



Instituto Politécnico de Gestão e Tecnologia

Curso PG em Analytics e Data Science Empresarial

ANÁLISE DO CATÁLOGO DE AMAZON KINDLE

Joana Araújo, A22507959

Pedro Correia, A22502161

Docente:

Jorge Duque

Unidade Curricular: Fundamentos de Business Intelligence e Análise de Dados

Vila Nova de Gaia

Ano académico 2025-2026

[página de verso, em branco]

Resumo

Este projeto apresenta o desenvolvimento de uma solução integrada de Business Intelligence (BI) e Data Mining aplicada ao catálogo de e-books da Amazon Kindle. O objetivo principal foi transformar dados brutos em conhecimento estratégico, seguindo a hierarquia DIKW (Dados, Informação, Conhecimento, Sabedoria). A metodologia adotada seguiu as etapas do processo KDD (Knowledge Discovery in Databases), iniciando-se com um processo de ETL em Python e Power Query para limpeza, normalização e binning de preços. A arquitetura de dados foi consolidada num modelo dimensional em estrela (Star Schema) no Power BI, integrando uma tabela de factos central com diversas dimensões. A componente analítica incluiu a criação de um dashboard para monitorização de KPIs, análise de elasticidade de preço e impacto da subscrição Kindle Unlimited. Complementarmente, explorou-se a modelagem preditiva para estimar a popularidade dos livros. Após uma tentativa inicial de classificação multiclasse com desempenho insuficiente, optou-se por um modelo de regressão (Random Forest), alcançando um R^2 de 0.8595. Os resultados demonstram que o sucesso na plataforma depende de um equilíbrio entre visibilidade (subscrições) e reputação das obras, e não apenas do preço baixo.

Palavras chave: Business Intelligence; Data Mining; Power BI; Machine Learning; Análise de Dados; Amazon Kindle.

ÍNDICE

ÍNDICE	4
ÍNDICE DE FIGURAS.....	5
ÍNDICE DE TABELAS	6
INTRODUÇÃO.....	7
DESENVOLVIMENTO DO TEMA	8
1. FUNDAMENTO TEÓRICO	8
1.1 DADOS, CONHECIMENTO E BUSINESS INTELLIGENCE	8
1.2 PROCESSO ETL	8
1.3 MACHINE LEARNING.....	9
2. CARACTERIZAÇÃO DO DATASET	12
3. PROCESSO ETL E MODELO DE DADOS	15
3.1 MEDIDAS BASE	18
3.2 MEDIDAS DE NEGÓCIO.....	19
3.3 MEDIDAS DE PREÇO E VALOR	20
3.4 MEDIDAS DE TIME INTELLIGENCE E CATEGORIZAÇÃO	21
4. VISUALIZAÇÃO DOS DADOS – DESENVOLVIMENTO DA SOLUÇÃO EM BI	24
4.1. DIAGRAMA DO LAYOUT E PRINCÍPIOS DE DESIGN.....	24
4.2. DESCRIÇÃO DAS PÁGINAS E VISUALIZAÇÕES.....	24
5. DESENVOLVIMENTO DO ALGORITMO DE DATA MINING – MODELAGEM PREDITIVA	27
5.1. ABORDAGEM METODOLÓGICA EXPLORADA: CLASSIFICAÇÃO	27
5.2. ABORDAGEM FINAL: REGRESSÃO	28
6. RESULTADOS, ANÁLISE E DISCUSSÃO	31
6.1 BUSINESS INTELLIGENCE.....	31
6.2 MODELAGEM PREDITIVA E DATA MINING	32
6.3 ANÁLISE DA IMPORTÂNCIA DAS FEATURES E O DILEMA DA SIMPLIFICAÇÃO.....	34
6.4. INTERPRETAÇÃO DO GRÁFICO LOG-LOG	34
LIMITAÇÕES E TRABALHO FUTURO	36
A. PROPOSTAS PARA TRABALHO FUTURO	38
CONCLUSÕES	40
BIBLIOGRAFIA.....	41
ANEXOS	42

ÍNDICE DE FIGURAS

Figura 1 Visualização do data model final	17
Figura 2 Gráfico de dispersão entre as Reviews Reais e Previstas, transformadas numa escala logarítmica	34

ÍNDICE DE TABELAS

Tabela 1 Dicionário de dados do dataset	12
Tabela 2 Transformações efetuadas ao DataFrame	16
Tabela 3 Resultados do test set	33

INTRODUÇÃO

No atual cenário da economia digital, a capacidade de extrair valor de grandes volumes de dados é um diferencial competitivo crítico. O mercado editorial, especificamente o segmento de e-books, gera terabytes de dados sobre preferências de leitura, estratégias de precificação e dinâmicas de publicação que, se devidamente analisados, podem orientar decisões estratégicas vitais para autores, editoras e plataformas.

Este trabalho surge no âmbito da Pós-Graduação em Analytics e Data Science Empresarial, com o objetivo de aplicar os conceitos fundamentais de Business Intelligence e Análise de Dados a um cenário real. O projeto foca-se na análise do dataset "Amazon Kindle Books Dataset 2023", obtido via Kaggle, que contém informações sobre mais de 100.000 títulos.

O problema central abordado é a identificação dos fatores que potenciam a popularidade e o sucesso comercial de uma obra na plataforma Kindle. Para tal, o trabalho estrutura-se em torno da construção de um sistema de apoio à decisão que permita responder a questões como: "Qual a relação entre o preço e a avaliação dos leitores?", "O programa Kindle Unlimited garante maior visibilidade?" e "É possível prever o número de reviews com base nas características do livro?".

A metodologia adotada percorre as fases clássicas de um projeto de dados: desde a ingestão e tratamento (ETL), passando pela modelagem dimensional (esquema em estrela) e visualização em Power BI, até à aplicação de algoritmos de Machine Learning (Python) para extração de conhecimento preditivo. O relatório encontra-se organizado de forma a espelhar este fluxo, detalhando as opções técnicas tomadas, desde o pré-processamento até à discussão dos resultados analíticos e preditivos.

DESENVOLVIMENTO DO TEMA

1. FUNDAMENTO TEÓRICO

1.1 Dados, Conhecimento e Business Intelligence

No contexto de data science e business intelligence, é fundamental compreender a hierarquia de valor dos dados, aqui representada pela pirâmide DIKW (Data-Information-Knowledge-Wisdom) (Duque, 2025). Na base desta pirâmide situam-se os dados, como matéria-prima de data science, composta por observações singulares de factos sem contexto direto. No âmbito deste projeto, um dado seria o valor "9.99" ou a string do atributo género "Romance", isolados na nossa base de dados. Um degrau acima surge a informação, quando os dados são organizados e contextualizados, transmitindo uma mensagem sobre um evento ou caracterizando algo. Por exemplo, ao processarmos os dados, obtemos a informação de que "O livro X, da categoria Romance, custa \$9.99". A seguir, conhecimento, é constituído pela informação estruturada e internalizada, permitindo a compreensão de padrões. No nosso projeto, o conhecimento emerge quando identificamos que "Livros da editora Y, com preço inferior a \$5, tendem a ter avaliações superiores a 4 estrelas". Por último, inteligência, sita no topo da pirâmide, refere-se ao conhecimento sintetizado e aplicado para ganhar profundidade de consciência e apoiar a decisão, neste projeto revestindo a forma de um algoritmo.

É com base nesta inteligência, aplicada ao contexto organizacional, que surge o Business Intelligence (BI), definido como o conjunto de processos, tecnologias e ferramentas que transformam dados brutos em insights para a tomada de decisão. O BI permite às organizações monitorizar o desempenho e identificar tendências de mercado, o que aplicaremos ao analisar o mercado de e-books.

Paralelamente, o conceito de analytics foca-se na interpretação dos dados para descobrir padrões. O nosso trabalho foca-se maioritariamente em duas tipologias: a análise descritiva, para responder a "Porque aconteceu?", visualizando a distribuição atual de preços e reviews dos livros por exemplo, e análise prescritiva, para explorar "O que vai acontecer?", para determinar a probabilidade de um livro ter um desempenho acima da média.

1.2 Processo ETL

Para possibilitar esta análise, recorreremos ao processo ETL (Extract, Transform, Load), para a ingestão e preparação dos dados, através dos 3 passos: extract (extração): obtenção dos dados

brutos a partir do repositório Kaggle (ficheiro CSV); transform (transformação): aplicação de regras de limpeza, normalização e criação de colunas derivadas. No nosso projeto, utilizamos Python e Power Query como área de staging para tratar valores nulos, binning, entre outras técnicas. É aqui que ocorre a “limpeza” dos dados: *“Data cleaning routines attempt to fill in missing values, smooth out noise while identifying outliers, and correct inconsistencies in the data.”* (Han, Kamber, & Pei, 2011, p. 88) com o intuito de obter uma análise congruente e sem enviesamento, assim como proporcionar dados fidedignos para algoritmos preditivos; por último, load (carregamento): inserção dos dados transformados no modelo de dados final (Power BI) para análise.

A arquitetura de BI adotada segue uma estrutura simples composta por fontes de dados (CSV), uma área de staging (Pandas e Power Query) e uma camada semântica onde implementámos um modelo dimensional, especificamente um Esquema em Estrela (Star Schema). Este esquema integra uma tabela de factos central (facts) ligada a várias tabelas de dimensões (dim_autor, dim_livro, etc.), otimizando o desempenho das consultas analíticas, e inclui medidas DAX adaptadas à lógica do negócio.

O projeto toca em conceitos de Data Mining, uma prática que visa descobrir padrões novos, úteis e válidos em grandes volumes de dados. Seguimos a lógica do processo KDD (Knowledge Discovery in Databases), percorrendo as suas etapas desde a seleção dos dados até à interpretação dos padrões. *“[KDD is] the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data.”* (Fayyad et al., 1996, p. 40) O output do processo KDD são modelos e insights que apoiam a predição e o *decision-making*.

1.3 Machine Learning

O Machine Learning é um subcampo da inteligência artificial que permite aos sistemas aprender a partir de dados, identificar padrões e tomar decisões com intervenção humana mínima. No contexto do processo de extração de conhecimento em dados, o machine learning é a ferramenta essencial para transformar a informação em conhecimento, através da criação de modelos preditivos. A modelagem preditiva é o processo de utilizar dados históricos (*features*) para construir um modelo matemático capaz de prever um valor ou resultado futuro (*target*).

Existem dois tipos de tarefas preditivas: supervisionada e não supervisionada (Müller & Guido, 2017). No contexto deste trabalho, iremos focar apenas na supervisionada. Neste tipo

de tarefa, o modelo é treinado com pares de input e output conhecidos, e destacam-se duas tarefas principais, ambas exploradas neste projeto:

- Classificação: é utilizada quando o *target* é categórico ou discreto. O modelo aprende a mapear os dados de entrada para rótulos de classe pré-definidos.
- Regressão: é utilizada quando o *target* é contínuo ou numérico. O objetivo é prever um valor real.
- Para garantir que o modelo não memorize os dados de treino (*data leakage*, que pode levar ao *overfitting*) e que tenha uma capacidade de generalização robusta para novos dados, são utilizadas estratégias de validação:
- Divisão em treino e teste (*train_test_split()*): O dataset é dividido em dois subconjuntos: o dataset de treino, que é usado para alimentar o algoritmo e onde o modelo aprende a relação entre as *features* (variáveis independentes) e o *target*; e o dataset de teste, que é usado para a avaliação final do modelo, simulando dados nunca vistos (Joseph & Vakayil (2022)).
- *Cross-Validation* e Otimização: a validação cruzada é uma técnica que divide o conjunto de treino em K partes, treinando o modelo K vezes, sendo uma técnica de otimização que automatiza a pesquisa da melhor combinação de hiperparâmetros. Esta técnica garante que o modelo final atinja a sua capacidade preditiva máxima de forma sistemática.

No final, é fundamental existirem métricas de avaliação para medirem o sucesso do modelo face ao objetivo de negócio. Para a classificação, as principais métricas são:

- Acurácia: mede a proporção das previsões corretas
- Recall: mede a proporção dos valores positivos corretamente identificados
- Matrix de confusão: é a tabela de sumariza os resultados de classificação (Verdadeiros Positivos, Falsos Negativos, Falsos Positivos, Verdadeiros Negativos)

Para a regressão, são utilizadas diferentes métricas para avaliar a performance do modelo:

- R^2 (Coeficiente de Determinação): métrica que varia entre 0 e 1, e indica a proporção da variância no *target* que é explicada pelas variáveis do modelo.
- MSE (Mean Squared Error): erro quadrático médio, que penaliza desvios maiores.

- RMSE (Root Mean Squared Error): indica a raiz quadrada do MSE, que devolve o erro para a unidade de medida do target. É a métrica mais interpretável do erro.

2. CARACTERIZAÇÃO DO DATASET

O conjunto de dados selecionado para este projeto intitula-se "Amazon Kindle Books Dataset 2023" e foi obtido através da plataforma Kaggle, uma plataforma de competição em data science detida pela Google onde é possível encontrar datasets livres e gratuitos. O autor do dataset, identificado como Asaniczka, compilou os dados através de técnicas de *web scraping* diretamente da livraria online Kindle Books detida pela Amazon. O autor não especificou os detalhes da captura de dados e não é possível garantir a representatividade total do catálogo Amazon Kindle Books. O DOI deste dataset encontra-se referenciado nas referências bibliográficas. A cobertura dos dados estende-se ao mercado norte-americano, pelo que o locale deve ser considerado o en-US.

Este dataset tem como objetivo fornecer uma visão abrangente do catálogo de e-books disponíveis, estático no tempo (sem dados transacionais), capturando métricas de desempenho (avaliações), metadados bibliográficos e indicadores de vendas (como etiquetas de best-seller). O ficheiro de origem é um flat file em formato csv (*kindle_data-v2.csv*), contendo aproximadamente 130.000 registos referentes a obras publicadas ou disponíveis até ao final de 2023, com as primeiras obras datadas do séc. XIX. Sendo um flat file, é composto por uma tabela única que agrega todas as dimensões do livro numa estrutura desnormalizada. Abaixo, apresenta-se o dicionário de dados das colunas presentes no ficheiro original:

Tabela 1 Dicionário de dados do dataset

<i>Atributo</i>	<i>Tipo de Dado (Original)</i>	<i>Descrição</i>
<i>asin</i>	String (Alfanumérico)	<i>Amazon Standard Identification Number.</i> Identificador único e chave primária natural de cada livro.
<i>title</i>	String	O título completo da obra.
<i>author</i>	String	Nome do autor da obra.
<i>soldBy</i>	String	Entidade responsável pela venda (geralmente a editora ou o autor em <i>self-publishing</i>).
<i>imgUrl</i>	String (URL)	Hiperligação para a imagem da capa do livro.
<i>productURL</i>	String (URL)	Hiperligação direta para a página do produto na Amazon.
<i>stars</i>	Decimal	Classificação média atribuída pelos utilizadores (escala de 0 a 5).

<i>reviews</i>	Inteiro	Número total de avaliações escritas recebidas pelo livro.
<i>price</i>	Decimal	Preço de venda do e-book (na moeda local capturada, USD).
<i>isKindleUnlimited</i>	Booleano	Indica se o livro está incluído no programa de subscrição <i>Kindle Unlimited</i> .
<i>category_id</i>	Inteiro	Identificador numérico da categoria do livro na taxonomia da Amazon.
<i>isBestSeller</i>	Booleano	Indicador se o livro possui a etiqueta de "Best Seller" na sua categoria.
<i>isEditorsPick</i>	Booleano	Indicador se o livro foi selecionado como uma "Escolha do Editor".
<i>isGoodReadsChoice</i>	Booleano	Indicador se o livro foi nomeado ou venceu prémios no Goodreads.
<i>publishedDate</i>	String / Short date	Data de publicação da obra.
<i>category_name</i>	String	Designação descritiva da categoria principal do livro (ex: "Mystery", "Romance").

Para contextualizar, o programa “Kindle Unlimited” (ao qual se refere o atributo bool “isKindleUnlimited”) é uma subscrição mensal que fornece acesso ilimitado a um subset do catálogo. Está disponível apenas em algumas regiões geográficas.

Sendo um dataset resultante de *web scraping*, o ficheiro original apresenta desafios típicos de dados não estruturados ou semiestruturados que necessitam de tratamento na fase de ETL antes da modelação, a saber:

- Valores em falta (Nulls e Zeros): existem registos onde o campo stars ou reviews é igual a 0. Isto pode significar tanto a ausência real de avaliações (em livros novos) como uma falha na captura do dado. O campo price também pode apresentar inconsistências, com valores a 0 que podem representar livros gratuitos ou promoções temporárias que distorcem a análise de receita. De acordo com o autor, valores 0 devem ser interpretados como nulos.
- Formatação de datas: a coluna publishedDate no ficheiro original pode ser interpretada como texto, exigindo conversão para um formato Short Date válido para permitir a criação da dimensão de calendário.
- Consistência de nomes próprios e cardinalidade: os campos author e soldBy podem conter duplicados semânticos tal como a mesma entidade com o nome escrito em uppercase ou title case. Isto gera uma cardinalidade elevada para estas dimensões. Existe ainda a dificuldade de encontrar equivalentes semânticos, por exemplo, a editora Penguin Random House, considerada a uma das "Big Five" editoras da língua inglesa,

resultou de uma fusão entre as editoras Penguin Books e Random House em 2013. Neste caso preservámos estas editoras para manter factualidade histórica.

- Caracteres especiais: sendo dados globais, os campos de texto (title, author, soldBy) contêm frequentemente caracteres especiais ou problemas de encoding que necessitam de normalização.

Contudo, o maior desafio advém da limitação deste dataset, que não contém dados de vendas e é um *snapshot* estático do catálogo no tempo. O desafio será converter este catálogo estático numa fonte de *insights* para a gestão.

3. PROCESSO ETL E MODELO DE DADOS

O dataset descarregado do Kaggle, conforme exposto anteriormente, está num formato flat file em csv. Dado que o objetivo deste trabalho é realizar uma análise de dados com base no dataset, uma das primeiras prioridades foi separar o dataset entre factos e dimensões, e estabelecer relações entre estas. Com este tipo de modelagem, visamos criar um modelo robusto para analytics, pois: “*Dimensional modeling [...] addresses two simultaneous requirements: [...] Deliver data that’s understandable to the business users. Deliver fast query performance*” (Kimball & Ross, 2013, p. 7). Com isto, temos um modelo de dados escalável com volume de transações e com necessidades de acrescentar novas dimensões, ao mesmo tempo minimizando o número de joins necessárias e melhorando a query performance.

Além disso, sendo os dados de preço contínuos, consideramos indispensável categorizá-los para permitir “*slicing and dicing*” na análise e desenvolver modelos de ML. A nossa primeira ordem de trabalhos foi então: criar uma chave "surrogate" entre autores e editoras (inexistentes no dataset de origem, que apenas usa o nome do autor e da editora, o que não é fiável devido a formatações diferentes, sendo que para o efeito usamos os métodos string em Python: strip() e title(), para padronizar os nomes antes de encontrar nomes únicos). De seguida, fizemos binning ao preço, categorizando-o por intervalos de preço de 5\$ em 5\$ até aos 40\$, dado serem os preços mais comuns, criando bins de 20\$ em 20\$ até aos 100\$ para os restantes. Esta categorização baseia-se no domain knowledge e visa facilitar a interpretação para o utilizador final, neste caso um gestor de e-commerce. O binning seguiu a psicologia de preços típica do mercado de e-books (preços psicológicos: 0.99, 2.99, 9.99), criando faixas que refletem a decisão de compra do consumidor (ex: compra por impulso < 5\$ vs. compra ponderada > 10\$). Na discussão de resultados iremos abordar um método alternativo que seria dividir em decís com frequências iguais, principalmente se o nosso foco fosse machine learning, para evitar enviesamento.

Em prol da eficiência, e discernir qual a melhor ferramenta para o trabalho em mãos, realizámos estes primeiros passos em Python com as libraries Pandas e NumPy. No fim, exportámos um ficheiro flat csv com todas as colunas, e procedemos à sua repartição em dimensões, renomeação e substituição de valores no Power Query, dado ser uma ferramenta eficiente e intuitiva para este tipo de transformação rápida e “ligeira”.

Tabela 2 Transformações efetuadas ao DataFrame

<i>Etapa</i>	<i>Operação Python</i>	<i>Finalidade</i>
1. <i>Extração</i>	pd.read_csv	Extração de flat para um DataFrame
2. <i>Limpeza</i>	.strip().title()	Padronização e correção de inconsistências
3. <i>Transformação</i>	pd.cut (Binning Preço)	Feature engineering (Discretização preço)
4. <i>Modelação</i>	pd.index (Criar IDs)	Criação de PKs para Star Schema
5. <i>Staging</i>	pd.to_csv	Preparação para carga no Power Query

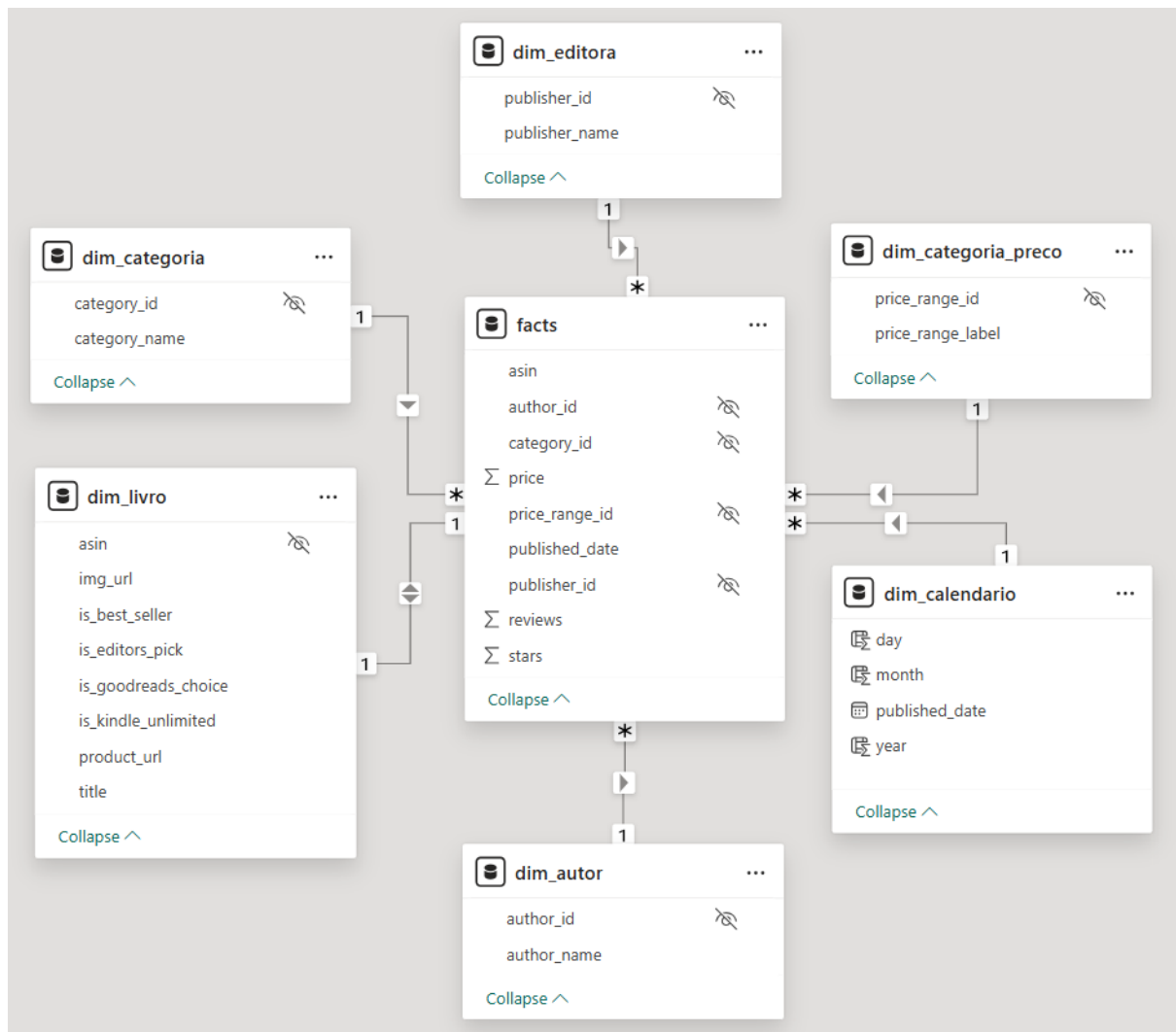
Já com este flat file intermédio em Power Query, procedemos à seguinte ordem de trabalhos:

- Separação das seguintes tabelas dimensões: autor, categoria de livro, categoria de preço, editora.
- Converter strings “Unknown” em nulls (entire cell content).
- Nas tabelas de dimensão supracitadas, procedemos à deduplicação de nomes e da PK de forma ser possível estabelecer uma relação 1:N com a tabela de facts. Ordenámos também os nomes por ordem ascendente e reordenámos as colunas colocando a PK em primeiro por uma questão de padronização;
- Criação da tabela de dimensão: calendário, por via das funções DAX calendarauto(), year(), month(), day() para permitir slicing temporal;
- Aproveitámos a tabela original que importamos para o Power Query para a transformar na tabela de facts, removendo as colunas que foram copiadas nas tabelas de dimensões, restando apenas as colunas: asin, author_id, category_id, price, price_range_id, published_date, publisher_id, reviews, stars;
- De acordo com o autor do dataset no Kaggle, nas colunas stars, reviews e price, o número 0 representa valores null. Para não afetar medidas DAX e análise de dados, substituímos o 0 por null nestas colunas. Efetivamente, estes livros serão ignorados na análise de preço, tal como no cálculo DAX de médias. Optámos também por tratar 0 nas reviews como null para evitar distorcer as médias com a grande quantidade de livros sem interação, focando a análise nos livros que já têm tração de mercado.

Por último, estabelecemos as relações e cardinalidade entre as tabelas de dimensões e a tabela facts. Estas relações foram de 1:N exceto com a tabela de dimensões dos livros, que caracteriza cada livro do catálogo, onde a cardinalidade é 1:1. Isto acontece pois estamos a analisar uma “snapshot” do catálogo estática no tempo. A tabela facts contém dados numéricos, mas não

propriamente um registro de ocorrências no tempo. Reconhecemos que entre a tabela `dim_livro` e a tabela de facts a granularidade dos dados é igual, mas como a escala do trabalho é pequena, a performance overhead do join entre as duas tabelas é negligível pelo que decidimos manter assim. Iremos discutir mais à frente esta limitação.

Figura 1 Visualização do data model final



Com base no descrito, até então o nosso modelo de dados afigura-se pelo diagrama acima apresentado. De seguida, para completar a camada semântica, procedemos à criação de medidas que nos permitam aplicar a lógica de negócio e descrever e classificar padrões e tendências no catálogo da Amazon Kindle, do tipo:

3.1 Medidas base

Essencial para servir de denominador em cálculos de rácios (percentagens) e para validar a granularidade dos dados após a aplicação de filtros. Compõe os dois primeiros níveis da nossa pirâmide DIKW. Estas medidas funcionam como uma "camada de abstração". Ao centralizar a lógica de agregação (ex: SUM, COUNTROWS), garantimos que qualquer alteração na regra de cálculo se propaga automaticamente por todo o modelo.

- **Total de Livros (Inventário)**

```
Total Livros = COUNTROWS('Facts')
```

- **Total de Avaliações (Reviews)**

```
Total Reviews = SUM('Facts'[reviews])
```

- **Contagem de Estrelas**

```
Contagem Estrelas = COUNT(facts[stars])
```

- **Média Ponderada de Estrelas**

```
Media Estrelas Ponderada =  
DIVIDE(  
    SUMX('facts', 'facts'[stars] * 'facts'[reviews]),  
    SUM('facts'[reviews]),  
    BLANK()  
)
```

Para compreender o rating efetivo dos leitores, optámos por uma média ponderada pelo volume de reviews, em detrimento de uma média aritmética simples ou Bayesiana

A média ponderada segue a fórmula:

$$\bar{x} = \frac{\sum(w_i x_i)}{\sum w_i}$$

Em que x_i são os valores e w_i os pesos.

- **Média de Reviews**

```
Média de Reviews = AVERAGE('facts'[reviews])
```

- **Total Autores**

```
Total Autores = DISTINCTCOUNT('Facts'[author_id])
```

3.2 Medidas de negócio

Segmentação e booleans que visam criar contexto e significado com os dados e informações recolhidas. Aplicam filtros específicos e regras de negócio sobre as medidas base para isolar segmentos críticos do catálogo (ex: Best Sellers, Kindle Unlimited) e assim elevam a informação ao nível do conhecimento, pois contextualizam os números brutos com regras de negócio específicas

- **Total de Best Sellers**

```
Total Best Sellers =  
CALCULATE(  
    [Total Livros],  
    'dim_livro'[is_best_seller] = TRUE()  
)
```

Permite analisar o comportamento exclusivo dos livros de sucesso sem o "ruído" dos livros com fraco desempenho.

- **Percentagem de Best Sellers**

```
% Best Sellers =  
DIVIDE(  
    [Total Best Sellers],  
    [Total Livros],  
    0  
)
```

Criámos a medida % Best Sellers para normalizar os dados, permitindo comparar editoras grandes com editoras pequenas de forma justa, algo que a contagem absoluta não permitiria.

- **Total Kindle Unlimited**

```
Total Kindle Unlimited =  
CALCULATE(  
    [Total Livros],  
    'dim_livro'[is_kindle_unlimited] = TRUE()  
)
```

Permite segmentar o catálogo pelo tipo de distribuição (Subscrição vs Venda Direta) para entender se a adesão ao programa Kindle Unlimited tem correlação com maior número de reviews ou melhores avaliações.

3.3 Medidas de Preço e Valor

Como não dispomos de dados de transações diretamente, usamos proxies que nos permitam derivar a performance financeira do catálogo. As próximas medidas constituem indicadores compostos que cruzam dimensões financeiras (Preço) com dimensões de interação (Reviews e Estrelas) para aferir a percepção de valor e estimar receitas.

- **Preço médio ponderado**

```
Preço Médio Ponderado =  
VAR ReceitaEstimada = SUMX('Facts', 'Facts'[price] * 'Facts'[reviews])  
VAR TotalReviews = SUM('Facts'[reviews])  
RETURN  
DIVIDE(ReceitaEstimada, TotalReviews)
```

Para compreender o preço efetivo pago pelo mercado, optámos por uma média ponderada pelo volume de reviews, em detrimento de uma média aritmética simples ou Bayesiana.

- **Preço médio ponderado (Apenas Best Sellers)**

```
Preço Médio BestSell =  
CALCULATE(  
    [Preço Médio Ponderado],  
    'dim_livro'[is_best_seller] = TRUE()  
)
```

Ao comparar esta medida com o "Preço Médio Ponderado", conseguimos responder à pergunta: "Livros mais caros vendem menos?". Se o preço médio dos Best Sellers for inferior à média geral, indica uma sensibilidade ao preço por parte do consumidor.

- **Indicador custo benefício**

```
Score Custo Beneficio =  
DIVIDE(  
    AVERAGE('Facts'[stars]),  
    AVERAGE('Facts'[price]),  
    0  
)
```

Criámos um indicador de valor que relaciona a qualidade (Stars) com o custo (Price), permitindo identificar livros subavaliados.

- **Proxy Receita Estimada**

```
Valor Catálogo (Proxy) =  
VAR Multiplicador_Dinamico = [Multiplicador Conversao Stars Value  
]  
RETURN  
SUMX(  
    'Facts',  
    'Facts'[price] * [Contagem Estrelas] * Multiplicador_Dinamico  
)
```

Na ausência de dados de vendas reais, aplicámos uma técnica de proxy analytics, estimando a receita gerada por um livro através do volume de estrelas multiplicado por um fator de conversão (percentagem de compradores que deixam um rating), assim transformando dados de interação em métricas financeiras. Este conhecimento vem do nosso domain knowledge na área de marketing, onde benchmarks de taxas de conversão são frequentemente utilizadas para inferência dentro de um intervalo de confiança. As “estrelas” foram escolhidas em detrimento das avaliações escritas devido a apresentarem uma taxa de conversão maior, e, portanto, mais data points para uma proxy mais robusta. Reconhecemos, contudo, que esta métrica abstrai alguma granularidade, tal como géneros de livros diferentes apresentarem taxas de conversão de rating diferentes, constituindo assim um limite discutido mais à frente. Com isto em conta, o multiplicador escolhido é um “slider”, criado através do menu “Modelling” → “New Parameter” → “Numeric Range”. O intervalo é de 10 a 100, com um valor pré-definido de 20, sendo este o valor referência de conversão de vendas para avaliações escritas de artigos online. Através do parâmetro What-If (o “slider”), o nosso objetivo não é chegar a um valor exato de faturação, mas sim criar um índice de performance relativa para comparar categorias e editoras. O valor absoluto interessa menos que a proporção entre as categorias.

3.4 Medidas de Time Intelligence e Categorização

Medidas que manipulam o contexto temporal (crescimento YoY) ou criam segmentações virtuais (classificação dinâmica) que não existem fisicamente na base de dados. Permitem monitorizar tendências ao longo do tempo (evolução temporal) e simular cenários (através da segmentação dinâmica).

- **Classificação da popularidade do livro**

```

Status do Livro =
VAR RatingAtual = [Media Estrelas Ponderada]
RETURN
IF(
    ISBLANK(RatingAtual),
    "Sem Avaliação",
    SWITCH(
        TRUE(),
        RatingAtual >= 4.5, "Excelência",
        RatingAtual >= 4.0, "Muito Bom",
        RatingAtual >= 3.0, "Médio",
        RatingAtual < 3.0, "Precisa Melhorar",
        "Sem Avaliação"
    )
)

```

Esta medida não retorna um número, mas ajuda a criar títulos dinâmicos ou classificação em visualizações. Isto facilita a leitura por parte de decisores não técnicos e permite criar filtros visuais intuitivos que não existem na fonte de dados original, assim como aplicar em algoritmos, compondo a nossa camada semântica.

- **Crescimento anual de publicações**

```

YOY Crescimento Publicacoes % =
VAR CurrentYearCount = [Total Livros] – Medida base
VAR PreviousYearCount =
    CALCULATE(
        [Total Livros],
        SAMEPERIODLASTYEAR(dim_calendario[published_date])
    )
RETURN
IF(
    PreviousYearCount = 0,
    BLANK(),
    DIVIDE(CurrentYearCount - PreviousYearCount, PreviousYearCount)
)

```

Esta métrica de *"time intelligence"* permite monitorizar a velocidade de expansão do catálogo da Amazon Kindle. Do ponto de vista de negócio, permite para identificar tendências de saturação ou de crescimento da oferta. Um valor positivo indica uma entrada líquida de novos títulos superior ao ano anterior, sugerindo uma categoria em expansão ou um aumento na atividade editorial. Valores negativos ou estagnação podem sinalizar um amadurecimento do mercado ou uma consolidação de categorias.

Juntamente com outras medidas de negócio, tal como o indicador da subscrição Kindle Unlimited, permite avaliar a expansão destes subsets do catálogo.

4. VISUALIZAÇÃO DOS DADOS – DESENVOLVIMENTO DA SOLUÇÃO EM BI

A presente solução de Business Intelligence foi implementada sobre a plataforma Microsoft Power BI Desktop, adotando uma metodologia de data storytelling. Esta abordagem narrativa conduz o utilizador desde uma perspetiva geral, evidenciado pela primeira página intitulada “Visão Geral”, para insights granulares referentes ao preço e à performance editorial tal como a página “Parceiros e Modelo de Negócio”, permitindo uma análise progressiva e estruturada dos dados.

Com isto, é de ressaltar que a visualização foi elaborada de forma holística e com a interatividade em mente, em que os visuais enriquecem-se mutuamente de contexto e podem ser filtrados de acordo com a temática que o utilizador queira explorar.

4.1. Diagrama do Layout e Princípios de Design

A conceção do layout visa maximizar a legibilidade e a facilidade de uso do dashboard. A estrutura base, transversal a todas as páginas do relatório, organiza-se de forma hierárquica e funcional. No topo da interface, situa-se o cabeçalho de contexto (header), uma zona reservada para a identificação da página e para a aplicação de filtros globais (slicers). Esta funcionalidade permite ao utilizador segmentar transversalmente todos os elementos visuais por variáveis críticas como o ano.

Ainda no cabeçalho, ao lado, dispõem-se os cartões de KPIs. Estes cartões apresentam métricas fundamentais, tal como o total de livros, o volume de críticas, médias de avaliação e receita estimada, proporcionando uma leitura imediata da saúde do catálogo. O corpo principal da página alberga os gráficos interativos, os quais se encontram dispostos segundo uma lógica de relevância de leitura, da esquerda para a direita e de cima para baixo.

4.2. Descrição das Páginas e Visualizações

A arquitetura analítica consubstancia-se em quatro páginas principais de análise consolidada. A primeira página, dedicada à visão geral do catálogo, atua como um painel de controlo central para a gestão. Nesta secção, os KPIs de topo oferecem uma fotografia instantânea da dimensão e saúde do catálogo, inclusive da subscrição Kindle Unlimited, permitindo aferir a escala da operação através do volume de inventário e das interações, bem como a qualidade percebida por via da média ponderada de estrelas. A análise temporal é

materializada num gráfico de linhas que monitoriza o crescimento das publicações ao longo dos anos, identificando tendências de expansão ou contração. Complementarmente, a distribuição por categoria revela a concentração da oferta editorial, enquanto um gráfico de rosca ilustra a taxa de penetração do programa Kindle Unlimited, um indicador vital para a estratégia de fidelização da plataforma.

A segunda página foca-se na estratégia de preço, dedicando-se à análise da elasticidade e do posicionamento de mercado. Esta secção cruza a oferta existente com a valorização da procura. Através de um histograma, analisa-se a distribuição de preços para identificar a concentração da oferta, distinguindo segmentos mais baratos dos mais caros. A elasticidade da procura é examinada num gráfico comparativo que contrapõe os intervalos de preço com o volume de ratings (estrelas), utilizado aqui como proxy da procura com o intuito de determinar o "preço ótimo" que maximiza a interação dos leitores. Adicionalmente, a relação preço-qualidade é escrutinada através da sobreposição da média de ratings aos intervalos de preço, testando a premissa de que livros com custo mais elevado são percecionados como detentores de maior qualidade.

A terceira página aborda os parceiros e modelo de negócio, centrando-se na análise dos parceiros de conteúdo que geram maior valor. Recorre-se a um ranking financeiro, baseado numa métrica de "Receita Estimada (Proxy) Por Editora", para listar as editoras estrategicamente mais relevantes. Esta visualização permite distinguir claramente entre editoras de volume, caracterizadas por muita oferta e baixa interação, e editoras de eficiência, que, apesar de um menor número de títulos, produzem Best Sellers. A análise conclui-se com o estudo do impacto do Kindle Unlimited, comparando a performance média entre os títulos incluídos e excluídos do programa, servindo de suporte à tomada de decisão para a inclusão de novas obras no plano de subscrição.

Por fim, a quarta página aborda a qualidade do catálogo, centrando-se na métrica de ratings (estrelas) para entender o desenvolvimento da qualidade percecionada ao longo do tempo. Com isto, no primeiro quadrante encontramos um scatterplot que relaciona o preço médio simples com a avaliação média ponderada, permitindo aferir a diferença no nível crítico para os preços médios praticados em cada categoria, e por esta via encontrar categorias de maior valor que são bem rececionadas pelos leitores, indicado produtos de maior valor percecionado e como tal, maiores margens e royalties. Esta página comporta também duas outras tabelas, uma no canto inferior esquerdo que classifica cada género literário com base nas avaliações, para uma interpretação rápida, e outra do lado direito, que compara a expansão do catálogo às mudanças

na percepção de qualidade ao longo dos anos, permitindo facilmente perceber se o catálogo se tem ou não adaptado aos gostos correntes dos círculos de leitores.

5. DESENVOLVIMENTO DO ALGORITMO DE DATA MINING – MODELAGEM PREDITIVA

Esta secção tem como objetivo principal a aplicação de metodologias de *Data Mining* ao nosso *dataset*, e desenvolver um modelo preditivo robusto.

5.1. Abordagem Metodológica Explorada: Classificação

A primeira abordagem metodológica experimentada foi uma classificação multiclasse.

a. Definição da Variável Alvo (Target Variable)

A variável-alvo inicial foi a métrica derivada do “Status do Livro”, com base nas “reviews” e nas “stars”, constituindo um problema de classificação multiclasse.

Ao prever o “Status do Livro”, o modelo permite ao editor tomar decisões estratégicas em áreas como o marketing, ao investir mais na promoção dos livros classificados como prováveis Top-Sellers, e na Revisão, ao reavaliar a estratégia de *pricing* ou a categoria de livros classificados como Low-Performers.

b. Desafios e Limitações

- Apesar do alinhamento estratégico, a implementação da classificação multiclasse enfrentou desafios críticos, que comprometeram a sua viabilidade:

- Tratamento do desequilíbrio de classes: A distribuição da variável-alvo revelou um forte desequilíbrio de classes, com a maioria dos registos a pertencer à classe “Low-Performer” (~63.5%), contrastando com a classe minoritária “Top-Seller” (~10.3%). Para mitigar o risco de o modelo negligenciar as classes minoritárias mais importantes para o negócio, optou-se por utilizar um parâmetro que balanceia o peso das classes no modelo de classificação (`class_weight = ‘balanced’`), ponderando o custo do erro de forma inversa à frequência das classes, combatendo o enviesamento.

- Redução de Dimensionalidade: As *features* categóricas de alta cardinalidade, nomeadamente `author_name` e `category_name`, exigiram uma técnica de redução de dimensionalidade. Para evitar um erro de memória (*MemoryError*), procedemos às seguintes estratégias:

- Autores: Definiu-se um critério de agrupamento para autores que ocorrem menos de 5 vezes, de modo a reter o sinal preditivo dos autores mais frequentes.
- Categorias: Definiu-se um critério de agrupamento para 200 ocorrências, reduzindo o número de colunas geradas pelos *One-Hot Encoding* e o ruído do modelo.

c. Conclusão da classificação

Foi selecionado o algoritmo `RandomForestClassifier`, dada a sua robustez e boa performance com dados categóricos, sendo otimizado com um número de estimadores (`n_estimators = 200`) e uma maior profundidade máxima (`max_depth = 20`), para combater o *underfitting* e melhorar a capacidade de generalização.

Apesar dos esforços para obter um bom resultado nas métricas de avaliação, estas foram constantemente insatisfatórias, pois o modelo falhava em identificar a grande maioria dos verdadeiros sucessos. Não sendo este desempenho fiável para sustentar as decisões do negócio, tomamos a decisão em mudar o foco para uma abordagem tecnicamente mais robusta.

5.2. Abordagem Final: Regressão

Face à inviabilidade técnica da classificação, a estratégia pivotou para a Regressão. Esta mudança foi metodologicamente justificada, pois, embora a regressão ofereça um insight menos direto para a tomada de decisão em BI, provou ser o método mais robusto e eficaz para a previsão quantitativa da popularidade do *dataset*, lidando melhor com dados esparsos e contínuos.

a. Definição do Target e Tratamento

A coluna “reviews” foi selecionada como o *proxy* mais fiável para a popularidade e sucesso comercial de um livro na plataforma Kindle. O número de *reviews* é um indicador direto do *engagement* e do volume de leitores.

A esta variável foi realizada uma transformação logarítmica, para mitigar o enviesamento positivo (*skewness*) e a heterocedasticidade (variância não constante), inerente aos dados de

contagem. Nisto, foi aplicada a transformação $\log(1+x)$. Esta etapa foi essencial para estabilizar a variância e assegurar que os modelos de regressão pudessem treinar de forma mais eficaz.

b. Engenharia e Seleção de Variáveis (Feature Engineering and Selection)

Para garantir que o modelo se mantivesse preditivo no contexto de um novo lançamento, foram mantidas as variáveis estáticas:

- variáveis numéricas: “price” e “stars”
- variáveis categóricas: “author_name”, “category_name”
- variáveis binárias: “isEditorsPick”, “isGoodReadsChoice”, “isKindleUnlimited”

A variável binária “Best Seller” foi propositalmente descartada para evitar data leakage, dado ser um dado futuro.

c. Desafio da alta cardinalidade e solução

As variáveis “author” e “category_name” apresentaram uma alta cardinalidade, resultando na criação de dezenas de milhares de colunas através do *OneHotEncoder*. Esta densidade excessiva causou erro de memória em algoritmos não lineares, inviabilizando o treino.

Para resolver este *MemoryError*, e devido à complexidade do *Random Forest Regressor*, a estratégia de redução de dimensionalidade foi ajustada, e foi aplicada a técnica de *Binning Manual*, que consistiu em agrupar valores menos frequentes em categorias únicas, focando o *OneHotEncoder* apenas nos top 50 autores e top 50 categorias. Preservando o sinal preditivo das entidades mais relevantes, enquanto o ruído das entidades raras era mitigado.

d. Implementação e Otimização do modelo

O modelo preditivo escolhido foi o *RandomForestRegressor* por ser um dos algoritmos mais robustos e eficazes para dados com relação não lineares e de alta dimensionalidade.

- Arquitetura: foi utilizado um Pipeline com um *ColumnTransformer* para processar as variáveis de forma rigorosa, onde as features numéricas foram sujeitas a um *StandardScaler*, e as features categóricas foram codificadas com *OneHotEncoder*.
- Otimização: a performance foi maximizada através do *GridSearchCV*, que explorou diferentes combinações de hiperparâmetros. Esta otimização resultou na

identificação da melhor configuração, que permitiu ao modelo atingir a sua capacidade preditiva máxima.

Foi também realizada uma análise de diagnóstico da importância das variáveis (Feature Importance) do modelo final treinado. Esta análise, baseada na redução da impureza, permitiu identificar as features mais influentes (as categorias). Contudo, testes subsequentes com um modelo simplificado (utilizando apenas as Top 5 variáveis mais importantes) não resultaram num aumento do poder preditivo, o que justificou a manutenção do modelo completo e otimizado para maximizar o desempenho.

6. RESULTADOS, ANÁLISE E DISCUSSÃO

6.1 Business Intelligence

A apresentação dos resultados decorre da exploração interativa do dashboard desenvolvido, estruturando-se em três vetores fundamentais: a validação das métricas de aproximação (*proxies*), a análise da estratégia de preço e a avaliação da performance editorial. A interpretação crítica destes dados permitiu transformar um *snapshot* estático do catálogo num instrumento de apoio à decisão estratégica. Dada a ausência de dados de vendas reais no dataset público, a fiabilidade dos resultados assenta na correlação assumida entre o volume de ratings e o volume de vendas.

A implementação da medida "Receita Estimada (Proxy)", ajustável via parâmetro (What-If), permitiu mitigar a incerteza. Observou-se que, independentemente do multiplicador de conversão selecionado, as tendências relativas mantêm-se constantes: as categorias com maior tração e os intervalos de preço mais rentáveis não se alteram, validando a robustez do modelo para análise comparativa, ainda que os valores absolutos sejam estimativos.

A análise da visão macroscópica (Página 1) revela um ecossistema editorial em franca expansão, dominado por uma forte concentração em categorias técnicas e académicas. Estas categorias representam o "motor" de volume neste dataset em particular. Contudo, pode ser uma sub-representação derivada da natureza do dataset via *web scraping*, um ponto discutido mais adiante.

Um dos insights mais relevantes extraídos reside na penetração do programa Kindle Unlimited. A visualização da distribuição do catálogo demonstra que a maioria dos títulos está integrada neste modelo de subscrição. A análise temporal (Crescimento YoY) sugere que a adesão a este programa tem sido um catalisador fundamental para a entrada de novos títulos no mercado, baixando as barreiras de entrada para novos autores e garantindo um fluxo constante de novidades para os leitores.

A investigação sobre a estratégia de pricing (Página 2) permitiu determinar o alinhamento entre a oferta e a procura. O histograma de distribuição de preços evidencia que a grande maioria do inventário se concentra na faixa "low-cost" (abaixo dos 10\$), uma consequência natural da forte presença de autores independentes e da pressão competitiva.

Ao cruzarmos estes dados com o volume de ratings (o nosso proxy de vendas) e com o indicador de qualidade percebida (Média de Estrelas Ponderada), corrobora-se que os livros com melhor performance comercial são os mais baratos.

A análise de performance editorial (Página 3) destaca uma transformação profunda na cadeia de valor do livro. O ranking de receita estimada é liderado não pelas editoras tradicionais centenárias, mas pelos serviços próprios da Amazon e pelas publicações independentes (Independently Published). Isto ocorre, pois, a maioria das publicações da Amazon decorrem da plataforma “Kindle Direct Publishing”, que é uma plataforma de auto-publicação online.

Por fim, a análise detalhada por categoria (Página 4) permitiu identificar "oceanos azuis" de oportunidade. As categorias de não-ficção e técnicas demonstram rácios de eficiência superiores face às categorias de ficção que apresentam sinais de saturação (elevadíssima oferta para uma procura fragmentada). A matriz de dispersão entre preço e avaliação evidencia que nestes nichos específicos é possível praticar preços significativamente acima da média do catálogo sem penalizar a satisfação do leitor, sugerindo uma oportunidade estratégica para a diversificação do inventário da plataforma em direção a conteúdos mais especializados e de maior valor acrescentado.

Em suma, os resultados demonstram que o sucesso na plataforma Amazon Kindle é multifatorial, dependendo de um equilíbrio delicado entre a inclusão em programas de subscrição (para visibilidade), uma estratégia de preço que sinalize qualidade (evitando o preço mínimo) e a gestão ativa da reputação (avaliações e reviews) junto da comunidade de leitores.

6.2 Modelagem Preditiva e Data Mining

A fase inicial de classificação multiclasse, com o target “Status dos Livros” foi uma tentativa metodológica de criar uma inteligência acionável de forma direta, permitindo a categorização de Top-Seller ou Low-Performer de forma binária e imediata para os decisores. Os esforços para mitigar o forte desequilíbrio das classes e as técnicas de feature engineering não se revelaram suficientes. As métricas de desempenho foram insatisfatórias, nomeadamente o recall para a classe minoritária Top-Seller que se manteve consistentemente abaixo de 21%. Este desempenho instável significava que o modelo falhava em identificar os verdadeiros sucessos com a precisão exigida pelo negócio, tornando a abordagem metodologicamente inviável, forçando o pivot para a Regressão.

Após a implementação da Pipeline de regressão, utilizando o RandomForestRegressor e a otimização com o GridSearchCV, com os melhores parâmetros $n_estimators = 150$ e $max_depth = 20$, o modelo apresentou os seguintes resultados no test set:

Tabela 3 Resultados do test set

Métrica de Avaliação	Resultado
R^2 (Coeficiente de Determinação)	0.8595
RMSE (Root Mean Squared Error)	1.1947
MSE (Mean Square Error)	1.4273

O R^2 de 0.8595 atesta a elevada capacidade do modelo de regressão implementado, indicando que este valor indica 85,95% da variância na popularidade dos livros (o volume de reviews na escala logarítmica) é explicada e prevista pelas *features* estáticas (preço, stars, autor, categoria...).

Este é um resultado estatisticamente robusto para um problema do mundo real, com dados ruidosos, e valida três aspetos cruciais da metodologia:

- Validação do feature engineering: o *binning* manual foi eficaz a extraír o sinal preditivo das variáveis de alta cardinalidade sem incorrer a erros de memória ou overfitting.
- Validação da escolha do target: a transformação logarítmica foi bem-sucedida a estabilizar o target, permitindo ao RandomForestRegressor capturar relações não lineares com alta precisão.
- Viabilidade: o sucesso do modelo de regressão justifica a sua escolha como solução principal, superando as limitações técnicas impostas pelo desequilíbrio do dataset de classificação.

A interpretação do RMSE de 1.1947, que representa a magnitude média do erro de previsão, pode ser considerado um valor controlado e baixo, o que confirma que o modelo está bem ajustado. É essencial notar que esta medida está na escala logarítmica, não na escala de contagem bruta. Um RMSE nesta escala indica que o erro não é excessivo em nenhuma região do dataset e, crucialmente, que não há evidências de overfitting, pois o modelo consegue generalizar bem para o test set.

6.3 Análise da Importância das Features e o Dilema da Simplificação

Foi realizada uma experiência de análise da importância das features, onde se verificou que as 5 categorias mais importantes a serem: Literature & Fiction, Science Fiction & Fantasy, Teen & Young Adult, Nonfiction e Biographies & Memoirs.

Contudo, a comparação metodológica demonstrou um dilema da simplificação:

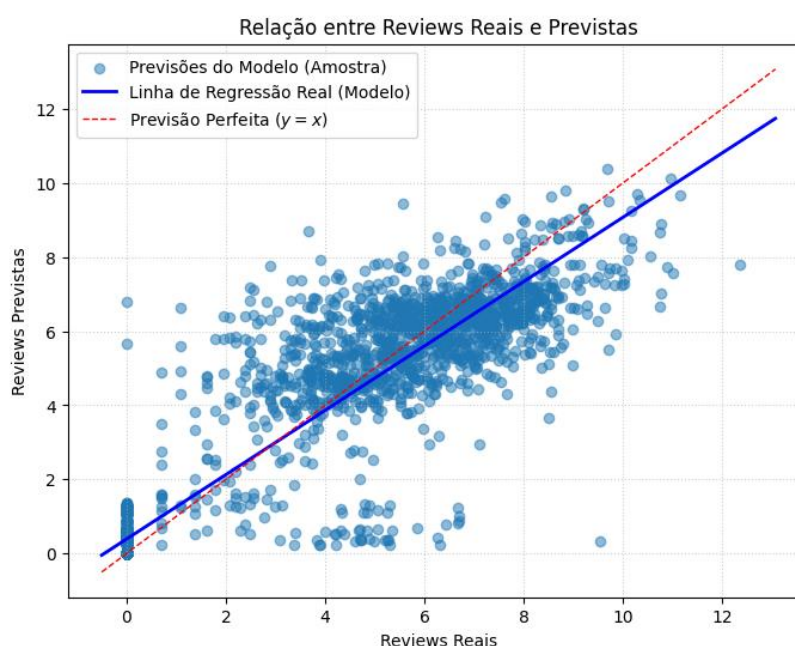
- A tentativa de criar um modelo mais simples e mais interpretável, restringindo as *features* apenas às Top 5 variáveis identificadas como mais importantes, resultou numa perda de poder preditivo, obtendo um R^2 de 0.8129 e um RMSE de 1.3782.

Esta conclusão é fundamental: O desempenho de R^2 de 0.8595 e o baixo RMSE do modelo final dependem da contribuição sinérgica de todas as variáveis *binned* e processadas (incluindo as centenas de autores binados e as categorias menos influentes). O modelo Random Forest exige esta complexidade de *features* para otimizar o resultado final.

6.4. Interpretação do Gráfico LOG-LOG

A performance robusta do modelo, quantificada pelo R^2 , é validada através da análise do gráfico de dispersão que compara os valores de Reviews Reais e Previstas, ambos na escala logarítmica $\log(1+x)$ (Figura 2).

Figura 2 Gráfico de dispersão entre as Reviews Reais e Previstas, transformadas numa escala logarítmica



Para garantir uma representação gráfica fiel foram adotados procedimentos, nomeadamente uma transformação logarítmica, domínio em que o modelo foi treinado, permitindo avaliar a sua acurácia no espaço onde a variância é estável, e também a mitigação de overplotting através de uma estratégia amostragem aleatória de 10% dos pontos do test set e uma baixa transparência ($\alpha=0.5$), permitindo revelar a banda de erro subjacente, que estaria oculta pela concentração extrema de dados.

A observação mais importante do gráfico é a forte sobreposição da linha de regressão real do modelo com a linha de tendência, sendo esta sobreposição é a evidência visual direta de três pontos analíticos:

- Validação do R^2 : A proximidade da nuvem de pontos à linha $y = x$ confirma que o modelo RF estabeleceu uma relação linear robusta, validando estatisticamente que 85,95% da variância é explicada pelas features de entrada.
- Bias negligenciável: A inclinação da reta de regressão indica que o modelo não sobrestima nem subestima sistematicamente as previsões, apresentando um bias muito baixo.
- Precisão nos valores comuns: A banda de dispersão é muito estreita na região de valores baixos e médios de reviews (que é a maioria do dataset). Isto demonstra que o modelo é ótimo na previsão de livros de popularidade comum, onde a maioria das observações está concentrada.

LIMITAÇÕES E TRABALHO FUTURO

O desenvolvimento desta solução de BI, embora robusto na sua arquitetura técnica, operou sob constrangimentos inerentes à natureza dos dados disponíveis, os quais condicionam a interpretação absoluta dos resultados. Primeiramente, não é possível garantir a representatividade total do catálogo Amazon, pois o método de extração original pode ter privilegiado livros com maior visibilidade (rankings) ou as recomendações algorítmicas, dependendo da técnica de *web scraping* e *crawling* utilizada. O autor não especifica os detalhes da captura de dados. Outra limitação primordial reside na ausência de dados transacionais reais, nomeadamente volumes de vendas e faturação. Para contornar esta lacuna, o estudo recorreu a técnicas de proxy analytics, assumindo uma correlação direta entre o volume de ratings (estrelas) e o sucesso comercial. Embora esta seja uma prática aceite na indústria na falta de dados proprietários, leva um enviesamento de representatividade, dado que a propensão para avaliar pode variar consoante o género literário ou a demografia do leitor, podendo inflacionar a perceção de sucesso em categorias com comunidades mais ativas em detrimento de outras. Nesta implementação, serve como um índice de performance relativa para comparar categorias e editoras.

Adicionalmente, a natureza estática do dataset, um *snapshot* único do catálogo, impõe restrições severas à análise temporal. Embora seja possível analisar a data de publicação, não dispomos do histórico de evolução de preços ou da acumulação de reviews ao longo do tempo. Esta característica introduz um "enviesamento de antiguidade", onde títulos publicados há mais tempo surgem naturalmente com métricas acumuladas superiores, dificultando uma comparação justa de desempenho com lançamentos recentes. A impossibilidade de analisar a sazonalidade de vendas ou o impacto de campanhas promocionais de curto prazo limita a capacidade do modelo em fornecer recomendações táticas de pricing dinâmico.

Ao nível do processamento de dados (ETL), as opções tomadas, embora justificadas pela eficiência, introduzem as suas próprias limitações. A decisão de categorizar os preços em intervalos fixos (binning de 5\$), fundamental para a segmentação visual, ocultou nuances de preços psicológicos (como a diferença entre 9.99\$ e 10.00\$) que poderiam ser determinantes na análise de elasticidade da procura. Já do ponto de vista de ML, o mais pertinente seria repartir os dados contínuos em decis para testar modelos que requeiram uma frequência distribuída. Do mesmo modo, o processo de limpeza de nomes de autores, apesar de ter reduzido a dispersão dos dados, não garante a desambiguação total de homónimos, o que pode

resultar numa ligeira sobrevalorização da performance de autores com nomes comuns. O mesmo se aplica para editores. Nos editores surge o problema de “Slowly Changing Dimensions” em que os nomes das editoras foram mudando ao longo do tempo com as fusões e mudanças do tecido económico. Neste caso foi preservada nomenclatura histórica, e não foi aplicado um SCD Type 2, que seria o mais adequado para garantir factualidade histórica. A estrutura de dados assumida, com uma relação de um-para-um entre a tabela de factos e a dimensão livro, embora adequada para este *snapshot*, careceria de revisão para um modelo de monitorização contínua onde o histórico de alterações de atributos do livro necessitaria de ser preservado.

Por fim, potencialmente a maior limitação deste trabalho que ultrapassa as barreiras técnicas, é a impossibilidade de comunicar com stakeholders da plataforma produtora destes dados, neste caso a Amazon Kindle, pois seriam as fontes de informação primárias para ajudar a criar a camada semântica: determinar as métricas a definir, aplicar lógica de negócio, levantar perguntas exploratórias a realizar ao dataset, assim como confirmar a sua utilidade prática, relevando a importância dos primeiros passos do processo CRISP-DM.

Incidindo sobre o modelo preditivo, o projeto foi desenvolvido sob um prazo e a recursos computacionais limitados. Por esses motivos, alguns conceitos foram pensados, mas não explorados:

- Exploração limitada de modelos: A otimização do RandomForestRegressor (via GridSearchCV) consumiu recursos significativos. Devido à falta de tempo e capacidade computacional, não foi possível explorar outras arquiteturas de ensemble (como o XGBoost) que poderiam potencialmente melhorar o desempenho, ao lidar com a alta dimensionalidade esparsa resultante do OneHotEncoder.
- Validação cruzada: A aplicação de GridSearchCV foi limitada a uma divisão simples de k-folds e um espaço de hiperparâmetros (o grid) restrito, para evitar tempos de processamento excessivos.
- Embora a classificação multiclasse tenha sido rejeitada devido ao baixo recall, o desequilíbrio de classes pode ser melhor explorado através de técnicas mais avançadas:
- Técnicas de balanceamento: Para além do `class_weight='balanced'` (que apenas penaliza os erros da minoria), poderiam ser testadas abordagens como SMOTE

(Synthetic Minority Over-sampling Technique) (Chawla et al. (2011)), ou métodos de amostragem que otimizam a fronteira de decisão.

- Modelos mais especializados: O algoritmo Support Vector Machine (SVM) ou redes neurais mais simples poderiam ter sido explorados, pois lidam de maneira diferente com a separação de classes em espaços de alta dimensão.

a. Propostas para Trabalho Futuro

Apesar dos resultados robustos alcançados com esta solução de BI, é fundamental reconhecer as limitações metodológicas e operacionais encontradas. Face ao exposto, numa instância futura consideramos pertinente aplicar técnicas de *string similarity* e fontes externas, tal como o número ISBN ou DOI, para desambiguar homónimos de autores e editores.

Já a mudança de nomes de editoras beneficiaria da aplicação de técnicas SCD, nomeadamente tipo 2. Conforme definido por Kimball e Ross (2013), a técnica de Slowly Changing Dimensions do Tipo 2 é fundamental para preservar o histórico, uma vez que: "add a new row in the dimension with the updated attribute values" (p. 54), garante a integridade referencial dos dados históricos para visualização histórica e reprodutibilidade de relatórios, mas ao mesmo tempo permite, para efeitos de análise, agregar pela mesma entidade atual.

Quanto à modelação de algoritmos preditivos, podem ser exploradas diferentes estratégias, nomeadamente features contextuais, através de técnicas de processamento de linguagem natural (NLP), como embedding de palavras (Word2Vec ou FastText), para capturar novas características de features categóricas, como tom e o tópico exato do livro, gerando um sinal preditivo ainda mais forte.

Além disso, também se podem testar novos algoritmos de Boosting (como XGBoost ou LightGBM), que são frequentemente mais rápidos e eficientes que o Random Forest em datasets de grande escala e podem proporcionar um aumento marginal do R^2 .

Também deve ser realizada uma reavaliação do modelo de classificação, e testar técnicas de balanceamento de classes mencionadas (SMOTE), para garantir que a classificação binária atinja um Recall aceitável pelo negócio (ex: >75%).

CONCLUSÕES

O presente trabalho permitiu consolidar as competências adquiridas na Unidade Curricular, demonstrando a aplicação prática da pirâmide do conhecimento na transformação de dados brutos em inteligência acionável.

Através da implementação de uma arquitetura de BI robusta, baseada num modelo dimensional em estrela, foi possível superar os desafios inerentes a dados não estruturados e de alta cardinalidade. A análise descritiva no Power BI revelou insights fundamentais, nomeadamente que a estratégia de "preço mínimo" não é garantia de sucesso e que o programa Kindle Unlimited atua como um catalisador de volume para novos autores.

Na vertente de Data Mining, o projeto evidenciou a importância da iteratividade do processo KDD. A transição de uma abordagem de classificação (que se revelou ineficaz) para uma abordagem de regressão com Random Forest permitiu atingir um modelo preditivo robusto (R^2 de 0.8595), validando a hipótese de que é possível estimar a popularidade de uma obra com base nos seus metadados.

Contudo, reconhecem-se limitações, principalmente a ausência de dados transacionais reais (vendas efetivas) e a natureza estática do dataset (snapshot), o que limitou a análise de tendências temporais de preços.

Como trabalho futuro, sugere-se a integração de técnicas de Processamento de Linguagem Natural (NLP) para analisar o conteúdo das sinopses e a implementação de um processo de Slowly Changing Dimensions (SCD) para rastrear alterações históricas nas editoras. Em suma, a solução desenvolvida oferece uma ferramenta analítica capaz de apoiar editores e autores na otimização das suas estratégias de publicação no ecossistema Amazon.

Fundamentalmente, concluímos este trabalho com uma visão muito mais clara das diferentes fases do processo CRISP-DM, e da importância de em BI, orientar o trabalho para extrair *insights* tangíveis e claros para os *stakeholders*, e não apenas construir um monumento técnico.

BIBLIOGRAFIA

- Asaniczka. (2023). *Amazon Kindle Books Dataset 2023 (130K Books)* [Data set]. Kaggle. <https://doi.org/10.34740/KAGGLE/DS/3808504>
- Botchkarev, A. (2019). A new typology design of performance metrics to measure errors in machine learning regression algorithms. *Interdisciplinary Journal of Information, Knowledge, and Management*, 14, 45–76. <https://doi.org/10.28945/4184>
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2011). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321–357. <https://doi.org/10.1613/jair.953>
- Duque, J. (2025). Fundamentos de Business Intelligence e Análise de Dados [Manual da Unidade Curricular Não Publicado]. Instituto Politécnico de Gestão e Tecnologia (ISLA).
- Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI Magazine*, 17(3), 37–54. <https://doi.org/10.1609/aimag.v17i3.1230>
- Han, J., Kamber, M., & Pei, J. (2011). *Data mining: Concepts and techniques* (3rd ed.). Morgan Kaufmann.
- Joseph, V. R., & Vakayil, A. (2022). SPlit: An optimal method for data splitting. *Technometrics*, 64(2), 166–176. <https://doi.org/10.1080/00401706.2021.1921037>
- Kimball, R., & Ross, M. (2013). *The data warehouse toolkit: The definitive guide to dimensional modeling* (3rd ed.). Wiley.
- Müller, A. C., & Guido, S. (2017). *Introduction to machine learning with Python: A guide for data scientists*. O'Reilly Media.

ANEXOS

Anexo 1: código Python da primeira fase de ETL.

```
import pandas as pd
import numpy as np

# [=====] Algoritmia [=====]
# Standardizar texto para obter vals únicos -> Criar colunas para tabela dim

def clean_text_column(series):
    """
    Standardizar uma series (coluna DF) para permitir deduplicar: preenche NAs, garante dtype
    string,
    remove espaços extra e aplica TitleCase (expectável TitleCase para
    nomes de entidades (autores e editoras)).
    """
    return series.fillna('Unknown').astype(str).str.strip().str.title()

def create_dimension_and_merge(df, source_col, dim_label_col, dim_id_col):
    """
    1. Extrai valores únicos da coluna de origem para deduplicar.
    2. Cria um DF de dimensão.
    3. Gera uma PK (surrogate key) baseado no index ordenado.
    4. Enriquece (merge) o DataFrame principal.

    df: DF de origem a enriquecer
    source_col: Coluna de origem a separar para dimensao
    dim_label_col: Nome da nova coluna do atributo
    dim_id_col: Nome da nova coluna PK
    """
    # Extrair únicos e criar DF
    unique_vals = df[source_col].unique()
    dim_df = pd.DataFrame(unique_vals, columns=[dim_label_col])

    # Criar PK (sort e reset index)
    dim_df = dim_df.sort_values(dim_label_col).reset_index(drop=True)
    dim_df[dim_id_col] = dim_df.index

    # Enriquecer a tabela original com o PK
    df_merged = df.merge(dim_df, left_on=source_col, right_on=dim_label_col, how='left')

    return df_merged

# [=====] Início do Processo [=====]
```

```

dataset_path = "./kindle_data-v2.csv"
kindle_df = pd.read_csv(dataset_path)

# ===== 1. Atributo Price (Binning)
ranges = [0, 5, 10, 15, 20, 25, 30, 40, 60, 80, 100, np.inf]
bin_labels = [
    '5 USD', '5-10 USD', '10-15 USD', '15-20 USD', '20-25 USD',
    '25-30 USD', '30-40 USD', '40-60 USD', '60-80 USD',
    '80-100 USD', '>100 USD'
]

kindle_df['price_range'] = pd.cut(kindle_df['price'], bins=ranges, labels=bin_labels)

# Converte missings em Unknown
kindle_df['price_range'] =
kindle_df['price_range'].cat.add_categories('Unknown').fillna('Unknown').astype(str)

# Criação da dimensão e merge (Price)
kindle_df = create_dimension_and_merge(
    kindle_df,
    source_col='price_range',
    dim_label_col='price_range_label',
    dim_id_col='price_range_id'
)

# ===== 2. Atributo Author
# Standardizar autor
kindle_df['author'] = clean_text_column(kindle_df['author'])

# Criação da dimensão e merge (Author)
kindle_df = create_dimension_and_merge(
    kindle_df,
    source_col='author',
    dim_label_col='author_name',
    dim_id_col='author_id'
)

# ===== 3. Atributo soldBy (editora (publisher))
# Standardizar soldBy
kindle_df['soldBy'] = clean_text_column(kindle_df['soldBy'])

# Criação da dimensão e merge (Publisher)
kindle_df = create_dimension_and_merge(
    kindle_df,
    source_col='soldBy',
    dim_label_col='publisher_name',

```

```

    dim_id_col='publisher_id'
)

# ===== Exportar para dar seguimento no PowerQuery
kindle_df.to_csv('kindle-dataset-python-staging.csv', index=False)

```

Anexo 2: Código Python para o desenvolvimento do modelo preditivo.

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split, KFold, GridSearchCV
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score

# --- 1. CARREGAMENTO E TRANSFORMAÇÃO DE DADOS ---
df = pd.read_csv("new_kindle_data.csv")
df = df.rename(columns={"reviews": "reviews_count"})

# Definir Features (X) e Target (y)
X = df.drop("reviews_count", axis=1)
# Aplicação da transformação log(1+x) ao target
y = np.log1p(df["reviews_count"])

# --- 2. FEATURE ENGINEERING (BINNING) ---
# Definir o número de bins/categorias principais
TOP_N_AUTHORS = 50
TOP_N_CATEGORIES = 50

# A) Binning do Autor
top_authors = X['author_name'].value_counts().nlargest(TOP_N_AUTHORS).index
X['author_binned'] = X['author_name'].apply(
    lambda x: x if x in top_authors else 'Author_Other'
)

# B) Binning da Category Name
top_categories = X['category_name'].value_counts().nlargest(TOP_N_CATEGORIES).index
X['category_name_binned'] = X['category_name'].apply(
    lambda x: x if x in top_categories else 'Category_Other'
)

```

```

# Definir as features finais a usar no modelo
numerical_features = ["price", "stars"]
categorical_features_binned = [
    "author_binned",
    "category_name_binned",
    "isEditorsPick",
    "isGoodReadsChoice",
    "isKindleUnlimited"
]
all_features_rf = numerical_features + categorical_features_binned

# Usar apenas as features processadas/binned para o modelo RF
X_rf = X[all_features_rf]

# --- 3. PRÉ-PROCESSAMENTO (COLUMN TRANSFORMER) ---
preprocess_rf = ColumnTransformer(
    transformers=[
        ('num', StandardScaler(), numerical_features),
        # OneHotEncoder usa sparse_output=False, adequado para RandomForest
        ('cat', OneHotEncoder(handle_unknown='ignore', sparse_output=False),
categorical_features_binned)
    ],
    remainder='drop'
)

# --- 4. DIVISÃO DOS DADOS ---
X_train_rf, X_test_rf, y_train_rf, y_test_rf = train_test_split(
    X_rf, y, test_size=0.2, random_state=42
)

# --- 5. PIPELINE FINAL (RANDOM FOREST) ---
# Modelo final com parâmetros do Grid Search
rf_model_final = RandomForestRegressor(random_state=42)

pipeline_rf = Pipeline(steps=[
    ("preprocess", preprocess_rf),
    ("regressor", rf_model_final)
])

# --- 6. GRID SEARCH (Otimização do Random Forest) ---
# Usando os melhores parâmetros identificados
param_grid_rf = {
    'regressor__n_estimators': [150],
    'regressor__max_depth': [20],
}

```

```

print("Iniciando Grid Search para o Random Forest...")

# Definir K-Fold para Cross-Validation (cv=3)
kf = KFold(n_splits=3, shuffle=True, random_state=42)

grid_search_rf = GridSearchCV(
    pipeline_rf,
    param_grid_rf,
    cv=kf,
    scoring='neg_mean_squared_error',
    verbose=1,
    n_jobs=-1
)

grid_search_rf.fit(X_train_rf, y_train_rf)

best_model_rf = grid_search_rf.best_estimator_
print("\nMelhores parâmetros do Random Forest:", grid_search_rf.best_params_)

# --- 7. AVALIAÇÃO FINAL ---
y_pred_rf = best_model_rf.predict(X_test_rf)

mse_rf = mean_squared_error(y_test_rf, y_pred_rf)
rmse_rf = np.sqrt(mse_rf)
r2_rf = r2_score(y_test_rf, y_pred_rf)

print("\n=== RESULTADOS FINAIS: RANDOM FOREST (OTIMIZADO) ===")
print(f"R² (Variação Explicada): {r2_rf:.4f}")
print(f"MSE: {mse_rf:.4f}")
print(f"RMSE: {rmse_rf:.4f}")

```