Ciência de Dados Prática: Predizer o surgimento de diabetes com base em medidas de diagnósticos

Francisco Thomás M. de Oliveira[1],**Ana Victória Araújo Maia**[1],**Regis Pires Magalhães**[1]

[1]Universidade Federal do Ceará (UFC) – Campus Quixadá
Caixa Postal 63.900-000 – Quixadá – CE – Brazil

{thomas_oliveira,victoria.maia}@alu.ufc.br, regismagalhaes@ufc.br

Abstract. *This article describes the resolution of a problem where the dataset is called Pima Indians Diabetes and is available on the Kaggle platform. The problem is presented and resolved on the YouTube channel of the Ciência de Dados Prática extension project, and documents are available detailing this resolution in Github. The problem is solved by using a supervised learning algorithm known as k-nearest neighbors (k-NN) to classify whether a particular sample of our data set is from a diabetic person or not diabetic. At the end, model evaluation metrics are applied to measure the final performance of our prediction model.*

Resumo. *Este artigo descreve a resolução de um problema onde o conjunto de dados (dataset) chama-se Pima Indians Diabetes e está disponível na plataforma Kaggle*[1]*. O problema é apresentado e resolvido no canal do YouTube do projeto de extensão Ciência de Dados Prática*[2]*, e disponibilizados documentos detalhando esta resolução no Github*[3]*. O problema é solucionado usando um algoritmo de aprendizagem supervisionada conhecido como k-vizinhos mais próximos(k-nearest neighbors – k-NN) para classificar se uma determinada amostra do nosso conjunto de dados é de uma pessoa diabética ou não diabética. Ao final são aplicadas métricas de avaliação de modelos para medir o desempenho final do nosso modelo de predição.*

1. Introdução

Ciência de Dados tem sido cada vez mais empregada em tudo mundo na tomada de decisão para melhoria da produtividade e redução de custos em instituições públicas e privadas [Provost and Fawcett 2013, Foreman et al. 2014, Siegel 2016], Entretanto, a falta de conteúdos didáticos gratuitos e disponíveis em língua portuguesa são aspectos para facilitar o acesso e estudo dessa área. Com base nesse foi criado o projeto, Ciência de dados Prática, que disponibiliza publicamente de forma gratuita vídeos e documentos explicativos sobre os principais temas na área de análise de dados e aprendizado de máquina. Os documentos explicativos mencionados apresentam a resolução comentada e detalhada dos problemas de uma forma simples e acessível até mesmo a leigos e iniciantes em ciência de dados. Objetivo principal é difundir as técnicas de análise de dados e aprendizado de máquina a um grande público, que poderá usá-las para uso eficiente de recursos,

[1]https://www.kaggle.com/
[2]https://www.youtube.com/channel/UCnR_-6nHlN-RrKl76IHOxcw
[3]https://github.com/ciencia-de-dados-pratica/praticas

eliminação de desperdícios, e em última instância, levando melhorias para a sociedade como um todo.

Este artigo apresenta um problema de classificação, usando aprendizado de máquina supervisionado, onde o conjunto de dados usados está descrito na plataforma Kaggle[4]. Tais dados pertencem originalmente ao Instituto Nacional de Diabetes e Doenças Digestivas e Renais (NIDDK, na sigla em inglês), O objetivo do conjunto de dados é diagnosticar se um paciente tem ou não diabetes, com base em determinadas medidas de diagnósticos incluídos no conjunto de dados. [UCIMachineLearning 2016, Kaggle 2018].

A classificação dos pacientes como diabéticos ou não diabéticos é realizada neste artigo através do algoritmo conhecido como k-vizinhos mais próximos (*k-nearest neighbors*, k-NN) da biblioteca *Scikit Learn* codificada da linguagem *Python* [Pedregosa et al. 2011]. O modelo preditivo é gerado a partir de um conjunto de treino. Ao final são calculadas as métricas acurácia, precisão e revocação (*recall*).

2. Metodologia

Os passos para resolução do problema proposto através do algoritmo kNN são descritos a seguir.

2.1. Seleção de características

O conjunto de dados possui 9 características (*features*) detalhadas a seguir.

Pregnancies refere-se ao número de vezes que a mulher engravidou.
Glucose é a concentração plasmática de glicose em um teste oral de tolerância à glicose.
BloodPressure representa a pressão arterial diastólica (mm Hg).
SkinThickness é a espessura da dobra da pele do tríceps (milímetro).
Insulin refere-se à insulina sérica de 2 horas (mu U / ml).
BMI é o índice de massa corporal (peso em kg /altura em m^2)
DiabetesPedigreeFunction é a função de pedigree de diabetes.
Age representa a idade em anos do paciente.
Outcome representa o rótulo (*label*), que neste caso é dado pelos valores categóricos 0 e 1, significando respectivamente as classes não diabético e diabético.

2.2. Analise e pré-processamento

Primeiramente aplica-se um pré-processamento e análise dos dados usando a biblioteca pandas do *Python*, a valores resultantes desse pré-processamento a partir desse método ira imprime informações sobre o conjunto de dados, incluindo o tipo de dado trabalhado (dtype) do índice e os tipos de coluna, a quantidade de valores nulos e uso da memória, o resultado está demonstrado na Figura 1.

Após essa análise preliminar, dos valores no conjunto, é feita a análise da distribuição dos dados.

Primeiramente são gerados estatísticas descritivas que resumem a tendência central, a dispersão e a forma da distribuição de um conjunto de dados, excluindo os valores não numéricos, demonstrado na Tabela 1.

[4]https://www.kaggle.com/

```
Data columns (total 9 columns):
Pregnancies                 768 non-null int64
Glucose                     768 non-null int64
BloodPressure               768 non-null int64
SkinThickness               768 non-null int64
Insulin                     768 non-null int64
BMI                         768 non-null float64
DiabetesPedigreeFunction    768 non-null float64
Age                         768 non-null int64
Outcome                     768 non-null int64
```

Figura 1. Informação dos dados

Por fim são analisadas as distribuições dos dados e apresentadas na Figura 2.

Essas análises foram feitas com o intuito de descobrir os tipos e como os dados então organizados no conjunto, em caso de valores faltantes serem removidos essas linhas e entender a distribuição dos dados e quais colunas influenciam mais no resultado.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

Tabela 1. Descrição dos dados

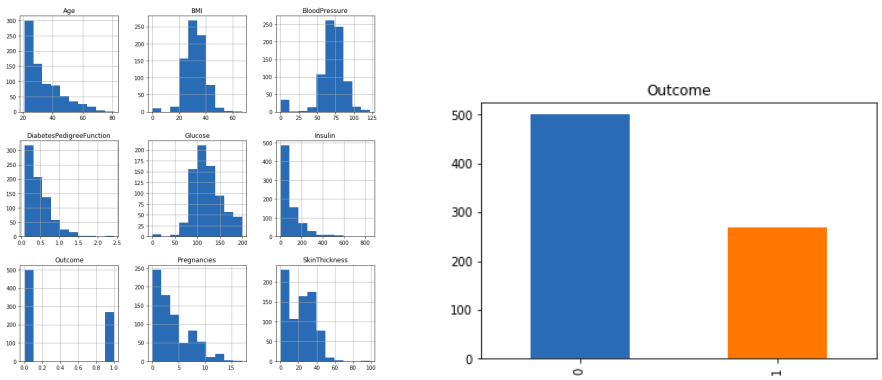

Figura 2. Distribuição dos valores de entrada (na esquerda) e saída (direita)

2.3. Criação e Treino do modelo

Durante a etapa de criação do modelo são usadas as implementações do algoritmo k-vizinhos mais próximos, além de métricas de classificação disponíveis na biblioteca *scikit-learn*. A geração do modelo preditivo é descrita a seguir.

2.3.1. Particionamento do conjunto de dados em conjunto de treino e conjunto de teste

O conjunto de dados originais é particionado em conjunto de treino e conjunto de teste na proporção de 70% e 30%, respectivamente. Outros percentuais poderiam ser usados. Os percentuais adotados são frequentemente usados na literatura [Raschka and Mirjalili 2017]. O conjunto de treino é usado para geração do modelo preditivo, enquanto que o conjunto de teste é usado para avaliar o modelo gerado.

2.3.2. Geração do modelo a partir dos dados de treino

O modelo preditivo é gerado através do uso do classificador *KNeighborsClassifier*, que é uma implementação do algoritmo k-vizinhos mais próximos (kNN) disponível na biblioteca *scikit learn*. Outros classificadores poderiam ser usados, mas este artigo foca na resolução através no kNN.

A ideia principal do KNN e determinar uma nova amostra em base do conjunto de dados usado, o valor de k define quantidade de vizinhos mais próximos que serão utilizados para averiguar de qual classe a nova amostra pertence, para calcular essa distancia entre os pontos, se define uma métrica de distancia, a métrica de distância a mais utilizada é a distância Euclidiana.

Onde para pontos n-dimensionais, $P = (p_1...p_n)$ e $Q = (q_1...q_n)$, a distância é computada como.

$$D = \sqrt{(p_1 + q_1)^2 + ... + (p_n + q_n)^2}$$

Foram testados valores de k entre os números inteiros 1 e 100, e para cada modelo testado foi calculado a acurácia em base do valor de acurácia de cada modelo foi escolhido o que tinha maior valor, e montado um gráfico para demonstrar os resultados. O melhor valor obtido foi o valor 49. Demonstrado na figura 3.

Figura 3. Pontos de teste para o valor de K

2.3.3. Avaliação do modelo com base nos valores preditos

Por último são calculadas as seguintes métricas de avaliação de classificadores: acurácia, precisão e revocação (*recall*). Os valores preditos pelo modelo são comparados com os valores reais do conjunto de teste. A partir dessa comparação é gerada uma matriz de confusão, cujos resultados são usados para cálculo das métricas anteriormente mencionadas.

A acurácia é a medida de desempenho mais intuitiva. É simplesmente a razão entre as observações corretamente previstas e o total de observações. Precisão é a razão entre as observações positivas previstas corretamente e as observações positivas preditas totais. Revocação diz respeito à razão entre as observações positivas corretamente previstas e todas as observações na classe real, quando o valor de saída é verdadeiro. Os valores das métricas para o conjunto de teste estão definidos na tabela da Figura 4.

K vizinhos mais próximos (KNN)	
Accuracy	0.77
Precision	0.75
Recall	0.51

	Positivos	**Negativos**
Positivos Preditos	136	14
Negativos Preditos	40	41

Figura 4. Resultados da validação do modelo e Matriz de confusão

3. Conclusões e Trabalhos futuros

O artigo apresenta didaticamente os paços para resolução de um problema prático de ciência de dados usando o algoritmo kNN. Ao final, foi obtido um modelo preditivo adequado para a resolução do problema de classificação de diabéticos.

Como trabalhos futuros pretende-se apresentar e disponibilizar mais problemas e técnicas de ciência de dados para acesso público e gratuito.

Referências

Foreman, J. W., Jennings, G., and Miller, E. (2014). *Data smart: Using data science to transform information into insight*. Wiley.

Kaggle (2018). Kaggle. `http://kaggle.com/`. Accessed: 2018-09-10.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Provost, F. and Fawcett, T. (2013). *Data Science for Business: What you need to know about data mining and data-analytic thinking*. "O'Reilly Media, Inc.".

Raschka, S. and Mirjalili, V. (2017). *Python machine learning*. Packt Publishing Ltd.

Siegel, E. (2016). *Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die.* Wiley.

UCIMachineLearning (2016). Pima indians diabetes database. `https://www.kaggle.com/uciml/pima-indians-diabetes-database`. Accessed: 2018-09-10.