Jornal LNCC - O Pacote CleanerR

Rafael Silva Pereira
Dr Fábio A. M. Porto

Fevereiro 2019

1 Sobre o Autor

Rafael Silva Pereira

Graduado com o titulo de Bacharel em Física com enfase em Física Computacional pela Universidade Federal Fluminense no campus de Volta Redonda e estudante da Pós Graduação em Modelagem Computacional na categoria de aluno do mestrado.

Ingressou em setembro de 2018 na pós graduação do LNCC. É orientado pelo professor Dr Fábio André Machado Porto, se especializando na área de ciência de dados, focando na parte de estatística e aprendizado de maquina.

2 A linguagem R

Desenvolvida a partir da linguagem S, R foi criado como uma linguagem para se tratar principalmente de operações estatísticas sobre diferentes estruturas de dados de forma simples. Seu código fonte se utiliza de C/C++, Fortran e R.

A linguagem está distribuída como software livre através da licença GNU General Public License, e se torna um competidor direto em relação a linguagem python para projetos de ciência de dados.

3 Problema a ser resolvido

Um problema comum que observei em discussões na área é o fato de que existem varias fontes de dados cujos datasets apresentam vários elementos faltantes.

Então sempre ocorre uma discussão do que fazer com as linhas em que isto ocorre. Observando na literatura, vemos que quando a informação a ser completada é numérica e continua, existem vários métodos diferentes sugeridos para preenche-los.

Mas além de variáveis continuas temos também variáveis categóricas, que podem ser representadas até por strings, por exemplo, e então o que fazer com estas variáveis?

4 Solução proposta

A fim de resolver este problema desenvolvi uma série de funções que englobei em um pacote da linguagem R chamado cleanerR.

Os pacotes da linguagem R se comportam como bibliotecas em algumas outras linguagens para os não familiarizados com o termo.

Este se utiliza basicamente de dois conceitos de áreas diferentes para tratar o preenchimento dos dados, que serão discutidos a seguir

4.1 Dependências Funcionais

Considere por exemplo que possuamos um atributo que tem valores $V = \{V_1, V_2, ..., V_n\}$. E possuamos M possíveis atributos que podem ajudar a descrevê-lo.

Seja K $\subset$ M tal que $K = \{M_1, M_2, .., M_J\}$.

Se pudermos definir que para cada possível tupla de K, existe um único valor V_i associado, dizemos que existe uma dependência funcional entre K e V uma vez que sabendo K retornamos V com o máximo possível de certeza.

E se a dependência funcional não existir?

Então definimos o conceito de dependência quase funcional onde existe um subconjunto de V que tem dependência funcional e traçamos uma PDF para os valores restantes. Para explicar melhor como escolher dependências aproximadamente funcionais nos baseamos na subseção seguinte.

4.2 Teoria da informação

Em teoria da informação, utilizando-se da definição de Entropia definido por Shannon $S = -\sum_i P_i log(P_i)$ podemos formular a seguinte hipótese:

- Dado coluna objetivo O_b que queremos prever os valores faltantes consideremos o seguinte:

  - Todos os possíveis valores de O_b caso este seja categórico já apareceram ao menos uma vez.
  - A distribuição dos valores atuais é representativa da distribuição final quando se utiliza o máximo de informação.

- Utilizando-se destas hipóteses temos que o máximo da entropia desta coluna é dada por esta distribuição.
- Então podemos verificar que se usarmos outra coluna associada a O_b podemos, a partir da PDF original, transforma-la em N distribuições, onde o somatório das entropias destas é menor que a entropia de O_b
- Caso utilizemos o subconjunto de V que geraria dependência funcional, converteríamos a distribuição de O_b com N_v possíveis valores tal que $\sum_{i=1}^{N_v} P_i = 1$ em N_v distribuições δ, e como sabemos esta tem entropia miníma.
- Não encontrando dependência funcional podemos verificar que subconjunto de V minimiza a entropia.

5 Implementação

Baseado nestes conceitos, implementei no laboratório DEXL o pacote cleanerR que dada a coluna objetivo analisa todas as combinações de tamanho dado por input na função e é capaz de definir o subconjunto que maximiza a informação sobre a coluna objetivo. Existem também opções para preencher o data frame, e ainda dado este conjunto de vetores verificar qual a acurácia média, melhor e pior caso. para usuários mais experientes na linguagem R é possível retornar todos os candidatos dada uma faixa de erro.

O pacote cleanerR então possui uma versão em desenvolvimento mantida no github, alem de um shinyApp que permite utilizar varias de suas features com uma interface gráfica

No dia 31/01/2019 o pacote foi aceito no CRAN que é o repositório oficial da linguagem R após passar pelo processo de review e pode ser instalado por meios oficiais via install.packages

O manual para utilização do pacote pode ser obtido na pagina oficial deste no CRAN

`https://cran.r-project.org/web/packages/cleanerR/index.html`

Para indivíduos que queiram utilizar se do shinyApp pode se utilizar se do comando:

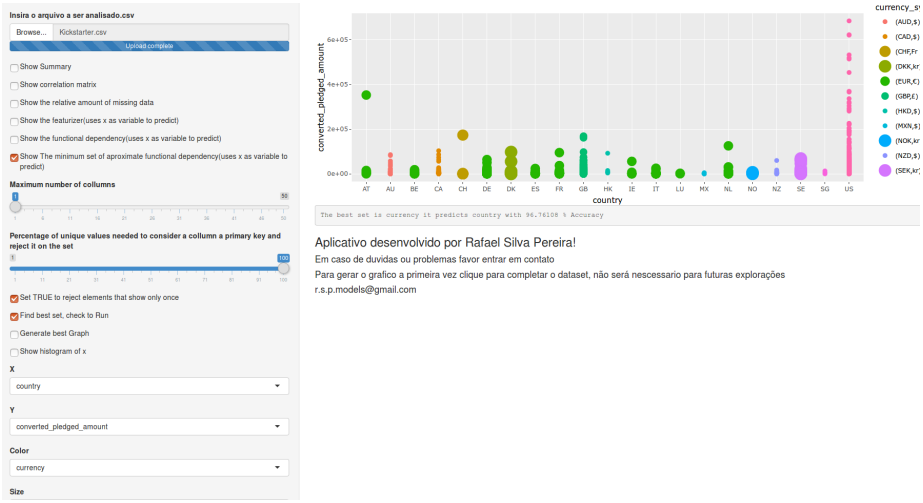

Figura 1: Análise de Dados com o cleanerR

```
runGitHub("CSV_DATA_Analysis","R-S-P-MODELS",launch.browser=TRUE)
```

Ele requer os pacotes shiny, ggplot2, plyr e plotly, o código do cleanerR está escrito dentro do arquivo do APP também, o App permite alem das funções do cleanerR automatizar o processo de visualização dos dados e verificar correlações entre os dados independente da classe destes.

6 Visualização das funções do cleanerR

Na Figura 1, pode-se observar o shiny App citado funcionando. Utilizamos neste exemplo um dataset com informações sobre projetos do Kickstarter, considerando as seguintes visualizações:

- eixo X: país de origem de um projeto
- eixo Y: quantidade de dolares que um projeto obteve
- cor dos pontos: moeda associada a este pais
- tamanho dos pontos: símbolo associado à moeda

Adicionalmente, utilizamos o método de dependência funcional para indicar o melhor conjunto de atributos de tamanho 1 que seja capaz de predizer os valores na coluna Pais. A função retornou o atributo *moeda* com 96% de acurácia .

o Aplicativo fornece ainda outras funcionalidades: o resumo do dataset, a matriz de correlação completa, ou seu corte para a variável de interesse, ou ainda a proporção de valores nulos do seu dataset. Finalmente, pode-se descobrir dado um ensemble de modelos estatísticos qual faz o melhor ajuste de curva, considerando-se duas possíveis métricas.