

Análise Preditiva



Grupo F:

Ana Pontes¹

Fábio Nunes²

Marta Pinhal³

¹up202005126@up.pt

²up202008145@up.pt

³up2020094372@up.pt

Mestrado em Ciência da Informação

Análise e Visualização de dados

2023/2024

Sumário

1	Introdução.....	6
2	Metodologia Utilizada.....	7
3	Compreensão do Negócio.....	8
3.1	Identificação do Problema	8
4	Identificação e Compreensão dos Dados.....	9
4.1	Descrição dos Atributos	9
4.2	Correlação entre Atributos	12
4.3	Qualidade dos Dados	14
5	Preparação dos Dados, Modelação e Avaliação	15
5.1	Preparação de dados	15
6	Modelação	20
6.1	Árvore de Decisão	21
6.1.1	Aplicação do Processo.....	22
6.1.2	Hiperparâmetros.....	22
6.2	Naïve Bayes	23
6.2.1	Aplicação do Processo.....	24
6.2.2	Hiperparâmetros.....	24
6.3	Redes Neurais Artificiais	25
6.3.1	Aplicação do Processo.....	25
6.3.2	Hiperparâmetros.....	26
6.4	Support Vector Machine	27
6.4.1	Aplicação do Processo.....	27
6.4.2	Hiperparâmetros.....	28
6.5	K-NN	29
6.5.1	Aplicação do Processo.....	29
6.5.2	Hiperparâmetros.....	30
7	Avaliação dos Resultados.....	31
7.1	Árvore de decisão	31
7.2	Naïve Bayes	32
7.3	Redes Neurais Artificiais	32
7.4	Support Vector Machine	33
7.5	K-NN	33

7.6	Comparação de Algoritmos.....	34
8	Conclusão	36
	Referências	37
	Anexos.....	38
	Anexo 1	38

Índice de Figuras

Figura 1 - Metodologia CRISP-DM	7
Figura 2 - Apresentação geral do dataset (parte 1)	9
Figura 3 - Apresentação geral do dataset (parte 2)	10
Figura 4 - Processo de Correlação no RapidMiner.....	12
Figura 5 - Resultado do Processo de Correlação	13
Figura 6 - Design da junção da tabela “games” e “recommendations”	15
Figura 7 - Lista de parâmetros dos papéis (“games” e “recommendations”)	16
Figura 8 - Lista de parâmetros dos atributos chave (“games” e “recommendations”).....	16
Figura 9 - Design da junção da tabela “games, “recommendations” e “users”	17
Figura 10 - Lista de parâmetros dos papéis (“games, “recommendations” e “users”)	17
Figura 11 - Lista de parâmetros dos atributos chave (“games, “recommendations” e “users”)	18
Figura 12 - Design da organização do título	18
Figura 13 - Rating	19
Figura 14 - Conversão de Nominal para Numérico.....	19
Figura 15 - Exemplo de Árvore de Decisão	21
Figura 16 - Primeiro Processo da Árvore de Decisão.....	22
Figura 17 - Segundo Processo da Árvore de Decisão.....	22
Figura 18 - Teorema de Bayes.....	23
Figura 19 - Primeiro Processo Naïve Bayes.....	24
Figura 20 - Segundo Processo Naïve Bayes	24
Figura 21 - Primeiro Processo Redes Neurais Artificiais	26
Figura 22 - Segundo Processo Redes Neurais Artificiais	26
Figura 23 - Primeiro Processo SVM.....	28
Figura 24 - Segundo Processo SVM.....	28
Figura 25 - Primeiro Processo K-NN.....	30
Figura 26 - Segundo Processo K-NN.....	30
Figura 27 - Matriz de Confusão da Árvore de Decisão	31
Figura 28 - Matriz de Confusão Naïve Bayes	32
Figura 29 - Matriz de Confusão Redes Neurais Artificiais.....	33
Figura 30 - Matriz de Confusão SVM	33
Figura 31 - Matriz de Confusão K-NN	34
Figura 32 - Árvore de Decisão.....	34
Figura 33 - Estatísticas relativas aos Atributos	38

Índice de Tabelas

Tabela 1 - Descrição dos Atributos	11
Tabela 2 - Quadro Comparativo de Modelos.....	20
Tabela 3 - Hiperparâmetros SVM	29
Tabela 4 - Valores de K no modelo K-NN.....	30
Tabela 5 - Comparação de Algoritmos.....	34

Resumo:

Este projeto tem como principal objetivo a análise preditiva de um dataset denominado “Game Recommendations on Steam” através da metodologia CRISP-DM. Neste dataset são encontradas informações acerca do rating, que possuem determinadas características, e conforme essas informações o principal objetivo é escolher o melhor modelo para prever o melhor rating de diferentes tipos de jogos. Para elaborar esta análise foram utilizados os modelos de classificação Árvore de Decisão, Naïve Bayes, Redes Neurais Artificiais, Support Vector Machine e K-NN e a partir da análise foi escolhido o modelo que melhor se adequava ao problema em questão.

Palavras-Chave: Análise de Dados, Análise Preditiva, RapidMiner, Árvore de Decisão, Naïve Bayes, Redes Neurais Artificiais, Support Vector Machine, K-NN, Jogos

Summary:

The main objective of this project is the predictive analysis of a dataset called “Game Recommendations on Steam” using the CRISP-DM methodology. This dataset contains information about the rating, which has certain characteristics, and according to this information the main objective is to choose the best model to predict the best rating for different types of games. In order to conduct this analysis, the classification models Decision Tree, Naive Bayes, Artificial Neural Networks, Support Vector Machine and K-NN were used, and based on the analysis, the model that best suited the problem in question was chosen.

Keywords: Data Analysis, Predictive Analysis, RapidMiner, Decision Tree, Naïve Bayes, Artificial Neural Networks, Support Vector Machine, K-NN, Games

1 Introdução

No âmbito da unidade curricular Análise e Visualização de Dados, lecionada no primeiro semestre do primeiro ano do Mestrado em Ciência da Informação, realizou-se o presente trabalho, com o objetivo de aplicar o conhecimento adquirido numa situação real através da metodologia CRISP-DM, com o propósito de aplicar os conhecimentos adquiridos sobre análise preditiva. Este trabalho baseia-se no dataset denominado "Game Recommendations on Steam", anteriormente utilizado na análise descritiva durante o Trabalho Prático 1. O nosso foco é agora elaborar uma análise preditiva dos ratings de jogos, explorando diferentes modelos de classificação.

O dataset "Game Recommendations on Steam" concentra informações cruciais sobre os ratings de jogos, sendo o nosso objetivo selecionar o modelo de classificação mais eficaz para prever o rating de diferentes tipos de jogos com base nas suas características específicas. Adaptando a metodologia CRISP-DM, as fases de modelação e avaliação dos resultados são ajustadas para atender às particularidades deste contexto.

Iniciamos o projeto compreendendo o domínio do negócio, identificando os objetivos e formulando o problema a ser solucionado. Em seguida, interpretamos os dados, descrevendo os atributos e assegurando a sua qualidade. A etapa seguinte envolveu a seleção de dados relevantes, passando por processos de limpeza, construção, integração e formatação para preparar o dataset para a fase de modelação.

Na modelação, optamos por cinco modelos preditivos: Árvore de Decisão, Naïve Bayes, Redes Neurais Artificiais, Support Vector Machine e K-NN. Após a aplicação e análise destes modelos, determinamos aquele que melhor se adequava à previsão de ratings de jogos. Posteriormente, realizamos uma avaliação detalhada do modelo escolhido, compreendendo a sua eficácia na previsão do rating para o problema em questão.

Ao concluir este trabalho, tentamos não só apresentar resultados precisos na análise preditiva de ratings de jogos, mas também extrair conclusões significativas que possam informar a aplicação prática destes modelos em contextos reais.

2 Metodologia Utilizada

A metodologia adotada neste trabalho prático é baseada na metodologia CRISP-DM. Esta metodologia é estruturada em fases distintas, tais como: compreensão do negócio, compreensão dos dados, preparação dos dados, modelação, avaliação e desenvolvimento, as quais serão detalhadamente abordadas de seguida.

A fase inicial, de compreensão do negócio, concentra-se na compreensão dos objetivos do negócio, realizando uma avaliação da situação para, a partir disso, definir um problema de análise de dados e elaborar um plano para superar esse desafio.

O próximo passo consiste na compreensão dos dados, onde ocorre a recolha do conjunto de dados a ser utilizado no projeto. Após a recolha, é fundamental descrever esses dados, examinando não apenas o volume, mas também as principais propriedades. Além disso, esta fase abrange a identificação da acessibilidade e disponibilidade dos atributos do conjunto de dados, descrevendo o tipo de atributos, intervalos e correlações entre os dados. Esses procedimentos visam verificar a qualidade dos dados recolhidos.

A terceira fase é dedicada à preparação dos dados, na qual, por meio do conjunto de dados, são selecionados e preparados os dados brutos iniciais que vão ser utilizados. Essa seleção inclui a correção e limpeza dos dados incorretos ou desnecessários para a resolução do problema em questão. Posteriormente, passamos à fase de modelação dos dados, na qual são selecionadas e aplicadas técnicas de modelação para enfrentar os problemas identificados. Em alguns casos, várias técnicas podem ser aplicáveis para resolver o mesmo tipo de problema, tornando inevitável a análise de todas as opções para determinar a mais adequada à situação em questão. Frequentemente, é necessário retornar à fase de preparação dos dados para viabilizar a utilização da técnica escolhida.

A fase subsequente é a avaliação dos dados gerados pelo modelo utilizado. Nessa avaliação, a ênfase recai sobre a qualidade por meio da análise dos dados, avaliando o modelo e, com base nessa análise, decidindo como utilizar os resultados obtidos para enfrentar os problemas identificados na fase de compreensão do negócio. Em seguida, realiza-se uma revisão minuciosa para verificar se o modelo atinge os objetivos e se há algum problema não identificado. Dependendo da revisão, são determinados os próximos passos, como a análise do potencial de cada resultado e a identificação de melhorias que podem ser implementadas no processo atual.

Como fase final, temos o desenvolvimento, cujo objetivo é aplicar o conhecimento adquirido na análise dos dados e compreender como esses modelos podem ser aplicados em contexto real.

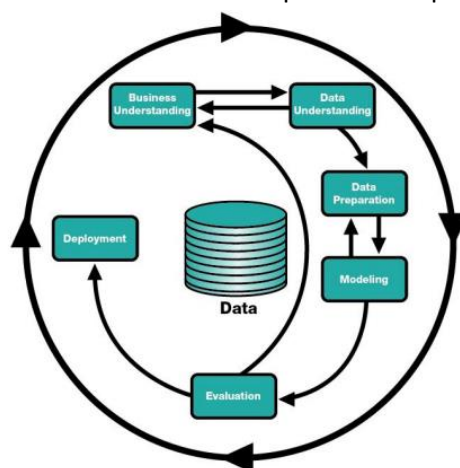


Figura 1 - Metodologia CRISP-DM

3 Compreensão do Negócio

Na fase da Compreensão do Negócio da metodologia *CRISP-DM*, o foco está na compreensão profunda dos objetivos e necessidades da organização. Durante esta etapa, realiza-se uma análise minuciosa da situação atual do negócio, avaliando os seus objetivos, desafios e contextos específicos. O objetivo é identificar claramente os problemas que podem ser abordados por meio da análise de dados.

Nesta etapa, o procurado é compreender a dinâmica operacional, os objetivos estratégicos e as questões-chave para o contexto da organização. Isso inclui a definição de um problema de análise de dados que seja relevante se dar resposta através do dataset para os objetivos globais da organização. No final da etapa da compreensão do negócio, espera-se ter uma visão clara do contexto organizacional e dos desafios que a análise de dados pode ajudar a resolver.

3.1 Identificação do Problema

O negócio a analisar diz respeito a uma empresa de plataforma online de jogos que detém um dataset cujos dados servem para uma boa e mais correta recomendação de jogos dependendo do perfil do utilizador. Assim, as recomendações de utilizadores, informações sobre jogos da Loja Steam, levou-nos à identificação de um potencial problema a: "Previsão de Popularidade Futura de Jogos".

Dado o conjunto de dados que inclui avaliações de utilizadores e informações sobre jogos, este problema preditivo possibilitava a previsão da popularidade futura de jogos na plataforma Steam. Isso envolveria a construção de um modelo que leva em consideração as características dos jogos, o histórico de avaliações, o feedback dos utilizadores e outras informações relevantes para prever quais jogos têm maior probabilidade de serem jogados, baseados nas avaliações positivas, no futuro. Essa previsão poderia ajudar a plataforma a destacar e promover jogos que provavelmente atrairão mais utilizadores, melhorando assim a eficácia do sistema de recomendação.

O objetivo é então, desenvolver um modelo preditivo capaz de analisar diversas características dos jogos (como preço, data de lançamento, título, plataforma onde é suportado, entre outros) e do comportamento dos utilizadores (avaliações, padrões de compra, etc.) para antecipar quais jogos terão maior adesão e popularidade entre os utilizadores no futuro.

A metodologia passaria pela:

- Identificação e compreensão de dados: Compreender as características e estruturas das tabelas nos conjuntos de dados.
- Agregação de Dados: Utilização de dados históricos da Steam, incluindo avaliações dos utilizadores, informações sobre jogos e perfis de utilizadores.
- Pré-processamento: Limpar e preparar os dados, identificando características relevantes para a previsão de popularidade.
- Modelagem Preditiva: Escolher algoritmos de *machine learning* adequados para criar um modelo preditivo com base nos dados preparados.

4 Identificação e Compreensão dos Dados

Nesta fase tornou-se essencial proceder a uma exploração mais detalhada dos dados para analisar.

O procedimento de compreensão de dados dividiu-se então em duas etapas: visualização e análise geral do dataset em excel e importação do dataset para o RapidMiner e exploração das estatísticas.

Sendo que tínhamos um elevado número de dados em cada uma das tabelas decidimos eliminar algumas linhas e seguirmos com 50873 linhas em cada uma das tabelas, sendo assim fizemos apenas uma análise de uma amostra do dataset inteiro por forma a ser possível analisarmos os dados nos computadores.

Ao fazer a junção das 3 tabelas (através do *inner join*): “games”, “Recommendations” e “users” conseguimos obter os seguintes dados para análise: 182 instâncias e 22 colunas, sendo que 19 destas correspondem a atributos do tipo: polinomial, número inteiro, data, real, e as outras 3 correspondem a diferentes ID’s.

4.1 Descrição dos Atributos

Antes da apresentação pormenorizada dos atributos, segue-se uma apresentação geral do dataset, assim sendo, na [Figura 2](#) e [Figura 3](#) é possível observar um excerto dos dados:

Open in

Turbo Prep

Auto Model

Filter (182 / 182 examples): all

Row No.	app_id	user_id	win	mac	linux	rating	is_recommended	title	date_release	positive_ratio	user_reviews	price_final	price_original	discount
1	346110	11457485	1	1	0	0	0	ARK: Survival Evolved	Aug 27, 2017 ...	83	495087	15	0	0
2	346110	6177428	1	1	0	0	0	ARK: Survival Evolved	Aug 27, 2017 ...	83	495087	15	0	0
3	346110	27882	1	1	0	0	1	ARK: Survival Evolved	Aug 27, 2017 ...	83	495087	15	0	0
4	1468860	5236314	1	0	0	0	0	Age of Empires IV: Anniversary Edition	Oct 28, 2021 ...	86	41462	40	0	0
5	270880	7977814	1	1	1	4	0	American Truck Simulator	Feb 2, 2016 1...	96	108202	20	0	0
6	270880	9120943	1	1	1	4	0	American Truck Simulator	Feb 2, 2016 1...	96	108202	20	0	0
7	1172470	5633784	1	0	0	0	0	Apex Legends‑S‑eason 1	Nov 4, 2020 1...	80	713182	0	0	0
8	107410	5298042	1	1	0	0	0	Arma 3	Sep 12, 2013...	91	154094	30	0	0
9	107410	13480540	1	1	0	0	0	Arma 3	Sep 12, 2013...	91	154094	30	0	0
10	107410	5624389	1	1	0	0	0	Arma 3	Sep 12, 2013...	91	154094	30	0	0
11	244210	1383854	1	0	0	0	0	Assetto Corsa	Dec 19, 2014...	92	79527	20	0	0
12	244210	4317627	1	0	0	0	0	Assetto Corsa	Dec 19, 2014...	92	79527	20	0	0
13	244210	5983710	1	0	0	0	0	Assetto Corsa	Dec 19, 2014...	92	79527	20	0	0
14	371970	8778538	1	1	1	0	0	Barony	Jun 23, 2015 ...	92	3713	20	0	0
15	602960	6701099	1	1	1	0	0	Barotrauma	Mar 13, 2023 ...	93	35639	24	0	0
16	284160	10765119	1	0	0	4	0	BeamNG.drive	May 29, 2015 ...	97	178635	25	0	0
17	284160	8119841	1	0	0	4	0	BeamNG.drive	May 29, 2015 ...	97	178635	25	0	0
18	620980	11359796	1	0	0	4	0	Beat Saber	May 21, 2019 ...	95	63695	30	0	0
19	582660	9247869	1	0	0	3	0	Black Desert	May 24, 2017 ...	76	49539	10	0	0
20	397540	4826886	1	0	0	0	0	Borderlands 3	Mar 13, 2020 ...	85	95243	60	0	0
21	397540	11129719	1	0	0	0	0	Borderlands 3	Mar 13, 2020 ...	85	95243	60	0	0
22	397540	13182821	1	0	0	0	0	Borderlands 3	Mar 13, 2020 ...	85	95243	60	0	0
23	1938090	8960193	1	0	0	2	0	Call of Duty‑Warzone	Oct 27, 2022 ...	59	429206	0	0	0
24	255710	11379804	1	1	1	0	0	Cities: Skylines	Mar 10, 2015 ...	93	178458	30	0	0

Figura 2 - Apresentação geral do dataset (parte 1)

helpful	funny	date	hours	review_id	products	reviews
0	0	Jul 3, 2017 1...	113.100	2310	367	2
0	0	Jun 19, 2015 ...	34.700	46278	67	2
3	0	Sep 26, 2017...	832.100	47070	300	8
0	0	Nov 1, 2021 1...	31.600	32486	317	11
0	0	Jan 5, 2022 1...	84.500	14524	73	3
0	0	Oct 15, 2022 ...	199.900	31809	25	5
3	0	Jun 8, 2021 1...	570	47368	365	1
0	0	Oct 31, 2018 ...	481.500	20131	65	12
0	0	Jul 1, 2019 1...	316.800	26624	175	13
0	0	Jan 15, 2018 ...	19.300	43867	140	15
0	0	Nov 9, 2013 1...	14.600	30122	100	9
0	0	Nov 26, 2019 ...	98.100	34275	47	1
0	0	Aug 4, 2021 1...	424.200	43463	16	5
0	0	Nov 1, 2021 1...	29.100	46477	49	3
2	0	Jun 5, 2019 1...	54.900	2025	583	65
0	0	Jun 17, 2015 ...	400.400	9086	156	5
0	0	May 6, 2020 1...	751.100	18561	5	1
0	0	Jan 31, 2020 ...	48.100	10206	717	6
0	2	Jul 16, 2018 ...	0.400	43507	78	5
0	0	Mar 29, 2020 ...	28	22841	46	1
0	0	Mar 18, 2020 ...	52.100	24057	1344	5
0	0	Nov 25, 2020 ...	118.100	24131	493	10
0	0	Nov 11, 2022 ...	118.900	21133	83	1
0	0	Nov 24, 2017 ...	35.800	14054	949	15

Figura 3 - Apresentação geral do dataset (parte 2)

Tendo em conta os atributos e respetivos dados de cada atributo, segue-se uma tabela com a explicação dos atributos após exploração dos mesmos:

Atributo	Descrição	Formato
App_id	Identificação única do jogo	Integer
User_id	Identificação única do user	Integer
title	Título do jogo	Polynomial
Date_release	Data de lançamento do jogo	Date
win	Se tem suporte no Windows	Polynomial
mac	Se tem suporte no MacOS	Polynomial
linux	Se tem suporte no Linux	Polynomial
rating	Classificação do jogo	Polynomial
Positive_ratio	Rácio de avaliação positiva	Integer
User_reviews	Nº de avaliações dos utilizadores	Integer
Price_final	Preço em dólares americanos \$ calculado após o desconto	Real
Price_original	Preço em dólares americanos \$ antes do desconto	Real
Discount	Percentagem de desconto do jogo	Real
Steam_deck	Se tem suporte na plataforma da Steam	Polynomial
Helpful	Nº de utilizadores que consideraram o jogo útil	Integer
Funny	Nº de utilizadores que consideraram o jogo divertido	Integer
Date	Data de disponibilização do jogo na steam	Date
Is_recommended	Se o jogo é recomendado ou não	Polynomial
hours	Nº de horas contabilizadas	Real
Review_id	Identificação única da classificação	Integer
Products	Nº de jogos que o utilizador possui	Integer
Reviews	Quantidade de vezes que o user classificou um jogo	Integer

Tabela 1 - Descrição dos Atributos

Esta descrição de atributos que possibilita que se obtenha uma primeira noção dos atributos mais relevantes para a análise, assim como dos que não adicionam informação relevante.

Posto isto, retiramos uma série de imagens estatísticas por forma a ser possível observar estatísticas referentes a cada atributo, como se pode observar no **Anexo 1**.

Ao analisarmos os dados a partir do RapidMiner é possível verificar que os atributos presentes são do tipo “Número inteiro”, “Número Real”, “Polinomial” e “Data”.

Posteriormente é feita uma análise estatística apresentando os dados sobre o formato de gráfico de barras, para os dados polinomiais, é indicado qual o valor com menor e maior ocorrência e a contagem final da ocorrência de cada valor. No caso dos dados em número inteiro é indicado o número mínimo e máximo, a média dos valores e o desvio. Para a data é indicada a data mais recente e data mais longínqua, bem como a duração.

Concluimos assim com esta análise dos dados que quanto à sua qualidade que não se verifica a presença de dados em falta, dados inconsistentes, dados redundantes, “noisy data” e outliers.

4.2 Correlação entre Atributos

Para uma compreensão mais aprofundada dos dados, é essencial investigar se os atributos estão inter-relacionados e, se assim for, compreender a relevância dessa relação para a nossa análise. Nesse sentido, iniciamos importando o conjunto de dados para o RapidMiner e aplicando a matriz de correlação. Para otimizar a análise, ajustamos parâmetros, como a seleção exclusiva de atributos quantitativos e o cálculo da “squared correlation”, em vez de apresentar apenas correlações simples.

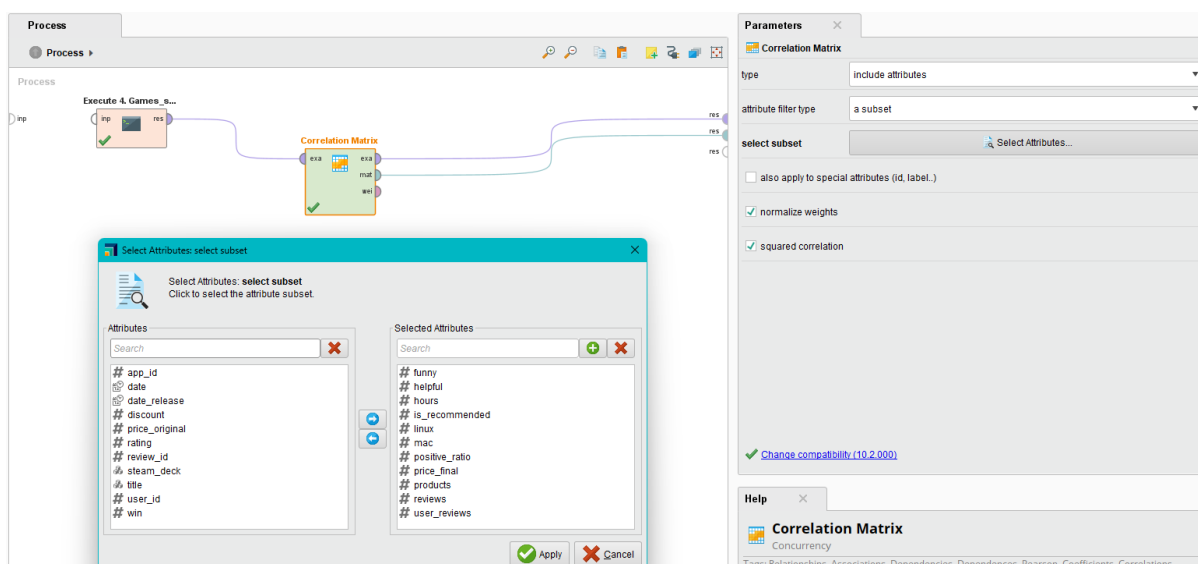


Figura 4 - Processo de Correlação no RapidMiner

Attributes	mac	linux	is_recommended	positive_ratio	user_reviews	price_final	helpful	funny	hours	products	reviews
mac	1	0.431	0.003	0.064	0.038	0.001	0.000	0.000	0.001	0.001	0.001
linux	0.431	1	0.003	0.029	0.078	0.000	0.004	0.000	0.005	0.000	0.002
is_recommended	0.003	0.003	1	0.044	0.001	0.002	0.011	0.001	0.015	0.009	0.000
positive_ratio	0.064	0.029	0.044	1	0.004	0.001	0.016	0.001	0.004	0.001	0.020
user_reviews	0.038	0.078	0.001	0.004	1	0.046	0.000	0.001	0.059	0.008	0.004
price_final	0.001	0.000	0.002	0.001	0.046	1	0.034	0.004	0.021	0.002	0.020
helpful	0.000	0.004	0.011	0.016	0.000	0.034	1	0.163	0.000	0.000	0.002
funny	0.000	0.000	0.001	0.001	0.001	0.004	0.163	1	0.003	0.000	0.001
hours	0.001	0.005	0.015	0.004	0.059	0.021	0.000	0.003	1	0.016	0.025
products	0.001	0.000	0.009	0.001	0.008	0.002	0.000	0.000	0.016	1	0.049
reviews	0.001	0.002	0.000	0.020	0.004	0.020	0.002	0.001	0.025	0.049	1

Figura 5 - Resultado do Processo de Correlação

Ao analisar a matriz de correlação é possível perceber sob o ponto de vista geral que os atributos não apresentam uma correlação entre si significativa.

A maior correlação que se pode verificar é entre "Mac" e "Linux", que é de 0,431, indica uma correlação positiva moderada entre a quantidade de sistemas operativos (Mac e Linux) compatíveis com os jogos. Existe ainda a relação entre o sistema operativo Windows (win) que não pode ser colocada na tabela de correlação, pois o mesmo têm uma taxa de compatibilidade com os jogos de 100%, sendo que este iria ter uma correlação de 1, podemos ainda assim concluir que o Windows é o sistema operativo mais usado pelos utilizadores que tem melhor compatibilidade com os jogos.

A segunda correlação mais alta que existe é entre "helpful" e "funny", com uma correlação de 0,163, sendo esta uma correlação positiva forte entre avaliações consideradas úteis e engraçadas.

As correlações positivas moderadas são as seguintes:

- "Linux" e "user_reviews" (0,078): Correlação positiva moderada entre a presença do sistema operativo Linux e o número de avaliações dos utilizadores. Verifica-se que alguns utilizadores recomendam o uso do sistema operativo Linux para um determinado jogo.
- "Products" e "reviews" (0,049): Correlação positiva moderada entre o número de produtos e o número de avaliações dos mesmos. Pode-se verificar que os jogos têm avaliações dos utilizadores.
- "price_final" e "helpful" (0,034): Correlação positiva moderada entre o preço final e se a avaliação foi útil. Verifica-se que o utilizador faz uma recomendação tendo em conta o jogo e o preço, sendo que estas demonstram-se úteis para os outros utilizadores que possam querer adquirir o jogo.
- "hours" e "reviews" (0,025): Correlação positiva moderada entre o número de horas jogadas e o número de classificações. Podemos verificar que os utilizadores ao jogarem um determinado número de horas fazem uma avaliação do jogo.
- "Is_recommended" e "hours" (0,015): Correlação positiva moderada entre a variável de recomendação e o número de horas jogadas. Quantas mais horas um jogador tiver no jogo irá fazer uma recomendação deste.

As correlações baixas e muito baixas que podemos encontrar foram as seguintes:

- "positive_ratio" e "user_reviews" (0,004): Correlação baixa entre o rácio de avaliação positiva e o número de avaliações dos utilizadores.
- "price_final" e "funny" (0,004): Correlação muito baixa entre o preço final e a variável de avaliações engraçadas.

- “Mac” e “Is_recommended” (0,003): Correlação muito baixa entre a presença de Mac e a variável recomendada. Podemos verificar que os utilizadores não recomendam usar o sistema operativo Mac para jogar.
- “price_final” e “Products” (0,002): Correlação muito baixa entre o preço final e o número de produtos.
- “positiveratio” e “price_final” (0,001): Correlação muito baixa entre o rácio de avaliação positiva e o preço final.

4.3 Qualidade dos Dados

A análise dos dados conduz-nos a um aspeto crucial: a sua qualidade, a qual desempenha um papel fundamental na possível influência sobre os resultados futuros. A qualidade dos dados revela-se na ausência significativa de elementos indesejáveis, tais como dados ruidosos, nulos, inconsistentes, redundantes, irregulares, valores em falta, *outliers*, entre outros.

Neste contexto, a nossa atenção concentrou-se predominantemente nestes aspetos, e é relevante salientar que não identificamos a presença dos referidos tipos de dados durante a nossa observação.

5 Preparação dos Dados, Modelação e Avaliação

5.1 Preparação de dados

A abordagem adotada para a preparação dos dados, conforme a Metodologia CRISP-DM aplicada neste relatório, incide sobre cinco pontos fundamentais: seleção, limpeza, construção, integração e formatação de dados. O propósito deste procedimento reside, primordialmente, na construção de um conjunto de dados final relevante a partir dos dados originais em estado bruto. Neste procedimento estão envolvidas tarefas como a escolha criteriosa dos dados que vão ser manipulados, a projeção das relações entre atributos para a resolução do problema apresentado na fase inicial, a preparação do conjunto de dados e a especificação do formato necessário para a análise subsequente.

Para fazer a junção das 3 tabelas foram realizados alguns processos que incluíram o dataset pretendido, o set role (para identificar o ID que era pretendido) e por fim o Join para agregar as tabelas tendo que na opção “key attributes” tivemos que identificar nos atributos chave da direita e da esquerda quais eram os pretendidos.

Posto isto realizou-se um processo inicial que iria juntar a tabela “games” com a tabela “recommendations”, para isso o set role de cada uma iria ser a “app_id”, pois é o ID comum dessas duas tabelas. Por fim foi então realizado um “join” para se juntar as duas tabelas sendo que queremos que este tenha em conta que para cada ID do jogo há uma recomendação diferente, posto isto é necessário identificar os atributos chave do lado direito e esquerdo por forma a nenhum ficar perdido.

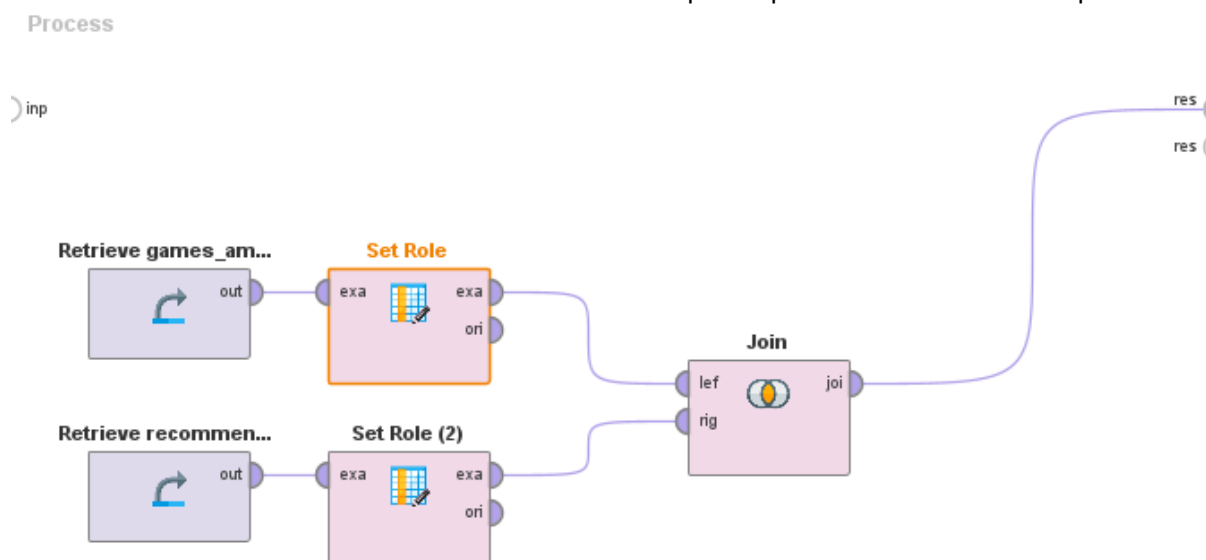
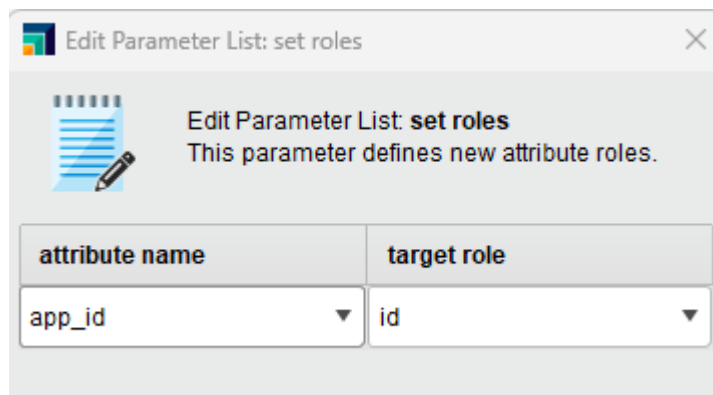


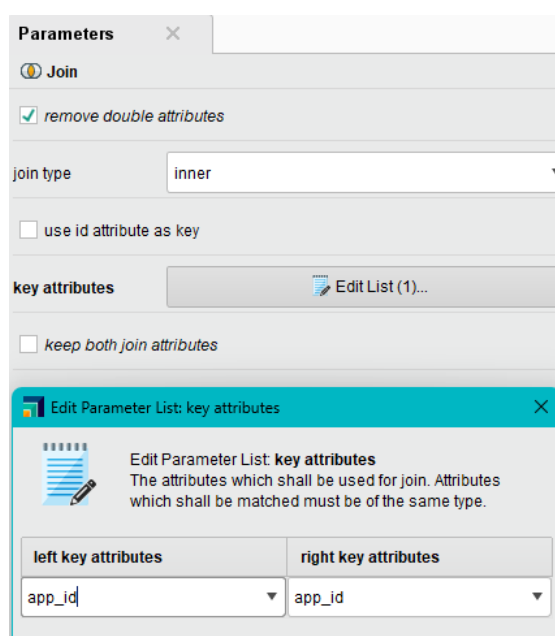
Figura 6 - Design da junção da tabela “games” e “recommendations”



Edit Parameter List: set roles
This parameter defines new attribute roles.

attribute name	target role
app_id	id

Figura 7 - Lista de parâmetros dos papéis (“games” e “recommendations”)



Parameters

Join

☒ remove double attributes

join type: inner

☐ use id attribute as key

key attributes [Edit List \(1\)...](#)

☐ keep both join attributes

Edit Parameter List: key attributes
The attributes which shall be used for join. Attributes which shall be matched must be of the same type.

left key attributes	right key attributes
app_id	app_id

Figura 8 - Lista de parâmetros dos atributos chave (“games” e “recommendations”)

Após a junção de duas tabelas criou-se um segundo processo que iria juntar o resultado inicial da junção das duas tabelas (“games” e “recommendations”) com a tabela “users”. Para isso o set role de cada uma iria ser a “user_id”, pois é o ID comum dessas tabelas. Por fim foi então realizado um “join” para se juntar as tabelas sendo que queremos que este tenha em conta que para cada ID do user tem um jogo diferente e uma classificação diferente, posto isto é necessário identificar os atributos chave do lado direito e esquerdo por forma a nenhum ficar perdido.

Process

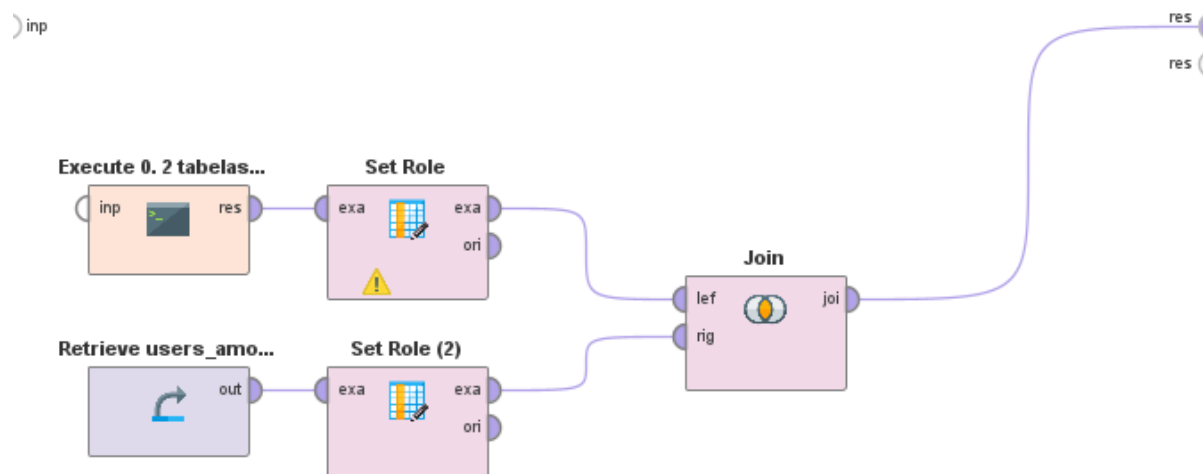


Figura 9 - Design da junção da tabela “games”, “recommendations” e “users”

The screenshot shows a dialog box titled 'Edit Parameter List: set roles'. It contains a text area with the text 'Edit Parameter List: set roles' and 'This parameter defines new attribute roles.' Below this is a table with two columns: 'attribute name' and 'target role'. The table has one row with 'user_id' in the 'attribute name' column and 'id' in the 'target role' column.

attribute name	target role
user_id	id

Figura 10 - Lista de parâmetros dos papéis (“games”, “recommendations” e “users”)

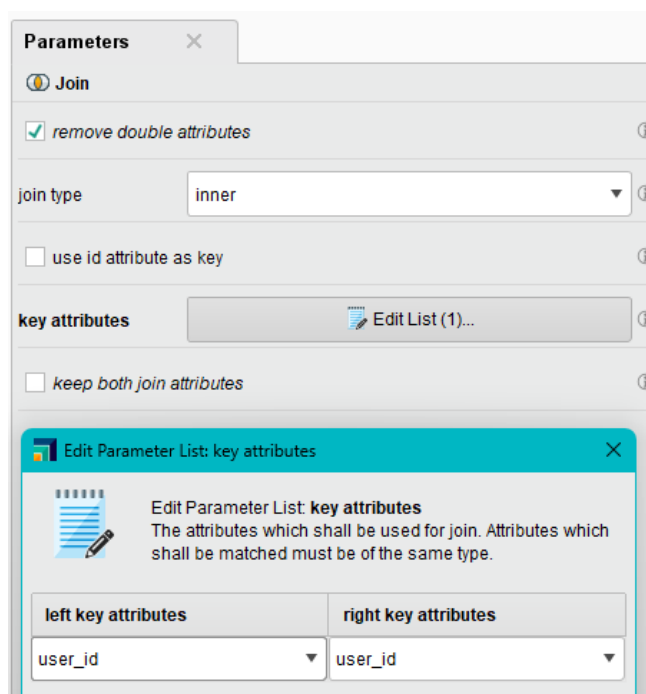


Figura 11 - Lista de parâmetros dos atributos chave ("games", "recommendations" e "users")

Após esses processos decidimos organizar o título dos jogos por ordem alfabética através do operador "sort".

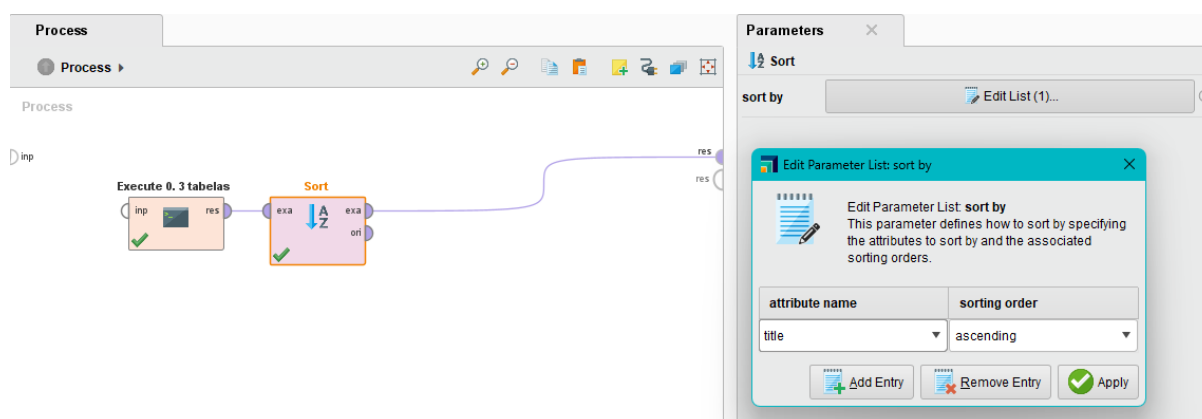


Figura 12 - Design da organização do título

No dataset original alguns atributos estavam em nominal (true or false) e decidimos converter esses mesmos atributos (win, mac, linux, is_recommended) para números inteiros únicos, ou seja, no caso de true e false, true = 1 e false=0. No caso de uma escala qualitativa do rating onde inicialmente antes da junção das tabelas tínhamos 9 classificações, [Figura 13](#), após a junção passamos apenas a ter 4, por isso o rating ficou definido de 0 a 4, [Figura 14](#).

Index	Nominal value	Absolute count	Fraction
1	Very Positive	110	0.604
2	Overwhelmingly Positive	42	0.231
3	Mostly Positive	24	0.132
4	Mixed	6	0.033
5	Mostly Negative	0	0
6	Negative	0	0
7	Overwhelmingly Negative	0	0
8	Positive	0	0
9	Very Negative	0	0

Figura 13 - Rating

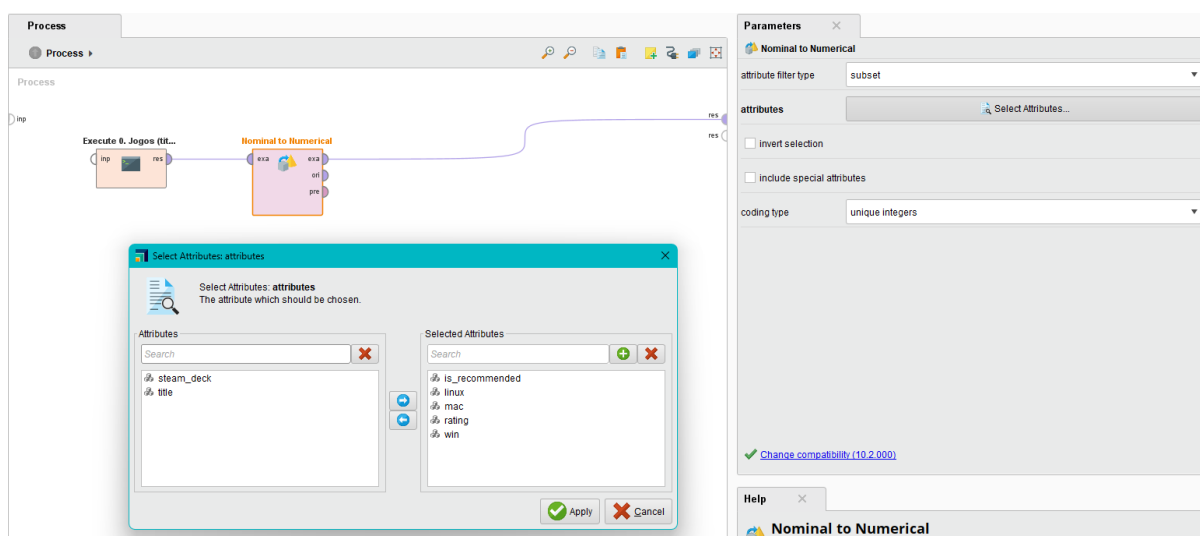


Figura 14 - Conversão de Nominal para Numérico

6 Modelação

Para realizar a análise preditiva deste conjunto de dados, alinhada com os objetivos propostos, foi essencial realizar uma seleção preliminar de modelos. Esse processo permitiu, posteriormente, a escolha do(s) melhor(es) que se adequam ao nosso conjunto de dados.

Após revermos os modelos abordados nas aulas, efetuamos uma pré-seleção e analisamos as características gerais que eram fundamentais para o nosso problema. Concentramo-nos no tipo de problema suportado pelo modelo (classificação, regressão ou ambos), na aceitação de atributos polinomiais, na capacidade de lidar com rótulos polinomiais, no desempenho preditivo e na facilidade de interpretação.

Desta forma, construímos o seguinte quadro comparativo:

Modelo	Tipo de Problema	Aceita atributos polinomiais?	Aceita Label polinomial?	Elevado desempenho preditivo?	Fácil interpretação?
Decision Tree	Ambos	Sim	Sim	Sim	Sim
Logistic Regression	Classificação	Não	Não	Não	Sim
Linear Regression	Regressão	Não	Não	Não	Sim
Neural Network	Ambos	Não	Sim	Sim	Não
Naïve Bayes	Classificação	Sim	Sim	Sim	Sim
Support Vector Machine	Classificação	Não	Sim	Sim	Não
K-NN	Ambos	Sim	Sim	Sim	Não * (não tem modelo para interpretar)

Tabela 2 - Quadro Comparativo de Modelos

Depois de construir a tabela de referência (Tabela 2), analisamos a mesma para a seleção dos melhores modelos a serem utilizados no projeto. É crucial que o algoritmo seja adequado para problemas de classificação, considerando a natureza do nosso problema. Por esse motivo, "Linear Regression" foi imediatamente excluído. Outro critério fundamental foi a capacidade de aceitar uma label polinomial, visto que nosso objetivo era prever o rating, que continha três classificações nominais diferentes. Portanto, "Logistic Regression" também foi excluído.

As técnicas escolhidas foram, portanto, Árvore de Decisão, Naïve Bayes, Support Vector Machine, Neural Network e K-NN, as quais serão detalhadas a seguir.

6.1 Árvore de Decisão

Optamos pela árvore de decisão, uma escolha fundamentada na sua facilidade de interpretação e na nossa familiaridade com o modelo. Este já tinha sido aplicado anteriormente a este conjunto de dados com o objetivo descritivo, e agora foi novamente utilizado com o intuito preditivo, passando por fases de teste e treino.

As árvores de decisão são modelos empregados em tarefas de classificação, capazes de hierarquizar todas as possíveis decisões de forma organizada. Através da sua estrutura, é possível visualizar o conjunto de ações e eventos que podem ocorrer em um determinado contexto.

O modelo de árvore de decisão apresenta uma estrutura simples e de fácil interpretação. Inicialmente, utiliza os nós raiz para representar os modelos preditivos, distribuindo-se pelos ramos que identificam os caminhos conforme as decisões e termina nas folhas, que se relacionam com as classes ou valores previstos. Na figura a seguir, apresentamos uma estrutura simplificada de uma árvore de decisão (Figura 15).

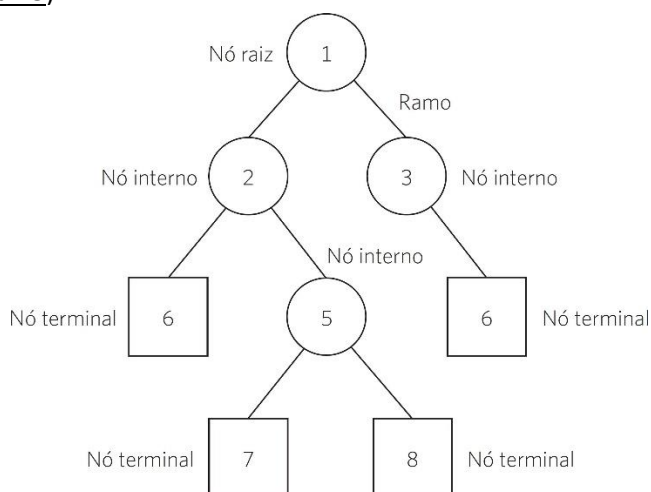


Figura 15 - Exemplo de Árvore de Decisão

Ao contrário de alguns modelos, as árvores de decisão operam tanto com dados numéricos quanto com dados categóricos, constituindo uma vantagem para qualquer tipo de estudo. Este modelo oferece uma vantagem significativa para quem o está a construir. Através de um pré-processamento automático, identifica outliers, dados ausentes e atributos irrelevantes, gerando a árvore com base nos valores processados. Apesar dessa vantagem, há um problema relacionado à definição das regras para dividir os nós, pois avalia localmente, sem ter informação suficiente, o que pode não garantir a utilização do melhor desempenho global.

Para criar uma árvore, é necessário utilizar um conjunto de dados de treino, uma vez que é um método de aprendizagem supervisionada. Na primeira fase, o conjunto de dados é dividido com base no resultado de testes aplicados a uma das variáveis presentes no conjunto de dados. O processo é repetido através de sub-partições dos dados identificados na divisão anterior, e é finalizado quando todas as ocorrências de um caminho estão descritas. Por outro lado, quando a sub-partição não resulta em previsões mais precisas, o caminho correspondente é encerrado. Esse último passo é repetido para todos os caminhos possíveis até que não haja mais variações, finalizando assim a construção da árvore. É importante destacar que, através de ferramentas de análise de dados, como o RapidMiner neste trabalho prático, é possível ajustar os parâmetros para tornar a árvore mais interpretável e estabelecer restrições aos dados desejados.

6.1.1 Aplicação do Processo

Após importar o conjunto de dados, foi essencial realizar a seleção de atributos para um subconjunto específico, incluindo "hours", "positive_ratio" e "rating". Em seguida, aplicamos a funcionalidade "Set Role" para designar a classe "Rating" como *label*. Posteriormente, incorporamos o operador "Cross Validation" para avaliar a precisão do modelo na prática. Na imagem a seguir, apresentamos a visualização do processo para este caso específico:

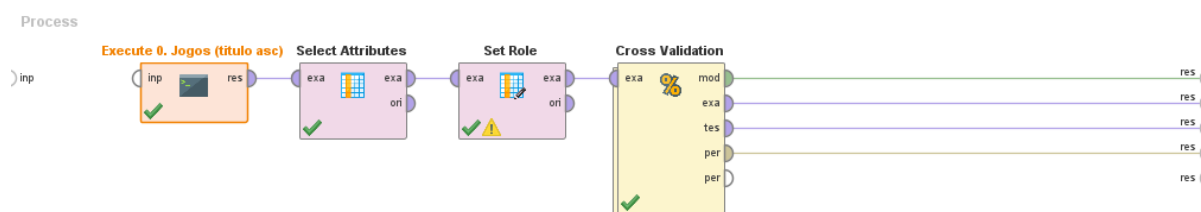


Figura 16 - Primeiro Processo da Árvore de Decisão

Incorporado no Cross Validation, encontramos a árvore de decisão e a implementação efetiva do modelo. Noutras palavras, na fase de treino, temos o operador "Decision Tree", representando o algoritmo utilizado para treinar os dados. Na etapa de teste, foram incorporados os operadores "Apply Model" e "Performance (Classification)", este último possibilitando a avaliação estatística do desempenho em tarefas de classificação.

Na imagem a seguir, é possível observar a interligação dos operadores nas etapas correspondentes de treinamento e teste:

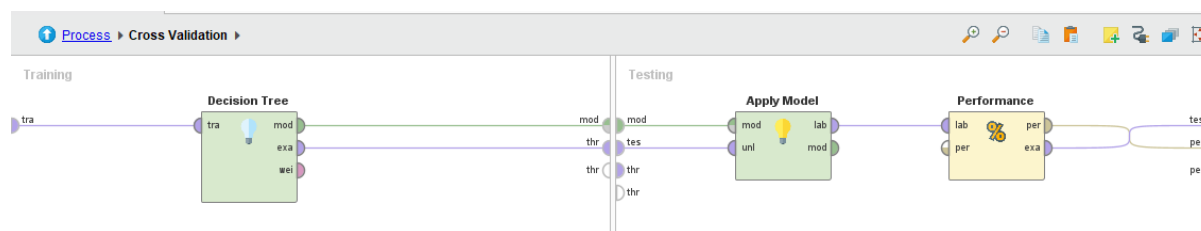


Figura 17 - Segundo Processo da Árvore de Decisão

Quanto aos operadores "Apply Model" e "Performance", não foram realizadas modificações nos parâmetros anteriormente definidos pelo RapidMiner. Apenas na árvore de decisão foram implementadas alterações.

6.1.2 Hiperparâmetros

Na definição dos hiperparâmetros da árvore de decisão, dois fatores principais foram considerados: maximizar a precisão da árvore e otimizar a sua interpretabilidade.

Nesse sentido, optou-se pelo critério "gain_ratio", que proporciona uma amplitude e uniformidade adequadas entre os valores dos atributos.

Com a escolha do critério, procedeu-se à seleção da "maximal depth". Diversos valores foram testados, uma vez que essa escolha impactava a precisão do modelo. O valor final para esse hiperparâmetro foi definido como 5, resultando num aumento na precisão de 92,31% para 99,44%.

A confiança foi ajustada para 0.25, e os demais parâmetros foram modificados de forma a possibilitar uma visualização da árvore mais simplificada e interpretável, garantindo ao mesmo tempo informações suficientes para a análise. Assim, foram atribuídos valores de 0.02 para o "minimal gain", 3 para o "minimal leaf size", 5 para o "minimal size for split", e 3 para o "Number of prepruning alternatives".

6.2 Naïve Bayes

Para ser possível alcançar uma análise preditiva mais robusta, foi decidido optar por outros modelos, de modo a realizar uma comparação e determinar qual seria o mais apropriado para a aplicação no problema preditivo em questão. Neste contexto, um dos modelos escolhidos foi o Naïve Bayes. Este algoritmo é frequentemente empregado em tarefas de classificação, tendo como objetivo principal a discriminação entre objetos distintos com base em características específicas.

O modelo Naïve Bayes é um algoritmo de classificação probabilístico que se baseia no Teorema de Bayes para realizar a categorização dos dados. Este método é especialmente útil em problemas de classificação, como classificação de documentos, sistema de recomendações, reconhecimento de padrões, entre outros.

O Naïve Bayes assume ingenuamente (daí o termo "naïve") que as características utilizadas para a classificação são independentes, o que significa que a presença ou ausência de uma característica não afeta a presença ou ausência de outras características. Apesar desta suposição simplificadora, o Naïve Bayes tem demonstrado eficácia em muitas situações práticas.

O processo de classificação do Naïve Bayes inicia-se com o treino do modelo, utilizando para isso um conjunto de dados para treino. Durante o treino, o algoritmo calcula as probabilidades de ocorrência de cada característica para cada classe. Em seguida, essas probabilidades são utilizadas para prever a classe de novos dados que ainda não foram classificados.

O Teorema de Bayes é fulcral para o funcionamento do algoritmo, permitindo a atualização das probabilidades à medida que novas evidências são apresentadas. A fórmula básica do Teorema de Bayes é:

$$P(c | x) = \frac{P(x | c)P(c)}{P(x)}$$

$P(c | X) = P(x_1 | c) \times P(x_2 | c) \times \dots \times P(x_n | c) \times P(c)$

Figura 18 - Teorema de Bayes

Depois de realizar os cálculos segundo a fórmula, o algoritmo produz resultados que representam as probabilidades de P. Esses valores permitem derivar a distribuição empírica para cada atributo preditivo por classe, de acordo com o conjunto de dados em análise.

Este modelo, de construção simples, é considerado benéfico para lidar com conjuntos de dados extensos. Além da sua simplicidade, destaca-se pela capacidade de desenvolver métodos de classificação sofisticados. Adicionalmente, é uma opção viável para realizar análises preditivas, uma vez que aceita Labels Polinomiais, tornando-o mais abrangente. Este modelo demonstra um desempenho preditivo elevado e é facilmente interpretado por quem o está a analisar.

6.2.1 Aplicação do Processo

O modelo foi construído começando por aplicar a tabela previamente trabalhada sobre os jogos, de seguida foram seleccionados os atributos pretendidos (positive_ratio, hours e rating) e posteriormente foi escolhido como o "Set Role" o atributo "Rating". Após isso aplicou-se o Cross Validation de forma a realizar uma validação cruzada no modelo, este processo pode ser visto na imagem seguinte:

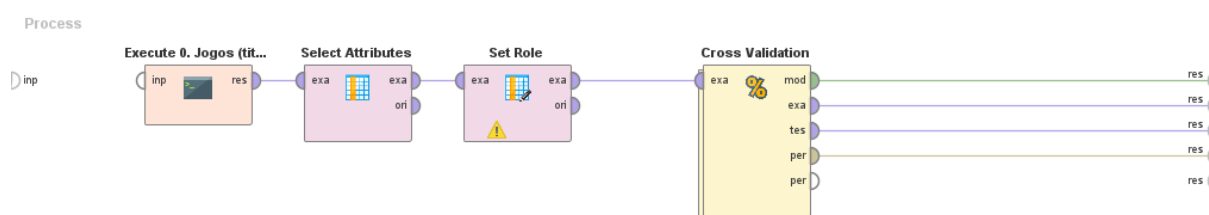


Figura 19 - Primeiro Processo Naive Bayes

Utilizou-se o Cross Validation para treinar o modelo, recorrendo ao operador "Naive Bayes", e realizou-se o teste através dos operadores "Apply Model" e "Performance (Classification)". É relevante salientar que não houve modificações nos parâmetros durante a execução destes operadores.

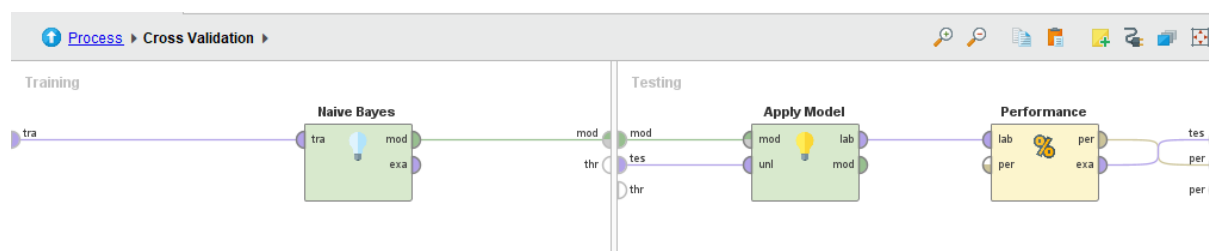


Figura 20 - Segundo Processo Naive Bayes

6.2.2 Hiperparâmetros

Neste algoritmo não existe qualquer parâmetro que possa modificar os resultados finais.

6.3 Redes Neurais Artificiais

Este modelo apresenta conjuntos de padrões representativos para cada uma das classes consideradas através do treino. Por forma a uniformizar os resultados o número de padrões deve ser o mesmo para cada classe. As RNAs¹ são sistemas distribuídos inspirados num sistema nervoso, este sistema é composto por várias unidades de processamento e possuem um grande número de conexões. O algoritmo aprende conforme os ajustes que são efetuados aos valores associados às conexões.

As unidades de processamento estão interligadas por meio de conexões. Nesse sentido, uma unidade de processamento recebe inputs de um conjunto de unidades A, realiza cálculos com base nessas entradas e envia o resultado obtido desse processo para um conjunto de unidades B. Cada conexão de entrada é associada a um peso específico.

Cada conjunto de unidades determina o seu valor de saída aplicando uma função de ativação à soma ponderada das entradas. É de realçar que na comunidade de redes neurais foram utilizadas diversas funções de ativação, variando desde funções lineares simples até funções não lineares mais complexas.

Quando é efetuado o treino dos dados, os valores dos pesos da rede são definidos por um algoritmo de aprendizagem, ou seja, os valores dos pesos são atualizados conforme a função objetivo definida. Além disso, o treino otimiza os parâmetros da função objetivo para cada unidade de processamento, de forma a diminuir o erro preditivo. Neste contexto, o erro preditivo é a diferença entre o valor de saída da unidade de processamento produzido pela função de ativação e o valor de saída desejado, o que explica o motivo de os algoritmos das redes neurais artificiais serem baseados em otimização.

Após a análise deste modelo, conseguimos perceber que pode ser aplicado tanto a problemas de classificação como de regressão. Apesar disso, tem a desvantagem de não aceitar atributos polinomiais, o que faz com que seja necessário converter dados em valores numéricos. Por outro lado, tem ainda vantagens como o facto de aceitar rótulos polinomiais, ter um grande desempenho preditivo e, para além disso, é fácil de interpretar, o que fez com que fosse considerado útil para análise preditiva.

6.3.1 Aplicação do Processo

Para este processo, foi essencial realizar a seleção de atributos para um subconjunto específico, incluindo "hours", "positive_ratio" e "rating". Em seguida, aplicamos a funcionalidade "Set Role" para designar a classe "Rating" como *label*. Posteriormente, incorporamos o operador "Cross Validation" para avaliar a precisão do modelo na prática.

¹ RNAs – Redes Neurais Artificiais

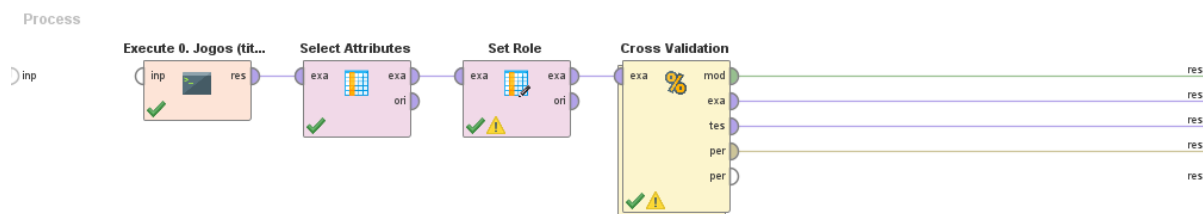


Figura 21 - Primeiro Processo Redes Neurais Artificiais

O operador "Cross Validation" possibilitou a aplicação do operador "Neural Net" para o treinamento do nosso modelo. Para testar o modelo, associamos ainda, na fase de teste, os operadores "Apply Model" e "Performance (Classification)" sem a necessidade de alteração de parâmetros.

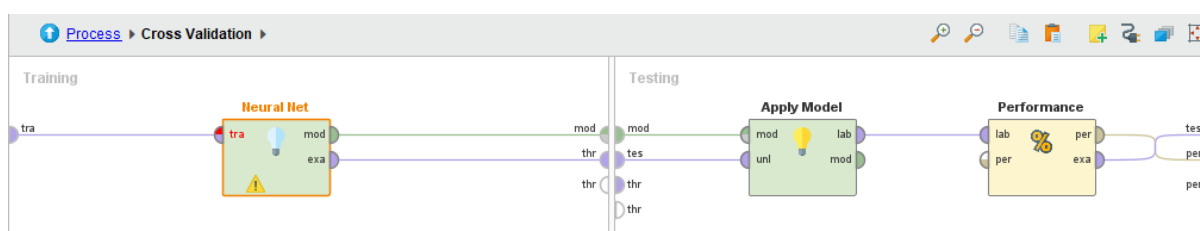


Figura 22 - Segundo Processo Redes Neurais Artificiais

6.3.2 Hiperparâmetros

Os hiperparâmetros definidos para a rede neural não foram substancialmente modificados em relação aos valores predefinidos pelo RapidMiner. As camadas ocultas permaneceram inalteradas. Quanto ao parâmetro "Training Cycles", que determina o número de ciclos de treino utilizados, foi aumentado para 400, visando alcançar uma precisão maior do modelo.

O "Learning Rate" e o "Momentum" mantiveram seus valores originais, 0.01 e 0.9, respetivamente, uma vez que qualquer alteração nesses parâmetros resultava em uma diminuição da precisão do modelo.

6.4 Support Vector Machine

Support Vector Machine (SVM) é um algoritmo de aprendizagem de máquina supervisionado, podendo ser usado para tarefas de classificação ou regressão. O principal objetivo do SVM é encontrar um hiperplano ótimo que separe os dados de diferentes classes no espaço de características. Alguns conceitos-chave associados ao SVM:

- Hiperplano: é um subespaço de dimensão. No caso de SVM bidimensional, um hiperplano seria uma linha que separa dois conjuntos de dados.
- Vetores de Suporte: São os pontos de dados que estão mais próximos do hiperplano de separação. Esses pontos são cruciais para a definição do hiperplano e, portanto, influenciam significativamente a posição e orientação do hiperplano.
- Margem: A margem é a distância entre o hiperplano e os vetores de suporte mais próximos. O objetivo do SVM é maximizar essa margem, o que ajuda a aumentar a robustez do modelo e reduzir o risco de overfitting.
- Kernel Trick: SVM pode usar o que é chamado de "truque do kernel" para mapear os dados para um espaço de características de maior dimensão, onde eles podem ser mais facilmente separados por um hiperplano. Isso é útil quando os dados não são linearmente separáveis no espaço original.
- Classificação: Para tarefas de classificação, o SVM tenta encontrar o hiperplano que separa melhor as classes. Novos dados são então classificados com base em qual lado do hiperplano eles caem.
- Regressão: O SVM também pode ser aplicado a problemas de regressão, onde o objetivo é prever um valor numérico em vez de uma classe. Nesse caso, o SVM tenta encontrar um hiperplano que melhor se ajusta aos dados.

Este modelo foi escolhido devido à sua aplicabilidade em problemas de classificação. Ao contrário do modelo de Redes Neurais Artificiais (RNA), que não aceita atributos polinomiais diretamente, o SVM permite essa flexibilidade através da conversão dos dados em valores numéricos. Além disso, o SVM demonstrou um alto desempenho preditivo com base nas pesquisas anteriores.

No entanto, é importante destacar que o SVM apresenta desafios em termos de interpretabilidade. A sua complexidade torna difícil de entender o raciocínio subjacente às decisões do modelo, tornando-o menos adequado em situações onde a interpretabilidade é crucial. No contexto deste relatório, é relevante observar que, apesar da sua precisão preditiva significativa, este modelo é identificado como o menos preciso entre os modelos analisados.

6.4.1 Aplicação do Processo

Para aplicar o algoritmo do Support Vector Machine (SVM) o primeiro passo foi selecionar os atributos, depois procedeu-se à seleção da label do Dataset sendo que essa seleção recaiu sobre o atributo "rating". Por fim inseriu-se o operador "Cross Validation".

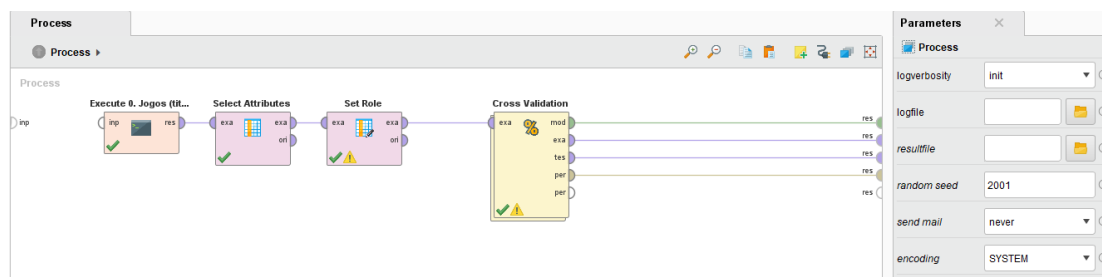


Figura 23 - Primeiro Processo SVM

Selecionado o operador SVM não foi feita alteração aos parâmetros, apesar de ter sido testada essa hipótese, era afetada uma distribuição balanceada na matriz de confusão pelo que se optou por usar os parâmetros pré-definidos pelo RapidMiner. Posteriormente acrescentou-se o “Apply Model” e a “Performance (Classification)”.

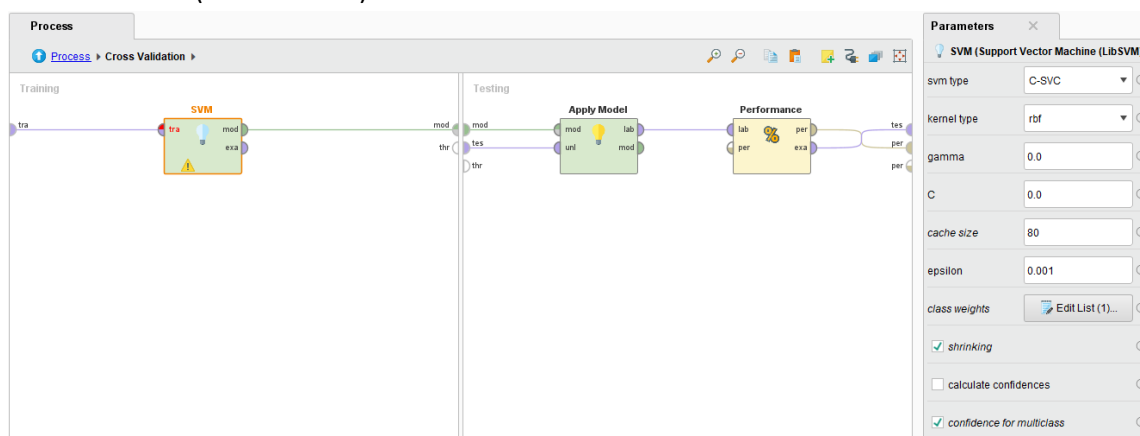


Figura 24 - Segundo Processo SVM

6.4.2 Hiperparâmetros

O SVM é influenciado por um conjunto de parâmetros que se tornam importantes de definir consoante o conjunto de dados e o objetivo da análise de dados.

Neste sentido, começou-se por definir o “SVM Type” como C-SVC uma vez que é ideal para tarefas de classificação. Posteriormente foi selecionado o “rbf” para o “Kernel Type” dado que é bom a lidar com casos em que a relação entre atributos é não linear, o que é o caso do nosso conjunto de dados. Além disso, este tipo de kernel também exige menos hiperparâmetros que podem influenciar o resultado final.

De seguida o foco da definição de hiperparâmetros voltou-se para o valor da gamma. A gamma pode ter um papel fundamental no modelo SVM uma vez que é um parâmetro que tem bastante influência na acurácia do modelo. Neste sentido foi testado mais do que um valor para perceber qual devolvia melhores resultados de “accuracy”.

O valor padrão definido pelo RapidMiner foi de 0.00 e por sua vez foi o que devolveu melhores resultados, no entanto será possível observar os diferentes resultados na tabela 3.

O parâmetro “C” define a tolerância à classificação incorreta, sendo que quanto mais elevados forem os valores de C mais suaves serão os limites. Este valor geralmente é definido em função do valor da gamma e juntos definem a acurácia de um modelo. Foi utilizado o valor padrão definido pelo RapidMiner (0.00) após várias tentativas efetuadas como podemos ver de seguida:

Gamma	C	Accuracy
0.0	0.0	74,18%
0.05	0.0	74,74%
0.1	1000	72,51%
0.2	100	69,77%
0.75	1	63,19%
1	10	63,16%
100	10	60,41%
0.1	0.1	60,41%

Tabela 3 - Hiperparâmetros SVM

6.5 K-NN

O K-NN, último modelo abordado, é um algoritmo de classificação simples baseado em aprendizagem lenta, pois não passa por uma fase explícita de treino. Em vez disso, ele memoriza os objetos do conjunto de treino, armazenando-os na memória. Quando precisa de prever a classe de um novo objeto, o modelo procura as classes dos k objetos mais semelhantes a esse novo objeto, seguindo uma abordagem de aprendizagem local.

A variável k determina quantos objetos vizinhos serão considerados. Escolher um valor adequado para k é crucial, pois um valor muito alto pode incluir vizinhos distantes que não são bons preditores, enquanto um valor muito baixo limita a consideração a objetos muito semelhantes, restringindo o modelo. É importante garantir que haja informação suficiente para uma análise robusta.

Um desafio do K-NN é que a classificação de um novo objeto pode ser lenta, devido ao cálculo da distância entre esse objeto e os demais no conjunto de treino. Estratégias como seleção de atributos ou redução do conjunto de dados podem ser empregues para mitigar esse problema.

O algoritmo utiliza a votação majoritária entre as classes dos k vizinhos mais próximos para determinar a classe do novo objeto. Ele tem apenas um hiperparâmetro, o valor de k, que precisa ser ajustado de acordo com as características do conjunto de dados. Apesar da sua simplicidade, a sua eficácia depende da escolha cuidadosa de k e do manuseamento adequado de seu desempenho computacional.

6.5.1 Aplicação do Processo

Para a aplicação desta técnica ao nosso problema preditivo foi necessário, numa primeira fase selecionar os atributos e depois definir o atributo “rating” como label e associar o operador “Cross Validation”.

Posteriormente para o treino de dados foi adicionado o operador “K-NN” com o valor de K igual a 3, e, na fase de teste, os operadores “Apply Model” e “Performance (Classification)”.

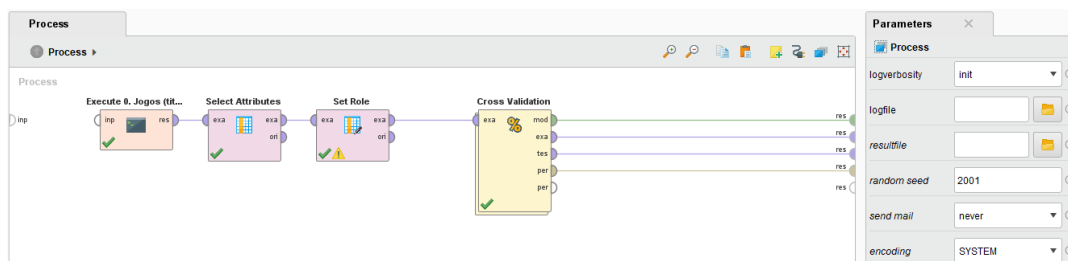


Figura 25 - Primeiro Processo K-NN

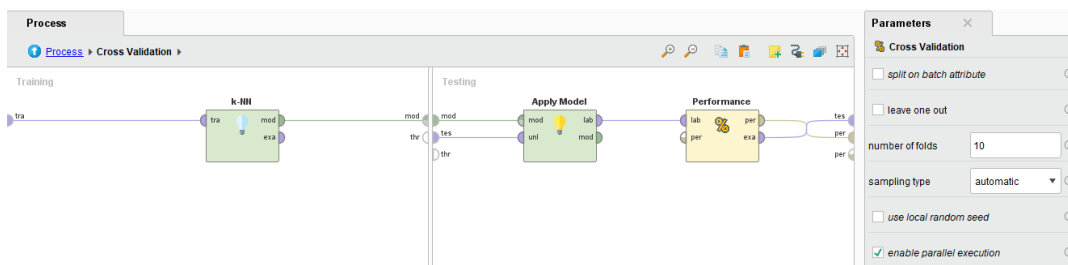


Figura 26 - Segundo Processo K-NN

6.5.2 Hiperparâmetros

No método de K-NN o único parâmetro necessário definir era o valor de K, ou seja, o número de exemplos de treino mais próximos. Para descobrir o K ideal fizemos uma tabela com base nos valores de accuracy.

k	Accuracy
2	74,71%
3	70,32%
4	73,10%
5	71,96%
6	71,96%
7	71,40%
8	71,40%
9	69,77%

Tabela 4 - Valores de K no modelo K-NN

Assim sendo, o valor de K foi definido como 3.

7 Avaliação dos Resultados

7.1 Árvore de decisão

A figura seguinte demonstra o resultado do processo de validação do modelo de árvore de decisão aplicado ao nosso conjunto de dados. O objetivo é classificar se os jogos são “Very Positive”, “Positive”, “Mixed”, “Mostly Positive”, “Overwhelmingly Positive”, “Negative”, “Mostly Negative”, “Overwhelmingly Negative” ou “Very Negative”.

Ao observar esta matriz, podemos inferir que:

- Very Positive: A árvore de decisão classificou corretamente todos os jogos nesta categoria, alcançando uma precisão de 100%. Nenhum jogo "Very Positive" foi erroneamente classificado como outra categoria.
- Positive: Todos os jogos "Positive" foram corretamente identificados, resultando em uma precisão de 100%. Nenhum falso positivo ou falso negativo foi registrado.
- Mixed: Houve 5 jogos classificados como "Mixed". Não ocorreram erros nesta categoria.
- Mostly Positive: A maioria dos jogos positivos (24) foi corretamente classificada, sem erros registrados.
- Overwhelmingly Positive: Todos os jogos com classificação "Overwhelmingly Positive" foram identificados corretamente, alcançando uma precisão de 100%.
- Negative, Mostly Negative, Overwhelmingly Negative, Very Negative: Nenhuma instância dessas categorias foi identificada. Considerando as restrições impostas para reduzir o número de linhas no trabalho, alguns dados foram perdidos, incluindo estas quatro instâncias que não foram encontradas na nossa amostra de dados. Posto isto, precisão para estas categorias é de 100%.

Relativamente ao valor de “accuracy” definido no modelo, que representa a percentagem de que, escolhida aleatoriamente uma amostra no conjunto de dados, o modelo classifique corretamente essa amostra, foi de 99,44% com um valor de 1,76 de desvio padrão.

accuracy: 99.44% +/- 1.76% (micro average: 99.45%)

	true Very Positive	true Positive	true Mixed	true Mostly Positive	true Overwhelming...	true Negative	true Mostly Negative	true Overwhelming...	true Very Negative	class precision
pred. Very Positive	110	0	0	0	0	0	0	0	0	100.00%
pred. Positive	0	0	0	0	0	0	0	0	0	0.00%
pred. Mixed	0	0	5	0	0	0	0	0	0	100.00%
pred. Mostly Positive	0	0	1	24	0	0	0	0	0	96.00%
pred. Overwhelmin...	0	0	0	0	42	0	0	0	0	100.00%
pred. Negative	0	0	0	0	0	0	0	0	0	0.00%
pred. Mostly Negat...	0	0	0	0	0	0	0	0	0	0.00%
pred. Overwhelmin...	0	0	0	0	0	0	0	0	0	0.00%
pred. Very Negative	0	0	0	0	0	0	0	0	0	0.00%
class recall	100.00%	0.00%	83.33%	100.00%	100.00%	0.00%	0.00%	0.00%	0.00%	

Figura 27 - Matriz de Confusão da Árvore de Decisão

7.2 Naïve Bayes

Ao examinar a matriz de confusão resultante da aplicação do modelo "Naïve Bayes", conclui-se que a precisão é de 98,36%, com um desvio padrão de 2,64%. Na categoria "Very positive", previu-se uma percentagem de verdade de 98,21%, enquanto a percentagem real foi de 100%, indicando uma previsão de menor ocorrência do que a observada.

Na categoria "Positive", a previsão de valor verdadeiro coincidiu com o valor real, ambos totalizando 0%. O mesmo padrão repetiu-se nas classificações "Negative", "Mostly Negative", "Overwhelmingly Negative" e "Very Negative".

Quanto à classificação "Mixed", a previsão de valor verdadeiro foi de 100%, mas a percentagem real foi de 83,33%, indicando uma expectativa de maior ocorrência do que a observada. Para a categoria "Mostly positive", previu-se uma percentagem de valor verdadeiro de 95,83%, que foi igual à percentagem real, indicando uma previsão totalmente precisa. Por fim, na categoria "Overwhelmingly Positive", esperava-se um valor verdadeiro de 100%, enquanto a percentagem real foi de 97,62%.

accuracy: 98.36% +/- 2.64% (micro average: 98.35%)

	true Very Positive	true Positive	true Mixed	true Mostly Positive	true Overwhelming...	true Negative	true Mostly Negative	true Overwhelming...	true Very Negative	class precision
pred. Very Positive	110	0	0	1	1	0	0	0	0	98.21%
pred. Positive	0	0	0	0	0	0	0	0	0	0.00%
pred. Mixed	0	0	5	0	0	0	0	0	0	100.00%
pred. Mostly Positive	0	0	1	23	0	0	0	0	0	95.83%
pred. Overwhelmin...	0	0	0	0	41	0	0	0	0	100.00%
pred. Negative	0	0	0	0	0	0	0	0	0	0.00%
pred. Mostly Negat...	0	0	0	0	0	0	0	0	0	0.00%
pred. Overwhelmin...	0	0	0	0	0	0	0	0	0	0.00%
pred. Very Negative	0	0	0	0	0	0	0	0	0	0.00%
class recall	100.00%	0.00%	83.33%	95.83%	97.62%	0.00%	0.00%	0.00%	0.00%	

Figura 28 - Matriz de Confusão Naïve Bayes

7.3 Redes Neurais Artificiais

Ao observar os valores apresentados na [Figura 29](#), proporcionadas pela Rede Neural Artificial, podemos inferir algumas considerações:

- **Very Positive:** A categoria "Very Positive" registou 107 classificações corretas, com 1 falso negativo.
- **Positive:** Não houve jogos classificados como "Positive". Todos foram classificados erroneamente, resultando em 0 verdadeiros positivos e 0 falsos positivos. Posto isto, podemos verificar que esta apresentou dificuldades, não identificando corretamente nenhum jogo nesta classificação.
- **Mixed:** Seis jogos foram incorretamente classificados como "Mixed", enquanto nenhum foi classificado corretamente.
- **Mostly Positive:** Dos jogos classificados como "Mostly Positive", 23 foram corretos, com 2 falsos positivos.
- **Overwhelmingly Positive:** Todos os 42 jogos classificados como "Overwhelmingly Positive" foram corretamente identificados.

- Negative, Mostly Negative, Overwhelmingly Negative, Very Negative: Não houve jogos identificados nestas categorias, indicando que o modelo não cometeu erros de classificação nesses casos.

Acrescenta-se ainda que a acurácia total deste modelo resultou num total de 94,53% com o desvio padrão de 5,03%.

accuracy: 94.53% +/- 5.03% (micro average: 94.51%)

	true Very Positive	true Positive	true Mixed	true Mostly Positive	true Overwhelming...	true Negative	true Mostly Negative	true Overwhelming...	true Very Negative	class precision
pred. Very Positive	107	0	0	1	0	0	0	0	0	99.07%
pred. Positive	0	0	0	0	0	0	0	0	0	0.00%
pred. Mixed	0	0	0	0	0	0	0	0	0	0.00%
pred. Mostly Positive	2	0	6	23	0	0	0	0	0	74.19%
pred. Overwhelmin...	1	0	0	0	42	0	0	0	0	97.67%
pred. Negative	0	0	0	0	0	0	0	0	0	0.00%
pred. Mostly Negat...	0	0	0	0	0	0	0	0	0	0.00%
pred. Overwhelmin...	0	0	0	0	0	0	0	0	0	0.00%
pred. Very Negative	0	0	0	0	0	0	0	0	0	0.00%
class recall	97.27%	0.00%	0.00%	95.83%	100.00%	0.00%	0.00%	0.00%	0.00%	

Figura 29 - Matriz de Confusão Redes Neurais Artificiais

7.4 Support Vector Machine

Relativamente ao tipo de risco “Very Positive”, o SVM fez uma previsão de 71,14% para a ocorrência do risco, no entanto, a verdadeira percentagem foi um valor de 95,45%.

O “Positive”, “Mixed”, “Negative”, “Mostly Negative”, “Overwhelmingly Negative”, “Very Negative” tiveram uma previsão de 0,00% tal como o valor real de 00,00%.

Para o “Mostly Positive” foi prevista uma percentagem de 100% sendo encontrados na verdade 12,50% dos valores.

Por fim, o “Overwhelmingly Positive”, o SVM previu 86,67%, sendo a sua verdadeira percentagem 61,90%.

A acurácia deste modelo apresentou um valor de 74,18% com um desvio padrão de 4,50%.

accuracy: 74.18% +/- 4.50% (micro average: 74.18%)

	true Very Positive	true Positive	true Mixed	true Mostly Positive	true Overwhelmingly ...	true Negative	true Mostly Negative	true Overwhelmingly ...	true Very Negative	class precision
pred. Very Positive	106	0	6	21	16	0	0	0	0	71.14%
pred. Positive	0	0	0	0	0	0	0	0	0	0.00%
pred. Mixed	0	0	0	0	0	0	0	0	0	0.00%
pred. Mostly Positive	0	0	0	3	0	0	0	0	0	100.00%
pred. Overwhelmi...	4	0	0	0	26	0	0	0	0	86.67%
pred. Negative	0	0	0	0	0	0	0	0	0	0.00%
pred. Mostly Negative	0	0	0	0	0	0	0	0	0	0.00%
pred. Overwhelmi...	0	0	0	0	0	0	0	0	0	0.00%
pred. Very Negative	0	0	0	0	0	0	0	0	0	0.00%
class recall	96.36%	0.00%	0.00%	12.50%	61.90%	0.00%	0.00%	0.00%	0.00%	

Figura 30 - Matriz de Confusão SVM

7.5 K-NN

Relativamente ao tipo de risco “Very Positive”, o K-NN fez uma previsão de 71,92% para a ocorrência do risco, no entanto, a verdadeira percentagem foi um valor de 95,45%.

O “Positive”, “Mixed”, “Negative”, “Mostly Negative”, “Overwhelmingly Negative”, “Very Negative” tiveram uma previsão de 0,00% tal como o valor real de 00,00%.

Para o “Mostly Positive” foi prevista uma percentagem de 25% sendo encontrados na verdade 8,33% dos valores.

Por fim, o “Overwhelmingly Positive”, o K-NN previu 85,71%, sendo a sua verdadeira percentagem 57,14%.

A precisão deste modelo é de 71,96% com desvio padrão de 4,19%.

accuracy: 71.96% +/- 4.19% (micro average: 71.98%)

	true Very Positive	true Positive	true Mixed	true Mostly Posi...	true Overwhelm...	true Negative	true Mostly Neg...	true Overwhelm...	true Very Negati...	class precision
pred. Very Posit...	105	0	1	22	18	0	0	0	0	71.92%
pred. Positive	0	0	0	0	0	0	0	0	0	0.00%
pred. Mixed	0	0	0	0	0	0	0	0	0	0.00%
pred. Mostly Po...	1	0	5	2	0	0	0	0	0	25.00%
pred. Overwhel...	4	0	0	0	24	0	0	0	0	85.71%
pred. Negative	0	0	0	0	0	0	0	0	0	0.00%
pred. Mostly Ne...	0	0	0	0	0	0	0	0	0	0.00%
pred. Overwhel...	0	0	0	0	0	0	0	0	0	0.00%
pred. Very Neg...	0	0	0	0	0	0	0	0	0	0.00%
class recall	95.45%	0.00%	0.00%	8.33%	57.14%	0.00%	0.00%	0.00%	0.00%	

Figura 31 - Matriz de Confusão K-NN

7.6 Comparação de Algoritmos

Concluída a escolha dos algoritmos, definição de hiperparâmetros e análise dos resultados dos mesmos torna-se fulcral neste ponto proceder à comparação dos diferentes resultados de forma a perceber qual(ais) se adequa(m) mais ao nosso problema e ao nosso conjunto de dados.

O desempenho dos algoritmos será descrito na seguinte tabela:

Algoritmo	Accuracy	Erro
Árvore de Decisão	99,44%	0,56%
Naïve Bayes	98,36%	1,64%
Neural Network	94,53%	5,47%
SVM	74,18%	25,82%
K-NN	71,96%	28,04%

Tabela 5 - Comparação de Algoritmos

Analisando a tabela, constatamos que a Árvore de Decisão obteve a maior taxa de precisão, em contraste com o K-NN, que, por sua vez, revelou os resultados menos favoráveis em termos de acurácia.

Diante disso, optamos por escolher a Árvore de Decisão como o modelo para a análise preditiva do nosso conjunto de dados, visando abordar a problemática de previsão de classificação de jogos com base nas avaliações dos utilizadores.

A estrutura da árvore gerada pelo RapidMiner é a seguinte:

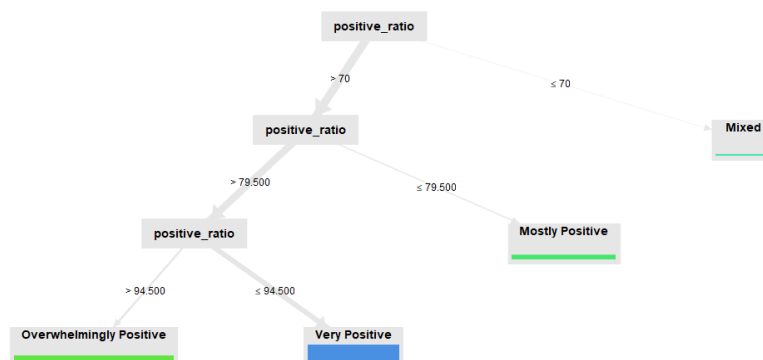


Figura 32 - Árvore de Decisão

Esta árvore de decisão fornece observações significativas sobre os atributos que influenciam a classificação de jogos pelos utilizadores, com foco na variável "positive_ratio". Ao analisar a estrutura da árvore e interpretar o seu significado podemos compreender o seguinte:

- Nó Raiz (positive_ratio > 70):
 - Este nó divide os jogos com base na proporção de avaliações positivas ("positive_ratio"). Se essa proporção for superior a 70%, a árvore progride para o próximo nível.
- Primeiro Nível (positive_ratio > 79.500 e positive_ratio ≤ 79.500):
 - No caso de "positive_ratio" superior a 79.500, a árvore verifica se é superior a 94.500. Se sim, o jogo é classificado como "Overwhelmingly Positive", indicando uma avaliação extremamente positiva. Caso contrário, o jogo é classificado como "Very Positive", sugerindo uma avaliação muito positiva, mas não necessariamente no nível extremo.
 - Se "positive_ratio" for inferior a 79.500, mas ainda superior a 70, o jogo é classificado como "Mostly Positive", indicando uma maioria de avaliações positivas.
- Segundo Nível (positive_ratio ≤ 94.500):
 - Se "positive_ratio" for inferior a 94.500 (caso do "Very Positive"), o jogo recebe a classificação "Very Positive". Caso contrário, se for superior, o jogo é categorizado como "Overwhelmingly Positive", destacando uma grande maioria de avaliações extremamente positivas.
- Terceiro Nível (positive_ratio ≤ 70):
 - Se a proporção de avaliações positivas for igual ou inferior a 70%, o jogo é classificado como "Mixed", sugerindo uma mistura de avaliações positivas e negativas.

Posto isto, é possível compreender o seguinte:

- A variável "positive_ratio" é claramente crucial na decisão do rating. Quanto maior essa proporção, maior a probabilidade de o jogo receber uma classificação mais positiva.
- A árvore enfatiza a distinção entre avaliações "Very Positive" e "Overwhelmingly Positive" com base em limites específicos de "positive_ratio".
- A categoria "Mixed" é atribuída quando a proporção de avaliações positivas é relativamente baixa, indicando uma diversidade de opiniões.

Esta árvore de decisão oferece uma interpretação clara e intuitiva de como a variável "positive_ratio" influencia a classificação dos jogos pelos utilizadores, fornecendo uma base sólida para tomada de decisões na categorização de jogos.

8 Conclusão

Em suma, a execução deste projeto enfatiza a aplicabilidade prática de diversos modelos de classificação na análise preditiva do dataset "Game Recommendations on Steam" utilizando a metodologia CRISP-DM. O principal desafio foi a seleção do modelo mais adequado para prever os ratings de diferentes tipos de jogos com base nas suas características, e onde foram escolhidos cinco modelos distintos: Árvore de Decisão, Naïve Bayes, Redes Neurais Artificiais, Support Vector Machine e K-NN.

Durante a fase de modelagem, foram realizados ajustes de hiperparâmetros e testes para aprimorar a precisão dos modelos, tendo como objetivo encontrar qual deles se melhor adequa ao nosso problema. Em última análise, este trabalho prático proporcionou não só insights valiosos para aplicação em cenários reais, mas também ressaltou a importância da integração entre a teoria discutida na análise e visualização de dados, conforme abordado na unidade curricular. A conjugação do conhecimento teórico com a experiência prática contribuiu significativamente para uma compreensão mais aprofundada e efetiva no campo da análise e previsão de dados específicos relacionados aos ratings de jogos.

Referências

- Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2020). *CRISP-DM 1.0. Step-by-step data mining guide*: https://s2.smu.edu/tfomby/eco5385_eco6380/data/SPSS/CRISPWP-0800%20Data%20Mining%20Standards.pdf
- Decision Tree Algorithm Examples In Data Mining*. (27 de 06 de 2023). Obtido de Software Testing Help: <https://www.softwaretestinghelp.com/decision-tree-algorithm-examples-data-mining/>
- How to Join Your Data [Tips + Tricks]*. (08 de 09 de 2016). Obtido de RapidMiner: <https://rapidminer.com/blog/tips-tricks-different-ways-to-join-data/>
- Modelo de suporte à decisão aplicado à identificação de indivíduos não aderentes ao tratamento anti-hipertensivo*. (2014). Obtido de Scielo: <https://www.scielo.br/j/sdeb/a/rPQMWDgncmxjZ9spRygfPMj/>
- Moreira, J. M. (2023). *Data Mining*. Obtido de Moodle: https://moodle2324.up.pt/pluginfile.php/105276/mod_resource/content/2/1%20-%20CRISP%20DM.pptx.pdf
- Naive Bayesian*. (2024). Obtido de An Introduction to Data Science: https://www.saedsayad.com/naive_bayesian.htm
- Replace Missing Values*. (2023). Obtido de RapidMiner: https://docs.rapidminer.com/latest/studio/operators/cleansing/missing/replace_missing_values.html
- Silva, A. F. (28 de 02 de 2016). *Using Data Mining to Predict Students' Academic Success*, p. 90. Obtido de <https://repositorio-aberto.up.pt/bitstream/10216/110241/2/117807.pdf>

Anexos

Anexo 1








Name	Type	Missing	Statistics	Filter (22 / 22 attributes)
app_id	Integer	0	 Min: 570, Max: 1967080, Average: 559978.681, Deviation: 464045.050	
user_id	Integer	0	 Min: 27862, Max: 13514834, Average: 7620401.330, Deviation: 3477239.547	
win	Integer	0	 Min: 1, Max: 1, Average: 1, Deviation: 0	
mac	Real	0	 Min: 0, Max: 1, Average: 0.363, Deviation: 0.482	
linux	Real	0	 Min: 0, Max: 1, Average: 0.236, Deviation: 0.426	
rating	Real	0	 Min: 0, Max: 4, Average: 1.385, Deviation: 1.764	
is_recommended	Real	0	 Min: 0, Max: 1, Average: 0.143, Deviation: 0.356	
title	Nominal	0	 Least: 8Y2RabbitWY2 (0), Most: Grand Theft Auto V (10), Grand Theft Auto V (10), Conan Exiles (5), Counter-Strike: Global Offensive (5), ... [48178 more] Details...	
date_release	Date-time	0	 Earliest date: Oct 19, 2010 12:00 AM, Latest date: Mar 13, 2023 12:00 AM, Duration: 4528d 1h 0m 0s	
positive_ratio	Integer	0	 Min: 55, Max: 98, Average: 87.121, Deviation: 8.487	
user_reviews	Integer	0	 Min: 1838, Max: 7494460, Average: 533510.786, Deviation: 1258563.581	
price_final	Real	0	 Min: 0, Max: 70, Average: 23.151, Deviation: 18.216	
price_original	Real	0	 Min: 0, Max: 0, Average: 0, Deviation: 0	
discount	Real	0	 Min: 0, Max: 0, Average: 0, Deviation: 0	
steam_deck	Nominal	0	 Least: false (0), Most: true (152), Values: true (152), false (0) Details...	
helpful	Integer	0	 Min: 0, Max: 50, Average: 1.648, Deviation: 5.503	
funny	Integer	0	 Min: 0, Max: 55, Average: 0.643, Deviation: 4.373	
date	Date-time	0	 Earliest date: Sep 8, 2012 12:00 AM, Latest date: Dec 30, 2022 12:00 AM, Duration: 3765d 1h 0m 0s	
hours	Real	0	 Min: 0.400, Max: 968.700, Average: 227.923, Deviation: 251.597	
review_id	Integer	0	 Min: 971, Max: 50714, Average: 25532.802, Deviation: 14875.270	
products	Integer	0	 Min: 0, Max: 5166, Average: 246.418, Deviation: 471.725	
reviews	Integer	0	 Min: 1, Max: 82, Average: 8.462, Deviation: 13.655	

Figura 33 - Estatísticas relativas aos Atributos