Desafio Módulo Analytics

Este desafio é construído para simular o fim a fim de um processo de ciência de dados: análise exploratória dos dados, aplicação de modelos de predição e seleção de melhor modelo.

A análise exploratória pode ser feito tanto em R quanto em Python. A parte de modelos de aprendizado de máquina deve ser feito utilizando a biblioteca sklearn do Python.

A interpretação dos resultados têm mais peso na nota do que o código ou a acurácia do modelo. Ao longo do desafio, escreva a sua interpretação dos passos tomados e resultados obtidos.

O desafio deve ser entregue individualmente, mas a discussão em dupla, trio ou grupo é bem-vinda. Isso significa que o código pode estar parecido entre as pessoas, mas a interpretação deve ser feita com as próprias palavras.

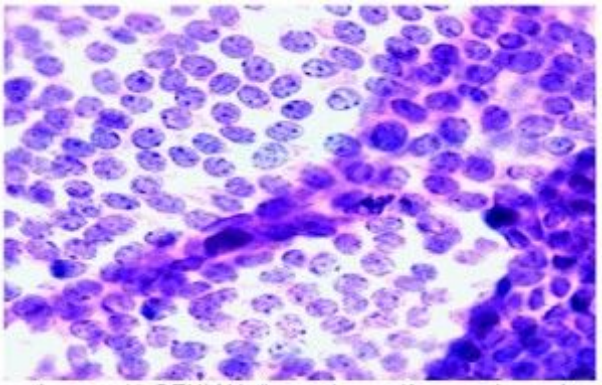
A aula de sexta-feira (10/08) terá espaço para perguntas. Dúvidas podem ser enviadas ao meu e-mail (afonso.menegola@keyrus.com.br). Entretanto, aproveite o desafio para aprender a pesquisar as dúvidas desta área na internet, existem tutoriais/materiais excelentes para aprender.

O desafio deverá ser entregue **até as 06:00 do dia 14/08**.

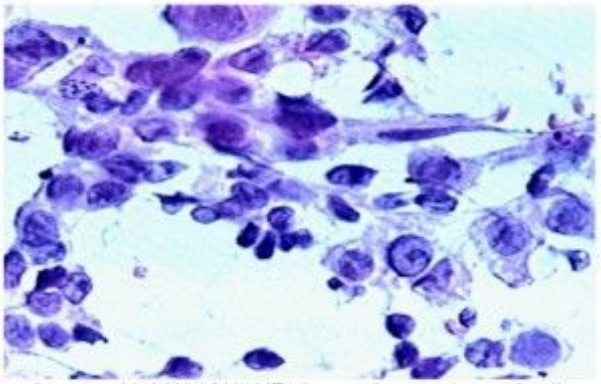
Descrição do dataset

Use os dados Breast Cancer Wisconsin (Diagnostic) Data Set do UCI Machine Learning Repository. O arquivo se encontra neste link.

O dataset consiste em 569 medições realizadas em células da mama, com o indicativo se era um tumor Maligno (M) ou Benigno (B).



Smear with BENIGN diagnosis – uniform nucleus of cells, symmetrical, homogeneous, with areas within normal size



Smear with MALIGNANT diagnosis – nucleus of cells without uniformity, asymmetrical, not homogeneous (multiple sizes) and with areas above normal size

Informações sobre as variáveis:

1) id - id do tumor
2) diagnosis (M = maligno, B = benigno)

10 variáveis de ponto flutuante foram medidas para cada núcleo de célula:

a) radius (média das distâncias do centro até o perímetro)
b) texture (desvio padrão da luminosidade da célula)
c) perimeter
d) area
e) smoothness (variação local dos comprimentos dos rais)
f) compactness (perimeter^2 / area - 1.0)
g) concavity (severidade das partes côncavas do contorno da célula)
h) concave points (número de regiões côncavas presentes na célula)
i) symmetry
j) fractal dimension ("coastline approximation" - 1)

Note que temos 30 variáveis disponíveis. As 10 medidas descritas acima foram realizadas diversas vezes na mesma célula. Para resumir as informações de todas as medidas em uma linha do dataset, para cada célula, os pesquisadores tiraram a média das medições (colunas 3 a 12), o desvio padrão das medições (colunas 13 a 22) e o "worst" (a média dos três maiores valores de cada medição - colunas 23 a 32)

Desafio:

1. Realize análise exploratória dos dados.
    a. Faça uma análise descritiva das colunas (média, mediana, std...)
    b. Plote gráficos de dispersão, histogramas, e boxplots.
    c. Existem dados faltantes?
    d. É possível encontrar outliers através dos boxplots?
        i. Indique quais são e remova-os caso julgar necessário
    e. Fique livre para realizar quaisquer outras análises
2. Use validação cruzada para avaliar qual dos algoritmos tem maior acurácia nos dados. Decida quantos folds você usará nos experimentos (não há resposta correta aqui, apenas alternativas)
    a. SVC(kernel='linear')
    b. SVC(kernel='rbf')
    c. KNeighborsClassifier()
    d. RandomForestClassifier()
    e. MLPClassifier()
3. Faça a normalização dos dados utilizando StandardScaler(). Houve melhora no resultado?

4. Utilize a função GridSearchCV() para encontrar os melhores hiperparâmetros dos algoritmos (acesse o link e implemente o código do primeiro exemplo):
    a. No caso de SVM(SVC):
        i. C: [‘0.01’,’0.1’,’1.0’,10’’]
        ii. kernel: [‘linear’,’rbf’]
    b. No caso de KNeighbors:
        i. k: [‘1’,’5’,’10’,’20’,’50’]
    c. No caso de RandomForest:
        i. Leia a descrição dos parâmetros n_estimators, max_features, max_depth e min_samples_split na documentação. Perceba qual é o “default” de cada parâmetro. Explique os parâmetros.
        ii. Escolha dois parâmetros para aplicar no GridSearchCV()
    d. No caso de MLPClassifier:
        i. hidden_layer_sizes: [‘(10,)’,’(30,0)’,’(100,)’,’(300,)’]
    e. Reporte as acurácias médias de cada algoritmo e os melhores valores dos hiperparâmetros