Classificação de Remédio através da utilização de Algoritmos Supervisionados

Lucas de Jesus Moreira dos Santos
Universidade Federal de São Paulo
São José dos Campos, São Paulo,Brasil
moreira.lucas@unifesp.br

Palavras Chave — Algoritmo Supervisionado, Classificação, Rémedios, Base de Dados

I. INTRODUÇÃO

Um diagnóstico é aquilo que pertence ou que se refere à diagnose. Este termo, por sua vez, refere-se à ação e ao efeito de diagnosticar (recolher e analisar dados para avaliar problemas de diversa natureza).

Na medicina, um diagnóstico é o ato de determinar e conhecer a natureza de uma doença pela observação dos seus sintomas e sinais. Também corresponde ao nome com que o médico qualifica a doença de acordo com os sinais detectados, isto é, é o nome dado à conclusão em si mesma.

Através do Diagnóstico, o médico é capaz de introduzir um medicamento ou um tratamento para seus pacientes.

No brasil segundo o ministério da saúde, foram realizadas cerca de 1,4 bilhão de consultas médicas pelo Sistema Único de Saúde (SUS) no período de um ano.

Durante o período de um ano o total de recursos investidos em ações e serviços públicos de saúde é de R$ 92,2 bilhões, será que os profissionais da área da tecnologia, podem de alguma forma ajudar os profissionais da saúde.

O presente trabalho visa através da coleta de alguns dados do paciente, realizar o diagnóstico médico, a fim de receitar algum medicamento para essas pessoas, assim otimizando o trabalho dos médicos. Fazendo com que eles foquem sua atenção em casos mais complexos.

Para realizar esse procedimento de análise e diagnóstico de algum medicamento para esses pacientes, iremos utilizar o conceito de Machine Learning,

O Aprendizado de Máquina ou Machine Learning, é um sistema que pode modificar seu comportamento autonomamente tendo como base a sua própria experiência. Fazendo com que a interferência humana seja mínima.

Os algoritmos de Machine Learning são subdivididos entre Algoritmos Supervisionados e não Supervisionados.

Os Algoritmos Supervisionados funcionam da seguinte forma, dado um conjunto de dados rotulados que já sabemos qual é a nossa saída correta e que deve ser semelhante ao conjunto, tendo a ideia de que existe uma relação entre a entrada e a saída.

Problemas de aprendizagem supervisionados são classificados em problemas de “regressão” e “classificação”. Em um problema de regressão, estamos tentando prever os resultados em uma saída contínua, o que significa que estamos a tentando mapear variáveis de entrada para alguma função contínua. Em um problema de classificação, estamos tentando prever os resultados em uma saída discreta. Em outras palavras, estamos tentando mapear variáveis de entrada em categorias distintas.

Os algoritmos não supervisionados, por outro lado, nos permitem abordar problemas com pouca ou nenhuma ideia do que nossos resultados deve se aparentar. Podemos derivar estrutura de dados onde nós não necessariamente saberíamos o efeito das variáveis.

Podemos derivar essa estrutura, agrupando os dados com base em relações entre as variáveis nos dados. Também pode ser usada para reduzir o número de dimensões em um conjunto de dados para concentrar somente nos atributos mais úteis, ou para detectar tendências.

Com aprendizagem não supervisionada não há feedback com base nos resultados da previsão, ou seja, não há professor para corrigi-la.

No presente trabalho iremos utilizar a técnica de algoritmos Supervisionados de classificação, já que dadas algumas características dos paciente, já existem Remédios que foram receitados. Utilizaremos esses dados para treinar nossos algoritmos.

Alguns trabalhos nesse sentido de classificar medicamentos já foram desenvolvidos, porém nenhum deles conseguiram atingir 100% de acerto nos testes.

A área de saúde é uma área muito delicada e precisamos obter o máximo de acertos para que e a técnica seja colocada

em prática, veremos que com a técnica certa conseguimos um nível de acerto nos testes de 100%.

II. Trabalhos Relacionados

A. O primeiro trabalho relacionado foi da Dulcie Jackson ela propôs classificar o mesmo Dataset utilizando um algoritmo chamado OneVsRestClassifier, esse algoritmo conseguiu um bom score 0.93, porém nesse trabalho usando Random Forest, veremos que ele se saiu melhor.

B.O segundo trabalho foi do Raghu, ele utilizou outro Dataset o Mushroom Classification, esse data set possuí apenas duas classes cogumelos comestíveis ou venenosos, ele aplicou alguns algoritmos de classificação, o que obteve o melhor score foi o KNN 0.9307, o trabalho dele é interessante, pois ele aplicou 6 algoritmos diferentes para classificar.

C. O terceiro trabalho foi o do Elie Kawerk, ele utilizou o Dataset Glass, dado uma série de característica o algoritmo tem que classificar o tipo de vidro, ele utilizou 7 algoritmos o que obteve o melhor resultado foi o Random Forest com um Score de 0.7819.

D. O quarto trabalho foi da própria pessoa que publicou o Dataset o Pratham Tripathi, ele propôs classificar os tipos de Remédio utilizando o algoritmo Decision Three, ele conseguiu um Score de 0.975, um Score muito bom.

E. O quinto e o ultimo trabalhado do Philip Purwoko, ele utilizou a técnica SVM, com os parâmetros que ele passou o resultado não saiu satisfatório o score foi de 0.4082, ele então usou o módulo GridSearch do Sklearn que testa todos os parâmetros e traz o melhor resultado, isso fez com que o desempenho diminuísse, porém o score aumentou para 0.9333.

III. Metodologia

O trabalho utilizara informações de alguns pacientes que tiveram suas características como Idade, Sexo, Pressão Arterial, Colesterol, Proporção de Sódio e Potássio no sangue e o tipo de Remédio inseridas em um banco de dados.

Para fazer a análise desses dados utilizaremos o Jupyter Notebook, em conjunto com algumas bibliotecas como a Pandas para fazer a leitura dos dados, a Matplotlib para visualizar os dados e a Sklearn que é uma biblioteca que já tem implementada uma serie de algoritmos de Machine Learning , sendo necessário só importar os módulos que contenham os algoritmos e as métricas que fazem a partição entre treino e teste.

Para conseguir colocar em prática os algoritmos KNN, SVM, E RD foi necessário tratar os dados, pois as informações das colunas sexo, Nível de Pressão Arterial, Colesterol e o Remédio foram Gravados na forma de Strings e precisamos transforma essas informações em inteiros para os algoritmos realizarem as contas.

Na própria Biblioteca do Sklearn existe um modulo que faz o processamento e tratamento dos dados. As Imagens a seguir são informações dos pacientes antes do processamento dos dados e após o processamento de dados

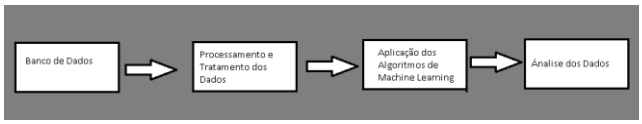
Fig.1- Variáveis do Banco de Dados na sua forma Original

```
Age
[23 47 28 61 22 49 41 60 43 34 74 50 16 69 32 57 63 48 33 31 39 45 18 6
 53 46 15 73 58 66 37 68 67 62 24 26 40 38 29 17 54 70 36 19 64 59 51 4
 56 20 72 35 52 55 30 21 25]

Sex
['F' 'M']

BP
['HIGH' 'LOW' 'NORMAL']

Cholesterol
['HIGH' 'NORMAL']

Na_to_K
[25.355 13.093 10.114  7.798 18.043  8.607 16.275 11.037 15.171 19.368
 11.767 19.199 15.376 20.942 12.703 15.516 11.455 13.972  7.298 25.974
 19.128 25.917 30.568 15.036 33.486 18.809 30.366  9.381 22.697 17.951
  8.75   9.567 11.014 31.876 14.133  7.285  9.445 13.938  9.709  9.084
 19.221 14.239 15.79  12.26  12.295  8.107 13.091 10.291 31.686 19.796
 19.416 10.898 27.183 18.457 10.189 14.16  11.34  27.826 10.091 18.703
 29.875  9.475 20.693  8.37  13.303 27.05  12.856 10.832 24.658 24.276
 13.967 19.675 10.605 22.905 17.069 20.909 11.198 19.161 13.313 10.84
 13.934  7.761  9.712 11.326 10.067 13.935 13.597 15.478 23.091 17.211
 16.594 15.156 29.45  29.271 15.015 11.424 38.247 25.395 35.639 16.725
 11.871 12.854 13.127  8.966 28.294  8.968 11.953 20.013  9.677 16.85
  7.49   6.683  9.17  13.769  9.281 18.295  9.514 10.103 10.292 25.475
 27.064 17.206 22.456 16.753 12.495 25.969 16.347  7.845 33.542  7.477
 20.489 32.922 13.598 25.786 21.036 11.939 10.977 12.894 11.343 10.065
  6.269 25.741  8.621 15.436  9.664  9.443 12.006 12.307  7.34   8.151
  8.7   11.009  7.261 14.642 16.724 10.537 11.227 22.963 10.444 12.923
 10.443  9.945 12.859 28.632 19.007 26.645 14.216 23.003 11.262 12.879
 10.017 17.225 18.739 12.766 18.348 10.446 19.011 15.969 15.891 22.818
 13.884 11.686 15.49  37.188 25.893  9.849 10.403 34.997 20.932 18.991
  8.011 16.31   6.769 34.686 11.567  9.894 14.02  11.349]

Drug
['DrugY' 'drugC' 'drugX' 'drugA' 'drugB']
```

Fig.2 – Banco de Dados após transformar Strings em Inteiros

```
Age
[23 47 28 61 22 49 41 60 43 34 74 50 16 69 32 57 63 48 33 31 39 45 18 65
 53 46 15 73 58 66 37 68 67 62 24 26 40 38 29 17 54 70 36 19 64 59 51 42
 56 20 72 35 52 55 30 21 25]

Sex
[0 1]

BP
[0 1 2]

Cholesterol
[0 1]

Na_to_K
[25.355 13.093 10.114  7.798 18.043  8.607 16.275 11.037 15.171 19.368
 11.767 19.199 15.376 20.942 12.703 15.516 11.455 13.972  7.298 25.974
 19.128 25.917 30.568 15.036 33.486 18.809 30.366  9.381 22.697 17.951
  8.75   9.567 11.014 31.876 14.133  7.285  9.445 13.938  9.709  9.084
 19.221 14.239 15.79  12.26  12.295  8.107 13.091 10.291 31.686 19.796
 19.416 10.898 27.183 18.457 10.189 14.16  11.34  27.826 10.091 18.703
 29.875  9.475 20.693  8.37  13.303 27.05  12.856 10.832 24.658 24.276
 13.967 19.675 10.605 22.905 17.069 20.909 11.198 19.161 13.313 10.84
 13.934  7.761  9.712 11.326 10.067 13.935 13.597 15.478 23.091 17.211
 16.594 15.156 29.45  29.271 15.015 11.424 38.247 25.395 35.639 16.725
 11.871 12.854 13.127  8.966 28.294  8.968 11.953 20.013  9.677 16.85
  7.49   6.683  9.17  13.769  9.281 18.295  9.514 10.103 10.292 25.475
 27.064 17.206 22.456 16.753 12.495 25.969 16.347  7.845 33.542  7.477
 20.489 32.922 13.598 25.786 21.036 11.939 10.977 12.894 11.343 10.065
  6.269 25.741  8.621 15.436  9.664  9.443 12.006 12.307  7.34   8.151
  8.7   11.009  7.261 14.642 16.724 10.537 11.227 22.963 10.444 12.923
 10.443  9.945 12.859 28.632 19.007 26.645 14.216 23.003 11.262 12.879
 10.017 17.225 18.739 12.766 18.348 10.446 19.011 15.969 15.891 22.818
 13.884 11.686 15.49  37.188 25.893  9.849 10.403 34.997 20.932 18.991
  8.011 16.31   6.769 34.686 11.567  9.894 14.02  11.349]

Drug
[0 3 4 1 2]
```

Fig.3 – Fluxograma das Etapas a serem realizadas



Para utilizarar o algoritmo KNN precisamos definir o valor de k, que será a vizinhança, para achar o melhor valor de k foi criado um laço que varia de 1 a 20, a cada rodada o score do treinamento é armazenado em um vetor. Depois disso nós utilizamos o maior valor de K para ser implementado. Garatindo assim o maior valor possível.

Para o algoritmo SVM, utilizamos o método GRIDSEARCH para acharmos o melhor valor, esse método acaba por usar mais recursos do computador, porém consegue encontrar as melhores pontuações.

Abreviações

KNN – K-Nearest Neighbors

RD – Random Forest

SVM – Support Vector Machine

BP – Pressão Arterial

Na_to_K – Proporção Sódio Potássio no Sangue

IV. ANÁLISE EXPERIMENTAL

- Conjunto de Dados

Como já foi comentado o Dataset Drugs, possuí um total de 6 colunas e 200 amostras.

Fig4 – Dataset

```
Database.head()
```

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	DrugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL	HIGH	7.798	drugX
4	61	F	LOW	HIGH	18.043	DrugY

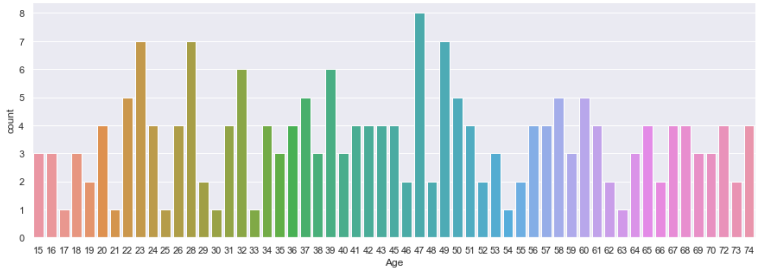
Fig5 – Média, Maiores e Menores valores, frequências.

```
Database.describe(include = 'all')
```

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
count	200.000000	200	200	200	200.000000	200
unique	NaN	2	3	2	NaN	5
top	NaN	M	HIGH	HIGH	NaN	DrugY
freq	NaN	104	77	103	NaN	91
mean	44.315000	NaN	NaN	NaN	16.084485	NaN
std	16.544315	NaN	NaN	NaN	7.223956	NaN
min	15.000000	NaN	NaN	NaN	6.269000	NaN
25%	31.000000	NaN	NaN	NaN	10.445500	NaN
50%	45.000000	NaN	NaN	NaN	13.936500	NaN
75%	58.000000	NaN	NaN	NaN	19.380000	NaN
max	74.000000	NaN	NaN	NaN	38.247000	NaN

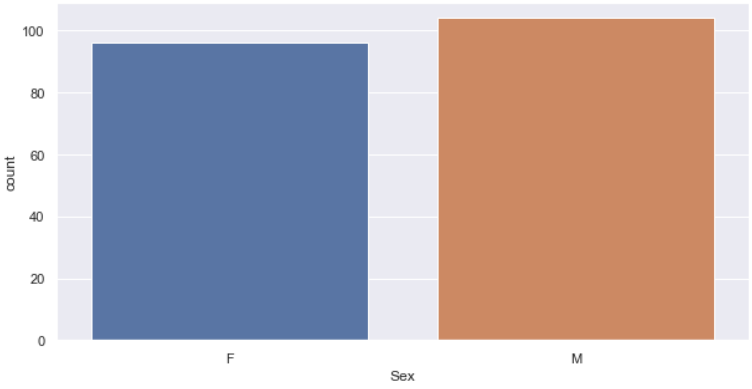
- Gráfico com algumas variáveis.

Fig6 - Variável Idades



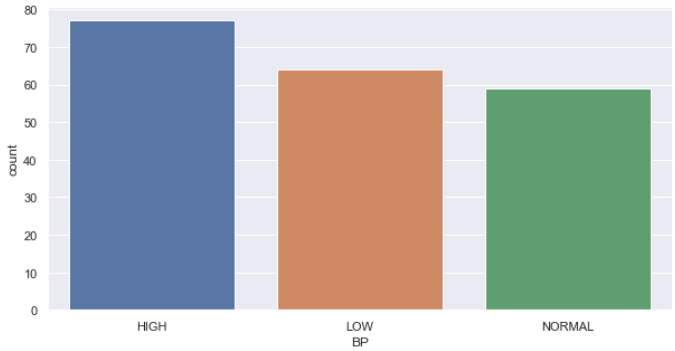
Maior Idade = 74
Menor Idade = 15
Média das Idades = 44

Fig7 - Variável Sexo



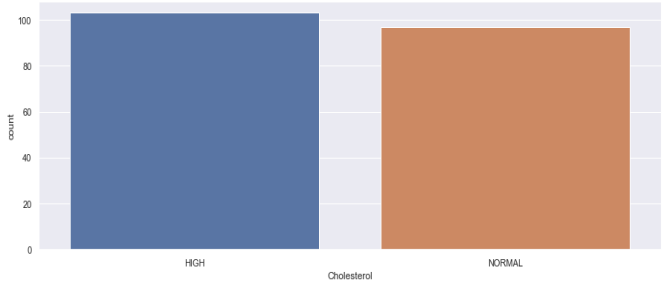
Masculino = 104
Feminino = 96

Fig8 - Variável Pressão Arterial



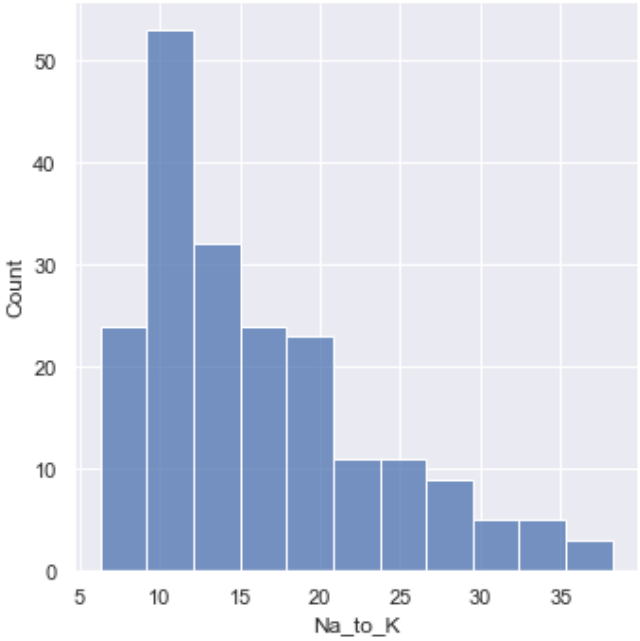
Alta = 77
Baixa = 64
Normal = 59

Fig9 - Variável Colesterol



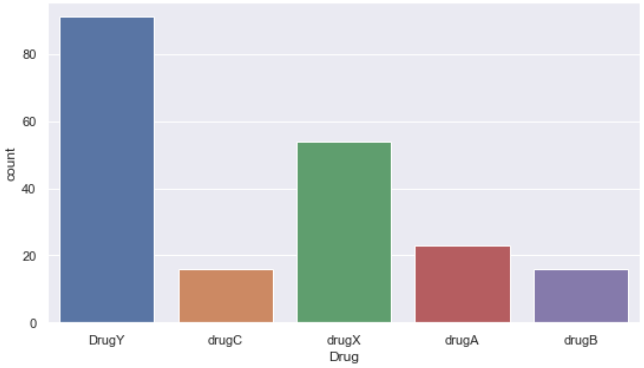
Alto = 103
Normal = 97

Fig10 - Variável Proporção de Sódio e Potássio



Maior Proporção de Sódio e Potássio = 38.247
Menor Proporção de Sódio e Potássio = 6.269
Média = 16.084

Fig11 - Variável Tipo de Medicamento



Remédio Y = 91
Remédio C = 16
Remédio X = 54
Remédio A = 23
Remédio B = 16

- Configuração do algoritmo e do ambiente computacional

Configurações de Hardware:

Processador Interl® Core™ I7 – 7700HQ CPU @ 2.80Ghz
Memória RAM 8GB
Placa de Vídeo GTX-1050 TI 4GB

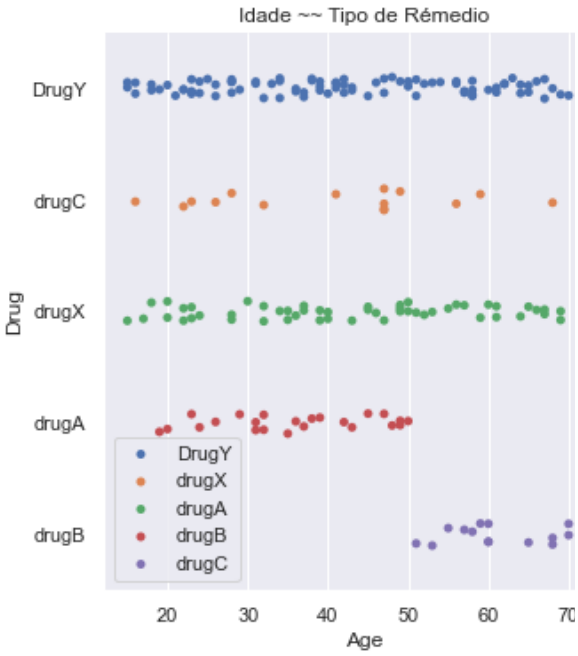
Configurações de Software:

Windows 10 Home
Versão 2004
Python 3.8.5
Jupyter Core 4.6.3
Jupyter Notebook 6.1.4
Pacotes:
Pandas
Matplotlib
Sklearn

- Resultados e Discussão

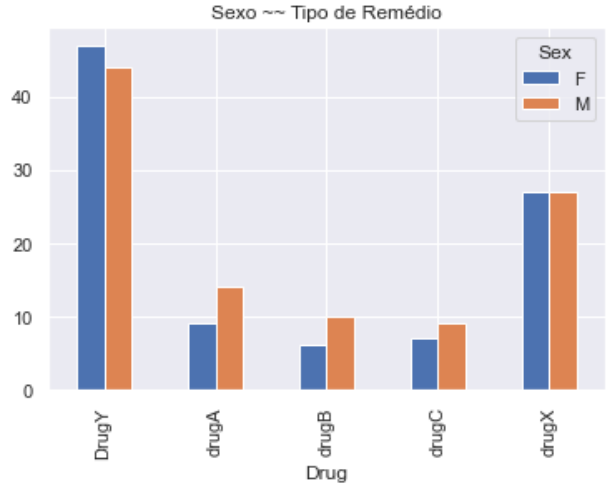
  Vamos fazer a comparação das características do paciente com o tipo de remédio para verificar quais característica são importantes para classificação.

  Fig12 – Idade com tipo de Remédio



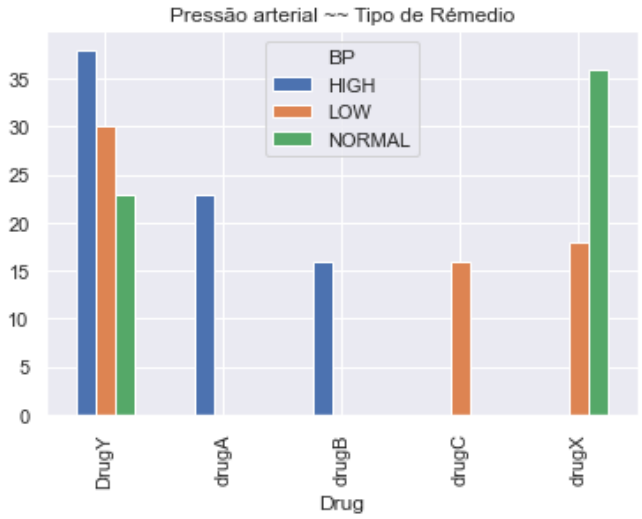
Através do Gráfico conseguimos constatar que o Remédio A, só é receitado para menores de 50 anos e o Remédio B só é receitado para pessoas maiores de 51 anos, os demais remédios não possuem restrição de idade.

Fig13 – Sexo com tipo de Remédio



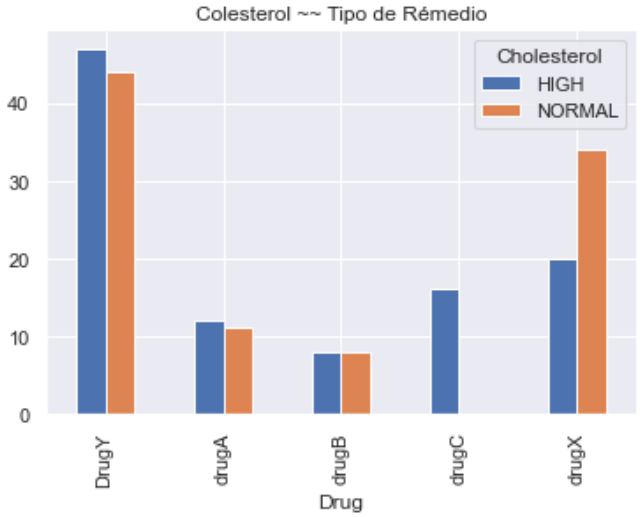
No geral os homens são mais medicados, porém eles estão em maior quantidade. Com isso é possível concluir que o sexo não é uma característica importante para classificação.

Fig14 – Pressão Arterial com Tipo de Remédio



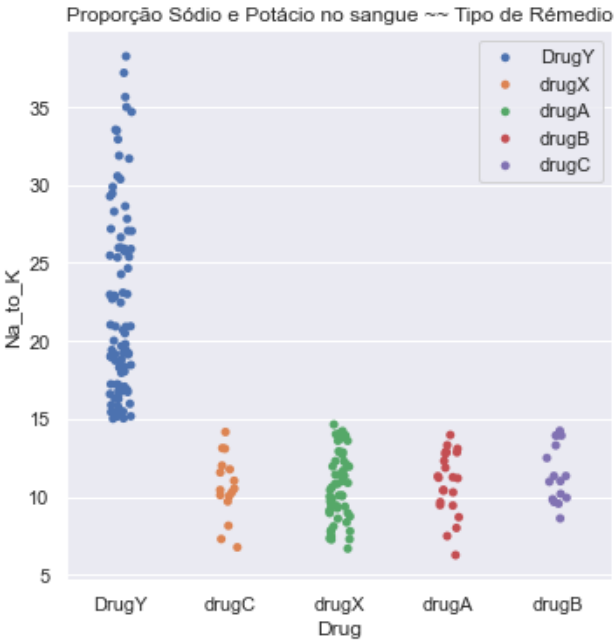
Percebemos que os Remédios A e B só são receitados quando os níveis de pressão estão altos, já o Remédio C só é receitado quando a pressão está baixa, o Remédio X não é receitado quando a pressão está alta.

Fig15 – Colesterol com Tipo de Remédio



Nesse gráfico vemos que todos remédios são utilizados independente se o colesterol está alto ou baixo, exceto o remédio C que só é receitado quando o colesterol está alto. Logo podemos ver que para o remédio C o nível de colesterol é importante para sua classificação.

Fig16 – Proporção de Sódio e Potássio com o Tipo de Medicamento



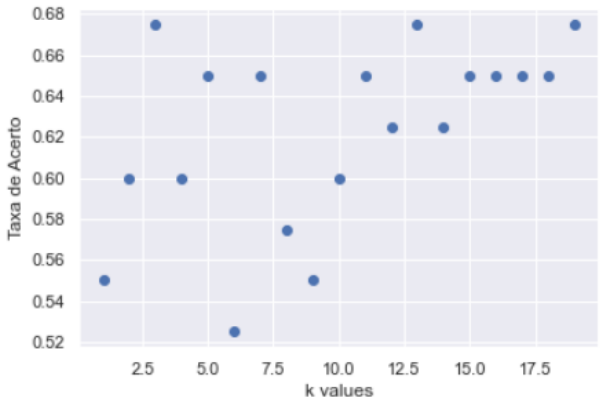
O remédio Y só é utilizado quando a proporção de Sódio e Potássio no sangue está acima de 15, logo essa característica é importante para classificar o Remédio Y, os demais só são receitados quando a proporção está abaixo de 15.

- Algoritmos Supervisionados

Para o trabalho foi utilizado 3 algoritmos de Classificação, o Dataset foi divido em 80% treino e 20% teste para todos os casos com estados aleatórios. O Primeiro deles foi o algoritmo KNN, que utiliza a estratégia de classificar com base na distância entre seus K vizinhos mais próximos.

Como já foi comentado foi construído um laço variando de 1 a 20 para acharmos o melhor valor de k.
Após percorrermos esse laço obtivemos o seguinte Gráfico:

Fig17 – Gráfico k vizinhos x treino score



Comprovamos esse gráfico colocando os valores de k = 6 menor valor e k = 3 maior valor

Fig18 – K = 6 Menor taxa de Acerto no teste

```
#Verificando menor taxa de acerto k = 6
knnBaixo= KNeighborsClassifier(n_neighbors=6)
accuracies = cross_val_score(knnBaixo, x_train, y_train, cv=5)
knnBaixo.fit(x_train,y_train)



print("Média do Train Score:",np.mean(accuracies))
print("Melhor Taxa de Acerto : ", knnBaixo.score(x_test,y_test))
```

```
Média do Train Score: 0.6625
Melhor Taxa de Acerto :  0.525
```

Fig19 – K = 3 Maior taxa de Acerto no teste

```
#Verificando o maior k = 3
knnBaixo= KNeighborsClassifier(n_neighbors=3)
accuracies = cross_val_score(knnBaixo, x_train, y_train, cv=5)
knnBaixo.fit(x_train,y_train)



print("Média do Train Score:",np.mean(accuracies))
print("Melhor Taxa de Acerto : ", knnBaixo.score(x_test,y_test))
```

```
Média do Train Score: 0.7
Melhor Taxa de Acerto :  0.675
```

O segundo algoritmo foi o Random Forest, que irá criar muitas árvores de decisão, de maneira aleatória, formando o que podemos enxergar como uma floresta, onde cada árvore será utilizada na escolha do resultado final.

O Algoritmo RD obteve média de 0.9875 no treino e Taxa de Acerto de 1.0

O terceiro Algoritmo que utilizamos foi o SVM, que usa a ideia de traçar uma linha que será usada para dividir as classes, com base nessa linha os novos exemplos serão classificados de acordo com o lado do espaço que eles estão colocados.

O resultado obtido foi:

Fig20 – Resultado SVM

```
# Terceiro Algoritmo SVM

SVM = svm.SVC(random_state = 42)
accuracies = cross_val_score(SVM, x_train, y_train, cv=5)
SVM.fit(x_train,y_train)



print("Média do Train Score:",np.mean(accuracies))
print("Porcentagem de Acerto:",SVM.score(x_test,y_test))

TrainScoreAlgoritmos["SVM Treino Score"] = np.mean(accuracies)
TestScoreAlgortimos["SVM Teste Score"] = SVM.score(x_test,y_test)
```

```
Média do Train Score: 0.7
Porcentagem de Acerto: 0.675
```

Como essa taxa de Acerto foi baixo, considerando os valores default do algoritmo, eu tive a ideia de testar o módulo chamado Gridsearch, que dados uma serie de parâmetros ele consegue retornar os melhores valores:

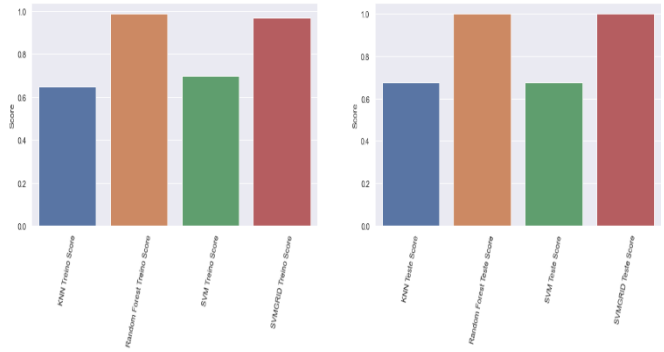
Fig21:Taxa de Acerto SVM, após usar o Gridsearch

```
#SVM achando o melhor score

grid = {
    'C':[0.01,0.1,1,10],
    'kernel' : ["linear","sigmoid"],
    'degree' : [1,3,5,7],
    'gamma' : [0.01,1]
}

SVMGRID  = svm.SVC();
SVMBEST = GridSearchCV(SVMGRID, grid, cv = 5)
SVMBEST.fit(x_train,y_train)
print("Best Parameters:",SVMBEST.best_params_)
print("Train Score:",SVMBEST.best_score_)
print("Test Score:",SVMBEST.score(x_test,y_test))

TrainScoreAlgoritmos["SVMGRID Treino Score"] = SVMBEST.best_score_
TestScoreAlgortimos["SVMGRID Teste Score"] = SVMBEST.score(x_test,y_test)
```

```
Best Parameters: {'C': 1, 'degree': 1, 'gamma': 0.01, 'kernel': 'linear'}
Train Score: 0.96875
Test Score: 1.0
```

Depois de testar os 3 algoritmos de Classificação obtivemos o seguinte gráfico:

Fig22 – Gráfico Consolidado Score Treino e Teste



Como podemos observar os algoritmos RD e SVM após usarmos o gridsearch obtiveram ótimos resultados, a pior performance ficou com o algoritmo KNN.

Conclusão

O trabalho desenvolvido obteve ótimos resultados, os algoritmos Random Forest e SVM de fato se sobressaíram, com esses resultados não seria utópico dizer que seria possível implementar esse trabalho na área da saúde, para otimizar a vida dos médicos.

Referências

[1] Russell, S. and Norvig, P., Artificial Intelligence – A Modern Approach, 2 nd Edition, Prentice-Hall, 2003.

[2]https://scikit-learn.org/stable/modules/neighbors.html#nearest-neighbors-classification

[3] https://scikit-learn.org/stable/modules/svm.html#classification

[4]https://scikit-learn.org/stable/modules/ensemble.html#forests-of-randomized-trees

[5] https://matplotlib.org/

[6]https://pandas.pydata.org/pandasdocs/stable/reference/api/pandas.read_csv.html

[7] https://medium.com/machina-sapiens/algoritmos-de-aprendizagem-de-m%C3%A1quina-qual-deles-escolher-67040ad68737#:~:text=Aprendizagem%20supervisionada%20%C3%A9%20a%20tarefa,a%20tarefa%20chama%2Dse%20regress%C3%A3o.

[8] http://www.blog.saude.gov.br/index.php/35602-populacao-teve-acesso-a-1-4-bi-de-consultas-medicas-pelo-sus-em-um-ano#:~:text=Somente%20em%202014%2C%20o%20sistema,R%24%2098%2C4%20bilh%C3%B5es.