Machine Learning: Decision Tree

Rúben Martins[1[202004531]] e Rúben Passarinho[2[20200095]]

[1] Universidade Europeia, IADE - Instituto de Artes Visuais, Design e Marketing, Av. Dom Carlos i 4, 1200-649
admissoes@iade.pt

Abstract. Machine Learning é um campo da inteligência artificial, sendo definido co-mo a capacidade de uma máquina imitar o inteligente comportamento hu-mano. Sistemas de inteligência artifical são utilizados para resolver tarefas complexas de uma maneira similar à que os seres humanos resolveriam. Para resolver estas tarefas podem ser utilizadas várias técnicas distintas. Neste artigo será abordada uma técnica especifica, as árvores de decisão (decision tree). RESULTADOS E CONCLUSOES

Keywords: Machine learning, Inteligência artificial, Decision tree..

1 Introdução

Para este artigo será explorada a técnica de Machine Learning decision tree. É classificada como sendo supervised learning, utilizando pares de dados para fazer previsões.
É uma das mais simples e mais sucedidas técnicas de machine learning, sendo possível visualizar o mapa de potenciais resultados para uma série de decisões. É recebido um vector de atributos e retornado um valor, uma decisão.
A árvore (tree) começa com um root node que tem a possibilidade de abrir vários child node. Os node vão sendo abertos de forma sucessiva, com base nos dados, até chegar à decisão final. Com isto é gerado uma estrutura muito semelhante a uma árvore.
Para o artigo foi utilizado como modelo uma abordagem da plataforma W3Schools, a referência estará no final do artigo.

2 Metodologia

2.1 Conjuntos de Dados e Parâmetros

Na realização deste artigo foi decidido tomar como exemplo uma ida normal a um cinema. Para tal, foram escolhidos os seguintes parâmetros:

- Time, horas a que será o filme
- Duration, duração do filme em minutos
- IMDb, rating do filme de 0 a 10
- Nationality, idioma do filme, português ou inglês
- GO, indica se a pessoa foi ao não ao cinema com certas condições

Com base nestes parâmetros foram fornecidas várias instâncias de da-dos, nomeadamente 20, estão indicadas na figura seguinte.

```
Time,Duration,IMDb,Nationality,Go
17,180,9,PT,YES
23,180,10,ENG,NO
23,120,9,ENG,NO
12,120,6,PT,YES
10,120,8,PT,NO
9,120,3,PT,NO
20,100,5,ENG,YES
13,200,10,PT,YES
8,100,10,PT,NO
12,60,4,PT,NO
15,150,5,PT,YES
7,90,2,ENG,NO
2,320,10,ENG,NO
18,140,9,PT,YES
23,68,4,ENG,YES
21,73,7,PT,YES
18,59,6,PT,YES
15,60,0,PT,NO
18,20,2,PT,YES
9,60,5,ENG,NO
```

Fig. 1. Conjunto de Dados

2.2 Implementação

No que toca a questões de implementação, foi escolhida a linguagem de programação Python. Com o mesmo é possível gerar uma árvore de decisões.

```
#3 linhas que permite ao compilador desenhar:
import sys
import matplotlib
matplotlib.use('Agg')

import pandas
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
import matplotlib.pyplot as plt

df = pandas.read_csv("data.csv")

#Os valores não numéricos são convertidos para numéricos
d = {'PT': 0, 'ENG': 1}
df['Nationality'] = df['Nationality'].map(d)
d = {'YES': 1, 'NO': 0}
df['Go'] = df['Go'].map(d)

#São separadas as colunas de features e targets
#Tentamos prever com base nas colunas de features e nos valores da de targets
features = ['Time', 'Duration', 'IMDb', 'Nationality']
X = df[features]
y = df['Go']

dtree = DecisionTreeClassifier()
dtree = dtree.fit(X, y)

#Permite ao compilador desenhar:
tree.plot_tree(dtree, feature_names=features)
plt.savefig(sys.stdout.buffer)
sys.stdout.flush()

#Permite efetuar o teste dos parâmetros
#print(dtree.predict([[9, 60, 7, 1]]))

#print("[1] Vai ao cinema!")
#print("[0] Não devias ir!")
```

Fig. 2. Implementação utilizada

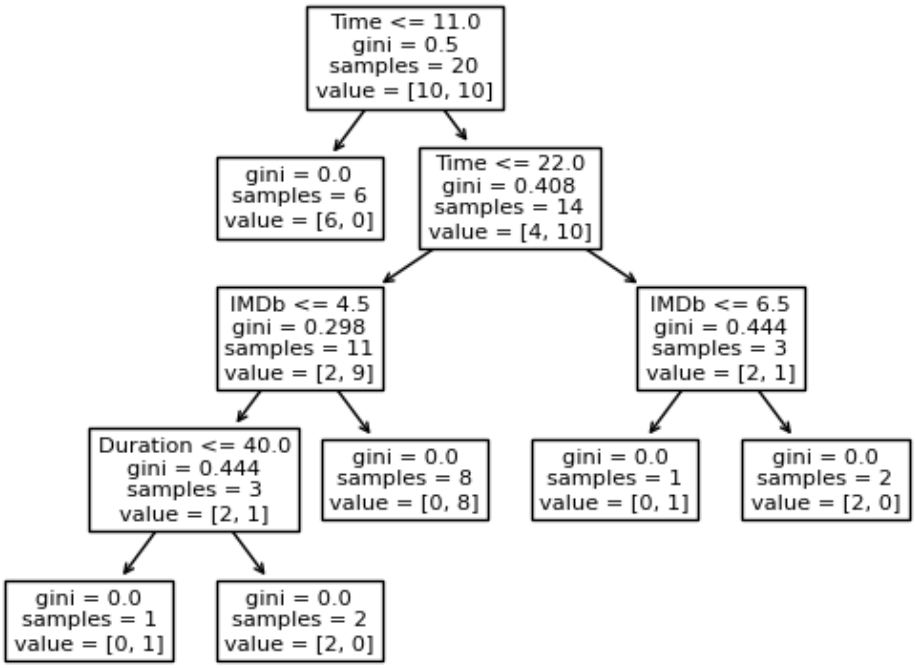

Fig. 3. Exemplo de árvore de decisões gerada

A árvore irá fornecer diferentes resultados se for "corrida" diversas vezes, mesmo sendo utilizados os mesmos dados. A árvore não fornece uma resposta 100% certa. É baseada numa probabilidade e a resposta irá variar.

O primeiro valor será o parâmetro a analisar, como por exemplo o “Time”. Gini refere-se à qualidade da divisão e é sempre um valor entre 0.0 e 0.5, onde 0.0 significa que todas as amostras obtiveram o mesmo resultado e 0.5 que a divisão é feita exatamente ao meio. Samples é a quantidade de amostras na atual decisão. No “value” o primeiro valor corresponderá à quantidade de “NO” dos dados, enquanto o segundo corresponderá ao “YES”.

2.3 Testes Efetuados

Depois de termos a nossa árvore formada e com diversos dados, foram efetuados diversos testes.

```
print(dtree.predict([[9, 60, 7, 1]]))
```

Fig. 4. Código utilizado para testes com os respetivos parâmetros, com dados teste

Foi utilizado esta linha de código de modo a efetuar os testes na árvore gerada. São utilizados os parâmetros descritos anteriormente, sendo eles a duração do filme (9), a duração do filme (60), o rating do filme (7) e a nacionalidade (1 para português ou 0 para Inglês).
O output gerado será o seguinte.

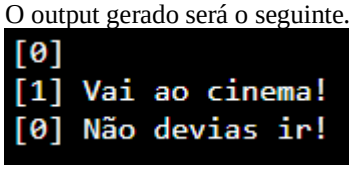

Fig. 5. Output gerado com o teste

Será mostrado se com base nos dados inseridos seria recomendado a pes-soa ir ou não ao cinema.

3 Resultados Obtidos

Depois de termos a nossa árvore criada e o nosso código de testes, podemos utilizá-la com diversos dados para testar.
Para uma primeira instância iremos utilizar os seguintes parâmetros.

```
print(dtree.predict([[9, 60, 7, 1]]))
```

Fig. 5. Dados do primeiro teste, 9 horas, 60 minutos, 7 de rating e English (1)

Foram testados 10 vezes os parâmetros, em todos eles o resultado foi "Não" (0).

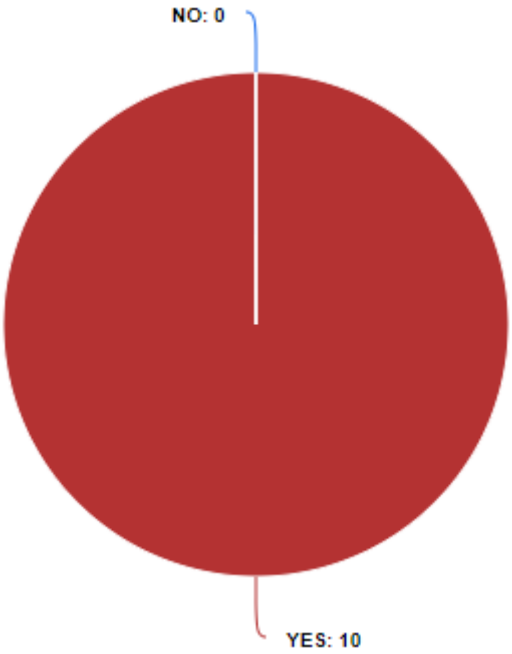

Fig. 6. Gráfico circular com os dados do primeiro teste

Para uma segunda instância foram utilizados os seguintes parâmetros

```
print(dtree.predict([[21, 500, 2, 1]]))
```

Fig 7. Dados do segundo teste, 21 horas, 500 minutos, 2 de rating e English (1)

Foram testados 10 vezes os parâmetros, ocorreu um split equivalente de 5 respostas “Sim” (1) e 5 respostas “Não” (2).

Fig. 8. Gráfico circular com os dados do segundo teste.

Para uma terceira instância foram utilizados os seguintes parâmetros

```
print(dtree.predict([[20, 120, 4, 0]]))
```

Fig 9. Dados do terceiro teste, 20 horas, 120 minutos, 4 de rating e português (1)

Foram testados 10 vezes os parâmetros, ocorreu um split de 2 respostas “Sim” (1) e 8 respostas “Não” (2).

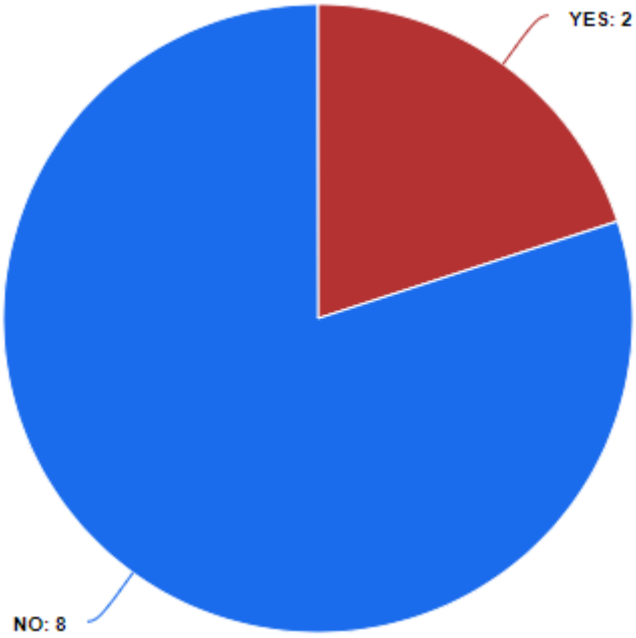

Fig. 10. Gráfico circular com os dados do terceiro teste

4 Conclusão

Árvores de decisão são um algoritmo muito útil para machine learning, sendo simples de compreender e ao mesmo tempo bastante eficazes. Possibilitam às empresas comparar possíveis outcomes e com isso decidir o melhor resultado com base nos parâmetros dados. Foram feitos vários testes conseguindo outcomes diferentes, sendo possível perceber bem o algoritmo.

5 Referências

6. W3Schools, https://www.w3schools.com/python/python_ml_decision_tree.asp, último aces-so a 23/12/2022.
7. Spiceworks, https://www.spiceworks.com/tech/artificial-intelligence/articles/top-ml-algorithms, último acesso a 23/12/2022.