UNIVERSIDADE EVANGÉLICA DE GOIÁS – UNIEVANGÉLICA

ENGENHARIA DE SOFTWARE

**Camila Souza**

**Bianca Abreu Félix**

**Felipe Franco**

Algoritmos de Classificação

Anápolis - GO

Maio, 2022

Link Github: https://github.com/FelpsFranco/Algoritmo\_De\_Classificacao

Dados: Presença de Doenças Cardíacas

**Introdução**

Algoritmos de Classificação ou Algoritmos de Machine Learning podem ser classificados em Aprendizagem Supervisionada, Aprendizagem Não supervisionada e Aprendizagem por Reforço. O uso desses algoritmos dependem do tipo de dado que será tratado e quais respostas precisam ser obtidas com sua implementação.

Para sua utilização é primeiramente necessário a extração de características, coleta de dados que serão utilizados. Após a coleta, será utilizado uma nova amostra de dados na qual ainda não temos informações e nem características, inserindo esses dados no modelo aprendido.

Alguns exemplos de algoritmos de classificação:

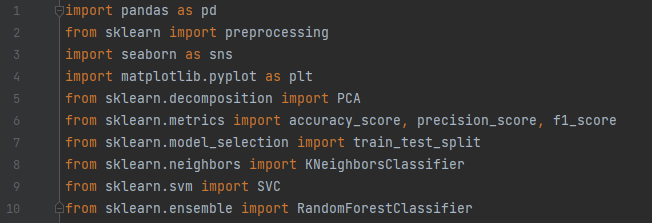
. **K-NN (Nearest Neighbors) ou K-vizinhos mais próximos:** ele se baseia na distância entre pontos para classificação dos dados. Para este modelo você precisa escolher um número K e averiguar qual classe mais se repete entre os K vizinhos mais próximos. A arbitrariedade na escolha do valor K é uma desvantagem deste método de classificação.

. **SVM (Support Vector Machine):** esse algoritmo de classificação binária, é utilizado em problemas no quais existem apenas duas classes, ele propõe traçar uma linha que divide o conjunto de dados em dois.

. **Random Forest (Florestas de decisão aleatória):** é um método que opera construindo uma infinidade de árvores de decisão em tempo de treinamento. A saída da floresta aleatória é a classe selecionada pela maioria das árvores.

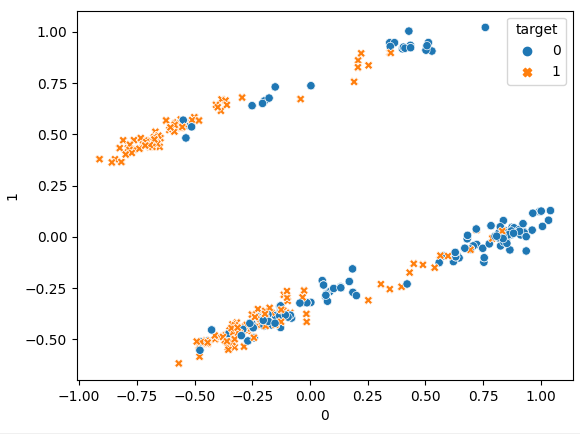
**Desenvolvimento:**

Bibliotecas utilizadas: pandas, sklearn, seaborn, matplotlib, sklearn.decomposition, sklean.metrics, sklearn.model\_selection, sklearn.neighbors, slearn.svm, sklearn.ensemble.



Antes a aplicação das formas de classificação foi realizado um tratamento de dados, carregando o Excel (.csv) e realizando uma normalização dos dados (Pré-Processamento) foi utilizado o transformado MinMaxScaler onde recursos são dimensionados para um determinado intervalo.

Feito a normalização de dados, realizamos a segmentação dos dados em Classes, para possibilitar sua visualização foi reduzido a dimensionalidade dos dados e registrado os novos valores. A partir da biblioteca matplotlib plotamos o seguinte dado colorindo pelas classes.

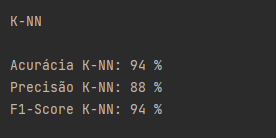


Após o gráfico é feito a separação de treinamento e testes, obtendo o resultado:

Treinamento: 717  
Teste: 308

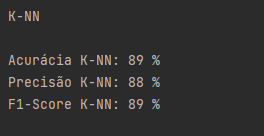
**K-NN (Nearest Neighbors)**

Aplicando K-NN, instauramos um modelo de dados utilizando primeiramente KNeighborsClassifier (**n\_neighbors = 2)** conseguimos obter:



O K-NN conseguiu classificar satisfatoriamente o conjuntos de dados.

Porém, realizando um novo teste, aumentando a quantidade de amostras para classificação KNeighborsClassifier (**n\_neighbors = 3)**



Conseguimos a mesma precisão porém perdemos acurácia.

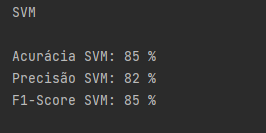
**Algoritmo Utilizado:**



**SVM (Support Vector Machines)**

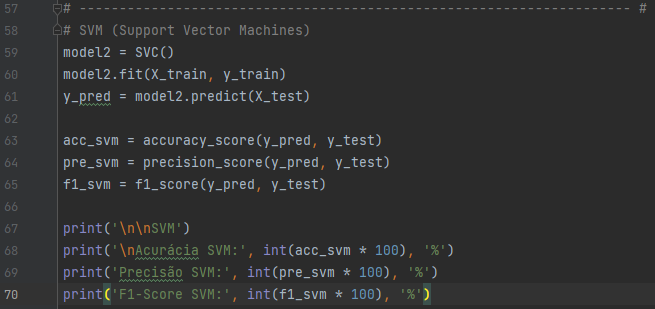
O modelo de SVM foi utilizado para termos de comparação com o K-NN, criando um segundo modelo de dados para aplicação. Novamente tratando Acurácia, Precisão e F1\_Score.

Obtivemos o seguinte resultado:



Conseguimos visualizar que o SVM obteve resultados piores que o K-NN ficando atrás em todos os pontos levantados (Para esse tipo de dados).

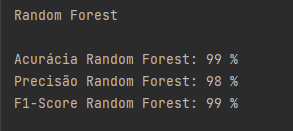
**Algoritmo Utilizado:**



**Random Forest**

Realizando mais uma comparação em termos de classificação, o Random Forest foram utilizados o parâmetros **RandomForestClassifier (n\_estimators=100, random\_state=26)**.

Obtivemos o seguinte resultado:



Com este resultado, o Random Forest fica à frente do K-NN e SVM para estes dados analisados.

**Algoritmo Utilizado:**

