Solução Lista 01

Nome: Pedro Cavalcante Mazzucca
E-mail: pedro.mazzucca@aluno.ufabc.edu.br
(Não é preciso informar os RAs)

25 fevereiro, 2025

Exercício 01

Problema de classificação

O problema da classificação são situações em que o objetivo é classificar os dados em diferentes categorias. No caso do aprendizado de máquina, se utiliza dados rotulados afim de, para cada dado que futuramente seja inserido, o sistema seja capaz de diferenciar nas categorias treinadas.

Por exemplo, para determinar se uma maçã é boa ou está podre, podemos utilizar imagens de maçãs comestiveís e de maçãs aprodrecendo, e assim criar um sistema capaz de separar maçãs com base em fotos tiradas por uma câmera. Entretanto, caso utilizamos o mesmo sistema, sem nenhuma alteração, para separar pera, o sistema não sabera diferenciar quais peras são saudáveis e quais estão podres.

Uma técnica comum é utilizar o kNN, que analisa os "k" vizinhos mais próximos para separar os dados. Caso o "k" seja muito baixo, o sistema "memoriza" os dados, e conforme o "k" aumenta, ele vai se tornando menos flexivel, se alinhando mais ao comportamento dos dados. # Problema de regressão

O modelo de regressão é utilizado para prever comportamentos de dados, sendo muito utilizado, por exemplo, na bolsa de valores, onde é muito difícil prever se uma ação irá subir ou descer dada a natureza caótica dos dados.

Diferente do problema de classificação, o modelo de regressão é capaz de retormar um resultado válido para dados nunca antes visto, por exemplo, se uma ação tenha um aumento ou uma redução subita, o mesmo modelo pode continuar retornando previsões validas, sem a necessidade de criar um novo sistema novamente.

Assim como no problema de classificação, a partir de técnicas como kNN é possível determinar o comportamento médio dos dados através de seus vizinhos, para assim prever o comportamento de futuros dados. # Problema de agrupamento

O problema de agrupamento de dados consiste em, dado uma enorme quantidade de informações, "separar em categórias distintas" (clustering) ao analisar comportamentos semelhantes. Esse problema não é supervisionado, visto que é o próprio computador que irá definir as categorias, e não um rotulo previamente estabelecido.

Uma das formas mais comuns de resolver esse problema é o "K-Means", que calcula a média entre "k" dados vizinhos, assim juntando dados que estejam muitos próximos na mesma categória. Entretanto, o formato dos dados, junto com a densidade, podem alterar o resultado, sendo necessário ou aumentar a quantidade de categórias separadas para ter resultado mais precisos ou mudar o método para outro que leve em consideração outros fatores, como a densidade.

O problema de agrupamento é extremamente útil para determinar padrões de vendas entre produtos, por exemplo.

Exercício 02

A maldição da dimensionalidade é um problema existente em diversas areas que lidam com dados, dentre elas o aprendizado de máquina.

Para organizar dados, são necessários "dimensões" para separá-los, por exemplo: se queremos determinar se alguém está acima do peso ou não, meramente utilizar o quanto alguém pesa é insuficiente para classificar, e para corrigir isso podemos utilizar o tamanho da pessoa e a idade como parâmetro, aumentado a "dimensão" do nosso sistema, e assim conseguindo resultados mais precisos. Isso ocorre pois o "volume" nos quais os dados podem estar confinados aumenta, e assim ficam mais dispersos do que caso representassemos numa única dimensão.

O problema é que, conforme a dimensão aumenta, fica mais difícil estabelecer estratégias para organiza-los e, por mais que aumentar os parâmetros ajude a prever, utilizar muitas dimensões também começa a ter o efeito oposto, já que os dados começam a ficar dispersos demais afim de separá-los apropriadamente.

Exercício 03

```
library(tidyverse)
library(magrittr)
calculaTudo <- function(k,x,D){
  D2 <- D %>%
    mutate( dist = (x[1] - x_1)^2 + (x[2] - x_2)^2 ) %>%
    arrange( dist ) %>% head(k) %>% count(y)
  return(D2)
}

x_1 = rnorm(100,1,1)
x_2 = rnorm(100,-1,2)
x <- c(x_1, x_2)

Data <- tibble( x_1 = rnorm(100,1,1),
             x_2 = rnorm(100,-1,2),
             y   = factor(sample(c("one","two","three"),100,replace = T)))
head(D)
```

```
##
## 1 function (expr, name)
## 2 .External(C_doD, expr, name)
```

```
print(calculaTudo(10,x,Data))
```

```
## # A tibble: 3 x 2
##   y         n
##   <fct> <int>
## 1 one       5
## 2 three     4
## 3 two       1
```

Exercício 04

```
library(tidyverse)
library(magrittr)

data("iris") # Carrega o banco no ambiente global
iris <- as_tibble(iris) %>%  # Converte para a dataframe tibble
  select(Petal.Length,Sepal.Length,Species) %>% # Seleciona colunas da dataframe
  rename( x_1 = Petal.Length, x_2 = Sepal.Length, y = Species) # Renomeia as colunas

head(iris)
```

```
## # A tibble: 6 x 3
##     x_1   x_2 y
##   <dbl> <dbl> <fct>
## 1   1.4   5.1 setosa
## 2   1.4   4.9 setosa
## 3   1.3   4.7 setosa
## 4   1.5   4.6 setosa
## 5   1.4   5   setosa
## 6   1.7   5.4 setosa
```

```
calculaTudo <- function(k,x,D){
  D2 <- D %>%
    mutate( dist = (x[1] - x_1)^2 + (x[2] - x_2)^2 ) %>%
    arrange( dist ) %>% head(k) %>% count(y)
  p=D #temp
  return(D2)
}


x_1 = rnorm(100,1,1)
x_2 = rnorm(100,-1,2)
x <- c(x_1, x_2)
##

print(calculaTudo(10,x,iris))
```

```
## # A tibble: 1 x 2
##   y          n
##   <fct>  <int>
## 1 setosa    10
```

```
print(calculaTudo(1,x,iris))
```

```
## # A tibble: 1 x 2
##   y          n
##   <fct>  <int>
## 1 setosa     1
```