



UNIVERSIDADE DA BEIRA INTERIOR

Predição personalizada de eventos de hipoglicémia

João Tomás Mota Cunha

Relatório no âmbito da UC Seminário
1º ciclo em Ciências Biomédicas
2020/20201

Tutor(es)
Prof. Doutor Nuno Garcia
Prof. Doutor Nuno Pombo
Mestre Virginie Felizardo

Covilhã, junho de 2021

Resumo

As novas tecnologias de inteligência artificial estão cada vez mais presentes na área da saúde. Exemplo disso é o uso de técnicas de *machine learning* na predição de eventos de hipoglicemia. Este trabalho apresenta uma proposta de predição personalizada, realizada em 5 pacientes, de eventos de hipoglicemia recorrendo aos classificadores: *AdaBoost Tree*, *Bagged Tree* e *Subspace KNN*. Foram usados *datasets* de treino e teste previamente selecionados e adaptados. Cada classificador possui um conjunto de parâmetros ajustados de forma a apresentar uma melhor predição para cada paciente. Com base nos resultados obtidos foi possível calcular o número de “falsos alarmes” e a percentagem de hipoglicemias previstas. Verificamos que o classificador mais eficaz foi o *Bagged Tree*, devido aos seus baixos valores de “falsos alarmes” sendo capaz de prever grande parte dos eventos de hipoglicemia. Os piores resultados dever-se, provavelmente, à escolha inadequada dos valores de cada parâmetro, influenciando os resultados obtidos. Com a aplicação destes métodos concluímos que utilização técnicas de *machine learning* é muito importante no papel desempenhado pelos engenheiros biomédicos, tornando mais acessível a melhoria destas aplicações possibilitando previsões cada vez mais precisas e eficazes, com o objetivo de melhorar a qualidade de vida dos diabéticos.

Abstract

New artificial intelligence technologies are increasingly present in the health area. An example of this is the use of machine learning techniques to predict hypoglycemia events. This work presents a proposal for personalized prediction, performed in 5 patients of hypoglycemia events using the classifiers: *AdaBoost Tree*, *Bagged Tree*, and *Subspace KNN*. Previously selected and adapted training and test datasets were used. Each classifier has a set of parameters adjusted to present a better prediction for each patient. Based on the results obtained, it was possible to calculate the number of “false alarms” and the percentage of predicted hypoglycemia. We found that the most effective classifier was the *Bagged Tree*, due to its low “false alarms” values, being able to predict most of the hypoglycemia events. The worst results are probably due to the inadequate choice of values for each parameter, influencing the results obtained. With the application of these methods, we conclude that the use of machine learning techniques is very important in the role played by biomedical engineers, making the improvement of these applications more accessible, enabling increasingly accurate and effective predictions, to improve the quality of life of diabetics.

Introdução

Diabetes *mellitus* (DM) é uma doença crônica caracterizada pela incapacidade de regulação normal da glicose no sangue (GS), principalmente devido à falha do corpo em produzir insulina (diabetes tipo I) ou à incapacidade do corpo de responder à ação da insulina (diabetes tipo II), como também diabetes gestacionais[1]. De acordo com a Federação Internacional de Diabetes (FID), em 2015, havia 415 milhões de pessoas com diabetes no mundo, tornando o diabetes uma das maiores emergências globais de saúde do século 21[2]. Pessoas com diabetes são caracterizadas por terem elevadas taxas de morbidade e mortalidade em comparação com a população normal[3].

A hipoglicemia ou queda da GS abaixo dos níveis críticos, é reconhecida como o processo de decaimento do nível de GS a baixo de 70 mg / dL ou 3,9 mmol / L [4]. Sendo assim uma das condições mais letais, mais comuns, para diabéticos de tipo I, tal como para diabéticos tipo II[5]. A hipoglicemia pode causar alguns sintomas como perda de consciência, confusão, convulsões e, em casos extremos, morte, no entanto, estes sintomas podem variar para diferentes indivíduos com base em vários fatores[6]. Além disso, com base no nível de GS, os sintomas de hipoglicemia podem ser classificados como leves, moderados e extremos. No entanto, a maioria dos casos a hipoglicemia não apresenta nenhum sintoma e ocorre silenciosamente[5].

Ao longo do tempo tem existido um grande aumento no uso de tecnologias para o controle dos diabetes, através de sistemas de monitoramento da glicose, sendo estes tópicos de tendência na biomedicina[7]. Por este motivo, os engenheiros biomédicos tencionam criar modelos de previsão cada vez mais eficientes de maneira a reduzir a incerteza e melhorar a qualidade de vida dos diabéticos.

A complexidade do prognóstico e gestão dos diabetes permitiu que técnicas de inteligência artificial (IA) e de *machine learning* (ML) se tornarem tecnologias-chave para o tratamento dos diabéticos[3]. Esta é a razão pela qual existe um aumento exponencial no trabalho de pesquisa focado em técnicas de ML para prever eventos glicêmicos adversos em geral e hipoglicemia em particular[8].

O ML tornou-se um dos principais campos da inteligência artificial (IA) nos últimos tempos e seu impacto na área da saúde tem sido enorme [9][10]. O conceito de ML tem suas raízes na ciência da computação, estatística e otimização. Toma como objetivo permitir que o computador se treine sem ser explicitamente programado, dando ao computador o poder de prever resultados até um certo nível de precisão. Em muitos cenários médicos, o conhecimento prévio de um evento adverso pode evitar uma emergência e, em muitos casos, salvar vidas. A qualidade do ML para prever o futuro torna-se assim uma ótima ferramenta para antecipar tais eventos [11]. Atualmente vários algoritmos são usados para prever diabetes, incluindo o método tradicional de ML, como é o caso do trabalho de Kavakiotis et al. (2017)[12], em que realizaram uma revisão sistemática de técnicas de ML e mineração de dados, usando diversos classificadores, na pesquisa de diabetes no contexto da previsão e diagnóstico do diabetes, complicações do

diabetes, histórico e ambiente genético, e cuidados com a saúde. Existe também o trabalho de Razavian et al. (2015)[13] que, a fim de lidar com os conjuntos de dados de alta dimensão, construíram modelos de previsão com base em regressão logística para diferentes *onsets* de previsão do diabetes tipo II. Ou até Georga et al. (2013)[14] que se focou na glicose e usou o *support vector regression* (RVS) para prever o diabetes, que é um problema de regressão multivariada.

Neste trabalho o objetivo será a predição personalizada de eventos de hipoglicemia recorrendo à linguagem de programação *python*, através do uso dos classificadores: *AdaBoost Tree*, *Bagged Tree* e *Subspace KNN*. De forma a obter predições personalizadas foram ajustados os parâmetros de cada classificador de acordo com os dados do paciente.

No restante relatório é constituído por “Materiais e métodos”, onde foram descritos alguns conceitos de *python* e *machine learning* além dos classificadores e *dataset* usados, passando aos “Resultados e discussão” onde foram avaliados e discutidos os resultados adquiridos, por fim a “Conclusão” onde descreveram as conclusões adquiridas neste estudo.

Materiais e métodos

1) Python e Machine Learning

Python, é o exemplo de uma linguagem de programação amplamente utilizada em *ML*. Esta provou ser uma muito eficaz e é usada em muitas aplicações de computação científica [15], usando algoritmos pré-empacotados é possível utilizá-la em processos de *ML*, usados para resolver tarefas de grande escala numa velocidade lenta e têm requisitos de memória e capacidade de computação[16].

ML é um tipo de *IA* que engloba métodos algorítmicos que permitem que os modelos “aprendam” sem sere programados explicitamente. Ou seja, os mecanismos de *machine learning algorithms* permite que os modelos aprendam por si mesmos uma vez que o *learning algorithms* é determinado.

Isso torna o modelo mais adaptável ao ambiente incerto e em constante mudança do mundo real. O *ML* aborda a questão de como construir programas computacionais que melhoram automaticamente através da experiência[17].

2) Classificadores

Ada-boost: é um dos classificadores de reforço de conjunto que combina vários classificadores para aumentar a precisão dos classificadores. O classificador *AdaBoost*, através de um método de conjunto iterativo, constrói um classificador forte combinando vários classificadores de baixo desempenho para que você obtenha um classificador forte de alta precisão. O conceito

básico por trás do *Adaboost* é definir os pesos dos classificadores e treinar a amostra de dados em cada iteração de forma que garanta as previsões precisas de observações incomuns[18].

Bagged Tree: Com recuso a métodos de *ensemble*, que combinam várias árvores de decisão para produzir melhor desempenho preditivo do que utilizar uma única árvore de decisão, é uma técnica de *ensemble* que cria uma coleção de *predictors*. Nesta são ajustadas árvores consecutivas (amostra aleatória) e, a cada etapa, o objetivo é resolver o *net error* da árvore anterior[19].

Subspace KNN: é um *non-parametric* e *lazy learning algorithm*. *Non-parametric* significa que não há nenhuma suposição para a distribuição de dados subjacente. Ou seja, a estrutura do modelo determinada a partir do conjunto de dados. O *lazy algorithm* significa que não precisa de nenhum ponto de dados de treino para a geração do modelo. Todos os dados de treino usados na fase de teste. Isso torna o treino mais rápido e a fase de teste mais lenta e cara (ocupa mais memória). Na pior das hipóteses, o KNN precisa de mais tempo para verificar todos os pontos de dados e a verificação de todos os pontos de dados exigirá mais memória para armazenar os dados de treino[20].

3) Datasets

O *design* do *framework* foi executado usando um *dataset* de diabetes [21] do *UCI Machine Learning Repository* que contém conjuntos de dados de 70 diabéticos tipo 1. Os registos de pacientes com diabetes do conjunto de dados UCI foram obtidos de duas fontes: um dispositivo de registo eletrônico automático que tinha um relógio interno para registar eventos e registos em papel que forneciam apenas intervalos de tempo relacionados à hora do dia, como pequeno-almoço, almoço, jantar, hora de dormir.

Os *datasets* disponibilizados já foram alvo de pré-processamento: manipulação de dados, extração de variáveis, imputação de glicose no sangue se um dado ausente for verificado, reamostragem, para obter um conjunto de dados balanceado.

4) Modelo proposto

O modelo proposto para predição personalizada de eventos de hipoglicémia consiste no uso de três classificadores de *machine learning*, um tipo de inteligência artificial, sendo eles: *AdaBoost Tree*, *Bagged Tree* e *Subspace KNN*.

Nota: O código computacional característico do classificador *AdaBoost Tree* está retratado no Anexo 1, com ao editor *Spyder* através da linguagem de programação *python*. A única parte do código que difere entre classificadores é a referente à “construção do modelo”, que está entre a linha 26 e a linha 30, esta parte do código tem de ser adaptada pois é a responsável pela escolha dos parâmetros usados para cada paciente.

Esta predição personalizada será realizada nos pacientes 55, 56, 65, 67 e 68. Para cada um dos pacientes realizamos um dataset de treino (dados de 52 pacientes+dados com 80% de hipoglicémias do paciente) e um dataset de teste (dados com 20% de hipoglicémias do paciente), estes dados são descritos na tabela 1.

Cada classificador possui um conjunto de parâmetros, os mesmos foram ajustados de forma a apresentar uma melhor predição para cada paciente, como é possível observar na tabela 2.

Nota: Conforme cada classificador alterou-se o nº estimators (*AdaBoost Tree*), nº trees e seed (Bagged Tree) e max features (Subspace KNN).

Tabela 1: Descrição do Dataset.

Pacientes	Dataset de treino Número de hipoglicémias do paciente (número total de observações)	Dataset de teste Número de hipoglicémias do paciente (número total de observações)
P55 (27 hipoglicémias)	21 (26349 observações)	6 (76 observações)
P56 (72 hipoglicémias)	57 (26249 observações)	15 (68 observações)
P65 (20 hipoglicémias)	16 (26289 observações)	4 (134 observações)
P67 (24 hipoglicémias)	19 (26217 observações)	5 (184 observações)
P68 (19 hipoglicémias)	15 (26090 observações)	4 (291 observações)

Tabela 2: Descrição dos parâmetros utilizados.

Classificadores	Parâmetro 1	Parâmetro 2	Parâmetro 3
AdaBoost Tree	Nº estimators = ..	Learning rate = 1	Algorithm = 'SAMME'

Bagged Tree	Base estimator = cart	Nº estimators = nº trees	Random state = seed
Subspace KNN	Base estimator = KNeighborsClassifier(2)	Bootstrap = False	Max features = ..

Resultados e Discussão

A tabela 3 representa os valores obtidos no classificador *AdaBoost* para cada um dos 5 pacientes: P55, P56, P65, P67 e P68. Tendo sido adaptado o valor de “Nº estimators” para cada um deles, sendo, respetivamente, os valores: 20, 630, 105, 1750 e 80. Com base nos resultados obtidos em cada classificador foi possível calcular o número de “falsos alarmes” e a percentagem de hipoglicémias previstas, sendo estes valores descritos nas 3 tabelas seguintes, para cada classificador.

Tabela 3: Resultados do classificador *AdaBoost*.

Nº estimators = 20	P55		
[63 0 0]	precision	recall	
[5 1 0]	0	0.93	1.00
[0 0 6]	1	1.00	0.17
Accuracy: 0.93	2	1.00	1.00
Nº estimators = 630	P56		
[11 8 0]	precision	recall	
[9 24 0]	0	0.55	0.58
[0 0 15]	1	0.75	0.73
Accuracy: 0.75	2	1.00	1.00
Nº estimators = 105	P65		
[68 48 0]	precision	recall	
[5 8 0]	0	0.93	0.59
[0 0 4]	1	0.14	0.62
Accuracy: 0.60	2	1.00	1.00
Nº estimators = 1750	P67		
[123 34 0]	precision	recall	
[14 7 0]	0	0.90	0.78
[0 0 5]	1	0.17	0.33
Accuracy: 0.74	2	1.00	1.00
Nº estimators = 80	P68		
[135 117 0]	precision	recall	
[15 19 0]	0	0.90	0.54
[0 0 4]	1	0.14	0.56

Accuracy: 0.54	2	1.00	1.00
AdaBoost Tree			
Pacientes	Falsos Alarmes	Hipoglicémias Previstas %	
P55	0	0,75	
P56	0,235294118	0,93	
P65	0,4	1	
P67	0,209876543	1	
P68	0,45703125	1	

De acordo com a tabela 3, apenas foram obtidos valores de “Falsos Alarmes” abaixo dos 30% para os 3 pacientes: P55, P56 e P67, enquanto os restantes pacientes obtiveram valores mais elevados. Levando a concluir que este classificador teve uma precisão relativamente baixa.

A tabela 4 representa os valores obtidos no classificador *Bagged Tree* para cada um dos 5 pacientes: P55, P56, P65, P67 e P68. Tendo sido adaptado os valores de “nº trees” e “seed” para cada um deles, sendo, respetivamente, os valores: 90 e 8, 30 e 7, 20 e 7, 100 e 7 e 50 e 20.

Tabela 4: Resultados do classificador *Bagged Tree*.

Nº trees = 90	seed = 8	P55	
[55 8 0]		precision	recall
[2 4 0]	0	0.96	0.87
[0 0 6]	1	0.33	0.67
Accuracy: 0.87	2	1.00	1.00
Nº trees = 30	seed = 7	P56	
[14 5 0]		precision	recall
[24 9 0]	0	0.37	0.74
[0 0 15]	1	0.64	0.27
Accuracy: 0.57	2	1.00	1.00
Nº trees = 20	seed = 7	P65	
[104 12 0]		precision	recall
[8 5 0]	0	0.93	0.90
[0 0 4]	1	0.29	0.38
Accuracy: 0.85	2	1.00	1.00
Nº trees = 100	seed = 7	P67	
[139 18 0]		precision	recall
[17 4 0]	0	0.89	0.89
[0 0 5]	1	0.18	0.19
Accuracy: 0.81	2	1.00	1.00
Nº trees = 50	seed = 20	P68	
[230 22 0]		precision	recall
[23 11 0]	0	0.91	0.91
[0 0 4]	1	0.33	0.32
Accuracy: 0.84	2	1.00	1.00
Bagged Tree			
Pacientes	Falsos Alarmes	Hipoglicémias Previstas %	
P55	0,115942029	1	

P56	0,147058824	0,4
P65	0,1	1
P67	0,111111111	0,4
P68	0,086614173	1

No caso da tabela 4, todos os valores obtidos de “Falsos Alarmes” foram abaixo dos 30%, demonstrando que este classificador teve uma precisão bastante elevada.

A tabela 5 representa os valores obtidos no classificador *Subspace KNN* para cada um dos 5 pacientes: P55, P56, P65, P67 e P68. Tendo sido adaptado o valor de “Max features” para cada um deles, sendo, respetivamente, os valores: 12, 5, 7, 4 e 3.

Tabela 5: Resultados do classificador *Subspace KNN*.

Max features = 12		P55		
[46 17 0]		precision recall		
[2 4 0]		0	0.96	0.73
[0 0 6]		1	0.19	0.67
Accuracy: 0.75		2	1.00	1.00
Max features = 5		P56		
[10 9 0]		precision recall		
[11 17 5]		0	0.48	0.53
[0 0 15]		1	0.65	0.52
Accuracy: 0.63		2	0.75	1.00
Max features = 7		P65		
[60 56 0]		precision recall		
[7 6 0]		0	0.90	0.52
[0 0 4]		1	0.10	0.46
Accuracy: 0.53		2	1.00	1.00
Max features = 4		P67		
[99 52 6]		precision recall		
[13 8 0]		0	0.88	0.63
[0 0 5]		1	0.13	0.38
Accuracy: 0.61		2	0.45	1.00
Max features = 3		P68		
[154 93 5]		precision recall		
[16 16 2]		0	0.91	0.61
[0 0 4]		1	0.15	0.47
Accuracy: 0.60		2	0.36	1.00
Subspace KNN				
Pacientes	Falsos Alarmes	Hipoglicémias Previstas %		
P55	0,246376812	1		
P56	0,264705882	0,93		
P65	0,466666667	1		
P67	0,333333333	1		
P68	0,370517928	1		

Por fim na tabela 5, apenas foram obtidos valores de “Falsos Alarmes” abaixo dos 30% para 2 pacientes: P55 e P56, enquanto os restantes pacientes obtiveram valores mais elevados. Com estes dados foi possível concluir que este classificador teve uma precisão muito baixa.

Por fim, é possível verificar que, neste trabalho, o classificador mais eficaz foi o *Bagged Tree*, devido aos seus baixos valores de “Falsos Alarmes” sendo capaz de prever grande parte dos eventos de hipoglicemia, enquanto os restantes classificadores demonstraram mais dificuldade a realizar essa tarefa.

Estes resultados “menos positivos” podem dever-se à escolha inadequada dos valores para cada parâmetro utilizado, que influencia em muito os resultados obtidos, os mesmos têm de ser adaptados a cada paciente tornando a escolha dos mesmos mais difícil e demorada.

Os valores de hipoglicémias previstas foram relativamente parecidos para os três classificadores, não permitindo grandes conclusões em relação à qualidade/eficácia de cada classificador.

Conclusão

Tem havido um grande aumento na quantidade de trabalhos de pesquisa realizados na área de predição de hipoglicémia através do uso de métodos de ML, no entanto para chegar à sua máxima eficiência e desempenho terá um longo caminho de melhorias pela frente.

Este trabalho teve como objetivo a predição personalizada de eventos de hipoglicemia, com recurso à linguagem programação python, através do uso de três classificadores: AdaBoost Tree, Bagged Tree e Subspace KNN. De forma a obter predições personalizadas foram ajustados os parâmetros de cada classificador de acordo com os dados do paciente. Para que fosse possível atingir este objetivo foi necessária a programar um código capaz de realizar esta predição, com recurso a técnicas de ML, realizando também uma seleção adequada dos valores dos parâmetros utilizados em cada classificador para cada um dos 5 pacientes (P55, P56, P65, P67 e P68).

Desta forma o seminário teve várias contribuições, como por exemplo, permitiu a comparação entre os classificadores utilizados, concluindo que existem classificadores mais precisos e eficazes na previsão de eventos de hipoglicémia do que outros; foi capaz também de alertar a importância dos parâmetros utilizados neste tipo de técnicas, que condicionam muito a qualidade dos resultados obtidos.

Exemplos como o usado neste trabalho são muito relevantes no papel de engenheiros biomédicos, tornando mais acessível o estudo e melhoria de aplicações capazes de fazer previsões cada vez mais precisas e eficientes, com o objetivo de reduzir a incerteza e melhorar a qualidade de vida dos diabéticos.

Referências bibliográficas

- [1] Q. Zou, K. Qu, Y. Luo, D. Yin, Y. Ju, and H. Tang, "Predicting Diabetes Mellitus With Machine Learning Techniques," *Front. Genet.*, vol. 9, p. 515, Nov. 2018, doi: 10.3389/fgene.2018.00515.
- [2] S. Oviedo, J. Vehí, R. Calm, and J. Armengol, "A review of personalized blood glucose prediction strategies for T1DM patients," *Int. j. numer. method. biomed. eng.*, vol. 33, no. 6, pp. 1-21, 2017, doi: 10.1002/cnm.2833.
- [3] A. Z. Woldaregay *et al.*, "Data-driven modeling and prediction of blood glucose dynamics: Machine learning applications in type 1 diabetes," *Artif. Intell. Med.*, vol. 98, no. August 2018, pp. 109-134, 2019, doi: 10.1016/j.artmed.2019.07.007.
- [4] J. F. Yale, B. Paty, and P. A. Senior, "Hypoglycemia," *Can. J. Diabetes*, vol. 42, pp. S104-S108, Apr. 2018, doi: 10.1016/j.jcjd.2017.10.010.
- [5] O. Mujahid, I. Contreras, and J. Vehi, "Machine learning techniques for hypoglycemia prediction: Trends and challenges," *Sensors (Switzerland)*, vol. 21, no. 2, pp. 1-21, 2021, doi: 10.3390/s21020546.
- [6] "IDF Diabetes Atlas." <https://www.idf.org/e-library/epidemiology-research/diabetes-atlas/159-idf-diabetes-atlas-ninth-edition-2019.html> (accessed Jun. 04, 2021).
- [7] P. Shende, P. Sahu, and R. Gaud, "A technology roadmap of smart biosensors from conventional glucose monitoring systems," *Therapeutic Delivery*, vol. 8, no. 6. Future Medicine Ltd., pp. 411-423, Jun. 01, 2017, doi: 10.4155/tde-2017-0012.
- [8] I. Contreras and J. Vehi, "Artificial intelligence for diabetes management and decision support: Literature review," *Journal of Medical Internet Research*, vol. 20, no. 5. JMIR Publications Inc., p. e10775, May 30, 2018, doi: 10.2196/10775.
- [9] J. Wiens and E. S. Shenoy, "Machine Learning for Healthcare: On the Verge of a Major Shift in Healthcare Epidemiology," *Clin. Infect. Dis.*, vol. 66, no. 1, pp. 149-153, Jan. 2018, doi: 10.1093/cid/cix731.
- [10] J. A. Roth, M. Battegay, F. Juchler, J. E. Vogt, and A. F. Widmer, "Introduction to Machine Learning in Digital Healthcare Epidemiology," *Infect. Control Hosp. Epidemiol.*, vol. 39, no. 12, pp. 1457-1462, Dec. 2018, doi: 10.1017/ice.2018.265.
- [11] R. C. Deo, "Machine learning in medicine," *Circulation*, vol. 132, no. 20, pp. 1920-

1930, Nov. 2015, doi: 10.1161/CIRCULATIONAHA.115.001593.

[12] I. Kavakiotis, O. Tsave, A. Salifoglou, N. Maglaveras, I. Vlahavas, and I. Chouvarda, "Machine Learning and Data Mining Methods in Diabetes Research," *Comput. Struct. Biotechnol. J.*, vol. 15, pp. 104-116, 2017, doi: 10.1016/j.csbj.2016.12.005.

[13] N. Razavian, S. Blecker, A. M. Schmidt, A. Smith-Mclallen, S. Nigam, and D. Sontag, "Population-level prediction of type 2 diabetes from claims data and analysis of risk factors," *Big Data*, vol. 3, no. 4, pp. 277-287, 2015, doi: 10.1089/big.2015.0020.

[14] E. I. Georga *et al.*, "Multivariate Prediction of Subcutaneous Glucose Concentration in Type 1 Diabetes Patients Based on Support Vector Regression," *IEEE J. Biomed. Heal. Informatics*, vol. 17, no. 1, pp. 71-81, 2013, doi: 10.1109/TITB.2012.2219876.

[15] E. Serrano, J. G. Blas, J. Carretero, M. Abella, and M. Desco, "Medical Imaging Processing on a Big Data Platform Using Python: Experiences with Heterogeneous and Homogeneous Architectures," in *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, 2017, pp. 830-837, doi: 10.1109/CCGRID.2017.56.

[16] K. Gao, G. Mei, F. Piccialli, S. Cuomo, J. Tu, and Z. Huo, "Julia language in machine learning: Algorithms, applications, and open issues," *Computer Science Review*, vol. 37. Elsevier Ireland Ltd, p. 100254, Aug. 01, 2020, doi: 10.1016/j.cosrev.2020.100254.

[17] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science (80-.)*, vol. 349, no. 6245, pp. 255 LP - 260, Jul. 2015, doi: 10.1126/science.aaa8415.

[18] "AdaBoost Classifier in Python - DataCamp."
<https://www.datacamp.com/community/tutorials/adaboost-classifier-python> (accessed Jun. 05, 2021).

[19] "How to Develop a Random Subspace Ensemble With Python."
<https://machinelearningmastery.com/random-subspace-ensemble-with-python/> (accessed Jun. 05, 2021).

[20] "KNN Classification using Scikit-learn - DataCamp."
https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn?utm_source=adwords_ppc&utm_campaignid=898687156&utm_adgroupid=48947256715&utm_device=c&utm_keyword=&utm_matchtype=b&utm_network=g&utm_adpostion=&utm_creative=332602034352&utm_targetid=aud-299261629574:dsa-429603003980&utm_loc_interest_ms=&utm_loc_physical_ms=20880&gclid=Cj0KCQjw--GFBhDeARIsACH_kdauP1-

JkBo1gPf26ducBzYyTBfkdQEP60eWd3hPUno8rJV69UujKuoaArcCEALw_wcB (accessed Jun. 05, 2021).

[21] “UCI Machine Learning Repository: Diabetes Data Set.”
<https://archive.ics.uci.edu/ml/datasets/diabetes> (accessed Jun. 05, 2021).

Anexos

Anexo 1: Código do classificador *AdaBoost Tree*.

```
AdaBoost_class.py × BaggedTree_class.py × SubspaceKNN_class.py ×
1  # -*- coding: utf-8 -*-
2  """
3  Spyder Editor
4
5  This is a temporary script file.
6  """
7  import pandas as pd
8  from sklearn.metrics import classification_report
9  from sklearn.metrics import f1_score
10 from sklearn.metrics import confusion_matrix
11 from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
12
13 #Boosted Tree
14 from sklearn.ensemble import AdaBoostClassifier
15
16 #Ler datasets:
17 data = pd.read_csv("P68_treino.csv", sep=';')
18 new = pd.read_csv("P68_teste.csv", sep=';')
19
20 #Extrair inputs e target:
21 X = data.iloc[:,0:12] #independent columns
22 y = data.iloc[:, -1]  #target column i.e price range
23 Xnew_p = new.iloc[:,0:12]
24 ynew_p = new.iloc[:, -1]
25
26 #Construir modelo:
27 Adaboost = AdaBoostClassifier(n_estimators=80, learning_rate=1, algorithm='SAMME')
28 modelo = Adaboost.fit(X, y)
29 prediction=modelo.predict(Xnew_p)
30
31 #Avaliação do modelo:
32 confusion = confusion_matrix(ynew_p, prediction)
33 print('Confusion Matrix\n')
34 print(confusion)
35 print('\nAccuracy: {:.2f}\n'.format(accuracy_score(ynew_p, prediction)))
36 print(classification_report(ynew_p, prediction))
37
```