Data exploration and enrichment for supervised classification

Video Games

Francisco Tavares 202205119

Rodrigo Batista 202206259

Introdução

- A ciência de dados tem desempenhado um papel fundamental na previsão e análise de dados em diversas áreas. Esta, a partir de uma grande base de dados, fornece-nos detalhes valiosos que poderão ser usados para prever possíveis resultados.

- Neste trabalho foi analisado um dataset de cerca de 6000 jogos com o objetivo de prever se os utilizadores deram ao jogo uma pontuação bad, medium, good ou great.

- Divimos o trabalho por etapas começando por uma análise e organização dos dados, passando depois para a análise exploratória destes através de gráficos, com o intuito de depois fazer a implentação dos modelos, os quais serão treinados e melhorados de forma a realizar melhores previsões das avaliações dos jogos.

Dataset

- O nosso dataset tem as seguintes características:

- 6000 jogos
- O nosso dataset apresenta um tamanho médio tendo em conta o seu contexto
- Possui uma dimensionalidade de 15 colunas:

Categorica: Name
Categorica: Category
Categorica: Users rating
Booleana: In a franchise
String: Genre
String: Platform
String: Companies
String: Summary

Identificador: Id
Numerica: Number of DLCs
Numerica: Number of expansions
Numerica: Release year
Numerica: Follows
Numerica: Average users score
Numerica: Number of reviews

- Problemas encontrados:

- Células nulas (21 'genres' 43 'companies');
- Colunas pouco relevantes para o nosso objetivo final ('summary', 'user_score').

	Number:	Percentage:
great:	3153	54.73%
good:	2500	43.4%
medium:	97	1.68%
bad:	11	0.19%

Figura 1 – " % Classificações "

Análise dos dados

- Numa parte inicial começamos por analisar e organizar os dados. Retiramos as colunas “summary” e “user_score” bem como as células sem dados (Nan values) nomeadamente 21 no parâmetro “genres” e 43 no “companies” (cerca de 0.08%).

- No passo seguinte passamos para uma análise exploratória do dataset onde realizámos vários gráficos (entre eles um scatterplot, um heatmap e violinplot) de maneira a perceber, inicialmente, as relações entre cada variável e posteriormente focando nos na relação entre o ‘user_rating’ e as outras variáveis.

- Um dado curioso que observamos através da figura 3 foi que todos os jogos têm pelo menos 11 reviews dadas pelos utilizadores.

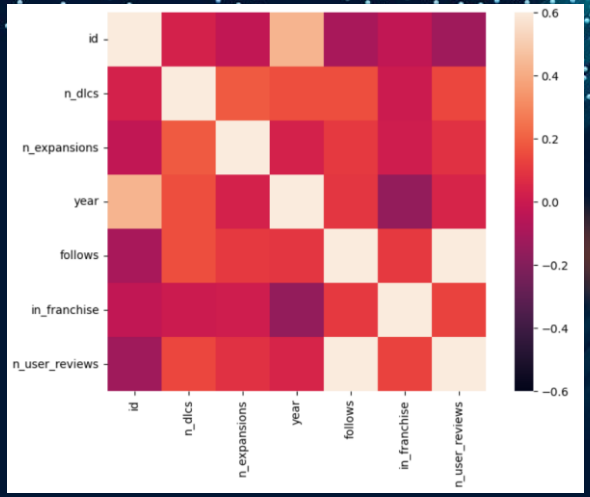

Figura 2 – “ HeatMap “

	id	n_dlcs	n_expansions	year	follows	user_score	n_user_reviews
count	5824.000000	5824.000000	5824.000000	5824.000000	5824.000000	5824.000000	5824.000000
mean	18508.000515	0.450721	0.120707	2007.293784	35.428056	75.115510	86.186470
std	30546.018723	2.150920	0.811236	9.342881	85.026280	10.197815	194.781917
min	1.000000	0.000000	0.000000	1971.000000	0.000000	10.266748	11.000000
25%	2271.750000	0.000000	0.000000	2001.000000	5.000000	69.937370	16.000000
50%	7258.000000	0.000000	0.000000	2009.000000	11.000000	76.258903	28.000000
75%	18581.250000	0.000000	0.000000	2015.000000	28.000000	81.695739	67.000000
max	178351.000000	44.000000	29.000000	2022.000000	1766.000000	99.738172	3369.000000

Figura 3 – “ Resumo estatístico do Dataset “

Análise dos dados (continuação)

- Com os gráficos que obtivemos conseguimos tirar algumas conclusões como:

- A relação entre a franquia e o número de seguidores;

- Os jogos com mais expansões são aqueles que têm classificações mais elevadas;

- Se um jogo pertence a uma franquia, a sua classificação tende a ser mais elevada do que os jogos que não pertencem a franquias;

- Com o passar dos anos, os jogos foram recebendo classificações mais elevadas assim como o número de expansões e dlcs foi aumentando.

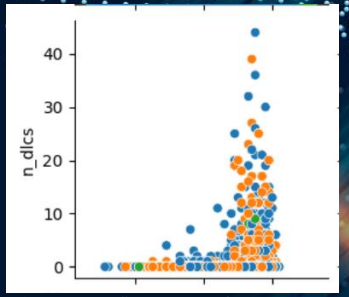

Figura 4 – " Scatterplt que relaciona o nº de dlcs com o ano"

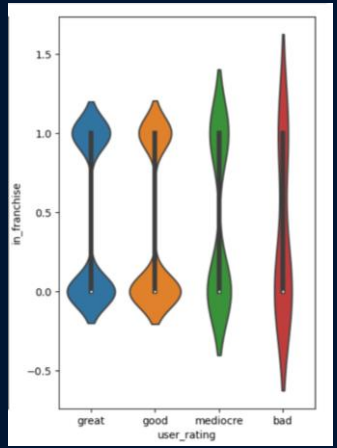

Figura 5 – " Violinplot que relaciona a classificação com a franquia"

Classificação dos dados

- Nesta etapa procedemos à classificação dos dados.
- Primeiramente definimos a variável ‘user_rating’ como **target** e as restantes como **features**.
- De seguida, procedemos à divisão dos dados em dois conjuntos, ***train*** e ***test***. O conjunto test terá um tamanho de 15% em relação ao conjunto train.
- Um cuidado a ter neste passo é não ter um conjunto de test de tamanho elevado pois poderá ocorrer overfitting.
- Usando uma função da biblioteca sklearn, podemos ver um exemplo do conjunto ‘target_test’.

```
target_test.value_counts()
```

```
user_rating
great          509
good           338
mediocre        12
bad              6
dtype: int64
```

Figura 6 – “ Conjunto target_test “

Implementação dos modelos

- Na quinta etapa, avançamos para a implementação dos modelos. Decidimos implementar o DecisionTreeClassifier, o K-NearestNeighbors e o RandomForestClassifier.

- Para cada modelo, foram feitos quatro testes:

  - 1º Teste- Para este primeiro teste começamos por aplicar o algoritmo usando apenas as 6 variáveis numéricas.

  - 2º Teste- Para o segundo teste transformamos as restantes variáveis em numéricas, usando o LabelEncoder e usamos assim 13 variáveis.

  - 3º Teste- No terceiro teste decidimos dividir o dataset em 4 grupos iguais, tendo em conta o user_rating.

  - 4º Teste- Com o intuito de melhorarmos a accuracy, neste teste retirámos o 'n_user_reviews'.

  - Após a realização dos 4 testes foi feito um Parameter Tuning em cada modelo, de forma a maximizar os resultados obtidos.

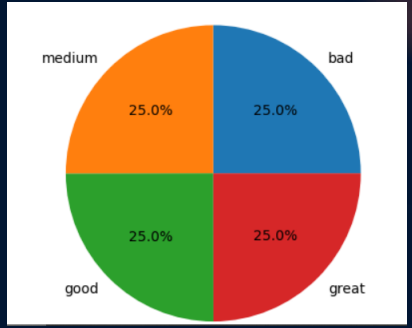

Figura 7 – " Divisão das classificações em 4 grupos iguais "

Análise e discussão dos resultados

Average score	Decision Tree	K-Nearest Neighbors	Random Forest
Teste 1	57,6%	59,4%	63,3%
Teste 2	59,0%	55,7%	67,4%
Teste 3	34,9%	29,8%	41,3%
Teste 4	58,8%	55,1%	66,6%
Final	60,0%	55,7%	62,2%

Pelos dados obtidos nos testes podemos concluir:

- O modelo com melhor accuracy foi o RandomForest;
- O segundo teste foi aquele que obteve melhor desempenho, menos no KNN;
- O terceiro teste foi o que obteve pior desempenho em todos os modelos.

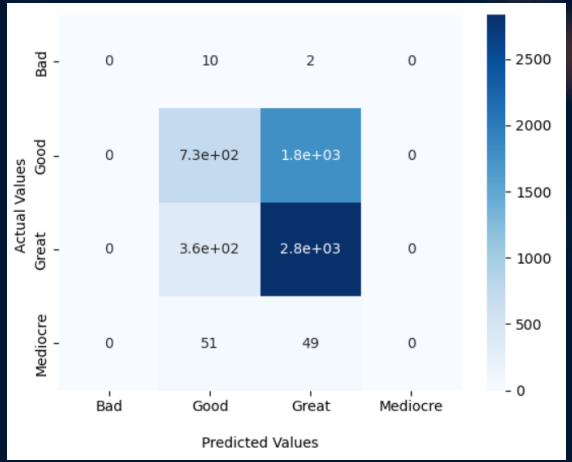

Figura 8 – “ Matriz confusão DecisionTreeClassifier“

Análise e discussão dos resultados (continuação)

- Depois de executados os testes em todos os modelos tirámos as seguintes conclusões:

  - No modelo DecisionTree no 4ºteste, obtivemos um aumento percentual de 250% em relação ao valor mais alto da CrossValidation até aquele momento;
  - No modelo KNN no 2ºteste, houve uma redução percentual de 7%;
  - Em todos os modelos foi notória uma redução percentual significativa no 3ºteste (de 40%);
  - As diferenças entre o 1º e 2º teste não foram muito notórias (aumento percentual de 4/5 %).

		Our Percentage:	Real percentage:
0	great:	81.06%	54.73%
1	good:	18.94%	43.4%
2	medium:	0.0%	1.68%
3	bad:	0.0%	0.19%

Figura 8 – “ % DecisionTree “

		Our Percentage:	Real percentage:
0	great:	60.08%	54.73%
1	good:	39.92%	43.4%
2	medium:	0.0%	1.68%
3	bad:	0.0%	0.19%

Figura 9 – “ % K-NearestNeighbors “

		Our Percentage:	Real percentage:
0	great:	55.48%	54.73%
1	good:	42.82%	43.4%
2	medium:	1.51%	1.68%
3	bad:	0.19%	0.19%

Figura 10 – “ % RandomForestClassifier “

Conclusão

- Com base na análise dos resultados obtidos, chegamos à conclusão de que o modelo RandomForestClassifier foi o melhor para o conjunto de dados fornecido.
- Apesar deste ter sido o que se saiu melhor, é de evidenciar que o tempo de execução é bastante elevado.
- Algo a ressaltar neste trabalho foi também a importância da análise do dataset. Tanto a eliminação de células nulas, como a eliminação de colunas que não estão diretamente ligadas ao objetivo final, são essenciais para obter uma alta precisão e acima de tudo resultados confiáveis.
- Em suma, através deste trabalho conseguimos concluir que o melhor modelo para o dataset fornecido é o RandomForestClassifier, e que a análise e tratamento dos dados é tão ou mais importante como a implementação do modelo.

Bibliografia

- http://localhost:8888/notebooks/OneDrive/Ambiente%20de%20Trabalho/2%C2%BA%20semestre/EIACD/data%20science/iris/Exercise5_IART_Notebook/Exercise5_IART1.ipynb#Step-5:-Classification
- https://moodle.up.pt/pluginfile.php/177966/mod_resource/content/0/Pandas_Cheat_Sheet.pdf
- https://moodle.up.pt/pluginfile.php/177968/mod_resource/content/0/Scikit_Learn_Cheat_Sheet_Python.pdf
- https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.mstats.mquantiles.html
- Chat GPT