Universidade de Brasília
Departamento de Ciência da Computação
Disciplina: Introdução a Inteligência Artificial
Professor: Díbio Leandro Borges

Random Forests

4 de outubro de 2021

Bruno Esteves Dalla Costa Filho, 17/0100863

1 Introdução

Este trabalho tem como objetivo aplicar algoritmos de Florestas Randômicas para predição de um diagnóstico de diabetes a partir de uma base de dados pre-definida.

A base de dados para o estudo se baseia em informações coletadas de um total de 768 pessoas, em específico mulheres indígenas com pelo menos 21 anos de idade, da etnia Pima. Foram escolhidas 8 variáveis independentes para a construção dos dados e uma alvo, que indicaria a classificação baseada no conjunto das anteriores.

Dentre as variáveis independentes temos:

Número de vezes grávida, Glucose, Pressão Sanguínea, Espessura da pele, Insulina, IMC, Pedigree de Diabetes e Idade.

2 Linguagem e Sistema Operacional

A linguagem de programação python3 foi utilizada na implementação.

O projeto foi compilado e executado no Windows utilizando o PowerShell.

3 Estruturação de dados

A partir do arquivo "diabetes.csv", utilizamos o pacote pandas para ler e formatar os dados em uma estrutura chamada de dataframe. Para treinar uma Random Forest nós separamos as variáveis independentes do alvo como é mostrado na figura.

```
data = pd.read_csv("diabetes.csv") #data = dataframe from pandas

X = data.drop("Outcome", axis=1) #axis = 1 indica colunas do dataframe
y = data["Outcome"]
```

4 Treinamento da Random Forest

Para nossa Random Forest foi utilizada a biblioteca sklearn, mais em específico a RandomForestClassifier.

Utilizando os estimadores descritos na descrição do projeto, sendo eles 10, 100 e a raiz quadrada do estudo do algoritmo baseado no tamanho da base de dados.

Para treinar a random forest separamos 80% dos dados para treino e 20% dos dados para testes futuros. Após esse processo realizamos o treino da random forest chamando "floresta.fit()".

```
estimadores = [10, 100, int(sqrt(768))] # n = 10, 100 e sqrt do estudo do algoritmo

for est in estimadores:

    Xtraining, Xtest, ytraining, ytest = train_test_split(X, y, test_size=0.2, random_state=0)
    #test_size = 0.2 seleciona 20% para teste e 80% para treino

    floresta = RandomForestClassifier(n_estimators=est, random_state=0, max_features="sqrt")

    floresta.fit(Xtraining, ytraining)
```

5 Precisão, Revocação e Medida F1

A precisão é calculada comparando o resultado da predição da floresta com o resultado real pela função da biblioteca sklearn accuracy_score().

```
pred = floresta.predict(Xtest)
precisao = accuracy_score(ytest, pred)
```

O resultado para os estimadores foi:

Para 10 estimadores: 0.7467532467532467

Para 100 estimadores: 0.7857142857142857

Para sqrt() estimadores: 0.7857142857142857

A partir da função classification_report() temos as medidas de revocação e F1 como descritas a seguir:

RESULTADO PARA 10 ESTIMADORES
0.7467532467532467

	precision	recall	f1-score	support
0	0.80	0.84	0.82	107
1	0.60	0.53	0.56	47
accuracy			0.75	154
macro avg	0.70	0.69	0.69	154
weighted avg	0.74	0.75	0.74	154

Figura 1: Estimadores = 10

RESULTADO PARA 100 ESTIMADORES
0.7857142857142857

	precision	recall	f1-score	support
0	0.84	0.86	0.85	107
1	0.66	0.62	0.64	47
accuracy			0.79	154
macro avg	0.75	0.74	0.74	154
weighted avg	0.78	0.79	0.78	154

Figura 2: Estimadores = 100

RESULTADO PARA 27 ESTIMADORES
0.7857142857142857

	precision	recall	f1-score	support
0	0.82	0.88	0.85	107
1	0.68	0.57	0.62	47
accuracy			0.79	154
macro avg	0.75	0.73	0.74	154
weighted avg	0.78	0.79	0.78	154

Figura 3: Estimadores = Sqrt

6 Ordens de Importância - Variáveis Independentes

O algoritmo Random Forests depois de treinado define pesos de importância para as variáveis independentes para suas predições. Com a feature "floresta.feature_importances_"do RandomForestClassifier do sklearn podemos ver esses pesos gerados pelo algoritmo.

Importancia das variaveis

	feature	importance
1	Glucose	0.210208
7	Age	0.176147
5	BMI	0.138669
6	DiabetesPedigreeFunction	0.111152
2	BloodPressure	0.104960
0	Pregnancies	0.090505
4	Insulin	0.085189
3	SkinThickness	0.083171

Figura 4: Estimadores = 10

Importancia das variaveis

	feature	importance
1	Glucose	0.242236
5	BMI	0.166750
7	Age	0.146922
6	DiabetesPedigreeFunction	0.129385
2	BloodPressure	0.090845
0	Pregnancies	0.080614
4	Insulin	0.073719
3	SkinThickness	0.069530

Figura 5: Estimadores = 100

Importancia das variaveis

	feature	importance
1	Glucose	0.227259
7	Age	0.160561
5	BMI	0.156754
6	DiabetesPedigreeFunction	0.124790
2	BloodPressure	0.092248
4	Insulin	0.081785
0	Pregnancies	0.081258
3	SkinThickness	0.075343

Figura 6: Estimadores = Sqrt

7 Matrizes de Confusão

Os resultados das matrizes de confusão para cada um dos estimadores foram os seguintes:

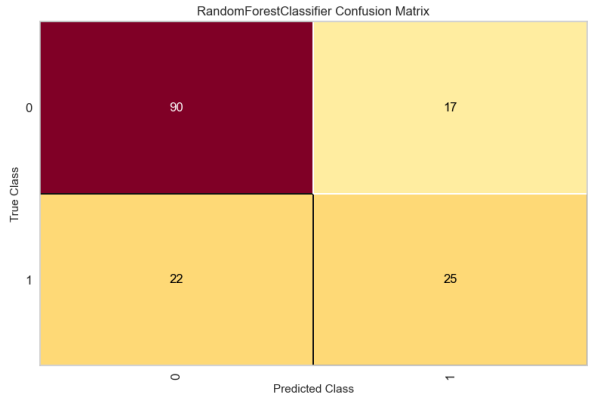

Figura 7: Estimadores = 10

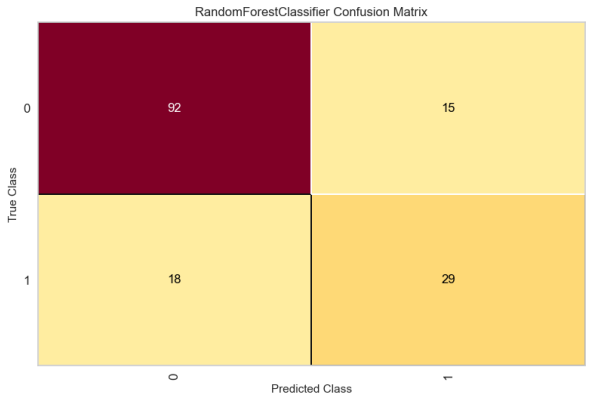

Figura 8: Estimadores = 100

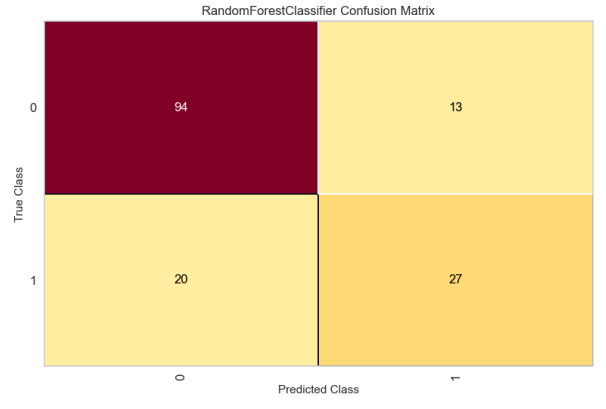

Figura 9: Estimadores = Sqrt

8 Conclusão

O algoritmo Random Forest prova-se ser um dos mais potentes, confiáveis e principalmente acessíveis, sendo que com uma linguagem de programação e com bibliotecas disponíveis que são muito poderosas o algoritmo pode ser implementado em poucas linhas.

Provou-se ser uma ótima ferramenta para predições de diagnósticos com uma boa confiança quando as variáveis independentes são escolhidas corretamente, podendo ser amplamente utilizável no meio da medicina e da aprendizagem de máquinas.

9 Referências

1. https://estatsite.com.br/2020/09/27/random-forest-em-python
2. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.htm
3. https://towardsdatascience.com/random-forest-in-python-24d0893d51c0