



Modelos Preditivos para Suporte ao Planeamento de Produção Têxtil

João Gonçalo Pinheiro Azevedo

Universidade do Minho
Escola de Engenharia





Universidade do Minho

Escola de Engenharia

João Gonçalo Pinheiro Azevedo

Modelos Preditivos para Suporte ao Planeamento de Produção Têxtil

Dissertação de Mestrado

Mestrado Integrado em Engenharia e Gestão de
Sistemas de Informação

Trabalho efetuado sob a orientação de:

Professor Doutor Paulo Alexandre Ribeiro Cortez

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

CC BY-NC

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

AGRADECIMENTOS

Do latim *“gratitudo”*, o sentimento de lembrança e agradecimento por um bem recebido é por nós reconhecido como "gratidão" e é, também, o sentimento que me urge a cada um de vós.

Ao Professor Doutor Paulo Cortez, um agradecimento especial pelo acompanhamento e pelos ensinamentos que tornaram possível a escrita desta dissertação e o despoletar do interesse por esta área.

À equipa de ML do CCG, pelo conhecimento que me passaram, pela paciência, pelos cafés e pelo companheirismo. Resta-me agradecer a todos, em especial ao André Pilastrri, Arthur, Luís Ferreira, Luís Matos e Rui Ribeiro pela facilidade e alegria com que partilham o vosso conhecimento, ao Hugo e aos “Luís Fernando” porque um local de trabalho deve sempre ter colegas como vocês. Deixar também uma palavra de agradecimento ao Rui Sousa, da Somelos, e ao João Paulo Silva, da Fluxodata, pela disponibilidade e pelo apoio que prestaram no desenrolar de todo o projeto.

Aos meus companheiros de caminhada: Campos, Cunha, Filipa, Helena, Hugo, Joana, Mafalda, Mariana Almeida, Mariana Pinto, Nuno e Pedro agradecer-vos dizendo que serão sempre o meu orgulho, o meu *“Peito ilustre lusitano”*.

Aos meus fieis companheiros Amarantinos, que me acompanham desde que queria ser jogador de futebol, um sincero obrigado por todos os momentos que vivemos e que me ajudaram a moldar a pessoa que sou hoje. Sem vocês, *“Sem esta terra funda e fundo rio/ Que ergue as asas e sobe em claro voo;/ Sem estes ermos montes e arvoredos/ Eu não era o que sou.”*

“Eu não sei senão amar-te,/Nasci para te querer.”, é a citar Fernando Pessoa, que te quero dedicar este espaço, Mariana, pela alegria com que vives e me fazes viver, pela companhia e pela compreensão, por confiares em mim mais do que até eu próprio, por seres colega, amiga e namorada, e, sobretudo por parte desta conquista também se dever a ti, obrigado!

Aos meus pais e irmão, a quem tudo devo, e por mais do que ninguém me apoiarem e me darem o melhor exemplo, fazendo-me superar todos os dias, o mais puro e honesto obrigado! *“A nossa família sempre é assim, maior que a humanidade.”*

Do latim *“solitas”*, a lembrança grata de um momento passado é por nós reconhecido como *“saudade”* e é o sentimento que perdurará.

“Tenho em mim todos os sonhos do mundo.”

Fernando Pessoa

This work is supported by the European Structural and Investment Funds in the FEDER component, through the Operational Competitiveness and Internationalization Programme (COMPETE 2020) [Project PPC4.0 - Production Planning Control 4.0; Funding Reference: POCI-01-0247-FEDER-069803].

DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho acadêmico 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.

RESUMO

Atualmente, as empresas do setor têxtil continuam a ter métodos de planeamento e controlo da produção pouco evoluídos e robustos. Algo que com a evolução do setor tecnológico e dos volumes crescentes de dados que as empresas da indústria têm, devido aos paradigmas da Indústria 4.0, não se justifica. Deste modo, com o intuito de melhorar o planeamento e controlo da produção da Somelos, foi feita uma parceria de investigação e desenvolvimento entre a Somelos, a Fluxodata, o Citeve e o Centro de Computação Gráfica (CCG) que deu origem ao projeto *PPC4.0*, onde a presente dissertação se enquadra.

Na fase experimental do projeto, foi adotada a metodologia *Cross Industry Standard Process for Data Mining (CRISP-DM)* para cada requisito, e foi dada maior atenção à fase de processamento de dados, utilizando para esse feito ferramentas de *Automated Machine Learning (AutoML)* na fase de modelação. Após a aplicação dos diversos cenários, os resultados obtidos foram debatidos com os profissionais do setor têxtil da Somelos que devolveram um *feedback* positivo.

Através da implementação de técnicas de *machine learning*, foi desenvolvida uma solução capaz de prever o rendimento dos artigos, prever os defeitos nos teares que causam a paragem das máquinas, a data de entrega de determinada matéria-prima e a duração dos tempos de operação. Sendo que, após a fase experimental, foi desenvolvido um protótipo adaptável ao surgimento de novos dados e que permite fazer a monitorização dos modelos, recorrendo a tecnologias como o *Kedro*, o *Mlflow* e o *Grafana*. Deste modo, é expectável que o protótipo desenvolvido suporte o planeamento e controlo da produção na Somelos.

Palavras-chave: *Machine Learning*, Análises Preditivas, Planeamento e Controlo da Produção, Previsão no Setor Têxtil

ABSTRACT

Currently, companies in the textile sector still have production planning and control methods that are poorly developed and robust. Something that with the evolution of the technology sector and the increasing volumes of data that companies in the industry have, due to the paradigms of Industry 4.0, is not justified. Thus, to improve Somelos production planning and control, a research and development partnership was made between Somelos, Fluxodata, Citeve and CCG that originated the *Production Planning and Control 4.0 (PPC4.0)*, where this dissertation came from.

During the project experimental phase, the *CRISP-DM* approach was used for each requirement, and more emphasis was placed on the data processing phase, with *AutoML* tools being used in the modeling phase for this reason. Following the implementation of numerous scenarios, the outcomes were discussed with Somelos textile professionals, who gave supportive remarks.

Through the implementation of machine learning techniques it was developed a solution capable of predicting the yield of the articles, predicting the defects in the looms that cause the machines to stop, the delivery date of certain raw materials, and the duration of the operation times. After the experimental phase, a prototype was developed adaptable to new data and allows the models to be monitored, using technologies such as Kedro, Mlflow and Grafana. As a result, it is envisaged that the prototype will help Somelos with production planning and control.

Keywords: Machine Learning, Predictive Analytics, Production Planning and Control, Forecast in the Textile Sector

ÍNDICE

Agradecimentos	7
Resumo	8
Abstract	9
Lista de Figuras	12
Lista de Tabelas	14
Lista de Abreviaturas, Siglas e Acrónimos	16
1 INTRODUÇÃO	1
1.1 Enquadramento e Motivação	1
1.2 Objetivos e Resultados Esperados	1
1.3 Organização do Documento	2
1.4 Abordagem Metodológica	2
2 REVISÃO DE LITERATURA	5
2.1 Estratégia de Pesquisa Bibliográfica	5
2.2 <i>Business Analytics</i>	5
2.2.1 <i>Descriptive Analytics</i>	6
2.2.2 <i>Diagnostic Analytics</i>	6
2.2.3 <i>Predictive Analytics</i>	7
2.2.4 <i>Prescriptive Analytics</i>	7
2.3 Inteligência Artificial	8
2.4 <i>Machine Learning</i>	9
2.4.1 Aprendizagem Supervisionada	9
2.4.2 <i>Automated Machine Learning</i>	10
2.4.3 Exemplos de algoritmos de <i>Machine Learning</i> :	11
2.5 <i>Deep Learning</i>	12
2.5.1 <i>Artificial Neural Network</i>	13
2.5.2 <i>AutoKeras</i>	13
2.6 Aplicação de <i>Machine Learning</i> no Setor Têxtil	14
2.7 Ferramentas Tecnológicas	19
3 CASO DE ESTUDO: PLANEAMENTO E CONTROLO DA PRODUÇÃO	24
3.1 Compreensão do Negócio	24
3.2 Compreensão e Preparação dos Dados	24
3.2.1 Requisito 1: Previsão do Rendimento dos Artigos	25
3.2.2 Requisito 2: Previsão de Defeitos nos Teares	26
3.2.3 Requisito 3: Previsão de Entrega de Matérias-Primas	29
3.2.4 Requisito 4: Previsão de Tempos de Operação (Urdissagem, Engomagem e Remetagem)	30
3.3 Modelação	31

3.4	Avaliação	32
3.4.1	Requisito 1: Previsão do Rendimento dos Artigos	33
3.4.2	Requisito 2: Previsão de Defeitos nos Teares	34
3.4.3	Requisito 3: Previsão de Entrega de Matérias-Primas	37
3.4.4	Requisito 4: Previsão de Tempos de Operação (Urdissagem, Engomagem e Remetagem)	38
4	PROTÓTIPO	40
5	CONCLUSÕES	43
5.1	Síntese do Trabalho Efetuado	43
5.2	Discussão	44
5.3	Trabalho Futuro	44
	Referências Bibliográficas	46
A	ANEXOS	54
A.1	Resultados das Previsões por <i>fold</i> em cada abordagem/ferramenta utilizada no requisito 2	54
A.2	Resultados das Previsões por <i>fold</i> em cada <i>target</i> utilizado no requisito 4	57
A.3	Screenshots do protótipo	57

LISTA DE FIGURAS

Figura 1	Modelo <i>CRISP-DM</i> . Adaptado de (Martínez-Plumed et al., 2021).	3
Figura 2	Modelo de Maturidade de Análises Analíticas de Dados. Adaptado de Schaap (2020).	6
Figura 3	Conceito de <i>IA</i> decomposto. Adaptado de Mondal (2020).	8
Figura 4	Tipos de Aprendizagem em <i>ML</i> . Adaptado de Shobha & Rangaswamy (2018).	9
Figura 5	Diagrama de venn representativo das relações entre <i>DL</i> , <i>ML</i> e <i>IA</i> . Adaptado de Bengio, Goodfellow, & Courville (2017).	12
Figura 6	Estrutura de um neurónio artificial. Adaptado de Neapolitan & Jiang (2018)	13
Figura 7	Arquitetura e funcionamento da ferramenta AutoKeras. Adaptado de Jin, Song, & Hu (2019).	14
Figura 8	Histograma do atributo relativo ao rendimento dos artigos	25
Figura 9	Distribuição do número de quebras de teia (esquerda), quebras de trama (meio) e rebentadela de fios (direita)	28
Figura 10	Distribuição do número de quebras de teia (esquerda), quebras de trama (meio) e rebentadela de fios (direita), após a aplicação da função logarítmica	28
Figura 11	Histograma do atributo relativo ao tempo de entrega de matérias-primas sem a aplicação da função logarítmica (esquerda) e com a aplicação da função logarítmica (direita)	29
Figura 12	Distribuição do tempo de urdissagem (esquerda), engomagem (meio) e remetagem (direita)	31
Figura 13	Exemplificação do processo de <i>cross-validation</i> . Adaptado de Sontakke, Lohokare, Dani, & Shivagaje (2019).	31
Figura 14	Código <i>Python</i> da aplicação da <i>cross-validation</i>	32
Figura 15	Curva REC e Reta de Regressão para o Fold7 do requisito 1	34
Figura 16	Curva REC para a quebra de teia (esquerda), quebra de trama (meio) e rebentadela de fios (direita).	35
Figura 17	Reta de Regressão do melhor modelo para a quebra de teia (esquerda), quebra de trama (meio) e rebentadela de fios (direita).	36
Figura 18	Conhecimento retirado através do <i>SA XAI</i> aplicado no modelo <i>GBM</i> : Importância das variáveis de entrada (esquerda) e curvas <i>VEC</i> dos atributos mais relevantes (direita).	36
Figura 19	Curva REC e Reta de Regressão para o Fold7 do requisito 1	37
Figura 20	Curva REC para o tempo da urdissagem (esquerda), da engomagem (meio) e da remetagem (direita).	38

Figura 21	Reta de Regressão do melhor modelo para o tempo da urdissagem (esquerda), da engomagem (meio) e da remetagem (direita).	39
Figura 22	Arquitetura tecnológica do protótipo	40
Figura 23	Pipelines do <i>Kedro</i>	41
Figura 24	Modelos registados no <i>Mlflow</i>	41
Figura 25	Visualização de <i>pipelines</i> no <i>Grafana</i>	42
Figura 26	Transição de estados no <i>Mlflow</i>	57
Figura 27	Selecionar modelos para comparar no <i>Mlflow</i>	58
Figura 28	Comparação de modelos no <i>Mlflow</i>	58

LISTA DE TABELAS

Tabela 1	Resumo das Técnicas de <i>Machine Learning</i> aplicadas ao setor têxtil	19
Tabela 2	Módulos Python utilizados neste trabalho.	20
Tabela 3	Resumo dos atributos relativos ao Requisito 1: Previsão do Rendimento dos Artigos	26
Tabela 4	Resumo dos atributos relativos ao Requisito 2: Previsão de Defeitos nos Teares	27
Tabela 5	Resumo dos atributos relativos ao Requisito 3: Previsão de Entrega de Matérias-Primas	29
Tabela 6	Resumo dos atributos relativos ao Requisito 4: Previsão de Tempos de Operação	30
Tabela 7	Resultados obtidos no Requisito 1: Previsão do Rendimento dos Artigos	33
Tabela 8	Resultados obtidos no Requisito 2: Previsão de Defeitos nos Teares	35
Tabela 9	Resultados obtidos no Requisito 3: Previsão de Entrega de Matérias-Primas	37
Tabela 10	Resultados obtidos no Requisito 4: Previsão de Tempos de Operação (Urdissagem, Engomagem e Remetagem)	38
Tabela 11	Resultados obtidos por <i>fold</i> para a abordagem A através da ferramenta H2O AutoML	54
Tabela 12	Resultados obtidos por <i>fold</i> para a abordagem B através da ferramenta H2O AutoML	54
Tabela 13	Resultados obtidos por <i>fold</i> para a abordagem A através da ferramenta AutoGluon	55
Tabela 14	Resultados obtidos por <i>fold</i> para a abordagem B através da ferramenta AutoGluon	55
Tabela 15	Resultados obtidos por <i>fold</i> para a abordagem A através da rede manual DFFN	55
Tabela 16	Resultados obtidos por <i>fold</i> para a abordagem B através da rede manual DFFN	56
Tabela 17	Resultados obtidos por <i>fold</i> para a abordagem A através da ferramenta AutoKeras	56
Tabela 18	Resultados obtidos por <i>fold</i> para a abordagem B através da ferramenta AutoKeras	56
Tabela 19	Resultados obtidos por <i>fold</i> para o Requisito 4: Previsão de Tempos de Operação (Urdissagem, Engomagem e Remetagem)	57

LISTA DE ABREVIATURAS, SIGLAS E ACRÓNIMOS

ANN	<i>Artificial Neural Network.</i>
API	<i>Application Programming Interface.</i>
AR	<i>Auto-Regression.</i>
ARIMA	<i>Autoregressive Integrated Moving Average.</i>
AutoML	<i>Automated Machine Learning.</i>
BA	<i>Business Analytics.</i>
BFG	<i>Broydon-Fletcher-Goldfarb-Shanno.</i>
BR	<i>Bayesian Regularization.</i>
CCG	<i>Centro de Computação Gráfica.</i>
CGF	<i>Fletcher-Reeves.</i>
CoxPH	<i>Cox Proportional Hazards.</i>
CRISP-DM	<i>CRoss Industry Standard Process for Data Mining.</i>
DFFN	<i>Deep FeedForward Neural Network.</i>
DL	<i>Deep Learning.</i>
DM	<i>Data Mining.</i>
DRF	<i>Distributed Random Forest.</i>
DS	<i>Data Science.</i>
DT	<i>Decision Tree.</i>
DW	<i>Data Warehouse.</i>
GAM	<i>Generalized Additive Models.</i>
GBM	<i>Gradient Boosting Machine.</i>
GEP	<i>Gene Expression Programming.</i>
GLM	<i>Generalized Linear Model.</i>
GNN	<i>Graph Neural Network.</i>
IA	<i>Inteligência Artificial.</i>
I&D	<i>Investigação e Desenvolvimento.</i>
IDF	<i>Inverse Document Frequency.</i>
IOT	<i>Internet of Things.</i>
ITV	<i>Indústria Têxtil e Vestuário.</i>
LM	<i>Levenber-Marquardt.</i>

LR	<i>Linear Regression.</i>
LSTM	<i>Long Short-Term Memory Recurrent Neural Network.</i>
MA	<i>Moving Average.</i>
MAE	<i>Mean Absolute Error.</i>
ML	<i>Machine Learning.</i>
MLP	<i>Multilayer Perceptron.</i>
MSE	<i>Mean Squared Error.</i>
NAS	<i>Neural Architecture Structure.</i>
NMAE	<i>Normalized Mean Absolute Error.</i>
NN	<i>Neural Network.</i>
PCA	<i>Principal Component Analysis.</i>
PCP	<i>Percentage Categorical Pruned.</i>
PPC	<i>Production Planning and Control.</i>
PPC4.0	<i>Production Planning and Control 4.0.</i>
PSO	<i>Particle Swarm Optimization.</i>
R^2	<i>R-squared ou Coefficient of Determination.</i>
RBF	<i>Radial Basis Functions.</i>
REC	<i>Regression Error Characteristic.</i>
RF	<i>Random Forest.</i>
RMS	<i>Root Mean Squared.</i>
RMSE	<i>Root Mean Squared Error.</i>
SA XAI	<i>Sensitivity Analysis eXplainable Artificial Intelligence.</i>
S.E.	<i>Stacked Ensemble.</i>
SVM	<i>Support Vector Machine.</i>
UM	<i>Universidade do Minho.</i>
VEC	<i>Variable Effect Characteristic.</i>

INTRODUÇÃO

1.1 ENQUADRAMENTO E MOTIVAÇÃO

A Indústria Têxtil e de Vestuário (*ITV*) é uma das mais antigas indústrias portuguesas e mantém-se como um dos maiores e mais importantes setores empresariais do país. Em 2016, a *ITV* empregava 135521 pessoas e atingiu os 7362 milhões de euros de volume de negócios, representando 4% do PIB nacional (DGAE, 2018). No sentido de continuar a progressão que a *ITV* tem vindo a demonstrar, urge a adesão a sistemas de informação flexíveis, que promovam a digitalização e tenham em perspetiva a Indústria 4.0 e os seus pilares fundamentais: *Big Data*, *Advanced Analytics*, *Cloud Computing* e *Internet of Things (IOT)* (Pinheiro, 2016).

Deste modo, a presente dissertação enquadrou-se no projeto *PPC4.0* e foi desenvolvida no CCG, com o apoio do Citeve e da Universidade do Minho (UM) como parceiros de *I&D* e com a participação da Fluxodata e da Somelos como parceiros empresariais. Por conseguinte, esta centrou-se na produção de uma solução inovadora de planeamento e controlo da produção, com o foco inicial na operação de tecelagem.

Assim, pretendeu-se desenvolver uma nova geração de ferramentas de suporte à atividade de planeamento e controlo da produção industrial, que seguisse os padrões da união europeia no que toca à reindustrialização e transformação digital, de modo a usufruir do potencial tecnológico que a Indústria 4.0 acarreta. A dissertação surgiu da tarefa “T4.3 - *I&D* Soluções de data science para planeamento e controlo da produção” que fazia parte da etapa “A4 - *I&D* *PPC4.0*”, onde se expectava que, através de técnicas de *Machine Learning (ML)*, fossem aplicados modelos e/ou algoritmos capazes de fazer análises preditivas por forma a melhorar a capacidade produtiva da Somelos, e, ainda, que estes modelos fossem flexíveis ao aparecimento de novos dados e ao seu consequente refrescamento.

1.2 OBJETIVOS E RESULTADOS ESPERADOS

Tendo em conta que a presente dissertação se inseriu no projeto *PPC4.0*, que tinha como principal objetivo o desenvolvimento de uma nova geração de ferramentas de suporte à atividade de planeamento e controlo da produção industrial, tirando partido da ciência de dados para aumentar a eficiência e sustentabilidade da manufatura, através de previsões de operações mais fiáveis e robustas, os objetivos e resultados esperados estão diretamente relacionados com os do projeto. Nesta senda, um conjunto de objetivos foi alvo de escrutínio nesta dissertação, sendo eles:

- Prever o rendimento dos artigos (Representa a percentagem de tecido introduzido que foi aproveitado para o artigo final, por exemplo, um tecido de 100 metros que origine um artigo de 85 metros, corresponde a um rendimento de 85%.);
- Prever a ocorrência de defeitos na tecelagem;
- Prever o tempo de entrega de determinada matéria-prima;
- Prever a duração das operações (Urdissagem, Engomagem e Remetagem);
- Transmitir os resultados das previsões através de uma *API*;
- Monitorizar os modelos de previsão.

No que concerne ao sucesso do projeto, este esteve, durante a sua execução, dependente do fornecimento e da qualidade dos dados, bem como do sucesso da implementação dos modelos de *ML*, e, conseqüentemente dependente da boa aplicabilidade das etapas do *CRoss Industry Standard Process for Data Mining (CRISP-DM)*, metodologia adotada para o desenvolvimento da presente dissertação.

1.3 ORGANIZAÇÃO DO DOCUMENTO

A presente dissertação incorpora cinco capítulos distintos na sua estrutura. O primeiro, relativo à introdução, apresenta o enquadramento e a motivação do projeto, os objetivos e resultados esperados da execução do mesmo, bem como a metodologia adotada no projeto, ou seja, o *CRISP-DM*. O segundo, que diz respeito à revisão de literatura, introduz os conteúdos de relevo do desenrolar do projeto. Assim sendo, inicia com a estratégia de pesquisa bibliográfica, introduzindo e explicando tendo em conta a literatura atual os conceitos de *Business Analytics (BA)*, Inteligência Artificial (*IA*), *ML* e de *Deep Learning (DL)*, bem como os seus constituintes. Ainda neste capítulo, são abordados outros trabalhos de relevo sobre a aplicação de *ML* no setor têxtil e é efetuado um estudo das ferramentas tecnológicas que serão utilizados no decorrer da dissertação. O terceiro capítulo apresenta os resultados obtidos nas experiências à luz da metodologia adotada, apresentando, nesta perspetiva, a compreensão de negócio, a compreensão e preparação dos dados, a modelação e a avaliação. O capítulo quatro, ilustra o protótipo desenvolvido para posterior implementação na organização. Por sua vez, o quinto e último capítulo apresenta as conclusões, com uma síntese do trabalho efetuado e uma secção para expor o trabalho futuro.

1.4 ABORDAGEM METODOLÓGICA

Metodologia é o quadro filosófico em que a investigação é conduzida ou a base sobre a qual a investigação se baseia (Brown, 2006). Deste modo, e tendo em conta a natureza prática desta dissertação, foi utilizada a metodologia *CRISP-DM* para conduzir a investigação, por forma a garantir que todos os conteúdos e técnicas fossem abordados e implementados de forma correta e na fase certa.

O *CRISP-DM* foi desenvolvido partindo de várias tentativas anteriores em definir metodologias de descoberta de conhecimento (Wirth, 2000). A metodologia atual fornece uma visão geral do ciclo de vida de um projeto de *Data Mining* (*DM*), expõe e explica as fases de um projeto, bem como as suas tarefas e as relações entre elas. Refere, ainda, que dependendo dos objetivos, do interesse do utilizador e, sobretudo, dos dados, podem existir relações entre quaisquer tarefas de *DM* (Chapman et al., 2000).

Na Figura 1 encontram-se as 6 fases delineadas nesta metodologia e, este é o modelo de referência na execução de projetos de *ML*. Em cada uma das fases está explícito o caminho a ser seguido, de notar, no entanto, que a utilização desta tecnologia deve ser contextualizada perante as especificidades de cada projeto.

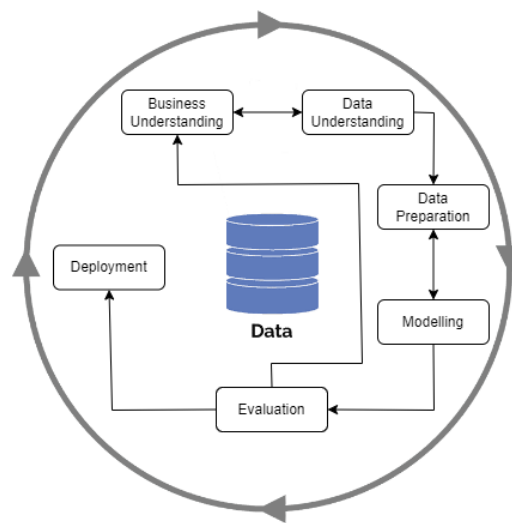


Figura 1: Modelo *CRISP-DM*. Adaptado de (Martínez-Plumed et al., 2021).

1. *Business Understanding* (Compreensão do Negócio): Esta fase direciona os seus esforços na compreensão dos objetivos e requisitos do projeto na perspetiva empresarial. Convertendo, posteriormente, o conhecimento obtido na formulação de um problema de *DM* e num plano preliminar concebido para alcançar os objetivos.
2. *Data Understanding* (Compreensão dos Dados): A fase de compreensão inicia-se com a recolha dos dados e prossegue com um estudo inicial para obter uma perspetiva da qualidade dos mesmos e encontrar padrões interessantes que possam formar hipóteses de informação oculta.
3. *Data Preparation* (Preparação dos Dados): Na fase de preparação dos dados, partindo dos dados iniciais é feito um conjunto de atividades, de modo a construir o conjunto de dados final. Estas atividades, por norma, incluem a seleção de tabelas, registos e atributos, bem como a transformação e a limpeza dos dados.
4. *Modeling* (Modelação): Nesta fase, são selecionadas e aplicadas as técnicas de modelação e são calibrados os seus parâmetros. Tendo em conta que existem várias técnicas para o mesmo tipo de problema, e que algumas delas têm requisitos específicos sobre o formatos dos dados, por vezes será necessário voltar à fase de preparação dos dados.

5. *Evaluation* (Avaliação): Na fase de avaliação é expectável que o modelo construído tenha alta qualidade, porém, antes de se proceder à implementação final é necessário que seja feita, de forma exaustiva, uma avaliação do mesmo, revendo as etapas executadas, a fim de se perceber se o modelo responde às expectativas empresariais previamente estipuladas. Um ponto fulcral desta fase é perceber se algum objetivo chave ficou por responder.
6. *Deployment* (Implementação): Criar o modelo por si só não representa o fim do projeto, dado que o conhecimento obtido terá de ser organizado e apresentado de forma a que o cliente o possa utilizar e, realmente, tirar valor do que foi desenvolvido. Para isso, na fase de implementação, são aplicados, frequentemente, modelos “ao vivo” dentro dos processos de tomada de decisão de determinada organização. Deste modo, esta fase, poderá ser executada de forma simples, através de um relatório ou, de forma mais complexa, envolvendo a implementação de um processo de *DM* em toda a organização. É importante que o cliente compreenda as ações necessárias para utilizar de forma efetiva os modelos criados.

REVISÃO DE LITERATURA

O capítulo de revisão de literatura tem como objetivo contextualizar a dissertação à luz da literatura atual. Posto isto, inicia-se com a estratégia de pesquisa bibliográfica adotada, passando pela introdução dos termos de *BA*, *IA*, *ML* e *DL*. Após a introdução dos termos, são apresentados trabalhos de outros autores que aplicaram *ML* no setor têxtil, seguido de uma tabela com as ferramentas tecnológicas a utilizar no desenrolar da dissertação.

2.1 ESTRATÉGIA DE PESQUISA BIBLIOGRÁFICA

Precedente à escrita da presente dissertação, foi delineada uma estratégia de pesquisa para que a base bibliográfica em que se assenta este projeto seja relevante e atual, contendo artigos científicos, artigos de conferência, livros e dissertações que validem a informação exposta. Deste modo, foram priorizados artigos publicados a partir de 2017, e os motores de busca preferenciais foram o *Google Scholar*, o *IEEE Explore*, o *Science Direct* e o *Elsevier*.

Numa primeira fase, a pesquisa incidiu sobre as concepções de relevo para a presente dissertação, através da pesquisa de algumas palavras-chave como: “*Business Analytics*”, “*Machine Learning*”, “*Automated Machine Learning*” e “*Predictive Analytics*”. Por sua vez, a pesquisa incidiu em conceitos cujo relevo incidia mais no *PPC*, através de palavras-chave como: “*Production Planning and Control*”, “*Industry 4.0*”, “*Machine Learning*” e “*Machine downtime*”.

Recolhidos os artigos considerados relevantes, estes foram estudados de forma complexa tendo em vista a aplicação do conhecimento adquirido, e, se necessário, a replicação futura de alguns procedimentos e métodos, servindo, também, como fundamentação para a presente revisão de literatura.

2.2 BUSINESS ANALYTICS

BA pode ser entendido como um conjunto de disciplinas e tecnologias interligadas, que, através de modelos estatísticos e análise de dados, auxiliam a tomada de decisão de gestão e, deste modo, resolvem problemas empresariais, efetuando uma investigação iterativa contínua do historial empresarial (Lutkevich & Burns, 2021). O sucesso da *BA* está dependente da qualidade dos dados, da compreensão do negócio e das tecnologias que os analistas detêm, bem como da forma como os dados são utilizados para trazer valor e criar condições para que a tomada de decisão seja o mais fundamentada possível. Em 2011, o *Gartner Group* apresentou

um modelo de dados com maturidade ascendente, que inclui as quatro principais fases de análise, bem como as questões que visa responder, tal como está apresentado na Figura 2.

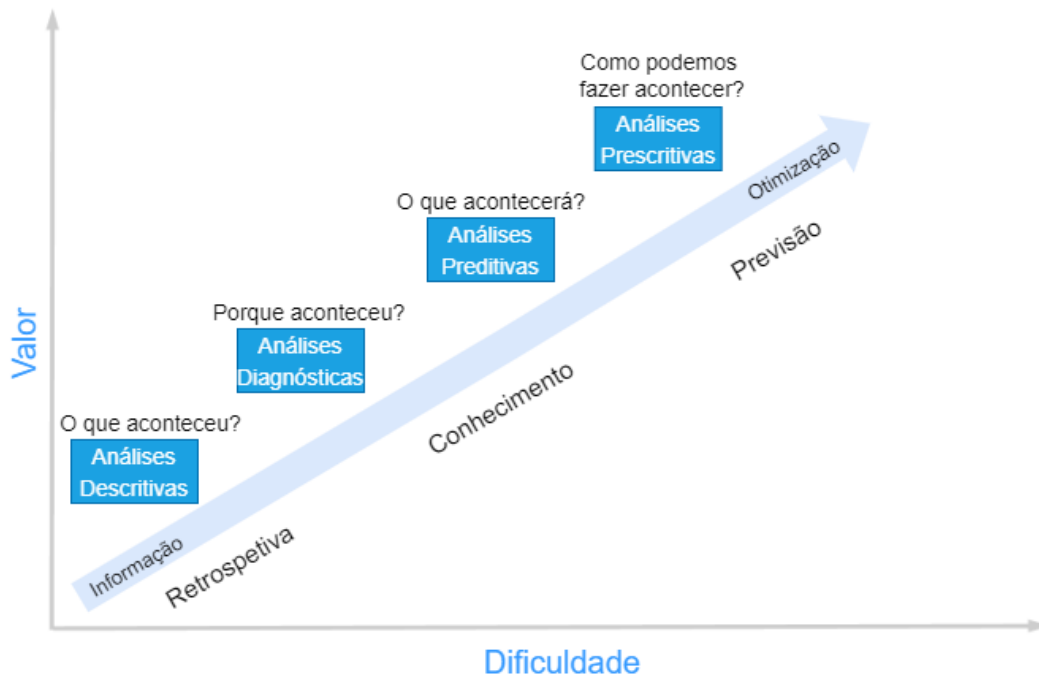


Figura 2: Modelo de Maturidade de Análises Analíticas de Dados. Adaptado de Schaap (2020).

2.2.1 *Descriptive Analytics*

A primeira etapa representada na Figura 2, *Descriptive Analytics*, é a análise mais comum e é composta por um conjunto de técnicas com o intuito de categorizar, caracterizar, consolidar e classificar os dados, para que gerem informação útil e que possa trazer valor para a organização. Este tipo de análise é efetuada em dados históricos e apresenta a informação em gráficos e/ou relatórios, para que os gestores consigam entender qual o impacto de determinada decisão, bem como analisar o desempenho do negócio através da padronização dos dados, procurando encontrar problemas ou oportunidades de negócio (Evans & Lindner, 2013). Deste modo, estas análises têm a capacidade de quantificar eventos e são o primeiro passo da transformação de dados em algum tipo de informação. A análise descritiva pode ajudar nas tarefas de gestão da saúde da população, identificando, por exemplo, quantos pacientes vivem com diabetes, bem como a aferir os resultados em relação às expectativas do governo, ou a identificar áreas para melhorar as medidas de qualidade clínica ou outros aspetos dos cuidados de saúde (El Morr & Ali-Hassan, 2019).

2.2.2 *Diagnostic Analytics*

Tal como acontece com as análises descritivas, este tipo de análise avalia os processos e causas, focando-se menos na ocorrência. De acordo com Vesset (2018), as *diagnostic analytics* apresentam três funções principais:

- Identificar Anomalias: com base nos resultados obtidos através da análise, devem ser identificadas as áreas que requerem um estudo mais aprofundado, dado que apresentam questões que não poderão ser respondidas apenas analisando os dados.
- Pesquisar a análise: identificar as fontes de dados que expliquem as anomalias. Por vezes, nesta etapa é necessário verificar a existência de padrões fora dos conjuntos de dados existentes, bem como o uso de dados de fontes externas para identificar correlações e determinar a probabilidade da sua natureza causal.
- Determinar relações causais: as relações ocultas são descobertas através da observação de eventos que podem ter resultado nas anomalias identificadas. A teoria da probabilidade, análise de regressão, filtragem e análises de dados em séries temporais podem ser úteis para descobrir histórias ocultas nos dados. Assim, para que os passos precedentemente mencionados sejam executados, aplicam-se técnicas de *DM* (Vesset, 2018).

2.2.3 Predictive Analytics

No que concerne às *predictive analysis*, é feita uma análise a dados históricos para que, com base nestes, se possam detetar padrões ou relações entre os dados, para extrapolar estas relações no tempo e, conseqüentemente, prever resultados futuros com maior assertividade. Este tipo de análise pode identificar anomalias, reconhecer oportunidades e, utilizando técnicas de *ML*, determinar padrões ocultos em grandes quantidades de dados, de modo a segmentar e agrupar dados em conjuntos coerentes para que a previsão de determinado comportamento seja o mais realista possível (Evans & Lindner, 2013). Assim, a escolha do algoritmo a utilizar nas análises preditivas depende do problema ou questão que estamos a tratar. Para a segmentação de pacientes, por exemplo, seriam necessários algoritmos de *clustering*; para um sistema de recomendação, a necessidade é de um algoritmo de classificação, e para prever o próximo resultado de eventos orientados pelo tempo, seria usado um algoritmo de regressão (El Morr & Ali-Hassan, 2019).

2.2.4 Prescriptive Analytics

A última etapa, *Prescriptive Analytics*, emprega a otimização para identificar as melhores soluções no que toca a minimizar/maximizar determinado objetivo. É uma técnica utilizada em variadas atividades profissionais, tais como marketing e finanças. Assim, combina análises matemáticas e estatísticas com otimização, tendo em consideração a incerteza dos dados para prever resultados e auxiliar a tomada de decisão (Evans & Lindner, 2013). Por conseguinte, os modelos analíticos prescritivos acrescentam à análise preditiva a capacidade não só de prever, mas também de explicar porque é que um evento aconteceu através de um conjunto de regras fáceis de interpretar, permitindo-nos agir com base no evento, utilizando essas regras (El Morr & Ali-Hassan, 2019).

2.3 INTELIGÊNCIA ARTIFICIAL

Segundo McCarthy (1989), podemos definir IA como “A ciência e engenharia de criar máquinas inteligentes, especialmente programas de computador inteligentes”. A origem da IA remete para 1950, quando *Alan Turing*, questionou sobre a capacidade das máquinas pensarem “*Can machine think?*” e propôs o “*Turing Test*” para avaliar a inteligência de uma máquina (Turing, 2009).

Tendo em conta que são vários os campos que constituem a IA: filosofia, matemática, neurociência, psicologia, engenharia informática, entre outros, fica difícil encontrar uma definição consensual para IA, na medida em que todos estes campos têm diferentes metodologias e paradigmas e, por conseguinte, não é fácil formar uma linguagem e compreensão comum a todos os campos (Ekmekci & Arda, 2020).

Mondal (2020) decompôs o conceito de IA em 5 componentes: *Natural language processing*, *Machine Learning*, *Decision Making*, *Computer Vision* e *Responding*, tal como ilustrado na Figura 3.

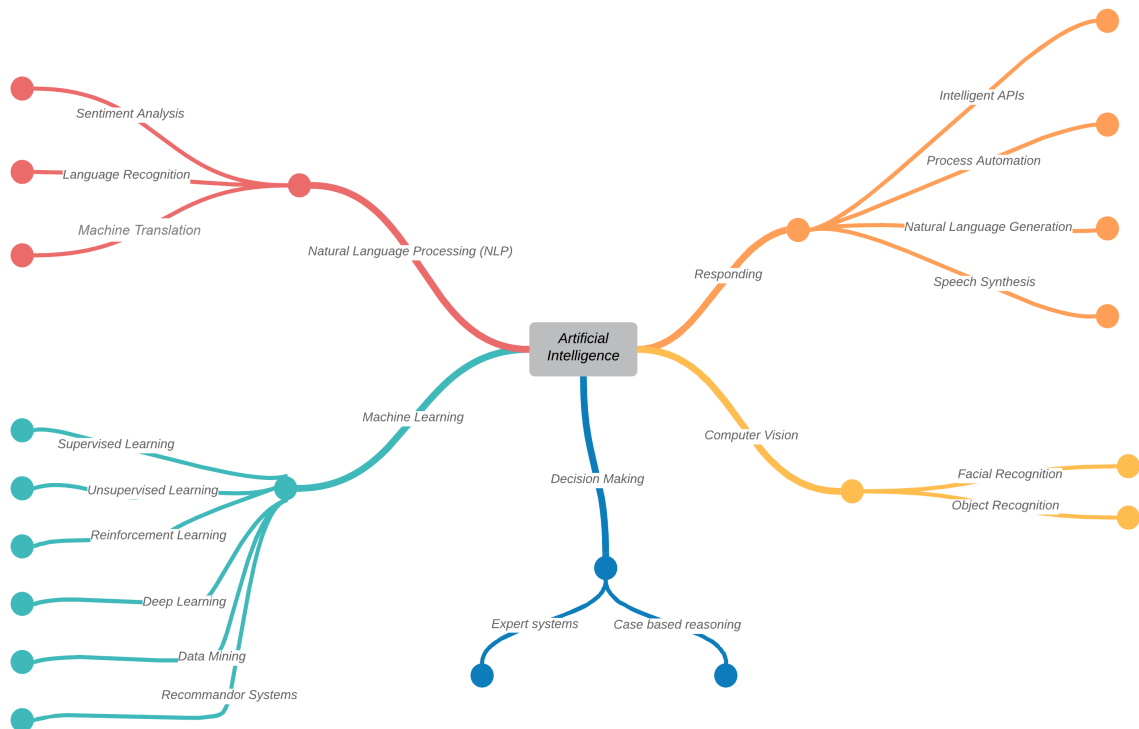


Figura 3: Conceito de IA decomposto. Adaptado de Mondal (2020).

2.4 MACHINE LEARNING

ML é um ramo, em evolução, dos algoritmos computacionais que são concebidos para recriar a inteligência humana, utilizando, assim, dados passados para aprender e melhorar determinado objetivo. Assim, o principal desafio é que os programas sejam capazes de aprender de forma automática e de reconhecer padrões para poderem tomar decisões com base nos dados fornecidos (El Naqa & Murphy, 2015). Nos últimos anos, várias aplicações de *ML* foram concebidas, por exemplo: modelos para a deteção de fraude em transações bancárias, sistemas de filtragem de informação que aprendem com as preferências de cada utilizador, veículos autónomos que aprendem a conduzir em autoestradas, entre outros.

Podemos subdividir o conceito de *ML* pelo tipo de aprendizagem em: Aprendizagem Supervisionada, Aprendizagem Não Supervisionada e Aprendizagem por Reforço, tal como está representado na Figura 4. A presente dissertação assume uma Aprendizagem Supervisionada.

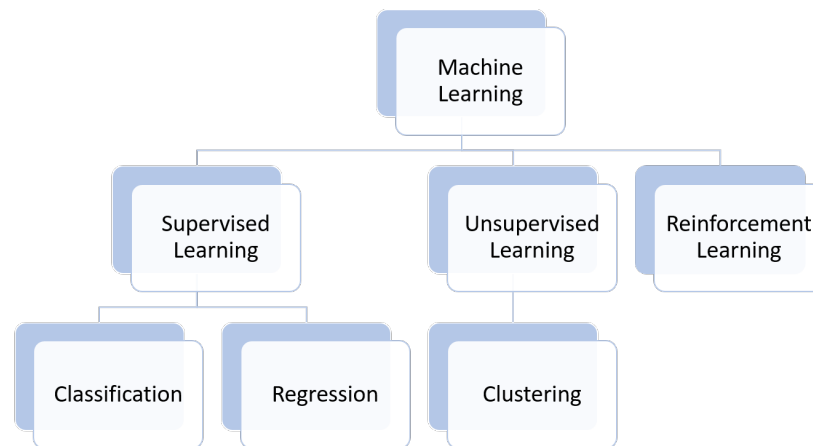


Figura 4: Tipos de Aprendizagem em *ML*. Adaptado de Shobha & Rangaswamy (2018).

2.4.1 Aprendizagem Supervisionada

Modelo de aprendizagem concebido para fazer previsões partindo de um *input* imprevisível, assim, dado um conjunto de dados de entrada, bem como as suas respostas é possível treinar o modelo (Regressão/Classificação) para que consiga prever a resposta pretendida aquando de um novo conjunto de dados. Desta feita, a aprendizagem supervisionada utiliza algoritmos de classificação e técnicas de regressão para desenvolver modelos preditivos. Por um lado, as técnicas de classificação preveem respostas discretas, recomendando-se que os dados sejam categorizados, etiquetados, ou separados em grupos específicos. Por outro lado, as técnicas de regressão preveem respostas contínuas. Uma regressão linear tem como objetivo modelar a relação entre duas ou mais variáveis, ajustando a equação linear aos dados observados (Shobha & Rangaswamy, 2018).

2.4.2 Automated Machine Learning

Nos últimos anos, a investigação e aplicação de *ML* tem vindo a aumentar e muito se deve a um conjunto de avanços tecnológicos que se têm vindo a notar, porém, o desempenho destes métodos continua muito relacionado com um conjunto de decisões que têm de ser tomadas aquando da implementação dos modelos, constituindo assim, um entrave aos novos desenvolvedores (Hutter et al., 2019).

Dado que a criação de modelos de *ML* é uma tarefa complexa e que, de forma iterativa, exige um procedimento de tentativa/erro (Zöller & Huber, 2021), que pode ser orientado pelo conhecimento especializado do analista ou heurístico (Gibert et al., 2016) é necessário obter um sólido conhecimento e compreensão dos tipos de algoritmos de *ML* e de como se podem ajustar os seus hiperparâmetros (Maher & Sakr, 2019).

Sendo assim, e no sentido de automatizar este processo desafiador e demorado surgem os sistemas de *AutoML*. Estes sistemas inteligentes aumentam a acessibilidade e a escalabilidade de variadas aplicações de *ML*. Assim, o *AutoML* liberta os utilizadores para aplicarem conhecimentos noutras componentes importantes, automatizando o processo de escolha do melhor modelo e dos seus hiperparâmetros (Le, Fu, & Moore, 2020).

Neste momento, já existem variadas ferramentas de *AutoML*, sendo que algumas das mais utilizadas são o *Auto-Sklearn*, *TPOT*, *H2O AutoML*, *Auto-Keras* e *AutoGluon*.

H2O AutoML

H2O é uma ferramenta *open-source*, escalável e distribuída. Utiliza compressão *in-memory* para processar grandes quantidades de linhas de dados, até mesmo num *cluster* de nível mais pequeno. Está desenhado para correr em modo *standalone*, no *Hadoop* ou num *cluster Spark*, através da aplicação *Sparkling Water*, fornecida pelo *H2O*. *H2O AutoML* é um algoritmo de aprendizagem supervisionada totalmente automatizado, implementado em *H2O*. Para facilitar a criação de fluxos de trabalho analíticos completos, a plataforma *H2O AutoML* está disponível em *Python*, *R*, *Java* e *Scala*, bem como através de uma interface *web*. Apesar deste algoritmo ser totalmente automatizado, muitas das suas configurações podem ser expostas, para que certos aspetos da etapa de modelação possam ser customizados (LeDell & Poirier, 2020).

Algoritmos usados e parâmetros:

O *H2O* inclui várias famílias de algoritmos de *ML*, bem como dois modelos de *Stacked Ensembles (S.E.)*, que podem ser utilizados em classificação e regressão, Ferreira, Pilastri, Sousa, Romano, & Cortez (2021). Estes estão organizados por tipo de aprendizagem. No que diz respeito à aprendizagem supervisionada são disponibilizados os seguintes modelos: *CoxPH*, *NN*, *DRF*, *GLM*, *GAM*, *GBM*, *Naïve Bayes Classifier*, *S.E.*, *SVM* e *XGBoost* (H2O.ai, 2021).

Para a aprendizagem não supervisionada são disponibilizados os seguintes modelos: *Generalized Low Rank Models*, *Isolation Forest*, *Extended Isolation Forest*, *K-Means Clustering* e *PCA* (H2O.ai, 2021).

AutoGluon

AutoGluon é uma biblioteca de *AutoML*, *open source*, que automatiza *DL* e *ML* para aplicações do mundo real envolvendo conjuntos de dados de imagem, texto e tabulares. A previsão tabular executa numerosos algoritmos de *ML* antes de devolver um *S.E.* com múltiplas camadas de diferentes modelos de *ML* (Ferreira et al., 2021). O *Ensemble* compreende os seguintes algoritmos: *LightGBM*, *CatBoost Boosted Trees*, *RF*, *Extra Trees*, *k-NN* e *Multiple Linear Regression*. O AutoGluon utiliza, também, uma arquitetura *DL* densa para o cenário *DL*, que emprega funções de ativação *ReLU*, regularização de *dropout* e camadas de normalização de lotes, bem como heurística para ajustar os tamanhos das camadas ocultas (Erickson et al., 2020).

2.4.3 Exemplos de algoritmos de Machine Learning:

- *AutoRegressive Integrated Moving Average*: O modelo *ARIMA* é baseado no modelo *Box-Jenkins* desenvolvido em 1960, que faz uso de valores passados para prever valores futuros de séries temporais (Singh et al., 2020). A abordagem de modelação *ARIMA* é caracterizada por apresentar três termos: p , d , q na qual p , representa a *Auto-Regression (AR)*, q referencia a *Moving Average (MA)* e d é a ordem de diferenciação necessária para tornar a série temporal estacionária. A *AR* é a regressão da variável em torno de si mesma, para prever a variável de interesse, correlacionado o padrão do período de tempo único com o seu historial de períodos de tempo, já a *MA* representa um modelo semelhante à regressão que utiliza os erros associados a uma previsão anterior para prever uma variável num tempo posterior (ArunKumar et al., 2021).
- *Decision Tree*: As *Decision Tree (DT)* são um modelo característico por apresentarem em cada árvore nós e ramos. Cada nó representa uma característica de uma categoria a ser classificada e, cada subconjunto, define um valor que pode ser tomado pelo nó e, deste modo, as observações sobre um item podem ser convertidas em conclusões com a análise da *DT* (Mathew et al., 2017). Tendo isto em conta, a simplicidade da análise de uma *DT*, bem como a sua precisão, têm sido fatores que justificam a elevada utilização e extensão deste modelo (Charbuty & Abdulazeez, 2021).
- *Linear Regression*: *Linear Regression (LR)* é o modelo estatístico mais simples e mais comum quando o objetivo é medir a relação entre variáveis contínuas (Hope, 2020). Para que um modelo de *LR* seja bem sucedido é de extrema importância compreender de que modo os dados poderão ser encaixados, quais os seus pontos fracos e fortes, bem como definir alternativas e escolher a altura certa para os adotar, caso seja necessário (Bonaccorso, 2017).
- *Random Forest*: O primeiro algoritmo de *Random Forest (RF)* data de 1995 e foi criado por *Ho*, sendo posteriormente extenso por *Breiman*, em 2001. Este é um método estatístico, baseado em árvores de decisão, tanto de classificação, como de regressão, que utiliza o método de *bootstrap resampling* para extrair múltiplas amostras provenientes da amostra original, modelando a árvore de decisão para cada amostra, com o intuito de combinar

as previsões das múltiplas árvores de decisão, de modo a obter a média dos resultados finais da previsão. Assim, *RF* tem sido amplamente utilizado e apresenta resultados muito satisfatórios tanto no que diz respeito à abordagem de classificação, como na da regressão (W. Zhang et al., 2021).

- *Support Vector Machine*: O poder de uma *Support Vector Machine (SVM)*, na abordagem de classificação, deriva da sua capacidade de aprender padrões de classificação de dados com precisão e reprodutibilidade equilibradas. Deste modo, o sucesso de uma *SVM* está diretamente relacionada com o sucesso da maximização da percentagem de categorias corretamente atribuídas pelo classificador, bem como com o sucesso da reprodutibilidade da mesma (Pisner & Schnyer, 2020). Já no que diz respeito à regressão, o objetivo da *SVM* é encontrar o melhor hiperplano de separação linear, tolerando um pequeno erro no ajuste dos dados (Cortez et al., 2009). Quando comparada com outras abordagens de regressão, a *SVM* apresenta duas claras vantagens: a primeira, prende-se com o uso de um *kernel* que pode fornecer uma forma eficiente de lidar com um problema de regressão não linear; o segundo benefício prende-se com o facto da *SVM* aprender um modelo para descrever a importância entre variáveis, ao contrário do que acontece noutros modelos de regressão onde é necessário assumir um modelo que pode não ser exato (F. Zhang & O'Donnell, 2020).

2.5 DEEP LEARNING

DL é um subcampo de *ML* que surge como uma ferramenta poderosa para resolver problemas, aprendendo através dos dados, impulsionando a sua aprendizagem pela disponibilidade de grandes conjuntos de dados, maior poder computacional, rápido armazenamento de dados, entre outros (Zuo et al., 2022). *DL* tem sido um bom método na resolução de problemas que persistiram durante muitos anos na comunidade de *IA*, como por exemplo no reconhecimento de imagens e no reconhecimento de fala (LeCun et al., 2015). A Figura 5 representa a relação entre *DL*, *ML* e *IA*, através de um diagrama de *venn*.

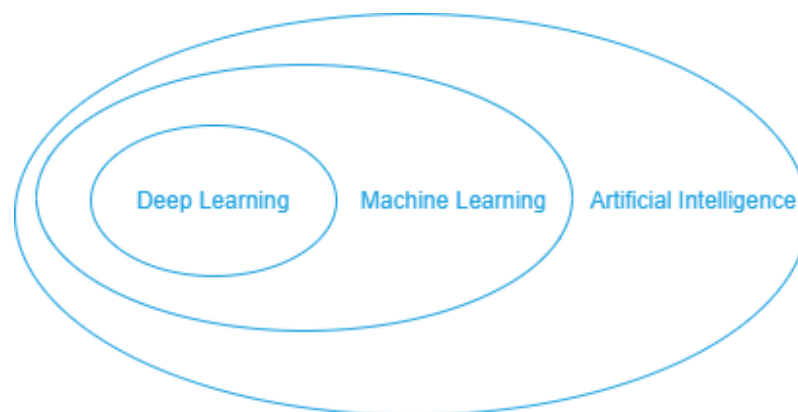


Figura 5: Diagrama de venn representativo das relações entre *DL*, *ML* e *IA*. Adaptado de Bengio et al. (2017).

2.5.1 Artificial Neural Network

Uma *Artificial Neural Network* (ANN) é uma rede inspirada pela funcionalidade dos cérebros humanos, onde milhões de neurónios interligados processam informações em paralelo. Esta consiste numa camada de entrada de neurónios, em várias camadas ocultas de neurónios, e numa camada final de neurónios de saída (S.-C. Wang, 2003). O neurónio artificial toma um vetor (a_1, a_2, \dots, a_k) como entrada e adiciona pesos $(p_0, p_1, p_2, \dots, p_k)$ para produzir uma soma ponderada. Deste modo, o neurónio aplica uma função de ativação à soma ponderada e produz o valor y de f , tal como representa a Figura 6 (Neapolitan & Jiang, 2018). Em suma, uma ANN é composta por um ou mais neurónios que interagem uns com os outros, sendo que a saída de um neurónio é a entrada de outro.

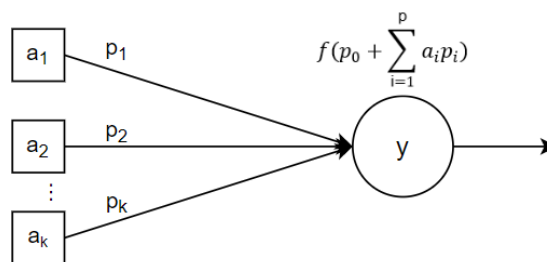


Figura 6: Estrutura de um neurónio artificial. Adaptado de Neapolitan & Jiang (2018)

2.5.2 AutoKeras

AutoKeras é uma biblioteca *Python*, baseada em *Keras* que funciona através de uma *Neural Architecture Structure* (NAS) automática, que se serve da *Bayesian Optimization* para configurar o número de camadas, o tipo de função de ativação utilizado, os valores de desistência, entre outros hiperparâmetros das redes de *DL*. A arquitetura adotada pelo *AutoKeras* é uma das suas principais vantagens dado que faz um uso eficiente de todos os seus componentes. O módulo de pesquisa (*Searcher* - 1) é executado em *CPU*, o processamento dos gráficos das redes neuronais (*Graph* - 2) é feita através da memória *RAM* para que o seu acesso seja mais rápido, enquanto que o módulo de treino (*Model Trainer* - 3) é executado em *GPU* e o armazenamento de modelos (*Storage* - 4) é feito de forma local (Jin et al., 2019). A Figura 7 representa a arquitetura descrita.

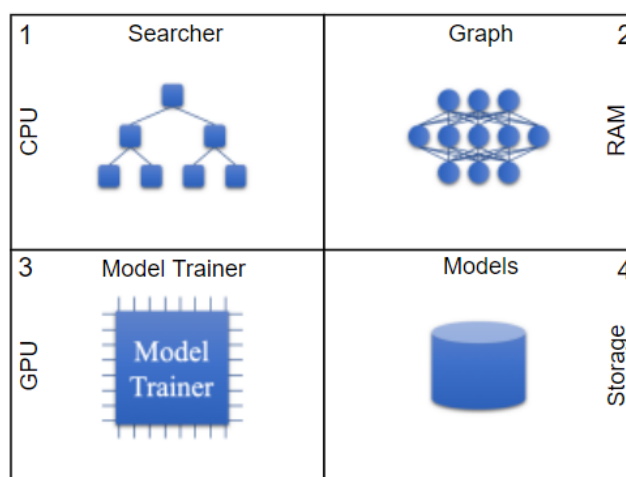


Figura 7: Arquitetura e funcionamento da ferramenta AutoKeras. Adaptado de Jin et al. (2019).

2.6 APLICAÇÃO DE MACHINE LEARNING NO SETOR TÊXTIL

Segundo Usuga Cadavid et al. (2020), o uso de *ML* para *PPC* divide-se em 7 grandes grupos, sendo que 48% da sua utilização visa planejar e obter uma calendarização inteligente, 14% obter uma estimativa de tempos, 10% direcionados para manutenção inteligente, 9% para controlo de qualidade e para controlo e monitorização do processo, 6% para controlo de inventário e distribuição e 4% para desenho inteligente de produtos e processos.

Kutschenreiter-Praszkiwicz (2008), no seu caso de estudo, tinha como objetivo a configuração de uma *ANN* para determinar o tempo de operação do processo de maquinação de determinada empresa. A rede adotada para o estudo foi a *RBF* que contempla três camadas: de entrada, oculta com não-linearidade *RBF* e uma camada de saída linear. O algoritmo utilizado no treino foi o *k-means algorithm*, e a configuração da rede que obteve melhores resultados foi a seguinte: (4:4-4-6:6), quatro neurónios na camada de entrada, quatro na camada oculta e seis na camada de saída, já o número de *epochs* utilizado foi 50. Para estas condições, o resultado obtido para o *RMS* foi, aproximadamente, 0.016.

Neste propósito, Eraslan (2009), utilizou uma arquitetura *MLP*, com diferentes tipos de algoritmos de aprendizagem, para calcular o tempo das operações numa empresa de moldagem. Foi efetuada uma seleção dos dados que tinham impacto para o estudo, bem como uma normalização destes. Foi utilizado um valor de *epochs* = 800, e utilizadas dois tipos de *hidden layers*, $6 \times 4 \times 2$ e $7 \times 5 \times 3$, aplicados a quatro algoritmos específicos, *Fletcher-Reeves*(*CGF*), *Broydon-Fletcher-Goldfarb-Shanno* (*BFG*), *Levenber-Marquardt* (*LM*) e *Bayesian Regularization* (*BR*). Os resultados obtidos foram mais satisfatórios para a primeira configuração ($6 \times 4 \times 2$) e para o algoritmo *LM*, com resultados de erro no treino e teste reduzidos.

Zakaria (2011) propôs um sistema de dimensionamento de uniformes escolares para crianças na Malásia. Nesse sentido, foram selecionadas 1000 raparigas de 29 escolas aleatórias, com idades compreendidas entre 7 e 17 anos. Para desenvolver o sistema, utilizou-se *clustering*, bem como uma árvore de classificação. Deste modo, a investigação levou à produção de 4 sistemas principais de dimensionamento, para as idades 7-12 e para 13-17 tanto para a parte

superior do corpo, como para a parte inferior. No total, foram desenvolvidos 93 tamanhos para as raparigas dos 7-17 anos de idade. A cobertura do sistema de dimensionamento para raparigas de 7-12 anos de idade é de 99,6%, para a parte superior do corpo e, 99,2% para a parte inferior do corpo. A cobertura para raparigas de 13-17 anos de idade é de 93,6%, para a parte superior do corpo, e de 98,2% para a parte inferior do corpo. Assim sendo, para o efeito, a aplicação de *ML* permitiu um ajuste melhorado do uniforme escolar das raparigas em idade escolar na Malásia, bem como a criação de uma diretriz para um sistema de dimensionamento e designação de tamanhos adequados.

Por sua vez, Nurwaha & Wang (2012) desenvolveu um sistema de previsão da qualidade do fio. Para o efeito, selecionou uma rede *ANN* com arquitetura *Multilayer Perceptron (MLP)*, uma *SVM*, uma *Radial Basis Functions (RBF)*, uma *Graph Neural Network (GNN)*, uma *Gene Expression Programming (GEP)* e uma *LR*. As redes foram alimentadas com dados como a resistência da fibra, o alongamento da fibra e o índice de uniformidade, entre outros. Após a aplicação dos modelos, verificou-se que os que tinham um *MAE* inferior foram a *SVM*, a *GNN*, a *GEP* e *LR*, com *MAE* de 82.87, 89.65, 93.69 e 97.73, respetivamente. Já as redes *MLP* e *RBF* conferiram *MAE* mais elevados, 100.91 e 131.63, respetivamente.

Akyol et al. (2015), com o objetivo de determinar o tempo de secagem da lã, utilizou uma abordagem de regressão que contou com vários métodos, desde algoritmos como o *MLP*, *LR*, *RBF* e, ainda, *rule-based algorithms* e *decision tree-based algorithms*. Segundo a investigação divulgada, a *REP Trees* inserida no grupo dos algoritmos *decision tree-based*, foi a que teve melhor resultado com um *MAE* e um *RMSE* de 0, ao contrário das restantes redes que tiveram um *MAE* entre 760 e 2660 e um *RMSE* entre 1260 e 3215.

No mesmo ano, Matusiak (2015), comparou 4 *ANN* com diferentes configurações para prever permeabilidade de tecidos ao ar, deste modo, a *ANN 1*, tinha uma arquitetura *MLP 7-4-1*, com 4 neurónios na camada oculta e com uma função de ativação linear na camada de saída. A *ANN 2*, tinha uma arquitetura *MLP 7-3-1*, com 3 neurónios na camada oculta e com uma função de ativação *hyperbolic tangente* na camada de saída. A *ANN 3*, tinha uma arquitetura *MLP 7-2-1*, com 2 neurónios na camada oculta e com uma função de ativação *hyperbolic tangente* na camada de saída. Por sua vez, a *ANN 4*, tinha uma arquitetura *MLP 7-2-1*, com 2 neurónios na camada oculta e com uma função de ativação exponencial na camada de saída. Em termos de resultados, a rede com melhores resultados foi a *ANN 1*, com um *MAE* de 0.1714 e um *RMSE* de 100.64, contrastando com os valores da *ANN 4*, que foram os piores, com um *MAE* de 0.8034 e um *RMSE* de 329.14. Também em 2015, Jaouachi & Khedher (2015) desenvolveram um modelo preditivo para o consumo de fio de costura numas calças jean, através de uma *ANN* com arquitetura *MLP* com *back-propagation*. A rede usada continha 6 nodos de entrada, 1 de saída e 4 camadas ocultas. Os resultados obtidos foram expostos num gráfico e comparados com um modelo de *LR* simples, onde foi possível perceber que a rede neuronal dá uma previsão mais precisa, fornecendo boas previsões quanto à quantidade de linha de costura necessária quando comparada com o modelo de *ANN*. O R^2 da rede foi de 0.973.

Com o intuito de prever a resistência de um tecido ao rasgo, Eltayib et al. (2016) providenciaram uma investigação nesse sentido, utilizando dados como o número de fios, a resistência

à tração do fio e a densidade linear do tecido, para alimentar uma *LR*. Os resultados foram apresentados sob duas perspectivas: no que diz respeito à força de rutura dos fios na direção da teia, o erro padrão foi de 0.467 e o R^2 foi de 0.663, já no que diz respeito à força de rutura dos fios na direção da trama, o erro padrão foi de 0.105 e o R^2 foi de 0.995.

Fallahpour et al. (2017) utilizou técnicas de *ML* para avaliar e selecionar os melhores fornecedores para uma cadeia de abastecimento (*supply chain management*). Para isso, aplicou o mesmo conjunto de dados, com uma validação cruzada de 5 *folds*, a diferentes algoritmos e comparou o *MSE* destes. O algoritmo que apresentou um *MSE* menor foi o *GEP*, uma abordagem robusta baseada em programação genética linear com um *MSE* de 16.390, seguida da *ANFIS*, uma rede neuronal artificial com 5 camadas que obteve um *MSE* de 17,396, seguindo-se a *RBF*, *SVM* e *MLP* com valores de *MSE* de 17.563, 19.418 e 21.518, respetivamente.

Lee & Lin (2018) na sua investigação, aplicaram técnicas de *ML* para melhorar a qualidade e eficácia dos processos/operações no setor têxtil. Nesta senda, incorporaram no seu artigo dados relativos ao histórico do processo de fabrico, parâmetros de operação de máquinas de urdissagem, processo de tecelagem, resultados do controlo de qualidade, entre outros. O estudo é dividido em dois setores, o primeiro utiliza modelos de regressão para prever parâmetros de operação têxtil e o segundo serve-se da classificação para prever a qualidade dos tecidos, para as duas abordagens foi efetuada uma *cross-validation* com 10 *folds*. Na abordagem de regressão, a métrica escolhida pelos investigadores foi o *MSE*, métrica onde a *LR* teve um valor 2.872, a *Lasso Regression* de 0.0001, a *Ridge Regression* de 1.0668 e a *ElasticNetRegression* de 0.0111. Por sua vez, na abordagem de classificação, os investigadores utilizaram a *Accuracy* como métrica de comparação para os modelos: *DT* com valores de 0.80334, *RF* com 0.88307, *AdaBoost* com 0.90243, *Gradient Boosting* 0.89895 e *XGBoost* 0.90849.

Onaran & Yanık (2020), utilizaram *ANN* com o intuito de prever o tempo de um ciclo de uma operação no setor têxtil. Para o estudo, foi escolhida uma *MLP*, um dos tipos de *ANN* mais comum, tendo sido aplicada com diferentes parâmetros, que, após a sua comparação, se concluiu que os que conferiam melhores resultados foram os seguintes: utilização da função de ativação *relu*, aplicação do *steepest gradient descent* no *solving method*, bem como $\alpha = 0.0001$, $\beta = 0.9$ e *hidden layer size* = 100. Após estas configurações, o modelo foi testado com todas as variáveis numa primeira fase, e com a exclusão de algumas variáveis, baseando-se na elevada relação “p” no modelo de regressão, numa segunda fase. Os resultados obtidos de R^2 , no modelo com todas as variáveis foi cerca de 0.859, contrastando com os cerca de 0.862 obtidos aquando da remoção de variáveis.

Ainda em 2020, na sua investigação, Ribeiro et al. (2020) utilizou dados reais de uma empresa têxtil portuguesa com o objetivo de reduzir o número de tentativas executadas para a produção de um tecido. Assim sendo, a previsão do resultado de determinado teste é benéfica para a empresa na medida em que reduz o número de amostras físicas necessárias para se conceberem novos tecidos. Posto isto, foram selecionadas duas tarefas de regressão têxtil, a resistência ao rasgamento em direção à teia e à trama. É ainda de realçar, que nesta investigação o foco maior foi na engenharia das características e na transformação de dados sendo que para a obtenção de resultados se aplicaram técnicas de *AutoML*. Para além do *baseline test*, foram efetuadas mais 3 operações, sendo que para a resistência ao rasgamento na

direção da teia o melhor resultado se obteve na segunda iteração com um *MAE* de 0.7 e na direção da trama na terceira iteração com um *MAE* de 1.16.

Mais recentemente, Shao et al. (2021) propôs um novo modelo de previsão baseado em *SVM* com *Particle Swarm Optimization (PSO)* com o intuito de conjecturar o tempo padrão do processo de costura. Foram selecionados os dados considerados relevantes para determinar o tempo padrão do processo de costura e, para a normalização dos dados, foi implementado o método *min-max normalization*. Os resultados obtidos no teste foram de 0.0844 e 0.852 para o *MSE* e R^2 , respetivamente. Também em 2021, C.-C. Wang et al. (2021), na sua investigação propôs a aplicação dos algoritmos *Autoregressive Integrated Moving Average (ARIMA)* e *Long Short-Term Memory Recurrent Neural Network (LSTM)* para prever a procura de encomendas com base em prazos de entrega curtos e em requisitos de entrega dentro do prazo estimado, para isso, seguiu um conjunto de tarefas: começou pelo ajuste e processamento de dados, prosseguiu com a implementação dos algoritmos, e, por fim, obteve os resultados preditivos e comparou os seus resultados através da métrica *RMSE*, sendo que os valores da *LSTM* foram melhores que os do *ARIMA* em todas as experiências, realçando-se ainda que todas as experiências, tanto da *LSTM* como do *ARIMA*, apresentavam valores mais satisfatórios e próximos da realidade que os que a empresa utilizava no seu conhecimento empírico.

Existem diversas características de um fio, como a uniformidade e a pilosidade, que podem aumentar a tensão na tecelagem e, quando combinadas com a tensão de enrolamento, podem causar falhas que interrompem a produção, como quebras de teia e de trama. A este respeito, vários autores apresentaram metodologias empíricas, estatísticas e instrumentais para resolver o problema das quebras de teia (Das et al., 2011). Nos últimos anos, foram também propostas diferentes abordagens baseadas em técnicas de *ML* para resolver este desafio.

Cioară et al. (2004) propôs um programa baseado em modelos matemáticos através de relações entre elementos específicos do fio e a sua resistência à quebra, com o objetivo de prever as quebras do fio no processo de tecelagem, e concluiu que as quebras da trama são significativamente influenciadas pelo comprimento de inserção da trama e pelo tipo de inserção, dado que o aumento da largura da máquina terá uma relação direta com a probabilidade de quebra.

Por sua vez, Yao et al. (2005) utilizou uma *feedforward back-propagation Neural Network (NN)*, para prever a taxa de quebras de teia, obtendo um modelo que utilizou uma camada oculta *sigmoid* com quatro neurónios, permitindo inferir que a previsão da quebra de teia é viável (foi obtido um coeficiente de correlação de 99.5% entre os valores previstos e reais).

Mwasiagi et al. (2008) também adotou uma *NN* para prever o alongamento da quebra do fio e, de seguida, inspecionou a influência dos parâmetros de entrada e das propriedades da *NN* na previsão. Das funções neuronais utilizadas, o melhor desempenho foi alcançado com o algoritmo *Purelin* e a melhor arquitetura utilizada foi a que utilizou a função de treino *Levenberg-Marquardt*. Os seis fatores mais influentes foram: a torção do fio, contagem do fio, alongamento da fibra, comprimento, uniformidade de comprimento e velocidade do fuso. Noutro estudo,

Zhao Bo (2010) propôs a utilização de uma *NN* com *feedforward back-propagation* para prever a taxa de quebras de teia utilizando o índice de qualidade de fio como atributo de entrada. O

modelo utilizado continha apenas uma camada oculta e oito neurónios, obtendo um coeficiente de correlação quadrado de 99,3%. Estes trabalhos apoiam a ideia de que as *NN* podem prever taxas de quebra de teia e que são uma abordagem útil com grande potencial para a indústria têxtil.

Almetwally et al. (2013) utilizou a técnica *ANOVA* para detetar os efeitos das variáveis nas quebras de fio na tecelagem e um modelo de regressão para prever o número de quebras de fio, com o objetivo de inferir o tipo de impacto da relação de mistura algodão/poliéster, tipo de algodão, torção de fio e contagem de fios nas quebras de fio no tear. Em termos de fator de impacto, os resultados revelam que a relação de mistura algodão/poliéster tem um impacto considerável nas quebras de fio, e, ainda, que o número de quebras varia diretamente com a quantidade de fio simples e de fio alisado.

De acordo com El-Tarfawy (2016) os fatores que mais afetam a eficiência de um tear de jato de ar são as quebras de teia e de trama. Os autores utilizaram um algoritmo de *backpropagation* para treinar uma *NN* de modo a prever o número de quebras de trama por milhão de metros, obtendo um valor de correlação quadrado de 0,955 entre o número real de quebras de trama e o número previsto, indicando que as quebras de trama podem ser previstas através de *NN*.

Mais recentemente, Seboka (2021) desenvolveu um modelo de *LR* para verificar se existe uma ligação entre as quebras de teia no tear e o estiramento mecânico do fio. Esta ligação ficou corroborada pelos resultados obtidos (Coeficiente de correlação quadrado de 84.4%).

A Tabela 1 contempla um resumo de todas as técnicas supramencionadas.

Tabela 1: Resumo das Técnicas de *Machine Learning* aplicadas ao setor têxtil

Autor	Algoritmo	Objetivo
Cioară et al. (2004)	Modelos matemáticos	Tempo de inatividade da máquina
Yao et al. (2005)	ANN	Tempo de inatividade da máquina
Kutschenreiter-Praszkiewicz (2008)	ANN-RBF	Tempo de Operação
Mwasiagi et al. (2008)	ANN	Tempo de inatividade da máquina
Eraslan (2009)	ANN-MLP	Tempo de Operação
Zhao Bo (2010)	ANN	Tempo de inatividade da máquina
Zakaria (2011)	DT	Desenvolvimento de um sistema de tamanhos
Nurwaha & Wang (2012)	SVM	Previsão da qualidade do fio têxtil
Almetwally et al. (2013)	ANOVA Modelos de regressão	Tempo de inatividade da máquina
Akyol et al. (2015)	Vários	Tempo de Operação
Matusiak (2015)	Várias ANN	Permeabilidade de Tecidos ao ar
Jaouachi & Khedher (2015)	ANN	Previsão do consumo de linha na produção de calças
Eltayib et al. (2016)	LR	Previsão da resistência de um tecido ao rasgamento
El-Tarfawy (2016)	ANN	Tempo de inatividade da máquina
Fallahpour et al. (2017)	Vários	Seleção e Avaliação de Fornecedores
Lee & Lin (2018)	Lasso Regression XGboost	Operação de Tecelagem
Onaran & Yanık (2020)	ANN	Tempo de Operação
Ribeiro et al. (2020)	AutoML	Previsão do resultado de um determinado teste têxtil
Shao et al. (2021)	SVM + PSO	Tempo de Operação
C.-C. Wang et al. (2021)	ARIMA LSTM	Procura de encomendas com base nos prazos de entrega
Seboka (2021)	LR	Tempo de inatividade da máquina

2.7 FERRAMENTAS TECNOLÓGICAS

Na presente secção reclama-se a apresentação das ferramentas tecnológicas que serão utilizadas no desenrolar da dissertação. Tendo em conta a sua componente de *DS* e de *ML*, o projeto será desenvolvido em Python¹.

A escolha desta linguagem está diretamente relacionada com a quantidade de bibliotecas disponíveis para esta linguagem que auxiliam os desenvolvedores em várias tarefas, como é o caso do *pandas* para o pré-processamento e para a modelação dos dados. Uma das vantagens da utilização desta linguagem é a quantidade de bibliotecas disponíveis para auxiliar em tarefas relacionadas. Na Tabela 2, encontram-se explicadas as bibliotecas consideradas relevantes para o desenrolar do projeto.

¹ <https://www.python.org/>

Tabela 2: Módulos Python utilizados neste trabalho.

Nome	Descrição	Área
AutoGluon-Tabular	Apresenta-se como uma estrutura de <i>AutoML</i> , <i>open source</i> , de fácil utilização, que permite treinar modelos de <i>ML</i> de elevada precisão, num conjunto de dados tabulares não processados. Ao contrário das demais ferramentas de <i>AutoML</i> , o sucesso da ferramenta provem do agrupamento de vários modelos em múltiplas camadas e não na otimização de hiperparâmetros. Erickson et al. (2020).	Modelação
AutoKeras	É um sistema de <i>AutoML</i> , <i>open source</i> , que pode ser instalado localmente. O sistema foi concebido por forma a poder ser utilizado por pessoas não especializadas em informática e <i>DS</i> . Por forma a garantir uma pesquisa mais otimizada, este pode ser executado na <i>CPU</i> e na <i>GPU</i> em simultâneo, Jin et al. (2019).	Modelação
Cane	Apresenta-se como um <i>package</i> simples, mas poderoso para pré-processamento de variáveis categóricas, oferecendo alguns métodos para o efeito, como é o caso do <i>PCP</i> que junta todos os dados menos frequentes e categoriza-os, numa só classe, por exemplo, "outros", do <i>IDF</i> que codifica os dados categóricos em valores de frequência, sendo que, quanto mais próximo de o for o valor, mais frequente ele é, conta também com a integração de um método simples de <i>One-Hot-Encoding</i> e com métodos de normalização baseados no <i>MinMax</i> e no <i>Standard Scaler</i> do <i>scikit-learn</i> , Matos et al. (2020).	Pré- -Processamento

Nome	Descrição	Área
Grafana	É uma multi-plataforma web, de análise e visualização interativa <i>open source</i> . Fornece gráficos e alertas para uma página web quando ligada a fontes de dados suportados. Os utilizadores finais podem criar painéis de monitorização complexos para os seus modelos de <i>ML</i> .	Monitorização
H2O AutoML	Apresenta-se como um <i>package</i> complexo, mas de fácil utilização, que permite automatizar o fluxo de trabalho em <i>ML</i> , inclui treino e afinação de parâmetros para vários modelos num determinado período de tempo estipulado pelo utilizador, permitindo que este se foque mais em tarefas relacionadas com a pipeline de <i>DS</i> , como por exemplo o processamento de dados e engenharia de características. Todos os modelos executados aparecem sob a forma de uma <i>leaderboard</i> , LeDell & Poirier (2020).	Modelação
Kedro	É uma ferramenta <i>Python</i> , <i>open source</i> , com o intuito de promover a criação de código reprodutível, modular e maneável para <i>DS</i> . Deste modo, incorpora as melhores práticas da engenharia de <i>software</i> ao desenvolvimento de <i>ML</i> .	Reprodutibilidade do código
Matplotlib	Apresenta-se como uma biblioteca abrangente para a criação de visualizações estáticas, animadas e interativas em <i>Python</i> , sendo uma extensão do <i>Numpy</i> . Inclui uma <i>API</i> orientada a objetos que permite a integração dos gráficos criados em aplicações através da utilização de ferramentas de <i>GUI</i> como o " <i>Tkinter</i> ", " <i>wxPython</i> ", " <i>Qt</i> " ou " <i>GTK</i> ", Hunter (2007).	Visualização

Nome	Descrição	Área
Mlflow	É uma ferramenta <i>open source</i> para gerir o ciclo de vida de um projeto de <i>ML</i> e oferece dois componentes de destaque: Rastreamento, registo e consulta de experiências (código, dados, configuração e resultados) e Modelo de Registo, possibilidade de armazenar, anotar, descobrir e gerir modelos num repositório central, Zaharia et al. (2018).	Rastreamento de modelos de <i>ML</i>
Numpy	Apresenta-se como uma biblioteca para <i>Python</i> que suporta o processamento de grandes e multi-dimensionais matrizes e apresenta um vasto conjunto de funções matemáticas que permitem ao utilizar fazer operações nestas mesmas matrizes, sendo por isso uma biblioteca imprescindível em projetos de <i>ML</i> , Harris et al. (2020).	Análise
Pandas	É uma biblioteca criada para <i>Python</i> com o intuito de auxiliar a manipulação e análise de dados. Particularmente, oferece estruturas e operações para a manipulação de tabelas numéricas e séries temporais. No que diz respeito a projetos de <i>ML</i> , esta biblioteca é particularmente usada pela facilidade que a manipulação do objeto <i>dataframe</i> oferece, McKinney (2010).	Análise Pre- -Processamento
Pandas Profiling	Apresenta-se como um módulo para a linguagem <i>Python</i> , <i>open source</i> , com o qual podemos fazer uma extensa análise exploratória dos dados, de forma rápida e em poucas linhas de código, poupando ao utilizador o trabalho de visualização e compreensão da distribuição de cada atributo, Brugman (2019).	Análise

Nome	Descrição	Área
Plotly	É uma biblioteca <i>open source</i> que permite aos utilizadores importar, copiar, colar ou transmitir dados para serem analisados e visualizados. Os gráficos podem ser desenhados com <i>Python</i> e com uma interface gráfica e partilhados através de um URL, para que outros interessados possam ver, colaborar ou guardar uma cópia, Plotly (2015).	Visualização
Scikit-learn	Apresenta-se como uma biblioteca para <i>Python</i> , <i>open source</i> e contém muitas ferramentas eficientes para <i>ML</i> e modelação estatística incluindo classificação, regressão e <i>clustering</i> , Pedregosa et al. (2011).	Pré- -Processamento Modelação
Seaborn	É uma biblioteca de visualização de dados <i>Python</i> , baseada em <i>matplotlib</i> . Fornece uma interface de alto nível para desenhar gráficos estatísticos atraentes e informativos, oferecendo bastantes facilidades para trabalhar com <i>dataframes</i> e com a biblioteca <i>pandas</i> , Waskom et al. (2017).	Visualização
Statsmodels	Apresenta-se como uma biblioteca para a linguagem <i>Python</i> que permite aos utilizadores, explorar dados, estimar modelos e realizar testes estatísticos, apresentando uma extensa lista de estatísticas descritivas, de testes e de funções para traçar gráficos dependendo do tipo de dados, Seabold & Perktold (2010).	Pré- -Processamento

CASO DE ESTUDO: PLANEAMENTO E CONTROLO DA PRODUÇÃO

O terceiro capítulo é correspondente ao estudo experimental da dissertação, deste modo, foi seguida a metodologia do *CRISP-DM* para cada requisito de *ML* a desenvolver. De realçar, que foram direcionados mais esforços para a componente de preparação dos dados, e, como tal, a etapa de modelação foi feita recorrendo a ferramentas de *AutoML*.

3.1 COMPREENSÃO DO NEGÓCIO

Atualmente, existe uma pressão na *ITV* para aumentar a eficiência e reduzir os prazos de entrega. Neste contexto, questões que afetam o planeamento e o controlo da produção como o rendimento efetivo dos artigos (requisito 1), a paragem da produção (requisito 2), o tempo de entrega de matérias-primas (requisito 3) e o tempo de *setup* (requisito 4) são das questões mais importantes dado que afetam diretamente a produtividade, a eficiência e a rentabilidade. Na presente dissertação, colaboramos no processo de transformação digital da Somelos, uma empresa têxtil portuguesa que produz tecidos, a fim de melhorar a autonomia e os níveis de desempenho do planeamento e controlo da produção. Concentrámo-nos particularmente em questões levantadas pela organização e que estão relacionadas com o processo de tecelagem, com o objetivo de fazer previsões mais eficientes melhorando o processo de planeamento da organização. A Somelos já implementou coletores de dados nos teares, assim, ao utilizar estes dados recolhidos pelo *IOT* como input para os algoritmos de *ML*, o objetivo é prever o rendimento dos artigos, prever o número de defeitos nos teares que causam a paragem da máquina, prever o tempo de entrega de matérias-primas, bem como prever o tempo de operações como a urdissagem, engomagem e remetagem, com o intuito de apoiar a tomada de decisão no momento de planear a produção.

3.2 COMPREENSÃO E PREPARAÇÃO DOS DADOS

Nas fases de compreensão e preparação dos dados, inserem-se os processos de recolha, exploração e seleção de dados tendo em perspetiva a correção de problemas e a construção do *dataset*. Desta forma, na etapa relativa à preparação dos dados recorreu-se à função *IDF*, proposta por Campos et al. (2016) e que se representa pela seguinte fórmula:

$$IDF(t) = \ln\left(\frac{N}{f_t}\right) \quad (1)$$

onde N representa o total de valores, e f_t simboliza o número de ocorrências do atributo t no *dataset* de treino. Para a preparação de dados foram também utilizadas as operações *Standard-Scaler* e *label-encoder*, a primeira é uma operação efetuada em valores numéricos que normaliza os dados, tornando a média da coluna em zero e o desvio padrão da mesma coluna em 1, enquanto a segunda substitui os valores de uma coluna categórica por índices numéricos, atribuindo 1 índice para cada valor distinto.

3.2.1 Requisito 1: Previsão do Rendimento dos Artigos

No requisito 1, previsão do rendimento dos artigos, após uma fase introdutória de contacto com os dados, o *target* (*rendi*), foi obtido através da divisão de 2 atributos provenientes da tabela *tbHqtturnMqOP* por indicação da Somelos Tecidos.

Para a criação do *dataset* final, foram aplicados processos de *inner joins* nas das chaves primárias de 7 tabelas distintas. A Tabela 3 representa um resumo dos 22 atributos selecionados. De realçar que o *dataset* utilizado contabiliza 20277 linhas, sendo que nenhum atributo contém valores em branco e/ou nulos.

No que diz respeito ao atributo do rendimento dos artigos ("*rendi*"), os seus valores compreendem-se no intervalo $[0, 100]$ sendo que a média se encontra nos 78%. A sua distribuição pode ser encontrada no histograma presente na Figura 8.

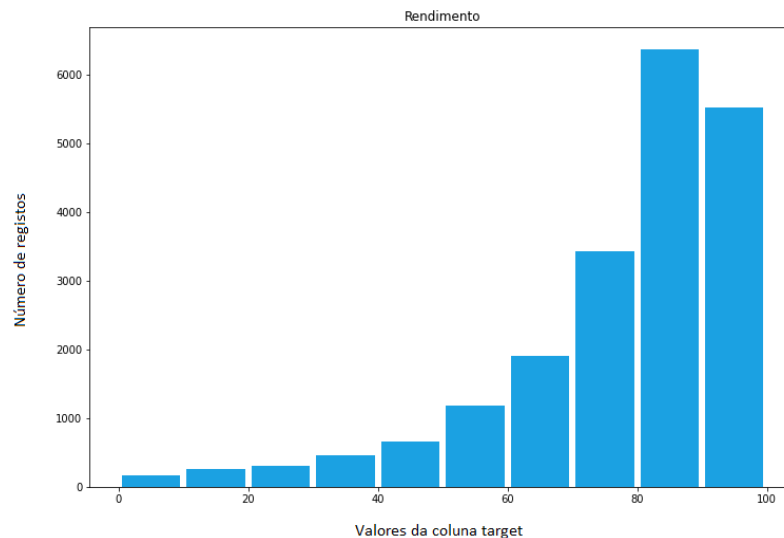


Figura 8: Histograma do atributo relativo ao rendimento dos artigos

A preparação dos dados consistiu na aplicação de processos de *encoding* de valores categóricos, bem como de valores numéricos que representavam códigos identificativos. Deste modo, o atributo *tipotecart* como apenas apresentava dois possíveis valores ('A', para artigo acabado e 'C', para artigo cru) foi transformado através de um *label-encoder*, isto é, o valor 'A' passou a ser 0 e o valor 'C' passou a ser 1. Para a codificação dos atributos identificativos (*codtecart*, *descart*, *destart* e *codtipmaq*) foi aplicada a função *IDF*.

Após a aplicação do *IDF*, todos os atributos, à exceção do *target*, passaram por uma etapa de normalização através da aplicação do *Standard-Scaler*. É ainda de notar que o atributo *numkcha*

Tabela 3: Resumo dos atributos relativos ao Requisito 1: Previsão do Rendimento dos Artigos

Atributo	Descrição	Dataset	Exemplo	Tipo
numkcha	Ordem de trabalho	Maqot Montage	411420	Int
idTurno	Identificativo do turno	hisMaquina	2	Int
rendi	Rendimento do artigo	tbHqturnMqOP	89.45	Double
codtecart	Código do artigo	Artiti Ofilig	4007313	Int
descart	Desenho da teia	Artiti	6301W	String
destart	Desenho da trama	Artiti	0010	Int
largart	Largura do tecido	Artiti	180	Int
tiptecart	Tipo de tecido	Artiti	A	String
nbfarm	Total de fios teia	Artiti	5440	Int
cpcart	Fios teia por polegada	Artiti	92	Int
cptart	Passagens por polegada artigo cru	Artiti	79	Int
acabpass	Passagens por polegada na trama artigo acabado	Artiti	82	Int
nbfrapart	Fios totais trama por rappord	Artiti	6	Int
empgnart	Largura pente do artigo	Artiti	165.1	Double
licart	Liços da ourela	Artiti	2	Int
plmiart	Peso linear artigo cru	Artiti	163	Int
complicart	Liços do corpo do tecido	Artiti	8	Int
qltec	Percentagem de segunda qualidade do artigo	Artiti	1.5	Double
p2maart	Peso do artigo acabado	Artiti	106	Int
codtipmaq	Tipo de máquina	Maquin	35	Int
velmaq	Velocidade da máquina	Maquin	600	Int
larpen	Largura do pente	Maquin	240	Int

foi removido por se apresentar como um código identificativo incremental que não acrescenta informação de valor.

3.2.2 Requisito 2: Previsão de Defeitos nos Teares

Para a criação do *dataset* analisado, juntamos atributos de dados relativos às características dos tecidos e das máquinas. Duas tabelas da base de dados da empresa têxtil foram utilizadas para recolher características dos tecidos: uma com 20.999 registos que apresentavam características genéricas de todos os tecidos, e outra com 28.249 registos que correspondiam às características dos tecidos produzidos nas máquinas. Por outro lado, os dados relacionados com as características das máquinas vieram de quatro tabelas distintas: uma com características gerais das máquinas (264 registos), duas tabelas com a ligação entre as ordens de trabalho e as máquinas (24.484 e 30.964 registos) e outra com dois anos de dados sensoriais sobre o tempo de inatividade das máquinas, com um total de 5.517.384 registos correspondentes a 29.959 ordens de trabalho, das quais apenas 11.855 tinham correspondência nas restantes tabelas.

Para efeitos de extração, foram aplicadas operações de *inner joins* entre as chaves primárias das tabelas para criar apenas um *dataset*, resultando em 1.966.217 registos. Para obter o número de quebras de teia, quebras de trama, e rebentadelas de fios, os dados tiveram de ser agrupados utilizando os atributos *codtecart*, *idTurno*, *numkcha*, e *codtipmaq*, o que resultou em três novas colunas que representam o número de quebras de teia, quebras de trama, e rebentadela de fios, reduzindo o *dataset* para 1.502 registos. Para além dos três *targets* (número de quebras de teia, número de quebras de trama e número de rebentadelas de fio), o *dataset* final tem vinte atributos diferentes, 18 dos quais numéricos, um categórica, e um nominal. Em particular, o atributo *numkcha* foi descartado dado que é um simples código de identificação incremental.

Tabela 4: Resumo dos atributos relativos ao Requisito 2: Previsão de Defeitos nos Teares

Atributo	Descrição	Dataset	Exemplo	Tipo
<i>numkcha</i>	Ordem de Trabalho	Maqot Montage	411421	<i>Int</i>
<i>codtecart</i>	Código do artigo	Artiti	4007313	<i>Int</i>
<i>tiptecart</i>	Tipo de tecido	Artiti	C	<i>String</i>
<i>descart</i>	Desenho da teia	Artiti	6301W	<i>String</i>
<i>destart</i>	Desenho da trama	Artiti	0010	<i>Int</i>
<i>largart</i>	Largura do tecido	Artiti	180	<i>Int</i>
<i>nbfart</i>	Total de fios teia	Artiti	5440	<i>Int</i>
<i>cpcart</i>	Fios teia por polegada artigo cru	Artiti	79	<i>Int</i>
<i>nbfrapart</i>	Fios totais trama por rappid	Artiti	6	<i>Int</i>
<i>empgnart</i>	Largura pente do artigo	Artiti	165.1	<i>Float</i>
<i>licart</i>	Liços da ourela	Artiti	2	<i>Int</i>
<i>plmiart</i>	Peso linear artigo cru	Artiti	163	<i>Int</i>
<i>complicart</i>	Liços do corpo do tecido	Artiti	8	<i>Int</i>
<i>qltec</i>	Percentagem de segunda qualidade do artigo	Artiti	1.5	<i>Float</i>
<i>idTurno</i>	Identificativo do turno	hisMaquina	3	<i>Int</i>
<i>codtipmaq</i>	Tipo de Máquina	Maquin	35	<i>Int</i>
<i>cptart</i>	Passagens por polegada artigo cru	Artiti	79	<i>Int</i>
<i>acabpass</i>	Passagens por polegada na trama artigo acabado	Artiti	82	<i>Int</i>
<i>p2maart</i>	Peso do artigo acabado	Artiti	106	<i>Int</i>
<i>velmaq</i>	Velocidade da máquina	Maquin	600	<i>Int</i>
<i>larpen</i>	Largura do Pente	Maquin	240	<i>Int</i>
<i>Qte_count</i>	Número de quebras de teia	hisMaquina	22	<i>Int</i>
<i>Qtr_count</i>	Número de quebras de trama	hisMaquina	36	<i>Int</i>
<i>RF_count</i>	Número de rebentadela de fios	hisMaquina	1	<i>Int</i>

A Tabela 4 resume os atributos do conjunto de dados final. As primeiras 14 linhas estão relacionadas com os atributos do tecido, as 6 linhas seguintes estão relacionadas com os atributos das máquinas têxteis, enquanto os últimos atributos apresentados são os *targets* do requisito em questão. Os intervalos de valores para os três *targets* (número de quebras de teia, quebras de trama e rebentadela de fios) são: [1, 226], [1,236], e [1, 15], respetivamente.

A fase de pré-processamento de dados foi dividida em duas abordagens. Na abordagem A, a fase de pré-processamento de dados envolveu a transformação dos dados categóricos e nominais, bem como os atributos numéricos que representavam códigos de identificação, tais como *codtecart*, *codtipmaq* e *descart*. Para a transformação do atributo nominal, tendo em conta que este tinha apenas dois níveis, foi aplicado um *label-encoder*. Nas restantes transformações, foi aplicada a função do IDF, mencionada no requisito anterior. Esta transformação é particularmente útil quando a cardinalidade dos atributos a transformar é substancialmente elevada (Matos et al., 2018, 2020). Com o objetivo de normalizar os valores do *dataset*, foi aplicado um *standard scaler* a todos os atributos (exceto aos *targets*). A abordagem B seguiu todos os procedimentos anteriores, mas a função $\log(y + 1)$ foi aplicada aos *targets*. Esta transformação é geralmente adotada quando a distribuição dos *targets* é enviesada para a esquerda, como mostra a Figura 9. É de notar que, uma vez treinado um modelo, a função logarítmica inversa é aplicada às previsões do modelo, a fim de analisar os resultados da abordagem B utilizando a escala original do *target* (tal como realizada para a abordagem A). A aplicação da função logarítmica fez com que a distribuição ficasse mais centrada, removendo a distribuição enviesada para a esquerda que se observava principalmente no número de quebras de teia e no número de quebras de trama, tal como apresentado na Figura 10.

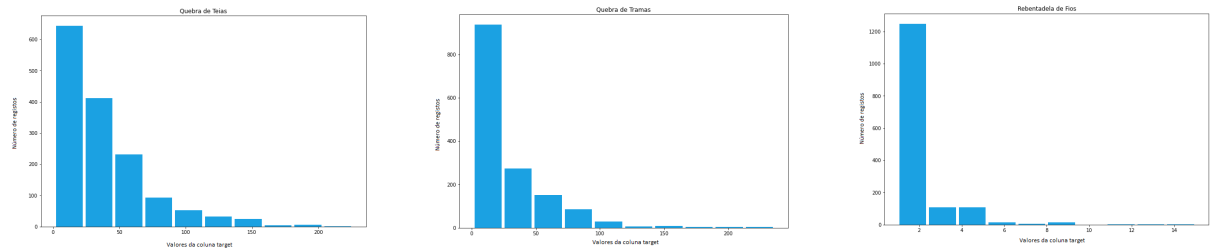


Figura 9: Distribuição do número de quebras de teia (esquerda), quebras de trama (meio) e rebentadela de fios (direita)

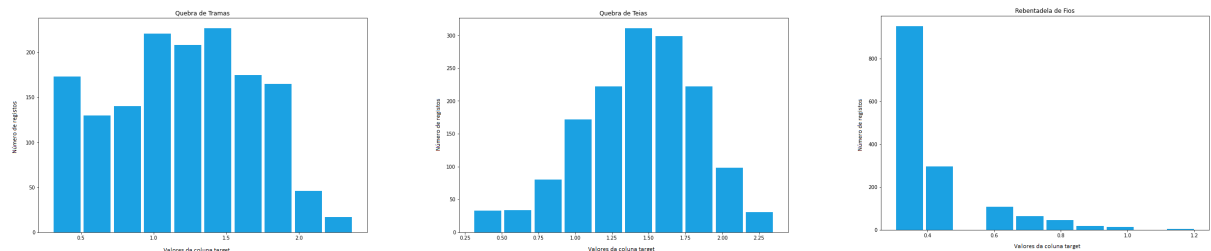


Figura 10: Distribuição do número de quebras de teia (esquerda), quebras de trama (meio) e rebentadela de fios (direita), após a aplicação da função logarítmica

3.2.3 Requisito 3: Previsão de Entrega de Matérias-Primas

Para a obtenção do *dataset* final, foram efetuados processos de “*inner joins*” entre as chaves primárias das tabelas *EncCab*, *EncLin* e *MovLin*. A Tabela 5 apresenta os atributos resultantes de uma *feature selection* posteriormente efetuada. É de notar que o *dataset* apresenta 1149 linhas, sendo que em nenhum atributo se verificam valores nulos e/ou brancos.

Tabela 5: Resumo dos atributos relativos ao Requisito 3: Previsão de Entrega de Matérias-Primas

Atributo	Descrição	Dataset	Exemplo	Tipo
linhaEnc	Código linha de encomenda fornecedor	EncLin MovLin	1	Int
qntEnc	Quantidade encomendada	EncLin	1500.0	Float
artigo	Artigo encomendado	EncLin MovLin	3602072	Int
codFornecedor	Identificativo do fornecedor	EncCab EncLin	120610	Int
armazem	Armazém para entrega	MovLin	630	Int
paisOrigem	País de origem da encomenda	EncCab MovLin	1	Int
dataEncomenda	Data em que foi efetuada a encomenda	EncCab	14/05/2020	Date
dataRecebida	Data em que foi recebida a encomenda	EncLin	15/05/2022	Date
qntRec	Quantidade recebida	MovLin	321.26	Float

Tendo em vista a criação de um atributo capaz de mensurar o número de dias desde que a encomenda foi efetuada até que a encomenda foi recebida no armazém, foi necessário criar um atributo novo (“tempoEfetivo”) que é resultante da subtração entre o atributo “dataRecebida” e o atributo “dataEncomenda”, sendo também o *target* desta previsão e cujo intervalo de valores se compreende em [1, 322]. O histograma do tempo efetivo que uma matéria-prima demora a ser entregue pode ser encontrado do lado esquerdo da Figura 11.

Pela análise do histograma à esquerda, conseguimos perceber que a distribuição do *target* se encontra enviesada para a esquerda, pelo que aplicamos a mesma técnica utilizada no requisito anterior para que a distribuição ficasse mais uniforme, tal como se apresenta no lado direito da Figura 11. Por forma a normalizar os dados, antes da fase de pré-processamento de dados terminar, foi aplicado o *standard-scaler* a todos os atributos com a exceção do *target*.

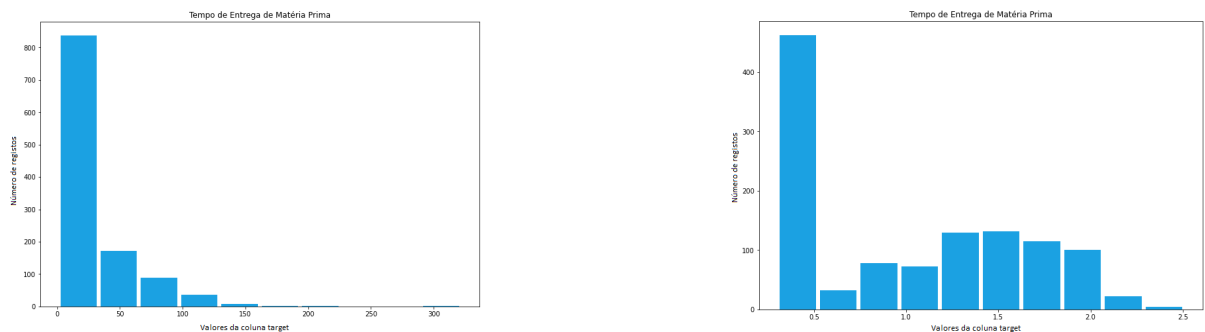


Figura 11: Histograma do atributo relativo ao tempo de entrega de matérias-primas sem a aplicação da função logarítmica (esquerda) e com a aplicação da função logarítmica (direita)

3.2.4 Requisito 4: Previsão de Tempos de Operação (Urdissagem, Engomagem e Remetagem)

A Tabela 6 resume os atributos utilizados para a previsão dos tempos de operação. A primeira secção é constituída pelos atributos comuns à urdissagem, engomagem e remetagem. A segunda secção engloba os requisitos adicionais utilizados na urdissagem, enquanto as terceira e quarta secções dizem respeito aos requisitos específicos da engomagem e da remetagem, respetivamente.

Tabela 6: Resumo dos atributos relativos ao Requisito 4: Previsão de Tempos de Operação

Atributo	Descrição	Dataset	Exemplo	Tipo
codtecart	Código do artigo	Artiti	4945318	<i>Int</i>
descart	Desenho da teia	Artiti	11011	<i>Int</i>
largart	Largura do tecido	Artiti	166	<i>Int</i>
tiptecart	Tipo de tecido	Artiti	C	<i>String</i>
nbfacha	Total de fios teia	Stkcha	1120	<i>Int</i>
num_got	Velocidade planeada	Stkcha	350	<i>Int</i>
qtecorkcha	Quantidade (em metros)	Stkcha	1135.00	<i>Float</i>
numsecprin	Nº de secções	Stkcha	28	<i>Int</i>
rapsecprin	Nº de rapports por secção	Stkcha	17	<i>Int</i>
rapsecult	Nº de rapports da última secção	Stkcha	14	<i>Int</i>
numrapcha	Nº de fios por rappord	Nomcha	10	<i>Int</i>
codurd	Código da urdissagem	Stkcha	2106	<i>Int</i>
horainiurdis	Hora de início da urdissagem	Stkcha	04:50	<i>Date</i>
datainiurdis	Data de início da urdissagem	Stkcha	2005-07-22	<i>Date</i>
horaurdis	Hora de fim da urdissagem	Stkcha	07:16	<i>Date</i>
dataurdis	Data de fim da urdissagem	Stkcha	2005-07-22	<i>Date</i>
peso_urd	Peso (em kg) por 1000 metros de teia urdida	Nomcha	47.22	<i>Float</i>
codengom	Código da engomagem	Stkcha	2501	<i>Int</i>
horainiengom	Hora de início da engomagem	Stkcha	08:11	<i>Date</i>
datainiengom	Data de início da engomagem	Stkcha	2005-07-22	<i>Date</i>
horaengom	Hora de fim da engomagem	Stkcha	09:25	<i>Date</i>
dataengom	Data de fim da engomagem	Stkcha	2005-07-22	<i>Date</i>
peso_engom	Peso (em kg) por 1000 metros de teia engomada	Nomcha	46.65	<i>Float</i>
codremet	Código da remetagem	Stkcha	2802	<i>Int</i>
horainiremet	Hora de início da remetagem	Stkcha	18:13	<i>Date</i>
datainiremet	Data de início da remetagem	Stkcha	2005-07-22	<i>Date</i>
horaremet	Hora de fim da remetagem	Stkcha	2005-07-22	<i>Date</i>
dataremet	Data de fim da remetagem	Stkcha	20:26	<i>Date</i>

De maneira a obter o tempo que cada operação demorou desde o seu começo até ao seu término, começou-se por unir as datas e as horas de cada operação, e, de seguida, fez-se a subtração entre a data final e a inicial, obtendo o tempo de cada operação.

Os histogramas dos *targets* para cada operação apresentam-se na Figura 12.

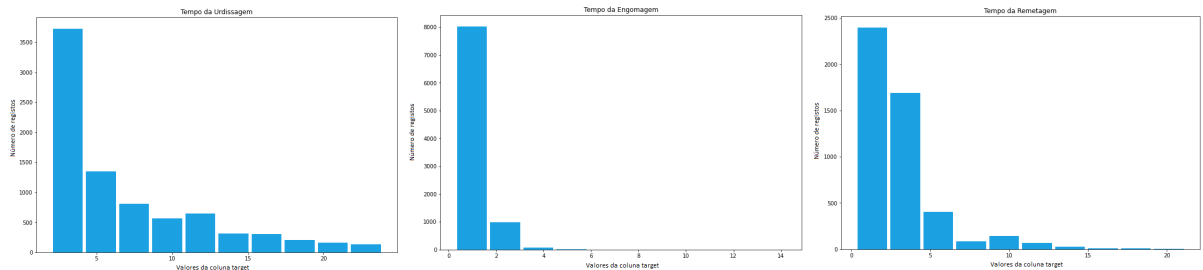


Figura 12: Distribuição do tempo de urdissagem (esquerda), engomagem (meio) e remetagem (direita)

Com o intuito de normalizar o *dataset*, todos os atributos não alvos de previsão foram alterados por um *standard-scaler*. Tal como efetuado nos requisitos anteriores, esta operação ocorreu após a aplicação do *IDF* aos atributos categóricos.

3.3 MODELAÇÃO

A conclusão da fase de processamento de dados pressupõe o início da fase de modelação, tal como ilustra a Figura 1. Nesta etapa, foi utilizada a linguagem de programação *Python* para a criação de modelos preditivos capazes de responder ao desafio inicial de cada requisito, recorrendo à ferramenta de H2O AutoML. Para o efeito, e de forma a conferir fiabilidade aos resultados finais, recorreu-se a uma *cross-validation* externa de 10 *folders*, sendo que, para cada *fold*, o H2O AutoML teve uma limitação de tempo de 1 hora, guardando o melhor modelo de cada *fold*, bem como os resultados obtidos. A Figura 13 representa o processo de *cross-validation* aplicado.

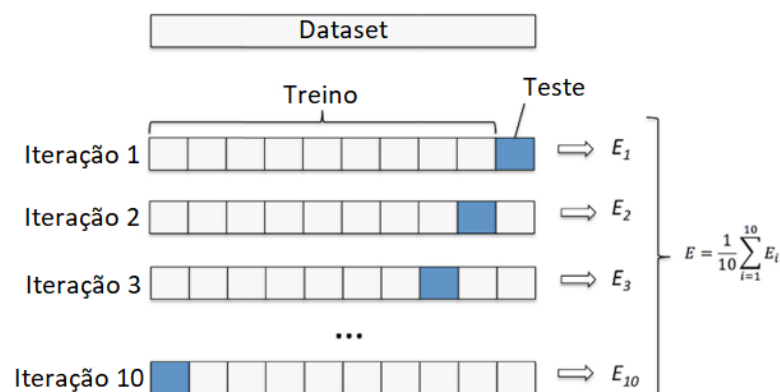


Figura 13: Exemplificação do processo de *cross-validation*. Adaptado de Sontakke et al. (2019).

O código desenvolvido para a aplicação da *cross-validation* através do H2O AutoML pode ser encontrado na Figura 14.

```

for target in targets:
    if target == "nome_target":
        df_fold1 = pd.read_csv(data_path + "-fold1.csv")
        df_fold2 = pd.read_csv(data_path + "-fold2.csv")
        df_fold3 = pd.read_csv(data_path + "-fold3.csv")
        df_fold4 = pd.read_csv(data_path + "-fold4.csv")
        df_fold5 = pd.read_csv(data_path + "-fold5.csv")
        df_fold6 = pd.read_csv(data_path + "-fold6.csv")
        df_fold7 = pd.read_csv(data_path + "-fold7.csv")
        df_fold8 = pd.read_csv(data_path + "-fold8.csv")
        df_fold9 = pd.read_csv(data_path + "-fold9.csv")
        df_fold10 = pd.read_csv(data_path + "-fold10.csv")

        folds = [df_fold1, df_fold2, df_fold3, df_fold4, df_fold5, df_fold6, df_fold7, df_fold8, df_fold9, df_fold10]

        for x in range(0, 10):
            h2o.init(ip="127.0.0.1", port="3000")

            fold_folder = "nome_target" + str(x+1)
            os.makedirs(fold_folder, exist_ok=True)
            folds = [df_fold1, df_fold2, df_fold3, df_fold4, df_fold5, df_fold6, df_fold7, df_fold8, df_fold9, df_fold10]
            test_df = folds[x]
            test = h2o.H2OFrame(test_df)

            del folds[x]
            train_df = pd.concat(folds)
            train = h2o.H2OFrame(train_df)

            x = train.columns
            y = target
            x.remove(y)

            aml = H2OAutoML(seed=42, sort_metric = "mae", nfolds=10, exclude_algos=["DeepLearning"], max_runtime_secs = 3600, max_models=1000)
            aml.train(x=x, y=y, training_frame=train)

```

Figura 14: Código Python da aplicação da *cross-validation*

No que diz respeito ao Requisito 2, previsão de defeitos nos teares, a fase de modelação teve variações quando comparada à dos outros requisitos, na medida em que foram delineadas duas estratégias de *ML*, uma *single* e outra *multi target*, isto é, na abordagem *single target* as previsões são feitas individualmente para cada um dos *targets*, enquanto na abordagem *multi target* as previsões são efetuadas em simultâneo. As experiências *single target*, correram em duas ferramentas de *AutoML*, o *AutoGluon* (Erickson et al., 2020) e o *H2O AutoML* (Cook, 2016). Estas ferramentas foram configuradas para escolher para cada *fold*, de forma automática, o modelo ótimo, bem como os seus hiperparâmetros, utilizando o *MAE* como métrica a otimizar. No que concerne às experiências *multi target*, foi utilizada a ferramenta *AutoKeras* (Jin et al., 2019) e uma *MLP*, também conhecida como *Deep FeedForward Neural Network* (*DFFN*) (Goodfellow et al., 2016), que utilizou uma configuração idêntica à utilizada em (Matos et al., 2019a, 2019b, 2021), com a salvaguarda de que a presente *DFFN* contém três *targets*. É de realçar que todas as experiências foram executadas com uma *cross-validation* interna e externa de 10 *folds*.

3.4 AVALIAÇÃO

A fase de avaliação pressupõe a avaliação dos modelos no que toca à qualidade de resposta aos objetivos de negócio propostos, traduzindo-se no cálculo de métricas entre os valores previstos e os valores reais.

Com o intuito de obter uma avaliação robusta, como referido anteriormente, todas as abordagens foram avaliadas utilizando um método de *cross-validation* externa de 10 *folds*. Para medir a qualidade dos resultados obtidos, as métricas *MAE* e *NMAE* foram calculadas para cada *fold*. O *NMAE*, é uma função matemática que normaliza o *MAE* pelo intervalo de valores do *target* no conjunto de teste, resultando numa percentagem independente de escala e de fácil compreensão, sendo representado pela seguinte fórmula (Oliveira et al., 2017):

$$NMAE = \frac{MAE}{(y_{max} - y_{min})} \quad (2)$$

na qual o y_{max} e o y_{min} representam os valores máximos e mínimos do *target* no conjunto de teste, respetivamente. É de notar que melhores previsões são indicadas por valores mais baixos de *NMAE* e *MAE*, ou seja, o valor perfeito é zero.

Por forma a conceder uma melhor visão sobre a qualidade das previsões, para o melhor *fold* de cada requisito, é efetuada a curva *REC*, bem como o gráfico da reta de regressão. Cada curva *REC* traça a percentagem de exemplos corretamente previstos (eixo y) para uma determinada tolerância ao erro (eixo x). A reta de regressão apresenta os *scatter-plots* dos valores reais (eixo x), com os valores previstos (eixo y).

3.4.1 Requisito 1: Previsão do Rendimento dos Artigos

No que diz respeito ao requisito da previsão do rendimento dos artigos, os resultados obtidos na experiência efetuada apresentam-se na Tabela 7.

Tabela 7: Resultados obtidos no Requisito 1: Previsão do Rendimento dos Artigos

Fold	MAE	NMAE	Melhor Modelo
1	9.96	9.97%	S.E.-AllModels
2	10.07	10.11%	S.E.-AllModels
3	9.8	9.83%	S.E.-Best1000
4	9.8	9.8%	S.E.-AllModels
5	10.13	10.17%	S.E.-Best1000
6	10.13	10.16%	S.E.-AllModels
7	9.71	9.72%	S.E.-Best1000
8	9.85	9.88%	S.E.-Best1000
9	9.79	9.81%	S.E.-Best1000
10	10.11	10.14%	S.E.-Best1000
Média	9.94	9.96%	-

Pela análise da tabela dos resultados obtidos, percebe-se que o *fold* cujos resultados foram mais satisfatórios foi o 7, confirmando um *MAE* de 9.71 e um *NMAE* de 9.72%.

No lado esquerdo da Figura 15, é possível verificar, através do *scatter plot*, que os pontos previstos estão próximos da diagonal que representa a previsão ideal, sendo que, em alguns casos, estes até se sobrepõem evidenciando a qualidade das previsões. No que diz respeito ao lado direito da Figura 15, podemos perceber que com uma tolerância de 0.2 mais de 80% das previsões do rendimento dos artigos estão corretas.

Em suma, os resultados obtidos demonstram qualidade e representam uma funcionalidade a aplicar no protótipo a entregar à Somelos Tecidos, na medida em que irá ajudar a empresa a planear a quantidade de tecido a comprar, sabendo de antemão qual será o rendimento deste mesmo tecido.

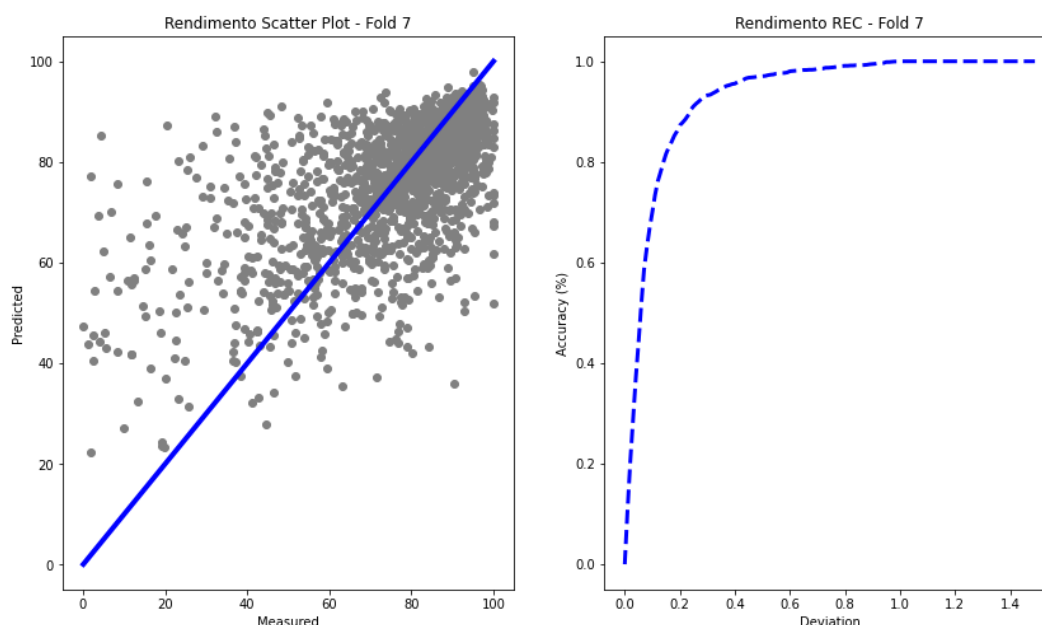


Figura 15: Curva REC e Reta de Regressão para o Fold7 do requisito 1

3.4.2 Requisito 2: Previsão de Defeitos nos Teares

A Tabela 8 resume os resultados obtidos na previsão. Para cada experiência foi detalhada a técnica de *ML* utilizada (Ferramenta), bem como a abordagem utilizada. Os resultados são apresentados sob a média dos resultados obtidos nos 10 *folds*, sendo que para o *MAE*, também foi incluído o desvio padrão dos 10 *folds* ($\pm s$, onde s é o valor do desvio padrão) e o modelo mais selecionado para cada abordagem.

Pela análise dos resultados obtidos, torna-se evidente que as abordagens que apresentaram um desempenho mais satisfatório foram executadas sob uma perspectiva *single target*. De forma geral, considerando as métricas *MAE* e *NMAE* a melhor abordagem foi a que utilizou a ferramenta H2O AutoML com a transformação logarítmica nos *targets* (abordagem B). Efetivamente, esta abordagem obteve os melhores resultados em termos de *NMAE* para as quebras de teia e de trama, obtendo valores muito próximos da melhor abordagem na rebentadela de fios. Os valores obtidos para cada *fold* encontram-se anexos.

Para efeitos de demonstração, a Figura 16 apresenta as curvas *REC* da 7ª iteração externa, para a previsão dos 3 *targets*, onde é possível verificar, por exemplo, que mais de 70% das previsões de quebras de teia são acertadas quando se utiliza uma tolerância de apenas 0,4. No que diz respeito à Figura 17, pode notar-se que as previsões para as quebras de teia e de trama estão próximas da diagonal ideal, verificando-se, em alguns pontos, uma sobreposição entre os valores reais e previstos. Fator que não se verifica para a rebentadela de fios, porém, através da Figura 16 percebe-se que, com uma tolerância de 0,4, as previsões para a rebentadela de fios seriam mais de 80% acertadas, denotando a qualidade das previsões obtidas.

Tabela 8: Resultados obtidos no Requisito 2: Previsão de Defeitos nos Teares

Objetivo	Ferramenta	Abordagem	MAE	NMAE	Modelo mais selecionado	
Quebra de Teias	H2O	A	14.11 ± 0.77	8.18%	GBM	
		B	13.93 ± 1.02	8.07%	GBM	
	AutoGluon	A	14.30 ± 0.76	8.28%	S.E.-AllModels	
		B	14.17 ± 0.93	8.22%	S.E.-AllModels	
	Keras	A	17.82 ± 1.55	10.19%	DFFN	
		B	18.51 ± 1.58	10.58%	DFFN	
	AutoKeras	A	17.97 ± 2.23	10.33%	DFFN	
		B	19.20 ± 2.09	10.97%	DFFN	
	Quebra de Tramas	H2O	A	10.96 ± 0.81	6.59%	GBM
			B	10.65 ± 0.98	6.47%	S.E.-Best1000
AutoGluon		A	11.25 ± 1.02	6.79%	S.E.-AllModels	
		B	10.77 ± 1.13	6.50%	S.E.-AllModels	
Keras		A	15.05 ± 2.58	8.99%	DFFN	
		B	14.69 ± 2.35	8.64%	DFFN	
AutoKeras		A	13.27 ± 1.33	8.03%	DFFN	
		B	15.29 ± 1.23	9.09%	DFFN	
Rebentadela de Fios		H2O	A	0.69 ± 0.08	7.93%	GBM
			B	0.68 ± 0.09	7.79%	XGBoost
	AutoGluon	A	0.68 ± 0.12	7.68%	S.E.-AllModels	
		B	0.68 ± 0.08	7.81%	S.E.-AllModels	
	Keras	A	0.83 ± 0.12	8.74%	DFFN	
		B	0.76 ± 0.11	7.70%	DFFN	
	AutoKeras	A	0.84 ± 0.12	8.55%	DFFN	
		B	0.73 ± 0.06	7.52%	DFFN	

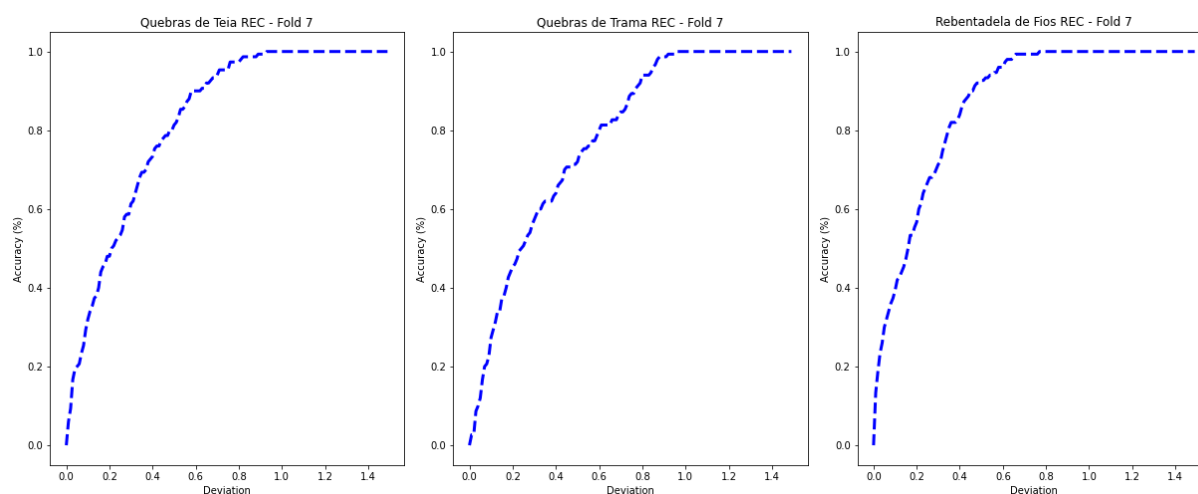


Figura 16: Curva REC para a quebra de teia (esquerda), quebra de trama (meio) e rebentadela de fios (direita).

Para demonstrar ainda mais o valor dos modelos selecionados (ferramenta *H2O* e transformação logarítmica), aplicamos uma abordagem *Sensitivity Analysis eXplainable Artificial Intelligence (SA XAI)* aos resultados da 7ª iteração da validação externa, nomeadamente a *SA 1-D*, tal como implementada pelo pacote *rminer* (Cortez, 2020; Cortez & Embrechts, 2013). Para esta iteração, a ferramenta *AutoML* selecionou um modelo *GBM*.

À esquerda da Figura 18 é ilustrada a relevância das variáveis de entrada (total de 20 características de entrada). Por exemplo, o *input* mais influente está relacionado com o número

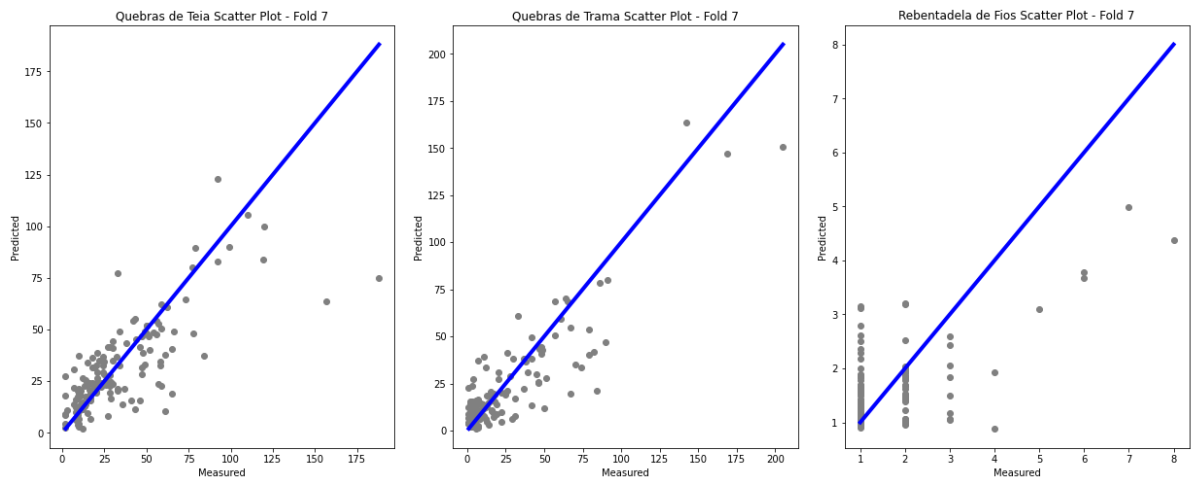


Figura 17: Retas de Regressão do melhor modelo para a quebra de teia (esquerda), quebra de trama (meio) e rebentadela de fios (direita).

de fios na trama por polegadas (relevância total de 14%). Além disso, os 9 principais *inputs* representam 66% da influência total no modelo *GBM*. Quanto ao lado direito da Figura 18, esta demonstra as curvas da *Variable Effect Characteristic (VEC)* para as 5 variáveis de entrada mais relevantes. A curva *VEC* mostra a influência média de uma entrada no *target* ao variar o seu intervalo através de $L = 7$ níveis distintos.

O gráfico revela claramente que o atributo mais influente (Número de fios trama por *rappord*) produz a maior mudança no *output* da *GBM* (impactando assim mais no modelo). Os conhecimentos *SA XAI* obtidos foram fornecidos aos peritos de produção têxtil, que confirmaram que, tanto a influência do *input* mencionado como os efeitos do mesmo, estão de acordo com os seus conhecimentos empíricos.

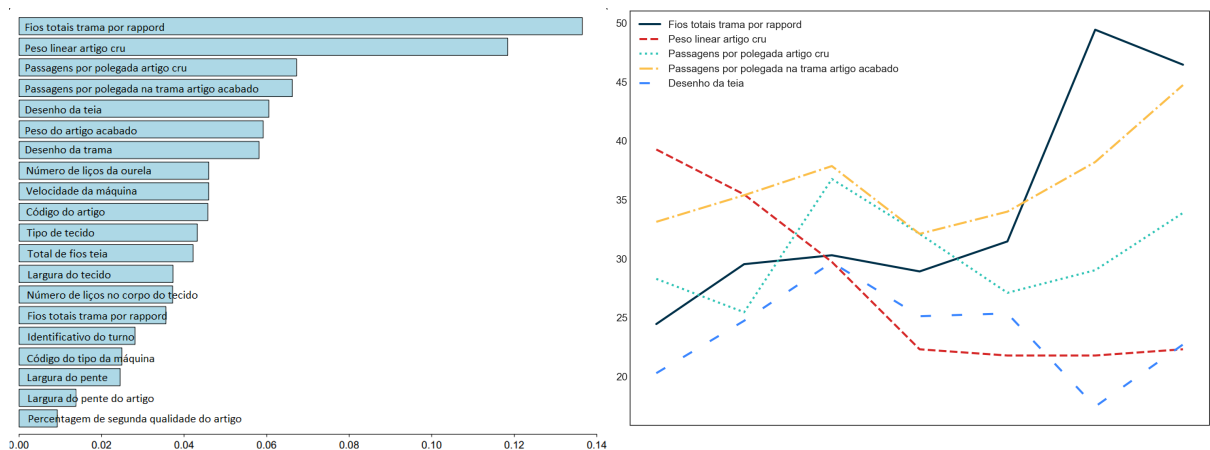


Figura 18: Conhecimento retirado através do *SA XAI* aplicado no modelo *GBM*: Importância das variáveis de entrada (esquerda) e curvas *VEC* dos atributos mais relevantes (direita).

3.4.3 Requisito 3: Previsão de Entrega de Matérias-Primas

Com o intuito de resumir o processo de experiências efetuado, a Tabela 9 apresenta os resultados obtidos. De realçar que a abordagem A se refere às experiências efetuadas sem a aplicação da função logarítmica no *target*, enquanto a abordagem B é relativa às experiências efetuadas com a aplicação da função supramencionada.

Tabela 9: Resultados obtidos no Requisito 3: Previsão de Entrega de Matérias-Primas

Fold	Abordagem A			Abordagem B		
	MAE	NMAE	Melhor Modelo	MAE	NMAE	Melhor Modelo
1	17.30	5.47%	S.E.-BestOfFamily	17.32	5.48%	XGBoost
2	16.18	7.97%	GBM	17.63	8.68%	XGBoost
3	14.97	7.64%	GBM	15.10	7.70%	XGBoost
4	15.21	4.74%	S.E.-AllModels	14.78	4.60%	XGBoost
5	15.02	7.82%	S.E.-BestOfFamily	15.33	7.99%	GBM
6	13.60	9.38%	S.E.-BestOfFamily	14.71	10.14%	GBM
7	16.91	12.25%	S.E.-BestOfFamily	17.36	12.58%	XGBoost
8	14.40	11.25%	S.E.-BestOfFamily	12.80	10%	GBM
9	15.71	11.99%	GBM	15.05	11.49%	GBM
10	14.65	9.58%	S.E.-BestOfFamily	12.59	8.23%	GBM
Média	15.40	8.81%	-	15.27	8.69%	-

Pela análise da tabela, é possível identificar que a abordagem que melhor responde ao desafio do requisito é a abordagem B, com um MAE médio de 15.27 dias, correspondendo a um NMAE médio de 8.68%, podendo também identificar que o *fold* que obteve melhores resultados foi o 4.

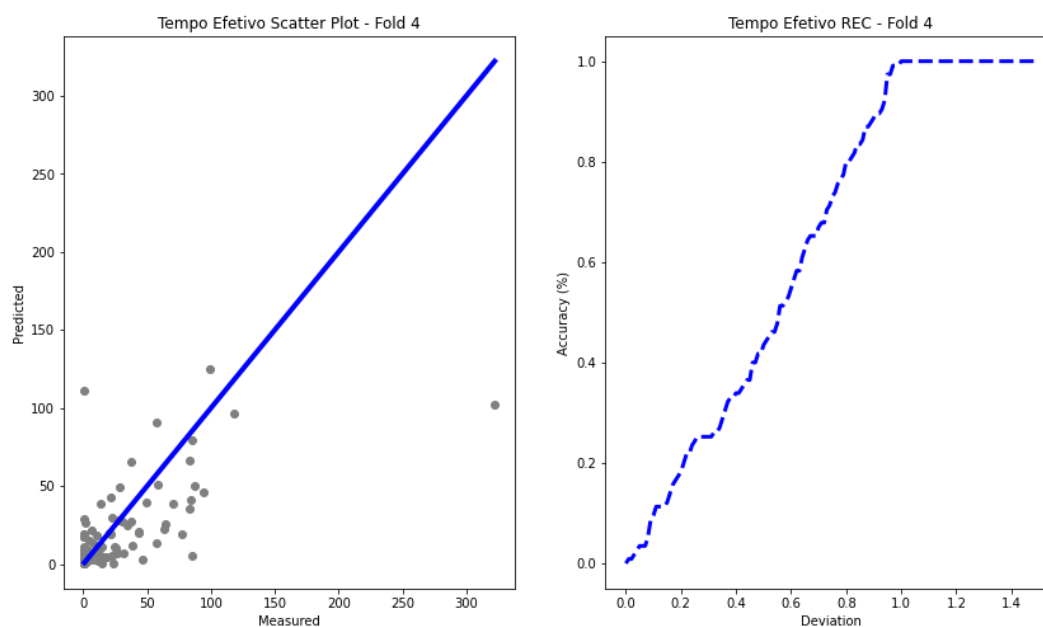


Figura 19: Curva REC e Reta de Regressão para o Fold7 do requisito 1

À esquerda da Figura 19, o *scatter plot* permite-nos identificar a qualidade do modelo na previsão de valores mais baixos, verificando-se mesmo sobreposição de alguns pontos previstos

com a diagonal ideal. Enquanto que pela análise da direita, percebemos que a fraca qualidade do modelo na previsão de valores mais elevados é impactante na qualidade global do modelo.

3.4.4 Requisito 4: Previsão de Tempos de Operação (Urdissagem, Engomagem e Remetagem)

No concernente ao requisito 4, a Tabela 10 apresenta para cada objetivo de previsão o MAE médio, bem como o desvio padrão associado, o NMAE e o modelo mais selecionado no conjunto dos 10 *fold*s.

Tabela 10: Resultados obtidos no Requisito 4: Previsão de Tempos de Operação (Urdissagem, Engomagem e Remetagem)

Requisito	MAE	NMAE	Modelo mais selecionado
Urdissagem	0.53 ± 0.07	2.46%	GBM
Engomagem	0.10 ± 0.01	1.20%	GBM
Remetagem	0.68 ± 0.10	3.76%	GBM

A evidência de erros inferiores a 4% corroboram o interesse e o sucesso deste objetivo de previsão. Para cada *target*, foi selecionado o *fold* que apresentou melhores resultados para formar a curva REC e a reta de regressão. A Figura 20 apresenta as curvas REC de cada um dos *targets* no seu melhor *fold*, enquanto a Figura 21 ilustra os *scatter plots* no mesmo *fold*.

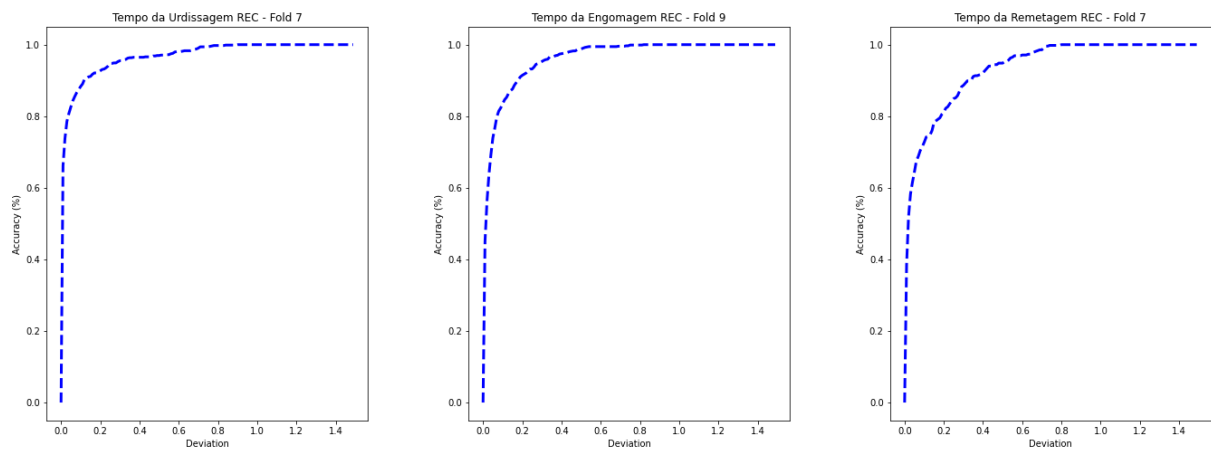


Figura 20: Curva REC para o tempo da urdissagem (esquerda), da engomagem (meio) e da remetagem (direita).

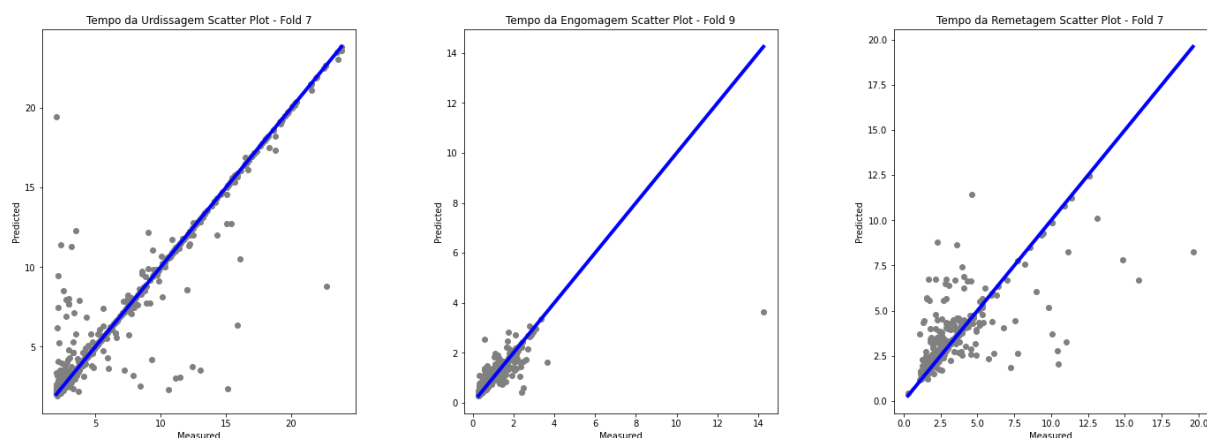


Figura 21: Reta de Regressão do melhor modelo para o tempo da urdissagem (esquerda), da engomagem (meio) e da remetagem (direita).

Pela análise das Figuras 20 e 21, percebe-se que, no que diz respeito à urdissagem, com uma tolerância de 0.2, cerca de 90% das previsões são acertadas e que a diagonal ideal de previsões se sobrepõe em muitos pontos previstos pelo algoritmo em toda a sua extensão, relatando a qualidade do modelo para este *fold*. Já no atinente à engomagem, uma tolerância de 0.2 também confere previsões que são 90% acertadas, e que a diagonal ideal se sobrepõe aos pontos previstos, também evidenciando o sucesso deste objetivo de previsão. Por sua vez, na remetagem, com uma tolerância de 0.2, 80% das previsões são acertadas, sendo que, apesar de haver uma maior dispersão de pontos previstos, ao longo da sua extensão, a diagonal ideal de previsão está sobreposta em vários pontos de previsão.

PROTÓTIPO

Após a realização do caso de estudo, e, com o intuito de desenvolver um protótipo passível de ser aplicado na organização, que contemplasse as quatro necessidades de previsão como um todo, foi tido em conta o volume crescendo de dados que a transformação digital sobre o paradigma da Indústria 4.0 acarreta, sendo que, dessa forma, o código foi adaptado para a linguagem *Pyspark*, uma biblioteca *Spark* escrita em *Python* que permite executar aplicações em paralelo e num *cluster* distribuído.

Deste modo, para a construção do protótipo, procedeu-se à ligação entre o *Data Warehouse* (DW) e o *Kedro*. No *Kedro*, encontram-se estruturadas as *pipelines* relativas ao pré-processamento, à divisão em treino/teste/validação, bem como a etapa de *encoding*. Para o treino dos modelos, criou-se uma *pipeline* que utiliza a ferramenta de *AutoML* do H2O, adaptada a *Pyspark*, o *Sparkling Water*. Após os modelos serem treinados, os resultados da previsão são enviados para a organização através de uma *API*. A arquitetura tecnológica presente na Figura 22, ilustra o processo supramencionado.

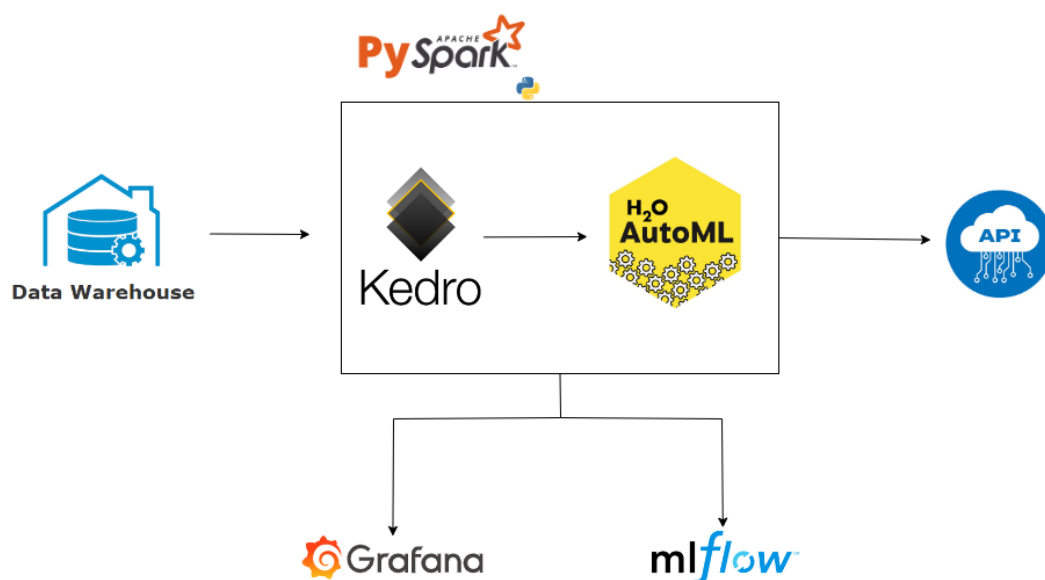


Figura 22: Arquitetura tecnológica do protótipo

No *Kedro* é possível fazer o treino de modelos para um requisito específico. Recorrendo ao *plugin Kedro-viz* é possível visualizar a *pipeline* implementada para cada requisito, desde o processo de carregamento de dados, passando pelo pré-processamento de dados, o *encoding*, a fase de separação em treino/teste/validação, bem como a fase de treino dos modelos. A Figura 23 apresenta a estrutura do protótipo no que diz respeito às *pipelines* implementadas no protótipo, sendo resultado da execução do *plugin Kedro-viz*.

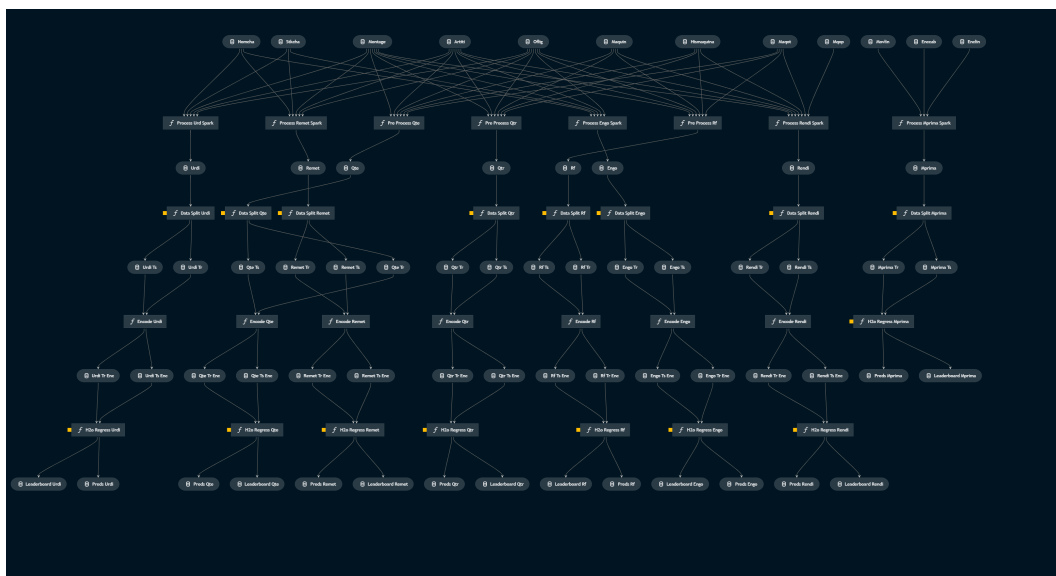


Figura 23: Pipelines do *Kedro*

Com o objetivo de monitorizar os modelos de previsão, implementou-se o *Mlflow*, uma ferramenta que permite o versionamento dos modelos treinados, registrando os seus parâmetros e métricas, a comparação de modelos, bem como a sua transição entre os estados de produção, estágio e arquivo. A Figura 24 apresenta um exemplo da plataforma com os modelos registrados para cada requisito, restantes imagens relativas ao funcionamento da ferramenta podem ser encontradas anexas a este documento.

Registered Models

Share and serve machine learning models. [Learn more](#)

Create Model

Name	Latest Version	Staging	Production	Last Modified
Rendim	Version 23	-	Version 23	2022-05-30 17:15:14
Qte_count	Version 5	-	Version 5	2022-05-30 17:56:41
Qtr_count	Version 3	Version 3	Version 2	2022-05-31 14:11:37
RF_count	Version 2	-	Version 2	2022-05-26 15:46:18
tempoEfetivo	Version 5	-	Version 2	2022-06-01 09:24:28
Urdi	Version 3	-	Version 3	2022-05-27 10:12:17
Engo	Version 3	-	Version 2	2022-05-27 12:52:28
Remet	Version 3	-	Version 2	2022-05-27 16:43:52

< Page 1 > 10 / page

Figura 24: Modelos registrados no *Mlflow*

Para facilitar o controlo dos *nodes* implementados no *Kedro*, foi feita a integração entre o *Kedro*, o *Mlflow* e o *Grafana*, permitindo a visualização de *dashboards* mais amigáveis, onde é possível, por exemplo, controlar o tempo de execução de cada componente da *pipeline*. Assim, a implementação do *Grafana* permitiu controlar métricas e registos, bem como monitorizar todas as *pipelines* implementadas, apresentando-as através de *dashboards*. Um exemplo de uma *dashboard* relativa à execução da *pipeline* da urdissagem no protótipo pode ser encontrada na Figura 25.

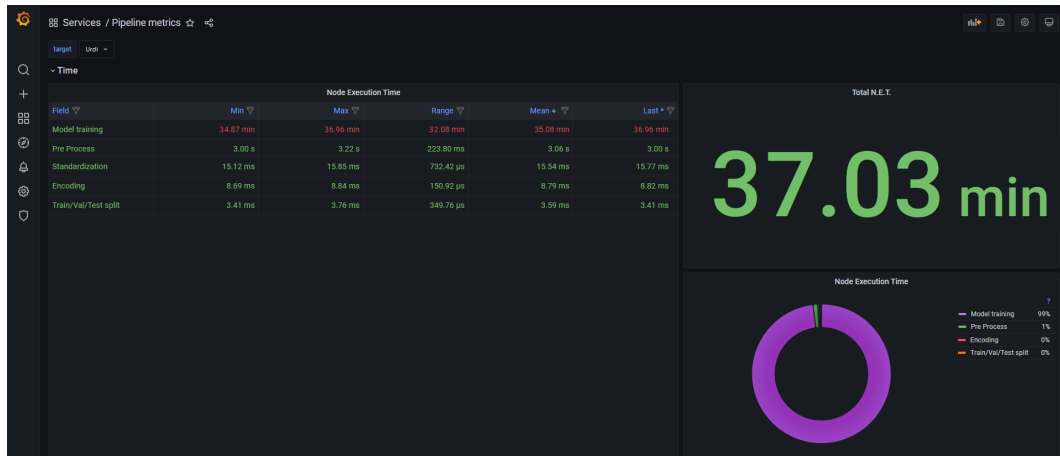


Figura 25: Visualização de *pipelines* no *Grafana*

Em suma, o processo tecnológico aplicado neste protótipo inicia-se com a transmissão dos dados do *DW* ao *Kedro*, onde os dados vão sofrer um conjunto de operações estipuladas na *pipeline*, em *pyspark*, antes de serem passados para o *Sparkling Water*, onde os modelos serão treinados. Os resultados da previsão serão passados para a organização via *API*. A implementação do *Mlflow* e do *Grafana* facilita o controlo e monitorização das *pipelines* e dos modelos à Somelos, auxiliando o processo de planeamento e controlo da produção.

CONCLUSÕES

O último capítulo da dissertação, a conclusão, tem como objetivo expor uma visão geral de todo o trabalho desenvolvido, com especial foco para as conclusões retiradas do projeto desenvolvido. Deste modo, este capítulo subdivide-se em três secções, a primeira visa recapitular o trabalho realizado no desenrolar da dissertação, a segunda secção avalia os contributos alcançados face aos previamente estipulados através da apresentação dos resultados, enquanto a terceira secção descreve que procedimentos podem ser feitos, no sentido de dar continuidade ao projeto.

5.1 SÍNTESE DO TRABALHO EFETUADO

A presente dissertação inicia-se com uma contextualização entre a área de incidência do projeto *PPC4.0* e do *ML* na *ITV*, através de um enquadramento e dos objetivos e resultados esperados. Apresentando também a metodologia adotada no desenrolar da parte prática, o *CRISP-DM*.

Em seguida, o capítulo de revisão de literatura iniciou-se com a exposição da estratégia de pesquisa bibliográfica adotada, passando para a exploração do conceito de *BA* e, explicando através do gráfico de *Gartner* (Figura 2), os seus constituintes: *Descriptive Analytics*, *Diagnostic Analytics*, *Predictive Analytics* e *Prescriptive Analytics*. Após a introdução do conceito de *Business Analytics*, os termos de *IA*, *ML* e *DL* foram introduzidos, começando por uma visão geral do conceito de *IA*, explicando a aprendizagem supervisionada, não supervisionada e reforçada como componentes de *ML*, expondo o conceito de *AutoML*, bem como exemplificando alguns algoritmos de *ML*. Por sua vez, o conceito de *DL* foi introduzido e foi exemplificada uma rede de *DL* “manual”, assim como o *AutoKeras*, uma forma de aplicar *DL* através de uma ferramenta de *AutoML*. Antes de o capítulo terminar, é feito um escrutínio sobre o trabalho de outros autores no que concerne à aplicação de *ML* no setor têxtil, sendo, também, apresentadas as ferramentas tecnológicas que serviram de base ao projeto.

O terceiro capítulo serve-se da metodologia do *CRISP-DM* para resolver os objetivos de previsão estipulados pela *Somelos*, com o intuito de melhorar o planeamento e controlo da produção da organização. Deste modo, para cada requisito é apresentada a estratégia adotada, bem como os resultados obtidos, os gráficos da curva *REC* e da linha de regressão que mostram o sucesso dos modelos.

Por fim, o quarto capítulo, apresenta o protótipo desenvolvido para posterior implementação na empresa, que tem em conta os requisitos da monitorização dos modelos de previsão e da transmissão dos resultados das previsões através de uma *API*, tendo em conta o volume crescendo de dados. Deste modo, os requisitos estipulados são tratados como um todo, e podem ser acedidos e treinados de forma mais fácil pelos profissionais do setor têxtil que serão os utilizadores finais desta solução.

5.2 DISCUSSÃO

Com o término da presente dissertação, ao fazer um ponto de situação entre os requisitos estipulados no começo do projeto e a sua concretização, podemos concluir que a sua implementação foi bem sucedida, na medida em que os objetivos propostos foram concretizados e que o objetivo de apoiar a tomada de decisão da Somelos, no que diz respeito ao planeamento e controlo da produção, foi concretizado. Para além dos requisitos de previsão definidos, foi, também, desenvolvido um protótipo preparado para o refrescamento dos modelos e o aumento do volume de dados, bem como respondendo aos dois requisitos não relacionados com a previsão: a transmissão dos resultados das previsões através de uma *API* e a monitorização dos modelos de previsão.

No domínio do projeto *PPC4.0*, a revisão de literatura efetuada permitiu perceber que estratégias de *AutoML* não são um método muito adotado para a resolução de problemas de *ML* no setor têxtil. Nesse sentido, a presente dissertação e os seus resultados, para além de trazerem valor para a organização e de suportarem a tomada de decisão no que diz respeito ao planeamento e controlo da produção da Somelos, têm relevância no sentido científico na medida em que corroboram a teoria de que é possível resolver problemas de regressão no setor têxtil partindo de modelos de *AutoML*. Para além disso, num dos requisitos de previsão foi aplicada uma abordagem *SA XAI*, que permitiu extrair ainda mais valor dos modelos de *ML*.

Em termos de contributos científicos, foi aceite, para publicação, um artigo de investigação, resultante do trabalho desenvolvido na presente dissertação, na conferência “*International Conference on Knowledge Based and Intelligent information and Engineering Systems (KES2022)*”, sendo esta indexada (*Scopus, CORE B*).

Por último, convém realçar que os resultados obtidos foram mostrados aos peritos de produção têxtil, que forneceram um *feedback* positivo, achando-os muito interessantes para apoiar o planeamento da produção.

5.3 TRABALHO FUTURO

A solução desenvolvida na presente dissertação constitui um rumo para a aplicação de *ML* no suporte ao planeamento e controlo da produção no setor têxtil, neste caso específico, na Somelos. Assim, os aspetos em consideração para a continuação deste trabalho estão interligados com as fases posteriores do projeto em que este trabalho se insere, o *PPC4.0*.

Deste modo, por forma a dar fiabilidade ao protótipo desenvolvido, os próximos passos estarão relacionados com os testes do mesmo, numa fase inicial num ambiente controlado, onde será possível perceber a capacidade do protótipo de se adaptar ao aparecimento de novos dados e, numa fase posterior, em ambiente real, onde será possível, para além de verificar a qualidade dos modelos em “*real-time*”, perceber de que forma os profissionais do setor têxtil se adaptam em torno da solução desenvolvida. Por sua vez, no que diz respeito à componente experimental, para verificar se os resultados obtidos na presente dissertação têm consistência seria interessante testar as abordagens propostas em mais *datasets* do domínio em questão e, ainda, experimentar outras ferramentas de *AutoML* na etapa de modelação para a comparação de resultados.

Por fim, o projeto deverá terminar com a fase de *deploy* descrita no *CRISP-DM* (Figura 1), seguida de uma fase de validação empresarial. Será após a implementação do protótipo que a Somelos começará, verdadeiramente, a tirar vantagens da solução final e a planear/controlar a produção com maior confiabilidade.

REFERÊNCIAS BIBLIOGRÁFICAS

- Akyol, U., Tüfekci, P., Kahveci, K., & Cihan, A. (2015, aug). A model for predicting drying time period of wool yarn bobbins using computational intelligence techniques. *Textile Research Journal*, 85(13), 1367–1380. Retrieved from <http://journals.sagepub.com/doi/10.1177/0040517514553879> doi: 10.1177/0040517514553879
- Almetwally, A. A., Mourad, M., & Mohammed, A. E. E. (2013). A Study of Yarn Breaks on Warping Machines. *Life Science Journal*. Retrieved from http://www.lifesciencesite.com/ljsj/life1001/016_12676life1001_108_114.pdf
- ArunKumar, K., Kalaga, D. V., Sai Kumar, C. M., Chilkoor, G., Kawaji, M., & Brenza, T. M. (2021). Forecasting the dynamics of cumulative covid-19 cases (confirmed, recovered and deaths) for top-16 countries using statistical machine learning models: Auto-regressive integrated moving average (arima) and seasonal auto-regressive integrated moving average (sarima). *Applied Soft Computing*, 103, 107161. Retrieved from <https://www.sciencedirect.com/science/article/pii/S1568494621000843> doi: <https://doi.org/10.1016/j.asoc.2021.107161>
- Bengio, Y., Goodfellow, I., & Courville, A. (2017). *Deep learning* (Vol. 1). MIT press Cambridge, MA, USA.
- Bonaccorso, G. (2017). *Machine learning algorithms*. Packt Publishing Ltd.
- Brown, R. B. (2006). *Doing Your Dissertation in Business and Management: The Reality of Research and Writing*. Sage Publications.
- Brugman, S. (2019). *pandas-profiling: Exploratory Data Analysis for Python*. <https://github.com/pandas-profiling/pandas-profiling>.
- Campos, G. O., Zimek, A., Sander, J., Campello, R. J. G. B., Micenková, B., Schubert, E., ... Houle, M. E. (2016). On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data Mining and Knowledge Discovery*, 30(4), 891–927. Retrieved from <https://doi.org/10.1007/s10618-015-0444-8>
- Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T. P., Shearer, C., & Wirth, R. (2000). *Crisp-dm 1.0: Step-by-step data mining guide*.
- Charbuty, B., & Abdulazeez, A. (2021). Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, 2(01), 20–28.
- Cioară, I., Cioară, L., & Onofrei, E. (2004, feb). Forecast of Yarn Breakages During The Weaving Process. *Research Journal of Textile and Apparel*, 8(1), 20–24. Retrieved from <https://www>

.emerald.com/insight/content/doi/10.1108/RJTA-08-01-2004-B003/full/html doi: 10.1108/RJTA-08-01-2004-B003

Cook, D. (2016). *Practical machine learning with h2o: powerful, scalable techniques for deep learning and ai*. O'Reilly Media. Retrieved from <https://books.google.pt/books?id=C5amDQAAQBAJ>

Cortez, P. (2020). rminer: Data mining classification and regression methods [Computer software manual]. Retrieved from <https://CRAN.R-project.org/package=rminer> (R package version 1.4.6)

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., & Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4), 547-553. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0167923609001377> (Smart Business Networks: Concepts and Empirical Evidence) doi: <https://doi.org/10.1016/j.dss.2009.05.016>

Cortez, P., & Embrechts, M. J. (2013). Using sensitivity analysis and visualization techniques to open black box data mining models. *Inf. Sci.*, 225, 1-17. doi: 10.1016/j.ins.2012.10.039

Das, B., Ishtiaque, S., Rengasamy, R., & Hati, S. (2011, aug). Prediction of Weavability of Warp Yarns: A Critical Review. *Research Journal of Textile and Apparel*, 15(3), 38-49. Retrieved from <https://www.emerald.com/insight/content/doi/10.1108/RJTA-15-03-2011-B005/full/html> doi: 10.1108/RJTA-15-03-2011-B005

DGAE. (2018). Indústria Têxtil e Vestuário. *Sinopse*, 26.

Ekmekci, P. E., & Arda, B. (2020). *Artificial intelligence and bioethics*. Springer International Publishing. doi: 10.1007/978-3-030-52448-7

El Naqa, I., & Murphy, M. J. (2015). What Is Machine Learning? In *Machine learning in radiation oncology*. Cham: Springer International Publishing. doi: 10.1007/978-3-319-18305-3_1

El Morr, C., & Ali-Hassan, H. (2019). Descriptive, predictive, and prescriptive analytics. In *Analytics in healthcare: A practical introduction* (pp. 31-55). Cham: Springer International Publishing. Retrieved from https://doi.org/10.1007/978-3-030-04506-7_3 doi: 10.1007/978-3-030-04506-7_3

El-Tarfawy, S. Y. (2016). Prediction of Weft Breaks in Air Jet Weaving Machine by Artificial Neural Network. *Journal of American Science*. Retrieved from http://www.jofamericanscience.org/journals/am-sci/am120516/05_30403jas120516_46_51.pdf doi: 10.7537/marsjas12051605

Eltayib, H. E., Ali, A. H. M., & Ishag, I. A. (2016). The Prediction of Tear Strength of plain weave fabric Using Linear Regression Models. *International Journal of Advanced Engineering Research and Science*, 3(11), 151-154. Retrieved from http://ijaers.com/uploads/issue_files/25IJAERS-NOV-2016-30-ThePredictionofTearStrengthofplainweavefabricUsingLinearRegressionModels.pdf doi: 10.22161/ijaers/3.11.25

- Eraslan, E. (2009). The Estimation of Product Standard Time by Artificial Neural Networks in the Molding Industry. *Mathematical Problems in Engineering*, 2009, 1–12. Retrieved from <http://www.hindawi.com/journals/mpe/2009/527452/> doi: 10.1155/2009/527452
- Erickson, N., Mueller, J., Shirkov, A., Zhang, H., Larroy, P., Li, M., & Smola, A. (2020, mar). AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data. Retrieved from <http://arxiv.org/abs/2003.06505>
- Evans, J. R., & Lindner, C. H. (2013). Business Analytics. In Wiley (Ed.), *Big data, big analytics* (pp. 99–125).
- Fallahpour, A., Wong, K. Y., Olugu, E. U., & Musa, S. N. (2017, aug). A Predictive Integrated Genetic-Based Model for Supplier Evaluation and Selection. *International Journal of Fuzzy Systems*, 19(4), 1041–1057. Retrieved from <http://link.springer.com/10.1007/s40815-017-0324-z> doi: 10.1007/s40815-017-0324-z
- Ferreira, L., Pilastrri, A., Sousa, V., Romano, F., & Cortez, P. (2021). Prediction of maintenance equipment failures using automated machine learning. In H. Yin et al. (Eds.), *Intelligent data engineering and automated learning – ideal 2021* (pp. 259–267). Cham: Springer International Publishing. doi: 10.1007/978-3-030-91608-4_26
- Gibert, K., Sànchez–Marrè, M., & Izquierdo, J. (2016, dec). A survey on pre-processing techniques: Relevant issues in the context of environmental data mining. *AI Communications*, 29(6). doi: 10.3233/AIC-160710
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. Retrieved from <http://www.deeplearningbook.org>
- H2O.ai. (2021). *H2O AutoML*. Retrieved from <https://docs.h2o.ai/h2o/latest-stable/h2o-docs/automl.html>
- Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... Oliphant, T. E. (2020, September). Array programming with NumPy. *Nature*, 585(7825), 357–362. Retrieved from <https://doi.org/10.1038/s41586-020-2649-2> doi: 10.1038/s41586-020-2649-2
- Hope, T. M. (2020). Chapter 4 - linear regression. In A. Mechelli & S. Vieira (Eds.), *Machine learning* (p. 67-81). Academic Press. Retrieved from <https://www.sciencedirect.com/science/article/pii/B9780128157398000043> doi: <https://doi.org/10.1016/B978-0-12-815739-8.00004-3>
- Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3), 90–95. doi: 10.1109/MCSE.2007.55
- Hutter, J., Frank and Kotthoff, & Lars and Vanschoren. (2019). *Automated Machine Learning* (F. Hutter, L. Kotthoff, & J. Vanschoren, Eds.). Cham: Springer International Publishing. Retrieved from <http://link.springer.com/10.1007/978-3-030-05318-5> doi: 10.1007/978-3-030-05318-5

- Jaouachi, B., & Khedher, F. (2015, apr). Evaluation of Sewed Thread Consumption of Jean Trousers Using Neural Network and Regression Methods. *Fibres and Textiles in Eastern Europe*, 23(3(111)), 91–96. Retrieved from <http://220.indexcopernicus.com/abstracted.php?level=5&ICID=1152518> doi: 10.5604/12303666.1152518
- Jin, H., Song, Q., & Hu, X. (2019, 7). Auto-keras: An efficient neural architecture search system. In (p. 1946-1956). ACM. doi: 10.1145/3292500.3330648
- Kutschenreiter-Praszkiewicz, I. (2008, apr). Application of artificial neural network for determination of standard time in machining. *Journal of Intelligent Manufacturing*, 19(2), 233–240. Retrieved from <http://link.springer.com/10.1007/s10845-008-0076-6> doi: 10.1007/s10845-008-0076-6
- Le, T. T., Fu, W., & Moore, J. H. (2020, jan). Scaling tree-based automated machine learning to biomedical big data with a feature set selector. *Bioinformatics*, 36(1). doi: 10.1093/bioinformatics/btz470
- LeCun, Y., Bengio, Y., & Hinton, G. (2015, 5). Deep learning. *Nature*, 521, 436-444. doi: 10.1038/nature14539
- LeDell, E., & Poirier, S. (2020). H2O AutoML: Scalable Automatic Machine Learning.
- Lee, C., & Lin, J. (2018, apr). Improve Quality and Efficiency of Textile Process using Data-driven Machine Learning in Industry 4.0. *International Journal of Technology and Engineering Studies*, 4(2). Retrieved from <https://kkgpublications.com/ijtes-volume4-issue2-article4/> doi: 10.20469/ijtes.4.10004-2
- Lutkevich, B., & Burns, E. (2021, oct). *Business Analytics*. Retrieved 2021-11-04, from <https://searchbusinessanalytics.techtarget.com/definition/business-analytics-BA>
- Maher, M., & Sakr, S. (2019, mar). SmartML: A Meta Learning-Based Framework for Automated Selection and Hyperparameter Tuning for Machine Learning Algorithms.
- Martínez-Plumed, F., Contreras-Ochando, L., Ferri, C., Hernández-Orallo, J., Kull, M., Lachiche, N., ... Flach, P. (2021). Crisp-dm twenty years later: From data mining processes to data science trajectories. *IEEE Transactions on Knowledge and Data Engineering*, 33(8), 3048-3061. doi: 10.1109/TKDE.2019.2962680
- Mathew, V., Toby, T., Singh, V., Rao, B. M., & Kumar, M. G. (2017). Prediction of remaining useful lifetime (rul) of turbofan engine using machine learning. In *2017 IEEE international conference on circuits and systems (iccs)* (p. 306-311). doi: 10.1109/ICCS1.2017.8326010
- Matos, L. M., Cortez, P., & Mendes, R. (2020). *Cane - Categorical Attribute traNsformation Environment*. Retrieved from <https://pypi.org/project/cane/>
- Matos, L. M., Cortez, P., Mendes, R., & Moreau, A. (2018). A comparison of data-driven approaches for mobile marketing user conversion prediction. In R. Jardim-Gonçalves, J. P. Mendonça, V. Jotsov, M. Marques, J. Martins, & R. E. Bierwolf (Eds.), *9th IEEE international conference on*

- intelligent systems, IS 2018, funchal, madeira, portugal, september 25-27, 2018* (pp. 140–146). IEEE. doi: 10.1109/IS.2018.8710472
- Matos, L. M., Cortez, P., Mendes, R., & Moreau, A. (2019a). Using deep learning for mobile marketing user conversion prediction. In *International joint conference on neural networks, IJCNN 2019 budapest, hungary, july 14-19, 2019* (pp. 1–8). IEEE. Retrieved from <https://doi.org/10.1109/IJCNN.2019.8851888> doi: 10.1109/IJCNN.2019.8851888
- Matos, L. M., Cortez, P., Mendes, R. C., & Moreau, A. (2019b). Using deep learning for ordinal classification of mobile marketing user conversion. In H. Yin, D. Camacho, P. Tiño, A. J. Tallón-Ballesteros, R. Menezes, & R. Allmendinger (Eds.), *Intelligent data engineering and automated learning - IDEAL 2019 - 20th international conference, manchester, uk, november 14-16, 2019, proceedings, part I* (Vol. 11871, pp. 60–67). Springer. doi: 10.1007/978-3-030-33607-3_7
- Matos, L. M., Domingues, A., Moreira, G., Cortez, P., & Pilastrri, A. L. (2021). A comparison of machine learning approaches for predicting in-car display production quality. In *Intelligent data engineering and automated learning - IDEAL 2021 - 22nd international conference, IDEAL 2021, manchester, uk, november 25-27, 2021, proceedings* (Vol. 13113, pp. 3–11). Springer. doi: 10.1007/978-3-030-91608-4_1
- Matusiak, M. (2015). Application of Artificial Neural Networks to Predict the Air Permeability of Woven Fabrics. *Fibres Textiles in Eastern Europe*.
- McCarthy, J. (1989). Artificial intelligence, logic and formalizing common sense. In *Philosophical logic and artificial intelligence* (pp. 161–190). Springer.
- McKinney, W. (2010). Data structures for statistical computing in python. In S. van der Walt & J. Millman (Eds.), *Proceedings of the 9th python in science conference* (p. 51 - 56).
- Mondal, B. (2020). Artificial Intelligence: State of the Art. In (pp. 389–425). Retrieved from http://link.springer.com/10.1007/978-3-030-32644-9_32 doi: 10.1007/978-3-030-32644-9_32
- Mwasiagi, J. I., Huang, X., & Wang, X. (2008, feb). Performance of neural network algorithms during the prediction of yarn breaking elongation. *Fibers and Polymers*, 9(1), 80–86. Retrieved from <http://link.springer.com/10.1007/s12221-008-0013-5> doi: 10.1007/s12221-008-0013-5
- Neapolitan, R. E., & Jiang, X. (2018). *Artificial intelligence: With an introduction to machine learning*. CRC Press.
- Nurwaha, D., & Wang, X. H. (2012). Using intelligent control systems to predict textile yarn quality. *Fibres and Textiles in Eastern Europe*, 90(1), 23–27.
- Oliveira, N., Cortez, P., & Areal, N. (2017). The impact of microblogging data for stock market prediction: Using twitter to predict returns, volatility, trading volume and survey sentiment indices. *Expert Syst. Appl.*, 73, 125-144.

- Onaran, E., & Yanık, S. (2020). Predicting Cycle Times in Textile Manufacturing Using Artificial Neural Network. In (pp. 305–312). doi: 10.1007/978-3-030-23756-1_38
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... others (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct), 2825–2830.
- Pinheiro, V. (2016). *Indústria 4.0 a Quarta Revolução industrial*. Retrieved 06-01-2022, from https://www.compete2020.gov.pt/destaques/detalhe/Industria_4ponto0
- Pisner, D. A., & Schnyer, D. M. (2020). Chapter 6 - support vector machine. In A. Mechelli & S. Vieira (Eds.), *Machine learning* (p. 101-121). Academic Press. Retrieved from <https://www.sciencedirect.com/science/article/pii/B978012815739800067> doi: <https://doi.org/10.1016/B978-0-12-815739-8.00006-7>
- Plotly. (2015). *Collaborative data science*. <https://plot.ly>. Plotly Technologies Inc.
- Ribeiro, R., Pilastri, A., Moura, C., Rodrigues, F., Rocha, R., & Cortez, P. (2020). Predicting the Tear Strength of Woven Fabrics Via Automated Machine Learning: An Application of the CRISP-DM Methodology. In *Proceedings of the 22nd international conference on enterprise information systems* (pp. 548–555). SCITEPRESS - Science and Technology Publications. Retrieved from <http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0009411205480555> doi: 10.5220/0009411205480555
- Schaap, P. (2020, mar). *The 4 levels of data maturity that you should absolutely know about*. Retrieved 2021-11-04, from <https://computd.nl/nl/demystification/4-levels-of-data-maturity/>
- Seabold, S., & Perktold, J. (2010). statsmodels: Econometric and statistical modeling with python. In *9th python in science conference*.
- Seboka, N. (2021). Regression model development for showing relation between mechanical yarn stretch (%) in sizing and Warp yarn breakage (cmpx) in looms using ANOVA model Regression model developmen Regression model development for showing relation between mechanical yarn . *Ethiopian Journal of Textile and Apparel (EJTA)*, 1(2), 2020. Retrieved from <https://www.researchgate.net/publication/348324833>
- Shao, Y., Ji, X., Zheng, M., & Chen, C. (2021, jul). Prediction of Standard Time of the Sewing Process using a Support Vector Machine with Particle Swarm Optimization. *Autex Research Journal*. Retrieved from <https://www.sciendo.com/article/10.2478/aut-2021-0037> doi: 10.2478/aut-2021-0037
- Shobha, G., & Rangaswamy, S. (2018). Machine Learning.. doi: 10.1016/bs.host.2018.07.004
- Singh, S., Parmar, K. S., Kumar, J., & Makkhan, S. J. S. (2020). Development of new hybrid model of discrete wavelet decomposition and autoregressive integrated moving average (arima) models in application to one month forecast the casualties cases of covid-19. *Chaos*,

- Solitons Fractals*, 135, 109866. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0960077920302666> doi: <https://doi.org/10.1016/j.chaos.2020.109866>
- Sontakke, S. A., Lohokare, J., Dani, R., & Shivagaje, P. (2019). *Classification of cardiocography signals using machine learning*. doi: 10.1007/978-3-030-01057-7_35
- Turing, A. M. (2009). Computing Machinery and Intelligence. In *Parsing the turing test* (pp. 23–65). Dordrecht: Springer Netherlands. Retrieved from http://link.springer.com/10.1007/978-1-4020-6710-5_3 doi: 10.1007/978-1-4020-6710-5_3
- Usuga Cadavid, J. P., Lamouri, S., Grabot, B., Pellerin, R., & Fortin, A. (2020, aug). Machine learning applied in production planning and control: a state-of-the-art in the era of industry 4.0. *Journal of Intelligent Manufacturing*, 31(6). doi: 10.1007/s10845-019-01531-7
- Vesset, D. (2018). Diagnostic analytics 101: Why did it happen.
- Wang, C.-C., Chien, C.-H., & Trappey, A. J. C. (2021, jul). On the Application of ARIMA and LSTM to Predict Order Demand Based on Short Lead Time and On-Time Delivery Requirements. *Processes*, 9(7). doi: 10.3390/pr9071157
- Wang, S.-C. (2003). *Artificial neural network*. Springer US. doi: 10.1007/978-1-4615-0377-4_5
- Waskom, M., Botvinnik, O., O’Kane, D., Hobson, P., Lukauskas, S., Gemperline, D. C., ... Qalieh, A. (2017, September). Seaborn. Retrieved from <https://doi.org/10.5281/zenodo.883859> doi: 10.5281/zenodo.883859
- Wirth, R. (2000). Crisp-dm: Towards a standard process model for data mining. In *Proceedings of the fourth international conference on the practical application of knowledge discovery and data mining* (pp. 29–39).
- Yao, G., Guo, J., & Zhou, Y. (2005, mar). Predicting the Warp Breakage Rate in Weaving by Neural Network Techniques. *Textile Research Journal*, 75(3), 274–278. Retrieved from <http://journals.sagepub.com/doi/10.1177/004051750507500314> doi: 10.1177/004051750507500314
- Zaharia, M. A., Chen, A., Davidson, A., Ghodsi, A., Hong, S. A., Konwinski, A., ... Zumar, C. (2018). Accelerating the machine learning lifecycle with mlflow. *IEEE Data Eng. Bull.*, 41, 39-45.
- Zakaria, N. (2011). Sizing system for functional clothing - uniforms for school children. *Indian Journal of Fibre and Textile Research*, 36(4), 348–357.
- Zhang, F., & O’Donnell, L. J. (2020). Chapter 7 - support vector regression. In A. Mechelli & S. Vieira (Eds.), *Machine learning* (p. 123-140). Academic Press. Retrieved from <https://www.sciencedirect.com/science/article/pii/B978012815739800079> doi: <https://doi.org/10.1016/B978-0-12-815739-8.00007-9>

Zhang, W., Wu, C., Zhong, H., Li, Y., & Wang, L. (2021). Prediction of undrained shear strength using extreme gradient boosting and random forest based on bayesian optimization. *Geoscience Frontiers*, 12(1), 469-477. Retrieved from <https://www.sciencedirect.com/science/article/pii/S1674987120300669> doi: <https://doi.org/10.1016/j.gsf.2020.03.007>

Zhao Bo. (2010, jun). The prediction of warp breakage rate of weaving by considering sized yarn quality using artificial neural network theory. In *2010 international conference on computer design and applications* (pp. V2-526-V2-529). IEEE. Retrieved from <http://ieeexplore.ieee.org/document/5541337/> doi: 10.1109/ICCDA.2010.5541337

Zöller, M.-A., & Huber, M. F. (2021, jan). Benchmark and Survey of Automated Machine Learning Frameworks. *Journal of Artificial Intelligence Research*, 70. doi: 10.1613/jair.1.11854

Zuo, C., Qian, J., Feng, S., Yin, W., Li, Y., Fan, P., ... Chen, Q. (2022, 12). Deep learning in optical metrology: a review. *Light: Science Applications*, 11, 39. doi: 10.1038/s41377-022-00714

-x

ANEXOS

A.1 RESULTADOS DAS PREVISÕES POR *fold* EM CADA ABORDAGEM/FERRAMENTA UTILIZADA NO REQUISITO 2Tabela 11: Resultados obtidos por *fold* para a abordagem A através da ferramenta H2O AutoML

Folds	Target	MAE	NMAE	Target	MAE	NMAE	Target	MAE	NMAE
1		15.31	8.10		11.33	6.00		0.63	5.72
2		13.24	7.12		11.60	5.58		0.79	5.63
3		14.60	10.21		10.13	4.54		0.65	9.30
4		14.00	9.03		11.19	7.08		0.68	6.19
5		14.71	6.54		12.41	5.28		0.72	10.24
6	Qte_count	14.00	9.86	Qtr_count	10.01	9.35	RF_count	0.80	6.70
7		12.62	6.78		9.86	4.83		0.56	8.025
8		13.76	7.36		10.38	6.97		0.58	11.69
9		14.92	7.89		11.81	7.62		0.70	10.07
10		13.93	8.88		10.84	8.68		0.80	5.71
Média		14.11	8.18		10.96	6.59		0.69	7.93

Tabela 12: Resultados obtidos por *fold* para a abordagem B através da ferramenta H2O AutoML

Folds	Target	MAE	NMAE	Target	MAE	NMAE	Target	MAE	NMAE
1		14.44	7.64		9.51	5.03		0.64	5.84
2		13.31	7.15		11.47	5.52		0.68	4.85
3		15.01	10.50		9.59	4.30		0.62	8.81
4		13.73	8.86		11.90	7.53		0.66	6.00
5		15.29	6.79		11.78	5.01		0.71	10.10
6	Qte_count	13.86	9.76	Qtr_count	10.46	9.78	RF_count	0.78	6.47
7		11.46	6.16		9.12	4.47		0.53	7.60
8		14.53	7.77		10.17	6.82		0.58	11.67
9		14.24	7.53		11.62	7.50		0.73	10.49
10		13.43	8.56		10.90	8.72		0.85	6.09
Média		13.93	8.07		10.65	6.47		0.68	7.79

Tabela 13: Resultados obtidos por *fold* para a abordagem A através da ferramenta AutoGluon

Folds	Target	MAE	NMAE	Target	MAE	NMAE	Target	MAE	NMAE
1		15.57	8.24		11.11	5.88		0.62	5.66
2		13.48	7.25		11.77	5.66		0.79	5.68
3		15.12	10.58		9.68	4.34		0.61	8.76
4		14.15	9.13		12.86	8.14		0.78	7.09
5		15.13	6.73		12.61	5.37		0.61	8.76
6	Qte_count	13.73	9.67	Qtr_count	10.15	9.48	RF_count	0.76	6.33
7		13.15	7.07		10.23	5.01		0.51	7.24
8		13.80	7.38		10.90	7.32		0.54	10.8
9		14.69	7.77		12.14	7.83		0.71	10.1
10		14.17	9.02		11.04	8.83		0.89	6.38
Média		14.30	8.28		11.25	6.79		0.68	7.68

Tabela 14: Resultados obtidos por *fold* para a abordagem B através da ferramenta AutoGluon

Folds	Target	MAE	NMAE	Target	MAE	NMAE	Target	MAE	NMAE
1		14.85	7.86		9.30	4.92		0.64	5.83
2		13.53	7.27		11.79	5.67		0.77	5.51
3		14.87	10.4		9.84	4.41		0.66	9.44
4		14.37	9.27		11.74	7.43		0.67	6.11
5		15.53	6.90		12.81	5.45		0.71	10.09
6	Qte_count	14.02	9.88	Qtr_count	10.05	9.39	RF_count	0.76	6.32
7		11.88	6.39		9.38	4.60		0.54	7.68
8		14.05	7.51		10.25	6.88		0.57	11.42
9		14.24	7.53		11.65	7.52		0.69	9.88
10		14.35	9.14		10.85	8.68		0.82	5.82
Média		14.17	8.22		10.77	6.50		0.68	7.81

Tabela 15: Resultados obtidos por *fold* para a abordagem A através da rede manual DFFN

Folds	Target	MAE	NMAE	Target	MAE	NMAE	Target	MAE	NMAE
1		18.95	9.97		17.65	9.29		0.76	6.34
2		17.61	9.37		14.03	6.71		0.91	6.04
3		17.62	12.24		13.54	6.05		0.86	10.81
4		18.16	11.57		19.68	12.38		0.80	6.69
5		20.11	8.90		17.09	7.24		0.78	9.80
6	Qte_count	16.39	11.30	Qtr_count	12.79	11.84	RF_count	0.93	7.17
7		14.30	7.60		11.03	5.38		0.59	7.41
8		19.35	10.29		17.21	11.47		1.07	17.79
9		18.41	9.69		14.54	9.32		0.76	9.48
10		17.30	10.95		12.93	10.26		0.87	5.83
Média		17.82	10.19		15.05	8.99		0.83	8.74

Tabela 16: Resultados obtidos por *fold* para a abordagem B através da rede manual DFFN

Folds	Target	MAE	NMAE	Target	MAE	NMAE	Target	MAE	NMAE
1		18.66	9.82		13.95	7.34		0.67	5.58
2		18.99	10.10		15.24	7.29		0.92	6.13
3		18.72	13.00		18.64	8.32		0.86	10.69
4		17.70	11.27		15.83	9.95		0.71	5.94
5		20.34	9.00		17.56	7.44		0.77	9.56
6	Qte_count	16.19	11.16	Qtr_count	12.23	11.32	RF_count	0.81	6.24
7		15.45	8.22		11.06	5.40		0.58	7.19
8		18.98	10.10		13.76	9.17		0.61	10.24
9		20.75	10.92		16.43	10.53		0.75	9.43
10		19.29	12.21		12.19	9.67		0.89	5.96
Média		18.51	10.58		14.69	8.64		0.76	7.70

Tabela 17: Resultados obtidos por *fold* para a abordagem A através da ferramenta AutoKeras

Folds	Target	MAE	NMAE	Target	MAE	NMAE	Target	MAE	NMAE
1		15.16	7.98		10.49	5.52		0.84	6.99
2		20.88	11.11		15.41	7.37		0.92	6.16
3		19.49	13.53		12.58	5.62		0.82	10.23
4		14.77	9.41		13.89	8.74		0.88	7.31
5		20.94	9.26		14.93	6.33		0.89	11.19
6	Qte_count	19.83	13.68	Qtr_count	13.29	12.31	RF_count	0.87	6.68
7		14.48	7.70		10.47	5.11		0.60	7.53
8		16.29	8.66		14.17	9.44		0.68	11.39
9		18.50	9.74		12.92	8.28		0.88	11.06
10		19.35	12.25		14.52	11.53		1.05	6.99
Média		17.97	10.33		13.27	8.02		0.84	8.55

Tabela 18: Resultados obtidos por *fold* para a abordagem B através da ferramenta AutoKeras

Folds	Target	MAE	NMAE	Target	MAE	NMAE	Target	MAE	NMAE
1		21.18	11.15		15.44	8.12		0.71	5.94
2		18.96	10.09		16.58	7.93		0.81	5.40
3		21.83	15.16		16.36	7.30		0.74	9.30
4		15.78	10.05		14.61	9.19		0.64	5.32
5		19.51	8.63		15.10	6.40		0.67	8.39
6	Qte_count	17.16	11.83	Qtr_count	13.24	12.26	RF_count	0.75	5.77
7		19.80	10.53		14.82	7.23		0.66	8.21
8		19.06	10.14		17.26	11.50		0.69	11.56
9		22.56	11.88		15.95	10.23		0.78	9.76
10		16.16	10.23		13.56	10.77		0.84	5.57
Média		19.20	10.97		15.29	9.09		0.73	7.52

A.2 RESULTADOS DAS PREVISÕES POR *fold* EM CADA *target* UTILIZADO NO REQUISITO 4

Tabela 19: Resultados obtidos por *fold* para o Requisito 4: Previsão de Tempos de Operação (Urdissagem, Engomagem e Remetagem)

Folds	Target	MAE	NMAE	Target	MAE	NMAE	Target	MAE	NMAE
1	Urdissagem	0.54	2.48	Engomagem	0.12	0.92	Remetagem	0.85	4.60
2		0.49	2.56		0.11	0.83		0.64	3.78
3		0.55	2.53		0.10	0.81		0.82	3.97
4		0.63	3.05		0.09	0.71		0.67	3.93
5		0.45	2.06		0.11	1.96		0.69	3.36
6		0.60	2.77		0.09	2.03		0.61	3.71
7		0.42	1.91		0.10	1.10		0.60	3.09
8		0.57	2.61		0.09	2.18		0.52	3.47
9		0.52	2.40		0.10	0.69		0.69	3.87
10		0.48	2.19		0.10	0.74		0.68	3.82
Média		0.53	2.46		0.10	1.20		0.68	3.76

A.3 SCREENSHOTS DO PROTÓTIPO

No que diz respeito ao *Mlflow* a Figura 24 representa os modelos registados, a Figura 26 representa a mudança de estado para o *target* relativo ao número de quebras de trama. Por outro lado, as Figuras 27 e 28 representam o processo de comparação de modelos para o rendimento dos artigos.

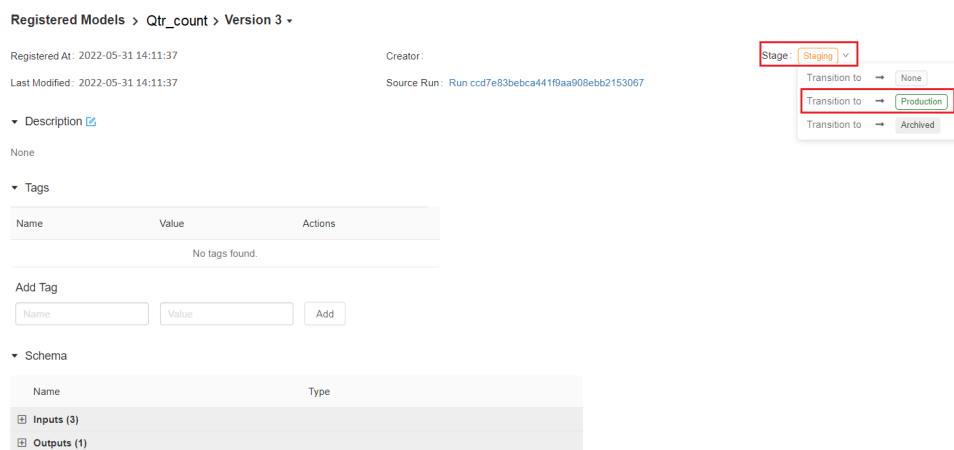


Figura 26: Transição de estados no *Mlflow*

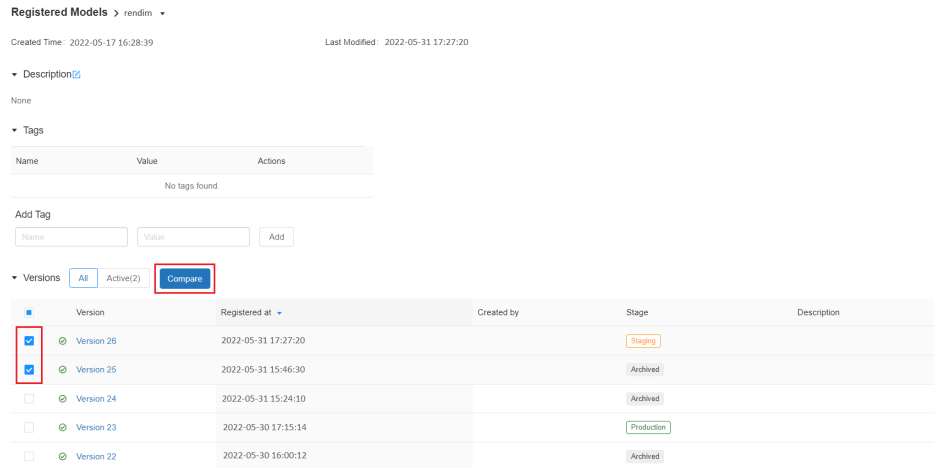


Figura 27: Selecionar modelos para comparar no Mlflow

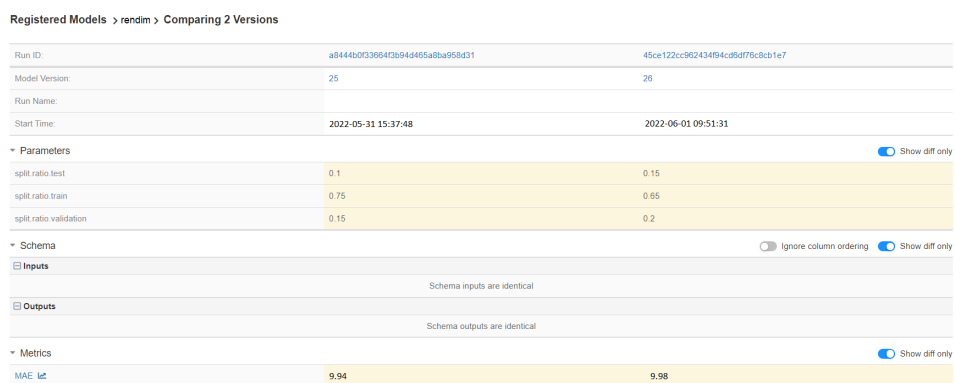


Figura 28: Comparação de modelos no Mlflow