Trabalho Prático 2: Video Games

FCUP

Elementos de Inteligência Artificial e Ciência de Dados

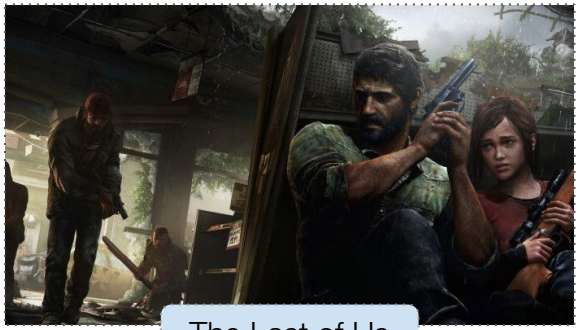
2º. Semestre 2022/23

Gonçalo Esteves (up202203947)

Nuno Gomes (up202206195)

Machine Learning Problem

God of War



The Last of Us

Atualmente, dada a elevada **competitividade** na indústria dos jogos eletrónicos, torna-se importante, para uma empresa, **prever o sucesso** adjacente ao lançamento do seu produto no mercado. Um dos principais aspetos a ter em conta nesta previsão são as ***reviews*** dos consumidores (*user_rating* subdividido em: *bad*, *mediocre*, *good* e *great*). Deste modo, obtém-se o seguinte problema de classificação:

“ Qual o *user_rating* do vídeo jogo? ”

De forma a resolvê-lo, foi-nos fornecido um dataset com dados de cerca de 6000 video jogos. Partindo destes, foi-nos possível realizar tanto a sua **análise exploratória** como a respetiva ***feature engineering***.

Por fim, após esta manipulação e tratamento de dados, apenas resta criar um **modelo preditivo** capaz de classificar qualquer vídeo jogo e avaliar a sua ***performance***.

Data Preprocessing | DataSet - Size & Problems

O DataSet é composto por **5824 entradas** às quais correspondem **15 atributos** diferentes:

→ ***name***

→ ***category***

→ ***genres***

→ …

Inicialmente, foi necessário analisar as possíveis **entradas nulas** (ou NaN values) e **entradas repetidas:**

Existiam 72 Entradas NaN / Null

Desta forma, dada a **quantidade extensa de dados** existentes no DataSet e ao número reduzido destas entradas, somos capazes de **removê-las sem afetar** significativamente os resultados a obter pelos **modelos preditivos**

Data Preprocessing | DataSet - Problems

Mais ainda, como o DataSet apresenta atributos que não irão influenciar o **desempenho** dos modelos preditivos, então estes poderão ser removidos:

→ ***id, summary*** e ***user_score***

Inicialmente foram removidos 3/15 atributos

Posteriormente, já com o DataSet mais reestruturado, criaram-se **novos atributos** (***Feature Engineering***) partindo de outros previamente existentes (Ex: A partir do atributo ***year*** foi criado o atributo ***age*** que reflete a idade de cada jogo).

Por fim, foi analisada a **correlação** entre os vários atributos o que levou à remoção dos que:

→ Proporcionavam uma **alta correlação** (+/- 1)

→ **Não eram úteis** para a restante análise exploratória

No Total foram removidos 6/19 atributos

Data Preprocessing | DataSet - In Suma

→ **Tratamento prévio dos Dados**:

- Remoção de **valores nulos, NaN e duplicados**
- Remoção de **colunas desnecessárias** (Ex: *summary*, *id, ...*) → Foram removidos **6** de **19** atributos trabalhados
- Conversão das colunas *genres*, *platforms* e *companies* (***string's***) para **listas** com os respetivos dados
- **Codificação** de Atributos (Ex: *category* e *in_franchise*) através do **One Hot Encoder** ou **Label Encoder**

→ ***Feature Engineering***:

- Criação de **4 Novas Colunas** (Ex: *age*, *n_genres*, *n_platforms*, *n_companies*) a partir de outras existentes

Predictive Models | Development & Evaluation

Através da Livraria ***ScikitLearn***, foi-nos possível implementar 2 Algoritmos de Aprendizagem computacional Supervisionados:

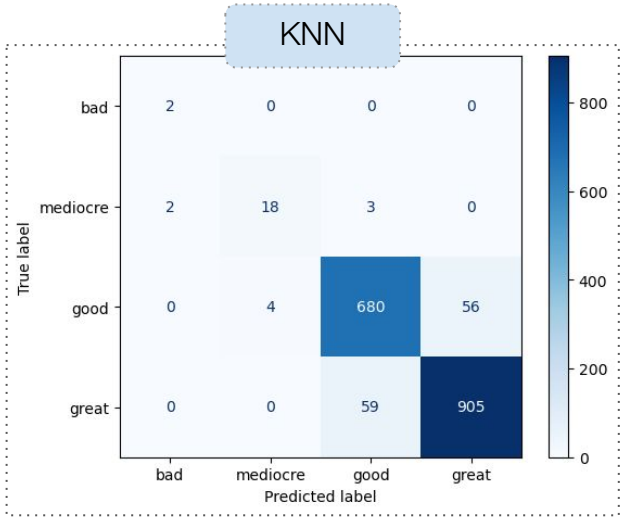
- **Decision Tree Classifier**
- **KNN Classifier**

No desenvolvimento destes modelos, é importante ter em conta vários parâmetros presentes nas **matrizes de confusão**:

- **Precision**
- **Sensitivity / Recall**
- **Average Accuracy**

Predictive Models | Confusion Matrices

As **matrizes de confusão** permitem avaliar a **performance** dos algoritmos de aprendizagem computacional supervisionados (**Decision Tree**, **KNN**) através dos parâmetros mencionados anteriormente.



Predictive Models | Precision

A **Precisão** permite identificar a frequência com que os dados certos no *set* de teste (para uma dada classe) foram previstos de forma correta, isto é:

$$Precision = \frac{TP}{TP + FP}$$

Deste modo, foram obtidos os seguintes **resultados experimentais**:

	Precision (%)
Decision Tree	28,8486
KNN	31,0275

Predictive Models | Sensitivity / Recall

A **Sensibilidade** é avaliada tendo em conta a proporção entre os dados previstos corretamente (forma positiva) e a quantidade total dos dados corretos (forma positiva) existentes no *set* de treino, isto é:

$$Sensitivity\ /\ Recall = \frac{TP}{TP + FN}$$

Assim, obteve-se:

	Sensitivity / Recall (%)
Decision Tree	28,9139
KNN	31,7242

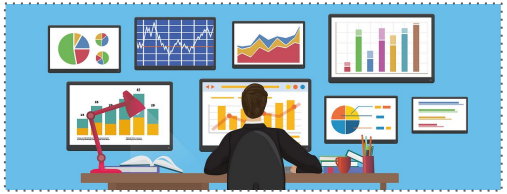
Predictive Models | Average Accuracy

	Average Accuracy (%) (K-Fold Cross Validation)
Decision Tree	53,7591
KNN	58,0857

A **Accuracy** avalia a frequência com que o modelo foi capaz de classificar corretamente os dados de teste, isto é:

$$Accuracy = \frac{TP + TN}{Total\ of\ Samples}$$

Porém, para obter um resultado conciso da **performance do modelo preditivo**, será necessário calcular a média de múltiplas Accuracies para **novos *set's* de treino e teste**, isto é, ir-se-á realizar uma **K-Fold Cross Validation**.



Conclusions

Tendo em conta os **Resultados Experimentais**, é-nos possível aferir uma ligeira superioridade do algoritmo **KNN** face à **Decision Tree** (relativamente à *Precision*, *Sensitivity* e *Accuracy*) ainda que nenhum tenha sido capaz de **prever** 100% dos casos (o que seria extremamente difícil)

Deste modo, podemos concluir que a **performance** de qualquer modelo preditivo advém principalmente tanto da **quantidade** como da **qualidade** dos **dados que lhe são fornecidos**