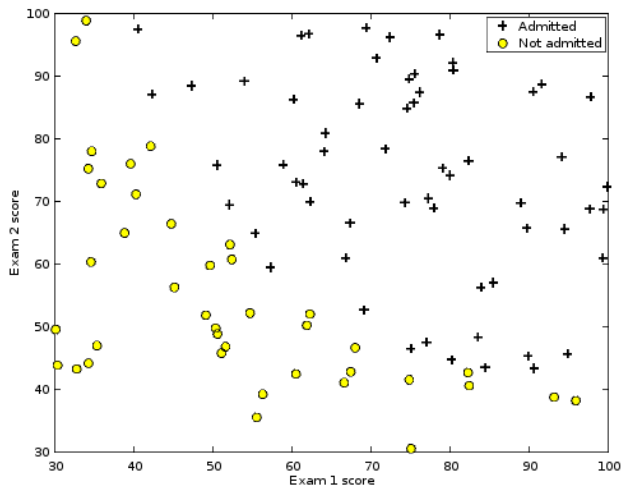
Utilizadas para prever
categorias

Regressão



Fonte: Andrew Ng. Exemplo retirado do curso "Aprendizagem Automática", Stanford University – Coursera.

Classificação



Fonte: Andrew Ng. Exemplo retirado do curso "Aprendizagem Automática", Stanford University – Coursera.

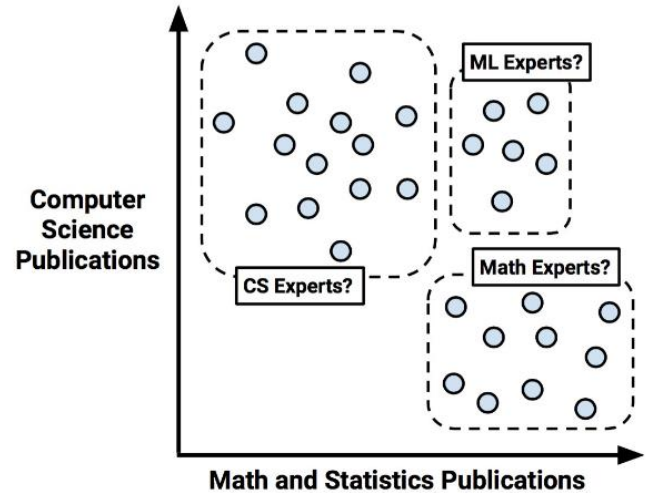
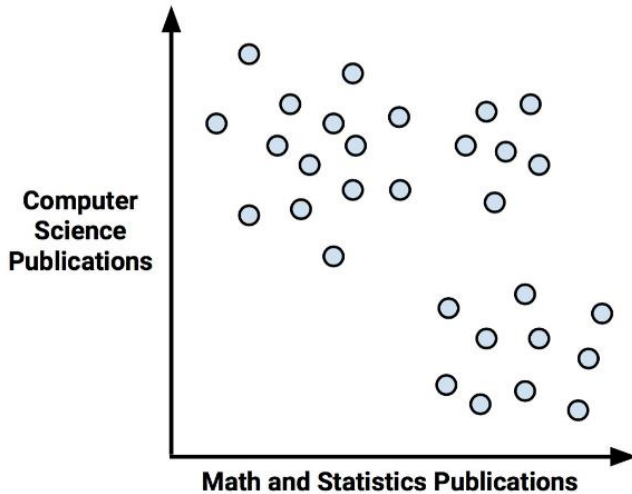
Técnica de Aprendizagem não Supervisionada abordada neste curso

Agrupamento (Clustering)



Utilizada para segmentação
de grupos com algumas
semelhanças

Clustering



Projeto R e RStudio

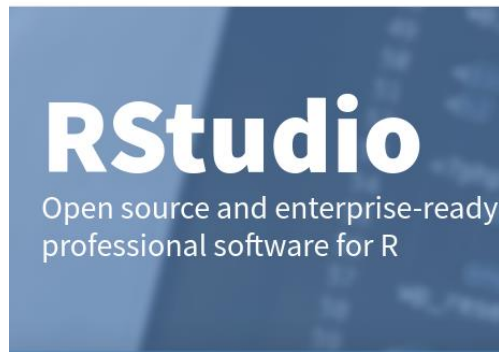


The R Project for Statistical Computing

Site R



 R Studio



Site RStudio



Projeto R e RStudio

Instalação de pacotes: `installed.packages('nome do pacote')`

Files

Plots

Packages

Help

Viewer

Install

Update

Name		Description	Version
System Library			
<input type="checkbox"/>	boot	Bootstrap Functions (Originally by Angelo Canty for S)	1.3-20
<input checked="" type="checkbox"/>	class	Functions for Classification	7.3-14
<input type="checkbox"/>	cluster	"Finding Groups in Data": Cluster Analysis Extended Rousseeuw et al.	2.0.7-1
<input type="checkbox"/>	codetools	Code Analysis Tools for R	0.2-15
<input type="checkbox"/>	compiler	The R Compiler Package	3.4.4
<input checked="" type="checkbox"/>	datasets	The R Datasets Package	3.4.4
<input type="checkbox"/>	foreign	Read Data Stored by 'Minitab', 'S', 'SAS', 'SPSS', 'Stata', 'Systat', 'Weka', 'dBase', ...	0.8-69
<input checked="" type="checkbox"/>	graphics	The R Graphics Package	3.4.4
<input checked="" type="checkbox"/>	grDevices	The R Graphics Devices and Support for Colours and Fonts	3.4.4
<input type="checkbox"/>	grid	The Grid Graphics Package	3.4.4
<input type="checkbox"/>	KernSmooth	Functions for Kernel Smoothing Supporting Wand & Jones (1995)	2.23-15
<input type="checkbox"/>	lattice	Trellis Graphics for R	0.20-35

Carregando os pacotes: `library(nome do pacote)`

Estruturas de dados/Objetos no R

- **Vetores:** Estrutura que armazena uma sequência de valores numéricos ou de caracteres. Todos os elementos devem ser do mesmo tipo (inteiros, caracteres, lógicos, etc...)
- **Fatores:** Um caso especial da estrutura de vetores. São utilizados apenas para representar variáveis categóricas e ordinais
- **Matrizes:** Estrutura bidimensional utilizada para armazenar vetores em linhas e colunas. Todos os vetores devem ser do mesmo tipo
- **Listas:** Estrutura semelhante aos vetores, porém permite armazenar elementos de diferentes tipos e tamanhos
- **Data frame:** Pode ser compreendida com uma lista de vetores ou fatores do mesmo tamanho
- **Funções:** Rotinas criadas para propósitos específicos

Vetores



```
# create vectors of data for three medical patients
subject_name <- c("John Doe", "Jane Doe", "Steve Graves")
temperature <- c(98.1, 98.6, 101.4)
flu_status <- c(FALSE, FALSE, TRUE)

# access the second element in body temperature vector
temperature[2]
[1] 98.6

## examples of accessing items in vector
# include items in the range 2 to 3
temperature[2:3]
[1] 98.6 101.4

# exclude item 2 using the minus sign
temperature[-2]
[1] 98.1 101.4

# use a vector to indicate whether to include item
temperature[c(TRUE, TRUE, FALSE)]
[1] 98.1 98.6
```

Fatores



```
# add gender factor
gender <- factor(c("MALE", "FEMALE", "MALE"))
gender
[1] MALE  FEMALE MALE
Levels: FEMALE MALE

# add blood type factor
blood <- factor(c("O", "AB", "A"),
               levels = c("A", "B", "AB", "O"))
blood
[1] O  AB A
Levels: A B AB O

# add ordered factor
symptoms <- factor(c("SEVERE", "MILD", "MODERATE"),
                  levels = c("MILD", "MODERATE", "SEVERE"),
                  ordered = TRUE)
symptoms
[1] SEVERE  MILD    MODERATE
Levels: MILD < MODERATE < SEVERE

# check for symptoms greater than moderate
symptoms > "MODERATE"
[1] TRUE FALSE FALSE
```

Listas



```
# display information for a patient
> subject_name[1]
[1] "John Doe"
> temperature[1]
[1] 98.1
> flu_status[1]
[1] FALSE
> gender[1]
[1] MALE
Levels: FEMALE MALE
> blood[1]
[1] O
Levels: A B AB O
> symptoms[1]
[1] SEVERE
Levels: MILD < MODERATE < SEVERE
```

Listas



```
# create list for a patient
```

```
subject1 <- list(fullname = subject_name[1],  
                 temperature = temperature[1],  
                 flu_status = flu_status[1],  
                 gender = gender[1],  
                 blood = blood[1],  
                 symptoms = symptoms[1])
```

```
# display the patient
```

```
subject1
```

```
> subject1  
$fullname  
[1] "John Doe"
```

```
$temperature  
[1] 98.1
```

```
$flu_status  
[1] FALSE
```






```
$gender  
[1] MALE  
Levels: FEMALE MALE
```

```
$blood  
[1] O  
Levels: A B AB O
```

```
$symptoms  
[1] SEVERE  
Levels: MILD < MODERATE < SEVERE
```

Listas



Environment	History	Connections
   Import Dataset ▾ 		
 Global Environment ▾		
Data		
<input checked="" type="radio"/> subject1		List of 6
fullname : chr "John Doe"		
temperature: num 98.1		
flu_status : logi FALSE		
gender : Factor w/ 2 levels "FEMALE","MALE": 2		
blood : Factor w/ 4 levels "A","B","AB","O": 4		
symptoms : Ord.factor w/ 3 levels "MILD"<"MODERATE"<...: 3		

Listas



```
## methods for accessing a list

# get a single list value by position (returns a sub-list)
subject1[2]
$temperature
[1] 98.1
# get a single list value by position (returns a numeric vector)
subject1[[2]]
[1] 98.1

# get a single list value by name
subject1$temperature
[1] 98.1

# get several list items by specifying a vector of names
subject1[c("temperature", "flu_status")]
$temperature $flu_status
[1] 98.1      [1] FALSE

## access a list like a vector
# get values 2 and 3
subject1[2:3]
$temperature $flu_status
[1] 98.1      [1] FALSE
```

Data frames

```
# create a data frame from medical patient data

pt_data <- data.frame(subject_name, temperature, flu_status, gender,
                      blood, symptoms, stringsAsFactors = FALSE)

# display the data frame
pt_data
```

Caso não seja especificado,
o R converte todas as
strings para fatores.






```
> pt_data
  subject_name temperature flu_status gender blood symptoms
1   John Doe       98.1      FALSE  MALE    O    SEVERE
2   Jane Doe       98.6      FALSE FEMALE  AB     MILD
3 Steve Graves    101.4       TRUE   MALE    A MODERATE
```

Data frames








```
pt_data <- data.frame(subject_name, temperature, flu_status, gender,  
                      blood, symptoms, stringsAsFactors = FALSE)
```



Environment	History	Connections
  Import Dataset 		
Global Environment 		
Data		
 pt_data 3 obs. of 6 variables		
subject_name: chr "John Doe" "Jane Doe" "Steve Graves"		
temperature : num 98.1 98.6 101.4		
flu_status : logi FALSE FALSE TRUE		
gender : Factor w/ 2 levels "FEMALE","MALE": 2 1 2		
blood : Factor w/ 4 levels "A","B","AB","O": 4 3 1		
symptoms : Ord.factor w/ 3 levels "MILD"<"MODERATE"<...: 3 1 2		

```
pt_data <- data.frame(subject_name, temperature, flu_status, gender,  
                      blood, symptoms, stringsAsFactors = TRUE)
```



Environment	History	Connections
  Import Dataset 		
Global Environment 		
Data		
 pt_data 3 obs. of 6 variables		
subject_name: Factor w/ 3 levels "Jane Doe","John Doe",...: 2 1 3		
temperature : num 98.1 98.6 101.4		
flu_status : logi FALSE FALSE TRUE		
gender : Factor w/ 2 levels "FEMALE","MALE": 2 1 2		
blood : Factor w/ 4 levels "A","B","AB","O": 4 3 1		
symptoms : Ord.factor w/ 3 levels "MILD"<"MODERATE"<...: 3 1 2		

Data frames



```
## accessing a data frame
```

```
# get a single column  
pt_data$subject_name
```

```
# get several columns by specifying a vector of names  
pt_data[c("temperature", "flu_status")]
```

```
# this is the same as above, extracting temperature and flu_status  
pt_data[2:3]
```

```
# accessing by row and column  
pt_data[1, 2]
```

```
# accessing several rows and several columns using vectors  
pt_data[c(1, 3), c(2, 4)]
```

```
> pt_data$subject_name  
[1] John Doe    Jane Doe    Steve Graves  
Levels: Jane Doe John Doe Steve Graves  
> pt_data[c("temperature", "flu_status")]  
  temperature flu_status  
1      98.1      FALSE  
2      98.6      FALSE  
3     101.4       TRUE  
> pt_data[2:3]  
  temperature flu_status  
1      98.1      FALSE  
2      98.6      FALSE  
3     101.4       TRUE  
> pt_data[1, 2]  
[1] 98.1  
> pt_data[c(1, 3), c(2, 4)]  
  temperature gender  
1      98.1    MALE  
3     101.4    MALE
```

Data frames



```
## Leave a row or column blank to extract all rows or columns
```

```
# column 1, all rows
```

```
pt_data[, 1]
```

```
# row 1, all columns
```

```
pt_data[1, ]
```

```
# all rows and all columns
```

```
pt_data[ , ]
```

```
# the following are equivalent
```

```
pt_data[c(1, 3), c("temperature", "gender")]
```

```
pt_data[-2, c(-1, -3, -5, -6)]
```

```
> pt_data[, 1]
```

```
[1] John Doe Jane Doe Steve Graves
```

```
Levels: Jane Doe John Doe Steve Graves
```

```
> pt_data[1, ]
```

	subject_name	temperature	flu_status	gender	blood	symptoms
1	John Doe	98.1	FALSE	MALE	O	SEVERE

```
> pt_data[, ]
```

	subject_name	temperature	flu_status	gender	blood	symptoms
1	John Doe	98.1	FALSE	MALE	O	SEVERE
2	Jane Doe	98.6	FALSE	FEMALE	AB	MILD
3	Steve Graves	101.4	TRUE	MALE	A	MODERATE

```
> pt_data[c(1, 3), c("temperature", "gender")]
```

```
temperature gender
```

```
1 98.1 MALE
```

```
3 101.4 MALE
```

```
> pt_data[-2, c(-1, -3, -5, -6)]
```

```
temperature gender
```

```
1 98.1 MALE
```

```
3 101.4 MALE
```