FACULDADE DE CIÊNCIAS DA UNIVERSIDADE DO PORTO

TRABALHO PRÁTICO DE INTELIGÊNCIA ARTIFICIAL - *MACHINE LEARNING*

Grupo 34

Joana Pinto (201905100)

Valentina Wu (201907483)

Junho 2021

Índice

1) Resumo

Neste trabalho foi abordado o tema machine learning, analisando os dados fornecidos e tratamo-los para serem avaliados/classificados através de dois algoritmos: decision tree ID3 e Naïve Bayes. Utilizamos duas métricas para avaliar o resultado de ambos os algoritmos, que foram: matriz de confusão e *cross-validation*, destas recolhemos os valores de *precision*, *accuracy* e *recall*, retirando daí as conclusões.

2) Introdução: descrição breve do problema e da metodologia, linguagem de programação e bibliotecas usadas

Para o projeto foi nos fornecidos dados de um estudo realizado pela *Columbia University*, no qual foram recolhidos dados sobre os participantes em eventos experimentais de *speed dating, durante os anos 2002 e 2004*. O projeto teve como objetivo a criação e avaliação de modelos de classificação utilizando dois algoritmos de *Machine Learning*, que são *Decision tree* e o *Naive Bayes* [1].

Machine Learning (ML) é um método de análise de dados que melhora automaticamente com experiências e uso de exemplos de dados, conhecidos como, "*Training Data*". Este mecanismo usa os seus erros e experimentações para fazer previsões sobre o conjunto de dados. Inteligência artificial beneficiou da evolução de ML, o seu elevado valor probabilístico permitiu a criação de avaliador linear.

Machine learning é influenciada pelo algoritmo de decision tree pois este cobre classificação e regressão.

Usamos o algoritmo de *decision tree Iterative Dichotomiser 3,* também conhecido com ID3, divide iterativamente os meios em dois ou mais grupos a cada iteração.

Aplicamos também o algoritmo *Naive Bayes*, que se baseia do teorema de *Bayes*, supondo a independência entre os recursos.

A linguagem utilizada foi python e as bibliotecas para resolver o problema deste projeto foram o pandas e sklearn:

Pandas: biblioteca onde podemos fazer leitura dos ficheiros e analisar de dados e que nos fornece estruturas e operações para manipulação de tabelas numéricas e series temporais.

Sklearn: uma biblioteca especializada para machine learning onde estão contidos os algoritmos de decision tree e de Naive Bayes, como também as métricas para analisarmos o algoritmo [2].

3) Algoritmos: descrição dos algoritmos usados

As decision trees são um algoritmo que cria um modelo que prevê o valor de uma variável destino, aprendendo através das regras de decisão simples implícitas pelos recursos de dados [3]. Estas também podem ser usadas para visualizar e representar decisões, e possíveis decisões a ser tomadas.

O algoritmo ID3 usa uma abordagem de cima para baixo para a criação de uma *decision tree*, ou seja, inicia se a construção da arvore no topo e a cada iteração seleciona se o melhor meio nesse momento para produzir um nó. Normalmente este algoritmo é usado exclusivamente para classificação de problemas de formas nominais [4].

O algoritmo *Naive Bayes*, se funda no teorema de *Bayes* que é uma fórmula matemática usada para calcular probabilidades condicionais:

$$P(A|B) = \frac{P(B|A)\,.\,P(A)}{P(B)}$$

P(A|B) é a probabilidade posterior: Probabilidade de hipótese A no acontecimento B analisado.

P(B|A) é a probabilidade de verossemelhança: Probabilidade da evidência dado que a probabilidade de uma hipótese é verdadeira.

P(A) é Probabilidade Prévia: Probabilidade de hipótese antes de observar as provas.

P(B) é probabilidade marginal: probabilidade de evidência.

É um algoritmo classificador probabilístico, logo prevê conforme a probabilidade de um acontecimento através de analise de recursos. O nome *naïve* tem o significado que uma determinada característica é independente da ocorrência de outras ocorrências, isto é, uma característica contribui individualmente para a identificação de um certo recurso [5].

4) Análise exploratória dos dados: descrição do conjunto de dados, análise das variáveis e descrição dos passos de pré-processamento

Sobre os dados, temos um conjunto de 13 atributos, tais que estão divididos em duas categorias, que se denominam de numéricas e categóricas.

Nas numéricas temos atributos como o age, age_o, int_corr, no que diz respeito o like e o prob, vamos tratar como atributos numéricos uma vez que é uma escala de [0,10].

	Média	Mediana	Desvio padrão	Máximo	Mínimo
Age	26.36	26	3.567	55	18
Age_o	26.36	26	3.564	55	18
Int_corr	0.196	0.21	0.304	0.91	-0.83
Like	6.134	6.0	1.841	10	0
Prob	5.208	5.0	2.130	10	0

Nas categóricas temos: goal, date, go_out, length, met, like, prob

	Moda
goal	1
date	6
go_out	2
length	1
met	0
like	7
prob	5

Goal:

	Frequência
Passar uma noite divertida (1)	3426
Conhecer novas pessoas (2)	3012
Conseguir um encontro (3)	631
Procurar um relacionamento sério (4)	301
Dizer que consegui (5)	510
Outro (6)	419

Date:

	Frequência
Várias vezes por semana (1)	94
Duas vezes por semana (2)	308
Uma vez por semana (3)	783
Duas vezes por mês (4)	2040
Uma vez por mês (5)	1528
Várias vezes por ano (6)	2094
Quase nunca (7)	1434

Go_out:

	Frequência
Várias vezes por semana (1)	2610
Duas vezes por semana (2)	2990
Uma vez por semana (3)	1940
Duas vezes por mês (4)	450
Uma vez por mês (5)	164
Várias vezes por ano (6)	99
Quase nunca (7)	37

Length:

	Frequência
Demasiado curta (1)	4227
Demasiado longa (2)	177
Adequada (3)	3059

Met:

	Frequência
Conhecer o par anteriormente (1)	3956
Não conhecer (0)	4047

O atributo match é o atributo objetivo sendo 1 como aceite e 0 com não aceite.

	Frequência
0	6998
1	1380

Removemos as colunas id e partner, porque ao analisarmos vimos que havia id repetidos, mas que não tinha a necessidade de ser a mesma pessoa.

No passo de pré-processamento, tivemos duas propostas, a primeira foi tirar todos as linhas com valor NA e a segunda foi dependendo dos atributos substituir pela sua média se for numérico e pela sua moda se for categórico, utilizando a ajuda das funções da biblioteca pandas, o dropna() e o fillna(), respetivamente.

6) Experiências e resultados: descrição da metodologia experimental adotada, métricas usadas e resultados obtidos

As métricas usadas para tirar conclusões foram a da matriz de confusão e cross-validation [6], a matriz de confusão como o nome indica é uma matriz que tem o objetivo de cálculo de a quantidade de verdadeiro positivo, falso positivo, falso negativo e verdadeiro negativo, com funções existentes nas bibliotecas é possível calcular a precisão(*precision*) e o *recall* (encontrar os verdadeiros positivos). A métrica de *cross-validation* tem o objetivo de avaliar como o modelo se comporta, a nível geral, isto é possível através da divisão dos dados de treino e de teste em k subconjuntos.

Como optamos por fazer de duas maneiras diferentes o pré-processamento, iremos ter duas experiências:

6.1) Remoção de valores de NA

6.1.1) Decision Tree:

Para obter uma melhor árvore de decisão decidimos verificar a accuracy, a precision e o recall com diferentes parâmetros e naquele que der um melhor valor juntarmos e usar essa decision tree. Os parâmetros a analisar são o max_depth que é a profundidade máxima da árvore, min_samples_split, o número mínimo de amostras necessárias para dividir um nó interno, min_samples_leaf o número mínimo de amostras para estar num nó da folha, max_features o número de recursos a considerar.

Max_depth	accuracy	precision	recall
5	0.83284883	0.3936170	0.1138461
10	0.82025193	0.3466666	0.16
15	0.79844961	0.3425605	0.3046153
20	0.7810077	0.3180515	0.34153846
25	0.77470930	0.3044692	0.3353846

Min_samples_leaf	accuracy	precision	recall
5	0.789244186	0.3133333	0.2910216
10	0.7989341085	0.30833333	0.22910216
15	0.807655038	0.309278350	0.18575851
20	0.8231589147	0.376490508	0.1981424
25	0.83333333	0.40540540	0.1393188

Min_samples_split	accuracy	precision	recall
5	0.76453488	0.317679550	0.3248587
10	0.768410857	0.31097561	0.2881355
15	0.775678295	0.31772575	0.2683616
20	0.775193798	0.306338028	0.2457627
25	0.791666	0.324074074	0.19774011

Max_features	accuracy	precision	recall
2	0.770348837	0.31123919	0.3148688046
4	0.7858527130	0.3355481727	0.2944606413
6	0.7577519379	0.280112044	0.291545189
8	0.765988372	0.309782608	0.332361516
10	0.763565891	0.2991689750	0.314868804

Tendo em conta os precision e recall mais próximos e um accuracy mais elevado que possível decidimos escolher os parâmetros com os valores: max_depth=15, min_samples_leaf=10, min_samples_split=10, max_features=8

Logo, obtemos uma árvore de decisão, com a seguinte matriz de confusão:

	Positive	Negative
True	85	1600
False	120	259

Com uma precision = 0.41463414 e recall = 0.247093023

Com a métrica cross-validation, utilizando uma partição de k=5, obtemos 5 accuracy, tal que com a média obtemos um valor de accuracy melhor igual a 0.80284968232

Accuracy obtidas
0.800145348
0.780523255
0.827761627
0.7876363636
0.818181818

6.1.2) Naive_Bayes:

Com este algoritmo, conseguimos obter a seguinte matriz de confusão:

	positive	Negative
True	83	1595
false	109	277

Sendo o precision = 0.432291666 e o recall = 0.23055555

Tendo a média da accuracy com a métrica do cross-validation:

Accuracy
0.8139534883
0.7870639534
0.84156976744
0.80509090909
0.8276363636

Sendo assim, uma média de accuracy = 0.8150628936

6.2) Preenchimento dos valores NA

6.2.1) Decision Tree:

Como foi explicado na alínea anterior, iremos verificar os valores para obter melhores valores de diferentes parâmetros.

Max_depth	Accuracy	Precision	Recall
5	0.8349244232	0.4710743802	0.1397058823
10	0.82020684168	0.3735632183	0.1593137254
15	0.7863961813	0.2912621359	0.2205882352
20	0.76571201272	0.2860520094	0.2965686274
25	0.76054097056	0.273364485	0.4402233874

Min_samples_leaf	Accuracy	Precision	Recall
5	0.795147175	0.33333333	0.28109452
10	0.8237867939	0.39800995	0.199004975
15	0.833333333	0.45355191	0.20646766
20	0.830548926	0.44230769	0.22885572
25	0.831344470	0.44270833	0.211441786

Min_samples_split	accuracy	Precision	Recall
5	0.772076372	0.3266331658	0.2988505
10	0.779634049	0.341333333	0.29425287
15	0.779634048	0.3342618384	0.27586206
20	0.791567223	0.3449477351	0.22758620
25	0.797931583	0.357976653	0.21149425

Max_features	Accuracy	Precision	Recall
2	0.772474144	0.291469194	0.3106060606
4	0.774065234	0.30000	0.325757575
6	0.7688941925	0.295805739	0.338383838
8	0.7613365155	0.2649769585	0.29040404
10	0.7677008751	0.286363636	0.3181818181

Sendo assim, utilizamos os valores max_depth=20, min_samples_leaf=5, min_samples_split=10, max_features=4

Com esses valores, conseguimos com a ajuda do algoritmo *decision tree* obter a seguinte matriz de confusão

	Positive	Negative
True	103	1913
False	159	339

Obtendo o precision = 0.3931297709 e recall = 0.23303167420

Com o *cross-validation*, usando a métrica *KFold* adquirimos uma média de accuracy, com os seguintes valores obtidos

Accuracy
0.786992840
0.763723150
0.8007159904
0.7719402985
0.821492537

A média é igual a 0.788929638

6.2.2) Naives Bayes:

Com o algoritmo de *Naives Bayes* conseguimos chegar à seguinte matriz de confusão:

	Positive	Negative
True	106	1973
False	109	326

Sabendo assim a precision = 0.49302355 e recall = 0.2453703703

Tendo a média da accuracy = 0.8234676023 utilizando a métrica cross-validation:

accuracy
0.8066825776
0.8084725537
0.8436754176
0.8143283582
0.8441791044

6.3) Discussão critica dos resultados

Como podemos verificar os valores obtidos, quer nas precision, recall ou accuracy são bastantes parecidos em ambos os casos, mas existindo uma mínima de diferença.

Mas foi possível denotar que o algoritmo de *Naïves Bayes* tem uma “performance” melhor em ambos os casos de tratamento de dados, uma vez que este algoritmo trata as variáveis de uma forma independente, conseguimos observar que as variáveis só esta direcionada de uma pessoa para o seu partner e não obtemos informação de volta, o que seria importante de saber pois achamos que o match teria haver com o comportamento de duas pessoas, se conseguiríamos acrescentar mais colunas sobre o que o *partner* pensava em relação a pessoa, os resultados iriam ser mais confiáveis.

Sobre a diferença entre ambos os casos, retirar todos os valores NA ou substituí-los por valores, no algoritmo *decision_tree*, os resultados parecem melhor se retiramos os NA, pois a árvore como é dependente dos atributos e da sua relação apesar de substituirmos por valores mais comuns existe alguma diferença entre os valores originais e os substituídos, pois este algoritmo é sensível a pequenas mudanças, por isso, o primeiro caso obtém melhores resultados.

Sobre o algoritmo de *Naïves Bayes*, obtivemos melhores resultados após a substituição por valores, como os atributos são independentes, quanto mais informação tivermos melhor o algoritmo funciona.

7) Conclusões

Em suma, para a realização deste trabalho exploramos dois algoritmos de analise e classificação de dados, *decision tree ID3* e *Naïve Bayes*. Ambos os algoritmos têm vantagens e desvantagens, particularmente para o conjunto de dados fornecido para este projeto, o algoritmo Naïve Bayes mais eficaz pois permite tratar de uma maior quantidade de dados, visto que trabalha com a independência dos atributos, bem como avalia os dados descantando os irrelevantes. Já o decision tree ID3 foca-se na relação entre atributos, logo é mais eficaz no menor conjunto de dados, o que aqui não é o caso, e são sensíveis a pequenas mudanças no conjunto de treino e teste gerando árvores diferentes.

8) Referências bibliográficas

[1]https://en.wikipedia.org/wiki/Machine_learning

[2] https://scikit-learn.org/

[3]https://scikit-learn.org/stable/modules/tree.html

[4] https://towardsdatascience.com/decision-trees-for-classification-id3-algorithm-explained-89df76e72df1

[5]https://www.kdnuggets.com/2020/06/naive-bayes-algorithm-everything.html

6]Como saber se seu modelo de Machine Learning está funcionando mesmo | by Paulo Vasconcellos | Paulo Vasconcellos—Cientista de Dados Brasileiro