



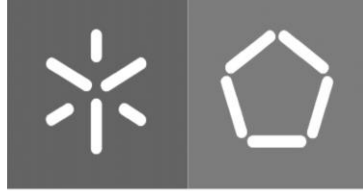
Universidade do Minho
Escola de Engenharia

Beatriz Barreto Henriques

Data Mining e Machine Learning:

Aplicação em Reaproveitamento de Stock

Dezembro de 2021



Universidade do Minho
Escola de Engenharia

Beatriz Barreto Henriques

Data Mining e Machine Learning:
Aplicação em Reaproveitamento de Stock

Dissertação de Mestrado

Mestrado em Engenharia de Sistemas

Trabalho realizado sob a orientação de

Professor Paulo Sérgio Lima Pereira Afonso

Professor Lino António Antunes Fernandes Costa

Dezembro de 2021

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

Este é um trabalho académico que pode ser utilizado por terceiros desde que respeitadas as regras e boas práticas internacionalmente aceites, no que concerne aos direitos de autor e direitos conexos.

Assim, o presente trabalho pode ser utilizado nos termos previstos na licença abaixo indicada.

Caso o utilizador necessite de permissão para poder fazer um uso do trabalho em condições não previstas no licenciamento indicado, deverá contactar o autor, através do RepositóriUM da Universidade do Minho.

Licença concedida aos utilizadores deste trabalho



Atribuição-NãoComercial-SemDerivações

CC BY-NC-ND

<https://creativecommons.org/licenses/by-nc-nd/4.0/>

AGRADECIMENTOS

Deixo algumas palavras de agradecimento às diversas pessoas e instituições que me apoiaram durante esta etapa.

Aos professores, Paulo Afonso e Lino Costa, pelo empenho e disponibilidade demonstrados e fornecerem as principais orientações para a conclusão desta dissertação.

Palavras de agradecimento ao Departamento de Produção da Decathlon Portugal pela oportunidade de realizar esta dissertação em contexto empresarial, assim como a toda a equipa pelo apoio.

Ao João Rodrigues e à Maria Pereira, por todo o apoio, ajuda e preocupação demonstrados ao longo do projeto desenvolvido.

Por fim, à minha família e amigos.

DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

Data Mining e Machine Learning: Aplicação em

Reaproveitamento de Stock

RESUMO

O desenvolvimento da tecnologia de informação gerou uma elevada quantidade de dados. Com este progresso, torna-se fulcral fornecer suporte ao agente de decisão através de análises estruturadas e em tempo real. Assim, instaura-se a necessidade de recorrer a técnicas de análise de dados que, a um nível estratégico, possibilitem estabelecer metas empresariais adequadas, e a nível operacional, auxiliem nas atividades funcionais da organização.

No quotidiano industrial, uma gestão de *stock* ineficiente e desajustada da realidade pode induzir a empresa em custos extraordinários. A presente dissertação aborda esta problemática para o universo de produção de bicicletas em Portugal pela Decathlon. Onde, o objetivo crucial incide na implementação de um sistema de apoio à decisão capaz de estabelecer padrões entre os dados e aprender de forma automática a analisar e associar diferentes tipos de componentes de uma bicicleta, de modo a usufruir do *stock* excedentário.

A metodologia implementada, com vista a obtenção do objetivo principal, inicialmente, recai sobre a implementação de métodos de *Machine Learning* que permitem transformar dados brutos em informação. Neste caso, a análise remete à tipologia de aprendizagem supervisionada para dados multiclasse e culmina no desenvolvimento de uma ferramenta de apoio ao agente de decisão. De modo a averiguar o melhor modelo mais indicado para responder à presente problemática foram implementados três modelos, *Decision Tree*, *Random Forest*, *Artificial Neural Network*. O modelo em rede apresenta, de forma geral, melhores resultados para os cenários desenvolvidos em comparação com os restantes modelos. Assim, a classificação dos componentes por tipologia incorre numa precisão de, aproximadamente, 95%.

Relativamente, à previsão de consumo médio do stock excedentário é de aproximadamente 85% num horizonte temporal de 5 anos.

Palavras-Chave: Análise de Dados, Apoio à Decisão, *Data Mining*, *Machine Learning*, *Stock*

Data Mining e Machine Learning: Stock Reuse Application

ABSTRACT

The development of information technology has generated a high amount of data, making it crucial to provide support to the decision agent through structured analysis in real-time. Thus, the need to implement data storage and analysis systems to support decision-making is required. At a strategic level, these systems can establish appropriate business goals, and at an operational level, they can assist the organization's functional activities.

In everyday industrial life, inefficient stock management and misadjusted to reality can induce the company in extraordinary costs.

This dissertation addresses this issue for the universe of bicycle production in Portugal by Decathlon. Where the crucial objective is the implementation of a system capable of establishing patterns between data and learning automatically to analyze and associate different types of components of a bicycle to take advantage of excess stock.

The methodology implemented, to obtain the main objective, initially falls on the implementation of Machine Learning methods that allow transforming raw data into information. In this case, the analysis refers to the supervised learning typology for multiclass data and culminates in the development of a decision support tool. To find the best model to respond to the present problem, three models were implemented, Decision Tree, Random Forest, Artificial Neural Network. The network model presents better results, in general, for the different scenarios compared to the other models. Thus, the classification of components by typology incurs an accuracy of approximately 95%.

Relatively, the forecast of the average consumption of the surplus stock is approximately 85% in a time horizon of 5 years.

Keywords: Data Analysis, Data Mining, Decision Making, Machine Learning, Stock

ÍNDICE

Agradecimentos.....	iii
Resumo.....	v
Abstract.....	vi
Índice.....	vii
Lista de Abreviaturas e Siglas	ix
Lista de Figuras.....	x
Lista de Tabelas	xii
1. Introdução.....	1
1.1 Enquadramento Geral	1
1.2 Motivação	2
1.3 Objetivos.....	3
1.4 Abordagem Metodológica	3
1.5 Estrutura do Documento.....	5
2. Estado de Arte	6
2.1 Gestão de inventário.....	6
2.1.1 Conceito de <i>Stock</i>	6
2.1.2 Custos Associados.....	7
2.1.3 Problemática de Manutenção de <i>Stock</i>	8
2.2 <i>Data Mining</i>	8
2.3 <i>Machine Learning</i>	10
2.3.1 Tipos de Aprendizagem.....	11
2.3.2 Modelos de Classificação	13
2.3.3 Análise de Desempenho	25
2.3.4 Exemplos de aplicação de <i>Machine Learning</i>	28
3. Caso de Estudo.....	30
3.1 Decathlon	30
3.2 Departamento de Produção Portugal	31
3.2.1 Produção de Bicicletas.....	32
3.2.2 Processo de Industrialização.....	33

3.2.3	Projeto <i>Zero Destruction</i>	34
4.	Métodos e Modelos Implementados	36
4.1	Exploração e Pré-processamento de dados	36
4.1.1	Análise Exploratória	36
4.1.2	Pré-processamento de Dados.....	42
4.2	Implementação de Modelos.....	42
4.2.1	<i>Decision Tree</i>	43
4.2.2	<i>Random Forest</i>	45
4.2.3	<i>Artificial Neural Network</i>	45
4.3	Ferramenta de apoio à decisão.....	47
5.	Análise e Discussão de Resultados	49
5.1	Experiência: Bicicleta <i>Adult</i>	49
5.2	Experiência: Bicicleta <i>Kids</i>	53
5.2.1	Comparação entre modelos	57
5.2.2	Impacto na análise financeira	57
6.	Conclusão	62
6.1	Limitações e Oportunidades para Trabalho Futuro	63
	Referências Bibliográficas	65

LISTA DE ABREVIATURAS E SIGLAS

ANN – Artificial Neural Networks

AUC – *Area under the curve*

BOM – Bill of Material

CRISP-DM - *Cross Industry Standard Process for Data Mining*

DM – *Data Mining*

DPP – Departamento de Produção Portugal

IG – *Information Gain*

ML – *Machine Learning*

MLP - *Multilayer Perceptron*

ROC – *Receiver Operating Characteristic*

LISTA DE FIGURAS

Figura 1 - Processo de CRISP-DM de acordo com (Chapman et al., 2000)	4
Figura 2 - Processo de DM de acordo com (Cabena et al., 1998)	9
Figura 3 – Relação DM com outras áreas de acordo com (Han et al., 2012).....	10
Figura 6 – Processo de Aprendizagem Supervisionada de acordo com (Mahesh, 2019).....	11
Figura 4 – Processo de aprendizagem não supervisionada de acordo com (Mahesh, 2019).....	12
Figura 5 – Processo de aprendizagem de Reforço de acordo com (Mahesh, 2019).....	13
Figura 7 – Estrutura base de uma árvore de decisão de acordo com (Janikow, 1998)	14
Figura 8 – Critério de divisão para classificação de acordo com (Han et al., 2012)	15
Figura 9 – Variação da Entropia perante o valor da probabilidade (Yin et al., 2019)	17
Figura 10 – Processo de desenvolvimento de Random Forest (Pal, 2005)	19
Figura 11 – Estrutura de uma ANN de acordo com (Gao & Su, 2020)	20
Figura 12 - Representação de uma operação de neurónio ANN (Vasilev et al., 2019)	21
Figura 13 – Fórmula e Representação gráfica da função ReLU (Vasilev et al., 2019).....	21
Figura 14 - Fórmula e Representação gráfica da função Softmax (Nwankpa et al., 2018)	22
Figura 15 – Exemplo para classificador binário da construção da matriz de confusão	26
Figura 16 – Representação gráfica de uma curva de ROC (Hanley & McNeil, 1982)	27
Figura 17 - Características da empresa à presente data (dezembro de 2021)	30
Figura 18 – Bicicleta e exemplo de uma frame e fork.....	32
Figura 19 - Diagrama de setor envolvidos na produção de bicicletas.....	33
Figura 20 – Comportamento por família em relação ao custo e quantidade de stock (adult).....	38
Figura 21 – Comportamento por tipologia de componentes para a quantidade de stock (adult).....	39
Figura 22 - Comportamento por tipologia de componentes para o custo (adult)	39
Figura 23 - Comportamento por família em relação ao custo e quantidade de stock (kids)	40
Figura 24 - Comportamento por tipologia de componentes para a quantidade de stock (kids)	41
Figura 25 - Comportamento por tipologia de componentes para o custo (kids).....	41
Figura 26 – Código referente à divisão de dados.....	43
Figura 27- Código do classificador em árvore	43
Figura 28 – Código para obtenção da accuracy do modelo.....	44
Figura 29 – Classificador em árvore com novos parâmetros.....	44
Figura 30 - Visualização de uma árvore de decisão.....	45

Figura 31 - Classificador de Random Forest	45
Figura 32 - Definição da estrutura de ANN	46
Figura 33 - Implementação de cross-validation	46
Figura 34 - Processo de construção de uma BOM	47
Figura 35 - Ferramenta de apoio à decisão	48
Figura 36 – Reporting presente na ferramenta de apoio à decisão.....	48
Figura 37 – Importância das variáveis para o estudo (adult)	51
Figura 38 - Matriz de Confusão (adult)	51
Figura 39 - Cross-Validation Performance (adult)	52
Figura 40 – Accuracy and loss of model (adult).....	53
Figura 41– Importância das variáveis para o estudo (kids)	54
Figura 42 - Matriz de Confusão (kids)	55
Figura 43 - Cross-Validation Performance (kids).....	56
Figura 44 - Accuracy and loss of model (kids)	56

LISTA DE TABELAS

Tabela 1 – Descrição de variáveis em estudo	37
Tabela 2 – Medidas descritivas (adult)	37
Tabela 3 – Componentes estruturais	38
Tabela 4 – Medidas descritivas (kids).....	40
Tabela 5 – Avaliação da performance das árvores de decisão (adult)	49
Tabela 6 - Medidas de avaliação (adult).....	50
Tabela 7 - Avaliação da performance de Random Forest (adult)	50
Tabela 8 - Avaliação da performance de ANN (adult)	52
Tabela 9 - Avaliação da performance das árvores de decisão (kids)	53
Tabela 10 - Medidas de avaliação (kids)	54
Tabela 11 - Avaliação da performance de Random Forest (kids)	54
Tabela 12 - Avaliação da performance de ANN (kids).....	55
Tabela 13 - Comparação entre modelos (kids)	57
Tabela 14 - Comparação entre modelos (adult)	57
Tabela 15 – Aproveitamento por tipo de componente (adult)	58
Tabela 16 – Previsão da capacidade produtiva (adult)	58
Tabela 17 – Lucro e preço médio e respetiva diferença (adult)	59
Tabela 18 - Previsão de liquidez de artigos (adult)	59
Tabela 19 - Aproveitamento por tipo de componente (kids).....	59
Tabela 20 - Previsão da capacidade produtiva (kids).....	60
Tabela 21 - Lucro e preço médio e respetiva diferença (kids).....	60
Tabela 22 - Previsão de liquidez de artigos (kids).....	60

1. INTRODUÇÃO

Neste capítulo introdutório procede-se à contextualização da temática abordada na presente dissertação, com o intuito de clarificar o enquadramento do problema em investigação. Em seguida, apresentam-se os objetivos propostos e é descrita, de forma sucinta, a abordagem adotada. Por fim, é apresentada a estrutura do documento.

1.1 Enquadramento Geral

Perante a atual era de dados, representada pela produção continua de dados em grande escala, surge a necessidade de recorrer a métodos capazes de transformar dados em informação à medida que estes são gerados. Este fator pode representar uma grande vantagem para as organizações, sejam estas de grandes ou pequenas dimensões, uma vez que permite obter uma elevada capacidade de adaptação a ambientes de elevada incerteza.

Por conseguinte, surge o interesse na recolha e análise dos dados existentes a todos os níveis da organização para extrair informações relevantes e, deste modo, dar suporte às organizações para o apoio à decisão (Sandryhaila and Moura 2014). A utilização efetiva de grandes quantidades de dados, aliada a ferramentas de visualização, atualmente, são vistos como essenciais para a obtenção de vantagem competitiva em relação a concorrentes, em todas as áreas de negócio (Kościelniak and Puto 2015).

O processamento e tratamento de dados e posterior análise através da implementação de modelos e técnicas para a resolução de problemas específicos de uma organização sintetiza o processo de transformação de dados em informação útil, como por exemplo, a exploração de tendências, otimização de métricas, deteção de padrões, entre outros. Perante os exemplos apresentados, é possível constatar que a extração de informação em grandes quantidades, recorre a técnicas com um grau de complexidade elevado, de modo a retirar o maior proveito dos dados que de outra forma seriam de difícil compreensão. Dentro das diversas técnicas existentes ressalta-se *Machine Learning* e *Data Mining* devido à sua aplicabilidade transversal a diversas áreas do conhecimento.

As várias entidades que operam nas mais diversas áreas podem, através da implementação destas técnicas, adquirir a capacidade de agir de forma eficaz de modo a evitar impactos negativos (Qiu et al. 2016). Além disso, *Machine Learning* remete à aprendizagem automática, ou seja, permite realizar e analisar processos sem que esta ação seja expressamente programada para tal.

Atualmente, no departamento de produção de bicicletas da Decathlon Portugal, onde é realizado o projeto, existem vários fatores determinantes e em constante evolução com vista à excelência. Um fator de extrema importância é a sustentabilidade. Sustentabilidade é geralmente definida como a utilização de recursos de forma a satisfazer as necessidades do presente sem comprometer a capacidade das gerações futuras satisfazerem as suas próprias necessidades (Ahi and Searcy 2013). Contudo, a adoção de medidas sustentáveis ao longo do processo de industrialização, provocou um desequilíbrio na gestão de *stocks*, especialmente, em casos em que os recursos a soluções de destruição de componentes eram vastamente implementados.

A gestão de *stocks* remete, de uma forma simplificada, ao controlo do nível de negócios ou fluxo de bens e serviços de acordo com sua procura (Chalotra 2013). De acordo com Aro-Gordon & Gupte (2016), a gestão de *stocks* pode também ser definida como a supervisão da oferta, do armazenamento e da acessibilidade. Uma vertente importante para o sucesso das operações, durante a gestão da cadeia de abastecimento, é a gestão de *stocks* dos artigos, a fim de assegurar um abastecimento adequado, sem excesso de oferta.

A necessidade de reduzir os *stocks*, sem prejudicar o nível de serviço é um dos maiores desafios das organizações. Do ponto de vista financeiro, a manutenção de elevados níveis de *stock* tem um impacto negativo na performance de uma organização.

1.2 Motivação

Com a emergente preocupação relativamente à sustentabilidade ambiental, muitas são as organizações que começaram a adotar novos mecanismos e/ ou a alterar processos de modo a incorrer em processos de menor impacto ambiental. Assim, o estudo enquadra-se no centro de Produção da Decathlon Portugal, organização que adotou políticas e processos sustentáveis de modo a obter uma industrialização mais sustentável.

A organização necessitou de interromper alguns processos, especialmente, o processo de liquidação de stock excedentário de componentes por incineração, processo comprovado como poluente para o meio ambiente. Aliado à necessidade da organização na obtenção de novas soluções para a otimização de processos operacionais com base em grandes quantidades de dados para o apoio à decisão, culmina no desenvolvimento da presente dissertação aplicada ao mundo da produção de bicicletas.

1.3 Objetivos

O principal objetivo recai sobre a implementação de modelos com capacidade de classificar os diferentes componentes presentes na montagem de uma bicicleta e, através dessa informação desenvolver uma ferramenta que a partir das semelhanças entre componentes seja capaz de fornecer apoio ao agente de decisão, de modo a obter informação das quantidades que é possível industrializar com vista a liquidação do *stock* excedentário, assim como à lista de componentes e respetivas quantidades necessárias para montar o produto final.

Assim, pretende-se implementar esta ferramenta de modo a escoar os materiais imobilizados em armazém com o intuito de retornar o capital investido, e consequentemente, diminuir os custos associados à manutenção de *stock*.

As etapas propostas no decorrer do projeto implementado são:

- Levantamento de dados, como o histórico de modelos de bicicleta (produto acabado), características de cada tipo de componente, condicionantes do processo de industrialização, o *stock* existente de componentes, entre outros;
- Análise exploratória e tratamento de dados;
- Implementação e avaliação de modelos de classificação para os componentes;
- Construção de algoritmo que, através dos componentes inutilizados previamente classificados, retorna a listagem de componentes que são passíveis de constituir o produto final consoante o tipo de utilização e dimensões pretendidas para a bicicleta;
- Desenvolvimento de uma ferramenta para o apoio ao agente de decisão, com informação sobre a capacidade de produção para cada tipo de produto acabado, assim como o impacto no *stock* existente.

A ferramenta de apoio à decisão tem como objetivo fornecer informação sobre o modelo de bicicleta desenvolvido, associado a uma análise da capacidade de produção, o custo operacional, o impacto no gasto de *stock* excedentário, assim como, o retorno especulado.

1.4 Abordagem Metodológica

A abordagem seguida neste projeto remete a um caso de estudo, onde a pesquisa e análise são realizadas em contexto real, neste caso uma organização. Contudo, no a estrutura de implementação do

estudo segue uma das metodologias mais utilizadas no suporte a projetos de *data mining* (Mariscal, Marbán, and Fernández 2010), denominada *Cross Industry Standard Process for Data Mining* (CRISP-DM). A metodologia CRISP-DM fornece um processo genérico de ações, que consoante as necessidades de uma organização podem ser mais específicas e estruturadas de modo a alcançar os objetivos propostos (Shearer 2000).

Por esta metodologia, é possível dividir o ciclo de vida de um projeto de *Data Mining*, em seis fases (Figura 1).

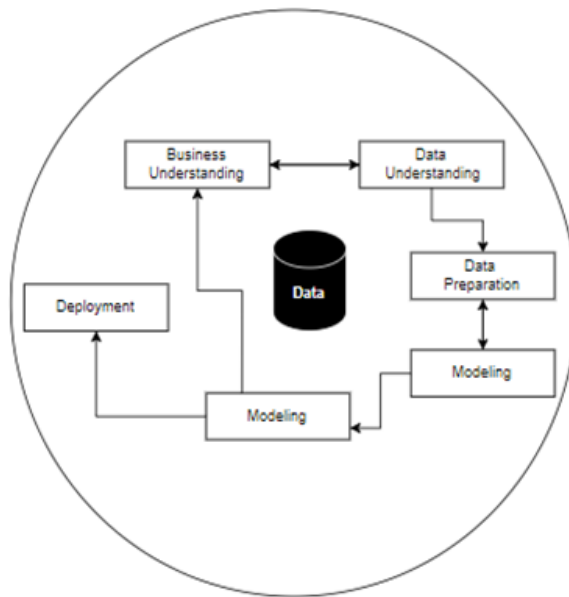


Figura 1 - Processo de CRISP-DM de acordo com (Chapman et al. 2000)

As setas (Figura 1) indicam as dependências mais frequentes entre as fases, enquanto o círculo externo simboliza a natureza cíclica das etapas de *data mining* e também, ilustra as informações adquiridas durante o processo podem desencadear novas necessidades, muitas vezes mais específicas (Shearer 2000).

Assim, as fases podem ser descritas como:

- **Compreensão do negócio:** Esta etapa inicial concentra-se na identificação dos objetivos e especificações do projeto e, a partir destas informações, definir um plano de ação para atingir os objetivos estabelecidos;
- **Compreensão dos dados:** Aquisição de dados e respetiva análise exploratória de modo a adquirir sensibilidade aos dados e compreender o seu comportamento;

- **Preparação de dados:** Processo de tratamento de dados, isto é, processos para identificar e corrigir eventuais problemas ou falhas presentes nos dados. Assim como, efetuar processos de transformação de modo a utilizar os dados no formato pretendido;
- **Modelação:** Diferentes técnicas de modelação podem ser selecionadas e implementadas nesta etapa. Além disso, esta fase permite o desenvolvimento de modelos, assim como, o ajuste de parâmetros do mesmo. Algumas técnicas têm requisitos específicos consoante o tipo de dados em estudo;
- **Avaliação:** Após a seleção de um ou mais modelos, este ponto do projeto pretende analisar a eficácia do modelo para garantir que este está a atingir assertivamente os objetivos estabelecidos;
- **Desenvolvimento:** Organizar o conhecimento extraído, bem como apresentá-lo de uma forma concisa e clara. Assim, através do conhecimento e informações adquiridas, é possível compreender rapidamente que ações necessitam ser colocadas em prática.

1.5 Estrutura do Documento

Este documento encontra-se estruturado em seis capítulos. O primeiro capítulo remete ao enquadramento geral e exposição dos objetivos e motivação da escrita do presente documento. O capítulo dois abrange a revisão bibliográfica dos conceitos abordados ao longo do estudo. O capítulo três ilustra o ambiente organizacional e as respetivas necessidades presentes no caso de estudo. O capítulo quatro remete à exposição das metodologias e modelos implementados. O capítulo cinco é referente à análise dos resultados obtidos, assim como a discussão dos mesmos. Este documento termina no capítulo seis, com uma síntese das conclusões retiradas, assim como, sugestões de trabalho futuro.

2. ESTADO DE ARTE

O presente capítulo remete à exposição de conceitos essenciais, de modo a fornecer uma contextualização teórica sobre os métodos e modelos aplicados no desenvolvimento do projeto. Deste modo, o estudo realizado foi corroborado através de uma investigação em artigos e livros anteriormente publicados.

2.1 Gestão de inventário

Organizações industriais requerem a necessidade de manutenção de *stocks*, por diversas razões, quer seja por satisfazerem atempadamente a procura, quer seja para evitarem ruturas no processo produtivo, ou até mesmo para beneficiarem de preços mais baixos ao encomendar em grandes quantidades ou por terem matérias-primas com preços muito oscilantes (Lisboa, João Veríssimo and Gomes 2006).

Os altos custos operacionais proporcionam uma forte motivação para que as organizações tenham um controlo mais exigente sobre o seu stock. A gestão de inventários ou *stocks* tornou-se, deste modo, uma parte crucial da gestão da cadeia de abastecimento (Aro-Gordon and Gupte 2016).

Esta gestão pode ter efeitos devastadores no desempenho da organização sobre o sistema de gestão de materiais implementado para determinar o desempenho do referido material e o desempenho geral da organização (Atnafu and Balda 2018).

Segundo Akindipe (Akindipe 2014), a gestão de *stocks* é recorrentemente considerada como uma oportunidade para a otimização dos recursos e para o alcance da eficiência operacional global entre indústrias.

2.1.1 Conceito de *Stock*

Para Chase et al. (Chase, R. B. & Aquilano 1995), os *stocks* são a existência de qualquer artigo ou recurso usado numa organização. Para Reis (Reis 2008) o *stock* pode ser definido como um conjunto de artigos que constitui uma determinada reserva de modo a satisfazer uma futura necessidade de consumo, fator útil para evitar situações de escassez.

Os *stocks* podem ser constituídos por diversos produtos, tais como (Zermati 2000):

- **Mercadoria:** produtos para revenda.
- **Matérias-primas:** Componentes ou artigos que constituem o produto final
- **Materiais auxiliares:** Materiais que se destinam à fabricação, mas não incorporam um processo de produção;
- **Produto acabado:** Produtos fabricados
- **Produtos de consumo:** produto adquirido para consumo interno da organização

Assim, também é possível identificar diferentes tipos de *stocks*, como (Reis 2008):

- **Stock:** Artigos de consumo regular;
- **Stock de segurança:** Parte do stock destinado a prevenir ruturas;
- **Stock alocado:** Com destino específico;
- **Stock global:** Soma das definições supracitadas;
- **Stock em trânsito:** com estadia no armazém por período limitado.

2.1.2 Custos Associados

A gestão de *stocks* e o *stock*, torna-se um mal necessário a todas as empresas, no entanto esta gestão acarreta custos inerentes à sua prática. Assim, para (Lisboa, João Veríssimo and Gomes 2006) estes custos são de várias naturezas:

Custos de armazenamento: consiste nos custos que uma empresa tem, em ter materiais não utilizados na produção no seu armazém.

Custos de oportunidade: que correspondem aos juros que uma empresa poderia usufruir se o valor das existências em armazém estivesse investido em fundos de rendimento.

Custo da matéria-prima: Consistem nos valores pagos pela empresa aos seus fornecedores. De forma a diminuir os custos relativos às compras, a empresa pode optar por comprar mais quantidades, e assim, beneficiar de descontos de quantidades muitas vezes propostos pelos fornecedores.

Custos de encomendar: Estes custos existem no momento de proceder a novas encomendas e são essencialmente de natureza administrativa.

Custos de ruturas de *stocks*: As ruturas de *stocks* podem acontecer por duas razões principais: uma procura superior à procura inicialmente prevista ou atrasos dos fornecedores nas entregas de materiais.

O gestor deve tomar as suas decisões considerando que a constituição de *stock* deve ser minimizada, para que o armazém não se verifique *stock* em excesso (sobredimensionamento de *stock*) mas também evitando o cenário de rutura de *stock* pois poderá colocar em causa a produção (Carvalho 2002).

2.1.3 Problemática de Manutenção de *Stock*

A existência de *stock* pode disfarçar ineficiências no seu processo de reabastecimento, uma vez que este desempenha um papel ambíguo na organização. Assim, por muitas razões, as empresas ficam perante cenários de excesso ou rutura de *stock* (Rosenfield 1989).

Ballou (Ballou 2004) refere que os gestores tendem a preferir *stocks* em excesso, como fator de segurança, em detrimento da falta dos mesmos e consequente insatisfação dos clientes. Porém, refere que os *stocks* absorvem capital que poderia ser investido em outras oportunidades de negócio, representando assim um custo de oportunidade. Além disso, os *stocks* podem encobrir problemas na gestão dos mesmos e, promover desintegração da restante cadeia de abastecimento. Carvalho et al. (2012) salienta que uma cadeia de abastecimento desalinhada, sem sincronização e colaboração enfrenta problemas de ineficiência, operações redundantes e sem valor acrescentado provocando o aumento de *stocks* nas fronteiras e o aumento de *stocks* de segurança.

O valor investido em *stocks* constitui um ativo para as empresas e, portanto, fazem parte do balanço de uma empresa. Além disso, os *stocks* influenciam os parâmetros financeiros das empresas e, a taxa média na qual os *stocks* são convertidos em dinheiro é um fator crucial na avaliação da condição financeira da empresa (Aro-gordon 2016). Os *stocks* provocam assim, um impacto significativo nos indicadores financeiros, sendo que o mau controlo de *stocks* pode congelar um elevado montante de capital e reduzindo a rentabilidade do investimento.

2.2 *Data Mining*

A necessidade recorrente de pesquisa em bases de dados por parte de organizações deu origem a novas abordagens para armazenar e manipular dados como o objetivo de obter informações fiáveis e atuais para dar suporte ao apoio à decisão. Uma das abordagens, apresenta-se sobre o conceito de *Data Mining* (DM) que consiste num processo de extração de informações e deteção de padrões em grandes quantidades de dados (Bharati and Ramageri 2010).

Assim, DM é um processo lógico utilizado para pesquisar em bases de dados de grande dimensão com o principal objetivo de encontrar padrões até então desconhecidos. A descoberta destes padrões permite

tomar decisões assertivas para o desenvolvimento de negócios. De forma geral, tal como mostrado na Figura 2, o seu processo resume-se a três etapas (Bharati and Ramageri 2010):

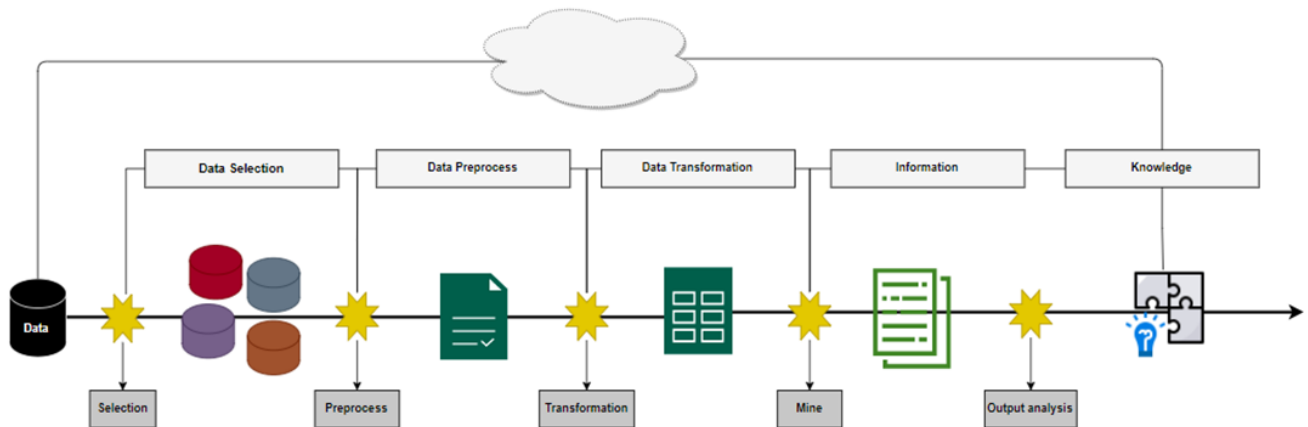


Figura 2 - Processo de DM de acordo com (Cabena et al. 1998)

1. Exploration: Data Selection, Data Preprocess and Data transformation

A primeira etapa remete à exploração de dados, onde se procede ao tratamento e transformação dos dados, assim como à seleção dos atributos importantes a serem estudados.

2. Pattern identification: Information

Esta etapa corresponde à identificação de padrões ocultos nos dados, o que remete à aquisição de informação relevante.

3. Deployment: Knowledge

Após a aquisição de informação é necessário analisar a sua veracidade de modo a transformar a informação em conhecimento. Esta fase remete à apresentação de um *output*.

DM tem um amplo campo de aplicação, sendo aplicável em quase todos os setores onde se verifica a existência de dados. Por isso, é considerada uma das técnicas mais importantes em tratamento de dados e também um dos desenvolvimentos mais promissores em Tecnologia da Informação (Ramos et al. 2016).

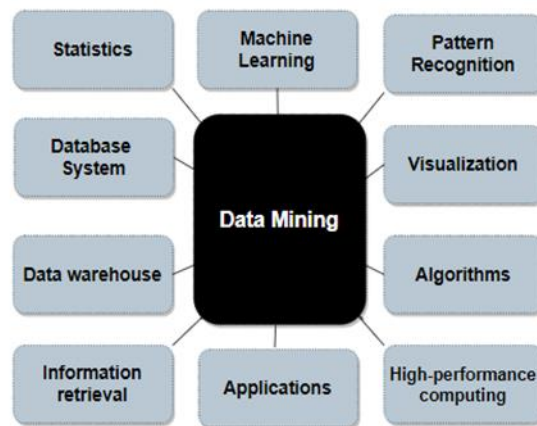


Figura 3 – Relação DM com outras áreas de acordo com (Han et al. 2012)

A Figura 3, mostra a relação de DM com outras áreas. Um bom exemplo, da interligação de DM com outros domínios do conhecimento, é o *machine learning* (ML). Embora tenham objetivos diferentes, DM utiliza métodos de ML e ML utiliza técnicas de DM como pré-processamento de dados para obter melhor precisão durante o processo de aprendizagem (Wang 2017).

2.3 *Machine Learning*

O ser humano possui a capacidade de reter informação através da experiência adquirida anteriormente e adaptar automaticamente o seu comportamento, aprendendo com o histórico (Carneiro et al. 2018). O mesmo não é possível afirmar sobre os sistemas computacionais, especialmente quando se analisa programas de computador considerados “tradicionais”, em que não é possível reanalisar os próprios resultados, portanto, não é possível adquirir conhecimento através dos acontecimentos passados.

Na sua essência, ML corresponde ao desenvolvimento de modelos e ferramentas com capacidade de aprender por meio da aquisição contínua de dados, resultando em experiência e conhecimento (Analide et al. 2006).

Por ser um campo da ciência da computação, mais especificamente um campo da Inteligência Artificial, o ML compartilha conhecimentos com diferentes áreas como: matemática, estatística, otimização, entre outros. Esta não pretende imitar o comportamento humano, mas sim complementar a inteligência humana com a capacidade de processamento dos computadores, permitindo o desenvolvimento de tarefas que vão além das capacidades humanas (Carneiro et al. 2015). Por exemplo, a capacidade de examinar e processar grandes quantidades de dados, uma vez que os modelos de ML podem detetar padrões inatingíveis à capacidade humana (Mitchell 1997).

Assim, é possível resumir ML à capacidade de um sistema de adquirir e integrar conhecimento por meio de observações em grande escala e de se aprimorar por meio da aprendizagem de novos conhecimentos, em vez de ser programado com esse conhecimento *à priori*.

2.3.1 Tipos de Aprendizagem

A aprendizagem é o processo de converter experiências em conhecimento (Bekkerman, Bilenko, and Langford 2011). Existem diversos paradigmas da aprendizagem em ML. Neste estudo são abordadas quatro principais tipologias, com foco na tipologia implementada, a aprendizagem supervisionada.

1. Aprendizagem Supervisionada

Na aprendizagem supervisionada, os dados utilizados pelo modelo do treino contêm informações sobre o resultado. Este tipo de aprendizagem utiliza exemplos já classificados, utilizando dados de entrada com *outputs* conhecidos de modo a treinar os modelos. Assim, esta técnica recorre a dados históricos com o objetivo de extrapolar futuros resultados, sendo possível analisar os resultados através da comparação com o resultado esperado (Qiu et al. 2016).

O fluxo de processamento desta técnica é apresentado na Figura 4.

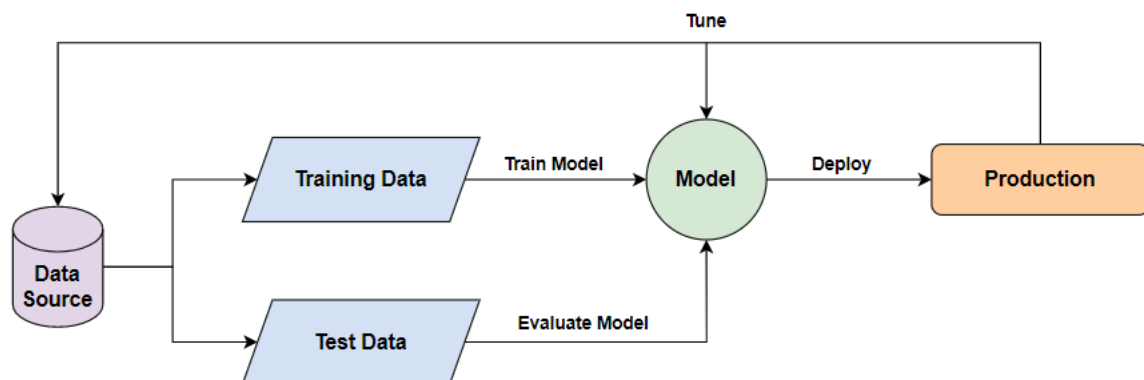


Figura 4 – Processo de Aprendizagem Supervisionada de acordo com (Mahesh 2019)

Existem muitos exemplos de modelos que utilizam aprendizagem supervisionada, tais como, *Decision Trees*, *Random Forest*, *Support Vector Machine*, *Naive Bayes Classifier*, *Bayesian Networks*, *Neural Networks*, entre outros (Qiu et al. 2016).

Uma das tarefas mais comuns da aprendizagem supervisionada, é o problema de classificação de instâncias desconhecidas em categorias previamente atribuídas. A classificação de um objeto é baseada em encontrar semelhanças com objetos predeterminados que são membros de classes diferentes, com

a semelhança entre os dois objetos é determinada pela análise das suas características. Em geral, a classificação em classes não é um valor numérico, logo significa que o atributo de classe, cujo valor deve ser determinado, é um atributo categórico (Novaković et al. 2017).

Existem dois tipos de classificação, classificação binária e classificação multiclasse. Se um problema de classificação remete somente a duas classes, é considerado classificação binária, se envolver duas ou mais classes, é um caso de classificação multiclasse.

Este estudo aborda um problema de classificação, assim, os métodos implementados são modelos de classificação multiclasse de aprendizagem supervisionada.

2. Aprendizagem Não Supervisionada

Os modelos baseados em aprendizagem não supervisionada, utilizam dados não classificados previamente, para o processo de treino do modelo. O sistema não sabe a resposta, apenas descobre os padrões dos dados. Neste tipo de modelo, estão disponíveis os dados de entrada, mas os resultados são desconhecidos antes dos modelos serem implementados e executados (Qiu et al. 2016).

Os algoritmos desenvolvidos nesta área de aprendizagem, identifica *clusters* ou grupos com base na semelhança entre os valores em avaliação (Rebala G., Ravi A. 2019). O processo de aprendizagem não supervisionada é representado na Figura 5.

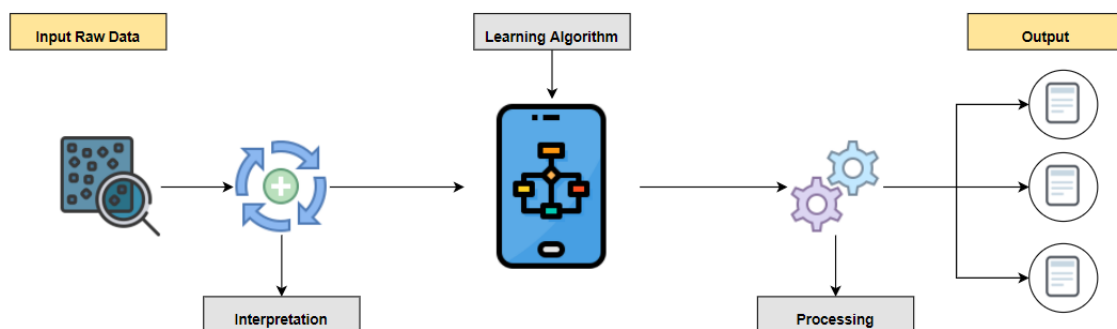


Figura 5 – Processo de aprendizagem não supervisionada de acordo com (Mahesh 2019)

Dentro da análise de clusters, os algoritmos de *K-means* e *K-Nearest neighbors* são exemplos de modelos de aprendizagem não supervisionados mais utilizados.

3. Aprendizagem Semi-supervisionada

Aprendizagem semi-supervisionada, como o nome indica, é uma combinação dos métodos de aprendizagem supervisionada e não supervisionada (Mahesh 2019). Neste caso, o conjunto de dados já apresenta alguns dos seus pontos classificados. O algoritmo utiliza técnicas de clusters, como na aprendizagem não supervisionada para identificar grupos dentro de um determinado conjunto de dados, e utiliza os pontos com dados classificados para classificar os restantes pontos de dados dentro do mesmo grupo ou *cluster*.

4. Aprendizagem de Reforço

A aprendizagem por reforço é o processo de treino de modelos de ML por meio de sequência de decisões. Nesta técnica, os resultados são encontrados por tentativa e erro, ou seja, é necessário definir os objetivos a alcançar, e perante as ações definidas pelo agente este recebe recompensas ou penalidades pelas ações que realiza. Deste modo, o objetivo é maximizar a recompensa total (Song et al. 2021). Este processo aparece representado na Figura 6.

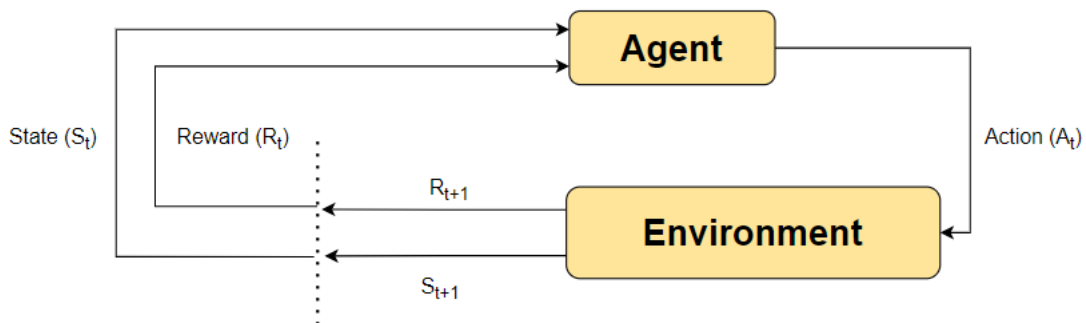


Figura 6 – Processo de aprendizagem de Reforço de acordo com (Mahesh 2019)

2.3.2 Modelos de Classificação

Perante os vários modelos de ML, serão abordados três modelos de classificação. Os modelos abordados correspondem a diferentes níveis de complexidade, correspondendo à ordem à qual foram implementados.

A. Decision Tree

As Árvores de Decisão ou Decision Tree (DT) são geradas a partir de um conjunto de dados de treino, o que resulta no desenvolvimento de uma estrutura em forma de árvore utilizada para classificar novos casos. Cada caso é descrito como um conjunto de recursos ou atributos associados com cada caso dos dados de treino. O *output* é o nome da classe. (Quinlan 1992).

Os algoritmos mais populares para a construção de Árvores de Decisão são o Dicotomizador Iterativo 3 (ID3), C4.5, e árvore de classificação e regressão (CART).

O ID3 foi desenvolvido por J. Ross Quinlan, durante os anos 70, que criou uma árvore de estrutura simplificada da raiz às folhas (construção de cima para baixo). Apesar disso, este algoritmo apresentou uma limitação clara, só foi capaz de lidar com variáveis nominais e, conseqüentemente, ser utilizado em problemas de classificação. Para resolver este problema, Quinlan criou um novo algoritmo, o C4.5, considerado uma melhoria do ID3. Este, permite variáveis numéricas contínuas e permite a implementação para problemas de regressão, além dos problemas de classificação. Outro algoritmo desenvolvido por um grupo de estatísticos (Myles et al. 2004), em paralelo com o mencionado antes, é o CART. Este é um algoritmo com uma abordagem muito semelhante a C4.5, também utilizado para modelos de classificação e regressão (Han, Kamber, and Pei 2012).

Além disso, outras melhorias surgiram, como a possibilidade de “podar” árvores. Este é considerado um acréscimo de extrema relevância, pois permite remoção de ramos que causam "ruído" nos dados e, conseqüentemente, uma melhoria na precisão. Normalmente, uma árvore de decisão adota a estrutura representada na Figura 7.

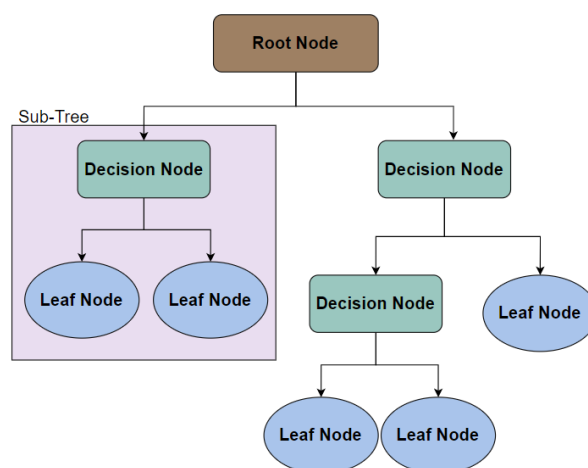


Figura 7 – Estrutura base de uma árvore de decisão de acordo com (Janikow 1998)

Como principais membros de uma árvore, surgem os conceitos:

- **Root Node:** representa uma variável e é o primeiro nó da árvore de decisão;
- **Branch:** representa o resultado de cada teste executado num nó. Os testes efetuados na árvore de decisão baseiam-se em condições *if-then* e o resultado de cada teste é utilizado para decidir qual o ramo a percorrer;
- **Decision Node:** Representa também uma variável e envolve todos os nós da árvore, exceto o nó raiz e os nós terminais;
- **Leaf Node:** Estes são os nós terminais da Árvore de Decisão. Quando um teste atinge um nó folha, uma classe é atribuída.

O processo de construção de uma árvore de decisão, segue etapas e parametrização de extrema importância para o modelo (Han et al. 2012).

Como *input*, o modelo utiliza um conjunto de dados de treino com as classes correspondentes e o método de seleção de atributos selecionado, procedimento responsável por estabelecer o método de divisão de dados em classes individuais (Figura 8). Como *output*, o modelo desenha uma Árvore de Decisão, capaz de classificar cada caso específico.

O processo de construção de uma árvore segue algumas etapas (Han et al. 2012):

1. A árvore inicia com um único nó (*root node*);
2. O algoritmo define o critério de divisão de acordo com o método de seleção de atributo previamente escolhido. O critério de divisão define, a melhor maneira de dividir o conjunto de dados de treino;
3. Este nó é nomeado como critério de divisão. Cada ramo que cresce a partir deste nó é um dos resultados do critério de divisão. Por exemplo, para valores discretos, os resultados são todos os diferentes valores possíveis, representados por cada ramo que cresce a partir do nó.

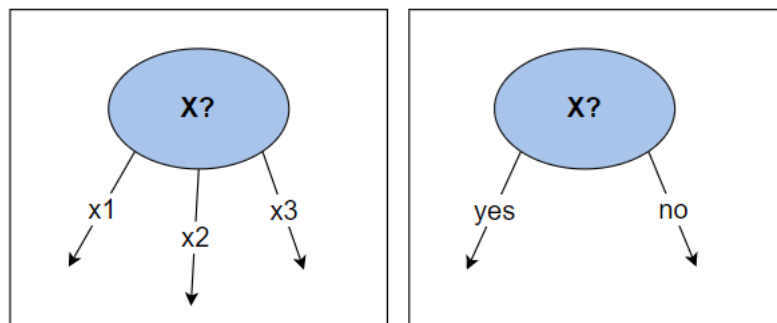


Figura 8 – Critério de divisão para classificação de acordo com (Han et al. 2012)

4. Este processo é interrompido quando para todos os casos nos dados de treino, estão alocados a uma classe ou não há mais possibilidade de dividir os dados, devolvendo assim uma folha (*leaf node*).

As Árvores de decisão são consideradas uma boa abordagem, pois permitem conhecer a importância e o impacto de cada variável no resultado. Ou seja, uma variável localizada mais próxima da raiz da árvore produz um efeito melhor no resultado do que uma variável mais próxima das folhas da árvore (Raglio et al. 2020).

Uma medida de seleção de atributos é uma heurística para determinar o critério de divisão que realiza a divisão dos dados em classes individuais. Assim, existem três medidas de seleção de atributos, descritas a seguir.

a. Information Gain

O *Information Gain* (IG) é uma medida de seleção de atributos baseada na entropia. A entropia é uma medida de incerteza que permite alocar cada atributo a cada posição da árvore. O cálculo da entropia é baseado na seguinte fórmula (Yin et al. 2019):

$$E = \sum_{i=1}^m -p_i \log_2 p_i \quad (1)$$

onde p_i representa a probabilidade de cada classe i aparecer no conjunto de dados para o total de m classes.

O valor de entropia oscila no intervalo de 0 a 1. Valores próximos de zero representam homogeneidade, ou seja, não se verifica a predominância de classes no conjunto de dados. O gráfico (Figura 9) demonstra como a entropia varia de acordo com a probabilidade (p) de cada classe i aparecer no conjunto de dados.

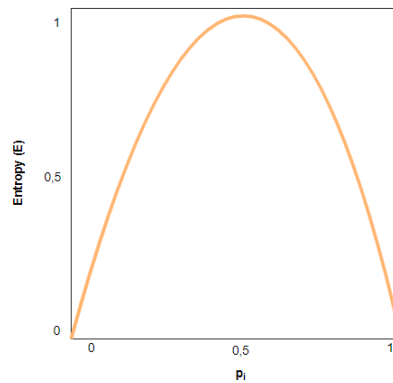


Figura 9 – Variação da Entropia perante o valor da probabilidade (Yin et al. 2019)

Assim, IG mede a redução esperada na entropia conforme o conjunto de dados é dividido. O atributo com maior redução de entropia e, conseqüentemente, maior IG deve ser alocado à raiz da árvore, o que permite reduzir a profundidade da árvore (Mingers 1989). O cálculo de IG é demonstrado pela expressão (Yin et al. 2019), onde G é o IG e E a entropia.

$$G(y, x) = E(y) - E(y, x) \quad (2)$$

onde $E(y)$ denota a entropia no nó de origem (ou seja, a distribuição da classe antes a divisão) e $E(y, x)$ a entropia para os nós criados (ou seja, depois da divisão da amostra pelo nó de decisão com base numa determinada característica, denotada como x).

b. Gain Ratio

Esta medida de seleção de atributos resolve algumas das limitações e especificidades do método *Information gain*, assim, é considerada uma extensão do mesmo. Deste modo, *Gain Ratio* é uma modificação do *Information gain* que reduz o viés (*bias*). Esta métrica utiliza o número de ramos que resultariam antes de aplicar a divisão, e assim, corrige a métrica mencionada através das informações intrínsecas a uma divisão. Finalmente, é possível calcular o *Gain Ratio* através da seguinte equação (Han et al. 2012):

$$Gain\ Ratio(X) = \frac{G(X)}{E(X)} \quad (3)$$

Após a seleção do critério de divisão, Quinlan propôs o seguinte procedimento (Quinlan 1992):

1. Determine o IG de todos os atributos e, em seguida, calcule o valor médio;

2. Em seguida, calcule a razão entre todos os atributos cujo IG é maior ou igual ao valor médio e, em seguida, selecione o atributo de maior relação de ganho para a divisão.

c. Índice de Gini

Esta medida é aplicada no algoritmo CART e é utilizada para medir a impureza de um conjunto de dados ou subconjunto.

O Índice de *Gini* é calculado como mostra a equação (Han et al. 2012):

$$Gini(F) = 1 - \sum_{i=1}^m p_i^2 \quad (4)$$

onde p_i representa a probabilidade de um registro pertencer a uma classe sobre um total de m classes. O atributo que maximiza o valor mínimo do índice de Gini é selecionado como o atributo de divisão (Han et al. 2012).

Outro conceito importante sobre Árvores de Decisão é *pruning*, ou seja, “podar” a árvore, este é o processo de exclusão de nós internos de uma árvore de decisão (Ayyadevara 2018). Métodos de *tree pruning* têm sido desenvolvidos para permitir que um modelo de árvore de decisão pare até que não seja mais possível dividi-lo (Mingers 1989). Este método, é utilizado, por exemplo em casos de *overfitting*, isto é, quando os dados se ajustam tão bem ao conjunto de dados de treino, que quando o modelo é confrontado com uma nova entrada (diferente do conjunto de dados de treino), o modelo acaba por não se adaptar (Shalev-Shwartz and Ben-David 2013). Assim, *tree pruning* permite eliminar redundâncias do modelo.

Por último, modelos de Árvores de Decisão têm algumas vantagens, mas também algumas limitações. Quando comparado com outros modelos, árvores de decisão são considerados modelos simples e eficazes, uma vez que alcançam boas previsões com esforços computacionais reduzidos (Kashyap 2018).

B. Random Forest

O modelo *Random Forest* possui árvores de decisão na sua constituição e é considerado um modelo eficaz, devido à apresentação de bons resultados e previsões de alta precisão (Kumar 2017).

No modelo de *Random Forest*, cada árvore de decisão é construída utilizando um subconjunto de dados escolhidos aleatoriamente a partir dos dados de treino. Cada subconjunto cria uma árvore, e assim, ao testar o modelo com um novo *input*, cada árvore irá classificar com uma respectiva classe. O *output* final corresponde à classe que aparece mais vezes entre as classes resultantes das árvores individuais (Pal 2005). O processo de desenvolvimento de *Random Forest* é apresentado na Figura 10.

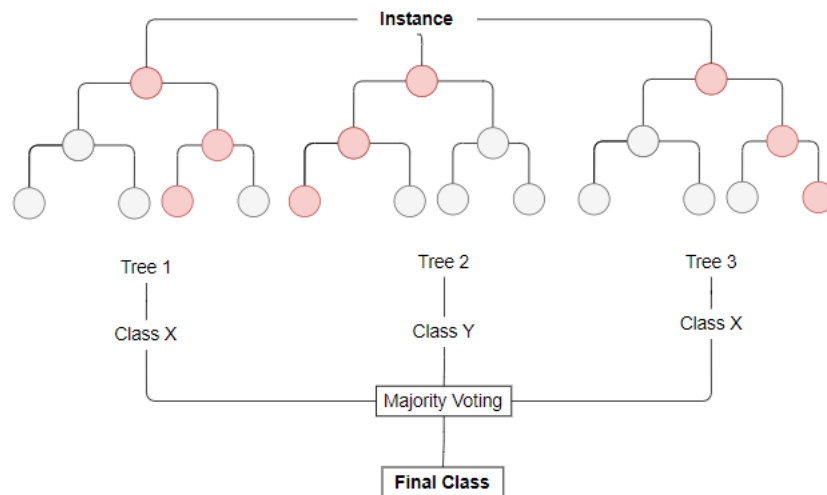


Figura 10 – Processo de desenvolvimento de *Random Forest* (Pal 2005)

Uma das principais vantagens, resulta da facilidade no tratamento de *outliers*. Além disso, assim como nas árvores de decisão, este modelo permite identificar as variáveis mais importantes para o estudo. Embora este modelo geralmente obtenha melhores resultados quando em comparação com árvores de decisão, requer maiores esforços computacionais (Kumar 2017).

C. Artificial Neural Network

Artificial Neural Network (ANN) ou Redes Neurais Artificiais apresentam um maior grau de complexidade, quando em comparação com os modelos mencionados anteriormente. ANN é um modelo em que a sua estrutura se baseia no funcionamento do cérebro humano. A unidade básica de uma ANN é o neurónio artificial, que representa o neurónio humano. Da mesma forma que o cérebro humano, os neurónios estão conectados entre eles, criando uma rede (Fausett 1994).

Em modelos ANN, os neurónios são responsáveis por processar informações. Assim, os sinais emitidos por cada neurónio são transmitidos para os próximos neurónios, através um elo de comunicação. Estas ligações têm um peso, que é responsável por multiplicar os sinais transmitidos. Para criar um *output*, cada neurónio aplica uma função de ativação para cada *input* que o atinge. Assim, o processo de treino

de ANN consiste na atualização dos pesos sobre iterações até produzir resultados. Estas iterações são chamadas de *epochs* e são um importante parâmetro em modelos de ANN (Heaton 2012).

A primeira pesquisa sobre redes neurais foi desenvolvida por Pitts e McCulloch, em 1943. Uma rede neural com mais de duas camadas é considerada como *deep learning* (Figura 11).

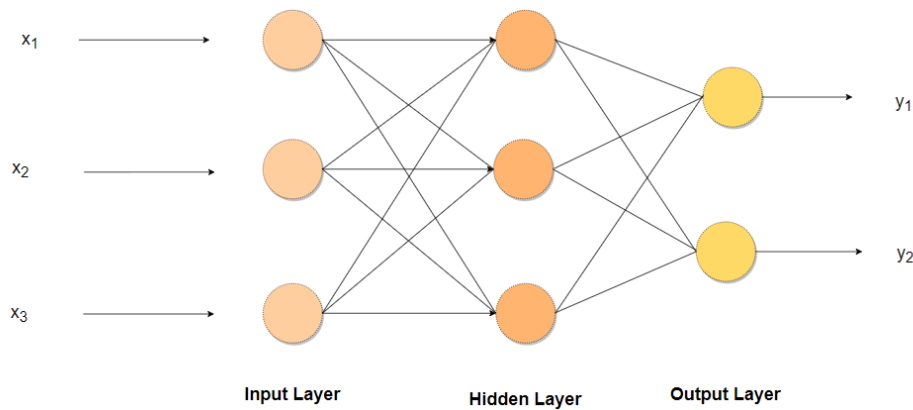


Figura 11 – Estrutura de uma ANN de acordo com (Gao & Su, 2020)

Os dois processos principais em ANN são o processo de *feedforward* e de *backpropagation*. O processo de *feedforward* envolve o fluxo de informações da camada de entrada de dados (*input layer*) até à camada de saída de conhecimento (*output layer*), enquanto o processo de *backpropagation* ocorre de forma contrária, ou seja, da camada de saída para a de entrada. Este último processo é responsável pelo processo de aprendizagem, e simultaneamente corrige os pesos atribuídos às ligações da rede.

O processo de *feedforward* matematicamente é representado pela seguinte equação:

$$\begin{bmatrix} x_1 \\ \dots \\ x_n \end{bmatrix} \times \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1n} \\ w_{j1} & w_{j2} & \dots & w_{jn} \end{bmatrix} + \begin{bmatrix} b_1 \\ \dots \\ b_j \end{bmatrix} = \begin{bmatrix} x_1 \times w_{11} + \dots + x_n \times w_{j1} + b_1 \\ \dots \\ x_1 \times w_{1n} + \dots + x_n \times w_{jn} + b_j \end{bmatrix}$$

Simplificando, para cada neurónio j da camada intermédia (*hidden layer*) ocorre a seguinte atribuição,

$$O_j = \sum_{i=1}^n w_{ji}x_i + b_j, \quad \forall j \quad (5)$$

onde b representa a constante *Bias*, isto é, é uma constante que ajuda o modelo a se ajustar da melhor forma aos dados fornecidos. A letra w , na equação 5, representa os pesos das ligações (do nó de entrada para o nó da camada intermédia), e o x representa os dados de entrada. A constante funciona como um

acréscimo, permitindo que à saída (que vem da função de ativação) desloca valores para a esquerda ou direita, no eixo das abscissas (Heaton 2012).

A estrutura mais simples em ANN é *multilayer perceptron* (MLP), na qual cada neurónio numa única camada recebe dados (*input*) e a transforma em resultados (*output*), como mostra a Figura 12.

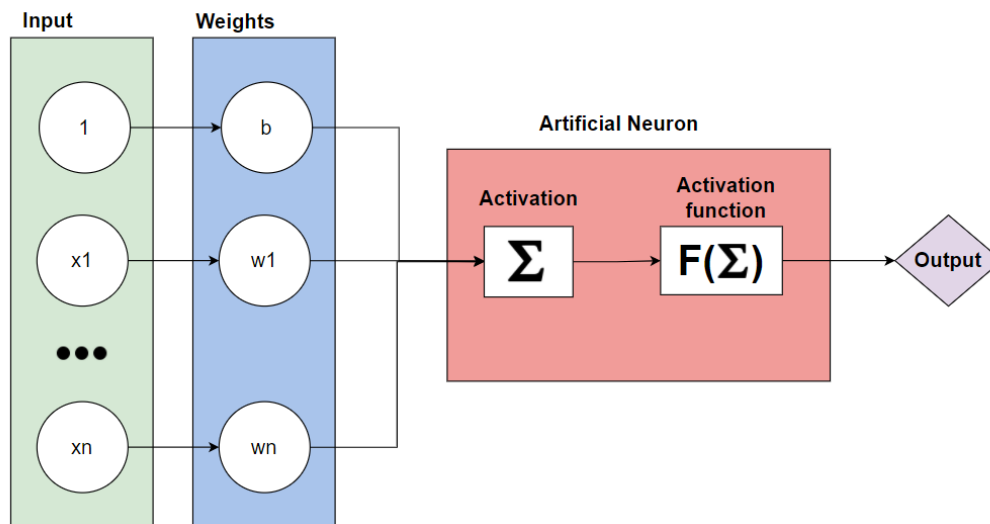


Figura 12 - Representação de uma operação de neurónio ANN (Vasilev et al., 2019)

Assim, de cada neurónio resulta uma saída, e esta, funciona como entrada para os neurónios da próxima camada. Assim, após o processo supracitado, para cada neurónio é aplicada uma função de ativação. Geralmente, as redes neuronais utilizam funções de ativação não lineares, que auxiliam na aprendizagem de dados complexos e a fornecer previsões precisas.

Existem muitas funções de ativação conhecidas, como a função de ativação linear retificada (ReLU), que é uma função não-linear que produzirá o *input* diretamente se for positiva, caso contrário, produzirá zero. Matematicamente, a função ReLU é mostrada e representada graficamente na Figura 13.



Figura 13 – Fórmula e Representação gráfica da função ReLU (Vasilev et al., 2019)

Esta é a função de ativação mais popular, e é implementada na camada intermédia e tem como vantagens a velocidade de construção dos modelos e o facto de não ocorrer erros no processo de *backpropagation*.

Outra função de interesse é a função *softmax*, que é utilizada em modelos multiclasse. Esta função calcula a probabilidade de distribuição de um vetor e retorna as probabilidades de cada classe, em que a classe alvo tem a maior probabilidade (Nwankpa et al. 2018). Assim, a função *softmax* retorna como *output* uma faixa de valores entre 0 e 1, em que a soma das probabilidades é igual a 1. Matematicamente, *softmax* é uma função onde z é vetor de entrada para camada de saída e j o índice das unidades de saída de dimensão K (Figura 14).

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad (7)$$

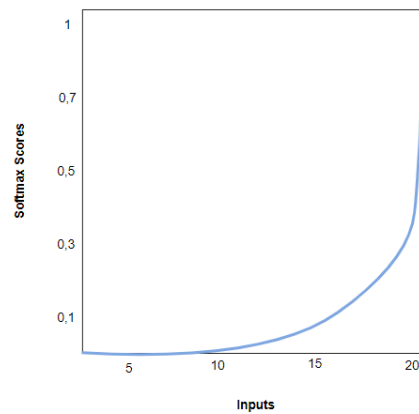


Figura 14 - Fórmula e Representação gráfica da função Softmax (Nwankpa et al., 2018)

A função *softmax* é frequentemente utilizada na camada final de um classificador em rede neural, ou seja, na camada responsável pelo *output*.

Resumindo, a função ReLU provoca o input, sendo aplicada na camada inicial e/ou intermédia e softmax provoca o output do neurónio, colocada na camada de *output* ou final.

A matriz de peso é o único parâmetro que varia durante o processo de treino em ANN. Como os pesos são corrigidos, a eficiência da rede neural melhora, convergindo para o menor erro possível. Esta atualização advém do processo de *backpropagation*, em que a informação flui no fluxo oposto ao processo de *feedforward*.

O processo de correção de pesos envolve as seguintes etapas (Heaton 2012):

a. Função de Perda

Para entender a *performance* de ANN, ou seja, perceber se os resultados estão próximos dos esperados, recorre-se à função de perda (*Co*).

A função de perda é utilizada para medir o desempenho do modelo e a inconsistência entre o valor real e o valor previsto. O desempenho do modelo aumenta com a diminuição do valor da função de perda (Murugan 2018).

Neste caso em específico, é mencionada a função *Cross Entropy* (*L*). Do ponto de vista teórico, *Cross Entropy* é utilizado para avaliar a similaridade entre duas funções. A métrica compara as duas distribuições ao longo de todo o domínio da variável e assume apenas valores positivos. De notar, que valores menores na função *Cross Entropy* denotam distribuições semelhantes (Grandini, Bagli, and Visani 2020).

Na classificação multiclasse, podemos considerar a variável de resposta e a previsão como duas variáveis discretas, uma vez que assumem valores reais e cada número representa uma classe diferente. Assim, a fórmula de *Cross Entropy* recorre a duas distribuições, *y*, a distribuição com valores reais, e $\hat{y}$ a distribuição estimada.

$$L = -y \log(\hat{y}) \quad (8)$$

Do ponto de vista prático, esta função é amplamente utilizada, embora, tenha apenas em consideração apenas a probabilidade verdadeira de classe sem se preocupar com a distribuição de massa de probabilidade entre as restantes classes (Grandini et al. 2020).

b. Gradient Descent

O *gradient descent*, é um algoritmo de otimização iterativo para encontrar o valor mínimo de uma função complexa através de iterações. A derivada é a taxa de variação da inclinação de uma função, que normalmente é representada pelas mudanças de uma função em qualquer ponto, desde que seja uma função contínua e diferenciável (Murugan 2018).

Como mencionado anteriormente, treinar ANN consiste na procura pelo melhor conjunto de pesos que produza *output* com o menor erro possível. Na impossibilidade de encontrar o melhor conjunto de pesos, através da análise de todas as combinações possíveis, uma vez que este processo é exaustivo e extremamente demorado (Heaton, 2012). Então, surge o processo de treino com recurso a iterações, ou seja, o único conjunto de pesos conhecido são os atuais de cada iteração.

Outro obstáculo ocorre da impossibilidade de encontrar uma curva de erro para todos os pesos, contudo é viável conhecer a inclinação da função de erro para um peso em específico, através do cálculo da derivada da função (Heaton, 2012).

É importante salientar que a função de perda é sensível a pequenas mudanças no peso, assim é necessário recorrer ao calculado das derivadas, logo, ao gradiente.

Este processo ocorre à regra da cadeia. A regra da cadeia foi desenvolvida para funções compostas, para casos em que uma função é utilizada como *input* de outra função.

Assim, este gradiente é calculado com base da implementação da regra da cadeia à função de perda, mencionada anteriormente.

c. Atualização de Pesos

O método de *backpropagation* é utilizado para treinar a rede neural e utilizar os valores do gradiente para atualizar os pesos. Logo, com a diminuição dos valores do gradiente, a disparidade do erro diminui e o erro da rede consequentemente também diminui (Heaton, 2012).

Nesse processo, existem dois conceitos cruciais a reter, nomeadamente, *batches* (lotes) e *epochs* (iterações). Cada lote é um conjunto de elementos de treino que são utilizados para treinar o modelo. Aqui, os pesos não são atualizados para cada elemento do conjunto, em vez disso, são totalizados os gradientes para cada elemento. Aquando da utilização de todos os elementos, os pesos da rede neural podem ser atualizados. Neste ponto, a iteração termina.

Aqui entra o conceito de *epochs*, esta agrega as iterações responsáveis pela atualização de pesos na rede neural (Heaton, 2012).

O objetivo principal é minimizar a função perda e, para isso, é necessário um método de atualização dos pesos das redes neurais, responsáveis por esta redução. Com a implementação da descida de gradiente, os pesos são atualizados através da seguinte equação:

$$\Delta W_{(t)} = \Delta W_{(t-1)} + \alpha \frac{\partial C_o}{\partial W} \quad (9)$$

onde α representa a taxa de aprendizagem. Esta taxa é responsável por dimensionar o gradiente e desacelerar ou acelerar o processo de aprendizagem (Heaton, 2012).

2.3.3 Análise de Desempenho

A tarefa de avaliar os modelos de classificação remete à comparação entre a classificação obtida pela utilização do modelo em relação à classificação real do caso. Dependendo do algoritmo selecionado, existem diferentes medidas para avaliar o desempenho do modelo. (Novaković et al. 2017).

A. Accuracy

Na avaliação de modelos de classificação, um conceito chave é a noção de falha. Se a implementação dos modelos de classificação tiver como resultado uma categorização diferente da classe real, então estamos perante um erro de classificação.

Assumindo a mesma importância para todas as classes, então, o número total de erros observados no conjunto pode ser um indicador de *performance* de um classificador.

Esta abordagem é baseada na *accuracy*, como uma medida para avaliar a qualidade da classificação modelo. Esta medida pode ser definida como a razão entre o número de casos classificados corretamente e o número total de registos classificados.

$$Accuracy = \frac{\text{number of correctly classified examples}}{\text{total number of cases}} \quad (10)$$

Por exemplo, se para uma amostra de 100 casos, um modelo conseguir classificar acertadamente 90 da totalidade dos casos então, a *accuracy* será de 90%.

As principais desvantagens desta métrica como medida de avaliação são (Novaković et al. 2017):

1. Negligencia diferenças entre tipos de erros, ou seja, não especifica o que causou o erro;
2. Depende da distribuição da classe no conjunto de dados.

B. Confusion Matrix

Nos casos em que é necessário distinguir os tipos de erros existentes, o resultado da classificação pode ser representado por uma matriz bidimensional, onde cada coluna da matriz corresponde a uma classe obtida pela aplicação do modelo e cada linha corresponde a uma classe teórica, como é apresentado na Figura 15 (Grandini et al. 2020).

		Predicted Class	
		Positive	Negative
Actual Class	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Figura 15 – Exemplo para classificador binário da construção da matriz de confusão

Na Figura 15, são indicadas as frequências obtidas como resultado da classificação,

TP: verdadeiro positivo (por exemplo, declarar X e realmente é X);

FN: falso negativo (por exemplo, declarar X quando, na verdade é Z);

FP: falso positivo (por exemplo, declarar Z quando, na verdade é X);

TN: verdadeiro negativo (por exemplo, declarar Z e realmente é Z);

Na diagonal da matriz está presente o número de exemplos classificados corretamente, enquanto os restantes elementos da matriz indicam o número de exemplos que foram classificados incorretamente. Através da estrutura da matriz e respetiva nomenclatura, é possível examinar novas métricas de desempenho.

Como, por exemplo, a *precision*, que responde à pergunta: “qual proporção de positivos previstos é verdadeiramente positiva?”. Este cálculo remete ao número de verdadeiros positivos sobre o número total de positivos.

$$Precision = \frac{TP}{TP + FP} \quad (11)$$

Assim, a precisão depende de verdadeiros positivos e falsos positivos. Existem duas formas diferentes de calcular esta métrica.

- **Macro averaged precision:** calcular precisão para todas as classes individualmente e, em seguida, calcular a média;
- **Micro averaged precision:** calcular verdadeiro positivo e falso positivo para uma classe e, em seguida, utilizar os valores para calcular a precisão geral.

As próximas métricas abordadas também são passíveis de determinar o seu cálculo através destas formas distintas.

Outra medida muito útil é a *recall*, que responde à pergunta, “que proporção de positivos reais está corretamente classificada?”. Neste caso, o cálculo avalia os verdadeiros positivos sobre o número real de positivos.

$$Recall = \frac{TP}{TP + FN} \quad (12)$$

Em situações de comparação entre modelos, em que os resultados obtidos são ambíguos entre as duas métricas mencionadas, é utilizada a métrica F1-score de modo a resumir o desempenho dos modelos numa única métrica através da *precision* e *recall*. A F1-score é calculada através da seguinte equação.

$$F1 - Score = \frac{2(Precision \times Recall)}{Precision + Recall} \quad (13)$$

C. Area under the ROC (Receiver Operating Characteristic) curve

Outra forma de avaliar o desempenho do classificador implementado é o recurso às curvas de ROC (Hanley and McNeil 1982). A curva de ROC é uma representação bidimensional da relação entre a taxa de falsos positivos e a taxa de verdadeiros positivos (Figura 16).

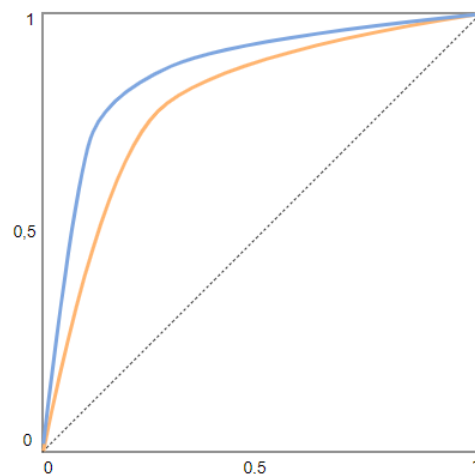


Figura 16 – Representação gráfica de uma curva de ROC (Hanley and McNeil 1982)

ROC é uma curva de probabilidade e a área abaixo da curva (AUC) representa o grau ou medida de separabilidade, ou seja, indica a capacidade do modelo de distinguir entre classes. Assim, quanto maior o AUC, melhor o modelo.

Para problemas de multiclasse, para calcular o valor de AUC é necessário dividir o conjunto de dados em vários problemas de classificação binária. Um classificador é então treinado em cada problema de classificação binária e são feitas previsões utilizando o modelo mais confiável. Esta técnica denomina-se um contra todos (*One vs All*) (Mahesh 2019).

2.3.4 Exemplos de aplicação de *Machine Learning*

As metodologias abordadas representam um grande potencial em diversas áreas de aplicação, deste modo verifica-se um aumento de artigos e de casos de estudo apresentados à comunidade científica.

Os casos de estudo são singulares, contudo é possível constatar fatores transversais aos dados e à área de aplicação que nos permite especular sobre o desfecho da implementação dos modelos de ML. Assim, recorre-se a casos de estudos com dados reais de modo a averiguar conclusões previamente constadas e comparar com os resultados deste projeto.

Devido à diversidade de modelos e técnicas de implementação de ML, um dos principais temas a abordar, remete à comparação entre modelos, especialmente, modelos em árvore e baseados em redes. Para Zhou (Zhou & Fujita, 2017), os modelos em árvore apresentam como principais vantagens, a fácil interpretação dos resultados e versatilidade, isto é, a capacidade de lidar com valores numéricos ou categóricos. Em contrapartida, estes modelos são mais suscetíveis a tendências e enviesamentos, e podem levar a cenários de *overfitting*, isto é generalização excessiva da informação.

Os modelos neuronais apresentam uma alta precisão e tolerância a ruído, assim como, um melhor balanceamento a falhas, visto que, quando um elemento da rede neural falha, não implica a falha do processamento inteiro devido à sua estrutura paralela. Contudo, requer um elevado tempo de processamento e de aprendizagem, e um nível de complexidade elevado na definição da regra de classificação.

Num estudo sobre a implementação de algoritmo de multiclasse, para avaliar o nível de atividade de pacientes que sofreram um AVC (Apao et al. 2020), recorre à construção de cada árvore de decisão de modo a criar uma floresta (*Random Forest*). A decisão final ou previsão é feita com base na maioria de votos de cada um dos nós da árvore de decisão. *Random Forest* utiliza a agregação *bootstrap* para reduzir o risco de *overfitting* e o tempo de treinamento, o que consequentemente melhora a performance em relação às árvores de decisão. Este classificador foi selecionado para o estudo porque vários estudos comprovaram que fornece a maior taxa de precisão em conjuntos de dados de saúde em comparação com os outros algoritmos de classificação.

Contudo, noutro estudo desenvolvido para prever o consumo de energia em edifícios (Ahmad, Mourshed, and Rezgui 2017), foram aplicadas outras técnicas de ML. Este estudo comparou o desempenho da rede neural artificial com *Random Forest* para avaliar a sua aplicabilidade na previsão do consumo de energia. Com base nas métricas de desempenho utilizadas no artigo, verifica-se que ANN teve um desempenho marginalmente melhor do que o algoritmo com base em árvores. Ambos os modelos tiveram um desempenho melhor em conjuntos de dados de treino e validação. ANN mostrou maior precisão em dados adquiridos recentemente (conjunto de dados de teste). No entanto, a partir dos resultados, conclui-se que ambos os modelos podem ser viáveis e eficazes para prever o consumo de eletricidade por hora. No que diz respeito, à classificação de componentes através de ML, grande parte dos estudos abordam os modelos de ML para um contexto diferente de dados, isto é, nesses casos a classificação ocorre através da leitura de informação através de imagens (Penumuru, Muthuswamy, and Karumbu 2020). Assim, a melhor técnica de aprendizagem em máquina não poderá ser escolhido *a priori*, e os diferentes algoritmos necessitam de uma avaliação com o intuito de validar a sua aplicabilidade para um determinado problema.

3. CASO DE ESTUDO

Neste capítulo é realizada uma breve descrição do grupo Decathlon, seguindo-se a apresentação da divisão Decathlon Produção Portugal. Por último, será apresentada a equipa responsável pela produção de bicicletas, onde se insere o presente projeto.

3.1 Decathlon

A Decathlon é uma empresa de artigos de desporto à escala mundial fundada em 1976 por Michel LecLercq em Englos na França. Desde a sua fundação cumpre a promessa de produtos desportivos inovadores e de excelente qualidade a preços baixos.

A Decathlon esforça-se para proporcionar tecnicidade e qualidade a preços competitivos, facto possível, uma vez que são criadores, produtores e distribuidores de produtos desportivos concebidos a partir de componentes exclusivos. Assim, a sua oferta é constituída maioritariamente por produtos de marcas exclusivas.

Até ao momento, dezembro de 2021, a empresa apresenta as seguintes características mostradas na Figura 17.

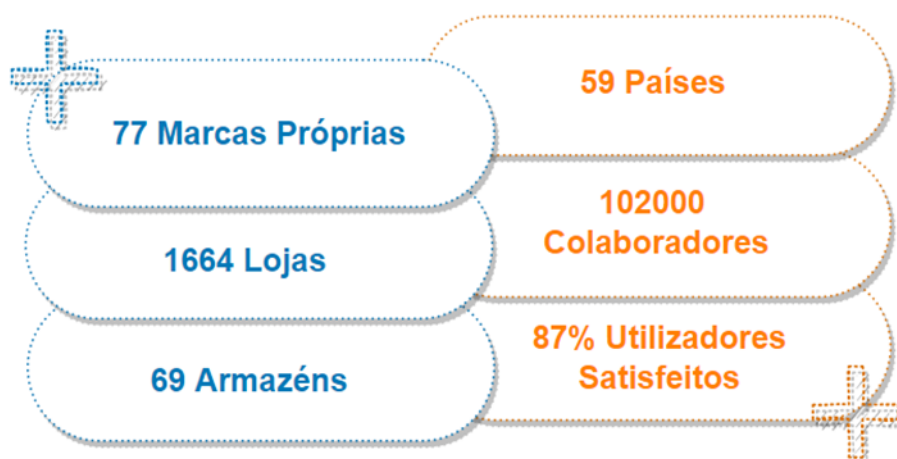


Figura 17 - Características da empresa à presente data (dezembro de 2021)

Em Portugal, inaugurou a primeira loja em 2000, a Decathlon Amadora, e atualmente tem 38 lojas dispersas de Norte a Sul do País e Ilhas. Também, possui um centro de logístico e um centro de produção.

A principal missão da Decathlon é ter clientes / desportistas satisfeitos. Uma empresa determinada em ouvir os desportistas de modo a entender as suas necessidades e oferecendo produtos atrativos aos preços mais baixos. As equipas de design assinam produtos adaptados aos utilizadores, para facilitar a sua prática, seja qual for a idade ou o nível de capacidade física, para que aproveitem os prazeres e os benefícios do Desporto.

A. Criar produtos seguros e de qualidade

Cada produto promete satisfazer os utilizadores ao longo da sua experiência. É este o objetivo essencial que a organização cultiva em todas as fases de vida dos seus produtos. Mobiliza a cadeia de negócios, desde o design do produto até o serviço pós-venda, para projetar e vender produtos que respondam aos mais exigentes critérios de qualidade.

As equipas trabalham para reduzir os riscos relacionados com a composição química, seja para nossos utilizadores, para os nossos vendedores, para os operadores que os fabricam ou para as pessoas que vivem perto dos locais de produção.

B. Compromisso ambiental

Há vários anos que a Decathlon reflete e aplica ações que contribuem na melhoria do seu impacto ambiental. Isso concretiza-se pela recolha de produtos usados, pelas rigorosas regras ambientais internas na produção dos seus artigos, pelas matérias-primas utilizadas, pela colocação de sistemas de iluminação de baixa tensão, pela redução do uso de embalagens, pela otimização dos transportes e pelo melhoramento das capacidades energéticas nas suas infraestruturas, mas também, pela mobilização e comprometimento de cada colaborador nos seus gestos diários.

Assim, o propósito evolui para esclarecer e aprofundar o significado da existência da Decathlon, passando do tradicional “Desporto acessível a todos” para “Para tornar o prazer e os benefícios do desporto acessíveis a todos de forma sustentável”.

3.2 Departamento de Produção Portugal

A Decathlon Produção estabeleceu-se inicialmente em Portugal, em 1993, na Maia. E tem como missão industrializar e produzir localmente para satisfazer as necessidades de artigos de desporto para o maior número de clientes.

A Decathlon apoia-se sobre uma rede de parceiros exclusivos e fornecedores para a produção dos seus produtos em 59 países. Produz cerca de 10 milhões de produtos desportivos por ano, entre eles bicicletas, capacetes, tabelas de basquetebol, calçado, entre outros artigos.

Atualmente, o DPP é o maior produtor de bicicletas em Portugal, com a produção de mais de 1.3 milhões de bicicletas para todo o mundo, representando mais de 30% das bicicletas que são vendidas mundialmente e mais de 40% das vendidas na Europa.

3.2.1 Produção de Bicicletas

Uma bicicleta é constituída por 65 componentes, os mais importantes remetem à sua estrutura, nomeadamente a *frame* e *fork* (Figura 18).



Figura 18 – Bicicleta e exemplo de uma *frame* e *fork*

Cada de modelo de bicicleta, enquadra-se num tipo de família de bicicletas consoante a semelhança entre as suas principais características, como por exemplo, bicicletas de estrada, de montanha ou cidade, assim como numa gama de preços que difere entre baixa, média e alta.

3.2.2 Processo de Industrialização

O processo de desenvolvimento de uma bicicleta envolve diversos setores e atividades. De modo a compreender na totalidade o processo de industrialização de uma bicicleta é necessário definir as etapas e a responsabilidade de cada setor (Figura 19).

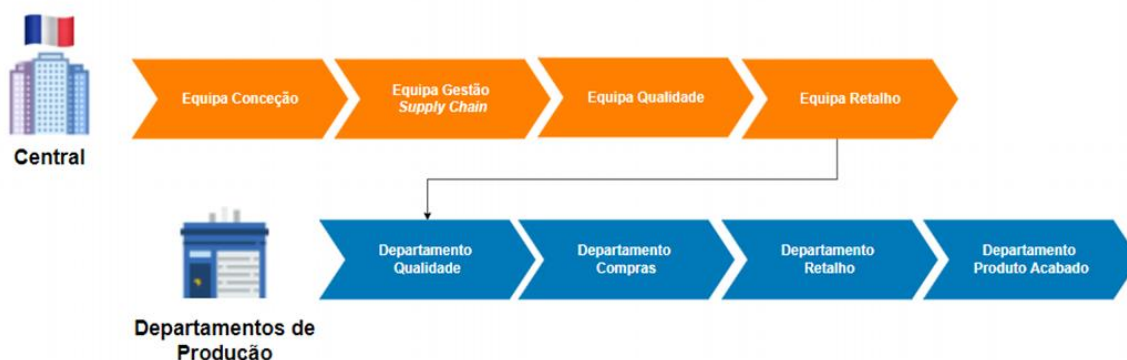


Figura 19 - Diagrama de setor envolvidos na produção de bicicletas

Na Decathlon, o processo de criação e processo inicial de desenvolvimento de um novo modelo de bicicleta é da responsabilidade do setor central, localizado em França. O processo envolve quatro equipas, com as seguintes responsabilidades:

- **Equipa de conceção:** Idealizam a bicicleta de acordo com as necessidades do mercado e enquadram a bicicleta numa família e numa gama.
- **Equipa de Gestão de *Supply Chain*:** avaliar fornecedores para cada componente e as opções de produtos que estes dispõem e selecionam o mais competitivo. Tendo em atenção a capacidade do fornecedor para abastecer os 47 Departamentos de Produção com as quantidades necessárias.
- **Equipa de Qualidade:** garantir que os *standards* de qualidade para os produtos são assegurados para todos os processos envolvidos no fabrico.
- **Equipa de Retalho:** Realizam análises de mercado, definem quantidades anuais de acordo com estudos de mercado e definem um preço *target*.

Após o processo descrito, a Central comunica com os seus departamentos de produção para averiguar a aplicabilidade do novo modelo e despoletar o processo de industrialização. Um dos principais relatórios a comunicar, de modo a uniformizar a comunicação, é a *bill of material* (BOM), que é uma lista dos

componentes necessários para construir um produto, ou seja, tem informação sobre os componentes a utilizar e as quantidades necessárias para a construção do produto acabado.

Os Departamentos de Produção (DP) enquadram-se em dois tipos distintos, *assembly* (montagem) e de componentes. Maioritariamente em países europeus, como Portugal, existem DP *assembly*, ou seja, são responsáveis pela montagem do produto acabado e os DP de componentes, responsáveis pelo fabrico de peças utilizadas no produto final, e estes encontram-se maioritariamente na Ásia. Assim, cabe ao DP a industrialização e fabrico dos produtos nos fornecedores subcontratados, sendo este um componente ou um produto acabado.

O processo de fabrico de uma bicicleta num DP *assembly* remete a quatro etapas principais divididas por quatro departamentos, responsáveis por:

- **Departamento Qualidade:** Industrializar o produto, isto é, realizar testes na linha de produção para garantir que o produto é possível de ser produzido com a qualidade necessária. A fase de testes inclui três etapas, protótipos (fora da linha de produção), pré-série (em linha com quantidades reduzidas) e *mass production* (produção);
- **Departamento de compra:** Realizam a compra de componentes que caracterizam o produto acabado aos respetivos fornecedores, garantindo as necessidades para manter a produção sem incorrer em imprevistos;
- **Departamento retalho:** Responsável por comprar os componentes necessários para responder às necessidades de produção estabelecidas por França.
- **Departamento Produto Acabado:** Gerir planos de produção, isto é, perante a informação de necessidades do mercado (estabelecida pela equipa central das vendas) e estabelecer um plano de ação perante as restrições existentes, como a disponibilidade de componentes ou a capacidade dos montadores.

3.2.3 Projeto *Zero Destruction*

O DP Portugal (*assembly*) apresenta-se responsável pela compra e gestão do seu *stock* que, em grande parte, advém de fornecedores asiáticos. O fator de impossibilidade de transporte entre a Europa e a Ásia, e o elevado tempo de viagem culminam na recorrência de optar por encomendas de quantidades superiores para manter o funcionamento das linhas de produção e evitar imprevistos. Antigamente, recorria-se à destruição de *stock* excedentário, ou seja, *stock* imobilizado, ou seja, que em nenhum

momento iria ser utilizado na produção, de modo a obter mais área de armazenagem assim como impedir custos de armazenagem extraordinários.

Contudo, com os novos conhecimentos sobre o impacto das indústrias nas alterações climáticas, assim com a organização empenhada em viver uma política mais sustentável surge um compromisso de terminar com a destruição de *stock* sem defeitos funcionais (defeitos visuais incluídos).

Com a adoção do compromisso ambiental e a eliminação do único mecanismo para liquidação de *stock* excedentário resultou na acumulação excessiva de material entre portas. Até à data de hoje, a manutenção de *stock* inutilizado advém em custos elevados de armazenagem, assim como o valor monetário investido nos componentes que se encontram inutilizados reflete num impacto negativo na análise financeira da empresa.

A questão atualmente recai em como escoar este *stock*, e retornar o valor investido. A solução idealizada para resolver este problema remete à reutilização de componentes classificados como *stock* excedentário, através da substituição de componentes em modelos existentes e através da criação de novos modelos localmente e remeter o valor investido através da venda dos modelos desenvolvidos através de plataformas existentes na Decathlon como a 2ª vida. Esta plataforma dispõe de artigos de exposição, devolvidos ou danificados previamente verificados por técnicos especializados para venda e assim, dar uma segunda vida a estes artigos.

Assim, os principais objetivos são a construção de uma base de dados única com a informação necessária, assim como detetar padrões e características numa grande quantidade enorme.

O projeto realizado descrito no presente do relatório, é referente a esta problemática e na necessidade de otimização deste processo, assim como fornecer suporte aos agentes de decisão, que neste caso são os responsáveis pela verificação da qualidade na linha de montagem.

4. MÉTODOS E MODELOS IMPLEMENTADOS

Neste capítulo são apresentados os métodos e modelos utilizados para atingir o reaproveitamento de *stock* excedentário. Como supracitado, a metodologia implementada recai sobre seis fases, agregadas em exploração e pré-processamento de dados, incluindo a análise exploratória, implementação e avaliação de modelos e representação de resultados.

4.1 Exploração e Pré-processamento de dados

A exploração de dados é uma fase crucial em projetos de *data mining*. Visualizar e entender o comportamento dos dados é fundamental para permitir a preparação dos mesmos, assim como, caso seja necessário corrigir falhas existentes.

Neste caso, antes do processamento de dados, ocorreu um levantamento dos dados disponíveis. Deste modo, os dados recolhidos perfazem 20 anos, desde o ano 2001 até 2021, e representam características relativas aos modelos de bicicletas, como o tamanho da bicicleta, o tipo de utilização pretendida para cada modelo e características sobre os componentes como, o custo do componente e a quantidade em *stock* classificado como excedentário.

É essencial realçar que, de modo a manter a confidencialidade da empresa, os valores e nomenclaturas descritos na análise são meramente exemplificativos, não remetendo à realidade empresarial.

A análise de dados e a exploração dos resultados foi realizada em linguagem de programação *python* (Lutz 2003). Além disso, esta linguagem foi utilizada para construir e treinar os modelos implementados e também desenvolver uma aplicação de suporte à tomada de decisão. Para ajustar e treinar esses modelos, recorreu-se ao ambiente de *Jupyter notebook* (Toomey 2017). As bibliotecas utilizadas mais importantes são: *pandas*, *matplotlib*, *NumPy*, *scikit-learn*, *TensorFlow*, *Keras* e *streamlit*.

4.1.1 Análise Exploratória

A base de dados utilizada neste estudo apresenta aproximadamente 10900 observações. Através do conhecimento de negócio, ocorre a necessidade de separar os modelos de criança (*kids*), dos modelos para adulto (*adult*) em duas análises distintas. Esta divisão ocorre uma vez que a estrutura base de uma bicicleta de criança, para a mesma família, diferem no número de componentes na sua estrutura.

Salientando, que para facilitar a compreensão e distinção entre as variáveis na fase de implementação, os nomes das variáveis são lúdicos, de forma a identificar rapidamente o seu significado. As variáveis descritivas e a sua respetiva explicação são apresentadas na Tabela 1.

Tabela 1 – Descrição de variáveis em estudo

Variável	Descrição	Tipo
Component	Nome do componente	Categórica
Type	Tipo de componente	Categórica
Bike	Nome do modelo de bicicleta	Categórica
Family	Tipo de bicicleta	Categórica
LABEL	Categorização	Categórica
QTY	Quantidade do componente utilizado na bicicleta	Númerica
€€	Custo unitário do componente	Númerica
Stock	Quantidade do componente em extra stock	Númerica

A. Bicicleta Adult

Com análises distintas, o cenário para bicicleta de adulto apresenta 10 mil registos e 20 classes previamente definidas, nomeadamente, *Elops L/XI, Elops One Size Fits All, Elops S/M, Riverside L, Riverside L/XI, Riverside M, Riverside M/XI, Riverside S, Riverside S/M, Road L, Road M, Road S, Road XI, Road Xs, Rockrider L, Rockrider M, Rockrider S, Rockrider Xs, Rockrider XI, Tilt One Size Fits All*.

Através da análise das medidas descritivas, é possível aprofundar a exploração ao conjunto de dados. Assim, a Tabela 2 mostra medidas estatísticas, relacionadas ao conjunto de dados como, mínimo, máximo, média, padrão desvio e variância, permitindo uma melhor compreensão da dispersão dos dados em relação ao custo unitário do componente e a quantidade em *stock*.

Tabela 2 – Medidas descritivas (adult)

Medidas Descritivas	Unit Cost	Stock
count	10000	10000
mean	22,65	144409,7
std	14,39	28792,9
min	0,1	1
25%	5	453
50%	8,8	2504
75%	13,4	17685
max	141,9	195534

Pela análise da Tabela 2, é possível compreender a disparidade presente na quantidade em *stock* para cada tipo de componente, uma vez que é possível encontrar componentes apenas com uma unidade em *stock* e outros com quantidades muito elevadas.

Na Figura 20 é ilustrada a proporção do *stock* e do custo em relação ao valor total, para cada tipo de família.

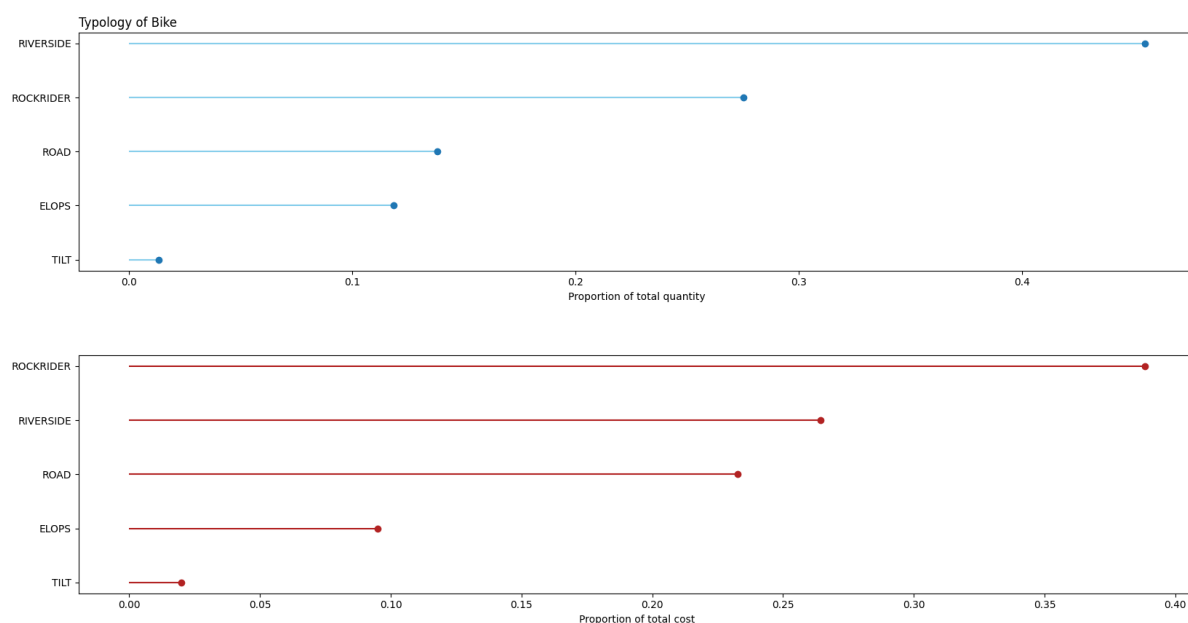


Figura 20 – Comportamento por família em relação ao custo e quantidade de stock (*adult*)

A Figura 20 permite averiguar que a família Riverside apresenta maior quantidade de componentes em *stock*, contudo em termos monetários a família Rockrider tem um maior impacto.

Na Figura 21, é representado a proporção do custo em relação ao valor total, para cada tipologia de componente. Analisa-se os dez componentes com custo mais elevado em comparação com o custo dos componentes estruturais, e ditos essenciais à construção de uma bicicleta (Tabela 3).

Tabela 3 – Componentes estruturais

Component
Frame
Fork
Derrailleur
Shifter
CHAIN
BOTTOM BRACKET
CRANKSET
CASSETTE
Free Wheel

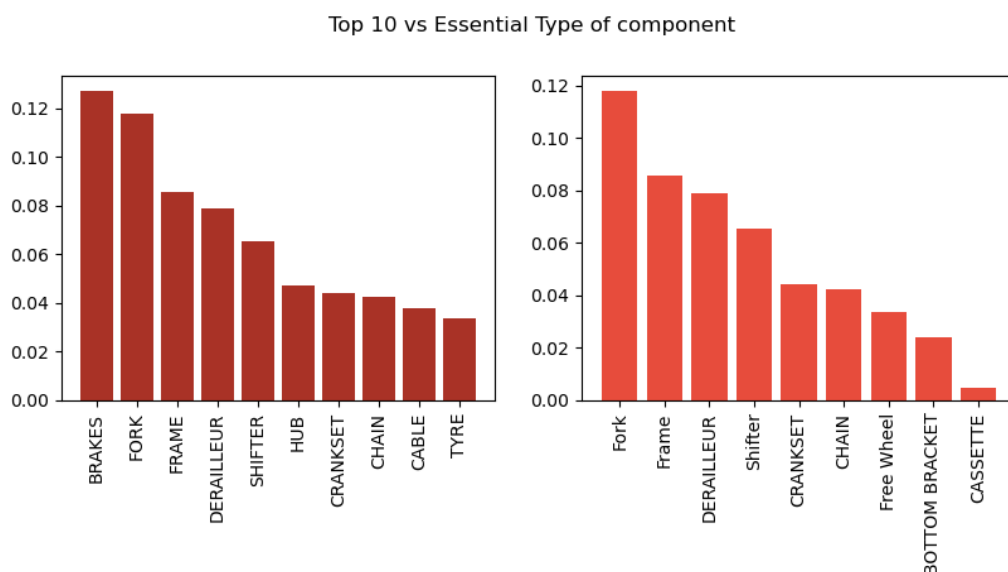


Figura 21 – Comportamento por tipologia de componentes para a quantidade de stock (*adult*)

Entre os tipos de componente com custo elevado, classificada como *stock* excedentário, 8 em 10 são componentes estruturais.

Na Figura 22, é representada a proporção do *stock* em relação ao valor total, para cada tipo de componente através da comparação do top 10 com os componentes estruturais.

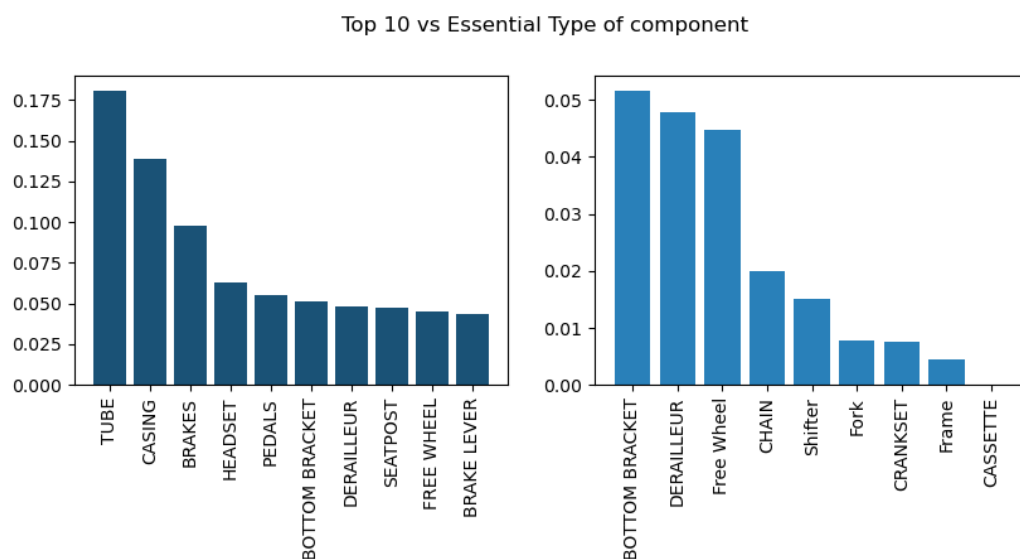


Figura 22 - Comportamento por tipologia de componentes para o custo (*adult*)

Entre os tipos de componente com quantidade elevada de *stock* excedentário, 3 em 10 são componentes estruturais.

B. Bicicleta Kids

O cenário de bicicleta para crianças apresenta 900 registos e 9 classes previamente definidas nomeadamente *Kids 14"*, *Kids 16"*, *Kids 20"*, *Kids 24"*, *Kids MTB 20"*, *Kids MTB 24"*, *Kids Riverside 24"*, *Riverside 20"*, *Rockrider 24"*.

A Tabela 4 apresenta as medidas estatísticas relacionadas ao presente conjunto de dados, como, mínimo, máximo, média, padrão desvio e variância.

Tabela 4 – Medidas descritivas (*kids*)

Medidas Descritivas	Unit Cost	Stock
count	900	900
mean	7,5	7334,4
std	5,6	13401,5
min	0,6	0,5
25%	4,1	166,2
50%	7	1147
75%	9,8	6433
max	66,5	81373

Na Figura 23, é ilustrada a proporção do *stock* e do custo em relação ao valor total, para cada tipo de família.

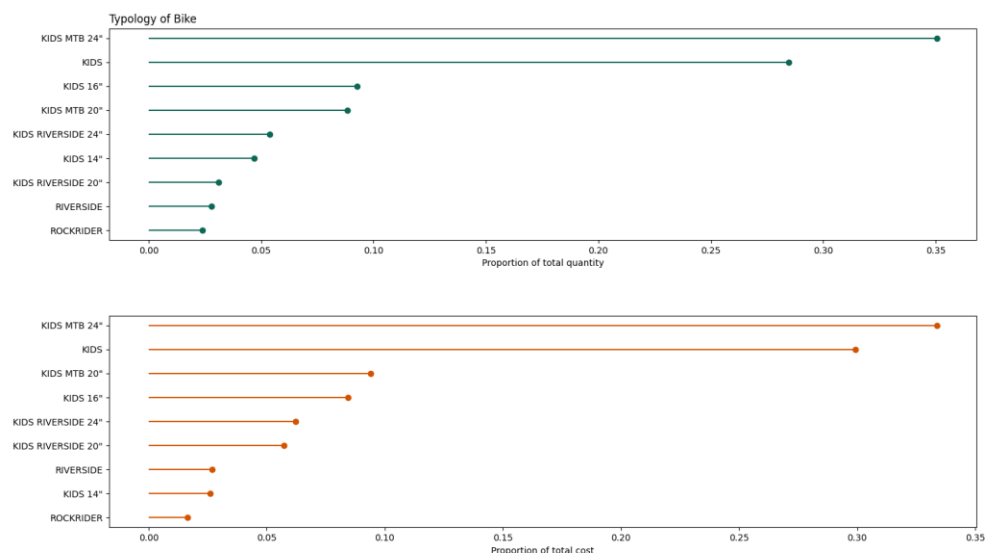


Figura 23 - Comportamento por família em relação ao custo e quantidade de stock (*kids*)

Neste caso, é possível averiguar similaridade entre a quantidade de componentes em *stock* e o custo unitário, uma vez que a família Kids MTB 24'' tem um maior impacto em ambos os casos.

A Figura 24 representa a proporção do custo em relação ao valor total, para cada tipo de componente. Comparação entre os dez componentes com custos mais elevados em extra *stock*, com o custo dos componentes estruturais.

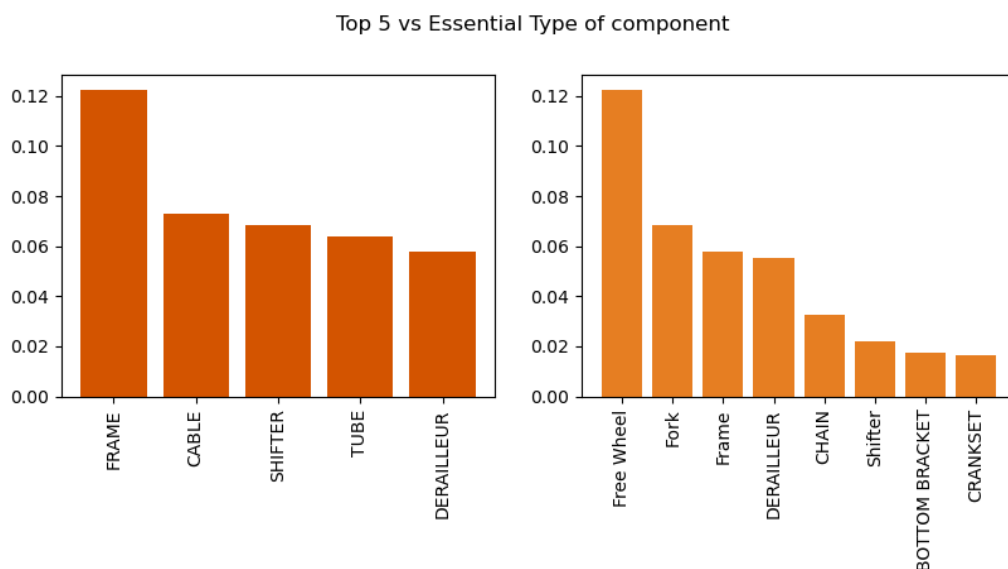


Figura 24 - Comportamento por tipologia de componentes para a quantidade de stock (*kids*)

Entre os tipos de componente com custo superior, 3 em 5 são componentes estruturais. Na Figura 25 é representado a proporção do *stock* em relação ao valor total, para cada tipo de componente, novamente recorrendo à comparação entre o top 10 e os componentes estruturais.

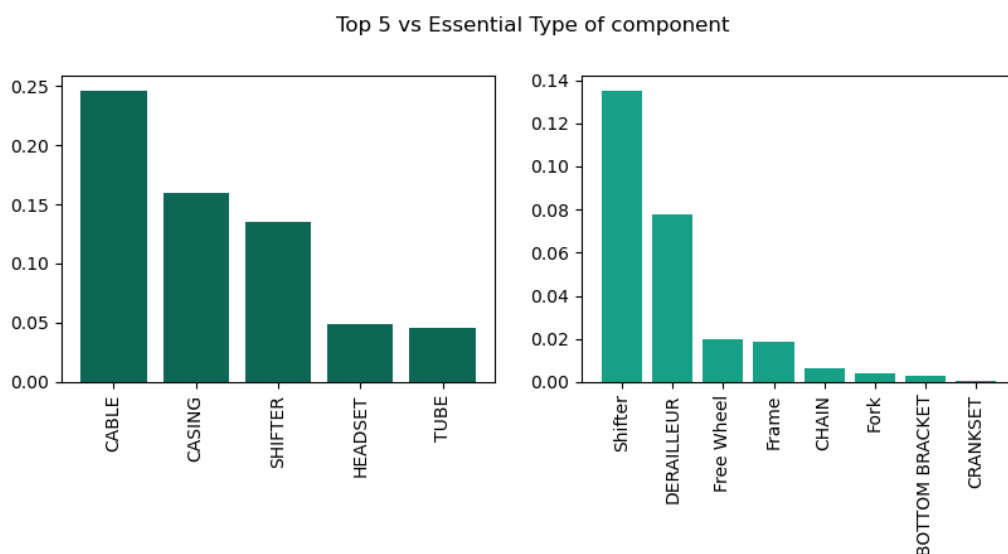


Figura 25 - Comportamento por tipologia de componentes para o custo (*kids*)

Neste caso os tipos de componente com quantidade elevada de *stock* excedentário, apenas um é considerado componente estrutural.

4.1.2 Pré-processamento de Dados

Após uma análise inicial, é essencial proceder à preparação dos dados para a análise futura, e se necessário recorrer a técnicas de tratamento de dados de modo a corrigir eventuais falhas, como valores em falta e *outliers*. No conjunto de dados, não foram detetados valores ausentes e valores discrepantes em relação ao normal, *outliers*.

No entanto, foi necessário realizar algumas alterações de modo a preparar a análise para a implementação de modelos, nomeadamente, codificação das variáveis categóricas e posterior normalização de dados numéricos.

A. Codificação One-Hot

Às vezes, em conjuntos de dados, encontramos colunas que contêm valores categóricos. Esses rótulos não têm uma ordem específica de preferência e, também, como os dados são valores de texto, os modelos de ML não conseguem interpretar corretamente esses dados.

Uma abordagem para resolver esse problema pode ser a codificação de texto, onde é atribuído um valor numérico aos rótulos e idealmente ambos os rótulos são igualmente importantes no conjunto de dados.

Para lidar com esse problema, recorre-se a técnica *One Hot Encoding*.

Nesta técnica, cada um dos parâmetros categóricos é dividido por colunas separadas em que regista o valor 1 se a observação corresponder ao cabeçalho da coluna, e o valor 0 nas observações díspar.

B. Normalização de dados

Uma vez que os dados possuem variáveis com diferentes intervalos e unidades de medida, foi necessário normalizar. Esta transformação de dados garante que todas as variáveis do conjunto de dados numéricos possuem uma escala comum.

4.2 Implementação de Modelos

Esta subseção mostra o processo de conceção e ajuste dos modelos explorados. Primeiro, iniciamos o desenvolvimento pelos modelos em árvore, árvore de decisão e *random forest*. Por fim, é apresentado o

modelo em rede. Para a demonstração a implementação e otimização dos classificadores, são utilizados os dados sobre bicicletas de criança (Kids).

4.2.1 Decision Tree

Um classificador de árvore de decisão é uma abordagem sistemática para classificação multi-classe. Coloca um conjunto de questões ao conjunto de dados (relacionadas aos seus atributos/características). Na raiz e em cada um dos nós internos, uma pergunta é colocada e os dados desse nó são posteriormente divididos em registos separados com características diferentes. As folhas da árvore referem-se às classes nas quais o conjunto de dados é dividido.

O primeiro passo é dividir as variáveis em dois tipos de variáveis dependentes (ou variáveis de destino) e variáveis independentes (ou variáveis de recursos). Neste caso, a variável dependente é denominada “label” e as restantes são variáveis independentes.

A. Divisão de dados

Para entender o desempenho do modelo, é necessário dividir o conjunto de dados em um subconjunto de treino e outro subconjunto de teste. Esta divisão ocorre de forma aleatória e esta divisão não é equitativa, uma vez que o subconjunto de treino recebe 70% dos registos e o subconjunto de teste os restantes 30%.

```
feature_cols = ['Component', 'Bike', 'Fam_spec', 'QTY', '€€', 'Stock']
X = kids[feature_cols]
y = kids.LABEL

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1)
# 70% training and 30% test
```

Figura 26 – Código referente à divisão de dados

B. Modelo de Árvore de Decisão

Para implementar o modelo de árvore de decisão recorre-se à biblioteca *Scikit-learn* que apresenta vários classificadores.

```
clf = DecisionTreeClassifier()
clf = clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
```

Figura 27- Código do classificador em árvore

C. Avaliação de Performance

Com o desenvolvimento do classificador, é necessário averiguar a performance do modelo assim, calcula-se o valor da *accuracy* que compara os valores reais do conjunto de teste e os valores previstos.

```
print("Accuracy:", metrics.accuracy_score(y_test, y_pred))
```

Figura 28 – Código para obtenção da *accuracy* do modelo

Neste caso, na primeira tentativa obteve-se um valor baixo de aproximadamente 25% de *accuracy*. Deste modo, é necessário recorrer à otimização e “poda” do modelo.

D. Otimizando o Desempenho da Árvore de Decisão e *Prunning*

O tamanho máximo da árvore pode ser utilizado como variável de controlo para o processo de *prunning*. Além deste fator também, é possível tentar outra medida de seleção de atributos, como a entropia.

- **Escolher medida de seleção de atributo:** Este parâmetro nos permite usar a medida de seleção de atributo diferente. O critério “gini” para o índice de Gini e “entropia” para o *information gain*.
- **Profundidade máxima de uma árvore:** Se nenhum dos nós são expandidos até que tenham todas as folhas. O valor mais alto da profundidade máxima causa *overfitting*, e um valor mais baixo causa *underfitting*.

Assim, opta-se por testar a alteração de vários parâmetros, até chegar ao melhor valor de precisão observado.

```
clf = DecisionTreeClassifier(criterion="entropy", max_depth=3)
clf = clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
print("Accuracy:", metrics.accuracy_score(y_test, y_pred))
```

Figura 29 – Classificador em árvore com novos parâmetros

É possível visualizar a construção de árvore representada pelo código acima descrito na Figura 30.

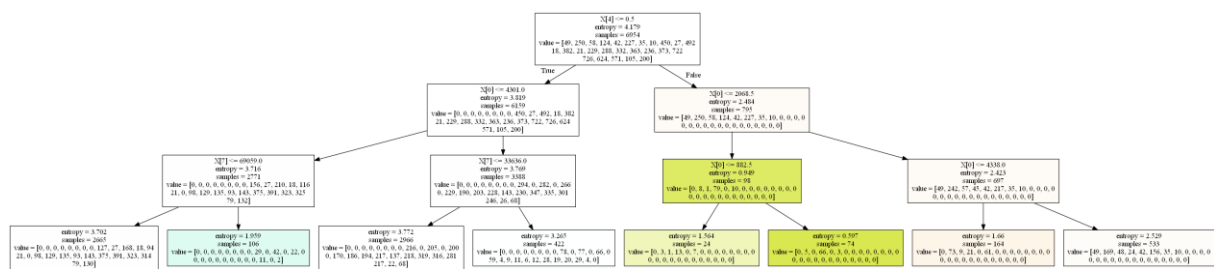


Figura 30 - Visualização de uma árvore de decisão

4.2.2 Random Forest

Random Forest recorre ao critério Gini para calcular a importância de cada característica. A importância de Gini também é conhecida como a diminuição total da impureza do nó. Representa o quanto o modelo se ajusta ou diminui a precisão quando uma variável é eliminada. Aqui, a diminuição média é um parâmetro significativo para a seleção de variáveis. O índice de Gini pode descrever o poder explicativo geral das variáveis. Contudo, primeiro é necessário implementar o modelo.

```
from sklearn.ensemble import RandomForestClassifier

clf=RandomForestClassifier(n_estimators=100)
clf.fit(X_train,y_train)
y_pred=clf.predict(X_test)
```

Figura 31 - Classificador de *Random Forest*

Algumas variáveis podem não ser tão representativas no processo de decisão, então é possível repetir o processo desde a identificação das variáveis independentes até este ponto, retirando as variáveis e observar o impacto que terá na performance.

4.2.3 Artificial Neural Network

Para a implementação e análise das redes neurais recorreu-se à biblioteca *keras* em python.

A. Definir Modelo

O primeiro passo é construir as camadas da rede e inicializar os parâmetros, assim é possível averiguar que para as três camadas os dados de inicialização são:

Input layer - um neurónio por característica

Hidden layer – como este caso é uma estrutura simples só tem um neurónio na camada oculta

Output layer – mesmo tamanho que o input

A seguir com a atribuição da função de perda e as funções de ativação a rede está definida, passando para a implementação no classificador.

```
def baseline_model():
    model = Sequential()
    model.add(Dense(input_len, input_dim = column_size, activation = 'relu'))
    model.add(Dense(1, activation = 'relu'))
    model.add(Dense(input_len, activation = 'softmax'))
    model.compile(loss = 'categorical_crossentropy', optimizer = 'adam', metrics = ['accuracy'])

    return model

estimator = KerasClassifier(build_fn = baseline_model, epochs = 20, batch_size = 1000, verbose = 0)
```

Figura 32 - Definição da estrutura de ANN

O tamanho do lote (`batch_size`) é um número de amostras processadas antes de o modelo ser atualizado e o número de épocas (`epochs`) é o número de passagens completas através do conjunto de dados de treino. Salientando que o tamanho de um lote deve ser superior ou igual a um e inferior ou igual ao número de amostras do conjunto de dados de formação. Estes dois parâmetros são normalmente submetidos a mudanças de modo a encontrar melhor valores de performance.

B. Definir *cross-validation* e treino do modelo

A implementação de *cross-validation* reduz a variação do erro estimado e melhora a estimativa do desempenho dos algoritmos de classificação multiclasse. Com alguns testes, modificando o número de divisões até entrar um valor ótimo.

```
kfold = KFold(n_splits = 5, shuffle = True, random_state = seed)
results = cross_val_score(estimator, X, Y, cv = kfold)
print("Result: %.2f%% (%.2f%%)")
```

Figura 33 - Implementação de *cross-validation*

4.3 Ferramenta de apoio à decisão

O principal objetivo deste projeto é fornecer apoio ao agente de decisão, sobre o desenvolvimento de novos modelos de bicicletas ou a substituição em modelos já existentes. Com a implementação dos modelos de ML para classificação multiclasse, tornou-se possível a máquina aprender a classificar os dados de forma mais precisa possível e assim, informar sobre novos *inputs* desconhecidos.

Para o desenvolvimento de um produto, é necessário declarar uma lista com os seus materiais constituintes, ou seja, desenvolver uma BOM. Assim, aos dados categorizados é aplicado um novo algoritmo que permita construir uma nova BOM tendo em conta a estrutura habitual de uma bicicleta.

```
def bom_build(type,model):  
    d = dict()  
    for i in list_comp:  
        if i in d:  
            d["type"].append(i)  
    bom = []  
  
    def recurse(bom, d, ds, code, stock, cost, name):  
        for n in len(range(ds)):  
            for i in d["type"]:  
                bom.append([code, stock, cost, name])  
            return bom
```

Figura 34 - Processo de construção de uma BOM

Este algoritmo devolve uma lista com os componentes e as quantidades necessárias para desenvolver um modelo.

Posteriormente e de modo a facilitar o acesso a informação, surge a necessidade de desenvolver uma ferramenta que forneça informação de forma fácil e rápida. Aliado à construção de uma BOM, recorre-se a estimativas sobre o capital retornado, assim como a capacidade de produção (quantidade que é possível industrializar do novo modelo) de modo a dar mais suporte ao agente de decisão, na questão se é viável a produção do modelo (Figura 35). A aplicação foi desenvolvida através do *package Streamlit* em *python*.

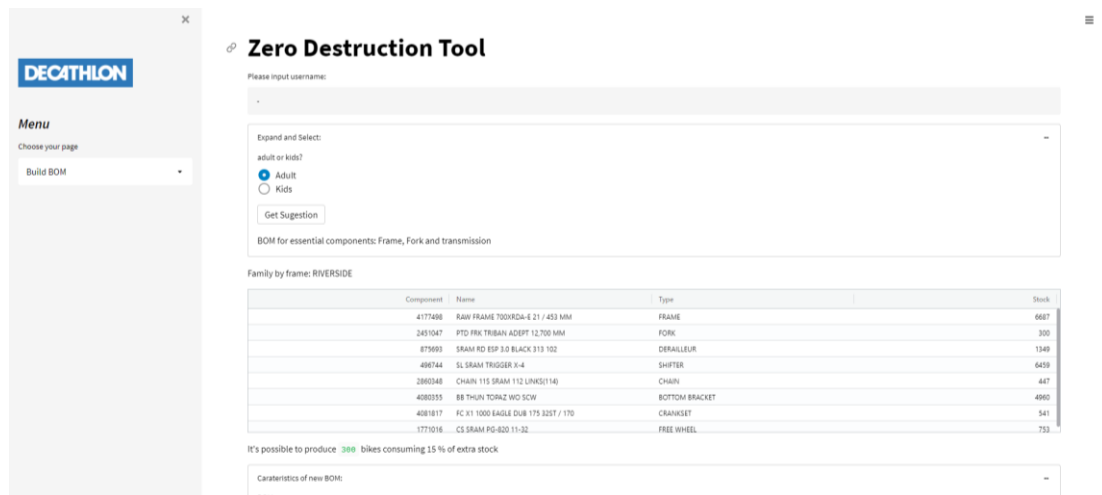


Figura 35 - Ferramenta de apoio à decisão

Contudo, também é interessante avaliar o impacto da industrialização da nova BOM na quantidade de *stock* excedentário (Figura 36).

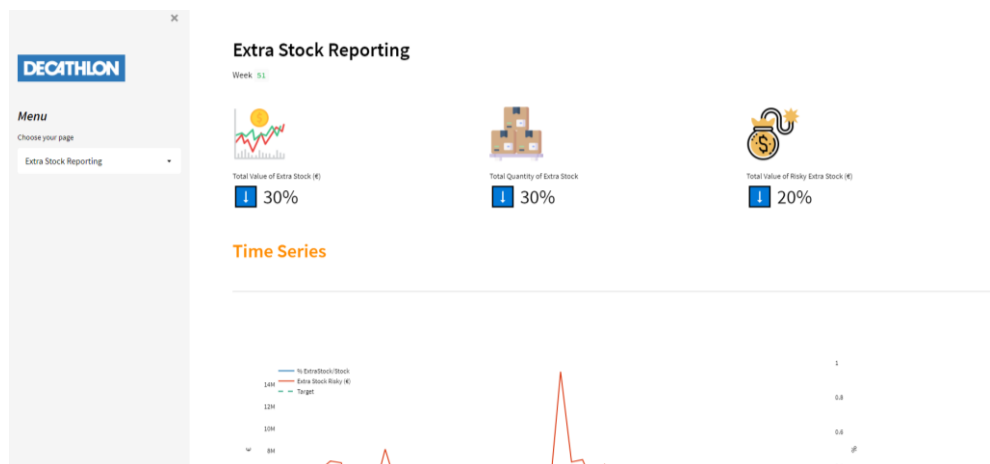


Figura 36 – Reporting presente na ferramenta de apoio à decisão

Deste modo, o agente de decisão reúne as informações necessárias para suportar a ação de desenvolvimento do modelo, aliado à avaliação do impacto da decisão na liquidação da quantidade e do retorno do valor investido.

5. ANÁLISE E DISCUSSÃO DE RESULTADOS

Este capítulo relata os principais resultados obtidos com a implementação dos modelos de ML, assim como, a posterior previsão do impacto financeiro da implementação do projeto na organização.

5.1 Experiência: Bicicleta *Adult*

Os métodos e modelos descritos no capítulo anterior foram aplicados aos dados de bicicletas de adulto. Assim, as classes alvo são a designação de cada família e tamanho existentes e remetem no total a 20 classes.

A. Árvores de Decisão

Os primeiros resultados registados são os da classificação em Árvores de Decisão. Os resultados obtidos, são apresentados na Tabela 5.

Tabela 5 – Avaliação da performance das árvores de decisão (*adult*)

Criterion	Decision Tree					
	Gini			Entropy		
	-	10	5	-	10	5
Max_length	-	10	5	-	10	5
Accuracy	88%	90%	58%	87%	89%	63%

Como é possível observar, o classificador foi testado para diversos cenários, com especial foco na alteração dos critérios de divisão e o tamanho / dimensão da árvore.

Para estas alterações e de modo a observar o método que otimiza a *performance* do modelo, foram alterados os parâmetros de forma consecutiva. A cada teste foi observada a *accuracy*, uma medida de desempenho.

Perante os resultados, no que diz respeito ao critério de divisão da árvore foram implementados o critério *gini* que recorre do índice de *gini*, e o critério *entropy* do *Information gain*.

Quando se recorre a outro método de otimização de árvores, o cenário altera-se. Para evitar o *overfitting* e *underfitting*, limita-se a dimensão da árvore e novamente compara-se a performance para cada caso. Então, todos os ramos são considerados importantes e a sua remoção leva a uma diminuição no desempenho do modelo.

Após os testes aplicados, é possível constatar que o modelo ótimo recorre a um classificador em árvore com dimensão igual a 10 com o critério de divisão *gini* com uma performance igual a 90%. Contudo, é necessário recorrer a outras medidas de avaliação para ter certeza que para este caso, realmente este é o modelo ótimo.

Tabela 6 - Medidas de avaliação (*adult*)

Class	Precision	Recall	F1-Score	ROC Curve
ELOPS L/XL	100%	100%	100%	100%
ELOPS ONE SIZE FITS ALL	100%	100%	100%	100%
ELOPS S/M	100%	100%	100%	100%
RIVERSIDE L	96%	81%	88%	88%
RIVERSIDE L/XL	100%	100%	100%	100%
RIVERSIDE M	83%	100%	91%	91%
RIVERSIDE M/XL	100%	100%	100%	100%
RIVERSIDE S	100%	79%	88%	89%
RIVERSIDE S/M	100%	100%	100%	100%
ROAD L	63%	55%	59%	59%
ROAD M	77%	100%	86%	88%
ROAD S	70%	85%	81%	79%
ROAD XL	69%	66%	68%	68%
ROAD XS	59%	81%	78%	73%
ROCKRIDER L	100%	100%	100%	100%
ROCKRIDER M	87%	81%	85%	84%
ROCKRIDER S	68%	75%	71%	71%
ROCKRIDER XS	50%	70%	60%	60%
ROCKRIDER XL	88%	81%	85%	85%
TILT ONE SIZE FITS ALL	100%	100%	100%	100%
Macro-Averaged	87%	90%	89%	-
Micro-Averaged	75%	86%	81%	-
Accuracy	90%			

Através da Tabela 6, é possível verificar as medidas de avaliação para cada classe e assim obter uma visão pormenorizada da alocação a cada classe.

As restantes medidas de avaliação corroboram o pressuposto, fornecendo uma avaliação mais profunda do modelo, uma vez que apresenta uma vista da performance para todas as classes.

B. Random Forest

Os resultados da classificação Random Forest são apresentados na Tabela 7.

Tabela 7 - Avaliação da performance de *Random Forest* (*adult*)

Random Forest	Todas as variáveis	Seleção de variáveis
Accuracy	84%	91%

Assim, a implementação do modelo para todas as variáveis obtém uma *accuracy* de aproximadamente 84%.

Para modelos de *Random Forest* é essencial averiguar a importância das variáveis, uma vez que as utilizações de variáveis com pouca relevância para o processo de classificação diminuem a *performance* geral do modelo. Deste modo, realiza-se um estudo sobre a importância das variáveis (Figura 37).

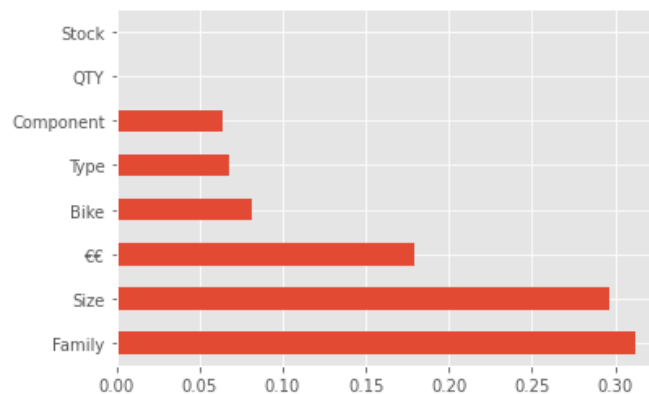


Figura 37 – Importância das variáveis para o estudo (*adult*)

Pela Figura 37, é possível concluir que as variáveis “*Stock*” e “*Qty*” não apresentam um impacto significativo no processo de classificação. Assim, é possível remover estas variáveis do modelo e averiguar novamente a *performance* do modelo.

Com esta alteração, o modelo obtém uma *accuracy* de aproximadamente 91%, logo uma melhoria em relação ao modelo anterior. Graficamente, é possível observar a *performance* do modelo através da sua matriz de confusão, onde os valores corretamente classificados são apresentados na diagonal (Figura 38).

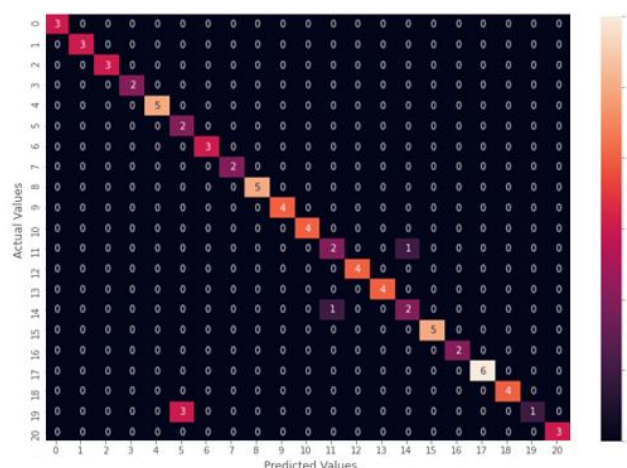


Figura 38 - Matriz de Confusão (*adult*)

Fazendo uma análise geral, pode-se concluir que a remoção de atributos irrelevantes, como a quantidade em *stock* ou a quantidade utilizada de cada componente na montagem da bicicleta, pode trazer melhores resultados.

C. Artificial Neural Network

A Tabela 8 mostra os resultados do processo de *tuning*, por cenário.

Tabela 8 - Avaliação da performance de ANN (*adult*)

	ANN			
Epoch	20	100	20	100
n_splits	5	5	10	10
Accuracy	79%	96%	80%	98%

Para modelos baseados em redes, os parâmetros para otimizar remetem ao número de iterações (*epochs*) e através da utilização de *cross-validation*, o parâmetro de número de divisões (n_splits).

Em relação ao número de iterações, pela Tabela 8, é possível verificar que a performance melhora para valores mais altos. Com a implementação de *cross-validation* também apresenta melhorias. Estes acontecimentos podem ser visualizados na Figura 39.

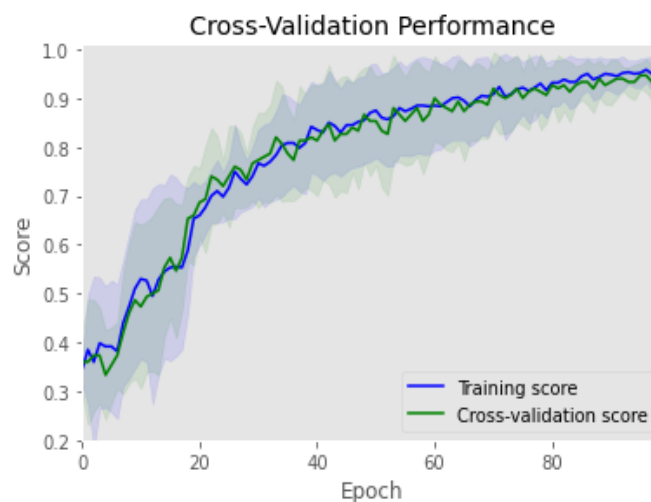


Figura 39 - Cross-Validation Performance (*adult*)

Pela análise efetuada é possível verificar que a performance é mais alta, ou seja, melhora com o aumento do número de iterações. Em relação, à implementação da *cross-validation* não é visivelmente significativa.

Assim, o modelo ótimo perante os cenários realizados é o modelo com o número de iterações igual a 100 e o número de divisões igual a 10.

Assim, este modelo e a sua performance pode observada na Figura 40.

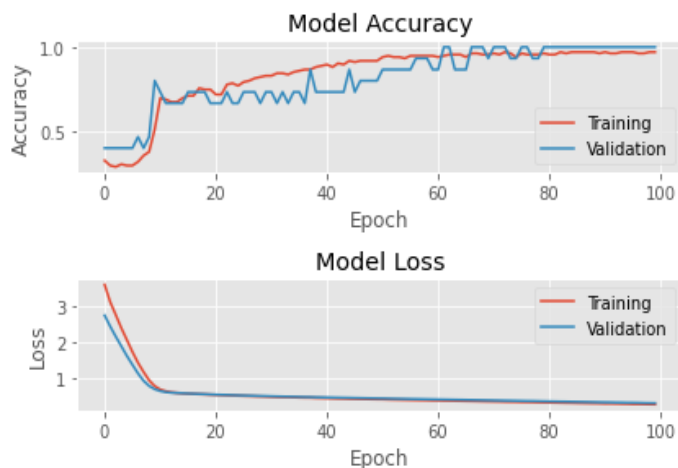


Figura 40 – Accuracy and loss of model (adult)

5.2 Experiência: Bicicleta Kids

Os mesmo métodos e técnicas foram aplicadas aos dados de bicicletas de criança. Assim, as classes alvo são a designação de cada família e tamanho existentes, para a totalidade de 9 classes.

A. Árvores de Decisão

Os primeiros resultados registados são os da classificação em Árvores de Decisão. Os resultados obtidos, são apresentados na Tabela 9.

Tabela 9 - Avaliação da *performance* das árvores de decisão (kids)

Decision Tree						
Criterion	Gini			Entropy		
Max_length	-	10	5	-	10	5
Accuracy	77%	79%	80%	79%	81%	83%

Perante os resultados, no que diz respeito ao critério de divisão da árvore foram, novamente, implementados o critério *gini* que recorre do índice de *gini*, e o critério “entropia” do *Information gain*. Em termos gerais, o critério *entropy* obteve melhores resultados.

Após os testes aplicados, é possível constatar que o modelo ótimo recorre a um classificador em árvore com dimensão igual a 5 com o critério de divisão *entropy* com uma performance igual a 83%. Contudo, é necessário recorrer a outras medidas de avaliação para ter certeza que para este caso, realmente este é o melhor modelo (Tabela 10).

Tabela 10 - Medidas de avaliação (*kids*)

Class	Precision	Recall	F1-Score	ROC Curve
KIDS 14"	100%	100%	100%	100%
KIDS 16"	100%	100%	100%	100%
KIDS 20"	65%	72%	67%	75%
KIDS 24"	100%	100%	100%	100%
KIDS MTB 20"	100%	100%	100%	100%
KIDS MTB 24"	68%	75%	71%	72%
KIDS RIVERSIDE 24"	0%	0%	0%	48%
RIVERSIDE 20"	100%	100%	100%	100%
ROCKRIDER 24"	100%	100%	100%	100%
Macro-Averaged	79%	82%	80%	-
Micro-Averaged	70%	70%	70%	-
Accuracy	83%			

A. Random Forest

Os resultados da classificação *Random Forest* são apresentados na Tabela 11.

Tabela 11 - Avaliação da performance de *Random Forest* (*kids*)

Random Forest	Todas as variáveis	Seleção de variáveis
Accuracy	79%	86%

Assim, a implementação do modelo para todas as variáveis obtém uma *accuracy* de aproximadamente 79%. Através do processo mencionado na experiência anterior, realiza-se um estudo sobre a importância das variáveis (Figura 41).

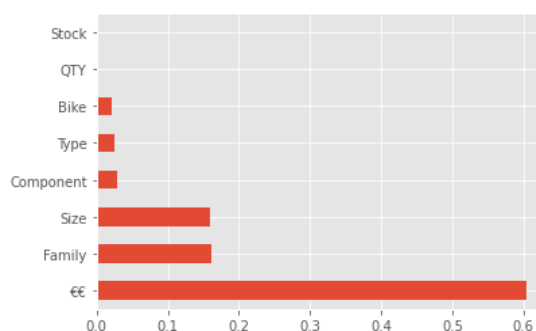


Figura 41– Importância das variáveis para o estudo (*kids*)

Pela Figura 41, é possível concluir que as variáveis “*Stock*” e “*Qty*” não apresentam um impacto significativo no processo de classificação. Assim, é possível remover estas variáveis do modelo e averiguar novamente a performance do modelo.

Com esta alteração, o modelo obtém uma *accuracy* de aproximadamente 86%, logo uma melhoria em relação ao modelo anterior. Graficamente, é possível observar a performance do modelo através da sua matriz de confusão, onde os valores corretamente classificados são apresentados na diagonal (Figura 42).

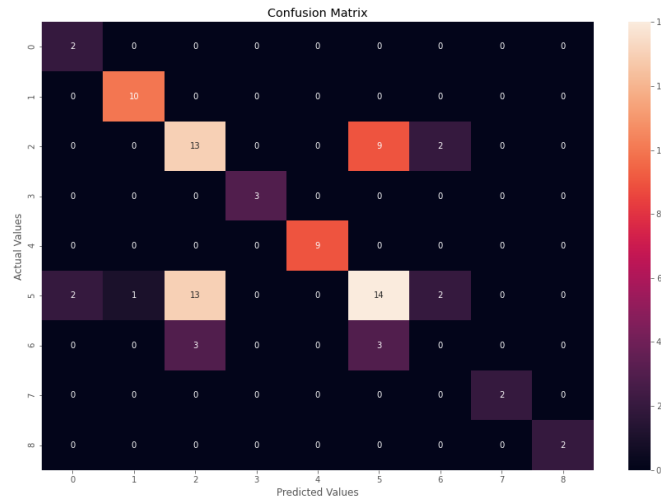


Figura 42 - Matriz de Confusão (*kids*)

A. Artificial Neural Network

A Tabela 12 mostra os resultados do processo de *tuning*, por cenário.

Tabela 12 - Avaliação da performance de ANN (*kids*)

	ANN			
Epoch	20	100	20	100
n_splits	5	5	10	10
Accuracy	75%	87%	81%	93%

Em relação ao número de iterações, pela Tabela 12, é possível verificar que a performance melhora para valores mais altos.

Com a implementação de *cross-validation* também apresenta melhorias. Estes acontecimentos podem ser visualizados através da Figura 43.

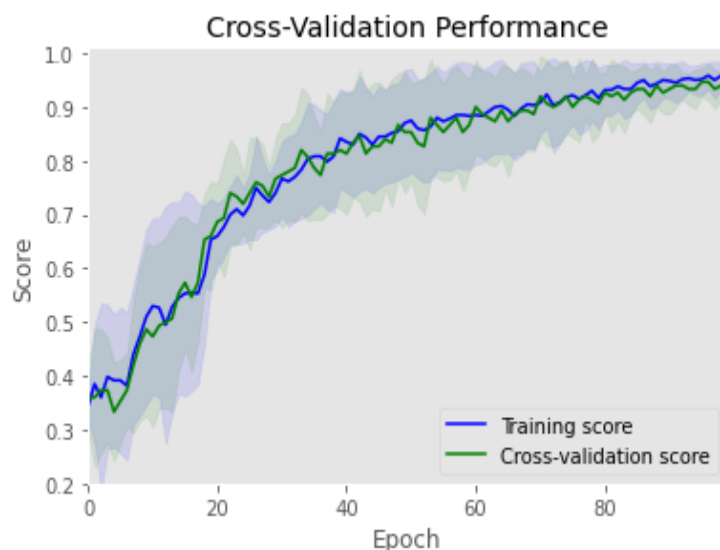


Figura 43 - Cross-Validation Performance (*kids*)

Pela análise efetuada é possível verificar que a performance é mais alta, ou seja, melhor com o aumento do número de iterações. Em relação, à implementação da *cross-validation* não é visivelmente significativa.

Assim, o modelo ótimo perante os cenários realizados é o modelo com o número de iterações igual a 100 e o número de divisões igual a 10, com *accuracy* de 93%.

Assim, este modelo e a sua performance pode ser observada na Figura 44.

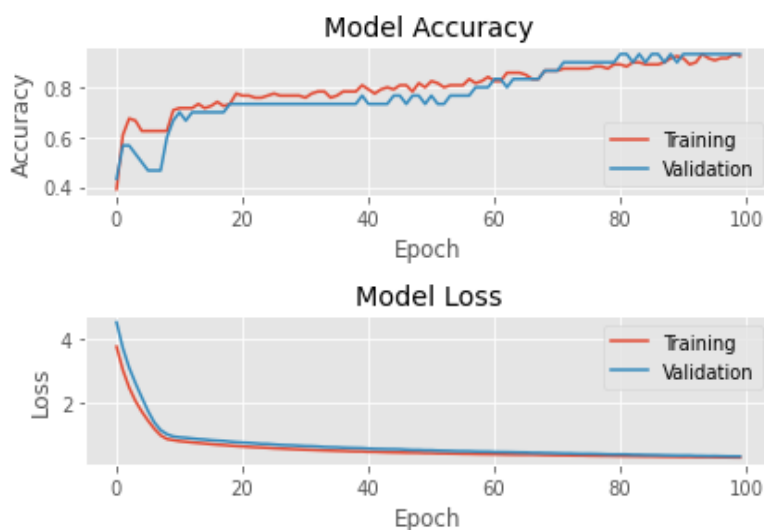


Figura 44 - Accuracy and loss of model (*kids*)

Pela Figura 44, é possível observar que o aumento de iterações provoca um efeito positivo na *performance* do modelo.

5.2.1 Comparação entre modelos

Depois de conhecidos todos os resultados dos modelos de classificação é possível fazer uma análise comparativa dos modelos de árvore implantados (*Decision Tree* e *Random Forest*) e ANN.

A Tabela 13 mostra o desempenho de cada modelo de classificação em relação à experiência de criança.

Tabela 13 - Comparação entre modelos (*kids*)

Kids	DT	RF	ANN
Accuracy	83%	86%	93%

Já a Tabela 14, mostra o desempenho de cada modelo de classificação para a experiência de adulto.

Tabela 14 - Comparação entre modelos (*adult*)

Adult	DT	RF	ANN
Accuracy	90%	91%	98%

A primeira conclusão é que o modelo que geralmente produz melhores valores de *accuracy* é ANN.

Para os cenários estudados, os modelos baseados em árvore produziram resultados semelhantes. Contudo, para o *Random Forest* apresentam no geral um melhor resultado, quando comparado com as árvores de decisão. Facto, que se alinha o levantamento de estudos em artigos com a mesma comparação mencionada previamente. O mesmo, acontece quando se compara o *Random Forest* e ANN, este último apresenta um resultado significativamente melhor para as ambas as experiências.

5.2.2 Impacto na análise financeira

O principal objetivo deste estudo é reaproveitamento de *stock* excedentário. Assim, após a implementação e avaliação dos modelos que permitem classificar os componentes para as respetivas famílias o próximo passo é construir novos modelos de modo a escoar o *stock* entre portas, retornar o capital investido e obter lucro sobre os produtos.

Perante a construção de novos modelos é possível observar a capacidade de produção para cada tipo de bicicleta. Para efetuar esta análise a divisão entre bicicletas de adulto e criança continua presente.

A. Bicicleta Adult

Inicialmente, é possível estimar o nível de consumo do *stock* para cada tipo de componente e avaliar quais representam uma maior taxa de aproveitamento de modo a originar novos artigos. A taxa de aproveitamento por tipo de componente é representada pela Tabela 15.

Tabela 15 – Aproveitamento por tipo de componente (adult)

Type	%
FRAME	25%
BRAKES	22%
FORK	21%
DERAILLEUR	16%
SHIFTER	15%

Tendo em consideração, o aproveitamento dos componentes estruturais de uma bicicleta, é possível prever, para as bicicletas de adulto, uma taxa de aproveitamento de *stock* excedentário de, aproximadamente, 87%. De notar, que os restantes componentes alocados e complementares à montagem, serão provenientes do stock normal. Deste modo, é possível averiguar a capacidade de produção de cada família e tamanho, como se apresenta na Tabela 16.

Tabela 16 – Previsão da capacidade produtiva (adult)

Família/Tamanho	L/XL	M/XL	M	L	S/M	S	ONE SIZE FITS ALL	XS	XL	Total
RIVERSIDE	24	0	5863	5274	24	4840	0	0	325	16350
ELOPS	1700	0	142	0	2899	3431	286	0	68	8525
TILT	0	0	0	0	0	0	1869	0	0	1869
ROCKRIDER	0	0	206	197	0	167	0	2881	80	3531
ROAD	0	0	3	0	0	521	0	297	1	821

De modo averiguar o impacto e o retorno da produção destas quantidades, avalia-se a taxa de investimento monetário por família, averiguando desde o custo de compra dos componentes até ao custo de produção associado. Na mesma medida, averigua-se o preço médio para cada tipo de bicicleta, através do preço médio de mercado da Decathlon. Posteriormente, avalia-se a margem de lucro estimada com a venda dos produtos nas plataformas específicas (Tabela 17).

Tabela 17 – Lucro e preço médio e respetiva diferença (*adult*)

Family	Preço médio de mercado	Margem de Lucro
RIVERSIDE	400 €	31%
ELOPS	700 €	47%
TILT	600 €	13%
ROCKRIDER	800 €	21%
ROAD	500 €	10%

Perante a capacidade produtiva e assumindo a quantidade média de unidades vendidas por ano para aproximadamente 5000 unidades, tendo em conta a demora de industrialização destes modelos, uma vez que não retêm capacidade de produção em massa. Assim, a liquidação de 87% do *stock* pode ser estudada para um horizonte temporal, de modo a prever a data alvo para o retorno total.

Tabela 18 - Previsão de liquidez de artigos (*adult*)

1 ano	5 anos	7 anos
14%	71%	100%

Pela Tabela 18, é espectável liquidar os modelos desenvolvidos e consequentemente o *stock* excendentário em 7 anos.

B. Bicicleta Kids

Como no exemplo anterior, avalia-se a taxa de aproveitamento por tipo de componente e a mesma é representada pela Tabela 19.

Tabela 19 - Aproveitamento por tipo de componente (*kids*)

Type	%
SHIFTER	15%
FRAME	22%
DERAILLEUR	14%
TUBE	11%
CABLE	8%

Neste cenário, a taxa de aproveitamento dos componentes estruturais é ligeiramente inferior e remete a uma previsão de utilização de *stock* excedentário de aproximadamente 83%. Deste modo, é possível averiguar a capacidade de produção de cada família e tamanho (Tabela 20).

Tabela 20 - Previsão da capacidade produtiva (*kids*)

Família/Tamanho	24"	20"	14"	No Size	16"	Total
ROCKRIDER	0	0	0	9342	0	9342
KIDS	0	0	0	4713	16	4728
KIDS MTB	0	0	0	1773	0	1773
RIVERSIDE	1978	9342	0	0	0	11320

Na mesma medida, averigua-se o preço médio para cada tipo de bicicleta, novamente, através do preço médio de mercado para modelos semelhantes e consosante a o gasto na produção, o lucro esteptado com a venda dos artigos (Tabela 21).

Tabela 21 - Lucro e preço médio e respetiva diferença (*kids*)

Family	Preço médio de mercado	Margem de Lucro
ROCKRIDER	300 €	49%
KIDS	150 €	12%
KIDS MTB	250 €	21%
RIVERSIDE	200 €	30%

Como no exemplo anterior, estipula-se a mesma capacidade produtiva e assim, as quantidades disponíveis para venda em 5000 unidades. Assim, avaliando o a progressão de vendas previstas para um horizonte temporal de 1 a 6 anos, é possível prever o gasto de 83% dos componentes estruturais nos próximos 6 anos (Tabela 22).

Tabela 22 - Previsão de liquidez de artigos (*kids*)

1 ano	4 anos	6 anos
18%	90%	99%

Embora, os horizontes temporais em estudo sejam relativamente elevados é necessário compreender as limitações de produção. Os modelos mencionados, sejam de *Kids* ou *Adult*, não são passíveis de produção em massa devido às suas diversas modificações dos componentes, sendo estes modelos praticamente de edição limitada.

Também se encontram passíveis de demora na alocação de outros componentes derivados do *stock* normal e alocado a outras industrializações, como por exemplo pequenos acessórios, para terminar o seu processo de montagem e posterior colocação no mercado, com o selo de qualidade Decathlon.

6. CONCLUSÃO

A precariedade de ações eficientes para colmatar o crescente aumento da quantidade de *stock* imobilizado, surge a necessidade de reaproveitar os componentes através da criação de modelos únicos em produto acabado. Desenvolvidos através da análise de semelhança entre as características de cada componente e das combinações possíveis de implementar. A capacidade produtiva é limitada às quantidades existentes.

Deste modo, é fulcral a aquisição de conhecimento sobre o processo industrial e, estruturar as principais características e restrições que influenciam no desenvolvimento de um modelo consoante o tipo de bicicleta. Assim, perante as diferentes características por tipologia de bicicleta, os modelos desenvolvidos serão agrupados por família e tamanho. No que diz respeito às bicicletas de adulto e criança, devido às suas diferenças estruturais, serão analisados em separado.

Perante a necessidade de deteção de semelhanças entre componentes e padrões na alocação a cada modelo, com base no histórico, recorre-se a implementação de modelos de *machine learning* de modo a automatizar o levantamento de informação. Devido à natureza categórica dos dados em estudo, os modelos contemplados remetem à classificação multiclasse em aprendizagem supervisionada. Com vista a perceção do modelo mais adequado para a resolução do problema, são implementados três modelos, nomeadamente, *Decision Tree*, *Random Forest* e *Artificial Neural Network*.

Para os três modelos implementados, estes foram avaliados para os dois conjuntos de dados em estudo, isto é, os dados relativos a bicicletas de criança (*kids*) e de adulto (*adult*). Algumas inferências são realizadas de modo a preparar os dados para a implementação dos modelos. Além disso, os modelos foram construídos e ajustados para alcançar as melhores classificações, em relação aos conjuntos de dados disponíveis.

Os primeiros modelos construídos e implementados foram os modelos baseados em árvore, as Árvores de Decisão e *Random Forest*. Em seguida, o modelo mais complexo, isto é, o modelo baseado em redes neurais artificiais, ANN.

No que diz respeito à implementação e otimização dos modelos, conclui-se que o modelo que apresenta melhores previsões é o ANN, apresentando maior *accuracy* nos dois conjuntos de dados.

O modelo *Random Forest*, aquando da remoção das variáveis referentes à quantidade em *stock* e à quantidade do componente utilizado na construção da bicicleta, identificadas como irrelevantes para a modelação, resulta numa melhoria significativa do desempenho.

O modelo de árvore de decisão, em geral, é o que apresenta os piores resultados. Finalmente, é possível concluir que, para os dados de bicicleta de adulto a melhor *accuracy* (98%) decorre da ANN e para os dados de bicicleta de criança obtém uma *accuracy* menor, mas muito satisfatória (93%) novamente através da ANN. A ANN requer maiores esforços computacionais em comparação com os restantes. Além disso, exigindo um tempo elevado para a construção e implementação deste modelo. Contudo, este modelo permitiu obter bons resultados para este problema.

No que diz respeito ao reaproveitamento de *stock* excedentário, é possível verificar que com a implementação dos modelos e técnicas mencionadas é possível liquidar 87 % do *stock*, no caso de bicicletas de adulto, e 83%, para bicicletas de criança. E, em média, é possível retornar na totalidade o capital investido e a devida margem de lucro associada à produção de cada artigo para um horizonte de seis anos.

Os maiores desafios desta pesquisa são baseados principalmente no processo de ajuste dos modelos. Este processo requer muito tempo e uma alta capacidade computacional (principalmente modelos em rede). Ainda assim, como existem vários parâmetros e cenários, leva um tempo considerável para obter resultados confiáveis.

Por fim, é possível concluir que a implementação da pesquisa explanada permite solucionar a problemática do *stock* no presente contexto empresarial, através do processo de classificação e do BOM para novos produtos. Este trabalho permite criar uma ponte entre o campo de ML e as necessidades empresariais para obter vantagem dos dados disponíveis.

6.1 Limitações e Oportunidades para Trabalho Futuro

Os modelos implementados e avaliados no decorrer do estudo, apresentam um grau de complexidade elevado. Deste modo, a principal limitação incorre no tempo de treino dos modelos até à obtenção de índices de performances adequados.

O modelo de ANN implementado remete à análise de dados categóricos. Como trabalho futuro, surge o aumento da complexidade dos modelos através do recurso a *deep learning* para classificação de imagens. Este fator seria de grande utilidade, uma vez que permite a leitura de informação através de uma imagem, e assim obter características específicas que poderiam tornar a análise e o desenvolvimento dos modelos mais ao pormenor e menos genérica à tipologia de utilização.

Também, face à construção de novos modelos de bicicletas surge a necessidade de melhorar a ferramenta de apoio à decisão, de modo a não só fornecer suporte ao agente de decisão, mas também como uma ferramenta capaz de fornecer suporte no processo operacional de industrialização.

REFERÊNCIAS BIBLIOGRÁFICAS

- Ahi, Payman, and Cory Searcy. 2013. "A Comparative Literature Analysis of Definitions of Green and Sustainable Supply Chain Management." *Journal of Cleaner Production* 52:329–41. doi: 10.1016/j.jclepro.2013.02.018.
- Ahmad, Muhammad Waseem, Monjur Mourshed, and Yacine Rezgui. 2017. "Trees vs Neurons: Comparison between Random Forest and ANN for High-Resolution Prediction of Building Energy Consumption." *Energy and Buildings* 147:77–89. doi: 10.1016/j.enbuild.2017.04.038.
- Akindipe, Olusakin S. 2014. "Inventory Management - A Tool for Optimal Use of Resources and Overall Efficiency in Manufacturing SMEs." *Journal of Entrepreneurship, Management and Innovation* 10(4):93–113. doi: 10.7341/20141044.
- Analide, Cesar, Paulo Novais, Jose Machado, and Jose Neves. 2006. "Quality of Knowledge in Virtual Entities." Pp. 436–42 in *Encyclopedia of Communities of Practice in Information and Knowledge Management*. IGI Global.
- Apao, Norma J., Larmie S. Feliscuzo, Cherry Lyn C. St. Romana, and Jennifer Aurea S. Tagaro. 2020. "Multiclass Classification Using Random Forest Algorithm to Prognosticate the Level of Activity of Patients with Stroke." *International Journal of Scientific and Technology Research* 9(4):1233–40.
- Aro-gordon, Stephen. 2016. "Review of Modern Inventory Management Techniques." *Global Journal of Business & Management* 1(September):2.
- Aro-Gordon, Stephen, and Jaideep Gupte. 2016. "Contemporary Inventory Management Techniques: A Conceptual Investigation." *International Conference on Operations Management and Research: "Towards Operational Excellence"* (January):20.
- Atnafu, Daniel, and Assefa Balda. 2018. "Inventory Management Practice in Micro and Small Enterprise : The Case of MSEs ' Manufacturing Sub Sector Arsi Zone , Ethiopia." *Industrial Engineering Letters* 8(1):13–20.
- Ayyadevara, V. Kishore. 2018. *Pro Machine Learning Algorithms*. Berkeley, CA: Apress.
- Ballou, R. 2004. *Business Logistics/ Supply Chain Management*. New Jersey: Pearson Prentice Hall.
- Bekkerman, Ron, Mikhail Bilenko, and John Langford. 2011. "Scaling up Machine Learning: Introduction." *Scaling up Machine Learning: Parallel and Distributed Approaches* 9780521192:1–20. doi: 10.1017/CBO9781139042918.002.
- Bharati, M., and Bharati Ramageri. 2010. "Data Mining Techniques and Applications." *Indian Journal of Computer Science and Engineering* 1.

- Cabena, Peter, Pablo Hadjinian, Rolf Stadler, Jaap Verhees, and Alessandro Zanasi. 1998. *Discovering Data Mining: From Concept to Implementation*. Prentice-Hall, Inc.
- Carneiro, Davide, Paulo Novais, José Miguel Pêgo, Nuno Sousa, and José Neves. 2015. "Using Mouse Dynamics to Assess Stress During Online Exams." Pp. 345–56 in *Hybrid Artificial Intelligent Systems*, edited by E. Onieva, I. Santos, E. Osaba, H. Quintián, and E. Corchado. Cham: Springer International Publishing.
- Carneiro, João, Pedro Saraiva, Diogo Martinho, Goreti Marreiros, and Paulo Novais. 2018. "Representing Decision-Makers Using Styles of Behavior: An Approach Designed for Group Decision Support Systems." *Cognitive Systems Research* 47:109–32. doi: <https://doi.org/10.1016/j.cogsys.2017.09.002>.
- Carvalho, J., Guedes, A., Arantes, A., Martins, A., Póvoa, A., Luís, C., et al. 2012. *Logística e Gestão Da Cadeia de Abastecimento*. Lisboa: Edições Silabo.
- Carvalho, J. C. 2002. *Logística*. 3ª. Lisboa: Edições Silabo.
- Chalotra, Vipul. 2013. "Inventory Management and Small Firms Growth: An Analytical Study in Supply Chain." *Vision: The Journal of Business Perspective* 17:213–22.
- Chapman, Peter, Janet Clinton, Randy Kerber, Tom Khabaza, Thomas P. Reinartz, Colin Shearer, and Richard Wirth. 2000. "CRISP-DM 1.0: Step-by-Step Data Mining Guide."
- Chase, R. B. & Aquilano, N. J. 1995. *Gestão Da Produção e Das Operações – Perspectiva Do Ciclo de Vida*. Lisboa: Projectos e Edições, Lda.
- Fausett, Laurene V. 1994. *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*. Englewood Cliffs (N.J.) : Prentice-Hall.
- Grandini, Margherita, Enrico Bagli, and Giorgio Visani. 2020. "Metrics for Multi-Class Classification: An Overview." *ArXiv* abs/2008.0.
- Han, J. W., M. Kamber, and J. Pei. 2012. *Data Mining Concepts and Techniques*. 3rd Editio. Morgan Kaufmann Publishers.
- Hanley, J. A., and B. J. McNeil. 1982. "The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve." *Radiology* 143(1):29–36. doi: 10.1148/radiology.143.1.7063747.
- Heaton, Jeffrey. 2012. "Introduction to the Math of Neural Networks." *Heaton Research, New York*.
- Janikow, Cezary Z. 1998. "Fuzzy Decision Trees: Issues and Methods." *IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics: A Publication of the IEEE Systems, Man, and Cybernetics Society* 28 1:1–14.
- Kashyap, Patanjali. 2018. *Machine Learning for Decision Makers: Cognitive Computing Fundamentals*

- for *Better Decision Making*. 1st ed. USA: Apress.
- Kościelniak, Helena, and Agnieszka Puto. 2015. "BIG DATA in Decision Making Processes of Enterprises." *Procedia Computer Science* 65(Iccmit):1052–58. doi: 10.1016/j.procs.2015.09.053.
- Kumar, U. Dinesh. 2017. *Business Analytics: The Science of Data - Driven Decision Making*. Wiley.
- Lisboa, João Veríssimo, and Carlos Ferreira Gomes. 2006. *Gestão Das Operações*. Porto: Editora Vida Económica.
- Lutz, Mark. 2003. *Learning Python*. 2nd ed. USA: O'Reilly & Associates, Inc.
- Mahesh, Batta. 2019. "Machine Learning Algorithms -A Review." *International Journal of Science and Research (IJSR)* 9(1):381–86.
- Mariscal, Gonzalo, Óscar Marbán, and Covadonga Fernández. 2010. "A Survey of Data Mining and Knowledge Discovery Process Models and Methodologies." *The Knowledge Engineering Review* 25(2):137–66. doi: 10.1017/S0269888910000032.
- Mingers, John. 1989. "An Empirical Comparison of Pruning Methods for Decision Tree Induction." *Machine Learning* 4:227–43. doi: 10.1023/A:1022604100933.
- Mitchell, Tom M. 1997. *Machine Learning*. New York: McGraw-Hill Science/Engineering/Math.
- Murugan, Pushparaja. 2018. "Implementation of Deep Convolutional Neural Network in Multi-Class Categorical Image Classification."
- Myles, Anthony, Robert Feudale, Yang Liu, Nathaniel Woody, and Steven Brown. 2004. "An Introduction to Decision Tree Modeling." *JOURNAL OF CHEMOMETRICS* 275–85. doi: 10.1002/cem.873.
- Novaković, Jasmina Dj, Alempije Veljović, Siniša S. Ilić, Željko Papić, and Tomović Milica. 2017. "Evaluation of Classification Models in Machine Learning." *Theory and Applications of Mathematics & Computer Science* 7(1):39–46.
- Nwankpa, Chigozie, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. 2018. "Activation Functions: Comparison of Trends in Practice and Research for Deep Learning." 1–20.
- Pal, Mahesh. 2005. "Random Forest Classifier for Remote Sensing Classification." *International Journal of Remote Sensing - INT J REMOTE SENS* 26:217–22. doi: 10.1080/01431160412331269698.
- Penumuru, Durga Prasad, · Sreekumar Muthuswamy, and · Premkumar Karumbu. 2020. "Identification and Classification of Materials Using Machine Vision and Machine Learning in the Context of Industry 4.0." *Journal of Intelligent Manufacturing* 31:1229–41. doi: 10.1007/s10845-019-01508-6.
- Qiu, Junfei, Qihui Wu, Guoru Ding, Yuhua Xu, and Shuo Feng. 2016. "A Survey of Machine Learning for Big Data Processing." *EURASIP Journal on Advances in Signal Processing* 2016(1):67. doi:

10.1186/s13634-016-0355-x.

Quinlan, J. Ross. 1992. *Programs for Machine Learning*. Vol. 7. 1st Editio. Morgan Kaufmann.

Raglio, Alfredo, Marcello Imbriani, Chiara Imbriani, Paola Baiardi, Sara Manzoni, Marta Gianotti, Mauro Castelli, Leonardo Vanneschi, Francisco Vico, and Luca Manzoni. 2020. "Machine Learning Techniques to Predict the Effectiveness of Music Therapy: A Randomized Controlled Trial." *Computer Methods and Programs in Biomedicine* 185:105160. doi: 10.1016/j.cmpb.2019.105160.

Ramos, Nuno M. M., João M. P. Q. Delgado, Ricardo M. S. F. Almeida, Maria L. Simões, and Sofia Manuel. 2016. "Data Mining Techniques." Pp. 13–30 in *SpringerBriefs in Applied Sciences and Technology*. Vol. 179.

Rebala G., Ravi A., Churiwala S. 2019. "Classification. In: An Introduction to Machine Learning." Springer, Cham.

Reis, L. d. 2008. *Manual Da Gestão de Stocks - Teoria e Prática*. 2ª. Lisboa: Editorial Presença.

Rosenfield, Donald B. 1989. "Disposal of Excess Inventory." *Oper. Res.* 37:404–9.

Sandryhaila, Aliaksei, and Jose M. F. Moura. 2014. "Big Data Analysis with Signal Processing on Graphs: Representation and Processing of Massive Data Sets with Irregular Structure." *IEEE Signal Processing Magazine* 31(5):80–90. doi: 10.1109/MSP.2014.2329213.

Shalev-Shwartz, Shai, and Shai Ben-David. 2013. *Understanding Machine Learning: From Theory to Algorithms*. Vol. 9781107057.

Shearer, Colin. 2000. "The CRISP-DM Model: The New Blueprint for Data Mining." *JOURNAL OF DATA WAREHOUSING* 5(4):480–84. doi: 10.1109/EITech.2016.7519646.

Song, Rui, Fengming Li, Wei Quan, Xuting Yang, and Jie Zhao. 2021. "Skill Learning for Robotic Assembly Based on Visual Perspectives and Force Sensing." *Robotics and Autonomous Systems* 135:103651. doi: <https://doi.org/10.1016/j.robot.2020.103651>.

Toomey, Dan. 2017. *Jupyter for Data Science*. 1st editio. Birmingham, England: Packt Publishing.

Wang, Lidong. 2017. "Data Mining , Machine Learning and Big Data Analytics." *International Transaction of Electrical and Computer Engineers System* 4(2):55–61. doi: 10.12691/iteces-4-2-2.

Yin, Hao Hua Sun, Klaus Langenheldt, Mikkel Harlev, Raghava Rao Mukkamala, and Ravi Vatrappu. 2019. "Regulating Cryptocurrencies: A Supervised Machine Learning Approach to De-Anonymizing the Bitcoin Blockchain." *Journal of Management Information Systems* 36(1):37–73. doi: 10.1080/07421222.2018.1550550.

Zermati, Pierre. 2000. *A Gestão de Stock*. 2ª. Lisboa: Editorial Presença.