



Aprendizagem Computacional no apoio à promoção do sucesso e redução do abandono escolar Projeto II

Miguel Alexandre Salvado Magueijo

N.º 20191374

Orientadores

Professora Doutora Ana Paula Neves Ferreira da Silva

Professor Doutor Arlindo Ferreira da Silva

Trabalho de Projeto II apresentado à Escola Superior de Tecnologia do Instituto Politécnico de Castelo Branco para cumprimento dos requisitos necessários à obtenção do grau de Licenciado em Informática e Multimédia, realizada sob a orientação científica do Professor Adjunto Doutor Ana Paula Neves Ferreira da Silva e coorientação do Professor Adjunto Doutor Arlindo Ferreira da Silva, do Instituto Politécnico de Castelo Branco.

Setembro 2024

Composição do júri

Presidente do júri

Doutor, Carlos Manuel de Oliveira Alves

Professor Adjunto da Escola Superior de Tecnologia de Castelo Branco

Orientador

Doutor, Ana Paula Neves Ferreira da Silva

Professor Adjunto da Escola Superior de Tecnologia de Castelo Branco

Arguente

Doutor, José Carlos Meireles Monteiro Metrôlho

Professor Coordenador da Escola Superior de Tecnologia de Castelo Branco

Agradecimentos

Num primeiro agradecimento, expresso a minha profunda gratidão à minha família por todo o apoio ao longo dos cinco anos de estudo no Instituto Politécnico de Castelo Branco. Um agradecimento especial aos meus pais, Carlos Magueijo e Elsa Magueijo, por acreditarem em mim e por me proporcionarem a oportunidade de dar continuidade aos estudos, especialmente, permitindo a elaboração deste projeto.

Em segundo lugar, quero deixar o meu agradecimento aos professores Ana Paula Silva e Arlindo Silva por toda a orientação e ajuda ao longo do desenvolvimento do projeto, nomeadamente às diversas respostas a todas as minhas perguntas e curiosidades. Deixo o meu agradecimento profundo a ambos professores por despertarem a minha curiosidade pela área de Inteligência Artificial e por estarem sempre disponíveis em conversar sobre temas e problemas da mesma.

Um adicional agradecimento pela oportunidade concedida pela professora Ana Paula Silva em permitir com que tivesse sido eu a desenvolver a parte de *Machine Learning* do projeto REVUP. Foi com grande entusiasmo e orgulho que contribuí com o meu conhecimento e habilidades para este projeto. Esta experiência não só permitiu aprofundar o meu conhecimento na área como também resultou com que me tivesse sido feita uma oferta de trabalho antes da conclusão da minha licenciatura. Estou imensamente grato à professora Ana Paula Silva por me ter proporcionado esta oportunidade.

Adicionalmente, gostaria de agradecer a todos os meus amigos que me acompanharam ao longo desta jornada académica. Sem o apoio e companhia deles, este percurso teria sido significativamente mais difícil.

Por fim, gostaria de estender os meus agradecimentos a todos os professores do IPCB com quem tive a oportunidade de interagir. Não só contribuíram para o meu desenvolvimento pessoal, como também se mostraram sempre disponíveis em ajudar e oferecer oportunidades em participar e organizar atividades que foram únicas e extremamente enriquecedoras.

Resumo

Perante um recente aumento no número de desistências no ensino superior, principalmente após o primeiro ano letivo, as instituições de ensino têm procurado soluções para combater o abandono e o insucesso escolar. Em resposta a este desafio, o Instituto Politécnico de Castelo Branco (IPCB) propôs, em 2024, o desenvolvimento e implementação do projeto REVUP. O objetivo do projeto é promover o sucesso escolar e combater o abandono e insucesso dos seus alunos. O REVUP propõe ainda, a utilização de modelos de *Machine Learning* (ML) para sinalizar os alunos em risco, para que possam ser tomadas medidas de ajuda a esses alunos.

De forma a ser possível utilizar modelos de ML para a predição do nível de risco de um aluno do IPCB, foi necessário estudar as áreas de Inteligência Artificial (IA) e ML dedicadas à criação de sistemas capazes de desempenhar tarefas humanas. Assim, foram estudados os diferentes tipos de aprendizagem computacional e identificado aquele que melhor se enquadra nos objetivos do trabalho, a aprendizagem supervisionada.

Para se perceber se é possível aplicar técnicas de ML dedicadas a ajudar as instituições do ensino superior a combater o abandono e insucesso escolar, foi realizada uma análise da investigação existente desta área. A análise permitiu constatar, de forma positiva, que o seu uso é possível e para determinados momentos de predição, os resultados obtidos são muito bons. Além disso, verificou-se que a investigação existente foi abordada por autores de todo o mundo.

Para o desenvolvimento deste trabalho, o autor recebeu dados históricos de alunos do IPCB, com o objetivo de treinar modelos de ML capazes de predizer o nível de risco relativo ao abandono e insucesso escolar. Adicionalmente, foi possível conceber um processo complexo de pré-processamento a ser aplicado aos dados de treino, o qual poderá ser reutilizado em previsões futuras. Esta reutilização permite evitar erros associados ao desconhecimento de atributos e/ou valores que o modelo de ML não tenha sido treinado para reconhecer. Além disso, foi demonstrado que a predição do nível de risco é possível; no entanto, deve ser devidamente validada pelos docentes, uma vez que o desempenho dos modelos de ML ficou aquém do esperado.

A realização deste projeto permitiu analisar o desempenho de modelos de ML para diferentes classes de predição e conceber um algoritmo de sinalização do nível de risco dos alunos do IPCB que se matriculam pela primeira vez. Além disso foi ainda possível desenvolver duas aplicações web onde os utilizadores têm de interagir diretamente com os modelos de ML.

Palavras-chave

Ensino superior; Abandono escolar; Sucesso escolar; Nível de risco; *Machine Learning*; Classificação; Regressão.

Abstract

Recently it was observed an increase in the number of dropouts in higher education, especially after the first academic year, thus educational institutions have been looking for solutions to combat school dropout and failure. In response to this challenge, the Polytechnic Institute of Castelo Branco (IPCB) proposed the development and implementation of the REVUP project in 2024. The aim of the project is to promote academic success and combat student dropout and failure. REVUP also proposes the use of Machine Learning (ML) models to identify students at risk, so that measures can be taken to help these students.

In order to be able to use ML models to predict the risk level of an IPCB student, it was necessary to study the areas of Artificial Intelligence (AI) and ML dedicated to creating systems capable of performing human tasks. Thus, the different types of computer learning were studied and the one that best fits the objectives of the work, supervised learning, was identified.

To determine whether it is possible to apply ML techniques dedicated to helping higher education institutions combat school dropout and failure, an analysis of existing research in this area was carried out. The analysis showed that it is possible to use ML techniques and that the results obtained for certain prediction times are very good. In addition, it was found that the existing research was covered by authors from all over the world.

For the development of this work, the author received historical data from IPCB students, with the aim of training ML models capable of predicting the level of risk related to school dropout and failure. In addition, it was possible to design a complex pre-processing process to be applied to the training data, which can be reused in future predictions. This reuse makes it possible to avoid errors associated with not knowing attributes and/or values that the ML model has not been trained to recognize. In addition, it was shown that predicting the level of risk is possible; however, it must be duly validated by teachers, since the performance of the ML models was lower than expected.

This project made it possible to analyze the performance of ML models for different prediction classes and to design an algorithm to signal the risk level of IPCB students enrolling for the first time. It was also possible to develop two web applications where users are able to interact directly with the ML models.

Keywords

Higher education; School dropout; School success; Risk level; Machine Learning; Classification; Regression.

X

Índice geral

1.	Introdução	1
1.1.	Enquadramento	2
1.2.	Objetivos	3
1.3.	Planeamento do projeto	4
1.4.	Estrutura do relatório	4
2.	Inteligência artificial	6
2.1.	Machine Learning	8
2.1.1	Aprendizagem supervisionada	11
2.1.2.	Aprendizagem não supervisionada	14
2.1.3.	Aprendizagem por reforço	15
2.2.	Deep Learning	16
2.3.	Aquisição de dados	16
2.4.	Avaliação e métricas para problemas de classificação	17
3.	Estudo do estado da arte	22
3.1.	Metodologia e processo de pesquisa	22
3.1.1.	Propósito e objetivos	22
3.1.2.	Fontes de dados	23
3.1.3.	Estratégia de pesquisa	24
3.1.4.	Critérios de elegibilidade para análise	24
3.1.6.	Extração de dados e análise	25
3.2.	Análise dos artigos	27
3.3.	Discussão dos resultados	36
3.4.	Principais conclusões	41
4.	Tecnologias e ferramentas utilizadas	44
4.1.	Python	44
4.2.	Jupyter Notebook	44
4.3.	Scikit-learn	45
4.4.	Pandas	46
4.5.	NumPy	47
4.6.	Matplotlib	47
4.7.	Seaborn	48
4.8.	PyCharm	49

4.9. Webstorm.....	50
4.9. GitHub.....	51
4.11. Weka.....	52
4.12. SvelteKit e Typescript.....	53
4.13. FASTAPI	54
4.14. PostreSQL.....	55
4.15. Bibliotecas adicionais.....	56
5. <i>Datasets</i>	58
5.1. Recolha.....	58
5.2. Composição	60
5.3. Pré-processamento.....	64
5.3.1. Classes investigadas.....	73
5.4. Análise	77
6. Treino e avaliação dos modelos de ML.....	91
6.1. Processo e sua evolução	91
6.2. Resultados.....	96
6.2.1. Risco Original.....	99
6.2.2. Risco Binário	101
6.2.3. Risco Otimizado	104
6.2.4. Continua estudos.....	107
6.2.5. ECTS Realizados.....	110
6.2.6. Intervalo ECTS Realizados	112
6.3. Combinação de modelos de ML explorada para a predição do nível de risco	116
6.3.3. Otimização dos melhores modelos de ML.....	118
6.4. Reflexão dos resultados obtidos	121
7. Aplicações Desenvolvidas.....	125
7.1. Aplicação de demonstração de conceito.....	126
7.1.1 Arquitetura e ferramentas utilizadas	126
7.1.2. Ecrãs da aplicação.....	127
7.2. Aplicação para apoio a investigação em ML.....	135
7.2.1 Arquitetura e ferramentas utilizadas	146
7.1.2. Ecrãs da aplicação.....	149

8. Contribuição externa do autor na Digitalis	178
9. Conclusões.....	188
9.1. Trabalho futuro.....	190
Referências.....	192
Anexos.....	204
A. Resultados obtidos	204

Índice de figuras

Figura 1 – Projeto REVUP, proposta de sinalização dos alunos por diferentes níveis de risco.....	3
Figura 2 - As três áreas principais de IA	7
Figura 3 – Fluxograma da abordagem de resolução de um problema sem ML	9
Figura 4 - Fluxograma da abordagem de resolução de um problema com ML.....	10
Figura 5 - Adaptação da abordagem de um problema com ML	10
Figura 6 –Exemplificação de dados de treino para o problema <i>SPAM</i> de mensagens em ML.....	12
Figura 7 - Exemplo de agrupamentos de dados através do uso da técnica de aprendizagem não supervisionada	14
Figura 8 - Matriz de confusão e posições dos VP, VN, FP e FN	19
Figura 9 - Exemplo real de matriz de confusão para classificação de frutas.....	19
Figura 10 - Exemplo do cálculo da precisão e <i>recall</i>	21
Figura 11 - Gráfico de barras com o tipo de previsões feitas pelos estudos analisados	36
Figura 12 - Gráfico circular das categorias de dados mais utilizados e o seu número de ocorrências	38
Figura 13 - Momentos de previsão das várias investigações do estado da arte	39
Figura 14 - Número de usos por cada algoritmo de ML utilizado	40
Figura 15 - Algoritmos de ML referenciados como o melhor para o treino	41
Figura 16 – Exemplo de divisão por células de código e <i>output</i> da sua execução num ficheiro <i>Jupyter Notebook</i>	45
Figura 17 - Exemplo de um <i>DataFrame</i> da biblioteca pandas após ser carregado um ficheiro CSV	46
Figura 18 - Exemplos de estilos de gráficos que é possível criar usando a biblioteca <i>Matplotlib</i>	48
Figura 19 - Comparação do código necessário para um gráfico simples usando as bibliotecas <i>matplotlib</i> e <i>seaborn</i>	49
Figura 20 – <i>PyCharm Professional Edition</i> , visualização de um objeto <i>DataFrame</i> em tempo real	50
Figura 21 - Exemplo de procura e pré-visualização de elementos <i>div</i> de todos os ficheiros de um projeto <i>SvelteKit</i> aberto na ferramenta <i>WebStorm</i>	51
Figura 22 - Exemplo de alterações de um ficheiro entre a penúltima e última versão no <i>GitHub</i>	52
Figura 23 – Interface gráfica da aplicação <i>Weka</i>	53
Figura 24 - Ficheiro da biblioteca <i>SvelteKit</i> responsável pelo layout de todas as páginas	54
Figura 25 - Exemplo de uma função em <i>FASTAPI</i> associada à rota “/user/”	55
Figura 26 – Composição do processo de pré-processamento aplicado aos <i>datasets</i>	65
Figura 27 - Mapa de renomeação de colunas em formato JSON.....	66

Figura 28 - Mapa para normalizar instâncias com valores da Guiné-Bissau.....	67
Figura 29 - Normalização dos valores para o atributo referente ao curso inscrito	68
Figura 30 - Mapas de normalização para os valores dos atributos: qualificação académica e grupo profissional do pai e da mãe.....	68
Figura 31 - Mapa de codificação ordinal para a qualificação do pai e mãe	69
Figura 32 – Exemplificação do funcionamento do algoritmo Binary Encoding	71
Figura 33 - Resultados experimentais do uso da codificação <i>Ordinal Encoding</i> nos atributos nominais do <i>dataset</i>	72
Figura 34 - Resultados experimentais do uso da codificação <i>One Hot Encoding</i> aos atributos nominais do <i>dataset</i>	72
Figura 35 - Resultados experimentais do uso da codificação <i>Binary Encoding</i> aos atributos nominais do <i>dataset</i>	72
Figura 36 - Pseudocódigo do Risco Original e os valores possíveis	75
Figura 37 - Pseudocódigo do Risco Otimizado e os valores possíveis.....	75
Figura 38 - Pseudocódigo do Risco Binário e os valores possíveis.....	76
Figura 39 - Pseudocódigo de Intervalo ECTS Realizados e os valores possíveis ...	76
Figura 40 – Número de instâncias por escola do <i>dataset</i> combinado de todos os anos letivos	78
Figura 41 – Número de instâncias por escola em cada ano letivo	78
Figura 42 - Número de alunos que continuaram a estudar no IPCB para os quatro anos letivos em análise	79
Figura 43 - Número de alunos que continuaram a estudar no IPCB em cada ano letivo	80
Figura 44 - Número de alunos que continuaram a estudar no IPCB por escola....	80
Figura 45 - Número de alunos, por curso, que continuaram a estudar após término do 1º ano.....	81
Figura 46 - Número de alunos por tipo de ingresso que continuam a estudar	82
Figura 47 - Número de instâncias por nível de risco (Risco Otimizado).....	83
Figura 48 - Número de instâncias para cada nível de risco (Risco Otimizado) em cada ano letivo.....	84
Figura 49 - Número de instâncias por nível de risco (Risco Otimizado) para cada escola do IPCB	85
Figura 50 - Número de instâncias por nível de risco (Risco Oitmizado) e por tipo de ingresso	86
Figura 51 - Número de instâncias por nível de risco (Risco Original).....	87
Figura 52 - Número de instâncias por nível de risco (Risco Original) para cada tipo de ingresso	87
Figura 53 - Número de instâncias com risco e sem risco (Risco Binário).....	88
Figura 54 - Número de instâncias com risco e sem risco por tipo de ingresso	88
Figura 55 - Número de instâncias por intervalo de ECTS realizados.....	89
Figura 56 - Número de instâncias por intervalo de ECTS realizados para cada tipo	89

Figura 57 - Fluxo de funcionamento de um primeiro processo de treino de modelos de ML implementado	92
Figura 58 - Novo processo de treino, avaliação e predição	93
Figura 59 - Exemplo de configuração de um treino de modelos de ML para a predição de continuação dos estudos por parte do aluno.....	93
Figura 60 – Representação gráfica da técnica de treino validação cruzada (<i>Cross-Validation</i>) (adaptada de [136])	94
Figura 61 – Valores das métricas exatidão e F1 de um modelo <i>Random Forest</i> treinado corretamente sem presença de classes como atributos ou avaliação em instâncias duplicadas	97
Figura 62 - Valores das métricas exatidão e F1 de um modelo <i>Random Forest</i> treinado com a presença de uma classe nos atributos de treino.....	97
Figura 63 – Valores das métricas exatidão e F1 de um modelo <i>Random Forest</i> quando a sua avaliação é feita em instâncias duplicadas	98
Figura 64 - Pseudocódigo do algoritmo de predição de nível de risco composto por dois modelos de ML.....	117
Figura 65 - Conjunto de pesquisa definido com os valores possíveis para cada parâmetro dos algoritmos de ML <i>Gradient Boosting</i> (<i>parameters_for_gb</i>) e <i>LightGBM</i> (<i>parameters_for_lgbm</i>)	119
Figura 66 - Melhores combinações de hiperparâmetros encontrados para o algoritmo <i>Gradient Boosting</i> e <i>LightGBM</i>	120
Figura 67 – Regra criada pelo algoritmo OneR para a classe "continua_estudos" e a sua exatidão	122
Figura 68 - Regra criada pelo algoritmo OneR para a classe "continua_estudos", e a sua exatidão, após remoção dos atributos relativos ao distrito do aluno	122
Figura 69 - Regra criada pelo algoritmo OneR para a classe "continua_estudos", e a sua exatidão, após remoção de qualquer atributo referente à nacionalidade ou indicativo que é um aluno internacional.....	123
Figura 70 - Regra criada pelo algoritmo OneR para a classe "risco_otimizado", e a sua exatidão, após remoção de qualquer atributo referente à nacionalidade ou indicativo que é um aluno internacional.....	123
Figura 71 - Arquitetura da primeira aplicação	127
Figura 72 - Único ecrã da primeira aplicação, opção de seleção de um ficheiro.	128
Figura 73 – Primeira aplicação, utilizador escolhe um ficheiro	129
Figura 74 - Primeira aplicação, apresentação resultados de predição de um ficheiro	129
Figura 75 – Primeira aplicação, conteúdo da resposta de um pedido <i>HTTP (POST)</i> à <i>REST API</i> para realizar uma predição no contexto de envio de um ficheiro	130
Figura 76 – Primeira aplicação, código <i>TypeScript</i> que permite realizar o pedido de predição e contar o número de instâncias que foi atribuído cada risco possível	130
Figura 77 - Primeira aplicação, código <i>HTML/SvelteKit</i> que cria dinamicamente os elementos <i>HTML ("div")</i> com a contagem de instâncias para cada classe de risco	131

Figura 78 – Primeira aplicação, apresentação do formulário com todos os campos necessário para predizer o risco de um único aluno.....	132
Figura 79 – Primeira aplicação, resultado da realização de uma predição ao preencher o formulário com os dados de um aluno	133
Figura 80 – Primeira aplicação, parte da configuração gerada e exportada em formato <i>JSON</i> referente ao treino do modelo de ML integrado	134
Figura 81 – Primeira aplicação, código <i>HTML</i> (com <i>SvelteKit</i>) que gera automaticamente elementos de <i>input HTML</i> para cada atributo especificado no ficheiro de configuração do modelo de ML integrado	135
Figura 82 – Mapa com todas as classes criadas para processos de pré-processamento e predição, inclui herança e dependências	137
Figura 83 - Mapa de classes para o contexto de pré-processamento de treino ..	138
Figura 84 - Exemplo de uma configuração de pré-processamento de treino com todos os anos letivos.....	139
Figura 85 - Parte do ficheiro de configuração resultante da execução do processo de pré-processamento de treino. Esta configuração é posteriormente importada pelo processo de pré-processamento no contexto de predição	142
Figura 86 - Mapa de classes para o contexto de pré-processamento de treino de predição e processo de predição.....	143
Figura 87 - Código de inicialização de um objeto <i>PredictionPreprocess</i>	144
Figura 88 - Código da função "predict" da classe <i>PredictorProcess</i>	145
Figura 89 – Representação da arquitetura da segunda aplicação de forma simplificada	147
Figura 90 -Estrutura da base de dados a ser utilizada na segunda aplicação desenvolvida	148
Figura 91 - Segunda aplicação, página inicial de um utilizador sem autenticação	150
Figura 92 – Segunda aplicação, código do <i>hook</i> de <i>SvelteKit</i> com instruções para redirecionar utilizadores para a página inicial quando tentam aceder ecrãs protegidos por autenticação ou páginas de administrador sem os privilégios suficientes	151
Figura 93 – Segunda aplicação, página de login.....	152
Figura 94 – Segunda aplicação, login com credenciais incorretas	152
Figura 95 – Segunda aplicação, login de uma conta desativada.....	152
Figura 96 – Segunda aplicação, página inicial de um administrador	153
Figura 97 - Segunda aplicação, página inicial de um utilizador autenticado sem privilégios de administrador	154
Figura 98 – Segunda aplicação, menu suspenso quando o utilizador passa o rato por cima do seu nome de utilizado na barra de navegação.....	154
Figura 99 – Segunda aplicação, página de definições da conta	155
Figura 100 – Segunda aplicação, utilizador tenta alterar password com a atual errada.....	155
Figura 101 – Segunda aplicação, painel de administração.....	157
Figura 102 – Segunda aplicação, página de criação de um novo utilizador.....	158

Figura 103 – Segunda aplicação, mensagem de sucesso na criação de um utilizador	158
Figura 104 – Segunda aplicação, página de consulta dos utilizadores.....	159
Figura 105 – Segunda aplicação, funcionalidade de filtrar utilizadores pelo seu nome ou nome de utilizador (<i>username</i>).....	159
Figura 106 – Segunda aplicação, página de criação de um novo tipo de predição	160
Figura 107 – Segunda aplicação, página de adição de tipo de predição com pré-visualização do conteúdo do ficheiro selecionado e da predição a ser adicionada.....	161
Figura 108 – Segunda aplicação, mensagem de erro apresentada ao administrador quando seleciona um ficheiro de <i>metadata</i> de predição inválido	161
Figura 109 – Segunda aplicação, página para consultar os tipos de predições existentes	162
Figura 110 – Segunda aplicação, página de consulta de tipos de predição quando o administrador seleciona uma predição existente.....	163
Figura 111 – Segunda aplicação, <i>popup</i> de confirmação de eliminação de conteúdo	164
Figura 112 – Segunda aplicação, página de adição de um novo pré-processamento	165
Figura 113 – Segunda aplicação, página de adição de um novo pré-processamento após ser selecionado um ficheiro de configuração válido.....	166
Figura 114 – Segunda aplicação, página de consulta de processos de pré-processamentos	167
Figura 115 – Segunda aplicação, página de adição de um novo modelo de ML a uma predição existente.....	168
Figura 116 – Segunda aplicação, erro apresentado ao administrador quando submete um modelo de ML diferente do algoritmo selecionado	169
Figura 117 – Segunda aplicação, página de consulta de modelos de ML existentes	170
Figura 118 – Segunda aplicação, página de realização de uma nova predição....	171
Figura 119 – Segunda aplicação, apresentação das colunas obrigatórias que o ficheiro com as várias instâncias a predizer deve conter	172
Figura 120 – Segunda aplicação, resultado apresentado depois da realização de uma predição de ficheiro	172
Figura 121 – Segunda aplicação, página de predição quando o utilizador pretende realizar uma predição manual (instância única).....	174
Figura 122 – Segunda aplicação, resultado apresentado ao utilizador depois da realização de uma predição manual	175
Figura 123 – Segunda aplicação, página de consulta de resultados das predições realizadas.....	175
Figura 124 – Segunda aplicação, página de consulta de resultados manuais	176
Figura 125 – Segunda aplicação, janela com os valores dos campos preenchidos (da instância predita) pelo utilizador.....	177

Figura 126 - Exemplo simplificado do <i>workflow</i> do módulo SI.PREVINA no contexto do IPCB até ao momento.....	179
Figura 127 - SI.PREVINA, página principal do docente	180
Figura 128 - SI.PREVINA, <i>dialog</i> de escolha da análise a ser visualizada	180
Figura 129 - SI.PREVINA, menu de navegação do docente	181
Figura 130 - SI.PREVINA, lista apresentada quando o utilizador seleciona a opção "Unidades Curriculares"	182
Figura 131 - SI.PREVINA, <i>dialog</i> com os alunos da UC selecionada	183
Figura 132 - SI.PREVINA, consulta das avaliações de um só aluno quando selecionada a opção "Avaliações" do <i>dialog</i> com os alunos de uma UC	183
Figura 133 - SI.PREVINA, consulta das diferentes épocas de avaliação de todos alunos de uma determinada UC.....	184
Figura 134 - PREVINA, consulta da assiduidade e atitude média das últimas quatro semanas de cada aluno de uma determinada UC.....	185
Figura 135 - SI.PREVINA, lista apresentada quando o utilizador seleciona a opção "Alunos"	186
Figura 136 – SI.PREVINA, <i>dialog</i> de lista de UCs de um determinado aluno.....	186
Figura 137 - SI.PREVINA, ecrã de acompanhamento do utilizador Funcionário	187

Índice de fórmulas

Fórmula 1 - Cálculo da exatidão	20
Fórmula 2 – Outra formulação possível para o cálculo da exatidão.....	20
Fórmula 3 - Cálculo da precisão	20
Fórmula 4 - Cálculo da medida <i>recall</i>.....	20
Fórmula 5 - Cálculo da medida <i>F1-score</i>.....	21

Índice de tabelas

Tabela 1 – Conjunto de dados a serem extraídos de cada artigo analisado	26
Tabela 2 - Sigla dos algoritmos de ML mais utilizados pelos estudos analisados... <td>27</td>	27
Tabela 3 - Ano de publicação, número de citações, momentos de predição e informação do <i>dataset(s)</i> utilizados.....	34
Tabela 4 - Extração dos algoritmos utilizados, incluindo os de treino (ML), e algoritmo que treina o melhor modelo de cada artigo analisado	35
Tabela 5 - Atributos que compõem os <i>datasets</i> originais sem qualquer alteração.....	60
Tabela 6 - Algoritmos de ML utilizados para o treino de modelos e sua sigla	95
Tabela 7 - Resultados do treino <i>holdout</i> para a classe "risco_original".....	99
Tabela 8 - Resultados do treino <i>holdout</i> com balanceamento SMOTE para a classe "risco_orginal"	100
Tabela 9 - Resultados individuais de <i>holdout</i> da classe "risco_orginal" para os níveis "Nenhum" e "Baixo"	100
Tabela 10 - Resultados individuais de <i>holdout</i> da classe "risco_orginal" para os níveis "Médio", "Alto", "Não comparece".....	101
Tabela 11 - Resultados do treino holdout para a classe "risco_binario"	101
Tabela 12 - Resultados do treino validação cruzada para a classe "risco_binario"	102
Tabela 13 - Resultados do treino validação cruzada para a classe "risco_binario" com balanceamento <i>SMOTE</i> no conjunto de treino.....	103
Tabela 14 - Resultados do treino holdout para a classe "risco_optimizado"	104
Tabela 15 - Resultados do treino <i>holdout</i> com balanceamento SMOTE para a classe "risco_optimizado"	105
Tabela 16 - Resultados individuais de <i>holdout</i> da classe "risco_optimizado" para os níveis "Nenhum" e "Baixo"	105
Tabela 17 - Resultados individuais de holdout da classe "risco_optimizado" para os níveis "Médio" e "Alto"	106
Tabela 18 - Resultados do treino <i>holdout</i> da classe "risco_optimizado", cujo conjunto de treino é composto pelas instâncias dos <i>datasets</i> dos anos letivos 2019/2020, 2020/2021, 2021/2022 e as instâncias de teste são pertencentes ao ano letivo 2022/2023	106
Tabela 19 - Resultados do treino validação cruzada para a classe "continua_estudos"	108
Tabela 20 - Resultados do treino validação cruzada para a classe "continua_estudos", cujo conjunto de treino é balanceado com o algoritmo <i>SMOTE</i>	109
Tabela 21 - Resultados do treino holdout da classe "continua_estudos", cujo conjunto de treino é composto pelas instâncias dos <i>datasets</i> dos anos letivos 2019/2020, 2020/2021, 2021/2022 e as instâncias de teste são pertencentes ao ano letivo 2022/2023	110
Tabela 22 – Resultados de regressão do treino <i>holdout</i> para a classe "ects_realizados"	111

Tabela 23 - Resultados de regressão do treino validação cruzada para a classe "ects_realizados"	111
Tabela 24 - Resultados do treino <i>holdout</i> para a classe "intervalo_ects_realizados"	113
Tabela 25 - Resultados do treino validação cruzada para a classe "intervalo_ects_realizados"	114
Tabela 26 - Resultados de <i>holdout</i> por valor possível da classe "intervalo_ects_realizados"	115
Tabela 27 - Resultados do treino <i>holdout</i> da classe "intervalo_ects_realizados", cujo conjunto de treino é composto pelas instâncias dos datasets dos anos letivos 2019/2020, 2020/2021, 2021/2022 e as instâncias de teste são pertencentes ao ano letivo 2022/2023.....	115
Tabela 28 - Resultados obtidos nos treinos do algoritmo de sinalização de nível de risco composto por dois modelos de ML	118
Tabela 29 - Resultados obtidos nos treinos do algoritmo de sinalização de nível de risco com a otimização de hiperparâmetros dos modelos de ML que o compõem....	120
Tabela 30 - Funcionalidade das classes pertencentes ao contexto de treino.....	140

Lista de abreviaturas, siglas e acrónimos

API (*Application Programming Interface*)

CSV (Comma-Separated Values)

DL (*Deep Learning*)

ECTS (*European Credit Transfer and Accumulation System*, em português Sistema Europeu de Transferência e Acumulação de Créditos)

FN (Falsos Negativos)

FP (Falsos Positivos)

IA (Inteligência Artificial)

IDE (*Integrated Development Environment*)

IPCB (Instituto Politécnico de Castelo Branco)

JSON (JavaScript Object Notation)

ML (*Machine Learning*)

REST (*Representational State Transfer*)

SMOTE (*Synthetic Minority Over-sampling Technique*)

UC (Unidade Curricular)

UCs (Unidades Curriculares)

URL (*Uniform Resource Locator ou web address*)

VN (Verdadeiros Negativos)

VP (Verdadeiros Positivos)

1. Introdução

Várias notícias publicadas este ano, 2024, referem que o abandono no ensino superior aumentou pela primeira vez em oito anos [1], [2], [3]. Segundo estatísticas nacionais do ano letivo 2022/2023, publicadas em InfoCurso [4], em média, 11,2% dos alunos inscritos em licenciaturas abandonam o ensino após o primeiro ano letivo. No relato destas estatísticas, o Diário de Notícias em [1] revelou que o Instituto Politécnico de Castelo Branco (IPCB) é uma das instituições de ensino superior (IES) que apresentou uma taxa de abandono superior à média, tendo sido registado uma taxa de 19,8%.

De acordo com informações prestadas pela Presidente do Conselho Pedagógico da Escola Superior de Tecnologia do IPCB e orientadora deste projeto, o IPCB tem, a nível do seu sistema de gestão da qualidade, mecanismos que tentam assegurar a qualidade das suas formações e tomar medidas no final de cada semestre no sentido de melhorar o desempenho das Unidades Curriculares que apresentem um desempenho menos bom. Adicionalmente, além de realizar sessões de acolhimento [5], possui uma comissão que ajuda os estudantes internacionais a integrarem-se mais facilmente na academia e na cidade. Apesar disso, como já foi referido em cima, os resultados destas medidas ficam aquém do esperado.

Uma vez que é um problema que afeta uma grande maioria das IES nacionais [6], a Direção-Geral do Ensino Superior (DGES), ao abrigo do Plano de Recuperação e Resiliência (PRR), lançou um conjunto de submedidas que têm como objetivo melhorar as condições de ensino nas IES. Uma das submedidas propostas, foi lançada ao abrigo do Investimento RE-C06-i07| Impulso Mais Digital: Inovação e Modernização Pedagógica no Ensino Superior - Programa de Promoção de Sucesso e Redução de Abandono Escolar no Ensino Superior [7].

O objetivo principal desta submedida é viabilizar as IES estimularem o desenvolvimento e aplicação de iniciativas e mecanismos de apoio à integração académica, nomeadamente voltada para os novos estudantes (primeiro ano) e promover o seu sucesso académico [7]. Assim, esta, procura financiar mecanismos de mentoria e acompanhamento, implementação de soluções inovadoras de ensino, diversificação das metodologias pedagógicas e mecanismos de predição de situações de abandono [7].

Desta forma, neste trabalho será explorada a vertente de desenvolvimento de um de sistemas de predição com objetivo de combater o abandono escolar. Ao longo deste capítulo será apresentado o enquadramento deste trabalho, os objetivos que se pretendem alcançar com o desenvolvimento do mesmo, as tarefas delineadas e a estrutura do documento.

1.1. Enquadramento

No final do ano de 2023, no âmbito do combate ao abandono escolar, o IPCB submeteu a candidatura do projeto "REVUP - Recursos e Ambientes Colaborativos de Aprendizagem", ao programa Inovação e Modernização Pedagógica no Ensino Superior - Programa de Promoção de Sucesso e Redução de Abandono Escolar no Ensino Superior [7], [8], ao abrigo do Investimento RE-C06-i07| Impulso Mais Digital. Esta candidatura foi aceite e o orçamento proposto foi aprovado quase na sua totalidade.

O projeto REVUP tem como objetivo desenvolver uma plataforma de acompanhamento académico para os alunos do IPCB [8]. Nesta plataforma, os estudantes poderão ser acompanhados por docentes, com a possibilidade de lhes serem associados tutores (docentes) e, adicionalmente, poderem ser acompanhados e motivados por outros alunos através de um sistema de mentoria. O IPCB pretende, também, dar uma ajuda financeira aos alunos que adiram ao projeto, tanto aos alunos 1ºano/1ª vez como aos que se voluntariem para dinamizar sessões de mentorias [8].

Um dos principais componentes do projeto REVUP é uma primeira identificação dos alunos que possam necessitar de um acompanhamento mais dedicado. Para tal, o projeto propõe a utilização de técnicas de *Machine Learning* (ML), nomeadamente dos seus modelos preditivos, para sinalizar os alunos que se matriculem pela primeira vez no IPCB. Dado que a professora orientadora Ana Paula Silva é a coordenadora institucional do projeto REVUP, esta abordou o autor do presente trabalho questionando-o se estaria interessado em desenvolver uma primeira versão do sistema de sinalização de alunos com base em modelos de ML. O autor manifestou o seu interesse e aceitou a proposta. Esta proposta surgiu na sequência das conclusões que tinham sido aferidas sobre a continuação do trabalho realizado na Unidade Curricular (UC) de Projeto I.

De facto, dado que a continuação do trabalho desenvolvido no âmbito da UC de Projeto I, predição de culturas a plantar para determinadas características do solo, revelou-se difícil devido à falta de dados portugueses para prosseguir com o estudo, optou-se por apresentar, em Projeto II, a investigação realizada pelo autor no contexto do desenvolvimento da componente inteligente do projeto REVUP. Assim, este trabalho visa investigar o uso de técnicas de ML para a predição do abandono e/ou insucesso escolar de alunos no ensino superior.

Neste trabalho, serão treinados e avaliados modelos de ML com dados históricos de alunos que se matricularam pela primeira vez no IPCB, e com base nesses dados pretende-se predizer um nível de risco do aluno com base no possível abandono escolar. Além disso, a solução proposta neste trabalho inclui o desenvolvimento de uma aplicação *web* onde os utilizadores (docentes do IPCB) poderão interagir com os modelos de ML para realizar a predição do nível de risco dos seus alunos

1.2. Objetivos

No âmbito da UC de Projeto II, pretende-se desenvolver uma ferramenta inteligente baseada em ML capaz de predizer potenciais casos de abandono e/ou insucesso escolar. A ferramenta proposta visa atribuir um nível de risco aos alunos do IPCB que se matriculam pela primeira vez, sendo esse nível indicativo da probabilidade de desistirem ou enfrentarem insucesso académico, onde níveis mais elevados correspondem a maior probabilidade e gravidade.

Adicionalmente, interessa referir que no convite da submedida apresentado pela DGES [7], fala-se em predição de situações de abandono escolar. Porém, o projeto REVUP define níveis de risco que contemplam além do abandono, o sucesso escolar. A **Figura 1**, mostra o diagrama que foi apresentado à DGES aquando da apresentação das candidaturas.

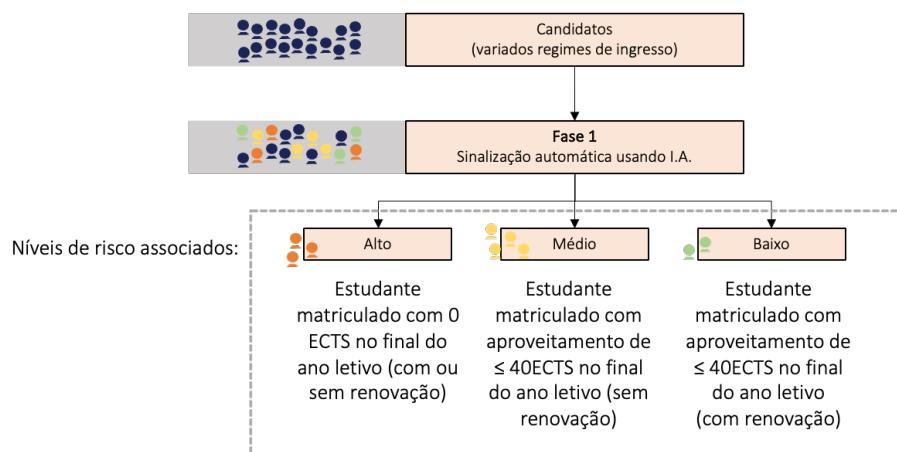


Figura 1 - Projeto REVUP, proposta de sinalização dos alunos por diferentes níveis de risco

Em ML, problemas deste tipo são conhecidos como problemas de classificação, uma vez que os modelos de ML predizem valores nominais [9]. Para o desenvolvimento deste projeto, foram delineados vários objetivos, nomeadamente:

- Realização de um estudo do estado da arte do uso de técnicas de ML na predição de abandono ou insucesso escolar de alunos do ensino superior;
- Verificar quais os dados disponíveis dos alunos do IPCB (*datasets*) que poderão ser utilizados para treinar os modelos de ML;
- Identificar e escolher as bibliotecas em *Python* adequadas para a aplicação de técnicas de ML neste contexto;
- Desenvolver competências técnicas através da utilização das bibliotecas escolhidas para o desenvolvimento deste trabalho;
- Investigar qual o algoritmo de ML mais promissor para identificar alunos com potencial risco de abandono ou insucesso escolar;
- Implementar uma aplicação que permita aos docentes do IPCB interagir com o(s) modelo(s) de ML treinado(s).

Por último, importa referir que todo o trabalho foi desenvolvido após a Comissão responsável da Proteção de Dados do IPCB ter sido informada e ter dado autorização para acesso aos dados necessários para o treino.

1.3. Planeamento do projeto

Para o desenvolvimento do trabalho aqui apresentado, foram definidas seis tarefas a serem realizadas ao longo do 2º semestre do ano letivo 2023/2024. No entanto, não foi estabelecido um cronograma para a execução das mesmas, uma vez que se previu que a sua execução não seria sequencial. Posto isto, as tarefas delineadas para este projeto foram:

- **Elaboração do relatório:** escrever o relatório de forma a documentar e apresentar resultados de todo trabalho desenvolvido;
- **Estudo do estado da arte:** investigar o uso atual de técnicas de ML para a predição de alunos de risco de abandono ou insucesso escolar;
- **Escolher tecnologias e ferramentas a usar:** escolher quais as bibliotecas de Python, tecnologias e ferramentas a utilizar para o desenvolvimento do trabalho;
- **Preparar datasets:** efetuar a devida preparação dos dados a serem dados aos algoritmos de ML;
- **Realizar um primeiro trabalho experimental:** para o(s) dataset(s) disponíveis, treinar e avaliar modelos de ML com objetivo de verificar se é possível utilizar ML para predizer o abandono e insucesso escolar de alunos do IPCB que se matriculam pela primeira vez no ano letivo 2024/2025;
- **Implementação de um protótipo:** desenvolver e alojar uma aplicação web funcional, onde os utilizadores poderão interagir com o modelo de ML previamente treinado.

1.4. Estrutura do relatório

O presente relatório é constituído por nove capítulos. Neste primeiro capítulo é descrito o problema e a solução procurada. Para além da contextualização, são apresentados os objetivos delineados e as tarefas que foram planeadas, para o seu desenvolvimento.

No segundo capítulo, explora-se a área de IA, com ênfase na sua subárea de ML, uma vez que esta é o foco central deste trabalho. Adicionalmente, são explorados os diferentes tipos de aprendizagem computacional existentes, bem como a relevância dos dados para o treino de modelos de ML e como se processa a sua avaliação. Este capítulo, constava do relatório apresentado na UC (Unidade curricular) de Projeto I.

No terceiro capítulo, apresenta-se o estudo do estado da arte que foi realizado e onde se relatada um conjunto de trabalhos publicados que fazem uso de ML para

predição de níveis de risco relacionados com o insucesso e o abandono escolar. Neste capítulo, são delineados os objetivos da pesquisa efetuada, identificados os diversos trabalhos que foram encontrados e posteriormente uma análise dos mesmos. Este capítulo encerra com uma discussão e as principais conclusões retiradas da análise.

No quarto capítulo, são apresentadas todas as tecnologias e ferramentas utilizadas neste trabalho. Sendo apresentado propósito da sua aplicação neste trabalho.

No quinto capítulo, são apresentados os *datasets*, e os atributos que os compõem, fornecidos ao autor para a realização deste trabalho. Adicionalmente, é descrito todo o processo de pré-processamento aplicado aos mesmos, e por fim é realizada uma breve visualização dos dados (análise).

No sexto capítulo, é apresentado o processo completo aplicado no treino e avaliação dos modelos de ML. Além disso, são apresentados os resultados obtidos com os diferentes modelos de ML, um algoritmo de predição que utiliza dois modelos de ML para predizer o nível de risco e por fim uma reflexão dos resultados.

O sétimo capítulo, apresenta e descreve duas aplicações web desenvolvidas no âmbito deste trabalho, que integram os modelos de ML treinados. Para cada um é realizada uma breve descrição da sua utilidade, detalhada a sua arquitetura e apresentados todos os ecrãs e funcionalidades que o compõem.

No oitavo capítulo, é apresentada a contribuição externa do autor para o projeto RevUp. Contribuição essa, feita enquanto o mesmo desempenha as suas funções de desenvolvedor de *software* na empresa Digitalis.

Por fim, o nono capítulo apresenta as principais conclusões retiradas do desenvolvimento deste trabalho, nomeadamente os resultados obtidos, os desafios enfrentados e o que se considera pertinente para trabalho futuro. O capítulo encerra com o delineamento de possíveis direções e melhorias a serem exploradas por alunos e/ou docentes que pretendam dar continuidade ao projeto.

2. Inteligência artificial

Em 1950, Alan Turing publicou um artigo, “*Computing Machinery and Intelligence*” [10], onde chocou os leitores com a proposta do “Teste de Turing” ou “Jogo da imitação” (“*Turing Test*” ou “*The Imitation Game*”) e por ter colocado a questão “Será que as máquinas conseguem pensar?” (“*Can machines think?*”) [11]. Turing propôs um jogo constituído por três jogadores, dois humanos e um computador. Um dos humanos age como um interrogador e encontra-se noutra sala. O interrogador deverá ser capaz de distinguir o humano do computador ao trocar mensagens de texto com o mesmo através de um terminal. Deste modo, Turing choca os leitores, pois, pressupõe de forma indireta que caso o interrogador não consiga distinguir o humano e o computador, isso deve indicar que o computador demonstrou sinais de inteligência - passando assim o teste.

Seis anos depois, em 1956, o termo “Inteligência artificial” (“*Artificial Intelligence*”) é adotado numa conferência da universidade de Dartmouth pelo grupo de cientistas e investigadores na área das ciências da computação John McCarthy, Marvin Minsky, Nathaniel Rochester e Claude Shannon - [12]. Segundo John McCarthy esta conferência teve como objetivo “proceder com base na conjectura de que todos os aspetos da aprendizagem ou de qualquer outra característica da inteligência podem, em princípio, ser descritos de forma tão precisa que uma máquina pode ser feita para os simular.” [13], nascendo assim o termo IA (Inteligência artificial, *AI* que abrevia o termo em inglês, *Artificial Intelligence*). Por fim, em 2004, John McCarthy define a IA como sendo a área da ciência e engenharia que se dedica à criação de máquinas inteligentes nomeadamente à criação de programas de computadores inteligentes [14].

Atualmente, segundo a Oracle em [15], IA é o termo utilizado para aplicações que desempenham funções que antes necessitavam de *input* humano, como por exemplo, comunicar com um cliente e jogar xadrez. Sendo assim, a IA pode ser definida como sendo a área que se dedica à criação de sistemas computacionais ou aplicações capazes de desempenhar funções humanas, capazes de observar, avaliar e interagir no ambiente em que se encontram.

No entanto, a IA engloba um conjunto vasto de outras subáreas de estudo como é o caso das áreas de ML (*Machine Learning*), DL (*Deep Learning*), Robótica, Redes Neuronais, Processamento de Linguagem Natural, Visão computacional, entre outras [16]. Tipicamente, é feita uma divisão em apenas três camadas sendo a mais exterior IA, seguindo-se ML e por fim DL, como se ilustra na **Figura 2** (fonte: [17]). No contexto deste trabalho é explorado a subárea de ML que possui como principal foco a aplicação de algoritmos capazes de aprender uma determinada tarefa com base em dados históricos e/ou padrões nos dados, criando para isso um modelo capaz de executar a tarefa em causa. O modelo desenvolvido pode assumir diferentes formas, dependendo do algoritmo que for adotado.

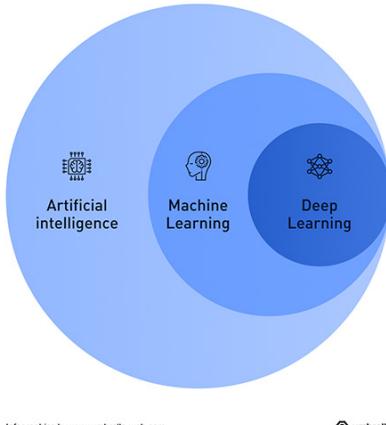


Figura 2 - As três áreas principais de IA

A IA está presente em muitas outras áreas como a indústria automóvel, saúde, cibersegurança, marketing, entre outras. Segundo Satya Ramaswamy, no estudo publicado em Harvard Business Review [18], as empresas, em particular, utilizam a IA em tarefas que permitem detetar e dissuadir invasões de segurança, reduzir a necessidade de controlar e gerir tarefas e/ou procedimentos de produção ao automatizar processos, antever qual o produto que o cliente comprará a seguir e automatizar a distribuição de chamadas em serviços de apoio ao cliente.

As formas em que se traduzem os modelos criados por técnicas de IA podem ser classificados em diferentes tipos. Esta classificação não é unânime, pois, existem opiniões e argumentos diferentes. Contudo, segundo o professor Arend Hintze da universidade de Michigan [19], existem quatro tipos de IA. O primeiro tipo, que ele denomina por “Máquinas reativas” (*“Reactive Machines”*), pode ser considerado o tipo mais básico uma vez que não utiliza memória sobre ações passadas para desempenhar uma determinada tarefa ou aprender algo novo. Hintze dá como exemplo o Deep Blue, o supercomputador criado pela IBM para jogar xadrez que conseguiu vencer ao campeão mundial de xadrez atual, Garry Kasparov, no ano 1997 [20].

O segundo tipo, que ele designou como “Memória Limitada” (*“Limited Memory”*), engloba as técnicas de IA capazes de consultar ações passadas, através de consulta dos dados guardados em memória e agir consoante esse conhecimento. Hintze exemplifica este tipo de IA com os assistentes de condução autónoma presentes nos carros de hoje, que conseguem observar a velocidade e direção de outros carros, sinais luminosos e reconhecer peões e sinais verticais de forma a conseguir tomar decisões de quando devem acelerar, travar, virar, trocar de faixa ou parar [19].

Hintze considera que as formas de IA que existem atualmente podem ser encaixadas num dos dois tipos anteriores. Propõe, contudo, dois outros tipos de IA que ainda não existem e que ele designou por “Teoria da mente” (*“Theory of mind”*) e “Autoconsciência” (*“Self-Awareness”*). Hintzen descreve que sistemas de IA da categoria “Teoria da mente” são capazes de entender que as pessoas, animais e objetos podem ter emoções e pensamentos que afetam o seu comportamento, sendo assim capazes de entender as nossas emoções e sentimentos (*“Theory of mind”*) [19].

Por último, Hintzen afirma que o quarto e último tipo de IA a ser criado será a “Autoconsciência” (“*Self-Awareness*”). Neste caso, as máquinas teriam capacidade de reconhecer a sua própria existência, tal como os humanos ao serem seres conscientes. Hintzen antecipa que este tipo de IA será capaz de compreender o seu estado interno atual e até mesmo prever sentimentos de outros após certas ações ou eventos [19].

2.1. Machine Learning

Aurélien Géron considera que “*Machine learning* é a ciência (e arte) de programar computadores de forma a que estes consigam aprender a partir de dados” [9]. Sendo assim, ML será a subárea de IA que se foca na aprendizagem a partir de exemplos (dados) reais, permitindo assim que o sistema crie as suas próprias regras em vez de tomar decisões com base em regras pré-definidas.

Géron explica em [12] o que é ML ao dar um exemplo real de uma das primeiras aplicações de ML para a criação de um filtro de *spam*. É possível criar um filtro de mensagens *spam* sem uso de ML ao analisar *as mesmas* e identificando palavras, frases ou expressões nelas, tipicamente, utilizadas. Depois disto poderia ser codificado um algoritmo que conseguisse identificar estes padrões e quando a mensagem a ser analisada apresentasse um deles, o algoritmo iria classificar a mensagem como *spam*. Posteriormente, seria avaliada a aplicação do filtro de *spam* com mensagens normais e *spam* já existentes e se o desempenho fosse considerado suficientemente bom, o algoritmo poderia ser adicionado à aplicação em questão, por exemplo, aplicação de emails ou mensagens do telemóvel.

No entanto, esta abordagem tem um problema. É que para além do tempo necessário para analisar e identificar padrões em milhares ou milhões de mensagens, quem as envia pode aperceber-se que esses padrões estão a ser detetados e as suas mensagens são filtradas. Logo os remetentes destas mensagens irão adaptar as mesmas para evitar a classificação como *spam*. Para resolver este problema será necessário voltar a analisar as novas mensagens mal classificadas e adicionar ou alterar padrões.

Sem a utilização de técnicas de ML certos problemas eram e são resolvidos usando o processo atrás descrito. A **Figura 3** adaptada de [9] permite ilustrar estes processos. Numa primeira fase, como se pode observar, é analisado o problema de forma a descobrir os padrões que devem ser ou não evitados, seguindo-se a criação de regras para classificação (no caso do *spam*, as mensagens). Após a criação e codificação de todas as regras o algoritmo é avaliado e caso tenha um bom desempenho é aplicado. Caso o desempenho não seja o desejado é necessário analisar os erros e o problema de forma a criar novas regras.

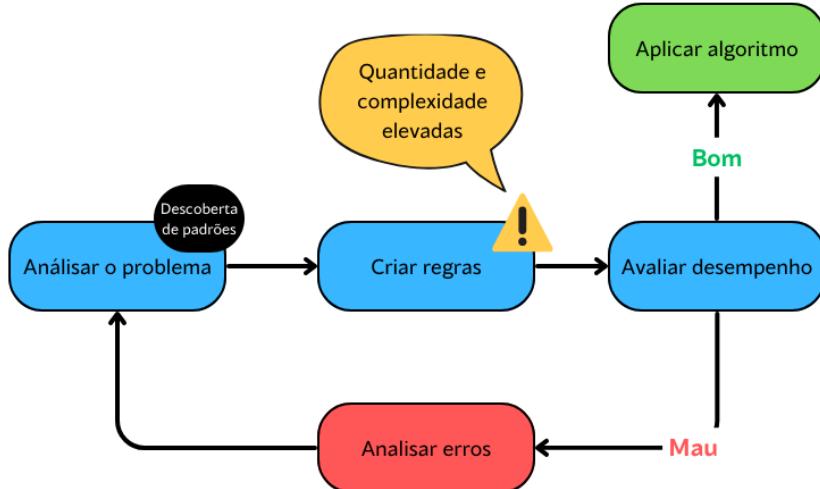


Figura 3 - Fluxograma da abordagem de resolução de um problema sem ML

Ao utilizar este processo para a criação de algoritmos, na verdade está-se a criar um programa com uma longa lista de regras complexas, tornando-se difícil a sua manutenção. A aplicação de técnicas de ML evita o uso deste processo ao permitir que o computador consiga aprender as regras com base em dados já previamente classificados. Desta forma, o computador utiliza algoritmos específicos para criar as suas próprias regras que resultam num modelo capaz de receber como *input* novas instâncias do problema e classificar as mesmas [9].

Continuando com o exemplo do filtro de *spam*, em vez de uma ou várias pessoas andarem a analisar um vasto número de mensagens de email de forma a descobrir padrões, apenas são guardadas mensagens classificadas como sendo, ou não, *spam*. Estas mensagens são posteriormente utilizadas para criar um modelo capaz de fazer a classificação pretendida. Para isso são utilizados algoritmos designados por algoritmos de treino. Estes algoritmos descobrem os seus próprios padrões e criam as suas próprias regras as quais são depois usadas para classificar novas mensagens como sendo *spam*, ou não. A **Figura 4** (adaptada de [9]) permite ilustrar o processo descrito[9].

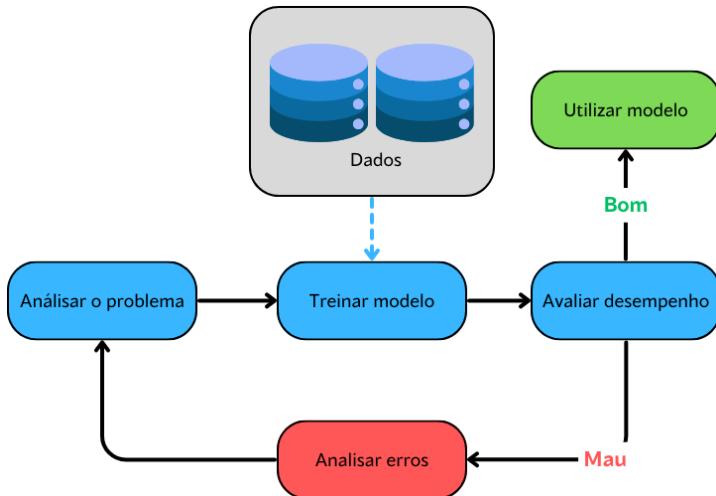


Figura 4 - Fluxograma da abordagem de resolução de um problema com ML

Os modelos resultantes de processos de treino usando técnicas de ML são significativamente melhores que os classificadores criados por humanos [9], [21]. Ainda assim, é possível descobrir falhas nos modelos e consegue-se enganar o modelo na sua classificação [9]. De forma a que este se consiga adaptar a essas falhas, necessita que os seus dados sejam atualizados com novos exemplos que traduzam essas falhas. Assim, o processo ilustrado na **Figura 4**, consegue ainda ser melhorado com a atualização dos dados com novos exemplos (ver **Figura 5**, adaptada de [9]).

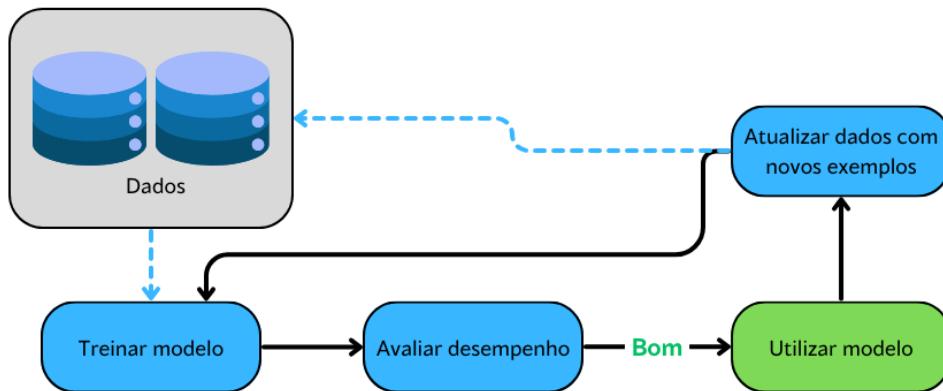


Figura 5 - Adaptação da abordagem de um problema com ML

Ao permitir que o computador aprenda a partir de dados (exemplos reais), as técnicas de ML têm sucesso em problemas cuja codificação é demasiado complexa ou para os quais não se conhece um algoritmo que o resolva. Géron dá como exemplo um algoritmo capaz de distinguir as palavras faladas “one” e “two” [9]. Numa abordagem mais tradicional é tido em conta as frequências de som de letras chave, como “T” na palavra “two”, para se conseguir distinguir. Esta abordagem rapidamente se descarta ao constatar-se que existem milhões de palavras com esta característica e que a mesma palavra pode ter tons diferentes ao ser pronunciada de pessoa para pessoa ou até mesmo poder ser afetada por ruído de fundo. Considerando estes factos, Géron afirma que a melhor solução, atualmente, passa pela utilização de algoritmos que aprendem sozinhos, à custa de vários exemplos de pessoas a falar, ou seja, utilizando ML. [9]

Adicionalmente, as técnicas de ML não só criam modelos que aprendem sozinhos, como, também, permitem que os humanos consigam aprender [9] através da análise desses modelos. Como já mencionado, para o problema da deteção de *email spam*, a utilização de ML permite criar de forma automática regras e padrões os quais depois podem ser analisados por um humano de forma a melhor visualizar combinações de palavras e/ou padrões que são usados nas mensagens *spam*. Géron menciona que por vezes até é possível descobrir novas correlações e tendências que levam a uma melhor compreensão do problema que se tenta resolver [9].

Atualmente, as técnicas de ML são das mais populares dentro da área da IA estando muitas vezes presentes em sistemas com funções que as pessoas assumem como automáticas, como por exemplo a recomendação de produtos. Alguns exemplos de aplicações reais, para além de filtros de *spam*, que utilizam ML são: a plataforma de *streaming* Netflix ao recomendar filmes e séries que o utilizador poderá querer ver a seguir [22]; DeepL que utiliza ML para obter traduções mais fidedignas [23]; Google Lens que utiliza ML para produzir fala a partir de texto capturado pela câmara do *smartphone* [24]; Stripe ao permitir que os clientes usem Stripe Radar para detetar e prevenir fraude nos pagamentos ao analisar os mesmos [25]; Gmail que para além da filtragem de *spam*, consegue recomendar respostas a emails e resumir os mesmos [26]; Facebook que utiliza ML para melhor recomendar anúncios aos seus utilizadores [27]; entre muitas outras.

Na área de ML os modelos são treinados a partir de dados designados por dados de treino, que na verdade são exemplos reais guardados sob a forma digital. No entanto, para que se consiga treinar o modelo deve ser usado um mecanismo de aprendizagem o qual depende do problema que se pretende resolver e dos dados disponíveis. Existem três tipos principais de aprendizagem automática: “Aprendizagem supervisionada”, “Aprendizagem não supervisionada” e “Aprendizagem por reforço” [9] os quais serão explicados nas próximas três secções.

2.1.1 Aprendizagem supervisionada

Aprendizagem supervisionada é talvez o tipo de aprendizagem mais comum no treino de modelos de ML. Este tipo de aprendizagem parte do princípio de que para o problema em questão existem dados corretamente identificados (*label* atribuída), como é o caso do exemplo apresentado do filtro de *spam* para o qual já existem grandes quantidades de mensagens identificadas como *spam* ou não *spam*. No contexto do projeto descrito neste documento, este é o tipo de aprendizagem utilizado.

Este tipo de aprendizagem é utilizado em duas categorias de problemas, problemas de classificação e problemas de predição ou regressão [9], [28]. No caso de classificação são conhecidos os possíveis tipos de resultados (*classes*). O modelo ao observar um novo *input*, que por norma nunca viu antes (não presente nos dados de treino), deve ser capaz de lhe atribuir uma classe [9]. Os problemas de classificação podem ser divididos em duas categorias, classificação binária, quando apenas existem duas

classes possíveis para classificar um novo *input* ou classificação *multiclasses* (múltiplas classes) quando existem mais que duas classes possíveis.

A **Figura 6** (fonte: [9]) ilustra o problema de classificação para o caso do filtro de *spam*, onde dentro dos dados de treino (“*Training set*”) estão presentes alguns exemplos corretamente etiquetados com a sua classe (“*label*”). Estes, por fim, são utilizados para treinar o modelo que posteriormente e de forma indireta são tidos em conta para a classificação um novo email (“*New Instance*”).

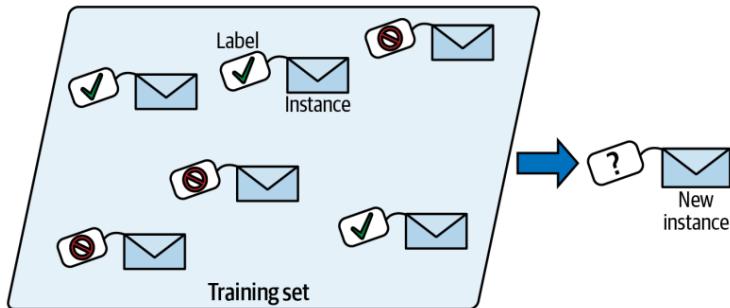


Figura 6 -Exemplificação de dados de treino para o problema SPAM de mensagens em ML

Na aprendizagem supervisionada também é possível predizer um valor numérico, como por exemplo, o preço de um carro [9] ou de uma casa dadas as características dos mesmos. Ao contrário dos problemas de classificação, cuja classe é nominal, os problemas de predição ou regressão têm como classe um valor numérico, não havendo um número pré-definido de possíveis classes, mas sim um intervalo numérico possível ou expectável. Assim, a denominação destes problemas é conhecida por problemas de regressão dado que utilizam técnicas de regressão numérica.

Modelos baseados em técnicas de regressão também são comuns em problemas de classificação, no entanto em vez de ser atribuída uma classe, é indicada a probabilidade de o *input* poder ser de determinada classe (classificação binária). Mais uma vez, para o filtro de *spam* o modelo poderia indicar que a nova mensagem tem uma probabilidade de 20% de ser *spam*. [9]

Para ser possível efetuar a aprendizagem automática é necessário recorrer a algoritmos específicos, sendo estes designados por algoritmos de aprendizagem ou de ML. Face à grande variedade dos mesmos para este tipo de aprendizagem, optou-se apenas pela utilização daqueles mais utilizados do estado da arte e já implementados nas bibliotecas de ML. Adicionalmente, em seguida, é realizado uma descrição breve, dos algoritmos de ML aplicados neste trabalho:

- Algoritmo *Naïve Bayes* - segundo [29], este algoritmo “assume que todos os atributos são independentes e não existe qualquer correlação entre eles”. Como se pode deduzir do nome, este algoritmo utiliza a fórmula do teorema de Bayes [29]. Assim, com o uso de valores de uma instância, calcula a probabilidade dessa instância pertencer a uma determinada classe [29]. Posteriormente, a este cálculo, a classe com maior probabilidade é aquela indicada pelo algoritmo.

- O algoritmo Regressão Logística (do inglês, *Logistic Regression, LoR*) - utiliza regressão para realizar a classificação. Este algoritmo calcula um valor com base numa equação de regressão linear e posteriormente utiliza esse valor na função de ativação, por norma a *sigmoid* [29]. Se o valor de saída da função de ativação for superior a 50%, este classifica com a respetiva classe positiva [29].
- Árvore de Decisão - trata-se de uma árvore idêntica a uma árvore binária, pois, à medida que são percorridos os diferentes nodos, os dados são divididos. Dessa forma, este algoritmo aprende a dividir as classes de forma iterativa ao alterar o atributo e valor a ser utilizado em cada nodo como comparação [29]. Por fim, quando aplicado, assim que se chega a uma folha, esta classifica a instância associada aos valores com a classe colocada nessa folha [29].
- *Random Forest* - é conhecido como um algoritmo *ensemble* dado a ser composto por várias árvores de decisão [29]. Este algoritmo treina várias árvores onde apenas são dados alguns dos atributos e instâncias de treino [29]. Por sua vez, estas devem ser capazes de dividir corretamente as instâncias pela sua classe. Por fim, quando dada uma instância para classificar, todas as árvores criadas irão indicar qual a classe que esta pertence, a classe com o maior número de referências pelas árvores que o compõem, é a classe que classifica a instância.
- *Gradient Boosted Decision (XGBoost e LightGBM)* – este tipo de algoritmos constroem um modelo forte composto por vários modelos mais fracos. Têm como base o uso de árvores de decisão [29]. O treino é realizado de forma iterativa (sequencial) e as várias árvores de decisão são ajustadas de forma a corrigirem os erros dos modelos anteriores [29].
- N-Vizinhos mais próximos (do inglês, *K-Nearest Neighbors, KNN*) - este algoritmo classifica uma nova instância com base numa votação [29]. Esta votação é realizada com base nas instâncias mais próximas da instância dada para classificar. A distância é calculada através de métricas escolhidas, podendo ser a distância Euclidiana, Manhattan, Minkowski entre outras [30].
- Máquina de Vetores de Suporte (do inglês, *Support Vector Machine, SVM*) - cria uma divisão (*decision boundary*) entre as classes projetadas num plano de uma determinada dimensão [29]. O valor da dimensão é igual ao número de atributos [29]. Adicionalmente, esta divisão é colocada o mais longe possível das instâncias que possibilitam essa divisão, pois, uma posição próxima dessas será sensível a más classificações e não generaliza bem [29].
- Perceptron de multicamadas (do inglês, *Multi-Layer Perceptron, NN_MLP*) é um tipo de rede neuronal composta por pelo menos três camadas, sendo a camada inicial designada como entrada (*input*) e a última como saída (*output*) [31]. Cada neurónio existente está diretamente ligado a todos os da próxima camada. Todas essas ligações têm um peso associado que

inicialmente é aleatório e que, ao longo do treino, é ajustado [31]. Esse peso é utilizado numa soma ponderada e inserido numa função de ativação, que posteriormente transfere esse valor para a camada seguinte [31]. Desta forma, ao longo do treino, os pesos de cada ligação são ajustados, de modo a que a última camada classifique corretamente a instância de entrada.

2.1.2. Aprendizagem não supervisionada

Na aprendizagem não supervisionada, os dados de treino não possuem qualquer identificação (*label*). Desta forma o modelo “aprende sem um professor” [9] pois, não há maneira do modelo saber se a sua predição ou recomendação está correta no treino. Consequentemente, a aprendizagem não supervisionada é utilizada em agrupamentos, associação, visualização e deteção de anomalias, descoberta de semelhanças, padrões e diferenças entre os dados e não em saber se dada predição ou classificação está correta [9], [32]. No caso dos agrupamentos, conhecido por *Clustering*, quando os dados são representados sob a forma dum gráfico e são considerados os valores dos seus atributos, pode-se visualizar que os mesmos começam a formar grupos (ver **Figura 7**).

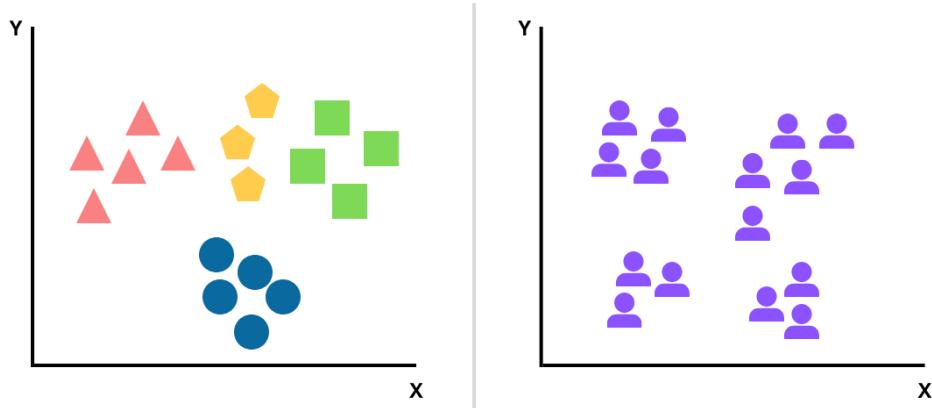


Figura 7 - Exemplo de agrupamentos de dados através do uso da técnica de aprendizagem não supervisionada

A formação destes grupos permite retirar certas conclusões, como, por exemplo, conhecer os utilizadores que visitam um blog [9]. Este conhecimento passa por saber quais são as atividades que os utilizadores partilham entre si, por exemplo os jovens gostarem de banda desenhada. Esta informação pode ser útil ao autor do blog uma vez que ele pode realizar novas publicações com base neste conhecimento [9].

Para a descoberta de associações a aprendizagem não supervisionada é utilizada, por exemplo, para conhecer hábitos dos consumidores. São o caso das recomendações de “pessoas que compraram este produto, também compraram estes” (Amazon) e de playlists que atualizam as músicas todas as semanas de forma a que o utilizador descubra novas músicas que possa gostar (Spotify) [33], [34].

Na deteção de anomalias este tipo de aprendizagem é utilizado para comparar novas instâncias com as existentes, permitindo detetar *outliers*. A deteção de anomalias

pode ser utilizada na prevenção de fraudes e descoberta de defeitos de fabrico ao comparar novas transferências ou produtos com os já existentes.

Quanto a algoritmos para este tipo de aprendizagem podem ser encontrados os algoritmos k-means e *DBSCAN* para *clustering* [9] e Gaussian Mixtures, Fast-MCD, *Isolation forest* e *Local outlier factor* para deteção de anomalias [9]. Realça-se que estes algoritmos não predizem um valor ou classificação uma instância, mas sim permitem uma visualização das instâncias em diversos *clusters* [9] que posteriormente podem ser identificados as anomalias e/ou conhecer os grupos existentes.

2.1.3. Aprendizagem por reforço

Por último, a aprendizagem por reforço segue uma dinâmica diferente das duas aprendizagens anteriores. Em [33], [35] este tipo de aprendizagem é considerada como a mais parecida com a aprendizagem humana. Neste tipo de aprendizagem não há propriamente o treino de um modelo, mas sim de um sistema, tipicamente designado por “agente” [36].

Na aprendizagem por reforço o agente é recompensado ou penalizado de acordo com as suas ações [9], [33], [36]. O agente observa o ambiente que se encontra, toma uma decisão e consoante a sua decisão pode ser recompensado ou penalizado [9]. Dado que é uma aprendizagem iterativa, antes de realizar uma nova ação e quando penalizado, o agente deve atualizar o seu conhecimento para não voltar a cometer o mesmo erro.

Segundo [36], [37] a melhor forma de explicar o funcionamento deste tipo de aprendizagem é através de uma analogia com os jogos digitais. Geralmente a personagem principal (agente) num dado ambiente possui um conjunto de ações possíveis e deve agir no mesmo até conseguir passar o nível ou cumprir os objetivos do mesmo. Quando o consegue fazer é recompensado e quando perde é penalizado. Assim, o agente irá tentar muitas vezes e começa a “entender” como deve reagir a certos eventos, levando o mesmo à vitória ou ao cumprimento dos objetivos.

Na aprendizagem por reforço o ambiente é representado à custa de um processo de decisão conhecido por *Markov Decision Process* (MDP) onde são definidos um número finito de estados, ações possíveis em cada estado, uma função de recompensa e um modelo de transição de uso de ações para alterar o estado [36], [37], [38]. Conjuntamente com o processo de decisão, geralmente, é utilizado o algoritmo *Q-Learning*, que procura encontrar o melhor conjunto de ações a serem executadas pelo agente com base nos seus estados [39], ou o algoritmo SARSA (*State-action-reward-state-action*), muito parecido ao *Q-Learning*, mas que tem em consideração a ação escolhida, para realizar a aprendizagem [36]. Apesar destes serem os mais utilizados, podem ser usados outros métodos ou algoritmos não mencionados.

2.2. Deep Learning

Como já foi referido (ver **Figura 2**), a área de DL é uma subárea de ML. Foca-se sobretudo no uso de técnicas com “circuitos algébricos com forças de ligação ajustáveis” [11]. Pode-se dizer que as origens de DL remontam de 1943 quando McCulloch e Pitts tentaram modelar o funcionamento do cérebro humano, criando uma rede de unidades simples cujo funcionamento era semelhante ao de um neurónio biológico [11]. Desta forma, quando se mencionam redes treinadas por técnicas e métodos de DL estas são denominadas por “redes neuronais” [11].

A palavra “*Deep*” refere-se ao facto de o processamento ser realizado em várias camadas [11], [40], [41]. Em [42] define-se DL como sendo a área das redes neuronais que possuem três ou mais camadas e tentam simular o comportamento do cérebro humano. Adicionalmente François Chollet em [41], menciona que nos dias de hoje, DL envolve dezenas ou até mesmo centenas de camadas sucessivas que aprendem automaticamente a partir dos dados de treino. Atualmente, é utilizado em áreas de maior complexidade, como, processamento de linguagem natural, visão computacional, geração de imagens, videojogos, robótica, entre outros [43].

2.3. Aquisição de dados

A aplicação de uma grande maioria dos algoritmos da área de IA (e suas subáreas) implica a utilização de um conjunto vasto de dados. No entanto, estes dados precisam de ser fidedignos e de qualidade, pois, se não forem, pode não ser possível obter o resultado pretendido. Na área de ML usa-se, frequentemente, a expressão “*garbage in, garbage out*” [9], ou seja, quando se fornece má informação, o *output* gerado será, também, mau. Assim, é importante que os dados fornecidos para o treino dos modelos sejam de fontes confiáveis e sejam de alta qualidade.

Cada vez que se menciona as áreas IA, ML ou DL, uma das palavras que por norma as acompanha é “dados”. Segundo o dicionário de Oxford [44] “dados” são “*factos ou informação, especialmente quando examinados e usados para descobrir coisas ou tomar decisões*”. Posto isto, dados são informação, a qual quando interpretada por nós, humanos, é utilizada para tomar decisões. Como já foi referido na secção **2.1. Machine Learning**, é possível utilizar dados para treinar modelos, que posteriormente são capazes de dar resposta a problemas ou até mesmo predizer o futuro.

Por norma, quando investigadores e empresas procuram obter dados para treino e criação de modelos de ML recorrem, geralmente, a quatro soluções. A primeira solução é aquisição de dados pertencentes a terceiros (outras empresas) ou de *brokers* (intermediários), pagando para adquirirem essas bases de dados ou parte delas [45]. O segundo método é a recolha direta através da respetiva aplicação (quando aplicável) que os clientes utilizam diariamente. Esta recolha pode ser de forma direta com inquéritos e questionários ou indireta com processos em *background* que recolhem e enviam dados periodicamente [45].

Outra solução popular é possível graças a instituições académicas, investigadores individuais, plataformas de dados abertos e defensores de código aberto que criam e publicam os seus *datasets* (base de dados ou ficheiros com dados) a partir de recolha previamente realizada, como é o exemplo a plataforma *Kaggle* [46] e diversos repositórios de universidades espalhadas por todo o mundo [45]. Por último, a aquisição de dados pode ser feita manualmente ao utilizar sensores de Internet das Coisas (do inglês, *Internet of Things, IoT*), *web scraping*, API (*Application Programming Interface*), entre outras maneiras onde o indivíduo ou empresa cria o seu próprio *dataset* [47], [48].

Ainda assim, existe uma quantidade considerável de barreiras na aquisição de dados para certos problemas. Algumas dessas barreiras são: questões de privacidade; segurança dos dados; custos associados na aquisição; diversidade dos dados; dados não imparciais; quantidade e qualidade [49], [50], [51], [52]. As últimas duas, quantidade e qualidade dos dados, possuem um impacto considerável no treino e surgiram no desenvolvimento deste projeto.

A primeira é a qualidade dos dados. Dados de qualidade, são dados que são completos, precisos e representativos do problema em questão [49]. Os problemas surgem quando existem dados com erro ou *outliers* (valores anómalos), ou seja, dados que variam bastante dos restantes [53]. Desta forma, é essencial verificar que os dados são de uma fonte confiável, bem identificados (no caso da aprendizagem supervisionada) e garantir que os mesmos estão limpos, removendo dados com erros, ou alterando os mesmos de forma a generalizar, e a retirar os *outliers*.

Por último, a segunda barreira é a existência de poucos dados e *datasets* para certos problemas. Isto porque, dependendo do problema, podem ser necessárias enormes quantidades de dados para treinar corretamente os algoritmos [9], [54]. Em outras situações pode mesmo não existir uma grande quantidade e/ou variedade de dados, nomeadamente em casos em que ainda não houve interesse suficiente para esse problema. Nestes casos o método de recolha de dados, geralmente, tem de ser feito de forma manual por parte do investigador ou empresa. No contexto deste projeto, esta foi uma das maiores barreiras, e será abordada no capítulo 5. **Datasets**.

2.4. Avaliação e métricas para problemas de classificação

Para verificar se um modelo de ML está apto para ser utilizado numa aplicação concreta, depois do seu treino deve-se realizar a sua avaliação. Esta avaliação deve ser feita utilizando dados do problema que o modelo não viu e não conhece. O caso mais comum, e talvez a melhor opção, passa por dividir os dados [9] a serem utilizados de forma a separar os mesmos para treino e para avaliação [12]. Tipicamente, esta divisão é feita em 80% para dados de treino e 20% para dados de teste. No entanto, esta divisão depende da quantidade de dados disponíveis, pois a utilização de apenas 1% de dez milhões de dados (100 000), para testar, pode ser aceitável [9]. Por norma, a seleção dos dados de teste é feita de forma aleatória [9], [55].

Dois dos objetivos da avaliação é evitar o *overfitting* e o *underfitting*. O primeiro acontece quando o modelo aprendeu muito bem os dados de treino, mas quando são fornecidos dados nunca antes vistos, possui um mau desempenho dado que não generalizou a sua aprendizagem [9], [11]. Diz-se que aconteceu *underfitting* quando o modelo não consegue aprender os padrões presentes nos dados [9], mostrando um mau desempenho durante o treino. O ideal será, portanto, treinar um modelo capaz de generalizar a sua aprendizagem de forma a conseguir obter um bom desempenho em dados nunca antes vistos [9], [56].

Existem diversas métricas de avaliação para os diferentes tipos de problemas. Pois os problemas de classificação utilizam métodos de avaliação diferentes dos problemas de regressão ou predição. Problemas de agrupamento (*clustering*), na aprendizagem não supervisionada, também utilizam outras métricas. Dado que existem várias métricas e que estas estão associadas ao tipo de problema, apenas serão apresentadas aqui as métricas de avaliação para problemas de classificação, uma vez que é o problema que se pretende resolver no âmbito deste projeto. Assim, as métricas utilizadas para problemas de classificação são: cálculo da exatidão; cálculo da precisão; cálculo de *recall* e *F1-score*.

Antes de serem apresentadas e explicadas as métricas mencionadas, interessa abordar a matriz de confusão. A matriz de confusão é bastante popular para visualizar o desempenho do modelo na atribuição das classes durante a fase de teste [56]. Para que seja possível compreender a matriz é necessário saber que num problema de classificação existem quatro tipos possíveis de resultados que são (com base de [57]):

- Verdadeiros Positivos (VP): quando uma instância é classificada como pertencendo a uma classe e ela realmente pertence a essa classe;
- Verdadeiros Negativos (VN): quando uma instância é classificada como não pertencendo a uma classe e realmente não pertence a essa classe;
- Falsos Positivos (FP): quando uma instância é classificada como pertencendo a uma classe e ela não pertence a essa classe;
- Falsos Negativos (FN): quando uma instância é classificada como não pertencendo a uma classe, mas na realidade pertencia a essa classe.

Com base nestes tipos de resultados é possível construir a matriz de confusão. Esta matriz possui um tamanho de N por N, onde N é o total de classes possíveis que o modelo pode classificar. Por norma, na matriz de confusão o eixo X (colunas) representa a classe que o modelo prediz e o eixo Y (linhas) é a classe real ou correta. Para melhor visualizar como uma matriz de confusão é constituída pode-se observar a **Figura 8** (fonte: [58]).

		Estimate		
		$c_0 \dots c_{k-1}$	c_k	$c_{k+1} \dots c_n$
annotated ground truth	$c_{k+1} \dots c_n$	TN	FP	TN
	c_k	FN	TP	FN
	$c_0 \dots c_{k-1}$	TN	FP	TN

TN	true negative
TP	true positive
FN	false negative
FP	false positive

Figura 8 - Matriz de confusão e posições dos VP, VN, FP e FN

Aplicando a matriz de confusão a um problema real obtém-se uma matriz semelhante à da **Figura 9** (retirada de [59]). O ideal seria obter apenas valores superiores a 0 nas células da diagonal principal, pois, são os casos em que o modelo prediz corretamente a classe.

		PREDICTED			
		APPLE	GRAPES	BANANA	ORANGE
ACTUAL	APPLE	10	2	1	2
	GRAPES	5	12	1	2
	BANANA	5	2	18	10
	ORANGE	11	3	1	15

4 x 4 Confusion matrix for fruit classifier

Figura 9 - Exemplo real de matriz de confusão para classificação de frutas

O cálculo da exatidão indica a percentagem de respostas corretas (quando foi atribuída a classe correta) tendo em consideração todas as respostas a que o modelo respondeu durante a fase de teste (ver **Fórmula 1**). Adicionalmente também se pode representar o cálculo de exatidão usando a **Fórmula 2**, caso seja pretendido utilizar os vários tipos de resultados possíveis, VP, VN, FP e FN [56], [57], [60].

$$\text{Exatidão} = \frac{\text{Respostas certas}}{\text{Total de respostas}}$$

Fórmula 1 - Cálculo da exatidão

$$\text{Exatidão} = \frac{VP + VN}{VP + VN + FP + FN}$$

Fórmula 2 - Outra formulação possível para o cálculo da exatidão

Apenas o cálculo da exatidão não permite obter uma boa percepção do desempenho do modelo. Em dados que tenham uma ou várias classes, só o cálculo da exatidão não indica se o modelo é bom ou não, pois, não permite saber o seu desempenho em classes minoritárias [56], [57]. Nestes casos é comum recorrer-se a duas outras medidas: precisão e *recall*.

A precisão permite conhecer a percentagem de respostas corretas tendo em consideração o total de respostas certas para uma determinada classe [57]. O cálculo da precisão é apresentado na **Fórmula 3** e deve ser aplicado para todas as classes do problema.

$$\text{Precisão} = \frac{VP}{VP + FP}$$

Fórmula 3 - Cálculo da precisão

A medida *Recall*, usando a designação *em inglês*, permite conhecer a percentagem das respostas efetivamente corretas tendo em consideração todas as respostas em que o modelo classificou uma instância como sendo de uma determinada classe (quer esteja ou não correta a classificação). Este cálculo é representado pela **Fórmula 4** e, tal como na precisão, deve ser aplicado para todas as classes possíveis.

$$\text{Recall} = \frac{VP}{VP + VN}$$

Fórmula 4 - Cálculo da medida *recall*

A **Figura 10** (retirada de [57]) permite entender melhor a diferença entre as medidas precisão e *recall* [56], [57]. Como se pode ver, a precisão, lado inferior esquerdo da **Figura 10**, tem em consideração as respostas certas e erradas para uma determinada classe e a medida *recall*, no lado inferior direito **Figura 10**, considera todas as respostas que atribuiu a classe em questão, mas com o intuito de saber apenas as classificações corretas.

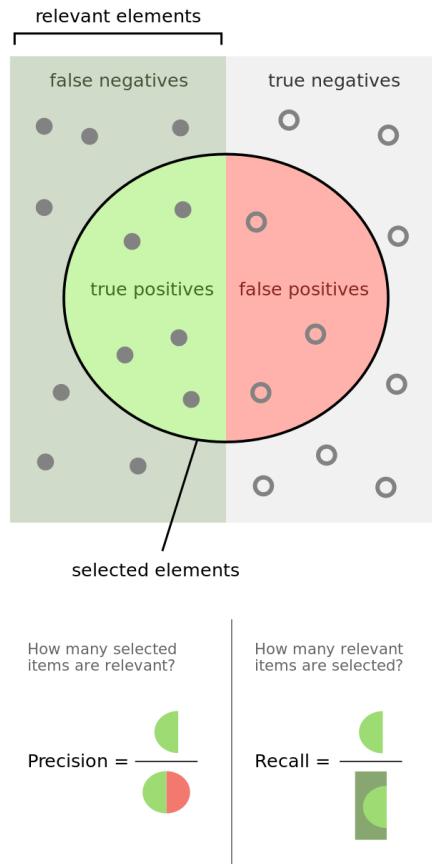


Figura 10 - Exemplo do cálculo da precisão e *recall*

Dependendo do tipo de problema, por norma deve-se ter em consideração a precisão ou o *recall*, no entanto, existem problemas que necessitam de bons resultados em ambas as métricas [56]. Nestes casos, utiliza-se a métrica *F1-score* que combina a precisão e *recall* numa só métrica, utilizando a média harmônica (ver a **Fórmula 5**).

$$F1 = \frac{2 * \text{Precisão} * \text{Recall}}{\text{Precisão} + \text{Recall}}$$

Fórmula 5 - Cálculo da medida *F1-score*

3. Estudo do estado da arte

Nesta secção, é realizado um estudo do estado da arte na utilização de ferramentas de IA (ML e/ou DL) para a deteção e prevenção de alunos em situação de risco que acabam por desistir do ensino superior ou tenham uma maior dificuldade a completar a sua graduação. O estudo permitirá ao autor conhecer e compreender o trabalho e a investigação já realizada, capacitando-o a explorar novas soluções, adaptar soluções existentes e/ou aprimorar a sua solução.

3.1. Metodologia e processo de pesquisa

Para a elaboração do estudo do estado da arte do uso de técnicas de ML para a detenção e prevenção de alunos de risco em risco de desistência, não foi utilizada nenhuma metodologia de revisão sistemática de literatura. Apesar disso, foram estabelecidos elementos no processo de pesquisa que se basearam na metodologia PRISMA [61], sendo eles:

- Propósito, objetivos e questões da investigação;
- Fonte de dados a usar;
- Estratégia adotada na pesquisa;
- Critérios de elegibilidade para análise;
- Extração e análise de dados;
- Conclusão e discussão.

Relativamente ao processo de pesquisa, foi estabelecido um número total de estudos a serem analisados igual a 10 e com preferência em estudos publicados em revistas (*journal*). Para além destes 10 estudos, abre-se espaço para estudos descobertos em pesquisas separadas em motores de pesquisa (exemplo: Google), bem como para artigos citados nos estudos analisados. Esta abordagem baseia-se principalmente no reconhecimento de que o problema específico de identificação de alunos em risco nas escolas e cursos do IPCB possui especificidades próprias, objetivos, regras, dados e estratégias de implementação. Complementarmente esta decisão foi aprovada e sugerida pelos professores orientadores do projeto.

3.1.1. Propósito e objetivos

Uma vez que o problema apresentado neste relatório é referente e destinado aos alunos das escolas e cursos do IPCB, é significativamente provável que a solução aqui proposta diferencie-se consideravelmente da pesquisa existente. Contudo, procurou-se conhecer e compreender a pesquisa existente em outras escolas e universidades. Deste modo, um dos propósitos deste estudo do estado da arte é confirmar e validar se a solução proposta pelo autor e docentes do IPCB vai ao encontro das soluções já existentes e/ou empregues. Complementarmente procura-se melhorar a solução que

será empregue no IPCB com base nos erros e descobertas de outros estudos em outras escolas e/ou universidades.

A elaboração deste estudo está fundamentalmente relacionada com o recente investimento do IPCB em ajudar os seus alunos em situação de risco e reduzir as suas taxas de desistência. O objetivo é incentivar os alunos a completar os seus estudos e a promover o seu sucesso escolar, profissional e até mesmo pessoal. No entanto, uma vez que muitos dos detalhes técnicos e a abordagem já estão definidos no programa REVUP e os recursos, nomeadamente dados, são limitados aos que o IPCB possui, foi realizada uma análise dos estudos mais referenciados, pois, espera-se que estes tenham uma qualidade superior e apresentem descobertas úteis para o trabalho aqui apresentado.

Com o propósito e objetivos definidos, procurou-se responder a um conjunto de questões, sendo elas: (1) quais são os dados utilizados para o treino dos modelos de ML ou DL, (2) em que altura do ano letivo é feita a deteção (exemplo: ato da matrícula, antes dos exames, no fim do primeiro semestre, no fim do primeiro ano, entre outras), (3) qual é o tipo de instituição de ensino (universidade, instituto politécnico, escola secundária ou outra) e o seu país, (4) quais foram os atributos (*features*) mais relevantes para a predição, (5) qual é a classe (valor a predizer) dos modelos de ML ou DL, (6) qual (ou quais) foram os algoritmos utilizados para treino de modelos de ML e qual aquele que depois de avaliado obteve os melhores resultados preditivos, e (7) se a solução foi empregue numa aplicação ou *website* que permite a utilização dos modelos treinados e se a sua utilização foi bem sucedida.

As respostas a estas perguntas permitiram, fundamentalmente, identificar as diversas soluções possíveis para combater a desistência escolar criadas para as instituições de ensino, identificar os dados utilizados e mais relevantes, bem como em que momento do ano letivo são realizadas as predições e com que nível de confiança. Permitindo, assim, obter uma visão abrangente da investigação mais referenciada nesta área, abrangendo não apenas as tecnologias e técnicas empregues, mas também a capacidade e utilidade das aplicações baseadas em técnicas de aprendizagem computacional.

3.1.2. Fontes de dados

Para a pesquisa de artigos foi utilizada a plataforma *Scopus* [62]. Esta é uma plataforma web que indexa milhões de artigos científicos, revistas, livros, entre outros. Adicionalmente, a *Scopus* indexa também estudos publicados em outras bases de dados e sistemas de indexação científica, incluindo os mais relevantes na área das Ciências da Computação, nomeadamente ACM [63], IEEE [64], Springer [65] e muitos outros.

A *Scopus* possui diferentes tipos de pesquisa, sendo uma delas a pesquisa avançada que permite procurar estudos através de uma *string* de pesquisa. Esta funcionalidade possibilita configurar termos de pesquisa em diversos campos, como título, palavras-chave, resumo, texto completo, autor, entre outros. Na construção da *string* de

pesquisa, é possível utilizar operadores lógicos, nomeadamente AND, OR e NOT, permitindo uma pesquisa mais seletiva de estudos.

Apesar de existirem outras plataformas de indexação com funcionalidades mais avançadas, como a B-On [66] que automaticamente remove estudos duplicados, a escolha da *Scopus* foi fundamentada pelo facto de que esta permite ordenar a pesquisa pelo número de referências enquanto a *B-On* não o permite. Além disso, uma vez que a *Scopus* indexa estudos publicados em outras plataformas e entidades, nomeadamente IEEE, ACM e Springer, optou-se por realizar a pesquisa exclusivamente na *Scopus*.

Assim, o uso da *Scopus* possibilita obter acesso a uma vasta variedade de publicações e ordenar os resultados pelo número de citações, permitindo identificar os estudos que foram amplamente referenciados por outros autores. Isto sugere uma boa qualidade e a presença de descobertas inovadoras ou essenciais para o problema em questão.

3.1.3. Estratégia de pesquisa

Para a procura e análise do estado da arte, mais concretamente para a pesquisa de artigos definiu-se um conjunto de palavras-chaves que se considerou melhor identificarem na área das publicações a pesquisar. Nomeadamente, a aplicação de técnicas de aprendizagem computacional no contexto de identificar e ajudar a combater o insucesso e abandono escolar. A procura de estudos foi realizada no dia 28 de maio de 2024 e a *string* utilizada na pesquisa avançada na plataforma *Scopus* foi a seguinte:

```
TITLE ( "early" AND ( "student* dropout" OR "student* performance" OR "academic failure" OR ( "student*" AND "risk" ) ) AND ( "machine learning" OR "ml" OR "deep learning" OR "dl" ) ) OR ABS ( "early" AND ( "student* dropout" OR "student* performance" OR "academic failure" OR ( "student*" AND "risk" ) ) AND ( "machine learning" OR "ml" OR "deep learning" OR "dl" ) ) OR TITLE-ABS-KEY ( "early" AND ( "student* dropout" OR "student* performance" OR "academic failure" OR ( "student*" AND "risk" ) ) AND ( "machine learning" OR "ml" OR "deep learning" OR "dl" ) )
```

A *string* de pesquisa abrange os campos título do documento, resumo e palavras-chave, como é possível, confirmar e no total foram obtidos 802 resultados no dia 28 de maio de 2024. É importante denotar que foram realizadas mais consultas posteriormente ao dia 28 de maio, porém a publicação de novos artigos não influenciou os estudos escolhidos, uma vez que os resultados eram ordenados por número de citações.

3.1.4. Critérios de elegibilidade para análise

Com os artigos ordenados por ordem decrescente de número de citações, torna-se necessário estabelecer critérios de elegibilidade, uma vez que nem todos os resultados

são adequados para análise. Quando analisados os títulos dos documentos resultantes da pesquisa, constatou-se que, apesar de a *string* de pesquisa ser complexa e seletiva, muitos dos estudos pertenciam a outras áreas, nomeadamente à saúde, e não estavam relacionados com a educação de nível secundário e/ou superior.

Assim, foram definidos os seguintes critérios de elegibilidade: (1) documentos escritos em inglês ou português; (2) publicação não é uma revisão sistemática (*review*); (3) autores utilizaram algoritmos de ML e/ou DL para treinar os seus modelos preditivos; (4) foi realizada a avaliação aos modelos produzidos pelos algoritmos e são apresentados os resultados dessa avaliação e (5) seja mencionado que o objetivo do estudo é identificar alunos em risco de desistência e/ou insucesso escolar.

Foram selecionados dez estudos com um número de citações entre 25 e 150. Além disso, foi incluído um estudo português, referenciado em [67], devido à sua autoria ser por professores da Universidade do Porto. Embora tenha sido mencionada a possibilidade de incluir estudos provenientes de fontes externas à *Scopus*, nenhum outro foi considerado, uma vez que não acrescentavam valor ao estudo do estado da arte já realizado.

3.1.6. Extração de dados e análise

A leitura aprofundada de cada um dos 11 artigos permitiu reunir para cada, um conjunto de dados que se considerou serem pertinentes para dar resposta às questões de investigação colocadas no subcapítulo **3.1. Metodologia e processo de pesquisa:** Ano, Número de citações, Dados utilizados (1), Momento da predição (2), País (3), Atributos mais relevantes (4), Classe (5), Algoritmos de codificação dos atributos do *dataset* (6), Algoritmos de balanceamento de *dataset* (7), Algoritmos de ML utilizados para o treino (8), Melhor algoritmo de ML (9), Solução implementada (10). Na **Tabela 1** é efetuado uma breve descrição de toda esta informação que se pretende reunir referentes a cada um destes campos. A extração destes dados permitiu identificar informação útil que contribuiu para melhorar a avaliação da solução a ser apresentada posteriormente e conhecimento de como outros autores abordaram este problema de identificação de alunos em risco.

Tabela 1 - Conjunto de dados a serem extraídos de cada artigo analisado

Dado	Descrição
Ano	Ano de publicação do artigo
Dados utilizados	Breve descrição dos dados utilizados pelos autores
Momento de predição	Em que momento é realizada a predição ou qual é o primeiro momento nos casos de vários momentos
País	País referente aos dados utilizados
Atributos mais relevantes	Atributos que os autores consideram mais relevantes para a recomendação
Classe	Valor a ser predito pelo(s) modelo(s) de ML e/ou DL
Algoritmos de codificação	Algoritmos utilizados para codificar os diversos atributos do <i>dataset</i>
Algoritmos de balanceamento	Algoritmos utilizados para equilibrar o número de instâncias por classe do <i>dataset</i>
Algoritmos de treino	Algoritmos utilizados para a criação e treino dos modelos preditivos
Melhor algoritmo de treino	Algoritmo que produziu o melhor modelo e respetiva pontuação na avaliação (referente ao momento de predição)
Solução implementada	Se a solução proposta no estudo foi implementada e/ou testada

Deste modo, a leitura e análise dos artigos selecionados permitiu sintetizar valores para cada um dos dados correspondentes aos campos da **Tabela 1**. Esta síntese pode ser consultada nas seguintes tabelas, **Tabela 3** e **Tabela 4**. Um resumo com a informação considerada pertinente em cada um dos trabalhos referidos é apresentado na secção seguinte, **3.2. Análise dos artigos**.

3.2. Análise dos artigos

Neste capítulo, apresenta-se um resumo para cada artigo analisado. Cada resumo realça o objetivo da investigação, a predição a ser realizada, os dados utilizados para o treino dos modelos de ML, os resultados de desempenho dos modelos treinados e as principais conclusões. Para facilitar a organização e síntese dos dados relevantes (**Tabela 3**), foram criadas duas tabelas: **Tabela 3** e **Tabela 4**. Ambas as tabelas são apresentadas a seguir à última análise.

Adicionalmente, dado que a referenciação e/ou enumeração dos algoritmos de ML pelo seu nome completo se tornaria repetitivo, levaria a frases e colunas extensas, optou-se pelo uso de siglas. O uso de siglas facilita a leitura, uma vez que depois de consultada não exige a leitura extensa do nome. Estas foram apenas utilizadas para os algoritmos com mais ocorrências. Posto isto, foi criada uma tabela, **Tabela 2**, auxiliar à leitura secção de análise, onde estão presentes nome do algoritmo correspondente a cada sigla.

Tabela 2 - Sigla dos algoritmos de ML mais utilizados pelos estudos analisados

Sigla	Nome do algoritmo
DT	Decision Tree
NB	<i>Naïve Bayes</i>
RF	<i>Random Forest</i>
SVM	Support Vector Machine
NN	Neural Network
LoR	Logistic Regression
KNN	k-Nearest Neighbors
NN (MLP)	MultiLayer Perceptron Neural Network
DNN	Deep Neural Network
GB	Gradient Boosting

Em [68], o autor relata um crescente interesse no uso de técnicas de data mining, que por sua vez são baseadas em ML, no setor do ensino superior. Neste contexto, o autor investiga uma solução capaz de prever a nota do exame final da unidade curricular *Turkish Language - 1*, sem recorrer a informações demográficas ou socioeconómicas dos alunos. Para tal, utilizou um *dataset* de alunos da Universidade Kırşehir Ahi Evran, composto apenas pela nota do exame intercalar da unidade curricular, faculdade e departamento frequentado. Este *dataset* totaliza 1854 instâncias referentes ao ano letivo 2019/2020. O objetivo do autor centra-se na identificação dos alunos que irão reprovar no exame final, de modo que medidas corretivas possam ser implementadas para ajudar a garantir a aprovação dos mesmos. Adicionalmente o autor refere ter discretizado as notas em intervalos para acomodar os algoritmos utilizados, uma vez que são classificadores. Para o treino, o autor recorreu à ferramenta *Orange Machine Learning*, onde definiu um *workflow* que treinará vários modelos de ML com diferentes algoritmos, sendo eles: RF, NN, LoR, SVM, NB e KNN. Complementarmente, é dito ter sido utilizada a estratégia de validação cruzada com um tamanho de 10 para o treino destes modelos. Entre todos os modelos preditivos treinados e testados, aquele treinado pelo algoritmo *Random Forest* apresentou o melhor desempenho, com uma exatidão de 74,6%, F1 de 72,1%, precisão de 75,2%, *recall* de 74,6% e AUC de 86%. O autor conclui o seu estudo indicando que os resultados apresentados demonstram a viabilidade da utilização de técnicas de ML na predição de aprovação ou reprovação numa determinada unidade curricular. Esta abordagem permite que, nos casos de predição de reprovação, sejam tomadas medidas antecipadas para evitar o insucesso escolar, contribuindo para a redução do número de reprovações.

Os autores de [69] procuram, através de técnicas de ML, identificar alunos que podem beneficiar de intervenção que os ajude a concluir os seus estudos com sucesso. O estudo focou-se na procura do momento temporal mais antecipado possível em que podem ser realizadas previsões, obtendo bons resultados, em vez de ser treinado o modelo preditivo mais eficiente. Para isso, os autores utilizaram dados da Universidade de Bangor referentes ao ano letivo de 2016/2017, totalizando 4970 instâncias. Os dados incluem o rácio de presença de cada aluno (instância) em 30 semanas do ano letivo, a escola/polo universitário e o ano letivo, levando assim a um total de 32 atributos de treino. A variável a ser predita indica se o aluno passa, reprova, necessita de repetir o semestre, de ir a exame ou de repetir a unidade curricular. Para o estudo os autores utilizaram a ferramenta *Weka* com diversos algoritmos, incluindo RF, DT, C4.5 e NN (MLP). Após a realização de treino, testes onde também efetuaram seleção sequencial de atributos, verificaram que é possível efetuar previsões ao fim das primeiras três semanas, tendo o algoritmo C4.5 obtido uma exatidão de 86.20% num ambiente de treino de *Leave-One-Out* e validação cruzada. Os autores concluem que soluções que recorrem a ML podem ser benéficas, sobretudo na deteção antecipada, porém não devem ser só consideradas as previsões dos modelos, mas também o próprio instinto dos professores e/ou comportamento dos alunos.

No estudo [70] é mencionado que cerca de 50 mil alunos sul coreanos do ensino secundário abandonam os seus estudos, assim, com o objetivo de identificar alunos com alta probabilidade de desistência, os autores procuraram uma solução com ML ao desenvolver e treinar modelos de ML. Estes utilizaram um conjunto de dados de 2014 fornecido pelo Instituto de Sistemas de Informação da Educação da Coreia do Sul (NEIS – *National Education Information System*), que abrange 12 mil escolas e 17 cidades/províncias. No total, dispuseram de 165.715 instâncias de treino e 12 atributos selecionados do conjunto de dados original com base nos estudos de referência. Estes 12 atributos abrangem o número de faltas e atrasos e o tempo gasto em atividades extracurriculares (clubes, voluntariado, etc.). Para o treino dos modelos de ML, os autores utilizaram a linguagem de programação R com a biblioteca Caret, dividiram os dados em 80% para treino e 20% para teste, e empregaram o algoritmo *Random Forest*. A variável a ser predita pelo modelo é binária, indicando se o aluno desiste ou não dos estudos. Os resultados obtidos foram: 95% de exatidão, 85% de sensibilidade, 95% de especificidade e 97% de AUC. Os autores concluem que soluções como esta são extremamente úteis para identificar alunos em risco de desistência e sugerem que tais sistemas sejam até mesmo incorporados nos sistemas de informação do NEIS, apesar das preocupações relacionadas à privacidade dos alunos.

Em [71], os autores afirmam que a educação é essencial para o sucesso económico de um país. Na Bélgica, identificaram que 26,9% dos alunos desistem do ensino superior. Para enfrentar este problema, os autores procuraram uma solução baseada em algoritmos de ML capazes de identificar os alunos que iriam desistir. Os algoritmos que estes escolheram foram: LoR, NN e RF. Como dados, foram utilizados dados da Universidade de Liége de três anos letivos (2011/2012, 2012/2013 e 2013/2014) e que, após removerem ruído e instâncias inválidas, obtiveram um total de 6845 instâncias para treino e teste. Dado que dispõem dados de diversos anos letivos, utilizaram o mais recente como conjunto de teste. Os dados incluem informações individuais, histórico educacional e informação socioeconómica dos alunos. A variável a ser predita pelos modelos é binária, indicando se o aluno desiste ou não. Após treinarem e testarem os diferentes modelos de ML, os autores obtiveram resultados semelhantes com os três algoritmos utilizados. Contudo, *Random Forest* produziu o melhor resultado, com 81% de exatidão na predição dos alunos que desistem. Os autores concluem o seu estudo com a recomendação do uso de níveis de incerteza quando a predição apresenta um nível de confiança reduzido, e afirmam que a sua solução pode ser aplicada a outro tipo de problemas.

Os autores em [67] mencionam a existência de estudos demonstrativos que o melhor período para prevenir a desistência de alunos no ensino superior é durante o primeiro ano. Com o crescente interesse em soluções baseadas em ML, os autores procuraram desenvolver uma solução utilizando técnicas dessa área. Para conduzirem este estudo, optaram pelo uso da estratégia AutoML (*Automated Machine Learning*) para treinar e otimizar os seus modelos ML, empregando a ferramenta Auto-Weka. Os dados utilizados foram recolhidos de diversas instituições de ensino dos Emirados

Árabes Unidos, totalizando 1491 instâncias com 12 atributos, incluindo informações individuais (sexo e grupo etário) e académicas (curso frequentado, estado da bolsa, entre outros). Dado que o conjunto de dados estava desequilibrado, os autores recorreram à utilização do algoritmo SMOTE para o balanceamento. O objetivo é predizer se o aluno transita de ano, verificando se o valor predito para a média de pontuação académica (GPA – *Grade Point Average*) é igual ou superior a 2. Os autores optaram por utilizar um *ensemble* (agrupamento) de vários algoritmos de ML, incluindo NN, DT, KNN, NB, SVM e LoR, com uma estratégia de validação cruzada de tamanho 10. Após a avaliação dos modelos de ML treinados, obtiveram um com 75,9% de exatidão geral e 83% de exatidão para os alunos que reprovam. Estes concluem que o uso de ensemble e AutoML é vantajoso comparado com os resultados existentes, e que a adoção de tais soluções pode reduzir os recursos necessários para a identificação manual deste tipo de alunos pelas instituições.

Em [72], os autores focaram-se no desenvolvimento solução capaz de identificar alunos em risco de desistência ou reprovação num único curso. A metodologia utilizada baseia-se em *quizzes* durante as aulas, conhecido como *Peer Instruction*, onde os professores fazem perguntas e os alunos respondem usando um dispositivo denominado por *clicker*. Ao contrário de outros estudos apresentados neste documento, não foram utilizados dados históricos, mas apenas as respostas às perguntas recolhidas durante as aulas. Assim, a solução proposta pelos autores envolve treinar modelos de ML com base nas respostas corretas e incorretas dos alunos, sendo agrupadas por semanas (num máximo de 10), para predizer se um determinado aluno irá desistir ou reprovar em uma determinada unidade curricular. Para isso, os autores utilizaram exclusivamente o algoritmo SVM com *Radial Basis Kernel*, comparando resultados preditivos em treinos iterativos com semanas adicionais (começando apenas com a primeira semana e terminando num conjunto de treino composto por 10 semanas). Os melhores resultados foram obtidos com um AUC de 79% para a unidade curricular *CS1-Python* enquanto o pior foi 65% para *Advanced DataStruct*. Constatou-se que a partir da terceira semana ou sexta semana, dependendo da unidade curricular, os resultados preditivos não apresentavam melhorias significativas com a utilização de mais semanas de dados. Os autores ressaltam que, para esta solução ser eficaz, as perguntas impostas pelos professores devem estar diretamente correlacionadas com os exames e concluem que esta abordagem permite aos professores identificar mais facilmente os alunos em risco.

Em [73], os autores propõem uma solução baseada em técnicas de ML para ajudar a universidade a reter alunos. A solução foca-se na identificação de alunos em risco de desistência, permitindo que, uma vez identificados, lhes seja oferecido suporte para concluir os seus estudos com sucesso. Estes utilizaram dados individuais dos alunos (sexo, idade, condição socioeconómica, entre outros) e dados académicos do primeiro ano (notas, presenças, faltas, entre outros) do ano letivo 2012/2013. A variável a ser predita é se um determinado aluno desiste ou não. Os autores utilizaram o algoritmo *Logistic Regression* para selecionar os melhores atributos de treino e *MultiLayer*

Perceptron Neural Network para realizar a predição para novas instâncias. Com uma divisão de dados em 95% para treino e 5% para teste, os autores relatam uma exatidão de 90% para a rede neuronal, porém, com apenas 16% de sensibilidade para os alunos que desistem. Por fim, concluem que a implementação de um sistema capaz de identificar alunos em risco de desistência é benéfica, permitindo que a universidade identifique estes alunos e forneça os meios necessários para que os mesmos não desistam e possam concluir os seus estudos com sucesso.

Os autores de [74] afirmam que a educação é essencial para o progresso de uma nação e para o sucesso pessoal. Com isso, os autores procuraram desenvolver uma solução baseada em técnicas de ML capaz de prever se um aluno conseguirá passar numa determinada unidade curricular. Para este estudo os autores utilizaram um *dataset* público denominado por "xApi-Edu-Data" da plataforma de gestão de ensino *Kalboard 360* referindo que este é composto por dados de quatro anos, entre 2016 e 2020, por 4266 instâncias e 11 atributos referentes ao desempenho académico dos primeiros dois anos. Inicialmente, a variável a ser predita era a nota na unidade curricular *Data Structures*, porém, os autores transformaram-na numa predição binária, indicando apenas se o aluno irá reprovar ou não. Para codificar os dados, utilizaram o algoritmo *Ordinal Encoding* enquanto para equilibrar número de instâncias, aplicaram os algoritmos *Random Oversampling*, *SMOTE* e variações deste último. Posto isto, os autores treinaram os seus modelos de ML com os algoritmos de ML: DNN, KNN, SVM, LoR, DT, RF e GB. Quando testados no *dataset* não equilibrado, o algoritmo *Gradient Boosting* obteve 91% de exatidão, mas apenas 9% de F1 para a classe com valor “reprovado”. Já o melhor resultado foi obtido esse foi obtido par ao modelo de ML treinado pelo algoritmo DNN e no *dataset* cujo a técnica de balanceamento foi a SMOTE, sem adaptações, onde alcançou uma exatidão de 89% e um F1 médio ponderado de 89% e de 40% para a classe com valor “reprovado”. Os autores deste estudo concluem que é evidente o impacto negativo de um *dataset* desequilibrado nos resultados e destacam a necessidade de garantir que os dados sejam equilibrados para obter melhores resultados.

Em [75], os autores afirmam que a desistência dos alunos causa diversos problemas económicos, sociais e políticos, entre outros. Adicionalmente relatam um estudo que menciona que 75% das desistências ocorrem nas primeiras semanas de aulas. Posto isto, com o objetivo de identificar alunos em risco de desistência, os autores investigaram uma solução baseada em técnicas de ML. Estes utilizaram um *dataset* da Universidade *Constantine the Philosopher*, na Eslováquia, contendo dados de 2016 a 2020, com 261 instâncias e 12 atributos referentes ao desempenho académico e acesso ao ensino superior (exemplo: nota de acesso). De forma a codificar os atributos nominais, os autores utilizaram o algoritmo *One-Hot Encoder*. O seu modelo preditivo foi treinado para utilizar a técnica de *stacking*, que envolve o treino e uso de um *ensemble* composto pelos algoritmos RF, *XGBoost*, GB e um modelo final treinado com o algoritmo *Feed-Forward Neural Network* (FNN). Os autores realizaram a otimização de hiperparâmetros com a técnica *grid-search*, relatando os melhores valores para cada

parâmetro no estudo. Após o treino e teste do modelo preditivo resultante da técnica *stacking*, compararam os resultados com aqueles obtidos por modelos ML treinados individualmente e afirmam que o ensemble obteve o melhor resultado, sendo ele 93% de precisão e recall e 92% de F1 de média ponderada. Estes concluem que soluções como a proposta são benéficas para a identificação e prevenção da desistência no ensino superior, e que a utilização de técnicas como *ensemble* e *stacking* oferece vantagens significativas em comparação com o uso de modelos ML treinados com um único algoritmo.

Os autores de [76] relatam que um em cada três alunos espanhóis desiste da universidade, tornando a Espanha um dos países com menor aproveitamento do ensino superior. Para abordar este problema, os autores investigam uma solução que realiza previsões em cinco momentos diferentes no percurso de um aluno de um curso, em que cada momento é predizido se um aluno vai desistir. Sendo a primeira previsão feita no ato da matrícula e as restantes no fim de cada semestre subsequente, que por sua vez utilizam o valor predito no momento anterior. Os dados recolhidos pelos autores em conformidade com o RGPD abrangem o período de 2012 a 2019. As categorias de dados incluem informações de matrícula, progresso académico e de bolsas. A variável a ser predizida pelo modelo preditivo é binária, indicando se o aluno concluirá o curso. Os autores antes de treinarem os seus modelos de ML, efetuaram um pré-processamento aos dados e equilibraram o *dataset* usando os algoritmos *SMOTE* e *Tomek Links*. Para o treino recorreram ao uso dos algoritmos GB, RF e SVM, onde adicionalmente também treinaram um ensemble destes três. Complementarmente, realizaram otimização de atributos via *GridSearchCV*. No momento da matrícula, o melhor resultado foi obtido pelo *ensemble* com 74,79% de *recall*, mas recomendam o uso do *Gradient Boosting* devido à sua menor complexidade com um resultado de 72,34% de *recall*. Já as previsões semestrais, o melhor resultado é observado no quarto semestre, onde o modelo produzido pelo algoritmo SVM alcança os 91,5% na métrica *recall*. Os autores concluem que os resultados documentados são promissores, especialmente logo após o primeiro semestre onde existe um aumento de 10% na métrica *recall* e esperam que futuros estudos consigam encontrar novas formas de melhorar os resultados.

Por último, em [77], os autores portugueses propõem uma solução baseada em técnicas de *data mining*, capaz de segmentar os alunos e prever o grupo a que pertencem, dependendo do seu desempenho escolar. Com esta segmentação, os autores indicam ser possível ajudar os alunos com maiores dificuldades e promover ainda mais o sucesso daqueles que já apresentam bom desempenho, a fim de melhorar a qualidade de ensino na instituição. Para determinar o desempenho de um aluno, os autores utilizaram uma fórmula denominada por *AP Score*, que mede o desempenho com base no número de ECTS concluídos e na nota associada a cada unidade curricular. Este cálculo é discriminatório, pois, penaliza os alunos que repetem unidades curriculares ou obtêm as suas melhores notas em unidades de menor peso (menor número de ECTS). O valor resultante deste cálculo foi dividido em cinco categorias

representativas do desempenho escolar, permitindo, adicionalmente, o treino de classificadores através de algoritmos de ML. Os dados utilizados referem-se a alunos matriculados entre 2003 e 2007, na Faculdade de Engenharia da Universidade do Porto, cuja inscrição estende-se até ao ano letivo de 2014/2015, independentemente de terem terminado ou não o curso. Contudo, os autores informam ter optado por remover todas as instâncias de alunos que desistiram do ensino, o que leva a um *dataset* composto por 2459 instâncias. Os atributos do *dataset* referem informações de matrícula, dados socioeconómicos, histórico académico (ex.: média do ensino secundário) e progresso académico (nota média e ECTS concluídos ao fim de cada semestre do primeiro ano) desses mesmos alunos. Os autores treinaram os seus modelos de ML com a utilização do software *RapidMiner* e de os algoritmos de ML: RF, NB, DT, SVM, *Bagged Trees* e *Adaptive Trees*. Após o treino e depois de serem avaliados os diferentes modelos resultantes, os autores destacam que o algoritmo *Random Forest* obteve o melhor resultado, alcançando 96,1% de exatidão e precisão e *recall* superiores a 90% para cada categoria da classe (*AP Score*). Os autores concluem o seu estudo apresentando diversos cenários possíveis de novos processos a serem criados para cada um dos grupos de *AP Score*, como a recomendação de tutoria pelos melhores alunos aos alunos identificados com pior desempenho. Complementarmente, destacam que o uso de modelos preditivos é uma solução benéfica aos responsáveis pelas estratégias e políticas das instituições ao permitir com que estes tomem decisões mais eficientes e consigam gerir melhor os recursos disponíveis.

Tabela 3- Ano de publicação, número de citações, momentos de predição e informação do dataset(s) utilizados

Ref	Ano	Totais citações	Dados	Momentos da predição	País	Atributos mais relevantes	Classe
[68]	2022	150	Nota do exame intercalar, escola e departamento	Posterior à receção da nota do exame intercalar	Turquia	Não mencionado	Nominal , representativa do intervalo de nota do exame final
[69]	2018	137	Presenças dos alunos	Ao fim das primeiras três semanas	Reino Unido	Escola, Semana 4, Semana 5 e Semana 3	Nominal , correspondente à situação final
[70]	2018	114	Assiduidade, pontualidade tempo de participação em outras atividades e outras	Não mencionado	Coreia do Sul	Não mencionado	Binário de desistência
[71]	2017	103	Informações individuais, socioeconómicas e histórico educacional	Após matrículação / inscrição do aluno	Bélgica	Não mencionado	Binário de desistência
[67]	2020	101	Informações individuais e escolares	Após matrículação / inscrição do aluno	Emirados Árabes Unidos	Não mencionado	Binário de aprovação/reprovação
[72]	2019	75	Respostas às perguntas colocadas pelos professores	Ao fim da terceira ou sexta semana dependendo da UC	Estados Unidos da América	Não mencionado	Binário de aprovação/reprovação
[73]	2020	74	Informações individuais e académicas	Ao fim do primeiro ano	Taiwan	Ranking percentual, se o aluno pediu empréstimo, número de faltas, número de UCs em alerta	Binário de desistência
[74]	2021	61	Notas de diversas UCs	Fim do primeiro semestre	Não referido	Não mencionado	Binário de aprovação/reprovação para uma UC
[75]	2022	49	Nota de acesso, testes, exames e projetos	Não mencionado	Eslováquia	"tests", "access", "project"	Binário de desistência

[76]	2021	25	Informações individuais e acesso (matrícula), progresso académico e bolsa	Após matrículação / inscrição do aluno	Espanha	Não mencionado	Binário de desistência
[77]	2018	148	Dados socioeconómicos , matrícula, histórico académico, resultados das UCs do primeiro ano académico	Após matrículação / inscrição do aluno	Portugal	Média da graduação anterior (acesso), Notas dos exames de acesso, Nota média do primeiro semestre e nota média do segundo semestre	Nominal , correspondente à performance no fim de curso

Tabela 4 - Extração dos algoritmos utilizados, incluindo os de treino (ML), e algoritmo que treina o melhor modelo de cada artigo analisado

Ref	Algoritmos de codificação	Algoritmos de balanceamento	Algoritmos de treino	Melhor algoritmo de treino	Solução implementada?
[68]	Nenhum	Nenhum	NB, RF, SVM, NN, LoR, KNN	RF com 74.6% exatidão, 75.2% precisão, 74.6% recall e 72.1% F1	Não
[69]	Nenhum	Nenhum	RF, C4.5, NN (MLP), NB, DT	C4.5 com 86.2% de exatidão	Não
[70]	Nenhum	Nenhum	RF	RF com 95% exatidão, 85% recall, 95% precisão e 97% AUC	Não
[71]	Não mencionado	Nenhum	RF, NN (MLP) e LoR	RF com 81% de exatidão	Não
[67]	Não mencionado	SMOTE	NB, DT, SVM, LoR, NN, KNN	Ensemble dos algoritmos com 75.9% de exatidão	Não
[72]	Nenhum	Nenhum	SVM	SVM com 79% de AUC	Não
[73]	Nenhum	Nenhum	NN (MLP), LoR	NN (MLP) com 90% exatidão	Não
[74]	Ordinal Encoder	SMOTE, Random Oversampling, ADASYN, SMOTE ENN	DNN, DT, LoR, SVM, KNN, RF, GB	DNN com 89% F1-Score (média ponderada)	Não
[75]	One-Hot Encoder	Nenhum	RF, XGBoost, GB, Feed-Forward NN	Stacking Ensemble dos algoritmos com 92% F1-Score (média ponderada)	Não

[76]	One-Hot Encoder	SMOTE e Tomek Links	GB, RF, SVM	Ensemble dos algoritmos com 75% de recall*	Não
[77]	Não mencionado	Nenhum	RF, DT, SVM, NB, Bagged Trees e Adaptive Trees	RF com 96.1% exatidão e recall e precisão superiores a 90% para todas as classes	Não

3.3. Discussão dos resultados

A análise dos artigos revelou que todos os autores investigaram, com base em técnicas de ML, soluções capazes de identificar alunos com maior risco de desistência ou com dificuldades em concluir os seus estudos com sucesso. Embora todos os estudos tenham procurado soluções para problemas semelhantes, constatou-se que os diferentes estudos realizaram diferentes tipos de predição pelos modelos de ML. Importa destacar que, apesar da diversidade nos tipos de predição, todas as soluções são fundamentadas em classificadores.

Estes diferentes tipos de predição analisados podem ser agrupados em quatro categorias: predição do intervalo de nota (de um exame ou avaliação futura); situação final ou desempenho académico; indicação de possível desistência do curso; ou indicação de que o aluno irá reprovar ao fim do ano ou na avaliação da unidade curricular (UC). Depois de serem contados o número de estudos em cada uma destas categorias, conforme ilustrado na **Figura 11**, verifica-se que a predição de desistência é a mais comum. Em contraste, a predição do intervalo de nota foi a menos frequente, pois, foi apenas abordada por um estudo.

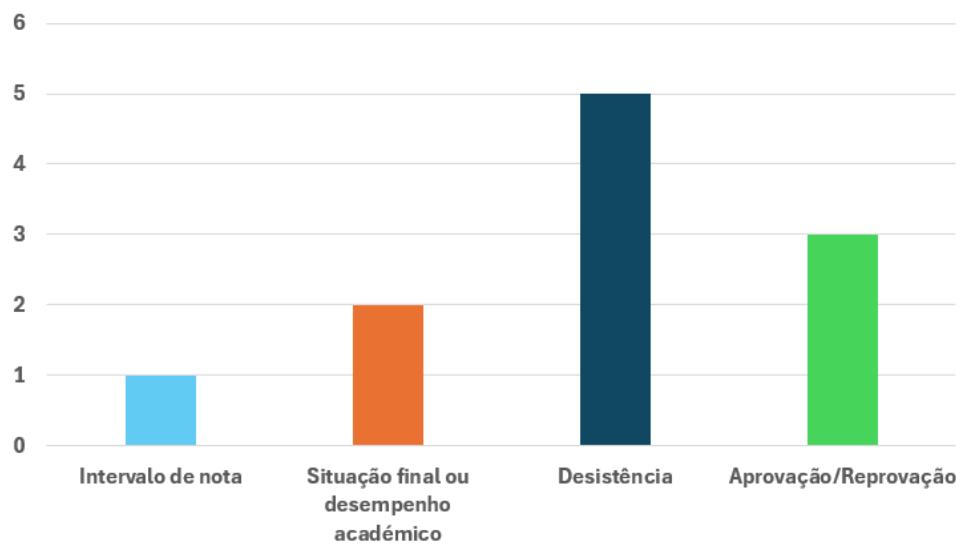


Figura 11 - Gráfico de barras com o tipo de predições feitas pelos estudos analisados

Adicionalmente, quando verificado o país associado a cada um dos estudos ou dos dados utilizados pelos autores, é evidente que soluções para este tipo de problemas são exploradas globalmente. Dos 11 estudos analisados, nenhum país aparece mais de

uma vez e confirma-se a existência de países de continentes diferentes, o que indica uma ampla distribuição geográfica nos estudos analisados. Contudo, os autores de [74] não fornecem qualquer indicação sobre o país ou a instituição de ensino associada aos dados utilizados. No entanto, pode-se inferir que os dados são provenientes da Universidade de Almançora, no Egito, uma vez que esta é a universidade à qual os autores estão vinculados, seja como estudantes ou profissionais.

Dado que os estudos analisados foram realizados por autores de diferentes países e continentes, observou-se também uma variação nos dados disponíveis e utilizados em cada investigação. Esta diferença é esperada, considerando que as diversas instituições operam de maneira distinta e, por conseguinte, possuem dados que refletem as suas necessidades específicas. Contudo, é possível identificar semelhanças nos dados utilizados, especialmente nos dados individuais (socioeconómicos ou sociodemográficos) dos alunos, como sexo, idade, habilitações académicas anteriores, habilitações académicas dos pais, entre outros. No entanto, os dados académicos são o que possuem mais diferenças entre estudos, uma vez que cada instituição possui métodos de avaliação diferentes.

Porém nem todos os estudos optaram por utilizar as mesmas categorias de dados no treino dos seus modelos de ML. Uns optaram pelo uso de dados individuais dos alunos bem como dados académicos, enquanto outros apenas consideraram dados académicos (avaliações, assiduidade, curso frequentado, entre outros). Uma vez que é raro o caso de utilização dos mesmos atributos, foi criado um gráfico circular (**Figura 12**) com apenas três categorias, mais abrangentes, e são contabilizadas as ocorrências do uso de cada categoria. Desta forma, ao ser observada a **Figura 12**, verifica-se que apesar de todos os estudos terem usado dados académicos, apenas cinco optaram também por utilizar dados individuais do aluno [67], [71], [73], [76], [77].

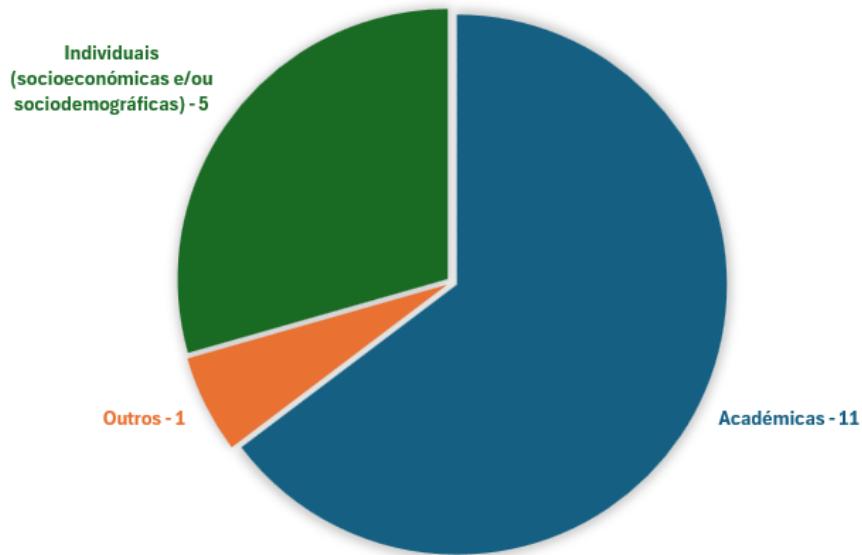


Figura 12 - Gráfico circular das categorias de dados mais utilizados e o seu número de ocorrências

A justificação por de trás da exclusão do uso de dados individuais do aluno só foi apresentada em dois estudos. Os autores de [68] referem que no seu caso é apenas pretendido analisar o impacto das avaliações anteriores na predição de uma avaliação futura, enquanto os do estudo [69] referem o desejo de verificar unicamente o impacto da assiduidade dos alunos na sua predição da situação final do ano letivo de um aluno. Adicionalmente, é importante destacar que a ocorrência da categoria “Outros” refere-se ao estudo [70], pois, este utiliza dados temporais de atividades extracurriculares.

Relativamente à identificação dos atributos mais relevantes do respetivo *dataset* utilizado no treino, apenas quatro estudos analisados forneceram essa indicação [69], [73], [75], [77]. Dado que os dados variam de instituição para instituição, como já mencionado anteriormente, não há semelhança evidente entre os atributos identificados como mais relevantes nestes quatro estudos. O estudo [69] destaca as semanas (onde cada uma contém informação das presenças) mais importantes e a escola frequentada; no estudo [73] os principais atributos são o ranking percentual do aluno, se o mesmo solicitou empréstimo, o número total de faltas e o número de cadeiras em alerta. Já no estudo [75], as notas em diferentes avaliações (“*tests*”, “*access*”, “*project*”) são os atributos mais relevantes, enquanto o estudo [77], que apresenta uma ligeira semelhança com [75], destaca a média da graduação anterior, a nota dos exames de acesso, a média do aluno no fim do primeiro semestre e a do segundo semestre.

No que diz respeito ao momento temporal em que são utilizados os modelos de ML para predizer o valor da classe, este depende sobretudo dos dados necessários e utilizados no treino. Caso os modelos utilizem dados académicos como a assiduidade e/ou avaliações, como é o caso dos estudos [68], [69], [72], [74], [75], [77], estes só conseguem realizar predições depois de serem recolhidos esses dados, que por sua vez só podem ser feitos depois de um período de tempo após o início das aulas. Contudo,

determinados estudos possuem como um dos objetivos a identificação do momento mais antecipado possível, como é o caso dos estudos [69], [72]. No entanto, também pode ser proposto a predição em vários momentos do percurso do aluno como é o caso do estudo [76].

Desta forma, observou-se uma diversidade no momento (ou primeiro momento como é o caso de [76]) em que a predição é realizada pelos modelos de ML, conforme se pode ver na **Figura 13**. Esta figura permite também concluir que o momento com mais ocorrências, embora um número reduzido (três), acontece depois do ato da matrícula e/ou inscrição no ensino superior [67], [71], [76]. Adicionalmente, é importante destacar que dois estudos, [70], [75], não indicaram o momento associado.



Figura 13 - Momentos de predição das várias investigações do estado da arte

Por vezes, parte dos dados utilizados no treino de modelos ML necessitam de uma transformação adicional, pois, os modelos não os conseguem utilizar na sua representação original. Posto isto, apenas em três estudos ([74], [75], [76]) os autores mencionam o algoritmo utilizado neste processo. Destes, dois utilizaram o algoritmo *One-Hot Encoder* ([75], [76]) e um ([74]) utilizou *Ordinal Encoder*. Já os restantes estudos, caso tenham recorrido a este processo, não indicaram o uso de um algoritmo específico.

Adicionalmente, a análise dos 11 estudos revelou que apenas três deles ([67], [74], [76]) optaram por utilizar algoritmos de balanceamento de dados. Este processo, considerado crucial por todos os três estudos, foi fundamental para equilibrar a representação de cada valor possível da variável predita (classe), o que levou os autores a obter melhores desempenhos dos seus modelos de ML treinados. Todos eles empregaram o algoritmo *SMOTE (Synthetic Minority Over-sampling Technique)*. No

entanto, o estudo [74] também comparou os seus resultados utilizando *Random Oversampling*, *ADASYN* e *SMOTE ENN*, enquanto o estudo [76] fez uma comparação com o algoritmo *Tomek Links*. Após a comparação dos resultados com e sem a aplicação desses algoritmos, os três estudos concluíram que o *SMOTE* proporcionou os melhores resultados.

No que diz respeito aos algoritmos utilizados pelos autores na sua investigação, verifica-se que, no total, foram referenciados e utilizados 12 algoritmos distintos. Contudo, dado que foram usadas várias tipologias de redes neurais, todas foram agrupadas sob a designação NN (*Neural Network*). Para uma visualização mais clara, e dado que a contagem de uso por simples observação e contagem das ocorrências na **Tabela 4** não é ideal, foi criado um gráfico de barras, conforme ilustrado na **Figura 14**.

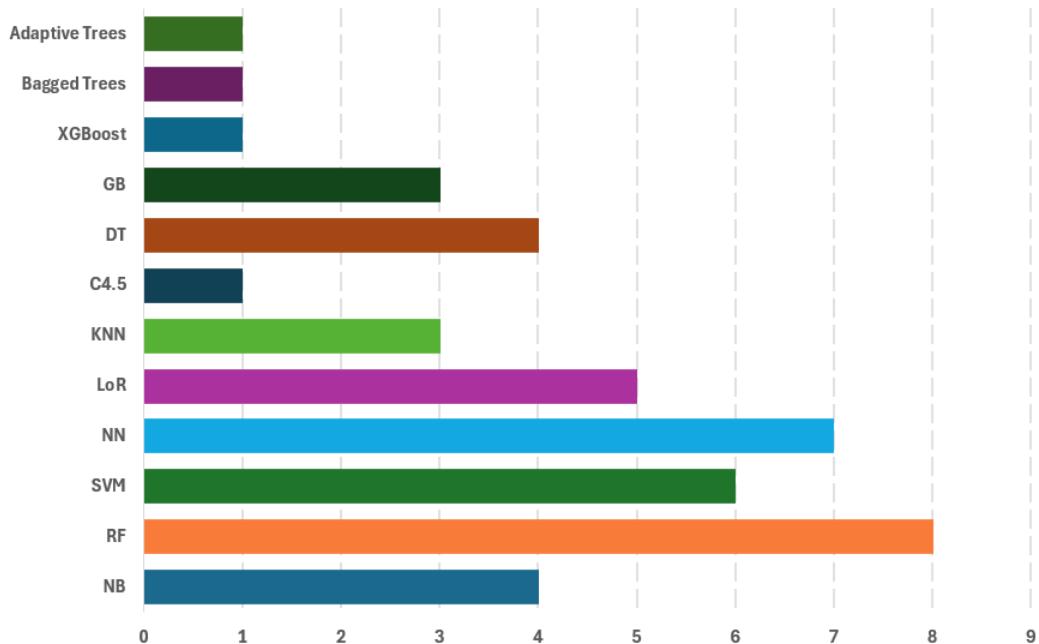


Figura 14 - Número de usos por cada algoritmo de ML utilizado

O gráfico da **Figura 14** revela que o algoritmo mais utilizado nos diferentes estudos é o *Random Forest*, com 8 ocorrências. Em seguida, destacam-se os algoritmos baseados em redes neurais (NN), com sete utilizações, e o algoritmo SVM, com seis. Adicionalmente, é importante denotar que dois estudos ([74], [76]) referenciaram e utilizaram apenas um algoritmo, sendo estes o *Random Forest* e o SVM, respectivamente. Por fim, no que diz respeito ao número de utilizações, observa-se que quatro dos 12 algoritmos foram referenciados apenas uma vez: *Adaptive Trees* e *Bagged Trees* por [77], *XGBoost* por [75], e, por último, *C4.5* por [69].

No que diz respeito ao algoritmo de ML que os autores indicam como sendo o mais eficaz na fase de criação e treino dos seus modelos preditivos, o *Random Forest* revela-se como o mais mencionado. Em seguida, a técnica de *Ensemble* com 3 menções, que, curiosamente, não está representada na **Figura 14**. Isto ocorre porque não se trata de um único algoritmo considerado como melhor, mas sim de uma técnica que combina diferentes algoritmos utilizados pelos autores, que por sua vez mostrou-se superior em

comparação ao uso isolado de cada algoritmo. Contudo, é importante destacar que uma das referências ao algoritmo *Random Forest* ([70]) e a única ao algoritmo *SVM* ([72]) provêm de estudos que utilizaram exclusivamente esses mesmos algoritmos.

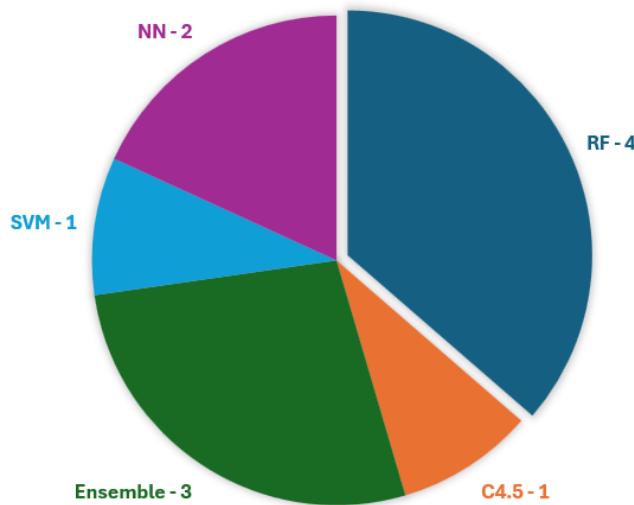


Figura 15 - Algoritmos de ML referenciados como o melhor para o treino

Por último, mas não menos importante, é relevante destacar que nenhum dos estudos analisados mencionou a implementação prática das soluções propostas. Esta observação é particularmente significativa, pois, embora muitos estudos apresentem resultados promissores, não há evidências de que o uso de técnicas de ML tenha efetivamente contribuído para a redução do insucesso escolar nas instituições.

3.4. Principais conclusões

A análise de 11 estudos distintos demonstrou que problemas relacionados ao ensino superior são investigados com técnicas de ML em diversas partes do mundo. No entanto, embora todos os estudos procurem soluções de identificação de alunos em risco ou com maiores dificuldades, as predições investigadas como parte da solução variam. A predição mais comum entre os estudos é se um aluno irá desistir ou não, o que está diretamente alinhado com uma das predições investigadas neste trabalho. Ainda assim, os autores desses estudos também identificaram outras predições, como a previsão do intervalo de nota de uma avaliação futura ou a aprovação/reprovação de uma UC, consideradas igualmente cruciais para a identificação de alunos em risco.

Esta análise também revelou diferenças significativas nos dados utilizados pelos diversos estudos, refletindo as particularidades das instituições que operam em diferentes países e, por sua vez, de maneiras diferentes. Curiosamente, verificou-se que apenas cinco estudos ([67], [71], [73], [76], [77]) incorporaram dados individuais dos alunos, além dos dados académicos, nos seus *datasets* de treino. Esta observação é especialmente relevante, uma vez que era esperado um maior uso do tipo de dados individuais, bem como um impacto preditivo maior deste tipo de dados. No entanto, depois de analisadas as avaliações dos modelos de ML dos diferentes trabalhos,

verificou-se que o uso exclusivo de dados académicos revela-se tão eficaz quanto a combinação destes com dados individuais dos alunos.

No que diz respeito à codificação de dados, constatou-se que poucos estudos mencionaram como modificaram os dados nominais para acomodar os algoritmos de ML, sendo que esta informação foi apenas fornecida por três estudos diferentes ([74], [75], [76]). Embora parte dos dados de maioria dos estudos analisados seja numérica, como as notas das avaliações realizadas pelos alunos, três estudos ([67], [71], [77]) não detalharam como realizaram de codificação dos seus dados nominais. Uma vez que utilizam este tipo de dados, seria vantajoso especificar como foram codificados, devido ao impacto que esse processo tem no treino dos modelos ML, especialmente neste contexto [78]. Uma exceção é o estudo [73], que apresentou dados nominais binários (apenas dois valores possíveis), o que leva a presumir que foi realizada uma codificação binária de cada atributo.

Adicionalmente, é importante destacar a diferença de momentos de predição abordados por cada estudo, uma vez que os autores devem considerar o tempo necessário para implementar intervenções ou processos que auxiliem os alunos identificados. Quanto mais cedo a predição for realizada, mais eficazes poderão ser essas intervenções, um ponto mencionado e desejado por todos os estudos analisados. No entanto, apenas três estudos ([67], [71], [76]) efetuaram predições no momento da matrícula ou inscrição do aluno, que é precisamente o momento de predição a ser investigado no presente trabalho. Já os restantes estudos, à exceção de dois que não especificam o momento de predição ([70], [75]), necessitam sobretudo de dados académicos do aluno, como as notas das avaliações ou as faltas, que por consequência, só conseguem realizar predições após a recolha dessas informações. Esta abordagem pode limitar a capacidade de intervenção antecipada, uma vez que as predições são feitas já ao fim do primeiro semestre ([74]) ou até mesmo só ao fim do primeiro ano ([73], [77]), altura em que alguns problemas académicos graves podem já estar a impactar negativamente os alunos.

Quando identificados os algoritmos de ML utilizados para o treino, constatou-se que o *Random Forest* se destaca como o mais promissor, sendo não apenas o mais utilizado, mas também frequentemente apontado como o algoritmo que treina o melhor modelo preditivo. É importante ressaltar que, embora as técnicas de uso de *ensembles* tenham sido identificadas três vezes ([67], [75], [76]) como as mais eficazes, estas combinam diferentes algoritmos de ML, e apenas o ensemble de [67] não inclui o *Random Forest*. Desta forma, estes resultados sugerem que o *Random Forest* se adapta particularmente bem a este tipo de problema, mostrando-se eficaz na identificação de alunos em risco.

Por fim, embora os resultados dos estudos analisados sejam positivos, os modelos preditivos permanecem teóricos e não foram aplicados em contextos reais. Adicionalmente, importa destacar que esta conclusão se fundamenta apenas nas informações disponíveis nos estudos, e é possível que certas soluções tenham sido implementadas, mas não divulgadas. Além disso, a necessidade de atualizar os modelos

de ML com dados de novos anos letivos não recebeu a devida atenção. A atualização contínua é fundamental para que os modelos se adaptem a possíveis alterações no perfil dos alunos. Esta necessidade também permite validar a eficácia dos modelos e avaliar se a adição de novos dados ou o treino com dados mais recentes (por exemplo, dos últimos dois anos letivos) proporciona melhorias significativas.

4. Tecnologias e ferramentas utilizadas

Para a elaboração deste trabalho foi necessário utilizar várias tecnologias e ferramentas. Estas serão apresentadas, neste capítulo, juntamente com a justificação da escolha de cada uma.

4.1. Python

Python é uma linguagem de programação de alto nível, orientada a objetos e interpretada. Adicionalmente *Python* é dinamicamente tipada, ou seja, uma variável ao longo do seu uso pode mudar de tipo. Por exemplo, da primeira vez que uma variável é usada pode-lhe ser atribuído um valor do tipo *string* e posteriormente ser-lhe atribuído um valor numérico do tipo *int* ou *float*. É uma linguagem de programação considerada de fácil aprendizagem devido à sua sintaxe muito semelhante ao inglês falado [79].

Uma notável vantagem do *Python*, no contexto de ML e DL, é a presença de bibliotecas bem estabelecidas, tais como a *Scikit-Learn*, *TensorFlow*, *pylearn2*, *Keras* e *PyTorch* [80], [81]. A comunidade de desenvolvimento em *Python* também conta com outras bibliotecas e/ou frameworks para análise e manipulação de dados, como *NumPy*, *Pandas* e *Matplotlib* [81].

Posto isto, devido à notável presença e progresso do *Python* na área de ML, análise e manipulação de dados, optou-se, para o desenvolvimento deste trabalho pela utilização desta linguagem. É importante salientar que esta escolha foi sugerida e recomendada pelos orientadores. Adicionalmente, o autor já possuía experiência prévia com a linguagem, adquirida através de estágios curriculares anteriores e do conhecimento obtido em algumas das unidades curriculares do curso, nomeadamente em Fundamentos de Inteligência Artificial e Inteligência Artificial, pelo que foi uma opção que se mostrou óbvia.

Por fim, interessa referir que a versão de *Python* escolhida e utilizada no desenvolvimento deste trabalho é a 3.11. Para a documentação do código *Python* desenvolvido, adotou-se o estilo *numpydoc* [82].

4.2. Jupyter Notebook

Jupyter Notebook é um projeto de código aberto que permite a criação de documentos divididos em células. Cada célula poderá conter código ou texto (no formato *Markdown*) e é possível a execução das células de código selecionadas ou de forma sequencial [83]. Esta divisão por células permite uma melhor divisão do código e apresentação de resultados de execução, pois, cada uma, consequentemente, possui uma célula de *output* associada.

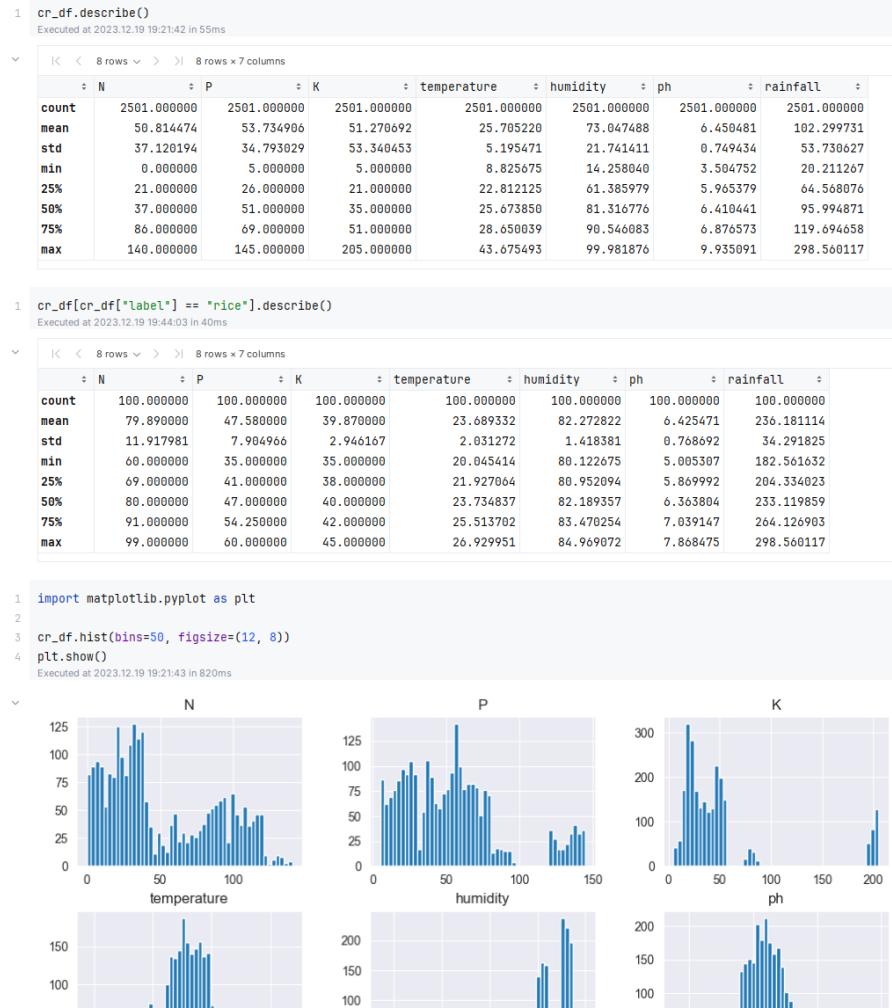


Figura 16 - Exemplo de divisão por células de código e *output* da sua execução num ficheiro *Jupyter Notebook*

A adoção desta tecnologia, com o uso dos seus ficheiros (*notebooks*), prende-se sobretudo pela vantagem, já referida, de melhor divisão de código e apresentação do respetivo *output*. Adicionalmente, versões semelhantes e/ou versões adaptadas de *Jupyter Notebook* (devido a ser código aberto) estão a ser adotados nas grandes plataformas de desenvolvimento de ML como o *Kaggle* e *Google Collab*.

4.3. Scikit-learn

Scikit-learn é uma biblioteca de código aberto para *Python* que permite a utilização de vários algoritmos de ML de treino supervisionado e não supervisionado. Esta biblioteca foi desenvolvida com o auxílio de outras bibliotecas como a *NumPy*, a *SciPy* e a *Matplotlib*. Sendo uma biblioteca com algoritmos de ML, mostra-se, particularmente, útil em problemas de classificação, regressão, agrupamento (*clustering*), redução de dimensão e métricas de comparação e validação dos modelos de ML criados [84].

Dado que o objetivo deste trabalho consiste na utilização de ML para a previsão de nível de risco de alunos do IPCB, esta foi a biblioteca escolhida para o treino dos modelos de ML. Esta foi, também, a recomendação dos orientadores no início do projeto, precisamente por já possuir uma vasta gama de algoritmos de ML implementados tais como: *Random Forest*; *SVM*; *Naïve Bayes*; Árvore de decisão e ferramentas de suporte a validação dos mesmos como o cálculo de exatidão, precisão, *recall*, *F1-score* e validação cruzada [84]. Salienta-se que a sua utilização exigiu diversas consultas à documentação disponível em [84] ao longo do desenvolvimento do presente trabalho.

4.4. Pandas

Pandas é uma biblioteca de código aberto em *Python* que permite a manipulação de conjuntos de dados (*datasets*). Esta biblioteca implementa a classe *DataFrame*, proporcionando uma manipulação de dados rápida, eficiente e indexada. Além disso, permite a alteração do *dataset* através de remodelação, cortes, junção de *datasets* e diversas outras operações disponíveis [85]. Na **Figura 17**, é demonstrado um exemplo de como carregar um ficheiro do formato *CSV* (*Comma-separated values*) para um objeto da classe *DataFrame*, seguido de uma visualização simples do mesmo.

```
In 1 import pandas as pd
2
3 dataset_name = "Manikanta_CR"
4
5 csv_df = pd.read_csv(f"../Data/Clean/{dataset_name}.csv")
6
7 print(csv_df)
Executed at 2024.01.26 19:15:51 in 326ms
```

	N	P	K	ph	EC	S	Cu	Fe	Mn	Zn	B	\
0	143	69	217	5.9	0.58	0.23000	10.20	116.35	59.96	54.85	21.29	
1	170	36	216	5.9	0.15	0.28000	15.69	114.20	56.87	31.28	28.62	
2	158	66	219	6.8	0.34	0.20000	15.29	65.87	51.81	57.12	27.59	
3	133	45	207	6.4	0.94	0.21000	8.48	103.10	43.81	68.50	47.29	
4	132	48	218	6.7	0.54	0.19000	5.59	63.40	56.40	46.71	31.04	
..
613	41	23	135	5.0	1.67	0.10655	26.00	39.20	206.89	31.09	20.64	
614	49	45	90	5.8	1.98	0.09229	19.00	40.20	91.12	32.68	14.91	
615	131	24	121	4.9	2.24	0.08775	22.00	40.00	94.34	24.93	23.74	
616	131	55	130	5.3	2.48	0.08983	15.00	41.00	92.58	45.73	21.48	
617	129	34	160	4.8	1.08	0.08869	25.00	39.00	259.93	33.49	14.16	
												label
0	pomegranate											
1	pomegranate											
2	pomegranate											
3	pomegranate											
4	pomegranate											

Figura 17 - Exemplo de um *DataFrame* da biblioteca pandas após ser carregado um ficheiro CSV

Devido à forma como a biblioteca *Scikit-learn* está implementada, o uso da biblioteca Pandas é um requisito. Assim sendo, o desenvolvimento deste trabalho exigiu a utilização desta biblioteca o que possibilitou o carregamento, manipulação e armazenamento dos *datasets* utilizados. Para conhecer a oferta presente nesta biblioteca, em particular para compreender o funcionamento da classe *DataFrame*, foi

consultada a documentação disponível em [86] ao longo do desenvolvimento do trabalho.

4.5. NumPy

Assim como as bibliotecas apresentadas anteriormente, *NumPy* é uma biblioteca de código aberto para *Python*. A *NumPy* considera-se como uma biblioteca essencial para a computação científica em *Python* [87]. Permite a utilização de *arrays* multidimensionais (*ndarray*) e outros objetos derivados de *arrays* e matrizes. Adicionalmente, oferece operações rápidas, como operações matemáticas sob *arrays*, manipulação da forma de um *array*, ordenação, seleção, álgebra linear, entre outras funcionalidades [87].

A utilização das bibliotecas *Scikit-learn* e Pandas necessitam da instalação da biblioteca *NumPy*. Consequentemente, esta é um requisito do trabalho, sendo essencial para manipulação de *arrays* ou outros tipos de armazenamento de dados utilizados com as bibliotecas atrás referidas. Para a compreensão do seu funcionamento e utilização, ao longo do desenvolvimento deste trabalho, foram efetuadas diversas consultas à sua documentação disponível em [88].

4.6. Matplotlib

No processo de análise de dados, a utilização de gráficos é fundamental. Como já referido anteriormente, *Python* é uma linguagem particularmente adequada para essa tarefa. Com o auxílio da biblioteca *Matplotlib* torna-se possível criar gráficos representativos dos dados. Nesse sentido, a biblioteca *Matplotlib*, de código aberto e desenvolvida em *Python*, permite a criação desses gráficos através de *plots* [89]. A *Matplotlib* oferece uma ampla variedade de estilos gráficos, nomeadamente gráficos de linha, dispersão, barras, caixa, *pie*, histogramas, entre outros presentes em [90]. Na **Figura 18** são apresentados alguns dos estilos de gráficos disponibilizados por esta biblioteca.

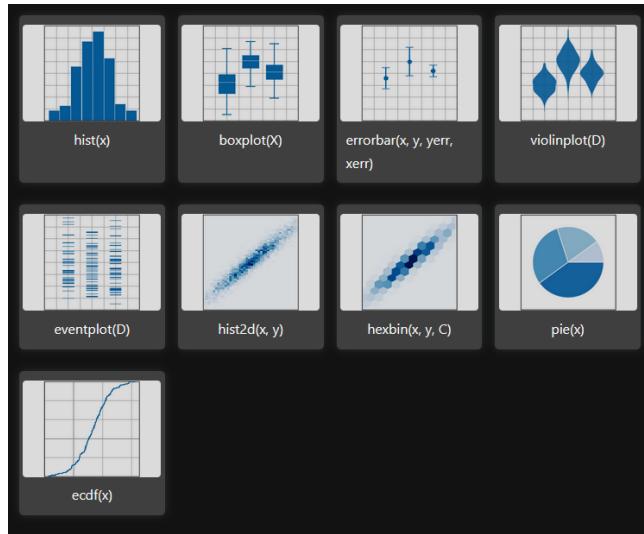


Figura 18 - Exemplos de estilos de gráficos que é possível criar usando a biblioteca *Matplotlib*

Uma vez que as bibliotecas *Pandas*, *NumPy* e *Scikit-learn* requerem a instalação desta biblioteca para o seu funcionamento, a sua utilização mostra-se essencial para análise dos *datasets* utilizados neste trabalho. Adicionalmente, esta biblioteca pode ser complementada com a biblioteca *seaborn*, que será apresentada de seguida. Tal como para as bibliotecas atrás descritas, a utilização da *Matplotlib* implicou a consulta e estudo da respetiva documentação disponível em [91].

4.7. Seaborn

A *seaborn* é uma biblioteca de visualização de dados em *Python*, baseada na biblioteca *Matplotlib*, que possibilita a exploração e representação gráfica de dados como anteriormente descrito [92]. Foi desenvolvida por Michael Waskom e continua a servir como uma *interface* de alto nível, melhorando a experiência de uso da biblioteca *Matplotlib* [92].

A adoção desta biblioteca neste trabalho deve-se, principalmente, à facilitação na criação de gráficos. Ao longo do desenvolvimento, percebeu-se que a criação de determinados gráficos com a *Matplotlib* apresentava-se mais complexa em comparação à utilização desta biblioteca. A **Figura 19** (retirado de [93]) ilustra uma comparação direta do código necessário para a criação de um simples gráfico com a biblioteca *Matplotlib*, presente na parte superior da **Figura 19**, e com a biblioteca *seaborn* presente na parte inferior da figura. Mais uma vez, relembra-se que a sua documentação, acessível em [94] foi consultada ao longo do desenvolvimento do trabalho.



Figura 19 - Comparação do código necessário para um gráfico simples usando as bibliotecas *matplotlib* e *seaborn*

4.8. PyCharm

A ferramenta *PyCharm* é um ambiente de desenvolvimento integrado (do inglês, *Integrated Development Environment*, IDE) para a linguagem de programação *Python*. *PyCharm* é uma ferramenta da propriedade da empresa *JetBrains* e possui suporte multiplataforma, estando disponível nas plataformas *Windows*, *macOS* e *Linux*. *JetBrains* considera esta como uma ferramenta indicada para desenvolvimento web e ciência de dados (inclui ML) [95], [96].

Esta ferramenta está dividida em duas versões, a paga denominada por *PyCharm Professional Edition*, e a gratuita, *PyCharm Community Edition*. Ambas as versões possuem suporte de desenvolvimento e escrita de código em *Python*, suporte a *notebooks* de *Jupyter* e controlo de versões (*Git*). Já a sua versão paga, *Professional Edition*, possui suporte de interpretadores remotos, visualização e manipulação de bases de dados, ferramentas adicionais de análise de dados (como visualização de *DataFrames*, em tempo real, da biblioteca *Pandas* (ver **Figura 20**, retirada de [97])) e um depurador de código mais avançado [95], [96], [98].

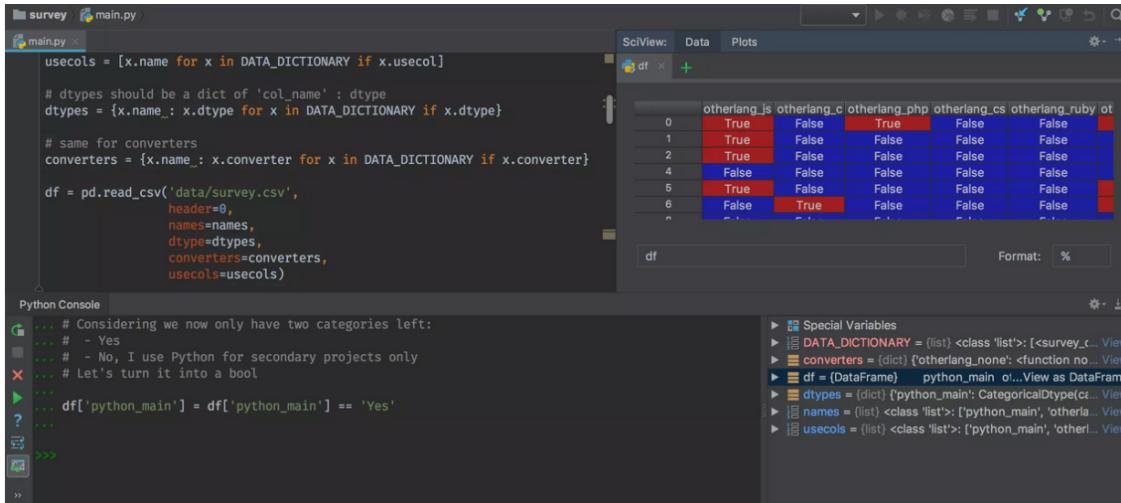


Figura 20 - PyCharm Professional Edition, visualização de um objeto DataFrame em tempo real

Uma vez que o autor do trabalho já estava familiarizado com esta ferramenta e possuía acesso às versões pagas da mesma, como estudante de uma instituição de ensino, a escolha desta ferramenta pareceu óbvia. A integração de controlo de versões, suporte a *notebooks* de *Jupyter* e ferramentas adicionais de análise de dados foram os fatores decisivos para a escolha desta ferramenta.

4.9. Webstorm

A ferramenta *Webstorm* [99] é também um IDE desenvolvido pela *JetBrains*. Esta é projetada para o desenvolvimento de código em *Javascript*, *TypeScript*, *HTML* e *CSS*, sendo assim ideal para o desenvolvimento web. A ferramenta oferece suporte abrangente às bibliotecas de *Javascript*, como *React*, *Angular* e *Node.js*, entre outras [100]. Como parte da suíte de IDEs da *JetBrains* [101], o *Webstorm* está disponível para instalação nas plataformas *Windows*, *macOS* e *Linux*.

Ao contrário do *PyCharm*, a ferramenta *Webstorm* não possui uma versão gratuita. No entanto, dado que o autor possui uma licença escolar, esta permite a descarga e acesso à mesma. Adicionalmente, devido à familiaridade do autor com o ambiente de desenvolvimento do *PyCharm*, foi escolhida a utilização do *Webstorm*, pois ambos possuem um comportamento de desenvolvimento muito semelhante. As funcionalidades que justificaram essa escolha incluem os atalhos para diversas ações, o preenchimento automático e inteligente de código (com suporte para diferentes bibliotecas), a procura instantânea de ficheiros, ações, funções e variáveis por nome em todo o projeto, e a procura e pré-visualização de excertos de código em qualquer ficheiro.

Além disso, a ferramenta permite a definição, programação e execução de diferentes ações da biblioteca em questão, tanto para o desenvolvimento (alojamento do ambiente de desenvolvimento) quanto para as etapas de compilação do código para ambientes de produção. Por fim, é importante destacar que esta ferramenta foi

utilizada para o desenvolvimento da interface da prova de conceito, que tem como base a biblioteca *SvelteKit*, uma biblioteca de *Javascript* e *TypeScript*.

Find in Files 15 matches in 6 files

File mask: *.js

Q <div|

In Project Module Directory Scope

```
<div class="h-full flex items-center justify-center flex-col">
<div class="svelte-announcer" aria-live="assertive" aria-atomic="true" style="position: absolute; left: 0; top: 0; width: 100%; height: 100%; clip: rect(0 0 0 0); z-index: 1000; font-size: 0; background-color: transparent; border: none; margin: 0; padding: 0; border-radius: 0; transition: all 0s ease 0s; ">
  <div>
    <div style="display: contents;"> + body + </div>
  </div>
</div>
<div style="display: contents;"> %sveltekit:body </div>
<div class="min-h-screen grid grid-rows-[auto_1fr_auto]">
  <div class="flex items-center gap-4">
    <!--<div>Footer content</div>-->
  </div>
</div>
```

routes|+page.svelte 1
root.svelte 52
internal.js 21
internal.js 22
app.html 11
+layout.svelte 9
+layout.svelte 12
+layout.svelte 25
+layout.svelte 27

```
root.svelte .svelte-kit/generated
39   return unsubscribe;
40 };
41 </script>
42
43 {#if constructors[1]}
44   <svelte:component this={constructors[0]} bind:this={components[0]} data={data_0}>
45     <svelte:component this={constructors[1]} bind:this={components[1]} data={data_1} {form} />
46   </svelte:component>
47 {:#else}
48   <svelte:component this={constructors[0]} bind:this={components[0]} data={data_0} {form} />
49 {/#if}
50
51 {#if mounted}
52   <div id="svelte-announcer" aria-live="assertive" aria-atomic="true" style="position: absolute; left: 0; top: 0; clip: rect(0 0 0 0); z-index: 1000; font-size: 0; background-color: transparent; border: none; margin: 0; padding: 0; border-radius: 0; transition: all 0s ease 0s; ">
53     {#if navigated}
54       {title}
55     {/#if}
56   </div>
57 {/#if}
```

Figura 21 - Exemplo de procura e pré-visualização de elementos *div* de todos os ficheiros de um projeto *SvelteKit* aberto na ferramenta *WebStorm*

4.9. GitHub

A plataforma *GitHub* permite a hospedagem de código aberto ou não (deixando ao critério do utilizador) com o auxílio da ferramenta de controlo de versões *Git*. A utilização da ferramenta *Git* e a plataforma GitHub permite uma melhor visualização das alterações para cada versão criada (ver **Figura 22**), colaboração com outras pessoas, criação de *branches* (ramificações), criação de *pipelines*, criação de *tickets* informativos de problemas ou novas funcionalidades e essencialmente a capacidade de acesso ao código ou ficheiros a partir de qualquer dispositivo com um navegador web [102].

A decisão de utilizar o *GitHub* e o *Git* revelou-se fundamental para o armazenamento, registo e gestão das diferentes versões do código fonte criadas ao longo do desenvolvimento. Esta plataforma permite a visualização da evolução do projeto através dos *commits* e, caso uma nova versão não funcione corretamente, possibilita a reposição do código fonte para uma versão funcional anterior. Adicionalmente, esta plataforma e ferramenta permitem que o desenvolvimento continue em diferentes dispositivos, uma vez que as versões são armazenadas nos servidores do *GitHub*. Assim, é possível efetuar a descarga do código fonte e modificar

o mesmo em qualquer dispositivo que permita a execução de comandos da ferramenta Git.

miguelmagueijo committed 2 weeks ago

1 parent e569be0 commit c5d4423

Showing 2 changed files with 6 additions and 2 deletions.

Whitespace Ignore whitespace Split Unified

UI/src/components/ModelForm.svelte

```
@@ -60,8 +60,11 @@
 60   60     <form action="{baseUrl}/predict/{id}" on:submit|preventDefault={predictClass}>
 61   61       <div class="grid gap-4 features-grid-columns">
 62   62         {#each features as fName }
+63 -         <div>
+64 -           <label for="input_{ fName }" class="block font-semibold
+63 +             <div class="relative">
+64 +               <div class="absolute hidden bottom-[100%] left-0 right-0 text-xs
+65 +                 { featuresMetadata[fName].help }
+66 +               </div>
+67 +               <label for="input_{ fName }" class="block font-semibold
+68 +                 capitalize truncate overflow-visible">
 65   68                 { fName } {featuresMetadata[fName].full_name ? `($
 66   69                   {featuresMetadata[fName].full_name}` : "")}
 67   70                 </label>
+68 +               <input class="w-full border-2 border-green-500 px-2 py-1"
+69 +                 type="number" id="input_{ fName }" name={ fName }
+70 +                 step="{featuresMetadata[fName].type.includes("int") ? "1" : "0.00001"}"
+71 +                 placeholder="0" required>
```

Figura 22 - Exemplo de alterações de um ficheiro entre a penúltima e última versão no GitHub

4.11. Weka

O Weka é uma aplicação de código aberto desenvolvida em Java, composta por uma variedade de algoritmos de ML direcionados para tarefas de pré-processamento e análise de dados [103], [104], também denominado por *data mining*. Esta aplicação permite treinar modelos de ML capazes de realizar tarefas como a seleção de atributos, classificação, regressão, agrupamento (*clustering*) e criação de regras de associação.

A **Figura 23** apresenta a interface gráfica desta aplicação após ser carregado um ficheiro com dados representativo do problema de identificação de alunos em risco de desistência. Nesta figura, é possível visualizar a distribuição de instâncias para o atributo “tipo_ingresso”, onde a cor azul representa alunos que não desistiram dos estudos e a cor vermelha representa os alunos que desistiram.

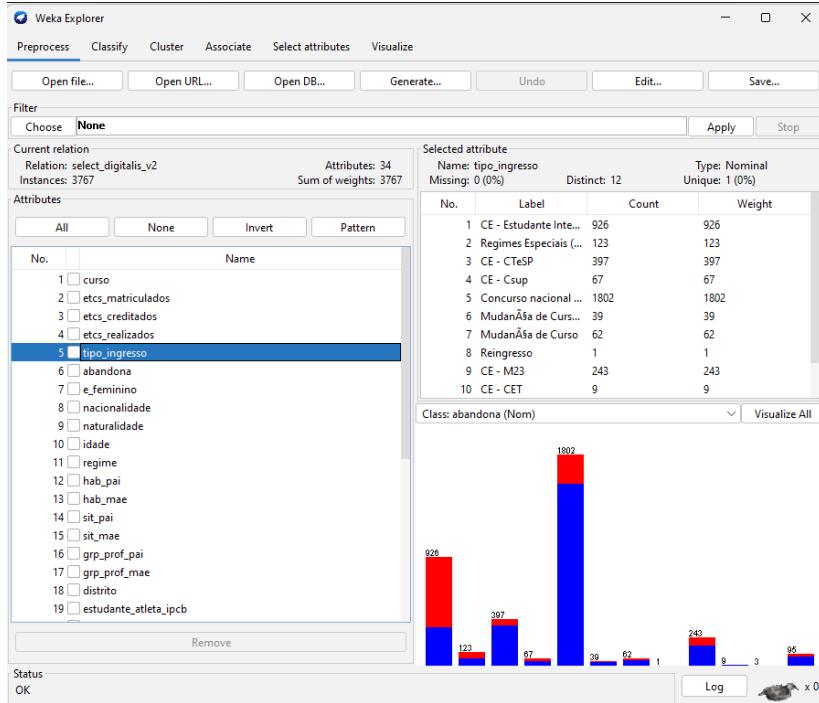


Figura 23 - Interface gr  fica da aplic  o Weka

A decis  o de utilizar esta aplic  o deve-se ao facto de esta possuir determinados algoritmos n  o implementados na biblioteca *Scikit-learn*, nomeadamente: *OneR*, *ZeroR*, *REPTree* e *J45*. Estes algoritmos foram utilizados para comparar os resultados de treino com aqueles obtidos a partir dos modelos treinados atrav  s da biblioteca *Scikit-learn*. Outra raz  o que motivou a escolha desta aplic  o em vez de bibliotecas de *Python*   a exist  ncia de interface gr  fica que por sua vez      il de interagir e compreender. J  a a utiliza  o de bibliotecas de *Python*, exigiria o desenvolvimento de c  digo adicional, que potencialmente poderia n  o ser enriquecedor para o trabalho, uma vez que o objetivo    apenas comparar resultados e n  o treinar modelos prontos para serem utilizados em contextos reais. Por   ltimo, a utiliza  o desta aplic  o permitiu, ocasionalmente, visualizar os dados de diferentes *datasets* sem a necessidade de escrever ou alterar c  digo.

4.12. SvelteKit e Typescript

SvelteKit [105]    uma biblioteca de *Javascript* de c  digo aberto destinada ao desenvolvimento web [106]. Esta permite a divis  o de uma p  gina web em diversos componentes, compostos por c  digo *HTML*, *CSS* e/ou *Javascript*, que podem ser reutilizados em outras p  ginas web. Al  m disso permite aos desenvolvedores escreverem c  digo *Typescript* [105], [107] nos seus componentes, layouts e p  ginas web.

Typescript, desenvolvida pela Microsoft,    um superconjunto de *Javascript* [108], [109] que introduz tipos est  ticos, tornando a codifica  o de c  digo *Javascript* semelhante    codifica  o de outras linguagens de baixo n  vel e estaticamente tipadas, como *C*, *Go* e *Java*.    importante referir que todo o c  digo *Typescript* codificado acabar   por ser convertido em c  digo *Javascript*, sendo assim poss  vel de executar nos

navegadores webs, como o Google Chrome, Firefox, Safari, entre outros, dos diferentes utilizadores.

A escolha do *SvelteKit* torna-se essencial devido às suas funcionalidades de criação de páginas web, componentes reutilizáveis e layouts de páginas. Esta escolha evita a necessidade de alterar inúmeros ficheiros *HTML* sempre que há modificações a serem feitas à barra de navegação ou no rodapé do website. A **Figura 24** mostra um exemplo de código de um ficheiro *Svelte*, que define um layout base aplicado a todas as páginas web. Neste exemplo, todas as páginas partilharam a mesma estrutura base, barra de navegação e rodapé, e o restante conteúdo (*HTML*, *CSS* e/ou *JavaScript*) será *HTML* "<slot />" (linha 20 da **Figura 24**).

Além disso, devido à experiência de desenvolvimento web do autor, que já possui familiaridade com o uso desta biblioteca, *SvelteKit*, a escolha de outra alternativa não foi ponderada. Assim, a biblioteca *SvelteKit* foi utilizada para o desenvolvido da interface da prova de conceito que será posteriormente apresentada neste documento. A versão da biblioteca *SvelteKit* utilizada é a 2.5.18 (*Svelte* - 4.2.18).

```
1 <script lang="ts"> Show component usages
2   import "../app.css";
3   import type { LayoutData } from "./$types";
4
5   export let data: LayoutData; Show usages ± Miguel Magueijo
6 </script>
7
8 <div class="min-h-screen grid grid-rows-[auto_1fr_auto]">
9   <nav class="p-4 border-b-2 border-gray-400 bg-gray-400/10 flex justify-between items-center">
10    <a class="text-xl font-bold" href="/">
11      <span class="text-theGray">Rev</span><span class="text-theBlue">Up</span>
12    </a>
13    {#if data.hasRevSession}
14      <div>
15        <a href="/logout" data-sveltekit-reload> Logout </a>
16      </div>
17    {/#if}
18  </nav>
19  <main>
20    <slot />
21  </main>
22  <footer class="flex justify-between items-center p-4 bg-gray-400/10 border-t-2 border-gray-400">
23    <a class="text-xl font-bold" href="/">
24      <span class="text-theGray">Rev</span><span class="text-theBlue">Up</span>
25    </a>
26    <p>
27      Desenvolvido por
28      <a class="font-bold underline" href="https://github.com/miguelmagueijo">
29        Miguel Magueijo
30      </a>
31      © 2024
32    </p>
33  </footer>
34</div>
```

Figura 24 - Ficheiro da biblioteca *SvelteKit* responsável pelo layout de todas as páginas

4.13. FASTAPI

O desenvolvimento da REST API da prova de conceito foi possível através da utilização da biblioteca *FASTAPI* [110]. Esta é uma biblioteca de código aberto

desenvolvida em Python, dedicada ao desenvolvimento de *APIs*, nomeadamente *REST APIs* para websites. A *FASTAPI* permite ao desenvolvedor definir múltiplas rotas *HTTP*, associando funções a cada uma delas. Dessa forma, quando é efetuado um pedido *HTTP* a uma rota definida, a função correspondente é executada, retornando uma resposta, que por omissão é no formato JSON. A **Figura 25** ilustra a associação de uma função à rota “/user/”, a qual recebe um nome de utilizador (“username”) como parâmetro e retorna os dados desse utilizador presentes na base de dados. Além disso, a biblioteca *FASTAPI* permite o retorno de respostas *HTTP* de erro quando determinadas condições são verificadas, como exemplificado nas linhas 141 e 142 da **Figura 25**, onde uma resposta de erro é gerada caso o utilizador não seja encontrado na base de dados.

```

135 @app.get("/user/{username}")
136     @app.get("/user/{username}")
137     async def get_user_by_username(username: str, request: Request):
138         with request.state.db.cursor(row_factory=class_row(DBUser)) as cursor:
139             cursor.execute("SELECT id, username, is_active FROM auth.user WHERE username = %s", [username])
140             user_data = cursor.fetchone()
141
142             if user_data is None:
143                 return JSONResponse(status_code=HTTPStatus.NOT_FOUND, content={"error": ErrorMessages.user_not_found})
144
145             return JSONResponse(status_code=HTTPStatus.OK, content={"user": {
146                 "id": user_data.id,
147                 "username": user_data.username,
148                 "is_active": user_data.is_active,
149             }})

```

Figura 25 - Exemplo de uma função em *FASTAPI* associada à rota “/user/”

A decisão de utilizar esta biblioteca teve em consideração o facto da biblioteca *Scikit-Learn* também ser em *Python*. A escolha de uma biblioteca para o desenvolvimento de uma API com uso de bibliotecas de outras linguagens de programação tornaria a utilização dos modelos de ML produzidos pelo *Scikit-Learn* um processo muito complexo e difícil de implementar, dado que estes modelos são criados e treinados num ambiente *Python*. Adicionalmente, embora existam outras bibliotecas para desenvolvimento de *APIs* em *Python*, como *Flask* e *Django*, a escolha do uso da biblioteca *FASTAPI* baseou-se no desejo do autor em aprender esta nova biblioteca, na sua fácil aprendizagem e na sua performance [111], [112] em comparação com as outras duas mencionadas.

4.14. PostgreSQL

PostgreSQL é uma base de dados relacional e de objetos de código aberto e o seu lançamento foi em 1996 por *PostgreSQL Global Development Group* [113]. Sendo uma base de dados relacional, esta permite a criação e gestão de várias bases de dados internas e tabelas, que podem ser manipuladas, alteradas, acedidas ou removidas através de código SQL. Esta base de dados suporta uma grande variedade de tipos dados, sendo eles: primitivos (inteiros, numéricos, texto e booleanos); estruturados (datas, arrays, *UUID*, entre outros); documentos (*JSON*, *XML* e outros); tipos menos comuns como dados geométricos e tipos customizados criados pelos seus utilizadores

[113]. Adicionalmente, sendo um projeto de código aberto, não existe qualquer custo associado ao seu uso e alojamento.

Dado que a prova de conceito projetada necessita de um sistema de autenticação, o uso de uma base de dados é crucial. Apesar de existirem outras bases de dados bem estabelecidas e ensinadas nas aulas, como *MySQL*, *SQL Server* e *Oracle*, o autor optou por utilizar o *PostgreSQL* devido à sua crescente popularidade e reconhecimento [114], [115], bem como ao custo associado (quando comparado ao *SQL Server* e *Oracle*). Além disso, a sua escolha também se fundamentou no facto do autor ter vindo a utilizar *PostgreSQL* em projetos pessoais e não existir qualquer requisito indicativo do uso de outra base de dados. No entanto, caso surja a necessidade de ser utilizada uma base de dados diferente, como *MySQL*, a transição não será morosa, dado que a estruturação e a complexidade da base de dados são reduzidas numa fase inicial (prova de conceito).

4.15. Bibliotecas adicionais

O desenvolvimento deste trabalho exigiu a utilização de mais cinco bibliotecas para além daquelas já apresentadas. Porém, uma vez que se considerou não ser necessária a apresentação destas bibliotecas, tendo em conta o seu papel na totalidade do trabalho desenvolvido, e o facto de que essa apresentação tornaria o documento mais extenso, optou-se apenas por fazer uma breve descrição das mesmas. Assim, as bibliotecas adicionais necessárias para este trabalho foram: *XGBoost* [116], *LightGBM* [117], *imbalanced-learn* [118], *skops* [119], *Psycopg3* [120] e *TailwindCSS* [121].

As duas primeiras bibliotecas (de *Python*), *XGBoost* [116] e *LightGBM* [117], permitem a utilização do algoritmo com a mesma nomenclatura que a biblioteca. Uma vez que a biblioteca *Scikit-learn* não implementa estes dois algoritmos, a sua instalação é um requisito para que possam ser utilizados ambos os algoritmos no treino de modelos de ML. Realça-se que ambas as bibliotecas dos algoritmos, *XGBoost* e *LightGBM*, são compatíveis com a biblioteca *Scikit-learn*.

A terceira biblioteca *Python* utilizada, *imbalanced-learn* [118], é destinada a problemas de classificação em que os dados apresentam um desequilíbrio no número de instâncias por classe [122]. Esta biblioteca oferece a implementação de diversos algoritmos de *oversampling* (sobreamostragem) e *undersampling* (subamostragem) para equilibrar a representação da classe a predizer do *dataset* [118], [122]. Uma vez que, os dados utilizados neste trabalho apresentaram um desequilíbrio, foi explorado o uso de três algoritmos de balanceamento: *SMOTE* (*Synthetic Minority Over-sampling Technique*), *Random Oversampling* e *Random Undersampling*, numa tentativa de melhorar os resultados obtidos.

A biblioteca *skops* [119] é uma biblioteca em *Python* que foi desenvolvida pelos autores da biblioteca *Scikit-learn*. A instalação desta biblioteca é necessária para exportar os modelos de ML treinados pelos algoritmos das bibliotecas *Scikit-learn*, *XGBoost* e *LightGBM*. Segue-se, assim, o que é recomendado na documentação da

biblioteca *Scikit-learn* [123] para que a exportação dos modelos de ML seja feita de forma mais segura.

A penúltima biblioteca, *Psycopg3* [120], é uma biblioteca de código aberto desenvolvida para permitir com que os desenvolvedores de código *Python* consigam integrar e utilizar a base de dados *PostgreSQL* nas suas aplicações e/ou programas. A sua instalação é essencial para que os pedidos da REST API (*FastAPI*) possam gerar respostas e/ou validar dados com a informação existente na base de dados.

A última biblioteca, *TailwindCSS* [121] é uma biblioteca de estilos *CSS* com funcionalidades semelhantes ao *Bootstrap* que é uma das bibliotecas mais reconhecidas no desenvolvimento web. Complementarmente, permite modificar as suas classes ou introduzir novas ao ser editado o seu ficheiro de configuração [124]. Além disso, esta biblioteca está preparada para gerar o ficheiro *CSS* com o tamanho mais reduzido possível, incluindo apenas as classes de estilo que foram efetivamente utilizadas em nos elementos de cada página [125].

5. Datasets

No âmbito do projeto REVUP, que abrange os alunos do IPCB, é necessário treinar os diferentes modelos com dados representativos desses alunos. Assim, todos os dados utilizados para o treino e teste dos modelos de ML neste projeto provêm dos sistemas de informação do IPCB, que armazenam dados pessoais e académicos dos alunos.

Numa fase inicial, não foram considerados dados provenientes de fontes externas ou de outras instituições, uma vez que estes não representam o universo dos alunos do IPCB e podem apresentar informações distintas, dificultando a sua integração com os dados existentes do IPCB. Além disso, prevê-se que futuras iterações e/ou alterações do trabalho aqui apresentado possam incluir novos atributos resultantes da recolha de novas informações que originam de novas necessidades ou requisitos emergentes.

Em seguida, neste capítulo, é descrito o processo de recolha de dados e os problemas encontrados no mesmo. É apresentada a composição dos *datasets* (atributos) originais antes de qualquer alteração, as etapas de pré-processamento aplicadas, as diferentes classes possíveis e investigadas para este problema e, por fim, uma breve visualização do *dataset* resultante da junção dos diversos *datasets*.

5.1. Recolha

Uma vez que os sistemas de informação do IPCB armazenam uma vasta diversidade de atributos, é necessário selecionar aqueles que são benéficos para o treino dos modelos de ML no contexto da identificação de alunos em risco de insucesso escolar. Esta escolha é crucial, pois dados como o número de telemóvel, nome e email não são úteis para o treino e introduzem ruído, já que são únicos para cada instância (aluno). Adicionalmente, é necessário garantir o cumprimento das regras e leis impostas pelo RGPD (Regulamento Geral sobre a Proteção de Dados), tendo sido solicitado a exclusão de qualquer dado identificativo dos alunos dos *datasets*.

Dado que existe uma diversidade significativa de atributos possíveis, foi necessário delinear um conjunto adequado para o treino dos modelos. Além dos atributos identificados na análise do estado da arte, foram consultadas diversas soluções para problemas semelhantes na plataforma Kaggle [46]. Com a definição de um conjunto possível de atributos, foi realizada uma sessão de brainstorming com os professores orientadores para validar a escolha dos atributos. Após esta validação, o respetivo departamento interno do IPCB foi contactado pela professora orientadora Ana Paula Silva para ser realizado a geração dos *datasets*, que por sua vez os reencaminhou ao autor ao longo que iam sendo gerados.

Assim, foi solicitado o pedido de extração de dados referentes aos alunos que se inscreveram pela primeira vez em determinado ano letivo. Esta recolha inclui, além dos dados relativos à matrícula/inscrição, informações académicas como o número de ECTS (aprovados, reprovados, creditados, realizados) e a continuidade dos estudos pelos alunos, entre outros. No entanto, sem qualquer dado referente a notas de

avaliações, exceto a nota de acesso. Dado que a predição proposta neste projeto é realizada no ato da matrícula/inscrição, o treino deve ser efetuado apenas com os dados disponíveis no momento da matrícula/inscrição (maioritariamente dados individuais), enquanto os restantes dados serão utilizados para definir as classes a serem previstas. Foram, assim, fornecidos ficheiros representativos de cada ano letivo, abrangendo os períodos de 2019/2020, 2020/2021, 2021/2022 e 2022/2023, a partir dos quais os dados foram extraídos.

A fim de validar os dados extraídos, foi realizada uma breve análise da distribuição das instâncias, seguida de pré-processamento, treino e testes de modelos de ML. Esta validação foi crucial, pois, nas primeiras versões dos *datasets* gerados, foram identificados diversos problemas de extração incorreta e incompleta, com várias instâncias apresentando valores incorretos ou inválidos para o tipo de dados associado.

Ao todo, foram realizadas quatro revisões, nas quais foram corrigidas as incorreções originadas na extração dos dados e acrescentados novos atributos. Todas estas revisões contaram com a participação e feedback do autor, dos professores orientadores, dos funcionários do IPCB responsáveis pela extração dos dados e da empresa Digitalis, que desenvolve parte do software utilizado nos sistemas de informação do IPCB.

Uma vez que a última versão dos dados é a que não apresenta erros identificados até ao momento e é a mais rica em termos de informação, as próximas secções deste relatório, nomeadamente a composição, pré-processamento, visualização dos dados, treino e os testes dos modelos de ML, serão fundamentadas nesta versão. Ocasionalmente, podem ser mencionados resultados de versões anteriores para efeitos de comparação, que por sua vez estão devidamente identificados e justificados. Esta escolha evita possíveis confusões de interpretação dos resultados presentes neste relatório, uma vez que seriam relatados resultados enganosos e estes não acrescentariam qualquer valor analítico.

Por fim, a última versão dos *datasets* inclui quatro ficheiros representativos de diferentes anos letivos: 2019/2020, 2020/2021, 2021/2022 e 2022/2023 e conta com um total de 4820 instâncias. Cada *dataset* possui 39 atributos, iguais entre eles, que por sua vez permite a combinação e correta integração dos quatro num só. Contudo, nem todos estes atributos serão utilizados para o treino, uma vez que alguns contêm informações irrelevantes para esse propósito. Todos eles atributos serão apresentados e descritos no próximo subcapítulo, onde também são indicados quais serão removidos e quais serão utilizados nas próximas etapas do projeto aqui apresentado.

5.2. Composição

Conforme já mencionado, foi necessário realizar quatro revisões aos *datasets*. Considerando que a apresentação detalhada da composição de cada revisão não acrescentaria valor significativo, visto que a última revisão (versão 4) inclui todos os atributos das revisões anteriores, bem como outros adicionados ao longo do processo, optou-se por apresentar apenas os atributos presentes na última revisão.

Antes de serem apresentados os diferentes atributos disponíveis na versão original dos *datasets* (sem qualquer modificação ou alteração), é necessário indicar o número total de instâncias que cada *dataset* contém. Assim, o *dataset* referente ao ano letivo 2019/2020 contém 1155 instâncias, 2020/2021 possui 1243 instâncias, 2021/2022 com 1191 instâncias e por último o *dataset* de 2022/2023 é composto por 1231 instâncias. Quando combinados, obtém-se um *dataset* composto por 4820 instâncias diferentes.

Todos estes *datasets* gerados possuem exatamente os mesmos atributos, permitindo assim que seja possível combinar os mesmos e não sejam descartadas instâncias por falta de valores. Ao todo existem 38 atributos distintos, que por sua vez podem ter correlação direta entre si, como é caso do código do curso e nome do curso.

Dado que o conjunto de dados é composto por 38 atributos, uma apresentação detalhada em texto tornar-se-ia tediosa e extensa para o leitor. Para facilitar a compreensão e organização da informação, foi criada a **Tabela 5**. Esta tabela tem como objetivo indicar o nome de cada atributo, o tipo de dado (numérico ou nominal) e fornecer uma descrição do atributo, incluindo os possíveis valores para cada atributo.

Tabela 5 - Atributos que compõem os *datasets* originais sem qualquer alteração

Nome da coluna	Tipo de dado	Descrição e valores possíveis
CD_LECTIVO	numérico	Descrição: ano letivo associada à instância. Valores possíveis: "201920", "202021", "202122", "202223".
DS_INSTITUIC	nominal	Descrição: nome da escola do IPCB frequentada pelo aluno. Valores possíveis: nome extenso da escola do IPCB das seis possíveis, exemplo: "Escola Superior de Tecnologia de Castelo Branco".
CD_CURSO	numérico	Descrição: código interno do curso frequentado pelo aluno Valores possíveis: valor inteiro positivo
NM_CURSO	nominal	Descrição: nome do curso frequentado pelo aluno.

		Valores possíveis: nome extenso do curso de licenciatura frequentado pelo aluno, exemplo: "Licenciatura em Engenharia informática".
Anulou no ano letivo	nominal	Descrição: indicação se o aluno anulou a sua matrícula a meio do ano letivo. Valores possíveis: "S" (Sim) ou "N" (Não).
Total ECTS Matriculados	numérico	Descrição: número de ECTS inscritos no primeiro ano pelo aluno. Valores possíveis: valor flutuante positivo.
Total ECTS Aprovados	numérico	Descrição: número de ECTS aprovados do aluno ao fim do primeiro ano. Inclui ECTS creditados. Valores possíveis: 0 ou valor flutuante positivo
Total ECTS Creditados	numérico	Descrição: número de ECTS creditados ao aluno em todo o curso. Valores possíveis: 0 ou valor flutuante positivo
Total ECTS Realizados	numérico	Descrição: número de ECTS realizados pelo aluno. Não inclui ECTS creditados. Valores possíveis: 0 ou valor flutuante positivo
Total ECTS Reprovados	numérico	Descrição: número de ECTS reprovados pelo aluno. Subtração do número de ECTS aprovados a ECTS matriculados Valores possíveis: 0 ou valor flutuante positivo
INGRESSO	nominal	Descrição: método de ingresso do aluno Valores possíveis: nome extenso do método, exemplos: "Concurso Nacional de Acesso", "CE - Estudante Internacional", "CE – CTeSP", entre outros
PROSSEGUIU	nominal	Descrição: indicação se o aluno prosseguiu estudos apesar transitar de ano ou reprovar. Valores possíveis: "S" (Sim), "N" (Não).
SEXO	nominal	Descrição: sexo do aluno Valores possíveis: "M" (Masculino), "F" (Feminino)
NACIONALIDADE	nominal	Descrição: nacionalidade atual do aluno Valores possíveis: nome do país, exemplo: "Portugal", "Brasil", "Moçambique", entre outros
NATURALIDADE	nominal	Descrição: naturalidade do aluno Valores possíveis: país no caso de alunos estrangeiros, freguesia no caso de alunos nascidos em Portugal
CD_POSTAL	numérico*	Descrição: primeiros quatro dígitos do código postal atual do aluno Valores possíveis: nulo, valor inteiro positivo de tamanho 4, exemplos: "6000", "2100", "6230", entre outros.
CD_SUBPOS	numérico*	Descrição: últimos três dígitos do código postal atual do aluno

		Valores possíveis: nulo, valor inteiro positivo de tamanho mínimo um e máximo três, exemplos: "8", "93", "356", "479", entre outros. Descrição: localidade associada ao código postal.
LOCALIDADE	nominal*	Valores possíveis: nulo (quando código postal também é nulo), nome extenso da localidade, exemplos: "Castelo Branco", "Fundão", "Cartaxo", entre outros. Descrição: data de nascimento do aluno
DT_NASCIMENTO	nominal	Valores possíveis: data no formato dia/mês/ano. Descrição: idade do aluno no ano da inscrição.
IDADE	numérico	Valores possíveis: valor inteiro positivo Descrição: tipos de aluno internos associados ao aluno da instância
TIPOS_ALUNO	nominal	Valores possíveis: descrição do tipo de aluno, quando vários os mesmos são separados por vírgulas. Exemplos: "Normal", "Normal, Trab. Estudante", "Parcial até 30,5 ECTS" (vírgula neste caso não separa vários tipos), entre outros.
DT_MATRIC	nominal	Descrição: data de registo da matrícula pelo aluno. Valores possíveis: data no formato dia/mês/ano, pode possuir sufixo de hora no formato hora:minutos:segundos.
REGIME	nominal	Descrição: regime de aulas inscrito pelo aluno. Valores possíveis: nome extenso do regime, exemplos: "Tempo Inteiro", "Tempo Parcial até 30,5 ECTS", entre outros.
DS_HABILIT_PAIS	nominal*	Descrição: habilitação académica do pai do aluno. Valores possíveis: nulo, nome da habilitação, exemplos: "Ensino Médio (11º ano)", "12º ano de escolaridade", "Ensino básico 1º ciclo (4ª classe)", entre outros.
DS_HABILIT_MAE	nominal*	Descrição: habilitação académica da mãe do aluno. Valores possíveis: mesmos de DS_HABILIT_PAIS
SIT_PROF_PAIS	nominal	Descrição: situação profissional do pai no ato da matrícula. Valores possíveis: descrição da situação, exemplos: "Desconhecida /Não Tem", "Trabalha por conta de outrem", "Reformado/a", entre outras.
SIT_PROF_MAE	nominal	Descrição: situação profissional da mãe no ato da matrícula. Valores possíveis: mesmos de SIT_PROF_PAIS
GRUPO_PROF_PAIS	nominal*	Descrição: grupo profissional associada à situação profissional do pai.

		Valores possíveis: nulo, descrição do grupo, exemplos: "Outra situação", "Membro das Forças Armadas", "Trabalhadores não qualificados", entre outros.
GRUPO_PROF_MAE	nominal*	Descrição: grupo profissional associada à situação profissional da mãe. Valores possíveis: mesmos de GRUPO_PROF_PAI
CD_SITUA_FINAL	nominal	Descrição: código interno representativo da situação final do aluno Valores possíveis: valor inteiro positivo
SITUACAO_FINAL	nominal	Descrição: descrição da situação final do aluno, no momento da extração dos dados. Valores possíveis: descrição da situação, exemplos "Normal", "Mudança de Curso", "RECOLOCADO", "Graduado", "Anulado", entre outros
NT_INGRESSO	numérico*	Descrição: nota de ingresso do aluno para os alunos que ingressaram pelo concurso nacional de acesso. Valores possíveis: nulo, nota do aluno de 0 a 200 (valor flutuante positivo)
Habilitação anterior	nominal*	Descrição: habilitação académica anterior do aluno Valores possíveis: nulo, descrição da habilitação, exemplos: "12º ano de escolaridade", "CTeSP", "Licenciatura (Pré-Bolonha)", entre outras
Inst. Hab. Anterior	nominal*	Descrição: nome da instituição académica onde o aluno se graduou com a habilitação anterior Valores possíveis: nulo, nome da instituição, valores possíveis: "Universidade Aberta", "ESTRANGEIRO", "Agrupamento de Escolas do Fundão", entre outras.
Curso Hab. Anterior	nominal*	Descrição: nome do curso graduado da habilitação anterior Valores possíveis: nulo, nome do curso, exemplos: "História", "ESTRANGEIRO", "Ciências da Cultura", entre outros.
País Ens. Secund.	nominal*	Descrição: país no qual o aluno se graduou do ensino secundário Valores possíveis: nulo, nome do país, exemplos: "Timor", "Portugal", "Cabo Verde", entre outros.
Ano conclusao hab ant.	numérico*	Descrição: ano de graduação da habilitação anterior Valores possíveis: nulo, valor inteiro positivo.
NIVEL	nominal*	Descrição: Nível de risco gerado pela Digitalis na query SQL de extração da informação. Valores possíveis: “-” (ausência de risco), “Nível 1”, “Nível 2”, “Nível 3”, “Nível 4”

*- pode conter valores nulos/vazios

Uma análise crítica da **Tabela 5**, sob a perspetiva de treino de modelos de ML, revela rapidamente a presença de diversos atributos que não são úteis para o seu treino. No entanto, apesar de não serem diretamente úteis para o treino, esses atributos desempenham um papel importante na criação de novas classes ou na identificação dos alunos, quando utilizados como dados a serem preditos.

Adicionalmente, é crucial destacar que os dados fornecidos aos modelos de ML na aplicação web devem manter esta estrutura, uma vez que a aplicação será responsável pelo pré-processamento. O pré-processamento dos *datasets* em formato original será detalhado na próxima subcapítulo.

Por fim, é importante destacar a existência de atributos que possuem valores apenas para casos específicos, como o atributo NT_INGRESSO, que se aplica exclusivamente aos alunos que ingressaram no ensino superior no IPCB através do Concurso Nacional de Acesso (CNA). Embora atributos como este contenham informações relevantes e potencialmente preditivas, as suas limitações podem impedir que sejam considerados no treino dos modelos, uma vez que se procura abranger todos os alunos, incluindo aqueles sem nota de ingresso.

A seguir, será apresentado todo o processo de pré-processamento e preparação dos *datasets* para treino e/ou predição. Adicionalmente, serão discutidas as restrições mencionadas e as ações tomadas para cada um dos casos.

5.3. Pré-processamento

Considerando que a maioria dos dados extraídos são nominais, especificamente 26 atributos, é necessário codificá-los (transformá-los) em um formato numérico para que os algoritmos de ML da biblioteca *Scikit-Learn* os possam utilizar [128], [129]. Além dessa transformação, é importante aplicar outros procedimentos, como a remoção de certos atributos que não são benéficos para o treino, conforme mencionado anteriormente. Assim, o processo de pré-processamento abrange todas as alterações ou transformações aplicadas ao *dataset* original, incluindo: remoção de atributos desnecessários ou irrelevantes; transformação de atributos nominais em numéricos; normalização de valores numéricos; entre outras. Embora este seja um processo relativamente simples, envolve várias etapas, o que levou à criação de um fluxograma (**Figura 26**) que ilustra todo o procedimento.

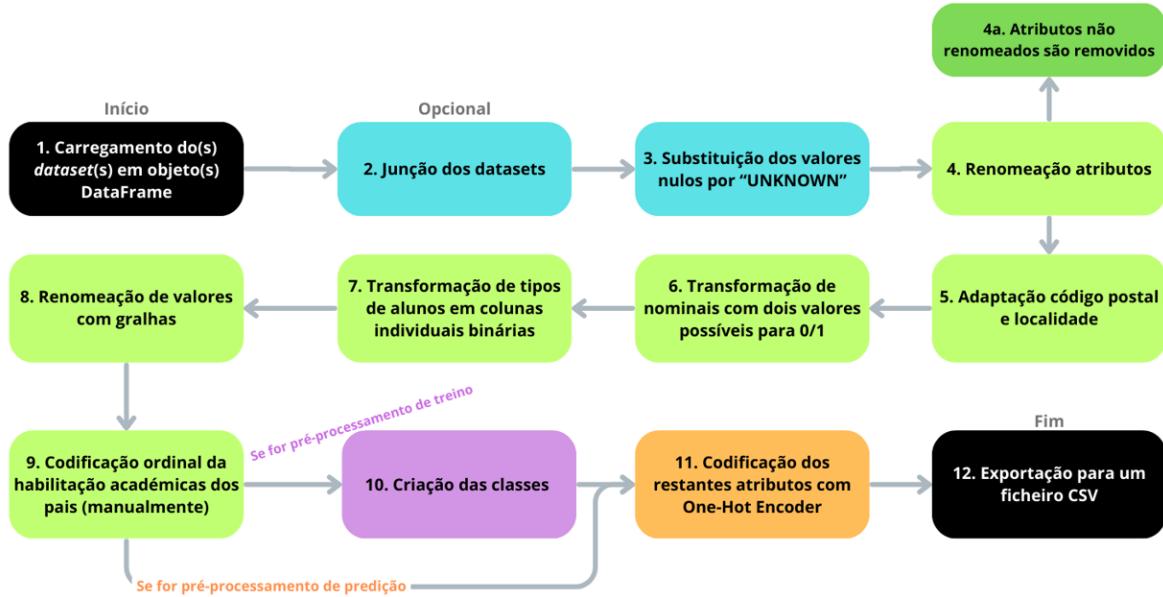


Figura 26 - Composição do processo de pré-processamento aplicado aos datasets

O processo de pré-processamento inicia-se com o carregamento de um ou vários *datasets* (ficheiros) em objetos *DataFrame* da biblioteca pandas (passo 1 da **Figura 26**). Caso sejam carregados vários os *datasets* a serem pré-processados, os mesmos serão combinados (passo 2 da **Figura 26**). Esta combinação é feita de forma incremental e consiste na realização de uma cópia do primeiro *dataset* e a essa, a adição do conteúdo dos restantes *datasets*.

Devido ao facto de existirem 13 atributos com possibilidade de conter valores nulos **Tabela 5**, é necessário tratar as suas ocorrências. Uma vez que existem casos específicos, como é o caso do atributo NT_INGRESSO, que só possui valor para um grupo restrito de instâncias (alunos que ingressaram pelo CNA), optou-se pela substituição dos valores nulos pelo valor "Desconhecido/a", passo 3 da **Figura 26**. Esta abordagem fundamenta-se principalmente na preservação de todas as instâncias, o que permite a utilização do mesmo *dataset* pré-processado em diferentes treinos de modelos de ML com distintos atributos.

No entanto, é importante destacar que as instâncias com o valor "Desconhecido/a" podem ser removidas antes do treino do modelo preditivo e por um processo posterior ao pré-processamento. Quando realizada a sua remoção, esta é devidamente mencionada no treino dos modelos de ML ou na apresentação dos seus resultados de treino.

Posteriormente, dado que os nomes originais dos atributos possuem uma nomenclatura incomum para manipulação, pois, incluem espaços, caracteres especiais, entre outros, é realizada uma renomeação dos mesmos (passo 4 da **Figura 26**). Esta renomeação é realizada com base num mapa armazenado num ficheiro *JSON*. O mapa utilizado neste trabalho está ilustrado na **Figura 27**. Adicionalmente, todos os atributos que não são renomeados por este passo serão removidos (passo 4a da **Figura**

26), permitindo a eliminação dos atributos não uteis para treino ou predição, conforme já mencionado anteriormente.

```
{  
    "DS_INSTITUIC": "school",  
    "NM_CURSO": "course",  
    "IDADE": "age",  
    "NACIONALIDADE": "nationality",  
    "Total ECTS Matriculados": "ects_enrolled",  
    "ECTS creditados": "ects_credited",  
    "Total ECTS aprovados": "ects_approved",  
    "ECTS reprovados": "ects_failed",  
    "ECTS realizados": "ects_completed",  
    "INGRESSO": "admission",  
    "PROSSEGUIU": "continued_studies",  
    "SEXO": "sex",  
    "NATURALIDADE": "place_of_birth",  
    "CD_POSTAL": "postal_code",  
    "TIPOS_ALUNO": "student_types",  
    "REGIME": "study_regime",  
    "DS_HABILIT_PAII": "academic_qualification_father",  
    "DS_HABILIT_MAEI": "academic_qualification_mother",  
    "SIT_PROF_PAII": "professional_situation_father",  
    "SIT_PROF_MAEI": "professional_situation_mother",  
    "GRUPO_PROF_PAII": "professional_group_father",  
    "GRUPO_PROF_MAEI": "professional_group_mother",  
    "NT_INGRESSO": "entry_grade",  
    "Habilitacao Anterior": "previous_qualification",  
    "Inst. Hab. Anterior": "previous_institution",  
    "Curso Hab. Anterior": "previous_course",  
    "Pais Ens. Secund.": "high_school_country",  
    "ano conclusao hab ant.": "year_previous_qualification"  
}
```

Figura 27 - Mapa de renomeação de colunas em formato JSON

Em seguida, é realizada uma adaptação dos dados referentes à localidade atual dos alunos (passo 5 da **Figura 26**). Nos dados originais a localidade é decomposta por três atributos (COD_POSTAL, CD_SUBPOS e LOCALIDADE). Sendo que a utilização destes três atributos referentes à localidade torna grande parte das instâncias únicas e uma vez que valores únicos são geralmente maus para treino de modelos de ML, foi optado pela adaptação em distrito. Esta transformação remove os três atributos referentes à localidade e adiciona um novo atributo, sendo ele representante do distrito da localidade. Esta transformação por sua vez permite agrupar em grupos maiores as instâncias do *dataset*, quando agrupados pelo distrito.

De forma a possibilitar a adaptação e transformação dos atributos da localidade no atributo referente ao distrito, foi necessário recorrer ao uso dos dados de códigos postais presentes no repositório de *GitHub* do autor João Antunes [126]. Contudo, a sua formatação não é indicada para esta transformação. Posto isto, recorreu-se a uma solução que formata os dados de João Antunes num formato próprio para este processo de pré-processamento. Adicionalmente, devido a que a solução apresentada em [126] é de código aberto, a solução desenvolvida neste projeto foi também publicada num repositório de código aberto, podendo ser consultada em [127].

O passo seguinte, passo 6 da **Figura 26**, tem como objetivo transformar todos os atributos nominais com apenas dois valores possíveis em atributos binários. Esta transformação consiste em associar o valor 0 a um dos valores e o valor 1 ao outro. No contexto deste trabalho, apenas dois atributos sofreram esta transformação: o atributo "sexo", em que o valor 1 foi atribuído ao valor "F" e o valor 0 ao "M" e adicionalmente foi renomeado para "e_feminino"; e o atributo referente à continuação de estudos, onde o valor 1 foi associado ao "S" e o valor 0 ao "N".

Após a transformação dos atributos para formato binário, procede-se à codificação do atributo TIPOS_ALUNOS (passo 7 da **Figura 26**). Este atributo possui múltiplos valores possíveis para a mesma instância, separados por vírgulas, o que levou a ser necessário utilizar o codificador *One-Hot Encoder*. Este algoritmo de codificação cria novas colunas, em que cada uma representa um dos valores possíveis do atributo original. Subsequentemente à criação das novas colunas, este processo de codificação altera os valores presentes no TIPOS_ALUNOS de cada instância para 1, na respetiva coluna, enquanto os valores não presentes, restantes colunas, são colocados a 0. Desta forma, é possível identificar todos os tipos de alunos associados a cada instância, resolvendo o problema de múltiplos valores no mesmo atributo, uma vez que os algoritmos de ML não suportam múltiplos valores num único atributo sem codificação prévia.

No passo 8 da **Figura 26**, são corrigidas diversas gralhas presentes nos dados. À medida foram inspecionados os *datasets* gerados nas diferentes versões, verificou-se a existência de diferentes gralhas, nomeadamente na existência de instâncias com denominações diferentes para o mesmo país, por exemplo Guiné-Bissau. Esta normalização dos valores foi também realizada através do uso de mapas *JSON*, como é ilustrado na **Figura 28**.

```
{
  "Guiné Bissau": "Guiné-Bissau",
  "Guine Bissau": "Guiné-Bissau",
  "Bissau": "Guiné-Bissau"
}
```

Figura 28 - Mapa para normalizar instâncias com valores da Guiné-Bissau

Como mencionado, existem diversas gralhas que necessitam de ser resolvidas ao normalizar os valores ou substituir pelo correto. Neste sentido, foi necessário normalizar os valores de dois cursos cujo seu nome foi alterado numa restruturação recente, ver **Figura 29**. Adicionalmente, procedeu-se à substituição dos valores nulos ("Desconhecido/a") nos atributos relativos à qualificação académica e grupo profissional do pai e da mãe. Neste caso específico, dado que existe um valor representativo para situações de desconhecimento, os valores nulos foram substituídos pelos termos corretos do seu contexto, "Outra" ou "Desconhecido/não tem", que por sua vez são apresentados na **Figura 30**.

```
{
    "Licenciatura em Tecnologias da Informação e Multimédia": "Licenciatura em Informática e Multimédia",
    "Licenciatura em Gestão Turística": "Licenciatura em Turismo"
}
```

Figura 29 - Normalização dos valores para o atributo referente ao curso inscrito

Qualificações	Situação profissional
<pre>{ "- Outra -": "Outra", "UNKNOWN": "Outra" }</pre>	<pre> 7 { 8 "UNKNOWN": "Desconhecido/não tem" 9 }</pre>

Figura 30 - Mapas de normalização para os valores dos atributos: qualificação académica e grupo profissional do pai e da mãe

Em seguida, após a correção das gralhas, procede-se à codificação ordinal dos atributos que representam a qualificação académica do pai e da mãe, passo 9 da **Figura 26**. A codificação ordinal (*Ordinal Encoding* [128]) é uma técnica utilizada para codificar atributos nominais cujos valores podem ser ordenados de uma forma lógica. Um exemplo comum do uso desta técnica, é a possibilidade de atribuir uma ordem a valores representativos de temperatura, como "quente", "ameno" e "frio" que possuem uma ordem natural, enquanto o mesmo não é possível no contexto de cores (à exceção de problemas específicos), como "vermelho", "azul" e "amarelo" [128].

Esta codificação não só discrimina os diferentes valores, mas também permite com que os algoritmos de ML reconheçam e diferenciem instâncias com base nesses valores ordenados. Considerando que as qualificações académicas já possuem níveis hierárquicos naturais [129], é possível ordenar os seus valores de uma forma lógica. A aplicação da codificação ordinal neste caso específico, recorreu também à utilização de um mapa em formato *JSON* (ver **Figura 31**), que associa os valores das qualificações aos respetivos níveis.

Contudo, é importante salientar a existência de valores desconhecidos nestes dois atributos, qualificação académica do pai e da mãe. Dado que os valores desconhecidos não possuem qualquer ordem lógica, optou-se por associar-lhes o valor 0, o qual é inferior à qualificação mais baixa (1- Não sabe ler nem escrever, ver **Figura 31**). Esta escolha fundamenta-se no caso de que os valores mais elevados estão reservados para as qualificações de nível superior, como é o caso de Doutoramentos, Mestrados e Licenciaturas.

```
{
    "- Outra -": 0,
    "Outra": 0,
    "UNKNOWN": 0,
    "Não sabe ler nem escrever": 1,
    "Sabe ler sem possuir a 4º classe": 2,
    "Ensino básico 1.º ciclo (4º classe)": 3,
    "Ensino básico 2.º ciclo (6º ano)": 4,
    "Ensino básico 3.º ciclo (9º ano)": 5,
    "Ensino Médio (11º ano)": 6,
    "12º ano de escolaridade": 7,
    "CET": 8,
    "CTeSP": 9,
    "Bacharelato": 10,
    "Licenciatura (Pré-Bolonha)": 11,
    "Licenciatura (Bolonha)": 12,
    "Pós-Graduação": 13,
    "Mestrado (Pré-Bolonha)": 14,
    "Mestrado (Bolonha)": 15,
    "Doutoramento": 16
}
```

Figura 31 - Mapa de codificação ordinal para a qualificação do pai e mãe

O próximo passo do processo de pré-processamento depende do contexto em que este está a ser realizado. No caso de se tratar de um pré-processamento para o treino de modelos de ML, é necessário computar e adicionar novas colunas representativas das classes a serem preditas (passo 10 da **Figura 26**). Por outro lado, quando o pré-processamento é efetuado com o objetivo de ser feita a predição das instâncias, o passo 10 é ignorado, uma vez que, além de os atributos necessários para a computação das classes não estarem presentes nesses dados, não se pretende adicionar classes nesse contexto.

Este passo, 10 da **Figura 26**, suporta a adição de várias classes simultaneamente. Contudo, é importante destacar que o treino só é realizado para ser predito uma única classe. Optou-se por permitir a adição de diversas classes no mesmo *dataset* de forma a reutilizar o mesmo, o que evita a criação de ficheiros diferentes cujo a única diferença entre si é uma coluna (classe). No entanto, é fundamental eliminar as colunas que representam as classes não preditas, uma vez que estas afetam negativamente o desempenho do modelo, apesar de os resultados serem quase perfeitos. Isto acontece, porque as colunas já fornecem a resposta durante o treino (apesar de o valor a predizer ser diferente) e adicionalmente essas colunas (atributos) nunca estarão disponíveis no momento de predição, sendo ele o ato da inscrição/matricula.

Adicionalmente, salienta-se que é devidamente feita a remoção das classes não preditas antes do treino. Uma vez que é esperado que já esteja interiorizado que um treino de modelos de ML só é feito com uma classe. No contexto deste trabalho houve todo o cuidado e preocupação de remover as classes não preditas.

Adicionalmente, ressalta-se que a remoção dos atributos representativos das classes não preditas é realizada antes do treino dos modelos de ML. Este cuidado é fundamental, uma vez que, no treino deste tipo de modelos, é esperado que seja feito unicamente com uma classe. Sublinha-se que no contexto deste trabalho, houve uma atenção especial em assegurar que as classes não preditas foram devidamente removidas, garantindo a integridade do processo de treino.

Além disso, destaca-se que diversos atributos já são considerados como classes, como é o caso de: “Total de ECTS Aprovados”, “Total de ECTS Realizados”, “Total de ECTS Reprovados” e “PROSSEGUIU”. Estes atributos no seu formato original já estão devidamente preparados a serem utilizados no treino, não necessitando qualquer transformação adicional.

No entanto, no contexto deste trabalho, foram adicionadas duas novas classes que requerem computação. Para melhor organização do relatório, a apresentação e a descrição detalhada de todas as classes utilizadas no treino dos modelos de ML são abordadas na seção seguinte, **5.3.1. Classes**.

O penúltimo passo, 11 da **Figura 26**, é realizado para ambos os casos anteriormente mencionados (treino versus predição). Este passo foca-se na codificação de todos os atributos nominais, que não foram eliminados pelo passo 4a (renomeação) e ainda não estão numa representação numérica.

Numa primeira solução, que é aquela proposta por este trabalho, optou-se pelo uso de um único algoritmo de codificação, o *One-Hot Encoder*. Esta escolha fundamentou-se sobretudo no tempo e resultados da escolha do melhor codificador associado a cada um dos atributos. Uma vez que toda essa procura requer todo um estudo mais aprofundado de cada atributo e da realização de inúmeros resultados, que por sua vez é adequado para um trabalho independente devido, novamente, ao tempo e complexidade necessários. Contudo, para este passo foram testadas três técnicas de codificação: *Ordinal Encoding*, *Binary Encoding* e *One-Hot Encoding*.

A técnica de codificação *Ordinal Encoding*, como já mencionado, é geralmente utilizada para atributos com uma ordem natural ou lógica, no entanto pode também ser aplicada a atributos sem ordem. Esta técnica atribui um número a cada valor possível do atributo e caso não seja indicada nenhuma ordem, baseia-se na ocorrência do valor.

O algoritmo *Binary Encoding* inicia com um processo idêntico ao de *Ordinal Encoding*, mas finaliza com uma abordagem diferente. Assim, primeiramente, realiza a codificação dos valores atributo em números e em seguida, converte esses números em binário. Após a conversão para binário, são criadas tantas colunas quantas forem necessárias para representar o maior número binário resultante [130]. Por fim, a cada coluna, uma vez que representa um bit da *string* do valor binário, irá-lhe ser associado o valor dessa mesma posição (bit) [130]. De forma a levar a uma melhor compreensão deste algoritmo, foi criada a **Figura 32** que apresenta todos os passos realizados por este.

Colunas	Cor	Cor	Cor	Cor_0	Cor_1	Cor_2
Valores	Vermelho	0	000	0	0	0
	Verde	1	001	0	0	1
	Azul	2	010	0	1	0
	Azul	2	010	0	1	0
	Codificação númerica Também conhecida por Ordinal Encoding	3	011	0	1	1
	Amarelo	4	100	1	0	0
	Rosa					

Figura 32 - Exemplificação do funcionamento do algoritmo Binary Encoding

Já o funcionamento da técnica *One-Hot Encoding*, foi mencionado anteriormente no passo 7 (da **Figura 26**) que aplica o algoritmo ao atributo TIPOS_ALUNOS. Relembrando que o algoritmo desta técnica cria novas colunas com valor binário (0 ou 1) representativas de todos os valores possíveis.

Para comparar a performance esperada dos modelos de ML utilizando diferentes técnicas de codificação, foram realizados treinos e avaliações extensivos com 11 algoritmos de ML selecionados para este trabalho, aplicados a um único *dataset* composto por todos os anos letivos (2019 a 2023), utilizando uma das três codificações mencionadas para os atributos nominais ainda não codificados. Estes testes revelaram que nenhum codificador se destacou consideravelmente, já que, em todos os testes realizados, as métricas de avaliação dos modelos de ML apresentaram valores semelhantes. Considerando que a apresentação de todos os resultados seria impraticável e aumentaria desnecessariamente a dimensão do relatório sem acrescentar valor de investigação, optou-se por apenas apresentar os resultados de um dos testes experimentais.

O teste experimental a ser apresentado envolveu o uso de 11 algoritmos de ML diferentes: Árvore de Decisão (DT), *Random Forest* (RF), *Naïve Bayes* (NB), *XGBoost*, *XGBoost* com *Random Forest* (*XGBoostRF*), *LightGBM*, *Gradient Boosting* (GB), k Vizinhos Mais Próximos (KNN), Regressão Logística (LoR), *Support Vector Machine* (SVM) e Rede Neural Perceptrão de Multicamadas (NN). O treino e a avaliação foram realizados com a técnica de treino Validação Cruzada de tamanho 10 (*KFolds* = 10). Todos os algoritmos foram avaliados utilizando as métricas: Exatidão, média de *Recall* (sem ponderação), média de Precisão (sem ponderação) e média de *F1* (com e sem ponderação). Desta forma, com a codificação *Ordinal Encoding* obteve-se os resultados da **Figura 33**, *One Hot Encoding* os da **Figura 34** e por fim a codificação *Binary Encoding* obteve os resultados da **Figura 35**. Adicionalmente, em cada figura, destacado a azul, está o melhor resultado da média não ponderada da métrica *F1*, que é uma das melhores métricas de avaliação para problemas de classificação de multiclasse.

	test_accuracy	test_balanced_accuracy	test_recall_macro	test_precision_macro	test_f1_macro	test_f1_weighted
DT	0.628008	0.425759	0.425759	0.418986	0.421281	0.631174
RF	0.727386	0.458780	0.458780	0.533036	0.466632	0.691055
NB	0.627178	0.412262	0.412262	0.394171	0.396799	0.622496
XGBoost	0.718880	0.466952	0.466952	0.514573	0.477074	0.692911
XGBoostRF	0.730913	0.449750	0.449750	0.534279	0.452683	0.686989
LightGBM	0.721577	0.464739	0.464739	0.522327	0.475195	0.693057
GRAD	0.731120	0.464225	0.464225	0.525109	0.470099	0.697687
KNN	0.698548	0.418846	0.418846	0.470101	0.430170	0.667045
LoR	0.683610	0.387184	0.387184	0.480914	0.366997	0.622127
SVM	0.682158	0.355500	0.355500	0.262897	0.286939	0.585614
NN	0.700415	0.415916	0.415916	0.469243	0.400073	0.650159

Figura 33 - Resultados experimentais do uso da codificação *Ordinal Encoding* nos atributos nominais do dataset

	test_accuracy	test_balanced_accuracy	test_recall_macro	test_precision_macro	test_f1_macro	test_f1_weighted
DT	0.647303	0.435282	0.435282	0.428789	0.430526	0.646436
RF	0.730290	0.465029	0.465029	0.555235	0.477319	0.694172
NB	0.680083	0.472123	0.472123	0.463146	0.460412	0.669307
XGBoost	0.722822	0.472321	0.472321	0.539527	0.484842	0.696715
XGBoostRF	0.731535	0.452055	0.452055	0.563110	0.454812	0.687670
LightGBM	0.726556	0.476049	0.476049	0.531504	0.486732	0.699470
GRAD	0.730498	0.464366	0.464366	0.531895	0.469693	0.698517
KNN	0.710166	0.441093	0.441093	0.498863	0.454866	0.680742
LoR	0.708921	0.418262	0.418262	0.461550	0.400833	0.659019
SVM	0.663278	0.287161	0.287161	0.278257	0.263249	0.553918
NN	0.657884	0.435867	0.435867	0.452054	0.441026	0.652961

Figura 34 - Resultados experimentais do uso da codificação *One Hot Encoding* aos atributos nominais do dataset

	test_accuracy	test_balanced_accuracy	test_recall_macro	test_precision_macro	test_f1_macro	test_f1_weighted
DT	0.610373	0.411738	0.411738	0.407488	0.408652	0.617166
RF	0.733610	0.465451	0.465451	0.556118	0.478461	0.696145
NB	0.687759	0.460902	0.460902	0.464432	0.451151	0.667640
XGBoost	0.715145	0.459749	0.459749	0.510642	0.469730	0.687130
XGBoostRF	0.733195	0.453198	0.453198	0.551206	0.456615	0.688508
LightGBM	0.726971	0.473862	0.473862	0.534015	0.485524	0.697856
GRAD	0.732780	0.470398	0.470398	0.539935	0.475075	0.698569
KNN	0.708091	0.433211	0.433211	0.496995	0.448756	0.677222
LoR	0.705809	0.411981	0.411981	0.476087	0.394731	0.653335
SVM	0.634025	0.240320	0.240320	0.294498	0.217065	0.512575
NN	0.702697	0.454748	0.454748	0.484543	0.459564	0.681144

Figura 35 - Resultados experimentais do uso da codificação *Binary Encoding* aos atributos nominais do dataset

Com a observação dos três resultados não é possível identificar de forma clara a melhor codificação a utilizar deixando, assim, a critério do autor a escolha da codificação. No contexto deste projeto, uma vez que um dos objetivos futuros é a extração de regras dos modelos de ML, de forma a verificar quais são os critérios mais importantes para a identificação de alunos em risco, optou-se pela escolha do uso único da codificação *One Hot Encoding*.

Embora esse algoritmo de codificação adicione um número considerável de novos atributos, é possível reduzir a quantidade de colunas utilizando a opção “infrequente” da biblioteca [131]. Esta opção permite definir um valor mínimo de ocorrências que um determinado valor do atributo precisa de ter para que uma coluna correspondente ao mesmo seja criada. Além disso, o uso da codificação *One Hot Encoding* facilita a

interpretação das regras que podem ser extraídas dos modelos de ML, já que cria uma coluna para cada valor específico, evitando comparações complexas como “curso = 5” (no caso de *Ordinal Encoding*) ou “curso_0 = 1 && curso_1 = 0” (no caso de *Binary Encoding*). Em vez disso, permite a criação de regras mais claras, como “curso_Licenciatura_Engenharia_Informática = 1 && curso_Licenciatura_Informatica_Multimedia = 0”. Além disso, outros resultados, como aqueles presentes em [130] e dois estudos do estado da arte ([75], [76]) demonstraram que, apesar do aumento no número de colunas, essa codificação oferece um bom desempenho.

Posto isto, ao ser aplicada a última transformação aos atributos nominais, o *dataset* encontra-se pronto a ser utilizado. Desta forma, optou-se por guardar o resultado de todo o processo de pré-processamento em formato *CSV*. Isto permite que os resultados de pré-processamento possam ser devidamente carregados nos scripts de *Python* de treino ou utilizados na aplicação desenvolvida neste trabalho.

Interessa referir que ao longo do desenvolvimento deste projeto, foram criados diversos pré-processamentos diferentes na sequência de várias tentativas de otimizar o desempenho dos modelos de ML. No entanto, devido a que a apresentação dos resultados de treino e avaliações de modelos de ML para cada pré-processamento demandaria muito tempo, optou-se pela apresentação dose resultados obtidos em apenas dois pré-processamentos específicos.

O primeiro pré-processamento combina e transforma todos os *datasets* fornecidos, abrangendo todos os anos letivos disponíveis. Já o segundo pré-processamento envolve os *datasets* dos anos letivos anteriores a 2022/2023, enquanto o *dataset* referente ao último ano letivo disponível (2022/2023) é apenas pré-processado no momento da predição. Esta segunda abordagem visa simular e verificar como será o desempenho dos modelos de ML em anos letivos subsequentes.

5.3.1. Classes investigadas

Todos os problemas que implementam soluções com base em técnicas de ML necessitam que seja definido qual o valor a predizer, denominado por classe. Relembra-se que todos os problemas de classificação, como já mencionado, procuram predizer uma classe para uma determinada instância, ou seja, um valor representante de um grupo conhecido de um conjunto de valores limitados cuja representação pode ser numérica ou nominal. Já os problemas de regressão procuram prever (calcular) um valor numérico para determinada característica. No entanto, ao contrário da classificação, o valor previsto é apenas numérico e continuo. Um exemplo simples para a classificação é a predição da raça de um animal com determinadas características, enquanto a regressão consegue prever o custo de uma casa dadas determinadas características da mesma.

Adicionalmente, em problemas de ML, é comum criar e associar novas classes às instâncias, uma vez que os dados extraídos, geralmente, não incluem a indicação explícita do que se pretende predizer. Um exemplo simples é o caso de treino de modelos de ML para emails de spam, onde os dados disponíveis consistem no conteúdo dos emails, mas apenas uma pessoa consegue identificar se um email é ou não spam. Neste mesmo problema, é necessário criar uma nova coluna (no *dataset*) e para cada instância associar-lhe o valor correspondente da nova classe, exemplo: "É spam?" que pode ser "Sim" ou "Não", permitindo que o modelo de ML aprenda a fazer essa distinção.

No âmbito deste trabalho, com base nos requisitos estabelecidos pela professora orientadora Ana Paula Silva e objetivos do projeto REVUP, ao todo foram identificadas e investigadas seis classes. Destas classes, apenas uma corresponde a um problema de regressão, enquanto as restantes cinco dizem respeito a problemas de classificação. A classe associada ao problema de regressão é já um atributo existente, "Total ECTS Realizados", que foi renomeado para "ects_realizados". Em relação às classes de classificação, quatro das cinco foram criadas com base em valores de outros atributos, enquanto a quinta é também um atributo pré-existente nos *datasets*, "PROSSEGUIU", tendo sido renomeado para "continua_estudos".

Como as classes "ects_realizados" e "continua_estudos" já existem como atributos no *dataset*, não houve necessidade de modificá-las, podendo ser utilizadas diretamente. No entanto, para as restantes quatro classes investigadas, foi necessário desenvolver pequenos algoritmos que criassem uma nova coluna no *dataset*. Estes algoritmos atribuem, para cada instância, um valor correspondente à nova classe que por sua vez é calculado com base em valores de outros atributos do *dataset*.

A lógica por de trás de cada um dos algoritmos desenvolvidos foi fundamentada nos objetivos do projeto REVUP. A professora orientadora, Ana Paula Silva, foi responsável pela definição inicial dos algoritmos que depois foram discutidos e validados em sessões de *brainstorming* com o professor coorientador Arlindo Silva e o autor deste trabalho. Dos quatro algoritmos definidos, três atribuem um nível de risco aos alunos do IPCB (referente ao abandono e insucesso escolar), enquanto o quarto algoritmo adapta a predição do atributo "ects_realizados" de regressão para classificação ao dividi-lo em três intervalos diferentes.

A atribuição de diferentes níveis de risco visa identificar os alunos com maiores dificuldades e maior probabilidade de desistirem do ensino no IPCB. A divisão em vários níveis permite adotar abordagens diferenciadas para cada nível de risco, garantindo que os alunos com maiores dificuldades recebam um apoio mais atento e cuidadoso comparativamente aos alunos sem risco. No entanto, é crucial destacar que a realização de intervenções de ajuda não deve depender exclusivamente da predição dos modelos de ML. Sendo fundamental que os docentes do IPCB validem e complementem a predição, assegurando uma intervenção correta, mais precisa e eficaz.

De forma a serem apresentados os diferentes algoritmos de criação de classes aplicados neste trabalho, foram criadas quatro figuras diferentes:

- **Figura 36:** apresenta o pseudocódigo e possíveis valores do primeiro algoritmo de associação de risco definido, denominado por “Risco Original”;
- **Figura 37:** apresenta o pseudocódigo e possíveis valores de um melhoramento do primeiro algoritmo, denominando-se de “Risco Otimizado”;
- **Figura 38:** apresenta o pseudocódigo e possíveis valores do algoritmo de atribuição de risco binário (com e sem risco), por sua vez denominado por “Risco Binário”;
- **Figura 39:** apresenta o pseudocódigo e possíveis valores do algoritmo de adaptação do ECTS realizados em três intervalos.

Algoritmo Risco Original	Nível	Valor associado
le(todos_alunos) de um ficheiro	0	Sem risco
Enquanto le(aluno) de todos_alunos faz:	1	Baixo risco
ects_realizados = aluno["ects_realizados"]	2	Médio risco
prossegue = aluno["continua_estudos"]	3	Alto risco
	4	Não comparece
Se ects_realizados == 0 && prossegue == Falso então:		
aluno["risco_original"] = 4		
Senão Se ects_realizados == 0 && prossegue == Verdadeiro então:		
aluno["risco_original"] = 3		
Senão Se ects_realizados < 40 então:		
Se prossegue == Falso então:		
aluno["risco_original"] = 2		
Senão:		
aluno["risco_original"] = 1		
Senão:		
aluno["risco_original"] = 0		
Fim do algoritmo		

Figura 36 - Pseudocódigo do Risco Original e os valores possíveis

Algoritmo Risco Otimizado	Nível	Valor associado
le(todos_alunos) de um ficheiro	0	Sem risco
Enquanto le(aluno) de todos_alunos faz:	1	Baixo risco
ects_realizados = aluno["ects_realizados"]	2	Médio risco
prossegue = aluno["continua_estudos"]	3	Alto risco
Se ects_realizados < 5 então:		
aluno["risco_optimizado"] = 3		
Senão Se ects_realizados < 40 então:		
Se prossegue == Falso:		
aluno["risco_optimizado"] = 2		
Senão:		
aluno["risco_optimizado"] = 1		
Senão:		
aluno["risco_optimizado"] = 0		
Fim do algoritmo		

Figura 37 - Pseudocódigo do Risco Otimizado e os valores possíveis

Nível	Valor associado
0	Sem risco
1	Com risco

```

Algoritmo Risco Binário
le(todos_alunos) de um ficheiro

Enquanto le(aluno) de todos_alunos faz:
    ects_realizados = aluno["ects_realizados"]
    prossegue        = aluno["continua_estudos"]

    Se prossegue == Falso || ects_realizados < 40 então:
        aluno["risco_binario"] = 1
    Senão:
        aluno["risco_binario"] = 0
Fim do algoritmo

```

Figura 38 - Pseudocódigo do Risco Binário e os valores possíveis

Nível	Valor associado
0	Aprova (>= 40 ECTS)
1	Reprova (5-39 ECTS)
2	0 ECTS

```

Algoritmo Intervalo ECTS Realizados
le(todos_alunos) de um ficheiro

Enquanto le(aluno) de todos_alunos faz:
    ects_realizados = aluno["ects_realizados"]
    Se ects_realizados < 5 então:
        aluno["intervalo_ects_realizados"] = 2
    Senão Se ects_realizados < 40 então:
        aluno["intervalo_ects_realizados"] = 1
    Senão:
        aluno["intervalo_ects_realizados"] = 0
Fim do algoritmo

```

Figura 39 - Pseudocódigo de Intervalo ECTS Realizados e os valores possíveis

Assim, foram acrescentadas quatro colunas adicionais ao *dataset* pré-processado, “risco_original” (ver **Figura 36**), “risco_otimizado” (**Figura 37**), “risco_binario” (ver **Figura 38**) e “intervalo_ects_realizados” (ver **Figura 39**). A inclusão destas colunas no mesmo *dataset* permite reutilizá-lo sem a necessidade de gerar ficheiros redundantes, onde todas as colunas, exceto a classe alvo, são idênticas. Contudo, para o treino dos modelos de ML, apenas é mantida a coluna da classe alvo, sendo que as restantes são removidas do objeto *DataFrame* a ser utilizado para o treino e teste do modelo. A criação destas quatro colunas e do uso dos algoritmos apresentados, é toda realizada recorrendo ao uso da biblioteca *Pandas* de *Python*. Todo este processo é executado no passo 10 do pré-processamento (ver **Figura 26**).

Adicionalmente, é relevante destacar que foram explorados outros algoritmos de associação de risco e uso de outros atributos como classe, como por exemplo o atributo “Total ECTS Reprovados”. No entanto, devido ao facto de que o número de resultados aumenta exponencialmente com o número de classes, assim como o tempo necessário para treinar, avaliar e otimizar os modelos de ML, neste trabalho são apenas apresentados e discutidos os resultados das seis classes mencionadas.

Por fim, embora todas as colunas estejam presentes no mesmo *dataset*, é importante salientar que cada modelo de ML só pode ser treinado para predizer uma única classe. Portanto, ao treinar, por exemplo, um modelo para predizer o intervalo dos ECTS Realizados (“intervalo_ects_realizados”), todas as outras classes devem ser removidas dos dados de treino. Essa observação, embora pareça óbvia, é por vezes esquecida, podendo levar a resultados extremamente bons à primeira vista. Porém,

apesar de um desempenho aparentemente perfeito, a predição é totalmente incorreta e ilusória, visto que os dados de treino contêm a resposta que o modelo deveria aprender. Esta observação é feita porque tal situação ocorreu durante o trabalho do autor, e será abordada com maior detalhe no capítulo **6. Treino e avaliação dos modelos de ML**.

5.4. Análise

Num primeiro trabalho desenvolvido, foi realizada uma breve visualização dos dados disponíveis para treino. Devido à grande variedade de atributos, esta visualização concentrou-se sobretudo nas distribuições de instâncias (alunos) para as classes “continua_estudos” e “risco_otimizado”. Além disto, é relevante realçar que toda a visualização dos dados foi possível através do uso das bibliotecas *Seaborn* e *Matplotlib* de *Python*.

Interessa ainda destacar que, em futuros desenvolvimentos e contribuições deste trabalho, é altamente recomendada a realização de uma análise mais aprofundada, com o objetivo de pré-selecionar e eliminar atributos de treino que apresentem variabilidade excessiva (atributos com valores únicos para cada instância) ou variabilidade quase inexistente (atributos com o mesmo valor para todas as instâncias). Uma análise detalhada, por sua vez, pode contribuir significativamente para a otimização do treino dos modelos de ML, possibilitando a descoberta de padrões ou informações relevantes.

Inicialmente, foi visualizado o número de instâncias por escola. Como não se pretende treinar um modelo de ML para cada escola do IPCB, torna-se essencial compreender a contribuição de instâncias de cada uma para o *dataset* de treino. Para esta análise preliminar, foram criados dois gráficos. Além disso, dado que o uso do nome completo das escolas comprometeria a legibilidade das figuras neste relatório, optou-se por substituir os nomes extensos pelas respetivas siglas, sendo as utilizadas:

- **ESA**: Escola Superior Agrária de Castelo Branco;
- **ESART**: Escola Superior de Artes Aplicadas de Castelo Branco;
- **ESE**: Escola Superior de Educação de Castelo Branco;
- **ESGIN**: Escola Superior de Gestão de Idanha-a-Nova;
- **ESALD**: Escola Superior de Saúde Dr. Lopes Dias;
- **EST**: Escola Superior de Tecnologia de Castelo Branco.

O primeiro gráfico, **Figura 40**, apresenta o número total de instâncias por escola dos quatro anos letivos disponíveis. O segundo, ilustrado na **Figura 41**, apresenta o número de instâncias por escola em cada ano letivo individualmente.

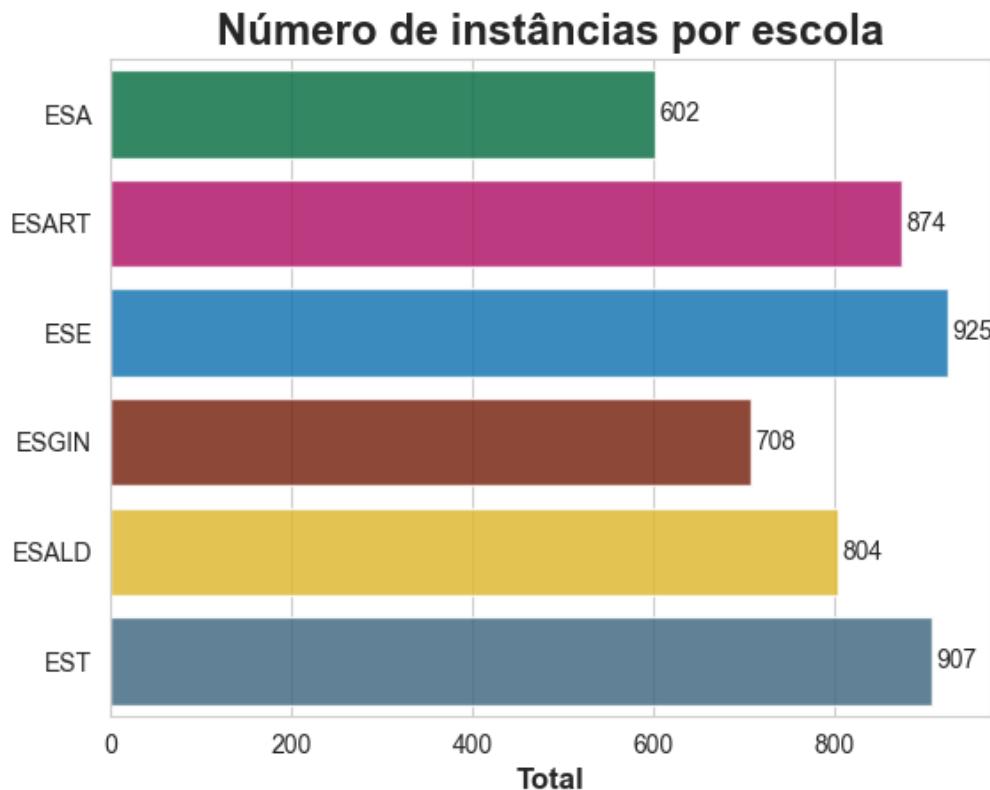


Figura 40 - Número de instâncias por escola do *dataset* combinado de todos os anos letivos

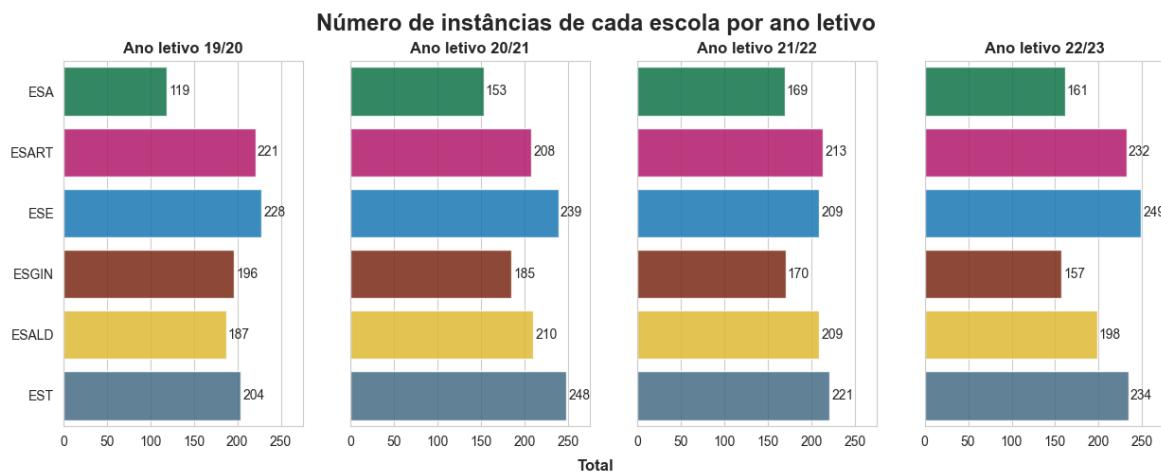


Figura 41 - Número de instâncias por escola em cada ano letivo

A observação do número de instâncias por escola permite concluir que entre os anos letivos 2019/2020 e 2022/2023 a escola que viu o menor número de alunos a se matricularem foi a ESA, com 602 alunos ao todo. Já o contrário, a EST, ESE e ESART são aquelas que possuem um número maior de alunos matriculados, superando os 800. Adicionalmente, esta visualização permite confirmar que não existe uma disparidade elevada no número de instâncias.

De seguida, foi visualizado o número de alunos que continuam a estudar no IPCB após o primeiro ano, classe “continua_estudos”. Para esta visualização foram criados quatro gráficos diferentes. O primeiro gráfico, **Figura 42**, apresenta no contexto geral

(quatro anos letivos) o número de alunos que continuaram a estudar e destes quantos desistiram ao fim do primeiro ano. Curiosamente, verifica-se que 25.9% (1247) dos alunos que se inscreveram pela primeira vez no IPCB desistem dos seus estudos. Esta observação reforça a necessidade de serem empregues soluções para combater o abandono escolar.

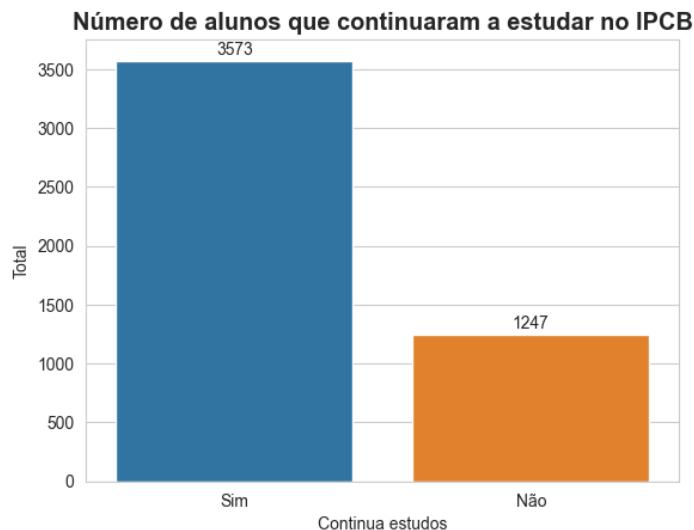


Figura 42 - Número de alunos que continuaram a estudar no IPCB para os quatro anos letivos em análise

Quando analisado o número de alunos desistentes por ano letivo, ilustrado na **Figura 43**, observa-se que o número de desistências se mantém relativamente constante ao longo dos anos. A observação destes dois gráficos, **Figura 42** e **Figura 43**, permite facilmente verificar um desequilíbrio significativo de representação das instâncias para a classe “continua_estudos”. Posto isto, caso os resultados de treino para esta classe fiquem aquém do esperado, deve-se considerar a realizar um balanceamento de instâncias. Esta observação é crucial, pois, provavelmente o modelo de ML terá facilidade em aprender os padrões dos alunos que continuam a estudar, mas enfrentará maior dificuldade em identificar os padrões dos alunos que desistem.

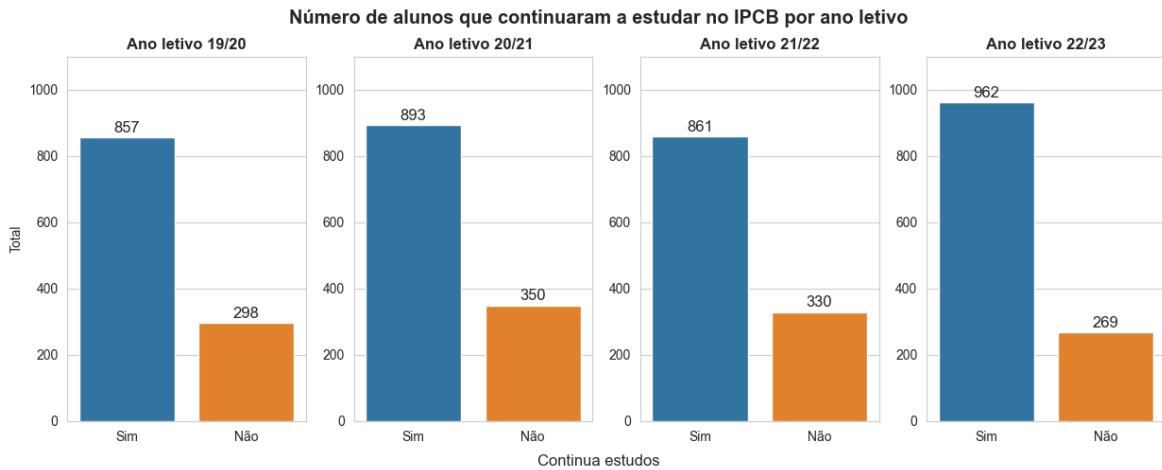


Figura 43 - Número de alunos que continuaram a estudar no IPCB em cada ano letivo

Adicionalmente, foi realizada uma visualização do número de alunos desistentes por escola, sendo essa ilustrada na **Figura 44**. A observação do gráfico revela a existência de três escolas, EST, ESGIN e ESA, com taxas de abandono superiores a 30% (39%, 33% e 32% respetivamente). Em contraste, a ESALD destaca-se pela positiva, pois, apresenta uma taxa de desistência significativamente baixa, igual a 9.5%. No entanto, é importante sublinhar que as razões por de trás destes valores são desconhecidas e, no âmbito deste trabalho, não foi procurada a resposta a esta observação.

Número de alunos que continuam estudos após 1º ano por escola

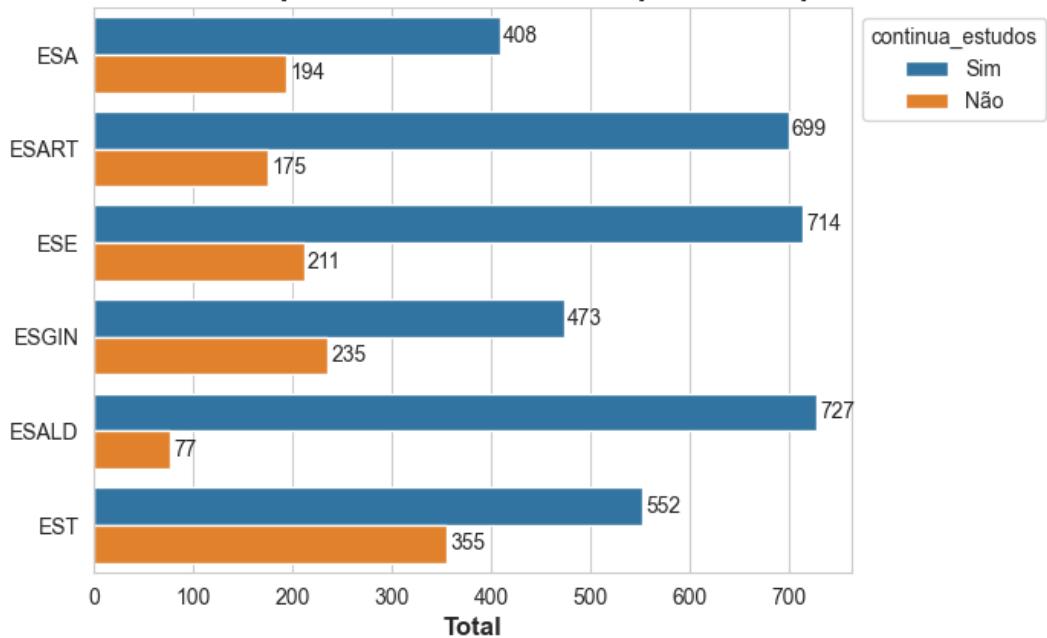


Figura 44 - Número de alunos que continuaram a estudar no IPCB por escola

Adicionalmente, foi analisado o número de alunos desistentes de cada curso de todos os anos letivos, sendo o seu gráfico ilustrado na **Figura 45**. A observação do gráfico permite verificar a existência de vários cursos que ao fim de quatro anos letivos,

tiverem mais alunos a desistir do que alunos a continuar. Esses cursos são: Licenciatura em Engenharia Civil; Licenciatura em Turismo; Licenciatura em Gestão Hoteleira; Licenciatura em Informática e Multimédia; Licenciatura em Engenharia Industrial.

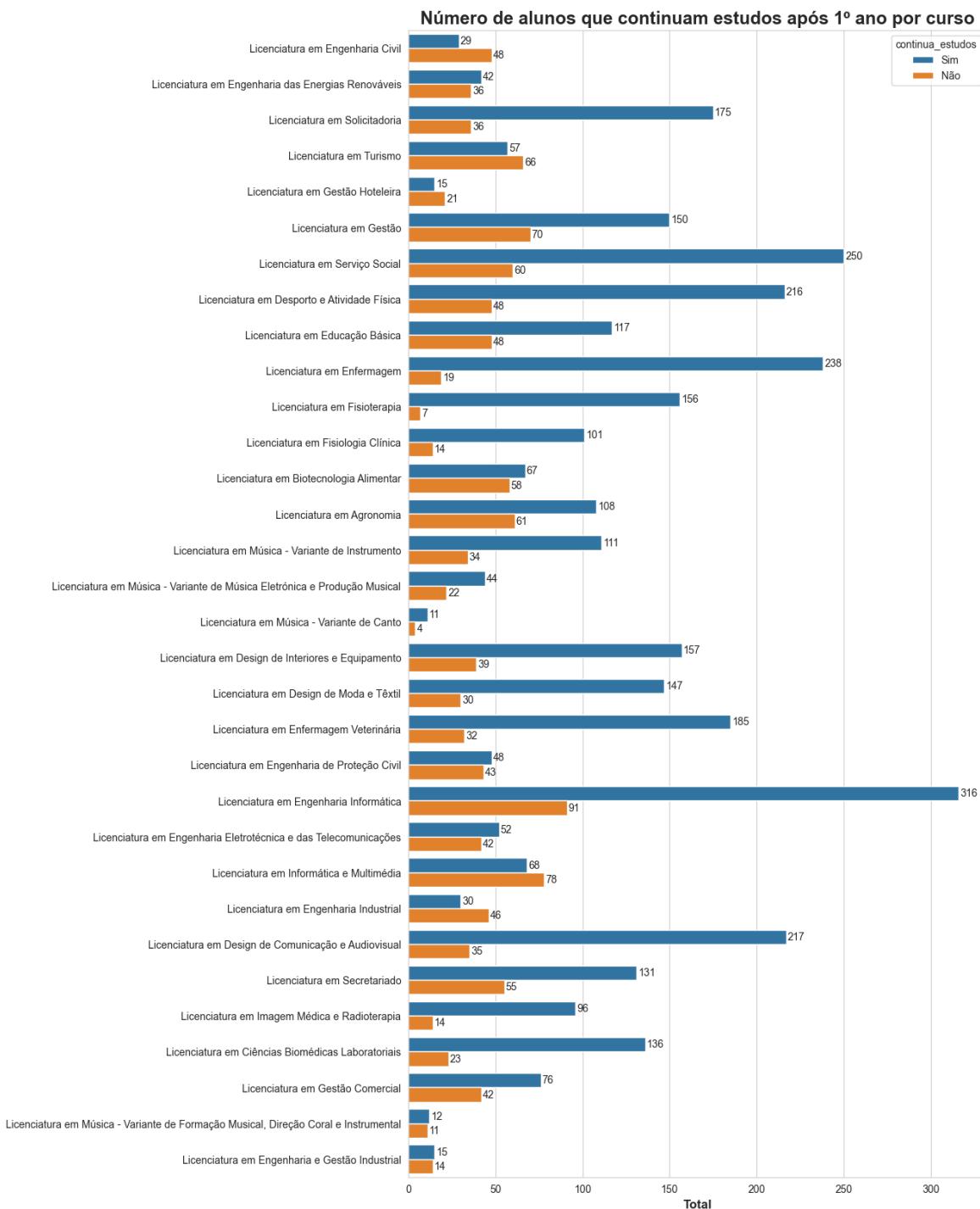


Figura 45 - Número de alunos, por curso, que continuaram a estudar após término do 1º ano

Uma outra análise realizada para a classe “continua_estudos”, foi a verificação de qual o tipo de ingresso que possui a maior taxa de abandono, assim, foi criado o gráfico apresentado na **Figura 46**. Os resultados observados são de certa forma surpreendentes, pois, verifica-se que cerca de 63% dos alunos internacionais acabam

por desistir do curso ao fim do primeiro ano. Esta observação pode ser justificada pelo facto de que os alunos vindos do estrangeiro têm uma maior dificuldade em integrar-se num novo país, dado que saem da sua zona de conforto e existe todo um processo moroso pelo qual têm de passar até conseguirem estar presentes nas escolas. A dificuldade de integração pode estar relacionada com o facto de virem sozinhos para um novo país, estarem longe da sua família e amigos o que acaba por afetar o seu dia a dia e motivação.

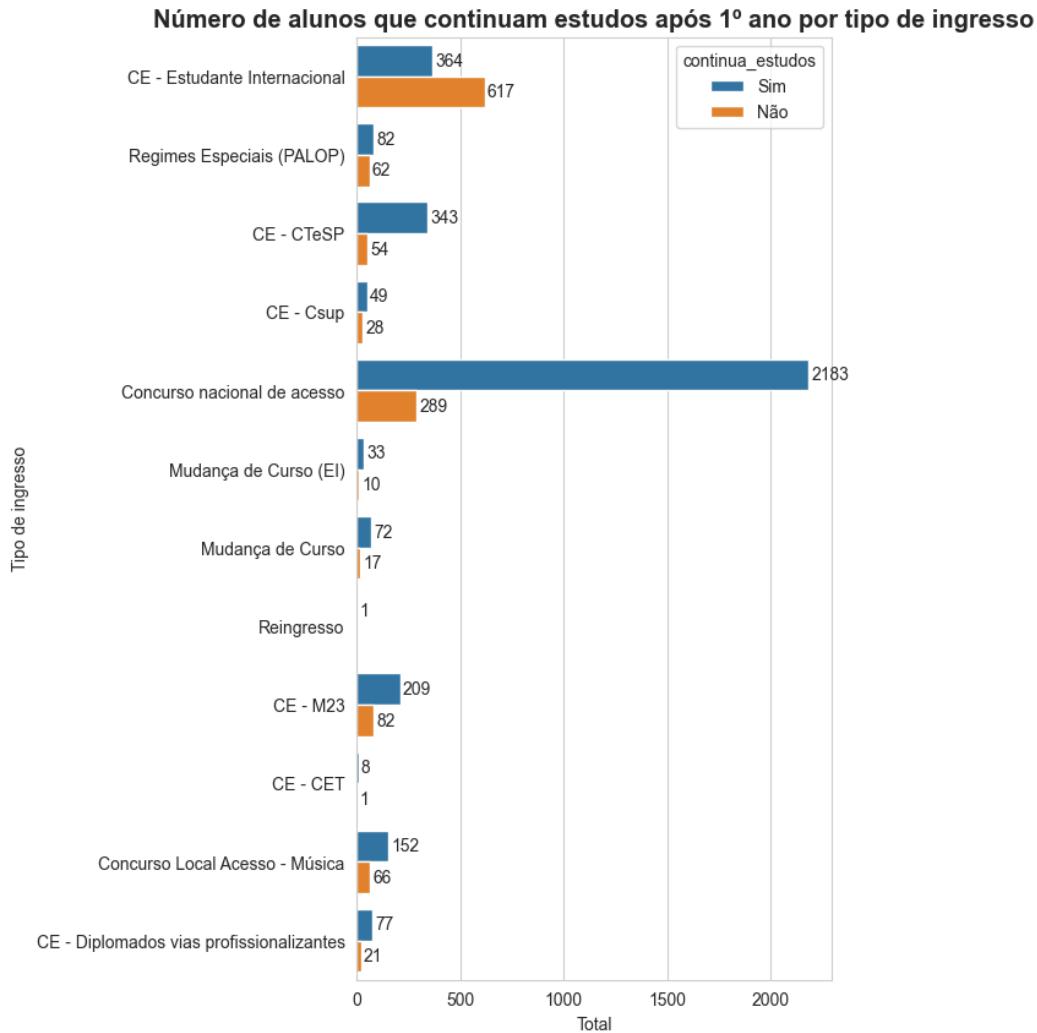


Figura 46 - Número de alunos por tipo de ingresso que continuam a estudar

Além disso, é possível verificar que existem três valores diferentes, “Mudança de Curso (EI)”, “Mudança de Curso” e “Reingresso”, que apesar de terem uma denominação diferente, representam a mesma situação. Assim, as instâncias destes três valores foram combinadas todas numa só, “Mudança de Curso”. Adicionalmente, dado que o valor “CE – CET” também possui um número muito reduzido de instâncias, estas foram incorporadas para o valor “CE – CteSP”, pois, também representa situação semelhante, dado que os cursos CET têm vindo a ser substituídos por CTeSTP.

Em última análise da classe “continua_estudos”, é possível prever que os modelos de ML terão a tendência de indicar que alunos internacionais irão abandonar ao fim do primeiro ano. Esta observação é perfeitamente normal, pois, como foi possível verificar no gráfico da **Figura 46**, as instâncias refletem esta mesma situação.

Relativamente à classe “risco_optimizado”, foram criadas visualizações semelhantes às da classe “continua_estudos”. Iniciando-se com a observação do número de instâncias por cada nível de risco no *dataset*, que agrupa os quatro anos letivos disponíveis. Assim, para esta primeira visualização foi criado o gráfico ilustrado na **Figura 47**. Ao analisar o gráfico, torna-se evidente que persiste um desequilíbrio de representação dos valores possíveis da classe. De todos os níveis possíveis, verifica-se que o risco “Médio”, representativo de alunos que reprovam e não continuam a estudar, é aquele com menor representação, o que por sua vez leva ao modelo de ML ter uma maior dificuldade em aprender os padrões das suas instâncias.

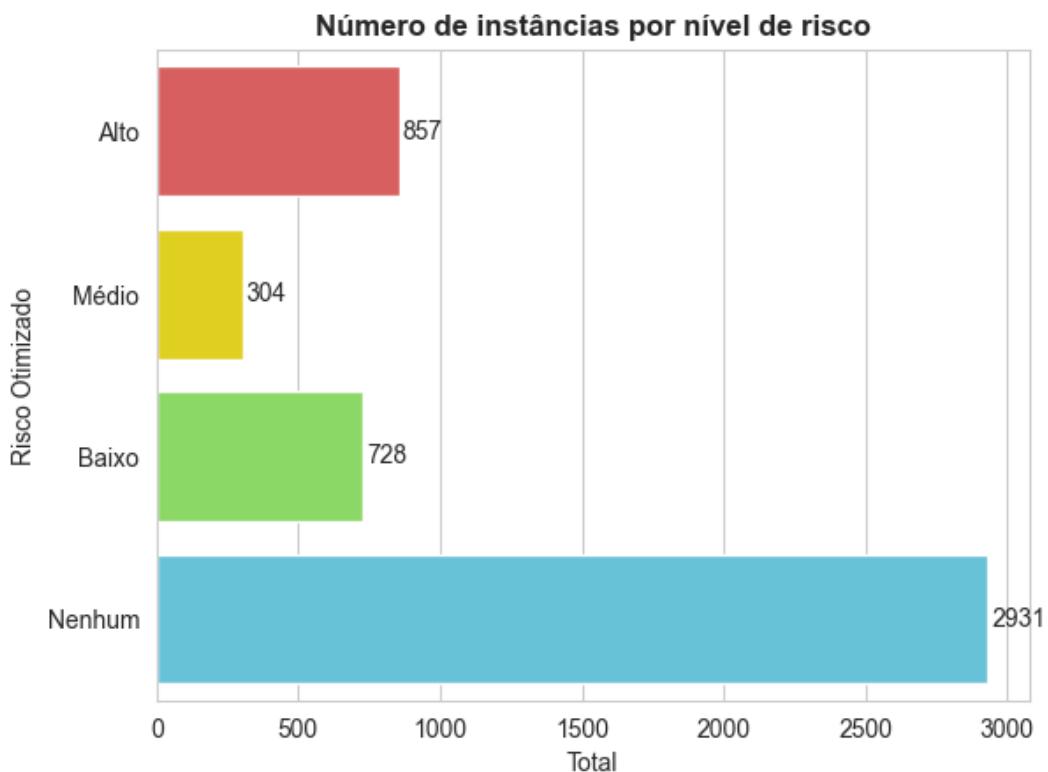


Figura 47 - Número de instâncias por nível de risco (Risco Otimizado)

No caso de serem visualizados os números de instâncias por nível de risco no contexto de cada letivo disponível, **Figura 48**, confirma-se novamente uma consistência ao longo dos anos.

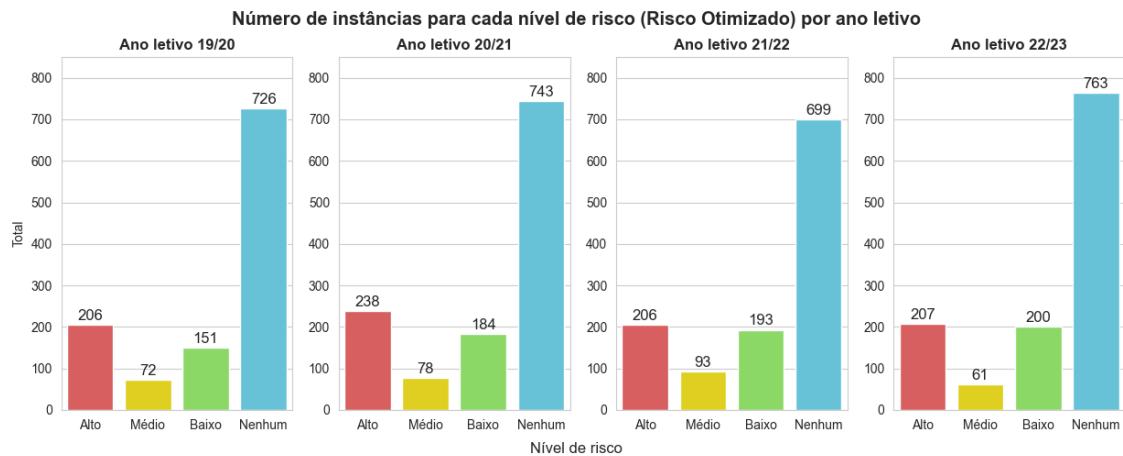


Figura 48 - Número de instâncias para cada nível de risco (Risco Otimizado) em cada ano letivo

Adicionalmente, foi analisada a distribuição de níveis de risco por escola, conforme ilustrado na **Figura 49**. Mais uma vez, verifica-se que a EST, ESGIN e ESA são as escolas mais preocupantes, pois, são aquelas que apresentam um maior número de alunos de risco. No entanto, é particularmente interessante denotar que, na EST, existem 289 alunos de risco a mais dos que não têm risco. Em contraste, a escola ESALD demonstra novamente um menor número de alunos de risco, em que apenas a 18% dos alunos é lhe associado um nível de risco.

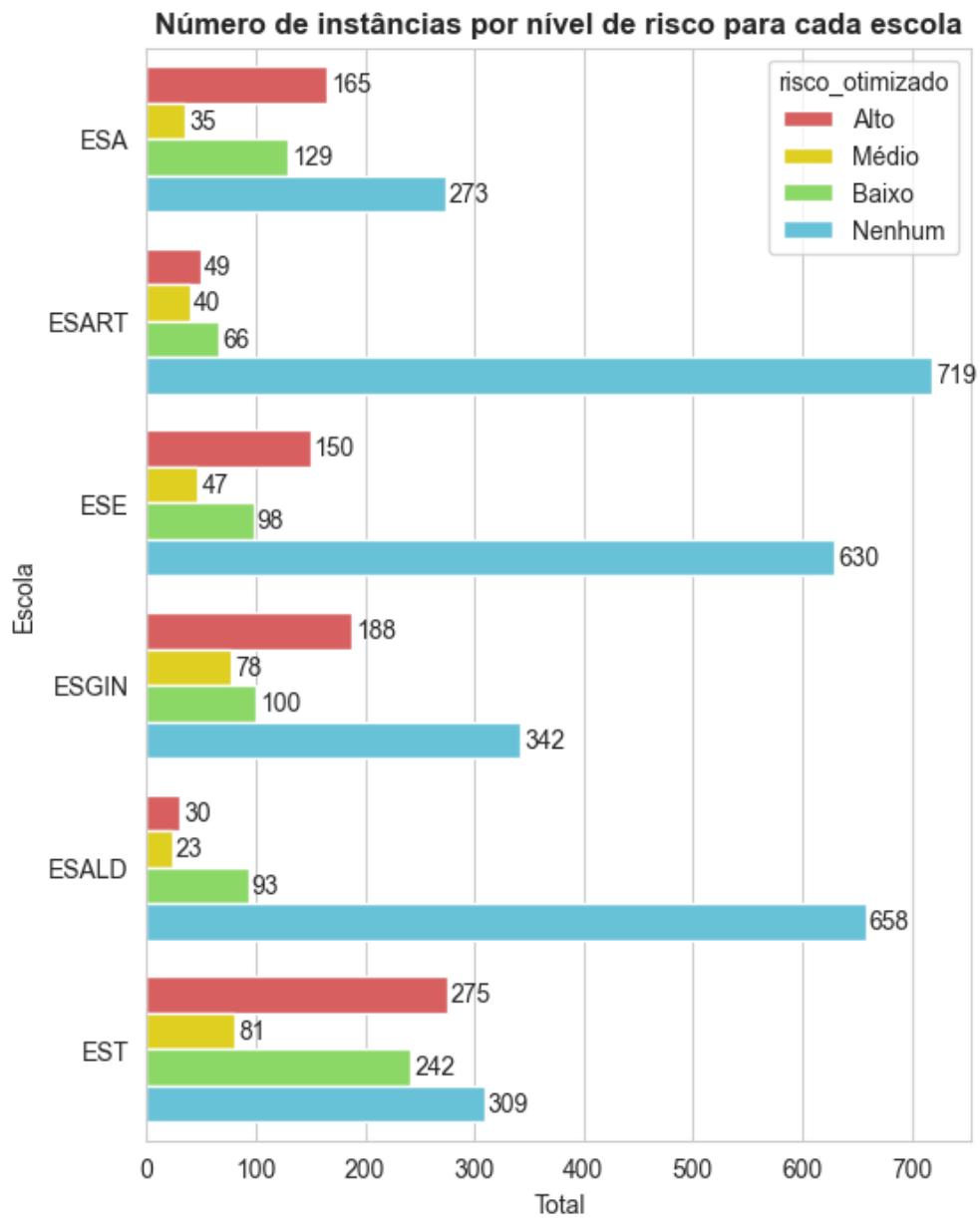


Figura 49 - Número de instâncias por nível de risco (Risco Otimizado) para cada escola do IPCB

Quando analisados os números de instâncias para os diferentes tipos de ingresso, conclui-se que quase todas as instâncias de nível de risco alto (82%, 706) provêm de alunos internacionais e de Países Africanos de Língua Oficial Portuguesa (PALOP), conforme ilustrado na **Figura 50**. Curiosamente, verifica-se que os alunos provenientes de CTesP (Cursos Técnicos Superiores Profissionais) apresentam um número considerável de instâncias de risco baixo que chega a superar o número de alunos sem risco. Adicionalmente, destaca-se que a criação do gráfico apresentado na **Figura 50** já possui as alterações dos valores dos tipos de ingresso semelhantes, cujo foram mencionados na análise da classe "continua_estudos", **Figura 49**.

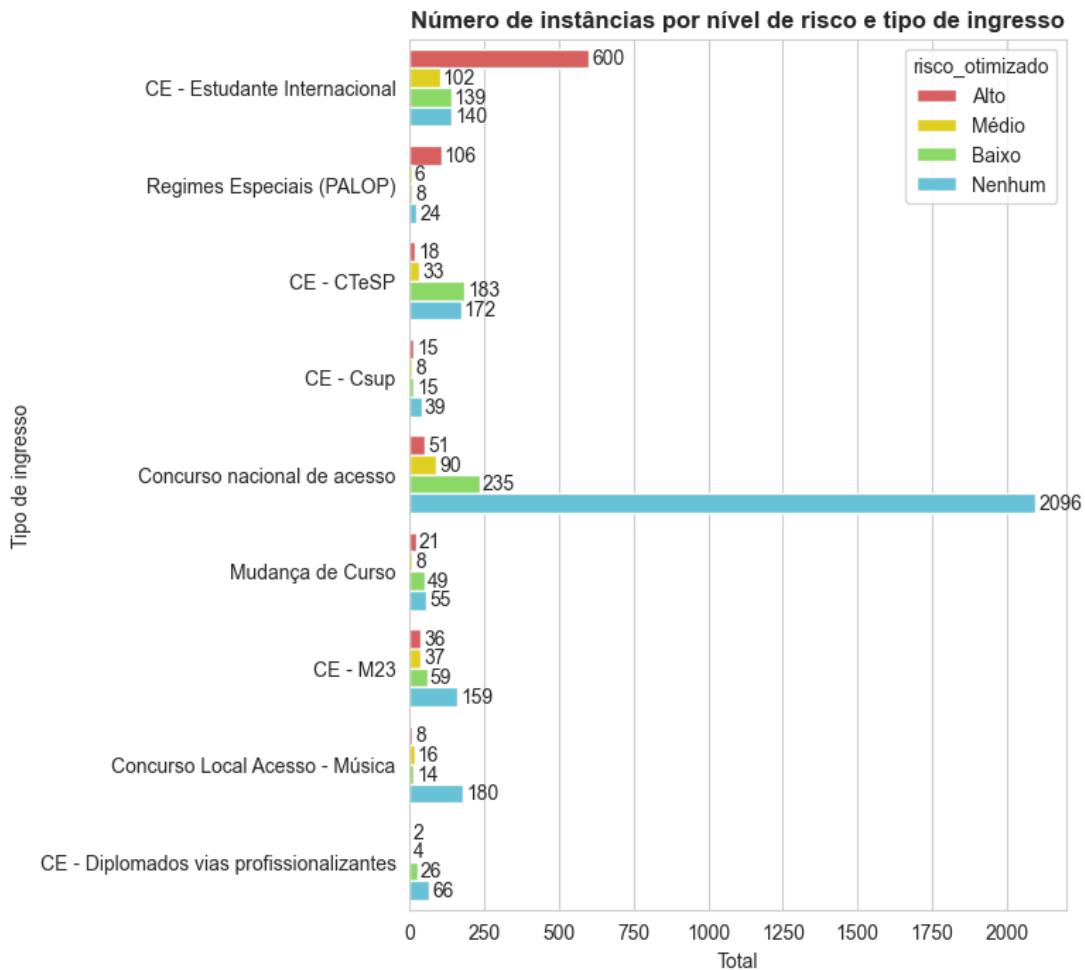


Figura 50 - Número de instâncias por nível de risco (Risco Oitmizado) e por tipo de ingresso

Interessa também destacar que também foram gerados gráficos de distribuição de instâncias para as três restantes classes de problemas de classificação (“risco_original”, “risco_binario” e “intervalo_ects_realizados”). No entanto, infelizmente, nenhuma dessas classes apresentou um equilíbrio de representação de instâncias para os seus possíveis valores melhor do que as classes já apresentadas.

Adicionalmente, dado que já se confirmou que as escolas com o maior número de alunos em risco são a EST, ESGIN e ESA, e em contraste a ESALD com o menor número, serão apenas apresentados dois gráficos ilustrativos da distribuição das instâncias para as restantes três classes. Para cada uma, o primeiro gráfico apresenta o número de instâncias por cada valor possível da classe e o segundo apresenta a distribuição das instâncias por tipo de ingresso. É também relevante mencionar que, em todas as classes, também se verificou que os alunos com maior risco ou com 0 ECTS realizados são, em grande parte, provenientes de países estrangeiros.

Assim, seguem-se seis figuras dos gráficos criados. As primeiras duas **Figura 51** e **Figura 52**, apresentam a distribuição de instâncias para a classe “risco_original”, a **Figura 53** e **Figura 54** para a classe “risco_binario” e por fim a **Figura 55** e **Figura 56** para “intervalo_ects_realizados”.

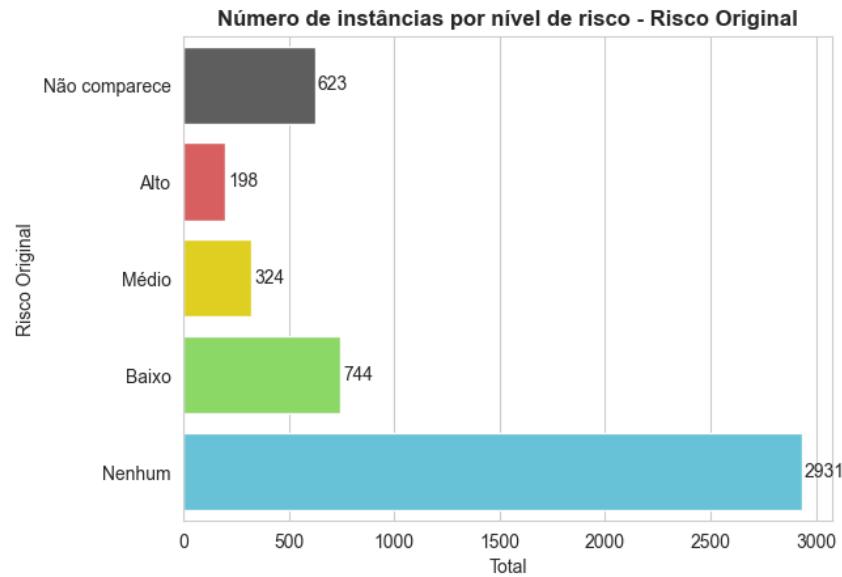


Figura 51 - Número de instâncias por nível de risco (Risco Original)

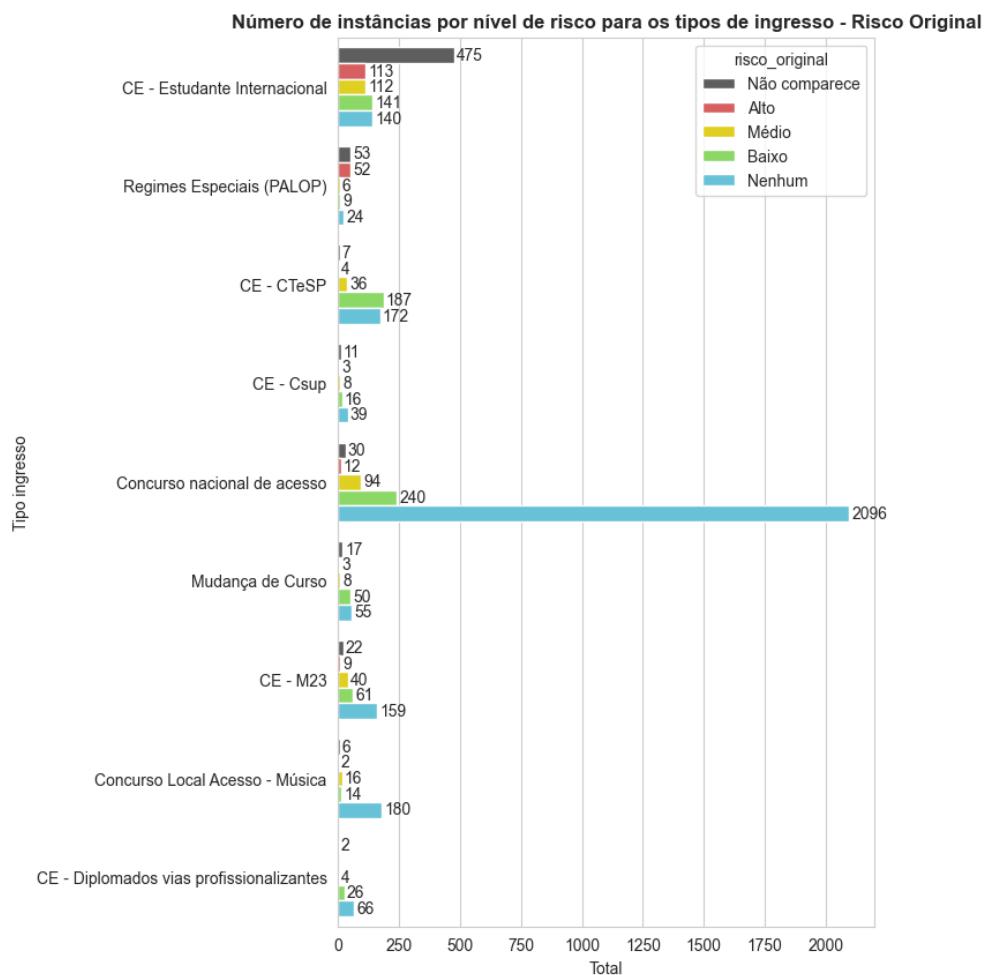


Figura 52 - Número de instâncias por nível de risco (Risco Original) para cada tipo de ingresso



Figura 53 - Número de instâncias com risco e sem risco (Risco Binário)

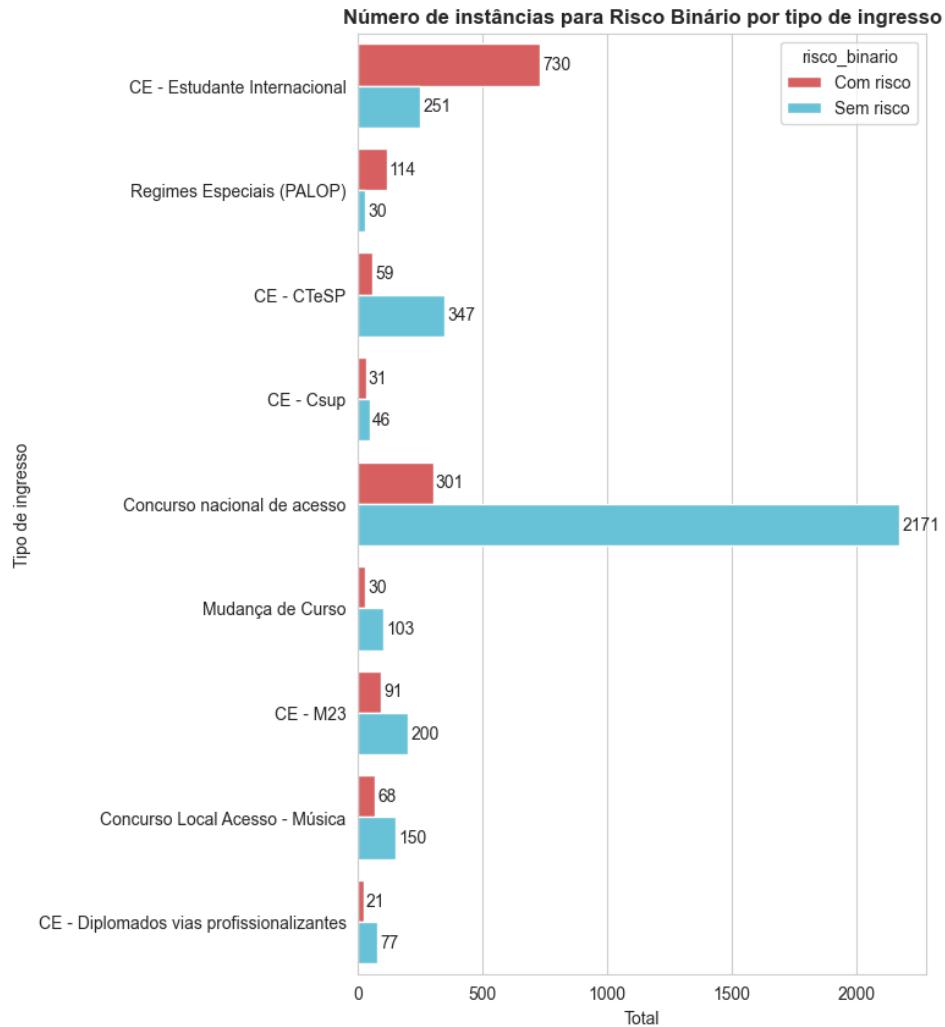


Figura 54 - Número de instâncias com risco e sem risco por tipo de ingresso

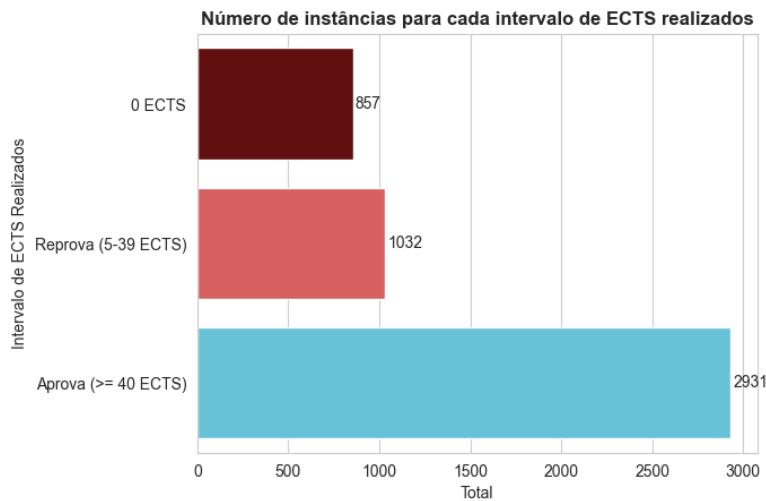


Figura 55 - Número de instâncias por intervalo de ECTS realizados

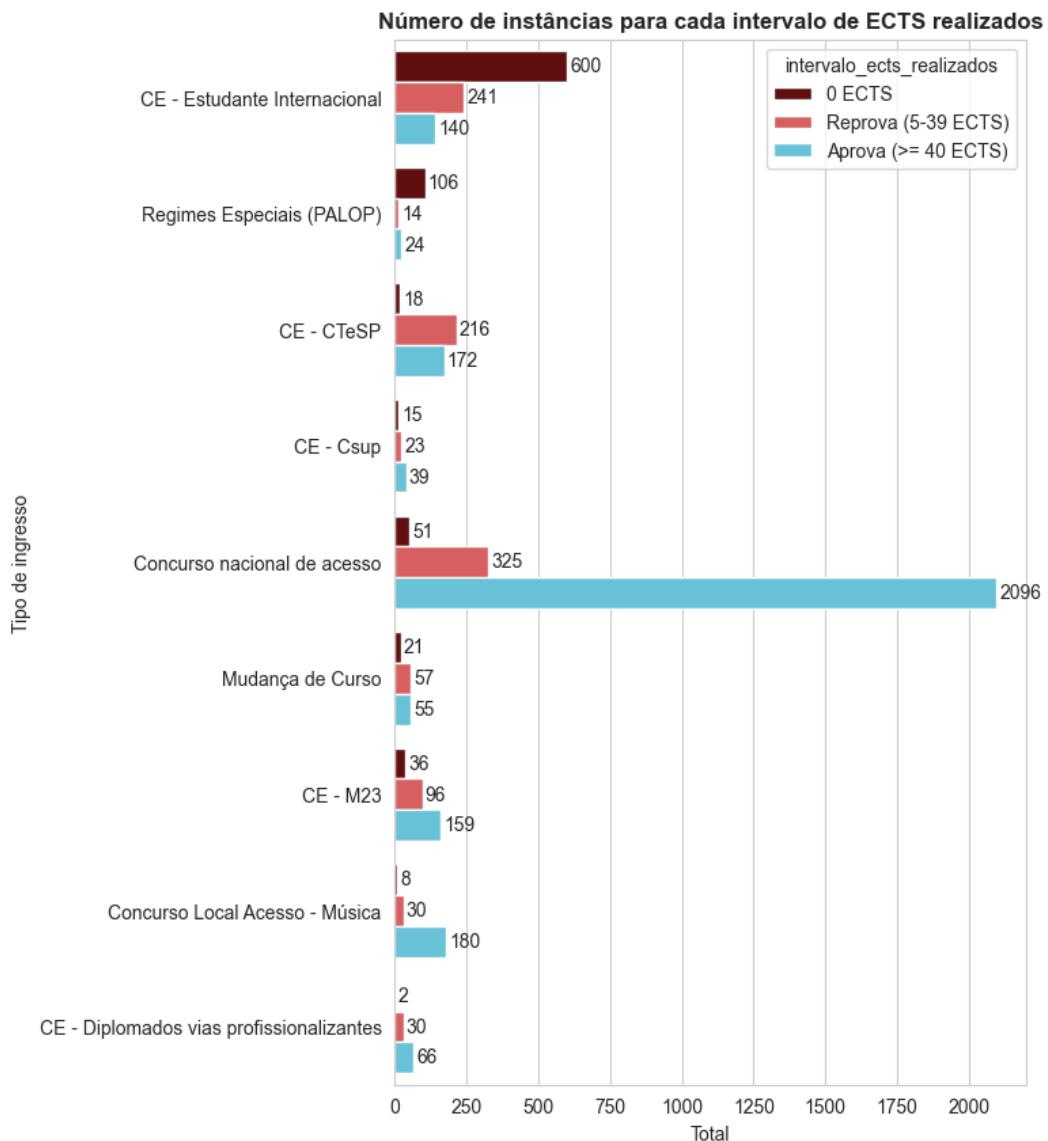


Figura 56 - Número de instâncias por intervalo de ECTS realizados para cada tipo

Com base na análise de dados realizada, é possível concluir que o IPCB, nomeadamente as escolas EST, ESGIN e ESA, poderá beneficiar significativamente da implementação de soluções focadas na prevenção do abandono e insucesso escolar. Isto porque verificou-se que cerca de 25% dos alunos que se matriculam pela primeira vez acabam por abandonar a instituição após o primeiro ano letivo. Perante este cenário, reforça-se a motivação para o uso de técnicas de ML, nomeadamente dos seus modelos preditivos, como uma ferramenta auxiliar na identificação de alunos em risco de abandono ou insucesso escolar.

Adicionalmente, interessa constar que grande parte dos alunos de risco identificados nos dados disponíveis de treino, provêm de ingressos internacionais e PALOP. Várias podem ser as razões que fazem com que esta situação ocorra, mas uma possível poderá ser o facto de os alunos vindos do estrangeiro possuírem uma dificuldade acrescida em integrar-se num país novo. Além disso, verificou-se que todas as classes apresentam um desequilíbrio significativo, e que por sua vez, pode apresentar-se como um obstáculo de treino dos modelos de ML.

6. Treino e avaliação dos modelos de ML

A próxima etapa do trabalho desenvolvido foca-se no treino e avaliação de modelos de ML. Modelos esses que foram treinados com os vários algoritmos de ML disponibilizados pelas bibliotecas de *Python*: *Scikit-Learn*, *XGBoost* e *LightGBM*. Nesta fase, é analisado o desempenho dos modelos de ML na predição das diferentes classes de predição apresentadas no capítulo **5.3.1. Classes investigadas**. Desta forma, procura-se investigar o desempenho do uso de modelos de ML como uma ferramenta auxiliar de combate ao abandono e insucesso escolar.

Devido ao elevado número de experiências que foram realizados para este trabalho, que produziram uma quantidade significativa de resultados, não é feita uma apresentação detalhada de todos eles. Contudo todos eles estão anexados a este trabalho (anexo **A. Resultados obtidos**). Além disso, como as primeiras versões dos *datasets* fornecidos ao autor, para os treinos experimentais, continham erros nos seus valores, os modelos de ML treinados com estes tinham um, por sua vez, um desempenho não refletor dos dados reais.

Desta forma, é inicialmente descrito o processo de treino e avaliação de modelos de ML aplicado neste trabalho. Posteriormente, de forma breve e concisa, são apresentados alguns dos resultados experimentais obtidos e os problemas encontrados durante a sua realização. Em seguida, é apresentado um processo que combina dois modelos de ML para realizar a predição do nível de risco dos alunos do IPCB. Neste penúltimo subcapítulo, são também apresentados os resultados de treino obtidos e uma tentativa de otimização de hiperparâmetros. O capítulo termina com uma breve reflexão dos resultados de treino obtidos no desenvolvimento deste trabalho.

6.1. Processo e sua evolução

Um processo de treino e avaliação de modelos de ML pode ser dividido em quatro passos fundamentais, sendo eles: carregar o *dataset*, pré-processar o *dataset*, treinar e avaliar modelo(s) de ML e visualizar os resultados de treino. Além disso, uma última etapa envolve a exportação de um modelo de ML treinado com o algoritmo de ML que apresentou o melhor desempenho. Só depois da realização e validação de todo este processo é que devem ser integrados os modelos de ML nas diferentes aplicações.

Num primeiro desenvolvimento do trabalho, todo este processo foi dividido em diferentes ficheiros de *Jupyter notebook* (“*.ipynb*”), dado que permitem uma execução separada de blocos de código e visualização do resultado da sua execução. Desta forma, seguiu-se uma implementação semelhante àquela apresentada na **Figura 57** para um primeiro trabalho experimental.



Figura 57 - Fluxo de funcionamento de um primeiro processo de treino de modelos de ML implementado

Porém, a primeira solução implementada revelou ter defeitos. Um dos defeitos significativos é a necessidade de criar múltiplas versões dos ficheiros para lidar com as diferentes evoluções dos dados, bem como manter um registo das implementações já desenvolvidas para permitir a comparação de resultados e desempenho. O defeito mais impactante relaciona-se com a falta da possibilidade de ser reutilizado o processo de pré-processamento. É essencial poder ser reproduzido o pré-processamento para que seja garantido o correto funcionamento do processo de predição a ser integrado nas diferentes aplicações.

Posto isto, foi necessário abordar os vários defeitos encontrados na primeira implementação. Assim, uma nova versão de todo o processo foi concebida e desenvolvida, sendo ela ilustrada na **Figura 58**. Esta nova versão resulta na criação de apenas três ficheiros: "*preprocess.py*", "*training.py*" e "*prediction.py*". Cada ficheiro é responsável por uma parte específica do processo, garantindo maior organização e permitindo armazenar configurações de pré-processamento para serem reutilizadas na etapa de treino ou predição.

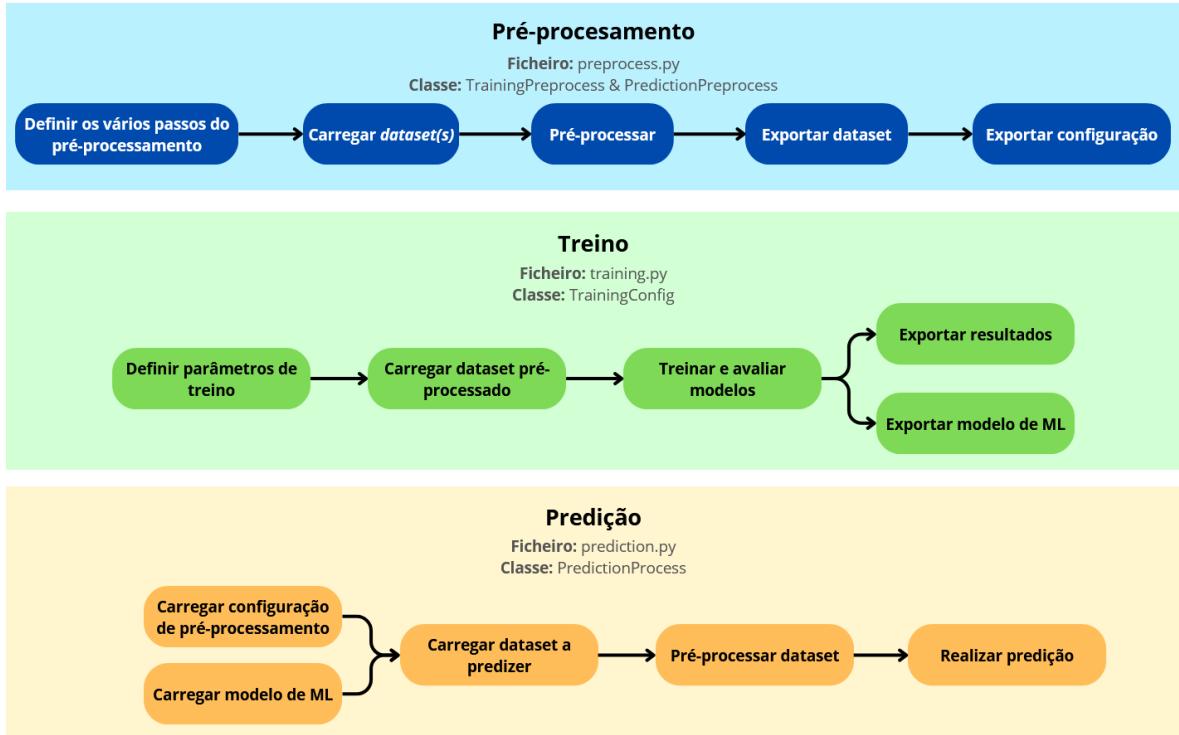


Figura 58 - Novo processo de treino, avaliação e predição

Uma breve descrição das classes criadas para as etapas de pré-processamento e predição é feita no capítulo **7.2. Aplicação para apoio a investigação em ML**, dado que todo este novo processo foi concebido paralelamente ao desenvolvimento da segunda aplicação. Relativamente à etapa de treino, nesta o programador poderá definir todo um processo de treino, conforme ilustrado na **Figura 59**. O processo de treino e avaliação de modelos de ML pode ser feito através do método *holdout* (divisão dos dados em duas partes, treino e teste) ou validação cruzada.

```

dropout_conf = TrainingConfig(df, target_col_name: "continua_estudos", is_cv=True, num_folds=10,
                               balance_strategy=BalanceStrategy.OVERSAMPLING_SMOTE,
                               technique_wrapper=TechniqueWrapper.ONE_VS_ALL,
                               cols_to_drop=["ects_realizados", "sf_aval_modelo", "risco_v1",
                                             "risco_v2", "risco_v3", "situacao_final"])

dropout_conf.run()
dropout_conf.export_results(to_excel: True, filename="Todos_CV_10")
dropout_conf.export_ml_model(algorithm_key: "rf", model_name: "random_forest_dropout", balance_strategy=BalanceStrategy.OVERSAMPLING_SMOTE)
  
```

Figura 59 - Exemplo de configuração de um treino de modelos de ML para a predição de continuação dos estudos por parte do aluno

Ao contrário da técnica *holdout*, a validação cruzada divide o *dataset* em *K folds* (*K* divisões), sendo o valor de *K* definido pelo programador. Esta técnica possui um total de iterações igual ao número de *folds*. Em que em cada iteração é treinado e avaliado um modelo de ML usando um algoritmo específico. Porém, em cada iteração, o *fold* utilizado para avaliar o modelo é diferente dos utilizados nas outras iterações, enquanto os restantes *folds* são utilizados para o treino. Com o objetivo de auxiliar a compreensão desta técnica, foi elaborada a **Figura 60** (adaptada de [132]) que ilustra todo um processo de treino usando a técnica de validação cruzada com *K folds* (com *K* igual a cinco).

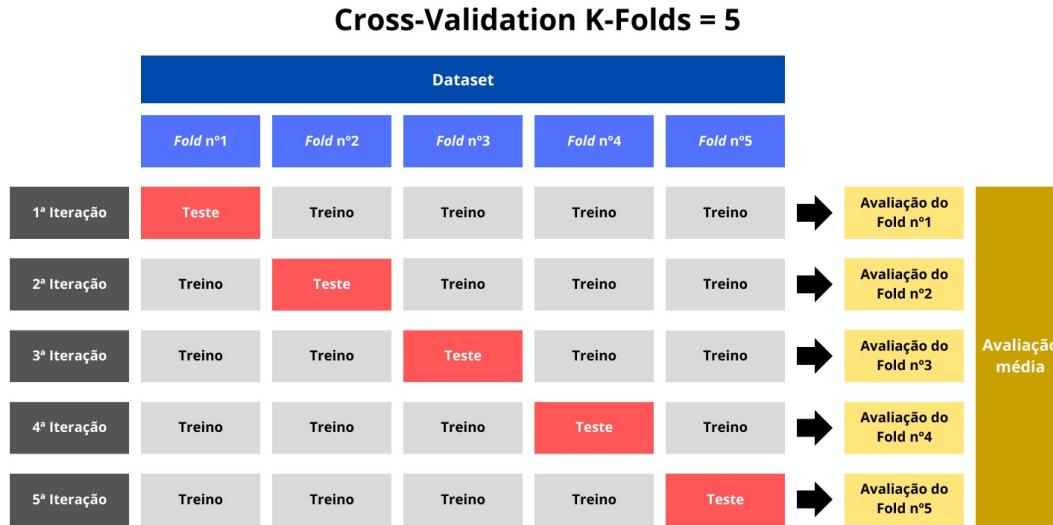


Figura 60 – Representação gráfica da técnica de treino validação cruzada (*Cross-Validation*) (adaptada de [136])

Interessa realçar que os *folds* não se sobrepõem [9] (ver **Figura 60**), possibilitando assim o uso de todas as instâncias do *dataset* tanto para treino como para teste. Para avaliar o desempenho de um algoritmo, é calculada a média das avaliações de cada modelo de ML que foram avaliados com a métrica escolhida pelo programador. Além disso, deve ser também calculado o desvio padrão com o objetivo de verificar se todos os modelos treinados são de certa forma semelhantes no seu desempenho. Posto isto, um dos objetivos da validação cruzada é a escolha do algoritmo que apresenta a melhor média da métrica escolhida e um menor desvio padrão. É importante realçar que a validação cruzada realizada neste trabalho utilizou um número K de *folds* igual a 10.

Para o treino de modelos de ML, foram utilizados 11 algoritmos de classificação e 10 para regressão (ECTS realizados). Interessa destacar que os 10 algoritmos utilizados para a regressão são os mesmos de classificação, no entanto diferenciam-se na sua implementação interna que é alterada para acomodar problemas de regressão. No entanto, um algoritmo, *Naïve Bayes*, não foi aplicado na regressão, uma vez que está disponível apenas para problemas de classificação. Desta forma, todos os algoritmos utilizados para o treino de modelos de ML estão presentes na **Tabela 6**. Cada algoritmo é também acompanhado pela respetiva sigla, sendo esta utilizada na apresentação dos resultados.

Tabela 6 - Algoritmos de ML utilizados para o treino de modelos e sua sigla

Algoritmo de ML	Utilizado para classificação?	Utilizado para regressão?	Sigla
Árvore de decisão	Sim	Sim	DT
Random Forest	Sim	Sim	RF
Gradient Boosting	Sim	Sim	GB
Naïve Bayes	Sim	Não	NB
XGBoost	Sim	Sim	XGB
XGBoost com Random Forest	Sim	Sim	XGBRF
LightGBM	Sim	Sim	LGBM
K Vizinhos mais próximos	Sim	Sim	KNN
Regressão Linear	Não	Sim	LR
Regressão Logística	Sim	Não	LoR
Support Vector Machine	Sim	Sim	SVM
Multi Layer Perceptron (rede neuronal)	Sim	Sim	NN_MLP

Adicionalmente, para o treino dos modelos de ML, uma vez que a análise dos dados (capítulo **5.4. Análise**) demonstrou que para todas as classes de classificação não existe um equilíbrio de representação dos valores possíveis, foi necessário aplicar algoritmos de balanceamento dos dados. Neste trabalho, optou-se pelo uso de dois algoritmos, *SMOTE* e *Random Oversampling*. Segue-se uma breve descrição do funcionamento de ambos:

- **SMOTE (Synthetic Minority Over-sampling Technique)**: a partir da população original de instâncias das classes minoritárias, com base no uso do algoritmo vizinho mais próximo, é selecionada uma das suas instâncias e N vizinhos. Após a seleção, é criada uma nova instância com base na interpolação dos atributos da instância selecionada e dos vizinhos selecionados [133], [134], [135]. Este processo repete-se até que todas as classes minoritárias possuam o mesmo número de instâncias que a classe maioritária;
- **Random OverSampling**: a partir da população original de instâncias para as classes minoritárias, iterativamente é escolhida uma instância e ela é duplicada [122], [133]. Este processo repete-se até que todas as classes minoritárias possuam o mesmo número de instâncias que a classe maioritária.

No entanto, neste trabalho, o processo de balanceamento dos dados foi aplicado apenas às classes de problemas de classificação. Esta escolha fundamentou-se pela falta de conhecimento do funcionamento do treino de regressores e não foi explorado muito mais para os resultados da predição do ECTS realizados, depois de realizado um treino básico e experimental.

6.2. Resultados

Logo após ser fornecido o primeiro *dataset* ao autor no mês de março de 2024, iniciaram-se de imediato o treino e avaliações de modelos de ML. No entanto, os dados referentes aos ECTS matriculados e aprovados nesse momento continham diversas gralhas nas suas instâncias, devido aos processos de reestruturação de vários cursos e aos respetivos planos de transição. Assim, uma vez que os modelos de ML estavam a ser treinados em dados malformados, todos os resultados obtidos dos treinos realizados nessa altura foram descartados e aguardou-se o envio de novos dados.

Interessa relembrar que ao longo do desenvolvimento deste trabalho foram entregues diferentes versões dos dados, em que nas mais recentes eram adicionados novos atributos. Dado que a apresentação dos resultados obtidos para cada uma das versões tornar-se-ia muito extensa, optou-se pela apresentação dos resultados obtidos para a última versão (versão 4). Adicionalmente, interessa referir que todos os resultados de treino dos modelos de ML para cada uma das versões encontram-se presentes no anexo **A. Resultados obtidos**.

A versão 4 dos dados é constituída por 19 atributos, que se passa a enumerar: escola, curso, ECTS matriculados, tipo de ingresso, sexo, nacionalidade, naturalidade, distrito, idade, tipos de aluno, regime de estudo, habilitação académica do pai, habilitação académica da mãe, situação profissional do pai, situação profissional da mãe, grupo profissional do pai, grupo profissional da mãe, nota de ingresso e habilitação anterior. Interessa referir que estes são os atributos antes da realização do pré-processamento apresentado anteriormente (ver **Figura 26**), pois, após a sua aplicação o *dataset* de treino é composto por 225 atributos.

Antes de serem apresentados os resultados finais para cada uma das classes de predição identificadas no capítulo **5.3.1. Classes investigadas**, é relevante destacar que, em alguns casos, os resultados obtidos, inicialmente, pareciam ser extremamente bons. No entanto, o autor descobriu que nesses casos o treino estava a ser feito incorretamente.

De forma a demonstrar o impacto de dois erros cometidos no treino de modelos de ML, é apresentada uma breve comparação entre os resultados obtidos com o treino adequado, **Figura 61**, e os resultados obtidos quando era cometido um dos erros, **Figura 62** e **Figura 63**. A comparação é feita com base nos resultados de um treino de validação cruzada utilizando o algoritmo *Random Forest* para predizer a continuação de estudos (classe "continua_estudos").

Iteração num.	Exatidão	F1 (média sem ponderação)
1	82.157676	72.871728
2	83.609959	75.527491
3	82.365145	73.668820
4	85.062241	77.930403
5	80.705394	70.771307
6	81.742739	72.443544
7	82.987552	74.322393
8	82.157676	74.173209
9	80.290456	69.695365
10	81.327801	73.325627

Figura 61 - Valores das métricas exatidão e F1 de um modelo *Random Forest* treinado corretamente sem presença de classes como atributos ou avaliação em instâncias duplicadas

O primeiro erro cometido num desenvolvimento inicial foi o esquecimento de remoção das classes dos dados de treino. Este esquecimento, por sua vez, resultou num treino com valores de exatidão e F1 extremamente bons. Porém, verificou-se que devia-se ao facto de que para o treino de predição de “continua_estudos” uma das classes de risco, “risco_v1”, não tinha sido removida dos dados. Através de uma comparação dos resultados realistas, **Figura 61**, com os resultados deste erro, **Figura 62**, verifica-se que a presença de uma classe como atributo de treino influência fortemente e negativamente o modelo preditivo. Além disso, destaca-se que caso estes modelos fossem exportados para uma aplicação, não seria possível o seu uso, uma vez que para a predição é necessário o valor de “risco_v1” que no momento de predição não estaria disponível.

Iteração num.	Exatidão	F1 (média sem ponderação)
1	93.360996	90.752332
2	94.398340	92.268069
3	91.286307	87.631965
4	92.738589	89.977126
5	93.153527	90.661273
6	91.493776	87.964897
7	92.323651	89.069290
8	92.531120	89.659611
9	91.286307	87.787162
10	91.701245	88.510679

Figura 62 - Valores das métricas exatidão e F1 de um modelo *Random Forest* treinado com a presença de uma classe nos atributos de treino

O segundo erro cometido durante o treino dos modelos de ML foi a realização do balanceamento de dados antes de ser feita a divisão nos conjuntos de treino e teste. Este erro só ocorre quando é realizado o balanceamento por *oversampling*, uma vez que esta duplica as instâncias existentes. Ao duplicar as instâncias antes de ser feita a divisão dos dados, é quase certo que irá existir uma fuga de instâncias iguais para ambos os conjuntos e isso não deveria acontecer, dado que o conjunto de teste deve ser

composto por instâncias totalmente diferentes do conjunto de treino. No entanto, em contraste com o erro de utilizar classes como atributos (**Figura 62**), neste caso só a avaliação do modelo de ML é afetada e espera-se que o modelo tenha um desempenho muito superior ao que realmente terá.

Ao comparar os resultados obtidos no treino sem erros, ilustrados na **Figura 61**, com os resultados onde houve a fuga de instâncias duplicadas, **Figura 63**, observa-se uma diferença significativa nas métricas. Desta forma, para a evitar resultados artificialmente otimistas, o balanceamento foi apenas realizado ao conjunto de treino, evitando qualquer fuga das suas instâncias para o conjunto de treino.

Iteração num.	Exatidão	F1 (média sem ponderação)
1	87.132867	87.086160
2	88.531469	88.489839
3	88.951049	88.928873
4	88.811189	88.774277
5	90.069930	90.066121
6	87.972028	87.943137
7	90.056022	90.043816
8	89.215686	89.210924
9	88.375350	88.350465
10	86.414566	86.404939

Figura 63 - Valores das métricas exatidão e F1 de um modelo *Random Forest* quando a sua avaliação é feita em instâncias duplicadas

Adicionalmente, é importante destacar que, para todos os algoritmos, sejam eles responsáveis por treinar modelos de ML, dividir dados (em conjunto de treino e teste ou *folds*) ou balanceamento, e que necessitam de utilizar aleatoriedade para o seu funcionamento, o valor de *random seed* (parâmetro "random_state") atribuído foi 500. Desta forma, é possível replicar todos os resultados apresentados neste trabalho, desde que os dados de treino sejam exatamente os mesmos e estejam ordenados da mesma forma.

Na investigação deste trabalho, foram realizados três tipos de treino: validação cruzada com 10 *folds*; *holdout* com uma divisão de 75% para treino e 25% para teste; e treino nos dados de 2019/2020 a 2021/2022 e teste (avaliação) nos dados do ano letivo 2022/2023. Dado que a apresentação detalhada para cada classe seria inexplorável, uma vez que, à exceção da classe "ects_realizados", cada classe possui nove resultados distintos. Isto porque, para cada tipo de treino, foi realizado um treino sem balanceamento de instâncias, um com balanceamento SMOTE e um com algoritmo de balanceamento *Random Oversampling*. Assim, para cada classe, é feita uma breve apresentação dos seus resultados, apenas sendo referidos os resultados mais relevantes para a validação do desempenho dos modelos de ML treinados. Interessa destacar que, apesar de não serem apresentados todos os resultados, os mesmos podem ser encontrados no anexo **A. Resultados obtidos**.

6.2.1. Risco Original

O primeiro treino realizado foi a predição da classe “risco_original”. Era esperado que esta seria a única classe a ser investigada, porém, os resultados ficaram aquém do pretendido. Conforme se pode observar na **Tabela 7**, os resultados de todos os algoritmos não foram bons, uma vez que nenhum consegue atingir os 80% de exatidão e a métrica F1-Score não chega a 60%. Assim, pode-se concluir que para esta classe os modelos de ML têm um desempenho insatisfatório, uma vez que para os diferentes níveis de risco, os modelos demonstraram não conseguir aprender os padrões pretendidos.

Tabela 7 - Resultados do treino *holdout* para a classe "risco_original"

Algoritmo	Exatidão	Precisão	Recall	F1 Score	F1 Score (ponderado)
DT	64,65	44,05	45,18	44,41	64,36
NB	68,22	45,88	48,55	46,79	67,06
RF	72,78	56,15	46,57	47,88	68,97
GB	72,70	50,19	45,88	46,23	69,50
XGB	73,53	55,83	49,82	51,45	71,16
XGBRF	72,78	52,59	44,18	44,12	68,33
LGBM	71,29	53,09	47,28	48,62	68,83
KNN	70,21	45,04	40,96	42,03	67,23
SVM	67,14	25,90	29,76	27,07	56,36
LoR	71,54	49,41	44,83	45,39	68,06
NN_MLP	65,73	43,33	44,62	43,72	65,38

Uma das causas que pode estar associada ao mau desempenho, é o facto de que o modelo de ML tenta predizer um valor de um conjunto de cinco. Dado que não existe um número significativo de instâncias, e não existe um equilíbrio de instâncias por nível, isso pode justificar este resultado. Embora tenha sido procurado balancear o conjunto de treino, os resultados continuaram a ser insatisfatórios.

Para ambos os algoritmos de平衡amento, os resultados mostraram-se, geralmente, ser semelhantes aos obtidos sem平衡amento. Para comparação, os resultados do treino com平衡amento *SMOTE* são apresentados na **Tabela 8**. A observação desta tabela, revela que a maioria dos algoritmos no treino balanceado obtiveram piores resultados, no entanto, o algoritmo *LightGBM* apresentou uma ligeira melhoria sendo o melhor num treino balanceado. Apesar disso, esperava-se que os resultados do treino balanceado fossem significativamente melhores, porém isso não aconteceu.

Tabela 8 - Resultados do treino *holdout* com balanceamento SMOTE para a classe "risco_orginal"

Algoritmo	Exatidão	Precisão	Recall	F1 Score	F1 Score (ponderado)
DT	61,74	42,01	42,20	42,09	62,15
NB	65,64	45,13	50,07	46,39	66,23
RF	71,62	51,57	47,41	48,16	69,01
GB	70,87	50,77	50,06	49,37	69,41
XGB	71,37	51,45	47,16	48,17	69,19
XGBRF	69,13	46,57	48,40	47,06	68,54
LGBM	72,61	55,90	50,43	51,58	70,51
KNN	47,97	39,46	41,89	38,04	53,58
SVM	64,23	44,78	47,13	43,97	65,37
LoR	60,08	45,02	49,77	46,31	62,72
NN_MLP	63,15	43,81	43,66	43,69	63,22

Além do balanceamento dos dados, foi investigado quais os níveis de risco apresentavam maior dificuldade para os modelos de ML aprenderem. Com base na análise dos resultados individuais, do treino sem balanceamento, para cada nível de risco, apresentados nas **Tabela 9** e **Tabela 10**, verifica-se que os níveis "Baixo" e "Alto" são aqueles onde os modelos de ML têm uma maior dificuldade em aprender os seus padrões. Em contraste, o nível "Nenhum" é o nível que os modelos aprendem com maior eficácia. Esta evidência é normal, uma vez que este nível é o valor em maioria em todos os dados (ver **Figura 51**), dando, assim, aos modelos de ML mais exemplos para aprender. Porém, como constatado na **Tabela 8**, o treino com o equilíbrio SMOTE não melhorou os resultados, levando a que os níveis "Baixo" e "Alto" continuassem com um mau desempenho.

Tabela 9 - Resultados individuais de *holdout* da classe "risco_orginal" para os níveis "Nenhum" e "Baixo"

Algoritmo	Nenhum			Baixo		
	Precisão	Recall	F1 Score	Precisão	Recall	F1 Score
DT	80,35	81,89	81,11	37,2	33,89	35,47
NB	83,22	83,68	83,45	42,5	37,78	40
RF	78,96	93,69	85,7	53,57	33,33	41,1
GB	81,1	93	86,65	51,11	38,33	43,81
XGB	81,83	92,04	86,64	51,88	46,11	48,82
XGBRF	79,58	94,65	86,47	56,12	30,56	39,57
LGBM	80,61	90,67	85,35	46,15	40	42,86
KNN	80	92,73	85,9	43,4	38,33	40,71
SVM	67,94	100	80,91	0	0	0
LoR	80,21	92,32	85,84	47,29	33,89	39,48
NN_MLP	82,3	81,62	81,96	39,13	45	41,86

Tabela 10 - Resultados individuais de *holdout* da classe “risco_orginal” para os níveis “Médio”, “Alto”, “Não comparece”

Algoritmo	Médio			Alto			Não comparece		
	Precisão	Recall	F1 Score	Precisão	Recall	F1 Score	Precisão	Recall	F1 Score
DT	15,58	13,95	14,72	30,16	41,3	34,86	56,96	54,88	55,9
NB	8,89	4,65	6,11	39,22	43,48	41,24	55,56	73,17	63,16
RF	33,33	4,65	8,16	53,85	30,43	38,89	61,05	70,73	65,54
GB	25	4,65	7,84	31,43	23,91	27,16	62,3	69,51	65,71
XGB	25,93	8,14	12,39	54,84	36,96	44,16	64,67	65,85	65,26
XGBRF	33,33	2,33	4,35	36	19,57	25,35	57,89	73,78	64,88
LGBM	29,17	8,14	12,73	50	34,78	41,03	59,54	62,8	61,13
KNN	16,67	5,81	8,62	23,08	13,04	16,67	62,07	54,88	58,25
SVM	0	0	0	0	0	0	61,54	48,78	54,42
LoR	11,11	2,33	3,85	50	26,09	34,29	58,46	69,51	63,51
NN_MLP	14	8,14	10,29	25	30,43	27,45	56,21	57,93	57,06

6.2.2. Risco Binário

Depois dos resultados obtidos do “risco_original”, foi investigado o desempenho dos modelos de ML perante uma classificação binária de risco, em que os alunos só eram sinalizados com ou sem risco. Os resultados obtidos, apresentados na **Tabela 11**, demonstram uma melhoria significativa comparativamente aos de “risco_original” (**Tabela 7**). Estes apresentam principalmente uma melhoria na métrica *F1-Score*, que passou dos 50% para superiores a 70%, que se deve ao facto de ter sido reduzido o número de valores possíveis a predizer, de cinco para dois. No treino *holdout* do “risco_binario” o algoritmo que treinou o melhor modelo foi o *Gradient Boosting*.

Tabela 11 - Resultados do treino holdout para a classe "risco_binario"

Algoritmo	Exatidão	Precisão	Recall	F1 Score	F1 Score (ponderado)
DT	72,45	72,29	72,41	72,32	72,49
NB	75,52	76,42	74,46	74,64	75,05
RF	79,34	79,90	78,56	78,81	79,10
GB	79,83	80,56	79,00	79,27	79,57
XGB	78,09	78,35	77,43	77,63	77,91
XGBRF	79,25	80,41	78,25	78,53	78,87
LGBM	79,09	79,53	78,36	78,59	78,88
KNN	76,51	76,79	75,79	75,98	76,29
SVM	73,53	74,99	72,20	72,23	72,76
LoR	78,42	78,92	77,65	77,88	78,18
NN_MLP	75,35	75,22	74,97	75,05	75,29

Com resultados positivos para o treino de *holdout*, foi analisado o desempenho dos modelos no treino de validação cruzada. O objetivo desta análise é verificar como os modelos de ML se desempenham perante diferentes populações (instâncias) de treino do utilizado *dataset*. A validação cruzada permite avaliar se as instâncias utilizadas para treino impactam significativamente a avaliação do modelo de ML, sendo que esta avaliação é realizada com base na verificação dos médios e de desvio padrão de cada métrica utilizada. Sendo que são treinados tantos modelos de ML como o número de *folds*, a média e desvio padrão de cada métrica é calculada com base neles todos. Adicionalmente, é importante referir, que o melhor modelo é aquele que apresente uma média da métrica o mais elevado possível e um menor desvio padrão.

Ao se observar os resultados obtidos para a validação cruzada (10 *folds*, ou seja, foram treinados 10 modelos de ML), apresentados na **Tabela 12**, verificou-se que o desvio médio para cada métrica é de aproximadamente 4% o que pode ser considerado relativamente reduzido e indicativo que existe uma distribuição boa de instâncias para os padrões existentes, uma vez que os resultados diferem, em média, 4%.

É importante destacar que as células das tabelas de resultados da validação cruzada contêm dois valores: o primeiro (que se encontra na parte superior) corresponde à média da métrica, enquanto o segundo (que se encontra na parte inferior) representa o desvio padrão. Desta forma, estes valores refletem o desempenho de todos os modelos treinados (ao todo 10) durante o processo de validação cruzada.

Tabela 12 - Resultados do treino validação cruzada para a classe "risco_binario"

Algoritmo	Exatidão	Precisão	Recall	F1 Score	F1 Score (ponderado)
DT	71,2230 3,5346	71,3550 3,4836	71,0830 3,3068	70,9420 3,4307	71,1490 3,4848
NB	76,2850 4,5121	77,4970 3,9244	75,2780 5,0106	75,2600 5,1639	75,6930 4,9663
RF	78,1340 4,4794	79,3550 4,8329	77,3490 4,5077	77,4370 4,6039	77,7630 4,5463
GB	79,1910 4,4774	80,4980 4,7109	78,3700 4,5871	78,4870 4,7086	78,8110 4,6219
XGB	78,2790 4,2789	79,1300 4,4578	77,5740 4,3458	77,6730 4,4505	77,9800 4,3803
XGBRF	79,3770 4,2070	81,2350 4,4187	78,3520 4,3918	78,4960 4,5434	78,8650 4,4297
LGBM	78,1950 4,7222	79,2610 4,9956	77,4420 4,7421	77,5390 4,8514	77,8570 4,7905
KNN	75,9330 4,4133	76,3220 4,7427	75,2900 4,3791	75,4060 4,4144	75,7180 4,3842
SVM	75,6640 4,2914	76,5580 3,8113	74,8080 4,8954	74,7460 4,9345	75,1490 4,7211
LoR	77,6340 4,2413	78,8670 3,9813	76,8390 4,5749	76,8270 4,7124	77,1770 4,5653
NN_MLP	71,7430 3,3734	71,8950 3,5079	71,6270 3,4085	71,4600 3,3430	71,6660 3,3328

Complementarmente, foi verificado se era possível melhorar os resultados ao aumentar a média e diminuir o desvio padrão das métricas quando utilizado um algoritmo de balanceamento. Porém, para a classe “risco_binario” quando balanceado o conjunto de treino, os resultados (**Tabela 13**) demonstraram-se ser muito semelhantes aqueles sem balanceamento (**Tabela 12**), o que poderá ser indicativo de que, para os dados disponíveis de treino para este trabalho, o desequilíbrio de representação de cada valor da classe não é um problema significativo.

Tabela 13 - Resultados do treino validação cruzada para a classe “risco_binario” com balanceamento *SMOTE* no conjunto de treino

Algoritmo	Exatidão	Precisão	Recall	F1 Score	F1 Score (ponderado)
DT	70,8510 3,9486	70,8590 3,9971	70,7020 4,0058	70,5750 3,9818	70,7950 3,9499
NB	76,1830 4,5405	77,3030 3,9770	75,2060 5,0250	75,1880 5,1775	75,6120 4,9821
RF	78,0490 4,3948	79,1440 4,7602	77,3550 4,3938	77,4190 4,4793	77,7240 4,4331
GB	79,0040 4,7231	80,0810 4,9577	78,3240 4,7263	78,4080 4,8583	78,6960 4,8025
XGB	77,7600 4,6330	78,7010 4,9082	77,1120 4,5603	77,1710 4,7008	77,4620 4,6667
XGBRF	78,9220 4,2261	80,6350 4,5005	78,0650 4,2658	78,1320 4,4651	78,4640 4,3889
LGBM	78,1530 4,8890	79,0630 5,3085	77,5290 4,7509	77,6170 4,8717	77,8950 4,8608
KNN	73,9410 4,9373	74,0250 4,9063	73,8300 4,7476	73,7250 4,8604	73,9060 4,8936
SVM	75,9950 3,5581	76,5150 3,4589	75,3600 4,0080	75,3610 3,9105	75,6890 3,7669
LoR	76,7230 4,5131	77,5950 4,5095	76,1880 4,5871	76,1280 4,6703	76,4120 4,6149
NN_MLP	72,2410 4,4272	72,4920 4,4703	72,1210 4,2051	71,9610 4,2965	72,1540 4,3518

No entanto, apesar dos resultados positivos, o uso desta classe limita, de certa forma, a predição de risco desejada, uma vez que esta classe apenas indica se um aluno é ou não de risco. Levando com que seja impossível distinguir os alunos que poderão querer continuar os estudos e aqueles que poderão desistir ao fim do primeiro ano. De forma a evitar esta limitação, foi explorado o uso da classe de risco otimizado, cujos resultados são apresentados a seguir.

6.2.3. Risco Otimizado

Em uma tentativa de ser possível predizer o nível de risco, foi experimentado diminuir o número de níveis de risco de cinco para quatro, com objetivo de verificar se os modelos de ML conseguiram melhor aprender os padrões de cada nível. Desse modo, foram treinados e avaliados modelos de ML para a classe “risco_optimizado”, que por sua vez prediz quatro níveis: “Nenhum”, “Baixo”, “Médio” e “Alto”. Contudo novamente os resultados ficaram aquém do pretendido e foram ligeiramente melhores aos de “risco_original”.

Quando visualizados os resultados obtidos no treino *holdout* para esta classe, apresentados na **Tabela 14**, verifica-se que dois algoritmos melhores algoritmos mostraram uma pequena melhoria nas suas métricas (comparativamente aos da **Tabela 7**). Contudo, era esperado que estes conseguissem atingir os 80% de exatidão e superar os 60% de *F1-Score*.

Tabela 14 - Resultados do treino holdout para a classe "risco_optimizado"

Algoritmo	Exatidão	Precisão	Recall	F1 Score	F1 Score (ponderada)
DT	67,3	49,2	49,05	49,12	67,14
NB	71,2	48,96	51,48	49,83	69,67
RF	74,44	56,58	50,55	50,7	70,67
GB	75,27	61,07	52,35	52,63	71,85
XGB	75,27	57,30	53,20	53,82	72,76
XGBRF	75,6	56,58	50,81	50,00	71,14
LGBM	74,85	56,11	53,04	53,29	72,34
KNN	72,61	50,76	48,51	48,78	69,73
SVM	72,45	35,38	44,04	39,21	63,60
LoR	74,61	50,29	50,34	49,42	70,72
NN_MLP	71,29	52,87	52,93	52,80	70,79

No entanto, mesmo realizando o balanceamento das instâncias de treino, os resultados não melhoraram, conforme apresentado na **Tabela 15**. Desta forma, pode-se concluir, e com base nos resultados de “risco_orginal” e “risco_binario”, que o balanceamento para os dados disponíveis e utilizados neste estudo não permite a obtenção de melhores resultados, pois, os modelos, mesmo com desequilíbrio de instâncias por valor da classe conseguem aprender os padrões dos dados.

Tabela 15 - Resultados do treino *holdout* com balanceamento SMOTE para a classe "risco_otimizado"

Algoritmo	Exatidão	Precisão	Recall	F1 Score	F1 Score (ponderada)
DT	67,22	47,96	48,04	47,96	67,00
NB	69,79	51,18	52,91	51,88	69,83
RF	74,44	56,57	52,39	52,42	71,60
GB	74,85	56,59	54,67	54,42	72,82
XGB	74,19	55,86	52,70	53,17	71,87
XGBRF	73,53	55,84	55,75	54,69	72,17
LGBM	72,78	52,66	50,33	50,38	70,12
KNN	52,45	46,32	46,24	43,84	57,73
SVM	69,05	51,48	51,18	50,52	68,96
LoR	64,56	52,06	54,58	52,46	67,14
NN_MLP	66,89	49,33	48,46	48,86	66,64

Também foi confirmado quais eram os níveis que os modelos tinham piores resultados preditivos. A observação das métricas para cada nível individual, **Tabela 16** e **Tabela 17**, permitiu observar que para o "risco_otimizado" os níveis "Baixo" e "Médio" são aqueles cujo modelos de ML não conseguem compreender e aprender os seus padrões de forma mais acertada. Adicionalmente, realça-se que o nível "Alto", comparativamente ao "risco_original" (**Tabela 10**), teve um aumento nos seus resultados, dado que este nível de "risco_otimizado" converge o nível "Alto" com "Não comparece" de "risco_original".

Além disso, continua a observar-se que o modelo de ML treinado por SVM não consegue de qualquer modo aprender os padrões das instâncias pertencentes aos níveis "Baixo" e "Médio". Pois, em todas as métricas para ambos estes níveis, o modelo de ML foi avaliado com o pior valor possível, zero.

Tabela 16 - Resultados individuais de *holdout* da classe "risco_otimizado" para os níveis "Nenhum" e "Baixo"

Algoritmo	Nenhum			Baixo		
	Precisão	Recall	F1 Score	Precisão	Recall	F1 Score
DT	81,22	81,89	81,56	33,7	34,27	33,98
NB	83,18	84,09	83,63	42,58	37,08	39,64
RF	78,65	93	85,23	54,64	29,78	38,55
GB	80,09	92,73	85,95	51,82	32,02	39,58
XGB	81,75	92,18	86,65	52,59	39,89	45,37
XGBRF	79,65	94,51	86,45	62,82	27,53	38,28
LGBM	81,37	91,08	85,95	50	38,76	43,67
KNN	80,02	92,32	85,73	43,62	36,52	39,76
SVM	73,79	96,16	83,5	0	0	0
LoR	79,83	92,87	85,86	51,92	30,34	38,3
NN_MLP	83,74	84,77	84,25	41,4	43,26	42,31

Tabela 17 - Resultados individuais de holdout da classe “risco_otimizado” para os níveis “Médio” e “Alto”

Algoritmo	Médio			Alto		
	Precisão	Recall	F1 Score	Precisão	Recall	F1 Score
DT	16	14,46	15,19	65,89	65,58	65,73
NB	4,76	2,41	3,2	65,31	82,33	72,84
RF	23,08	3,61	6,25	69,96	75,81	72,77
GB	41,67	6,02	10,53	70,71	78,6	74,45
XGB	24	7,23	11,11	70,85	73,49	72,15
XGBRF	16,67	1,2	2,25	67,19	80	73,04
LGBM	20,83	6,02	9,35	72,25	76,28	74,21
KNN	10,53	2,41	3,92	68,88	62,79	65,69
SVM	0	0	0	67,72	80	73,35
LoR	0	0	0	69,42	78,14	73,52
NN_MLP	16,67	12,05	13,99	69,68	71,63	70,64

Por último, foi também investigado o desempenho dos modelos de ML quando treinados com instâncias dos anos letivos 2019/2020 a 2021/2022 e testados com as instâncias do ano letivo 2022/2023. Este treino permite simular, de forma teórica, o desempenho dos modelos num cenário real. Assim, os resultados obtidos neste cenário são apresentados na **Tabela 18**

Tabela 18 - Resultados do treino *holdout* da classe “risco_otimizado”, cujo conjunto de treino é composto pelas instâncias dos datasets dos anos letivos 2019/2020, 2020/2021, 2021/2022 e as instâncias de teste são pertencentes ao ano letivo 2022/2023

Algoritmo	Exatidão	Precisão	Recall	F1 Score	F1 Score (ponderada)
DT	60,03	37,34	34,03	35,26	59,9
NB	67,99	46,38	45,46	45,15	66,67
RF	70,35	47,04	39,59	41,62	65,73
GB	70,02	46,60	40,44	41,94	66,01
XGB	70,19	50,82	42,56	45,29	67,12
XGBRF	70,84	49,48	38,71	40,10	65,26
LGBM	70,11	48,99	42,13	44,42	67,03
KNN	68,89	44,75	37,67	39,79	65,14
SVM	61,98	12,40	20,00	15,31	47,43
LoR	67,42	30,30	34,53	30,78	60,16
NN_MLP	61,98	12,40	20,00	15,31	47,43

A observação destes resultados (**Tabela 18**), que simulam um cenário real, revela que os modelos de ML apresentam um desempenho preditivo inferior quando comparado com o treino realizado para todos os dados disponíveis (**Tabela 14**). Contudo, esta ocorrência era esperada, dado que cada ano letivo apresenta uma população de instâncias (alunos) completamente diferente, que por sua vez pode

conter novos valores de atributos cujo modelos de ML (depois do pré-processamento existente) os trata como desconhecidos. Além disso, esses resultados podem também refletir o impacto do número reduzido de instâncias de treino, visto que o *dataset* de 2022/2023 contém aproximadamente um quarto das instâncias disponíveis de treino neste trabalho. Posto isto, acredita-se que com mais dados de treino em trabalhos futuros, essa diferença poderá ser menos significativa, o que permitirá que os modelos estabilizem a sua aprendizagem e melhorem as suas regras padrões.

6.2.4. Continua estudos

Para além da predição do nível de risco dos alunos, também foi realizada a predição da continuidade dos estudos de um aluno do IPCB após um ano. Esta predição está diretamente alinhada com predições exploradas no estado da arte (**3. Estudo do estado da arte**) e com a predição do abandono escolar que vinha referida na submedida de Inovação e Modernização Pedagógica no Ensino Superior - Programa de Promoção de Sucesso e Redução de Abandono Escolar no Ensino Superior [7]. Além disso, sendo esta uma classificação binária, tal como a classe “risco_binário”, obteve melhores resultados, uma vez que o problema se torna mais simples devido a um menor número de classes possíveis.

A **Tabela 19** apresenta os resultados do treino de validação cruzada para a classe “continua_estudos”. Estes resultados mostram que, embora sejam positivos, a métrica *F1-Score* não atinge o objetivo pretendido de 80%. Além disso, ao comparar com a classe “risco_binario”, verifica-se um ligeiro aumento na exatidão, mas um decréscimo nas restantes métricas, sendo a métrica *F1-Score* a mais preocupante. Este fenómeno pode ser explicado pelo maior número de instâncias da classe minoritária no caso de “risco_binario”, o que permite que os modelos de ML aprendam melhor os seus padrões, uma vez que dispõem de mais exemplos (instâncias) para se basearem.

Tabela 19 - Resultados do treino validação cruzada para a classe "continua_estudos"

Algoritmo	Exatidão	Precisão	Recall	F1 Score	F1 Score (ponderada)
DT	72,4670 2,1532	64,2950 2,5272	64,2940 3,0074	64,1380 2,6382	72,4530 1,9970
NB	78,1330 3,3040	72,1290 4,0805	72,5290 4,4132	71,9370 3,9197	78,2590 3,0923
RF	80,5200 2,4226	76,3130 4,4048	69,0720 4,0492	70,9100 4,2038	78,9580 2,8219
GB	81,0180 2,3003	77,0660 4,2284	69,7950 3,3499	71,8100 3,5768	79,5710 2,5101
XGB	79,1500 2,4831	73,6780 4,2357	68,0150 3,4032	69,5480 3,5855	77,7780 2,5556
XGBRF	80,3740 2,5541	75,9540 4,6100	69,0230 4,0437	70,8250 4,2527	78,8570 2,8953
LGBM	79,9590 2,5079	75,1760 4,4456	68,7440 3,5658	70,4720 3,7706	78,5280 2,6570
KNN	78,4870 2,8255	72,0910 4,8188	67,4870 4,0903	68,8770 4,4354	77,2120 3,1239
SVM	74,2120 0,3721	60,6360 21,5447	50,3690 0,5352	43,5120 1,0928	63,6080 0,6613
LoR	80,5620 2,3484	76,1780 4,2181	69,3080 3,8692	71,1430 4,0345	79,0810 2,7225
NN_MLP	73,7970 4,1587	65,8870 5,4939	65,2180 4,9346	65,3820 5,0675	73,5940 3,9949

Tal como nos restantes treinos apresentados até aqui, também foi analisado o desempenho dos modelos de ML quando as instâncias do conjunto de treino eram devidamente balanceadas. Contudo, tal como observado também nas outras previsões, os resultados mantiveram-se semelhantes aos obtidos no treino sem balanceamento, conforme apresentado na **Tabela 20**.

Tabela 20 - Resultados do treino validação cruzada para a classe "continua_estudos", cujo conjunto de treino é balanceado com o algoritmo *SMOTE*

Algoritmo	Exatidão	Precisão	Recall	F1Score	F1Score (ponderada)
DT	72,2400 2,9482	64,0980 3,7410	64,3760 4,1847	64,1130 3,9028	72,3310 2,9362
NB	77,4700 3,4443	71,4250 4,0590	72,2650 4,6138	71,3970 4,0099	77,7070 3,1953
RF	79,6670 2,7982	74,3680 4,8155	68,8620 4,1278	70,4120 4,3512	78,3780 3,0449
GB	80,0420 3,0209	74,8090 4,7989	70,4980 4,2045	71,7770 4,2498	79,1200 3,0934
XGB	79,1930 2,8671	73,4640 4,5661	68,8800 3,8737	70,2420 4,0071	78,0960 2,9293
XGBRF	78,7150 4,3691	73,0290 6,0751	69,7580 4,6899	70,6820 5,1709	78,0420 4,1032
LGBM	79,8970 2,5850	74,8560 4,3750	69,2250 3,6309	70,8090 3,7995	78,6450 2,7023
KNN	61,6200 4,1295	61,2910 3,0480	64,6400 3,9318	59,0710 3,8749	63,9490 3,9137
SVM	77,8850 3,1509	71,8230 3,9376	72,3620 4,4766	71,6790 3,8391	78,0310 2,9708
LoR	74,1310 5,1440	67,8290 5,5237	69,0660 5,0576	68,0020 5,4046	74,7030 4,6764
NN_MLP	72,6560 2,6177	65,0930 3,2003	65,7310 3,6980	65,1450 3,2313	72,9080 2,4735

Adicionalmente, uma vez que um modelo de ML desta classe será utilizado em um algoritmo de sinalização do nível de risco (a ser apresentado em **6.3. Combinação de modelos de ML explorada para a predição do nível de risco**), foi observado o desempenho dos modelos quando treinados nos primeiros três anos letivos dos dados disponíveis (2019 a 2022) e testados no último ano letivo (2022/2023). Os resultados para este cenário são apresentados na **Tabela 21** que permite verificar que, ao contrário da classe "risco_otimizado", os resultados são muito próximos daqueles no *dataset* com todos os anos letivos. Assim, esta predição mostra-se ser estável para ser utilizada em predições de anos letivos futuros.

Tabela 21 - Resultados do treino holdout da classe “continua_estudos”, cujo conjunto de treino é composto pelas instâncias dos datasets dos anos letivos 2019/2020, 2020/2021, 2021/2022 e as instâncias de teste são pertencentes ao ano letivo 2022/2023

Algoritmo	Exatidão	Precisão	Recall	F1Score	F1Score (ponderada)
DT	72,79	60,74	61,16	60,94	73,05
NB	78,64	69,34	71,07	70,09	79,09
RF	80,99	72,48	64,81	66,91	79,06
GB	81,40	73,02	66,67	68,70	79,92
XGB	80,83	71,76	66,85	68,56	79,62
XGBRF	81,72	73,88	66,48	68,70	80,07
LGBM	80,67	71,45	66,61	68,29	79,44
KNN	79,04	68,27	63,29	64,80	77,40
SVM	78,15	39,07	50,00	43,87	68,56
LoR	81,48	73,84	64,85	67,15	79,36
NN_MLP	21,85	10,93	50,00	17,93	78,40

Na simulação do uso dos modelos de ML num cenário real de predição para este problema de classificação (continuação de estudos), verificou-se que os modelos de ML treinados com os algoritmos *Naïve Bayes* e *XGBoost* com *Random Forest* obtiveram os melhores resultados. Curiosamente, observa-se um bom desempenho por parte do modelo de ML treinado com *Naïve Bayes*, uma vez que, em termos de complexidade e requisitos computacionais, este é um dos algoritmos mais leves treinados no âmbito deste trabalho. Assim, caso sejam necessárias soluções mais simples e com menor exigência de recursos computacionais, a utilização do modelo de ML treinado com *Naïve Bayes* apresenta-se ser viável.

Em contraste, um dos modelos mais complexo e exigente em termos de computação, rede neuronal (Perceptrão de multicamadas, NN_MLP), apresentou-se ser extramente mau neste cenário, com uma diminuição elevada dos seus resultados. Contudo, não foi possível determinar a causa desta observação, dado que as redes neurais são consideradas caixas negras, não sendo possível compreender que padrões aprendeu.

6.2.5. ECTS Realizados

Um dos treinos experimentais focou-se na predição do número de ECTS realizados, o qual se trata de um problema de regressão. No entanto, este treino foi realizado de forma puramente experimental e não foi devidamente aprofundado, uma vez que o autor não possui o mesmo nível de conhecimento sobre o funcionamento e avaliação de modelos de ML para problemas regressão como possui para problemas de classificação. Assim, os resultados obtidos no treino *holdout* são apresentados na **Tabela 22**, e os da validação cruzada na **Tabela 23**.

Tabela 22 - Resultados de regressão do treino holdout para a classe "ects_realizados"

Algoritmo	R2	Erro Médio Quadrático	Raiz do Erro Quadrático Médio	Erro Absoluto Médio	Erro Absoluto Mediano
DT	0,2058	426,7712	20,65844	12,6361	6
RF	0,5882	221,2595	14,8748	10,253	6,585
GB	0,6187	204,8863	14,31385	10,3358	7,2212
XGB	0,5713	230,3837	15,1784	10,5139	6,6381
XGBRF	0,6072	211,0899	14,52893	10,5942	7,1476
LGBM	0,6002	214,8456	14,65761	10,2927	6,7683
KNN	0,5339	250,4643	15,82606	10,9734	7,4
SVM	0,011	543,3715	23,31033	17,1536	8,7354
LR	-2,5E+18	1,37E+21	3,7E+10	1,09E+09	8,1009
NN_MLP	0,5261	254,5729	15,95534	11,5293	8,0009

Tabela 23 - Resultados de regressão do treino validação cruzada para a classe "ects_realizados"

Algoritmo	R2	Erro Médio Quadrático	Raiz do Erro Quadrático Médio	Erro Absoluto Médio	Erro Absoluto Mediano
DT	0,1738	434,8130	20,7371	13,0253	6,5250
	0,1712	93,1998	2,1879	2,0900	2,1692
RF	0,5506	237,3298	15,3094	10,7117	7,1648
	0,0933	53,6149	1,7182	1,6595	1,6004
GB	0,5842	219,3410	14,7153	10,6676	7,3547
	0,0892	50,1439	1,6734	1,4715	1,3624
XGB	0,5455	239,7586	15,3998	10,8440	7,1504
	0,0881	50,3070	1,6138	1,5524	1,6184
XGBRF	0,5692	227,3242	14,9931	10,9128	7,5013
	0,0841	48,1921	1,5910	1,4225	1,3930
LGBM	0,5717	225,9009	14,9415	10,5913	7,1199
	0,0879	49,4559	1,6286	1,5213	1,4580
KNN	0,5024	263,0856	16,1349	11,0271	7,1050
	0,0886	53,4652	1,6581	1,5507	1,7034
SVM	-0,0004	530,8452	22,9646	17,0839	9,6016
	0,0836	85,0326	1,8636	1,7011	1,1962
LR	< -1000	>1000	>1000	>1000	8,4211
	>1000	>1000	>1000	>1000	1,4029
NN_MLP	0,4713	279,1543	16,6200	11,9705	8,2760
	0,0984	7,5436	1,7112	1,4551	1,1867

A análise dos resultados obtidos permite verificar que os modelos de ML apresentaram uma média de erro (Erro Absoluto Médio) na predição dos ECTS realizados de cerca de 10 (dez), o que equivale a duas UCs. No entanto, o valor mediano de erro é inferior, situando-se em torno de 7 (sete) ECTS, o que pode indicar a existência de predições com erros significativamente superiores, resultando numa média mais alta.

Adicionalmente, uma das métricas mais relevantes no contexto de problemas de regressão é a métrica R2 ou *R-Squared* que permite verificar a variação do desempenho do modelo de ML [136], [137], [138]. O indicado pela documentação de *Scikit-Learn* em [138], indica que quanto mais próximo o seu valor se encontra de 1, melhor é o desempenho do modelo. Assim, com base nessa indicação, o modelo que apresentou o melhor desempenho, tanto no treino *holdout* como no de validação cruzada, foi o *Gradient Boosting*. No entanto, apesar de não ter sido o melhor nas métricas de média e mediana de erro absoluto, o mesmo ficou muito perto e a sua diferença não é impactante de forma a ser optado pelo uso de outros algoritmos.

Por último, é importante destacar que o algoritmo de Regressão Linear apresentou resultados atípicos. Contudo, devido à falta de conhecimento aprofundado sobre o funcionamento de modelos de regressão, a causa desse desempenho não foi investigada. Assim, pode-se concluir que ou o algoritmo de Regressão Linear não consegue aprender de nenhuma forma os padrões presentes nos dados, ou está a ser cometido um erro no seu treino que ainda não foi detetado.

6.2.6. Intervalo ECTS Realizados

Ainda numa última fase de investigação do desempenho preditivo dos modelos de ML, foi feito o treino e avaliação de modelos para a classe “intervalo_ects_realizados”. Esta foi uma escolha suportada pelos professores orientadores, uma vez que o autor não possui um conhecimento detalhado do funcionamento de problemas de regressão.

Quando analisados os resultados obtidos para esta classe num treino *holdout*, apresentados na **Tabela 24**, constatou-se que apesar de os resultados ficarem abaixo do pretendido (com métricas a rondar os 80%), os modelos apresentaram um desempenho relativamente positivo. Dado que esta classe, possui apenas três valores possíveis, confirmou a tendência de melhoria nas métricas associadas a problemas de classificação multiclasse, Precisão, *Recall* e *F1-Score*, com um aumento registado de cerca de 20% para as mesmas (comparativamente à **Tabela 7**, “risco_original”, e **Tabela 14**, “risco_optimizado”). Adicionalmente, interessa realçar que, apesar de não serem apresentados neste relatório, os resultados obtidos nos treinos com balanceamento do conjunto de treino, foram, novamente, idênticos aos obtidos sem balanceamento. Estes podem ser consultados no anexo **A. Resultados obtidos**.

Tabela 24 - Resultados do treino *holdout* para a classe "intervalo_ects_realizados"

Algoritmo	Exatidão	Precisão	Recall	F1Score	F1Score (ponderada)
DT	68,96	62,68	62,02	62,33	69,01
NB	73,44	65,60	67,82	66,09	72,59
RF	75,93	70,05	66,49	67,25	74,24
GB	77,10	71,47	68,44	69,15	75,66
XGB	77,01	71,77	68,67	69,63	75,87
XGBRF	76,76	71,65	66,43	66,69	74,15
LGBM	77,51	72,65	69,24	70,38	76,38
KNN	74,27	67,61	63,93	65,31	73,02
SVM	71,70	56,78	56,83	51,82	63,18
LoR	76,35	70,12	67,15	67,75	74,71
NN_MLP	71,78	66,22	65,02	65,54	71,92

Para validar se os modelos de ML apresentavam um desempenho consistente com diferentes conjuntos de treino, através da **Tabela 25**, observa-se que as métricas apresentam um desvio padrão médio de 3,5%, o que pode ser considerado bom para este tipo de problemas de classificação. Além disso, verifica-se que não há uma diferença significativa entre o valor médio das métricas quando comparado ao treino *holdout*. Em ambos os tipos de treino, confirma-se que o algoritmo de ML com melhor desempenho foi o LightGBM. Em contraste, o SVM mostrou, novamente, ser o pior algoritmo para esta classificação.

Tabela 25 - Resultados do treino validação cruzada para a classe "intervalo_ects_realizados"

Algoritmo	Exatidão	Precisão	Recall	F1Score	F1Score (ponderada)
DT	67,2410 4,1647	60,2250 3,4367	59,3600 3,6316	59,4340 3,3861	67,1880 3,4809
NB	72,3870 2,9101	64,8290 4,4203	66,4850 1,9259	64,6000 3,1919	71,6100 2,2096
RF	75,6840 4,0230	71,1690 6,8828	65,8940 3,9575	66,6800 4,2190	73,9110 3,5794
GB	76,1420 3,3944	71,0960 5,2283	66,9350 3,7181	67,5110 3,5017	74,6350 3,0455
XGB	75,2910 3,5167	69,8680 5,0684	66,4540 3,7014	67,2590 3,8138	74,1210 3,2537
XGBRF	76,2450 3,3761	71,4010 6,0164	66,3770 3,9184	66,4630 4,3444	73,9620 3,3859
LGBM	75,5600 3,6614	70,7350 6,0166	66,7720 3,4896	67,6380 3,6701	74,3360 3,1804
KNN	74,2940 3,4546	68,3890 4,9562	64,2070 3,7278	65,4600 3,8887	72,9240 3,3063
SVM	72,3450 2,0901	62,65401 4,1790	57,4900 2,6881	52,2540 2,8964	63,9050 2,0801
LoR	74,5430 3,0460	68,7060 4,5012	65,1040 3,9399	65,2770 3,5142	72,7160 2,9120
NN_MLP	69,2940 5,1317	63,1220 5,1320	61,2790 4,0112	61,6890 4,2680	68,9800 4,2365

Complementarmente, foi analisado o desempenho dos modelos de ML para cada um dos valores possíveis da classe "intervalo_ects_realizados", cujos resultados estão apresentados na **Tabela 26**. A tabela permite constatar que os modelos de ML têm uma maior dificuldade aprender os padrões dos alunos que reprovam. Esta observação não foi surpreendente, uma vez que a mesma tendência foi identificada nas classes "risco_original" e "risco_optimizado". Esta observação pode ser indicativa de que os dados disponíveis e utilizados para o desenvolvimento deste trabalho não são suficientemente representativos das razões pelas quais os alunos reprovam, que por sua vez, levam a desempenhos abaixo do esperado.

Tabela 26 - Resultados de *holdout* por valor possível da classe “intervalo_ects_realizados”

Algoritmo	Aprova (>= 40 ECTS)			Reprova (5-39 ECTS)			0 ECTS		
	Precisão	Recall	F1 Score	Precisão	Recall	F1 Score	Precisão	Recall	F1 Score
DT	79,54	79,97	79,75	42,01	43,3	42,64	66,5	62,79	64,59
NB	83,24	83,81	83,53	47,98	36,4	41,39	65,57	83,26	73,36
RF	80,63	91,36	85,66	58,75	36,02	44,66	70,78	72,09	71,43
GB	81,92	91,36	86,38	59,65	39,08	47,22	72,85	74,88	73,85
XGB	81,89	90,53	85,99	60,87	42,91	50,34	72,56	72,56	72,56
XGBRF	80,21	93,96	86,54	64,17	29,5	40,42	70,56	75,81	73,09
LGBM	81,95	90,95	86,22	60,75	43,3	50,56	75,24	73,49	74,35
KNN	80,66	90,4	85,25	51,26	39,08	44,35	70,9	62,33	66,34
SVM	72,61	96,71	82,94	28,57	0,77	1,49	69,16	73,02	71,04
LoR	81,44	91,5	86,18	56,97	36,02	44,13	71,95	73,95	72,94
NN_MLP	81,97	82,3	82,14	45,71	49,04	47,32	70,98	63,72	67,16

Além disso, foi verificado qual seria o desempenho esperado pelos modelos de ML quando enfrentados com o cenário pretendido, predizer a classe para alunos de um novo ano letivo. A **Tabela 27** apresenta os resultados obtidos neste cenário e que, por sua vez, são muito semelhantes aos obtidos no *dataset* composto por todos os anos letivos. Desta forma, tal como para a classe “continua_estudos”, os modelos de ML para este problema demonstram um desempenho, teórico, estável para anos letivos futuros. No entanto, e em contraste com os resultados já apresentados, neste cenário o algoritmo de ML que melhor treinou o seu modelo foi o *XGBoost*, contudo realça-se que a sua diferença para *LightGBM* não é significativa.

Tabela 27 - Resultados do treino *holdout* da classe “intervalo_ects_realizados”, cujo conjunto de treino é composto pelas instâncias dos datasets dos anos letivos 2019/2020, 2020/2021, 2021/2022 e as instâncias de teste são pertencentes ao ano letivo 2022/2023

Algoritmo	Exatidão	Precisão	Recall	F1Score	F1Score (ponderada)
DT	67,42	60,66	58,46	59,35	67,44
NB	72,54	64,26	65,23	64,18	71,28
RF	74,49	69,65	61,22	63,72	72,08
GB	74,49	68,44	61,60	63,89	72,52
XGB	75,06	69,58	64,09	66,14	73,68
XGBRF	74,09	67,34	59,65	61,58	71,01
LGBM	74,74	68,90	63,82	65,79	73,54
KNN	73,27	68,26	59,81	62,54	71,22
SVM	61,98	20,66	33,33	25,51	47,43
LoR	71,73	59,69	58,06	56,24	66,80
NN_MLP	61,98	20,66	33,33	25,51	47,43

Por último, é importante destacar que um modelo de ML desta predição será utilizado num algoritmo de sinalização de nível de risco com base na predição de dois modelos. Este algoritmo de sinalização será já apresentado de seguida.

6.3. Combinação de modelos de ML explorada para a predição do nível de risco

Dado que os resultados obtidos para os modelos de ML treinados com as classes de risco ("risco_original" e "risco_otimizado") foram insatisfatórios e ficaram aquém do esperado, foi explorada a possibilidade de prever o nível de risco utilizando dois modelos de ML separados. Esta abordagem foi implementada com o objetivo de validar se o uso combinado de dois modelos de ML poderia melhorar os resultados. Além disso, o risco a ser predizido é dividido em quatro níveis: "Nenhum", "Baixo", "Médio" e "Alto".

Assim, com base nos resultados obtidos e apresentados (que também se demonstraram positivos), para a predição do nível de risco foi adotado um modelo de predição do intervalo de ECTS realizados e outro de predição da continuação dos estudos. Para ser possível integrar ambos os modelos, de forma a associar um nível de risco às suas predições, foi necessário definir um algoritmo.

Algoritmo esse que se inicia com a submissão do aluno ao modelo de predição do intervalo de ECTS realizados. Após o modelo realizar essa predição, caso seja classificado com "0 ECTS" é automaticamente atribuído o risco "Alto", enquanto se for classificado com "Aprova (>= 40 ECTS)" é lhe atribuído o nível "Nenhum". No caso de ser classificado com "Reprova (> 4 e < 40 ECTS)", será submetido ainda ao segundo modelo de ML, treinado para predizer se continuará os estudos no IPCB. Se o modelo predisser que o aluno irá continuar a estudar, a instância é classificada com o nível de risco "Baixo", caso contrário é classificada com "Médio". A **Figura 64** ilustra, em pseudocódigo, o funcionamento deste algoritmo aqui proposto.

```

Algoritmo Predição de Risco
    clf_ects:    Modelo de ML para predizer o intervalo de ECTS
    clf_continua: Modelo de ML para predizer se continua os estudos

    le(alunos_a_predizer) do ficheiro com instâncias a predizer

    Enquanto le(aluno) de alunos_a_predizer faz:
        ets_previstos = clf_ects.predizer(aluno)

        Se ets_previstos == "0 ECTS" então
            associa o nível "Alto" ao aluno
        Senão se ets_previstos == "Aprovado ( $\geq 40$  ECTS)" então
            associa o nível "Nenhum" ao aluno
        Senão
            continua_estudos = clf_continua.predizer(aluno)

            Se continua_estudos == Verdadeiro
                associa o nível "Baixo" ao aluno
            Senão
                associa o nível "Médio" ao aluno
    Fim do algoritmo

```

Figura 64 - Pseudocódigo do algoritmo de predição de nível de risco composto por dois modelos de ML

Visto que já foram apresentados os desempenhos dos vários algoritmos de ML para cada uma das predições realizadas pelos modelos de ML deste algoritmo de predição do nível, optou-se apenas pelo treino e avaliação com os melhores de cada um. Assim, para a predição do intervalo de ECTS Realizados foi utilizado o algoritmo de ML *LightGBM* e para a predição de continuação de estudos foi escolhido o algoritmo *Gradient Boosting*. Embora ambos os algoritmos selecionados não terem sido considerado os melhores na simulação de predição real, estes foram escolhidos com base no seu desempenho no *dataset* com todos os anos letivos. Além disso, os resultados demonstraram que estes apresentavam um desempenho muito próximo do melhor algoritmo na simulação.

Assim, para este sistema de sinalização, serão apresentados dois resultados de treinos realizados: um primeiro treino *holdout* (75/25) no *dataset* composto por todos os anos letivos e um segundo treino que simula a sua utilização real, cujo dados de teste são os do ano letivo 2022/2023 e os restantes dados dos anos letivos são utilizados para treino. Com estes resultados pode-se ter uma estimativa, teórica, do seu desempenho.

É importante salientar que, até o momento, os dados referentes ao ano letivo de 2023/2024 ainda não foram disponibilizados. Sendo que assim que esses forem fornecidos, será possível realizar um novo treino dos modelos de ML. Dessa forma, espera-se melhoria nos resultados preditivos, pois, haverá um aumento de exemplos (instâncias) a serem dados para o treino esperando assim que os modelos consigam fundamentar ainda mais a sua aprendizagem, ou até mesmo aprender novos padrões.

Na **Tabela 28** são apresentados os resultados obtidos para os dois treinos realizados. A sua análise permite verificar que o desempenho deste algoritmo ficou aquém do esperado, pois, procurava-se melhorar os resultados preditivos de um risco com base no uso de dois modelos de ML. Estes resultados podem ser comparados com os resultados obtidos na classe “risco_ottimizados” (**Tabela 14** e **Tabela 18**), pois, este algoritmo é uma tentativa de melhoria desses resultados. Quando comparados os resultados confirma-se que estes são muito semelhantes.

Tabela 28 - Resultados obtidos nos treinos do algoritmo de sinalização de nível de risco composto por dois modelos de ML

Tipo treino	Exatidão	Precisão	Recall	F1-Score	F1-Score (ponderado)
Com todos os dados dos anos letivos	74,2739	55,1444	52,7051	53,4464	72,6093
Simulação do cenário real	72,4614	52,2026	47,1573	48,871	70,2497

Para este algoritmo, não foi experimentado o balanceamento dos dados treino, uma vez, que se confirmou que o seu uso não melhorava o desempenho dos modelos de ML. Desta forma dado que os resultados continuaram a ser insatisfatórios e o algoritmo apresentou um desempenho semelhante a um único algoritmo de ML, foi realizado uma breve e simples otimização dos modelos que compõem o algoritmo de sinalização apresentado.

6.3.3. Otimização dos melhores modelos de ML

Um dos passos que pode ser realizado com objetivo de melhorar o desempenho dos modelos de ML é a otimização dos hiperparâmetros, cujo objetivo é ajustar os parâmetros de treino dos algoritmos de ML para encontrar uma combinação que produza o melhor resultado possível. Dado que os resultados dos algoritmos por omissão foram insatisfatórios e o algoritmo de sinalização será utilizado num cenário real, foi feita uma breve e simples otimização dos hiperparâmetros dos modelos que compõem o algoritmo referido, *Gradient Boosting* e *LightGBM*.

É importante salientar que este processo de otimização não foi investigado de uma forma detalhada, pois, uma análise detalhada é um processo moroso que poderia só por si ser trabalho para um outro projeto de investigação. No entanto, mesmo assim, e face aos resultados obtidos, procurou-se fazer uma simples otimização de hiperparâmetros, com o objetivo de melhorar ligeiramente os modelos de ML finais.

Uma vez que a otimização de hiperparâmetros é realizada através de uma procura extensa de combinações de parâmetros e a sua procura depende muito do espaço de pesquisa que por sua vez pode ser muito grande, foi optado pelo uso da pesquisa aleatória, *RandomizedSearchCV*. Com esta técnica as combinações de parâmetros serão selecionadas aleatoriamente, de um conjunto pré-definido, evitando processos de

procura muito morosos e computacionalmente intensivos. Assim com base nos parâmetros possíveis para cada os algoritmos *Gradient Boosting* e *LightGBM*, foi definido um conjunto de pesquisa de parâmetros, conforme ilustrado na **Figura 65**. Nesta figura a variável “*parameters_for_gb*” apresenta o conjunto de valores definidos para o algoritmo Gradient Boosting, e a variável “*parameters_for_lgbm*” apresenta o conjunto para o algoritmo *LightGBM*.

```

parameters_for_gb = {
    "n_estimators": [int(x) for x in np.linspace(start=100, stop=1000, num=100)],
    "loss": ["log_loss", "exponential"],
    "max_features": ["auto", "log2", "sqrt"],
    "max_depth": [int(x) for x in np.linspace(start=5, stop=100, num=5)],
    "subsample": [0.8, 1.0],
    "min_samples_split": [int(x) for x in np.linspace(start=5, stop=100, num=5)],
    "min_samples_leaf": [int(x) for x in np.linspace(start=5, stop=100, num=5)],
}

parameters_for_lgbm = {
    "n_estimators": [int(x) for x in np.linspace(start=100, stop=1000, num=100)],
    "learning_rate": [0.01, 0.05, 0.1, 0.2],
    "num_leaves": [int(x) for x in np.linspace(start=5, stop=100, num=5)],
    "max_depth": [-1] + [int(x) for x in np.linspace(start=5, stop=100, num=5)],
    "min_child_samples": [-1] + [int(x) for x in np.linspace(start=5, stop=100, num=5)],
    "subsample": [0.6, 0.8, 1.0],
    "colsample_bytree": [0.6, 0.8, 1.0],
    "reg_alpha": [0, 0.1, 0.25, 0.5],
    "reg_lambda": [0, 0.1, 0.25, 0.5],
    "boosting_type": ["gbdt", "dart"],
}

```

Figura 65 - Conjunto de pesquisa definido com os valores possíveis para cada parâmetro dos algoritmos de ML *Gradient Boosting* (*parameters_for_gb*) e *LightGBM* (*parameters_for_lgbm*)

No entanto, é importante mencionar que os valores dos hiperparâmetros foram definidos por intuição, assim, a sua definição não é fundamentada em investigações existentes ou históricas do autor. Adicionalmente, esta é a primeira vez que o autor realiza esse processo de otimização e dado que o mesmo não é devidamente fundamentado, é importante realçar que existe a possibilidade de que o conjunto de valores escolhidos não seja o ideal. Desta forma, propõem-se que em trabalhos futuros se deva explorar mais detalhadamente este processo de otimização, de forma que a definição de valores seja devidamente fundamentada e seja possível obter melhores resultados.

Complementarmente, é importante mencionar que a procura pela melhor combinação de parâmetros foi realizada com base num treino de validação cruzada de 10 *folds*. Já a métrica de avaliação utilizada para identificar a melhor combinação de parâmetros foi *F1-Score*, uma vez que é esta a que se pretende melhorar. No total, para cada algoritmo e conjunto de dados de treino (todos os anos letivos ou anos letivos de 2019/2020 a 2021/2022), foram avaliados 250 treinos diferentes. Ao calcular o

número total de modelos de ML treinados, Número de *folds* (10) x Número de iterações (250) x Número de algoritmos (2) x Número de conjuntos de treino (2), verifica-se que ao todo foram treinados 10000 modelos.

Posto isto, os melhores conjuntos de parâmetros para cada algoritmo de ML utilizados neste sistema de sinalização de risco estão ilustrados na **Figura 66**. Na parte esquerda da figura, encontram-se os parâmetros para o algoritmo *Gradient Boosting*, enquanto na parte direita estão os parâmetros de *LightGBM*. Com a visualização da **Figura 66**, é possível verificar que, para o algoritmo *Gradient Boosting*, os parâmetros “*n_estimators*” e “*min_samples_leaf*” variaram entre os dois tipos de treino realizado. Já o algoritmo LightGBM apresentou a mesma combinação de parâmetros para ambos os treinos.

Dados de treino da procura	Gradient Boosting (continua_estudos)	LightGBM (intervalo_ects_realizados)
Em 75% dos dados de todos os anos letivos	{'subsample': 1.0, 'n_estimators': 372, 'min_samples_split': 76, 'min_samples_leaf': 5, 'max_features': 'sqrt', 'max_depth': 5, 'loss': 'log_loss'}	{'subsample': 0.8, 'reg_lambda': 0, 'reg_alpha': 0.1, 'num_leaves': 5, 'n_estimators': 372, 'min_child_samples': 5, 'max_depth': 5, 'learning_rate': 0.05, 'colsample_bytree': 0.6, 'boosting_type': 'gbdt'}
Nos dados dos anos letivos 2019/2020, 2020/2021, 2021/2022	{'subsample': 1.0, 'n_estimators': 100, 'min_samples_split': 76, 'min_samples_leaf': 100, 'max_features': 'sqrt', 'max_depth': 5, 'loss': 'log_loss'}	{'subsample': 0.8, 'reg_lambda': 0, 'reg_alpha': 0.1, 'num_leaves': 5, 'n_estimators': 372, 'min_child_samples': 5, 'max_depth': 5, 'learning_rate': 0.05, 'colsample_bytree': 0.6, 'boosting_type': 'gbdt'}

Figura 66 - Melhores combinações de hiperparâmetros encontrados para o algoritmo *Gradient Boosting* e *LightGBM*

Assim, voltou-se a treinar os dois modelos de ML que compõem o algoritmo de predição proposto. Após o seu treino e integração no algoritmo, foi realizada a predição de teste (avaliação) e os resultados obtidos foram aqueles apresentados na **Tabela 29**. Ao comparar os novos resultados obtidos com os anteriores (**Tabela 28**) sem otimização, verifica-se uma ligeira melhoria, no entanto a mesma não foi muito significativa.

Tabela 29 - Resultados obtidos nos treinos do algoritmo de sinalização de nível de risco com a otimização de hiperparâmetros dos modelos de ML que o compõem

Tipo treino	Exatidão	Precisão	Recall	F1-Score	F1-Score (ponderado)
Com todos os dados dos anos letivos	75,1867	58,4117	53,8699	54,8041	72,969
Simulação do cenário real	74,736	54,6806	47,7899	49,6301	71,4965

Desta forma, após a otimização dos hiperparâmetros e confirmação de uma melhoria, ainda que reduzida, no desempenho dos modelos de ML, pode finalmente ser feita a exportação final do sistema de sinalização do nível de risco. Depois da sua exportação para um ficheiro “*.skops*”, o mesmo poderá ser carregado no momento da predição, sendo adicionalmente integrado na aplicação web desenvolvida. Desta forma, dá-se por terminado a apresentação dos resultados de treino e avaliação de modelos de ML no contexto deste trabalho.

6.4. Reflexão dos resultados obtidos

Como foi possível constatar ao longo da apresentação dos resultados obtidos, estes foram insatisfatórios e ficaram aquém das expectativas quando se tentou prever níveis de risco que contemplassem o desempenho académico dos estudantes. Para as seis classes analisadas, apenas duas, “risco_binario” e “continua_estudos”, apresentaram resultados relativamente bons, com uma exatidão em torno de 80% e *F1-Score* de 70%. Para as restantes classes, observou-se que, para alguns valores da classe, como o nível de risco “Baixo” ou intervalo de ECTS realizados “Reprova (5-39 ECTS)”, os modelos de ML não conseguiram aprender corretamente os padrões presentes nos dados.

Inicialmente, considerou-se que os resultados desfavoráveis poderiam estar associados ao desequilíbrio de representação dos valores possíveis de cada classe. Dessa forma, investigou-se o uso de algoritmos de balanceamento de dados para equilibrar o conjunto de treino, permitindo aumentar o número de instâncias para as os valores minoritários e de forma que os modelos de ML pudessem basear seus padrões em mais exemplos sintetizados. No entanto, como foi apresentado, os resultados obtidos com o balanceamento dos dados de treino mostraram-se muito semelhantes aos do treino sem balanceamento. Esta observação indica claramente que os modelos de ML não conseguiram aprender os padrões pretendidos. Dado que o balanceamento não melhorou os resultados, resta-se investigar se os mesmos poderiam ser melhorados fazendo uma a seleção detalhada e aprofundada de atributos e/ou serem experimentados treinos com base em outros atributos.

Além disso, interessa referir que numa análise independente foi experimentado o uso do algoritmo *OneR* da ferramenta *Weka*, o qual cria uma única regra (*if*, condição lógica) para classificar as instâncias. Curiosamente, este algoritmo obteve um desempenho muito próximo dos modelos de ML treinados neste trabalho. Esse desempenho pode ser explicado pelo facto de que os alunos internacionais representam a larga maioria das instâncias associadas aos níveis de risco ou ao abandono escolar. Para ilustrar esta ocorrência, são apresentadas duas figuras, **Figura 67** e **Figura 68**, onde é possível observar que a única regra criada pelo *OneR* para a classe “continua_estudos” se baseia na identificação de que se o aluno (instância) é estrangeiro. Por exemplo, essa regra pode estar associada à ausência de um distrito (**Figura 67**) ou à indicação explícita de que o aluno é um estudante internacional (**Figura 68**).

```

distrito-Desconhecido/a:
  < 0.5  -> True
  >= 0.5 -> False
(3879/4820 instances correct)

Time taken to build model: 0.07 seconds

== Stratified cross-validation ==
== Summary ==

  Correctly Classified Instances      3879          80.4772 %
  Incorrectly Classified Instances    941           19.5228 %
  Kappa statistic                   0.3841
  Mean absolute error              0.1952
  Root mean squared error          0.4418
  Relative absolute error          50.8916 %
  Root relative squared error     100.8947 %
  Total Number of Instances        4820

```

Figura 67 - Regra criada pelo algoritmo OneR para a classe "continua_estudos" e a sua exatidão

```

tipo_ingresso-CE__Estudante_Internacional:
  < 0.5  -> True
  >= 0.5 -> False
(3826/4820 instances correct)

Time taken to build model: 0.05 seconds

== Stratified cross-validation ==
== Summary ==

  Correctly Classified Instances      3826          79.3776 %
  Incorrectly Classified Instances    994           20.6224 %
  Kappa statistic                   0.4222
  Mean absolute error              0.2062
  Root mean squared error          0.4541
  Relative absolute error          53.758 %
  Root relative squared error     103.6971 %
  Total Number of Instances        4820

```

Figura 68 - Regra criada pelo algoritmo OneR para a classe "continua_estudos", e a sua exatidão, após remoção dos atributos relativos ao distrito do aluno

No entanto, após a remoção de todos os atributos relativos à nacionalidade do aluno ou da indicação de que se trata de um estudante internacional, observou-se que outros atributos ainda conseguem obter resultados próximos daqueles já apresentados. No caso da classe "continua_estudos", na **Figura 69** verifica-se que o algoritmo OneR alcançou uma exatidão de 74,4% considerando apenas a idade do aluno. Em contraste, para a classe "risco_otimizado", o algoritmo, em **Figura 70**, obteve uma exatidão de 64,5% ao basear-se unicamente no número de ECTS matriculados pelo aluno. Estas observações são indicativas da necessidade de explorar o uso de novos atributos, de forma a verificar se existem outros padrões menos comuns, mas que melhorem o desempenho dos modelos de ML.

```

idade:
  < 26.5 -> True
  < 30.5 -> False
  < 42.5 -> True
  < 43.5 -> False
  < 47.5 -> True
  < 48.5 -> False
  >= 48.5 -> True
(3618/4820 instances correct)

Time taken to build model: 0.02 seconds

== Stratified cross-validation ==
== Summary ==

Correctly Classified Instances      3586      74.3983 %
Incorrectly Classified Instances    1234      25.6017 %
Kappa statistic                      0.0929
Mean absolute error                  0.256
Root mean squared error              0.506
Relative absolute error              66.7378 %
Root relative squared error         115.5397 %
Total Number of Instances            4820

```

Figura 69 - Regra criada pelo algoritmo OneR para a classe "continua_estudos", e a sua exatidão, após remoção de qualquer atributo referente à nacionalidade ou indicativo que é um aluno internacional

```

ects_matriculados:
  < 30.25 -> Alto
  < 53.0 -> Baixo
  < 56.0 -> Alto
  < 66.75 -> Nenhum
  < 75.25 -> Baixo
  < 78.5 -> Nenhum
  < 84.25 -> Baixo
  < 86.25 -> Nenhum
  < 91.75 -> Baixo
  < 100.75       -> Nenhum
  < 103.75       -> Baixo
  < 144.5 -> Nenhum
  >= 144.5       -> Baixo
(3127/4820 instances correct)

Time taken to build model: 0.02 seconds

== Stratified cross-validation ==
== Summary ==

Correctly Classified Instances      3111      64.5436 %
Incorrectly Classified Instances    1709      35.4564 %
Kappa statistic                      0.1699
Mean absolute error                  0.1773
Root mean squared error              0.421
Relative absolute error              61.9883 %
Root relative squared error         111.3607 %
Total Number of Instances            4820

```

Figura 70 - Regra criada pelo algoritmo OneR para a classe "risco_optimizado", e a sua exatidão, após remoção de qualquer atributo referente à nacionalidade ou indicativo que é um aluno internacional

Com a apresentação dos resultados e a análise da investigação realizada por outros autores, é possível concluir que as predições binárias obtêm os melhores resultados. Embora os resultados obtidos tenham sido insatisfatórios, aqueles relativos à predição do abandono escolar (classe "continua_estudos"), realizada no momento da matrícula do aluno, coincidem com os resultados dos estudos analisados ([67], [71], [76]), onde

a métrica de exatidão varia entre 70% e 80%. Por outro lado, para as restantes classes, que representam o nível de risco, e têm em consideração o sucesso académico, não é possível fazer uma comparação direta, pois os estudos analisados não realizam esse tipo de predição. Contudo, o único estudo que realiza uma predição semelhante, segmentação dos alunos por vários grupos [77], realiza as suas predições depois do início das aulas e baseia-se em dados de desempenho já disponíveis do aluno, como as notas.

Apesar dos resultados obtidos do sistema de sinalização proposto serem insatisfatórios, é importante salientar que o mesmo é uma implementação inicial que será devidamente continuada e melhorada nos próximos anos. Contudo, mesmo que os resultados não tenham sido ideais, o seu desempenho no cenário real ainda é desconhecido, dado que ainda não foi implementado e suas predições não foram validadas. Assim, só após a sua aplicação e validação final (ao fim do ano letivo) é que será possível obter conclusões definitivas sobre a eficácia do sistema proposto.

Além disso, aquilo que foi proposto pela submedida da DGES, a predição do abandono, foi conseguido com resultados significativamente satisfatórios e dentro daquilo que são os apresentados na literatura ([67], [71], [76]). No entanto, a tentativa de englobar num nível de risco as duas realidades, abandono e sucesso escolar, é que se mostrou ser uma tarefa muito difícil para os algoritmos de ML quando lhes são apenas fornecidos os dados disponíveis da matrícula dos alunos. Desta forma, pode-se concluir ser mais vantajoso considerar as duas tarefas de forma independente. O desenvolvimento de ferramentas de acompanhamento, como o proposto no projeto REVUP, representará um papel extramente importante para a recolha de dados que poderão ser usados para a predição de diferentes aspetos relacionados com o sucesso escolar dos alunos, uma vez que permite guardar a assiduidade dos alunos, atitude em aula, resultados dos momentos de avaliação das avaliações por frequência e por exame, entre outros.

Por último, realça-se que neste trabalho não foi realizada qualquer seleção de atributos. Por isso, futuras melhorias do trabalho aqui apresentado, devem realizar a devida seleção de atributos e adicionalmente explorar o uso de outros atributos de dados. Além disso, deve ser investigado com um maior detalhe todo o processo de otimização de hiperparâmetros dos modelos de ML a serem utilizados. Isto porque, como foi possível verificar, uma simples otimização possibilitou que os modelos tivessem obtido um melhor desempenho, e assim, uma procura mais detalhada, aprofundada e fundamentada permitirá obter ainda melhores resultados.

7. Aplicações Desenvolvidas

A integração de modelos de ML em aplicações é um passo crucial para tornar a sua utilização prática e acessível. Sem uma interface gráfica, a interação com os modelos ficaria limitada à linha de comandos ou à execução de código *Python* (no contexto deste projeto), exigindo conhecimentos técnicos e de programação avançados. Desta forma, o desenvolvimento de aplicações com interfaces gráficas, que integram modelos de ML, desempenha um papel fundamental, pois permite que utilizadores sem qualquer experiência em programação ou funcionamento de ML possam interagir com os modelos de forma intuitiva e rápida, bastando apenas inserir os dados necessários para que o modelo realize as previsões.

No âmbito deste projeto, foram desenvolvidas duas aplicações distintas. A primeira foi concebida como uma prova de conceito simples, capaz de realizar previsões de alunos a partir de um ficheiro Excel ou através do preenchimento manual dos atributos do modelo. Esta pequena aplicação/protótipo, também, permitiu apresentar a ideia de produto à empresa Digitalis, a qual demonstrou grande interesse no conceito.

Na sequência da proposta do sistema apresentada pelo IPCB, a empresa concordou concretizar o seu desenvolvimento como um novo módulo na aplicação de gestão académico do IPCB (NetP@), denominado SI.PREVINA. Este novo módulo terá como objetivo servir como uma plataforma de acompanhamento aos alunos identificados com risco de poder vir a cair em situações de insucesso e/ou abandono escolar. O autor deste projeto acabou também por integrar a equipa de desenvolvimento da Digitalis, nesse novo modulo, e o trabalho por este produzido (até ao momento de escrita do relatório), será apresentado no capítulo **8. Contribuição externa** do autor na Digitalis.

Relativamente à segunda aplicação, esta é mais complexa e foi desenvolvida com o objetivo de ser utilizada pelos docentes do IPCB no contexto do projeto REVUP nos próximos anos. A decisão de desenvolver esta segunda aplicação baseou-se principalmente na possibilidade de permitir que os docentes do IPCB, ou até mesmo alunos que desejem dar continuidade ao trabalho de investigação deste projeto, possam realizar estudos de previsões de forma independente e sem realizarem alterações significativas no sistema principal de gestão académica, NetP@. No entanto, devido a limitações temporais, algumas funcionalidades, que não são essenciais para o seu uso, não foram implementadas, sendo estas descritas no subcapítulo dedicado à sua apresentação.

Posto isto, seguem-se as apresentações das duas aplicações referidas. Para cada uma, inicialmente será relembrado o seu objetivo, as funcionalidades disponíveis, a arquitetura (incluindo as ferramentas utilizadas no desenvolvimento) e, por fim, a apresentação dos diversos ecrãs que as compõem. É, no entanto, importante salientar que neste relatório não é apresentado todo o código produzido no desenvolvimento de ambas as aplicações. Em vez disso, ocasionalmente, são fornecidos alguns pequenos excertos com o objetivo de ilustrar certas funcionalidades. Esta escolha deve-se principalmente a limitações temporais na elaboração do relatório e à necessidade de

evitar um documento excessivamente extenso e detalhado, focando apenas nos aspetos mais relevantes da implementação.

7.1. Aplicação de demonstração de conceito

A primeira aplicação consiste numa única página web que permite com que o utilizador realize previsões do nível de risco para um conjunto de alunos (via ficheiro) ou para um único aluno (através do preenchimento de um formulário). Esta foi desenvolvida exclusivamente para ser apresentada à empresa Digitalis, com o intuito de demonstrar parte do produto concebido pelo projeto REVUP. A aplicação, apesar de muito simples, cumpriu com sucesso o seu objetivo, permitindo que os colaboradores da Digitalis compreendessem claramente a função das técnicas de ML (modelos preditivos) no produto concebido. Além disso, o interesse demonstrado pela Digitalis foi significativo, levando não apenas a uma melhor compreensão dos requisitos do SI.PREVINA, mas também à realização de uma proposta de emprego ao autor deste projeto, que por sua vez foi aceite.

A aplicação disponibiliza apenas duas funcionalidades. A primeira permite a previsão do risco de alunos (Risco Otimizado, ver **Figura 37**) a partir de um ficheiro Excel (.XLS ou .XLSX), e a segunda permite a previsão do risco para um único aluno. Em ambos os casos, o resultado é limitado à apresentação do nível de risco, não sendo possível descargar os resultados. Quando um ficheiro com múltiplas instâncias é fornecido, apenas é indicado o número de alunos identificados em cada nível de risco. Por outro lado, ao predizer uma única instância (através da inserção e seleção de valores nos campos de input), o risco é apresentado exclusivamente para essa instância individual. Adicionalmente, o uso desta aplicação não requer qualquer autenticação.

Interessa referir que o modelo de ML integrado nesta aplicação, aquando a sua apresentação, não foi devidamente treinado e otimizado, bem como os seus dados não foram devidamente codificados (não passaram pelo processo de pré-processamento mencionado em **5.3. Pré-processamento**). Esta decisão deveu-se ao facto de os dados utilizados possuírem erros, uma vez que eram provenientes da primeira exportação (primeira versão). Como mencionado anteriormente, as versões iniciais de extração e exportação de dados apresentavam gralhas significativas nos valores dos atributos em maioria das instâncias, o que resultou num treino inadequado e em previsões de baixa qualidade, inadequadas para um uso em produção. No entanto, para uma primeira demonstração do comportamento e da função de modelos de ML no produto à Digitalis, o desempenho foi considerado suficiente, uma vez que as previsões geradas eram automaticamente descartadas e não importavam para avaliação geral do conceito.

7.1.1 Arquitetura e ferramentas utilizadas

Sendo uma aplicação muito simples, a sua arquitetura é também muito simples. Desta forma, para o seu desenvolvimento foram unicamente utilizadas as bibliotecas

SvelteKit (Javascript e Typescript), Flask (Python), Scikit-Learn, Pandas e Skops. Dado que o código desenvolvido com a biblioteca *Flask* é muito reduzido, esta não foi mencionada nas ferramentas utilizadas neste projeto. Assim, toda a arquitetura da aplicação pode ser visualizada através da **Figura 71**.



Figura 71 - Arquitetura da primeira aplicação

Nesta arquitetura, o utilizador acede a um website que possui um único ecrã de utilização e poderá selecionar o ficheiro com as instâncias dos alunos a predizer o risco ou os valores dos atributos de um único aluno. Ao fim de submeter o seu *input*, a interface gráfica (*SvelteKit*) é responsável pelo envio dessa informação à REST API (*Flask*) que por sua vez pré-processa os dados para que o modelo consiga predizer uma classe. Com a realização desse pré-processamento é originado um objeto *DataFrame* com a(s) instância(s) a predizer e com este é realizado um pedido ao modelo de ML (*Scikit-Learn* e *Skops*). Após receber esse pedido o modelo de ML irá realizar a predição de risco para cada instância e ao fim de realizar todas as predições devolve o valor da classe associada a cada uma. Por fim este retorna o resultado à resposta à *REST API* que por sua vez envia para a interface gráfica.

Para o desenvolvimento do código desta aplicação foi unicamente utilizada a ferramenta *Visual Studio Code*. Dando-se a mesma situação que o uso da biblioteca *Flask*, uma vez que o seu uso foi consideravelmente reduzido comparativamente às restantes ferramentas, este não foi mencionado na apresentação das ferramentas utilizadas. Adicionalmente, é importante salientar que todo o código desenvolvido está disponível num repositório GitHub do autor. No entanto, devido à natureza dos dados e do segredo de negócio do produto, o repositório não é de código aberto, ou seja, só o autor possui acesso de leitura e escrita ao mesmo.

7.1.2. Ecrãs da aplicação

Como já mencionado a interface da aplicação é constituída unicamente por um ecrã, sendo este apresentado na **Figura 72**. Ao entrar na aplicação web, o utilizador poderá inicialmente visualizar o título, “Classificador RevUp”, e logo em baixo a indicação de que a interface é referente a um protótipo de desenvolvimento (demonstração). Adicionalmente, é acompanhado da data atual para eventuais capturas de ecrã que possam vir a ser tiradas.

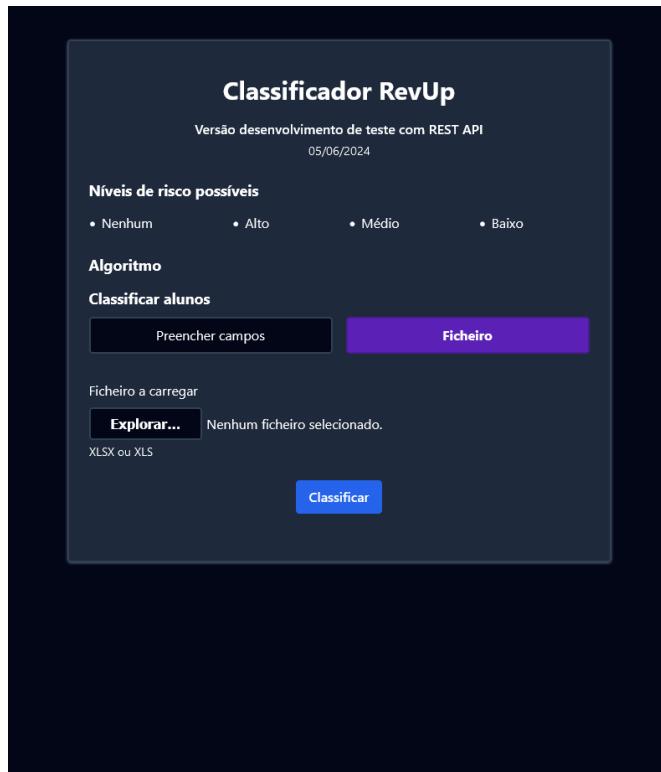


Figura 72 - Único ecrã da primeira aplicação, opção de seleção de um ficheiro

Nesta pequena página, o utilizador visualiza também quais são os valores possíveis a serem predizidos pelo modelo de ML. Esses podem ser encontrados por baixo do subtítulo “Níveis de risco possíveis” da **Figura 72**. Em seguida, o utilizador dispõe de duas opções, a primeira “Preencher campos” que irá apresentar todos os campos do modelo para uma predição manual, e a segunda “Ficheiro” (selecionada por omissão) que permite a escolha de um ficheiro Excel com as instâncias a predizer.

Nesta primeira aplicação web desenvolvida não são mencionadas quais as colunas obrigatórias do ficheiro, ao contrário da segunda aplicação que já possui essa funcionalidade. Desta forma, apenas um utilizador informado conseguirá selecionar um ficheiro válido, pois, é possível selecionar um ficheiro não suportado e tentar classificar. Este caso resultará em erro, mas não será dado nenhum feedback visual de erro ou sucesso.

Quando selecionado um ficheiro o utilizador poderá consultar o nome do mesmo ao lado do botão “Explorar” como é apresentado na **Figura 73**. Após a seleção, o utilizador poderá carregar no botão “Classificar” que por sua vez enviará o ficheiro para a *REST API* que irá pré-processar o mesmo e dar ao modelo de ML para predizer.



Figura 73 - Primeira aplicação, utilizador escolhe um ficheiro

Após carregar no botão “Classificar” o utilizador necessita de aguardar que a *REST API* retorne uma resposta, tipicamente demorando menos de 1 segundo. Com a receção dos dados da *REST API*, são apresentados os resultados de predição, resultando no estado do ecrã visível na **Figura 74**.

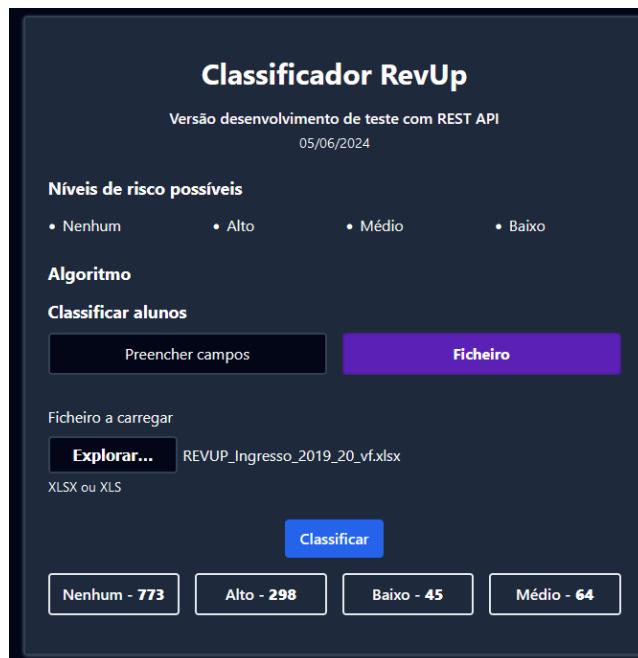


Figura 74- Primeira aplicação, apresentação resultados de predição de um ficheiro

Dado a que os resultados do modelo são de baixa qualidade e são apenas demonstrativos do funcionamento de interação com o modelo de ML, não é possível visualizar o valor associado a cada instância, mas sim a contagem de instâncias associadas a cada valor da classe predita. No entanto, a resposta da *REST API* é composta por cada predição, ou seja, quando consultado o valor de retorno do pedido *HTTP*, podem ser visualizados os diferentes riscos para cada instância (índice), ver **Figura 75**.

Todo o código desenvolvido para realizar o pedido *HTTP* de predição de um ficheiro e para apresentar a contagem de classificações para risco predito, é ilustrado na **Figura 76**. As linhas 97 a 103 correspondem ao envio do pedido *HTTP POST* (através do uso da função “fetch” de Javascript) para a rota “/file” da *REST API*, enquanto as linhas 105 a 113 tratam da obtenção do resultado em formato *JSON* (linha 105) da resposta e da contagem de instâncias por cada valor de risco gerado pelo modelo. Esta contagem é realizada utilizando um mapa, onde a chave é o nome do risco e o valor é a contagem de ocorrências, que é gradualmente incrementada conforme cada valor é visto na

iteração dos resultados retornados (ver **Figura 75**). É importante notar que, caso o modelo não identifique qualquer ocorrência para um risco específico, este não será incluído nos resultados apresentados. Por exemplo, se as instâncias forem classificadas unicamente com “Nenhum”, “Alto” e “Médio”, o valor Baixo não será apresentado nos resultados.

Figura 75 - Primeira aplicação, conteúdo da resposta de um pedido *HTTP (POST)* à *REST API* para realizar uma predição no contexto de envio de um ficheiro

```
93 let predictedRisksCounts: { [key: string]: number } = {};
94 async function submitFile(ev: SubmitEvent) {
95     const formElement = ev.target;
96
97     const formData = new FormData(formElement as HTMLFormElement);
98     console.log(formData);
99
100    const res = await fetch("http://127.0.0.1:5000/file", {
101        method: "POST",
102        body: formData,
103    });
104
105    const resData = await res.json();
106    predictedRisksCounts = {}
107    for (let predict of resData) {
108        if (Object.keys(predictedRisksCounts).includes(predict)) {
109            predictedRisksCounts[predict] += 1;
110        } else {
111            predictedRisksCounts[predict] = 1;
112        }
113    }
114 }
```

Figura 76 - Primeira aplicação, código *TypeScript* que permite realizar o pedido de predição e contar o número de instâncias que foi atribuído cada risco possível

O código responsável pela apresentação dos resultados, conforme ilustrado na **Figura 74**, é apresentado na **Figura 77**. Este código apresenta definição de elementos HTML e ainda código de criação dinâmica de elementos da biblioteca *SvelteKit*, que permite a criação dinâmica de elementos HTML para cada registo presentes no mapa

“predictedRiskCounts”. Quando este mapa está vazio (sem contagens de instâncias para cada risco), o *SvelteKit* é eficiente o suficiente para não gerar nenhum componente HTML. No entanto, quando o mapa contém elementos, o *SvelteKit* itera sobre cada entrada e apresenta um elemento “div” correspondente para cada valor presente. No contexto da **Figura 77**, o valor de “n” corresponde ao nome do nível de risco (classe) e o “c” a contagem de instâncias atribuídas com esse valor.

```

    </button>
    <div class="grid grid-cols-4 gap-4 font-semibold mt-4">
      {#each Object.entries(predictedRisksCounts) as [n, c]}
        <div class="py-2 text-center border-2 rounded">
          { n } - <b>{ c }</b>
        </div>
      {/each}
    </div>
  
```

Figura 77 - Primeira aplicação, código HTML/*SvelteKit* que cria dinamicamente os elementos HTML (“div”) com a contagem de instâncias para cada classe de risco

Como já referido, o utilizador dispõe da possibilidade de realizar a predição para um único aluno. Para este caso, em vez de o utilizador estar a criar um ficheiro com os atributos e valores de uma única instância, poderá apenas preencher campos de um formulário. Esse formulário será apresentado assim que o mesmo carrega no botão “Preencher campos”, como se pode verificar através da **Figura 78**. Após o preenchimento dos diferentes campos o utilizador poderá carregar no botão “Classificar” que por sua vez irá realizar um pedido *POST HTTP* à *REST API* para que o modelo de ML realize uma única predição. Quando efetuado o pedido de predição, que por costume demora menos de um segundo a ser realizado, aparecerá o nível de risco predito por baixo do botão “Classificar”, como é exemplificado na **Figura 79**.

Classificador RevUp

Versão desenvolvimento de teste com REST API
05/06/2024

Níveis de risco possíveis

• Nenhum • Alto • Médio • Baixo

Algoritmo

Classificar aluno

Preencher campos **Ficheiro**

escola
Escola Superior Agrária de Castelo Branco

curso
Licenciatura em Agronomia

etc_s_matriculados

e_feminino ■

forma_ingresso
CE - CTeSP

nacionalidade
Portugal

naturalidade
Brasil

desc_codigo_postal
CASTELO NOVO

idade

regime_estudo
Tempo inteiro

habilidade_mae
Licenciatura (Pré-Bolonha)

habilidade_pai
Licenciatura (Pré-Bolonha)

situacao_profissional_mae
Reformado/a

situacao_profissional_pai
Reformado/a

grupo_profissional_mae
Técnicos e Profissionais de Nível Intermédio

grupo_profissional_pai
Outra situação

estudante_atleta_ipcb ■

estudante_dirigente_ae ■

estudante_nec_especial ■

estudante_internacional ■

estudante_pai_mae ■

estudante_parcial_20-5 ■

estudante_parcial_30-5 ■

estudante_parcial_40-5 ■

estudante_trabalhador ■

estudante_normal ■

estudante_bombeiro ■

estudante_borseiro ■

Classificar

Figura 78 - Primeira aplicação, apresentação do formulário com todos os campos necessário para predizer o risco de um único aluno

Classificador RevUp

Versão desenvolvimento de teste com REST API
05/06/2024

Níveis de risco possíveis

- Nenhum
- Alto
- Médio
- Baixo

Algoritmo

Classificar aluno

Preencher campos Ficheiro

escola: Escola Superior de Tecnologia de Castelo Branco

curso: Licenciatura em Tecnologias da Informação e Multimédia

etc_s_matriculados: 60

e_feminino:

forma_ingresso: CE - CTeSP

nacionalidade: Portugal

naturalidade: Fundão

desc_codigo_postal: FUNDÃO

idade: 20

regime_estudo: Tempo inteiro

habilidade_mae: 12º ano de escolaridade

habilidade_pai: 12º ano de escolaridade

situacao_profissional_mae: Trabalha por conta própria - (como empregador)

situacao_profissional_pai: Trabalha por conta de outrém

grupo_profissional_mae: Outra situação

grupo_profissional_pai: Outra situação

estudante_atleta_ipcb:

estudante_dirigente_ae:

estudante_nec_especial:

estudante_internacional:

estudante_pai_mae:

estudante_parcial_20-5:

estudante_parcial_30-5:

estudante_parcial_40-5:

estudante_trabalhador:

estudante_normal:

estudante_bombeiro:

estudante_bolseiro:

Classificar

Aluno classificado com risco: Nenhum

Figura 79 - Primeira aplicação, resultado da realização de uma predição ao preencher o formulário com os dados de um aluno

Dado que os modelos de ML são treinados com um conjunto específico de atributos e seus respetivos valores, não é recomendável criar manualmente os campos do formulário. Em vez disso, esses campos podem e devem ser gerados de forma dinâmica. Para permitir a geração dinâmica, foi desenvolvida uma funcionalidade capaz de gerar um ficheiro de configuração em formato *JSON* que contém toda a informação sobre cada atributo utilizado para o treino de um modelo. Neste ficheiro de configuração gerado, uma das chaves, denominada por “*features*”, contém um mapa que associa o nome de cada atributo (coluna do *DataFrame* utilizado para treino) aos valores aceites pelo modelo de ML. A **Figura 80** apresenta uma parte do ficheiro de configuração gerado referente ao modelo integrado nesta aplicação, onde é possível visualizar os nomes dos diferentes atributos (chaves do mapa “*features*”) e os valores possíveis para cada um deles.

```

{
    "target_classes": [...],
    "features": {
        "escola": [...],
        "curso": [...],
        "etc_s_matriculados": "float",
        "e_feminino": "bool",
        "forma_ingresso": [...],
        "nacionalidade": [...],
        "naturalidade": [...],
        "desc_codigo_postal": [...],
        "idade": "integer",
        "regime_estudo": [...],
        "habilidade_mae": [...],
        "habilidade_pai": [
            "Licenciatura (Pr\u00e1-Bolonha)",
            "Ensino b\u00f3lico 1.\u00ba ciclo (4\u00aa classe)",
            "12\u00ba ano de escolaridade",
            "DESCONHECIDO",
            "Ensino b\u00f3lico 2.\u00ba ciclo (6\u00ba ano)",
            "Ensino b\u00f3lico 3.\u00ba ciclo (9\u00ba ano)",
            "Outra",
            "Bacharelato",
            "Licenciatura (Bolonha)",
            "Doutoramento",
            "Sabe ler sem possuir a 4\u00aa classe",
            "Ensino M\u00e1dio (11\u00ba ano)",
            "Mestrado (Pr\u00e1-Bolonha)",
            "CTesP",
            "Mestrado (Bolonha)",
            "P\u00f3s-Gradua\u00e7\u00e3o",
            "- Outra -",
            "N\u00fao sabe ler nem escrever",
            "CET"
        ],
        "situacao_profissional_mae": [...],
        "situacao_profissional_pai": [...],
        "grupo_profissional_mae": [...],
        "grupo_profissional_pai": [...],
        "estudante_atleta_ipcb": "bool",
        "estudante_dirigente_ae": "bool",
        "estudante_nec_especial": "bool"
    }
}

```

Figura 80 - Primeira aplicação, parte da configuração gerada e exportada em formato *JSON* referente ao treino do modelo de ML integrado

No ficheiro de configuração (ver **Figura 80**), quando o valor de uma chave referente a um atributo é um *array*, indica que o campo correspondente deve ser um elemento do tipo “*select*”. Já quando o valor é uma *string*, esta define o tipo de dado esperado, podendo ser flutuante (*float*), inteiro (*integer*) ou booleano (*bool*). Dessa forma, ao

fornecer esta configuração ao *SvelteKit*, que tem a capacidade de interpretar ficheiros *JSON*, torna-se possível iterar sobre cada atributo existente e gerar automaticamente o elemento *HTML* de input associado a ele. Todo o código *HTML* (com *SvelteKit*) que permite a criação de elementos de forma dinâmica é apresentado na **Figura 81**.



```

{#each Object.entries(features) as [name, value]}
  {#if value === "integer"}
    <div class="mt-2">
      <label for="input_{name}">{name}</label>
      <input class="w-full rounded border text-black px-2 py-1" type="number" step="1" name="input_{name}" id="input_{name}" min="16" required>
    </div>
  {:#else if value == "float"}
    <div class="mt-2">
      <label for="input_{name}">{name}</label>
      <input class="w-full rounded border text-black px-2 py-1" type="number" step="0.5" name="input_{name}" id="input_{name}" required>
    </div>
  {:#else if value == "bool"}
    <div class="mt-2">
      <label for="input_{name}">{name}</label>
      <input class="rounded border text-black px-2 py-1" type="checkbox" name="input_tipo_estudante" value={name} id="input_{name}" required>
    </div>
  {:#else}
    <div class="mt-2">
      <label for="input_{name}">{name}</label>
      <select class="block w-full px-2 py-1 border text-black" name="input_{name}" id="input_{name}" required>
        {#each value as possible_value}
          <option value={possible_value}>{possible_value}</option>
        {/each}
      </select>
    </div>
  {/if}
{/each}

```

Figura 81 - Primeira aplicação, código *HTML* (com *SvelteKit*) que gera automaticamente elementos de *input HTML* para cada atributo especificado no ficheiro de configuração do modelo de ML integrado

Embora nesta aplicação tenha sido integrado apenas um modelo de ML, o desenvolvimento das funcionalidades de exportação de configurações e geração de elementos HTML serviram de base para o desenvolvimento da segunda aplicação web. Esta, que será apresentada a seguir, é mais complexa e suporta múltiplos modelos de ML, cujo cada um possui o seu próprio conjunto de atributos e valores específicos.

7.2. Aplicação para apoio a investigação em ML

A segunda aplicação desenvolvida no contexto deste projeto é uma aplicação web que permite aos docentes do IPCB, nomeadamente aqueles que trabalham em investigação relacionada com ML, adicionar diferentes predições, processos de pré-processamento e modelos de ML. Esta aplicação, possibilita a interação com os modelos preditivos adicionados e consultar e descarregar as diferentes predições realizadas. Apesar de ter sido desenvolvido com foco especial no projeto REVUP, e na predição de níveis de risco de abandono e sucesso escolar, complementarmente, funciona, também, como uma aplicação independente onde podem ser realizadas predições de trabalhos de investigação independentes que não se pretendem integrar no módulo desenvolvido pela Digitalis.

A aplicação foi projetada com modularidade em mente de forma a facilitar a reutilização da mesma em diferentes contextos, exigindo apenas pequenas alterações, como o título e a descrição da aplicação. Dessa forma, qualquer aluno do IPCB, docente ou entidade pode adotar, reutilizar e/ou modificar a aplicação conforme necessário. No

entanto, apesar do autor apoiar soluções de código aberto, devido à sensibilidade dos dados e informações presentes nesta aplicação, os repositórios do GitHub com todo o código da aplicação não estão visíveis publicamente, sendo o autor o único com acesso aos mesmos. Contudo, estes repositórios podem ser disponibilizados a interessados em continuar o projeto ou reutilizar a aplicação em outros contextos, desde que possuam interesse, conhecimento e capacidade de trabalhar com as tecnologias utilizadas.

Além da aplicação web desenvolvida, foram criados diversos scripts em *Python* que, de forma modular, permitem a especificação de processos de pré-processamento (como o apresentado na **Figura 26** do capítulo **5.3. Pré-processamento**), bem como a exportação da configuração para reutilização no processo de pré-processamento antes da predição (não treino) e treino (sem seleção de atributos ou otimização de hiperparâmetros) de diferentes algoritmos de ML. Todo o código destes *scripts* está, também, armazenado em um repositório *Git* no *GitHub* do autor, que, devido à sensibilidade dos dados, só o mesmo possui acesso. No entanto, e tal como a aplicação web, o autor tem interesse em continuar e concluir o desenvolvimento, com o objetivo de publicar versões genéricas e de código aberto no *GitHub*. Essas versões não estarão associadas a nenhum contexto específico e permitirá que outros investigadores da área de ML as utilizem em seus projetos e/ou provas de conceito.

Sendo o foco a modularidade, foi necessário criar classes em *Python* para executar diferentes tarefas. A codificação dessas classes foi essencial para o desenvolvimento de uma solução que possibilitasse a criação, exportação e reutilização de processos de pré-processamento. Conforme apresentado no capítulo **5.3. Pré-processamento**, um processo desse tipo pode ser dividido em várias etapas, permitindo o desenvolvimento de funções específicas para cada uma delas.

O autor tomou a liberdade de desenvolver do zero toda uma solução modular, capaz de pré-processar *datasets* e exportar todo o processo aplicado (no formato *JSON*) para ser reutilizado em futuras predições. Esta abordagem foi praticamente obrigatória, já que, no contexto do pré-processamento de dados de alunos, os dados variam de ano para ano. Por exemplo, novos valores de atributos podem surgir, como no caso da nacionalidade, onde nem todos os países possíveis podem estar representados no *dataset* de treino. Complementarmente, os pré-processadores conseguem associar apenas valores já observados e tratados previamente, o que significa que novos valores não terão uma associação direta. No entanto, a solução desenvolvida pelo autor foca-se em resolver esta mesma limitação, permitindo que valores nunca antes observados sejam associados a um novo valor, como por exemplo "Desconhecido", ou simplesmente descartados (não é feita nenhuma associação, mas a instância pode ser predita), garantindo assim a flexibilidade do processo de pré-processamento e garantindo que o mesmo funciona para dados de anos subsequentes.

No total foram desenvolvidas 19 classes em *Python*, das quais quatro são abstratas. De forma a ser visualizado as relações de herança e dependência (uso) entre as classes,

foi elaborado um gráfico semelhante a um diagrama de classes, conforme ilustrado na **Figura 82**. Contudo, devido à sua complexidade e dimensão, o diagrama não inclui os métodos e variáveis de classe, uma vez que a sua inclusão dificultaria ainda mais a sua leitura e o autor não achou pertinente colocar.

Com o objetivo de simplificar a visualização dos diferentes processos a quais estas classes são aplicadas, o diagrama com todas classes foi repartido em dois diagramas mais pequenos. A **Figura 83** ilustra as classes de pré-processamento no contexto de treino de modelos de ML, enquanto a **Figura 86** representa as classes relativas ao pré-processamento anterior à predição e do processo responsável pela execução da predição (comunicação com o modelo de ML). Uma vez que uma explicação detalhada de ambos estes dois contextos, treino e predição, acabaria por ser muito extenso, o autor optou por uma apresentação mais resumida de cada um. Assim, para cada contexto é realizada uma breve explicação da sua importância e uso, sendo acompanhada por uma tabela descriptiva da função de cada classe e excertos de código ou apresentação das exportações.

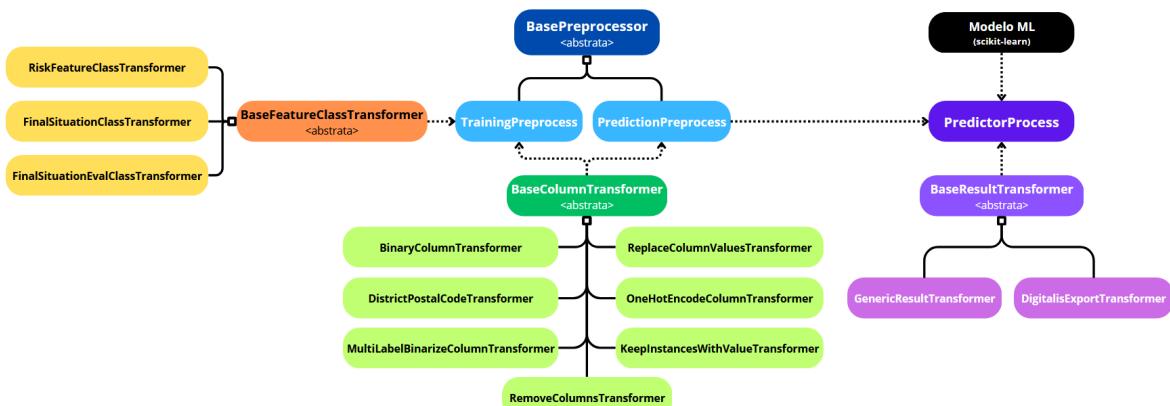


Figura 82 - Mapa com todas as classes criadas para processos de pré-processamento e predição, inclui herança e dependências

Iniciando pelo contexto de treino (ver **Figura 83**), foram programadas 14 classes, das quais 9 também estão presentes no contexto de predição (como a classe "*BasePreprocessor*" e todas aquelas com fundo verde). Para conseguir a solução modular pretendida capaz de pré-processar qualquer *dataset*, independentemente do problema de ML que se pretende resolver foi criada a classe "*TrainingPreprocess*" (que estende a classe "*BasePreprocessor*"), capaz de aceitar diferentes transformadores de atributos e/ou geradores de novas classes para predição.

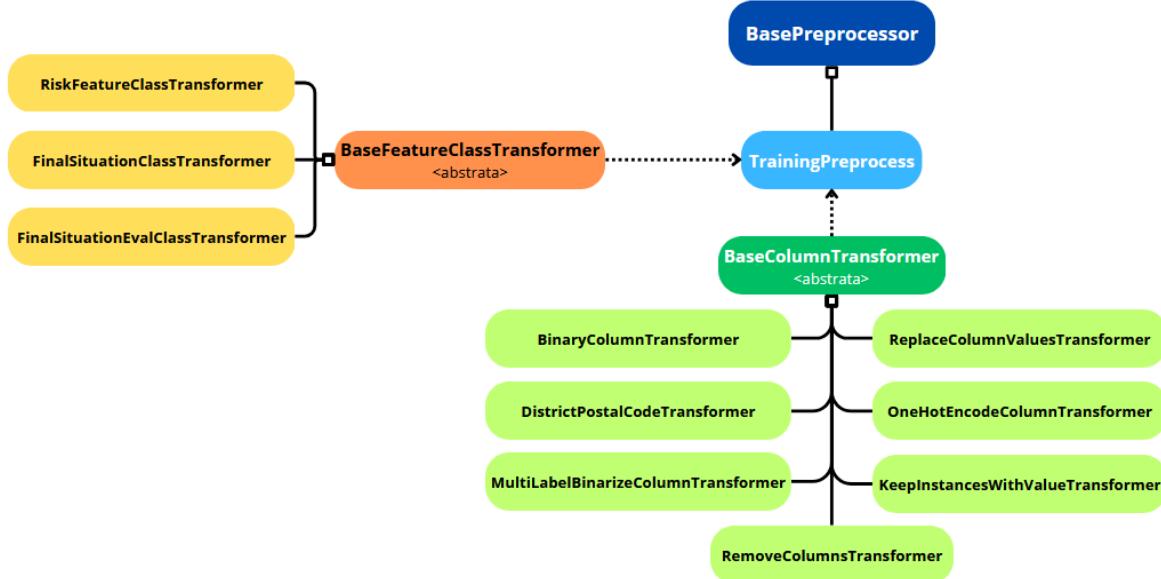


Figura 83 - Mapa de classes para o contexto de pré-processamento de treino

O processo de pré-processamento de treino pode ser configurado através do uso da classe “*TrainingPreprocess*”, para a qual o programador necessita de definir vários parâmetros importantes. Estes incluem a seleção dos *datasets* a carregar, *datasets* a combinar (opcional), valor nominal a substituir os valores nulos, lista com nome de atributos a remover antes de iniciar o pré-processamento, lista com os diferentes transformadores de atributos (a sua ordem de definição importa), o codificador para todos os restantes atributos nominais (a fim de serem aplicados os transformadores individuais), bem como a definição de uma lista com transformadores responsáveis pela criação de novas classes. Toda esta configuração é realizada através de código *Python*, cuja estrutura será equivalente ao exemplo ilustrado na **Figura 84**, que reflete a configuração do pré-processamento apresentada na **Figura 26**.

```

all_v4_config = TrainingPreprocess(data_1920_v4, training_preprocess_config,
    dataframes_to_append=[data_2021_v4, data_2122_v4, data_2223_v4],
    fill_na_value="Desconhecido/a",
    columns_to_drop=["ects_aprovados", "ects_creditados", "ects_reprovados", "ano_conclusao_habilitacao_anterior",
                      "curso_anterior", "instituicao_anterior", "nota_acesso"],
    columns_transformers=[

        DistrictPostalCodeTransformer(col_name: "codigo_postal", new_column_name: "distrito"),
        BinaryColumnTransformer(col_name: "continua_estudos", true_value="s", is_class=True),
        BinaryColumnTransformer(col_name: "sexo", true_value="f", new_column_name="e_feminino"),
        MultiLabelBinarizeColumnTransformer(col_name: "tipos_aluno", value_map=STUDENT_TYPE_MAP, cols_prefix="tipo_aluno"),
        ReplaceColumnValuesTransformer.from_json_file(col_name: "curso", file_path: "maps/courses_value_map.json"),
        ReplaceColumnValuesTransformer.from_json_file(col_name: "nacionalidade", file_path: "maps/country_value_map.json"),
        ReplaceColumnValuesTransformer.from_json_file(col_name: "naturalidade", file_path: "maps/country_value_map.json"),
        ReplaceColumnValuesTransformer.from_json_file(col_name: "habilitacao_academica_pai",
                                                       file_path: "maps/academic_qualifications_typo_map.json"),
        ReplaceColumnValuesTransformer.from_json_file(col_name: "habilitacao_academica_mae",
                                                       file_path: "maps/academic_qualifications_typo_map.json"),
        ReplaceColumnValuesTransformer.from_json_file(col_name: "habilitacao_academica_pai",
                                                       file_path: "maps/parents_qualifications_value_map.json", int,
                                                       save_map_as_metadata: True),
        ReplaceColumnValuesTransformer.from_json_file(col_name: "habilitacao_academica_mae",
                                                       file_path: "maps/parents_qualifications_value_map.json", int,
                                                       save_map_as_metadata: True),
        ReplaceColumnValuesTransformer.from_json_file(col_name: "grupo_profissional_pai",
                                                       file_path: "maps/parents_professional_group_value_map.json"),
        ReplaceColumnValuesTransformer.from_json_file(col_name: "grupo_profissional_mae",
                                                       file_path: "maps/parents_professional_group_value_map.json"),
        OneHotEncodeColumnTransformer(col_name: "nacionalidade", min_frequency=15),
        OneHotEncodeColumnTransformer(col_name: "naturalidade", min_frequency=15),
    ],
    final_nominal_transformer=OneHotEncodeColumnTransformer(col_name="", min_frequency=5),
    features_to_classes_transformers=[

        RiskFeatureClassTransformer(col_name: "risco_v1", version: 1),
        RiskFeatureClassTransformer(col_name: "risco_v2", version: 2),
        RiskFeatureClassTransformer(col_name: "risco_v3", version: 3),
        FinalSituationClassFeatureClassTransformer("situacao_final"),
        FinalSituationClassFeatureClassTransformer("sf_aval_modelo"),
    ],
    verbose=True
)

```

Figura 84 - Exemplo de uma configuração de pré-processamento de treino com todos os anos letivos

Os transformadores são os elementos mais importantes de uma configuração, pois sem eles o processo não poderia ser executado. Para clarificar o papel de cada classe neste contexto, foi criada a **Tabela 30**, que apresenta uma explicação simples da função ou tarefa de cada classe. Para facilitar a distinção entre as classes abstratas e as classes que as implementam, os nomes das classes abstratas foram destacados a negrito. Adicionalmente, as classes que estendem (herdam) o comportamento das abstratas, o seu nome é acompanhado da indicação de qual estende por baixo.

Tabela 30 - Funcionalidade das classes pertencentes ao contexto de treino

Nome da classe	Função
<i>BasePreprocessor</i>	Implementa o comportamento comum de ambos os processos de pré-processamento (treino e predição), declara variáveis comuns e declara funções abstratas
<i>TrainingPreprocess</i> - Herda <i>BasePreprocessor</i>	Responsável por juntar <i>datasets</i> , renomear e remover colunas, aplicar transformadores de colunas e exportar configuração e <i>metadata</i>
<i>BaseColumnTransformer</i>	Implementa o comportamento comum a todos os transformadores de atributos
<i>BaseFeatureClassTransformer</i>	Implementa o comportamento comum a todos os transformadores criados de novas classes no dataset
<i>BinaryColumnTransformer</i> - Herda <i>BaseColumnTransformer</i>	Transforma os valores de um atributo em booleano, sendo necessário especificar qual a associar verdadeiro enquanto os restantes são atribuídos o valor falso
<i>DistrictPostalCodeTransformer</i> - Herda <i>BaseColumnTransformer</i>	Cria uma nova coluna, Distrito, atribui o nome do distrito com base nos primeiros quatro dígitos do código postal
<i>MultiLabelBinarizeColumnTransformer</i> - Herda <i>BaseColumnTransformer</i>	Para um atributo com múltiplos valores para a mesma instância, divide os mesmos em atributos booleanos indicadores se a instância possui ou não esse valor (caso dos tipos de aluno)
<i>RemoveColumnsTransformer</i> - Herda <i>BaseColumnTransformer</i>	Permite remover atributos a meio do pré-processamento
<i>ReplaceColumnValuesTransformer</i> - Herda <i>BaseColumnTransformer</i>	Para um dado mapa de chave-valor, os valores originais do atributo serão substituídos pelos valores da chave do mapa (denominação da chave é igual ao valor original)
<i>OneHotEncodeColumnTransformer</i> - Herda <i>BaseColumnTransformer</i>	Realiza a codificação <i>One Hot Encoding</i> . Manipula de forma controlada o treino e transformação do algoritmo de codificação (<i>OneHotEncoder</i>) da biblioteca <i>Scikit-learn</i>
<i>KeepInstancesWithValueTransformer</i> - Herda <i>BaseColumnTransformer</i>	Remove todas as instâncias cujo valor do atributo definido não pertence a um conjunto de valores definidos (caso de apenas querer as instâncias de cursos piloto)
<i>RiskFeatureClassTransformer</i> - Herda <i>BaseFeatureClassTransformer</i>	Criar uma nova classe representativa do nível de risco do aluno, cujo valor é computado com base nos valores de outros atributos de cada instância. Possui versões diferentes, sendo necessário indicar qual a pretendida
<i>FinalSituationClassTransformer</i> - Herda <i>BaseFeatureClassTransformer</i>	Cria uma nova classe indicativa da situação final / intervalo de ECTS realizados – Aprova, Reprova ou 0 ECTS. Baseia-se no valor de ECTS Realizados
<i>FinalSituationEvalClassTransformer</i> - Herda <i>BaseFeatureClassTransformer</i>	Idêntico à classe (“ <i>FinalSituationClassTransformer</i> ”), no entanto considera os alunos que desistem ou não no caso de reprovarem (Reprova com renovação e Reprova sem renovação)

Desta forma, utilizando as classes previamente apresentadas, o programador pode configurar e executar um processo de pré-processamento no contexto de treino. Após a sua execução, o *dataset* pré-processado será guardado localmente, ficando disponível para ser utilizado por um algoritmo de ML para treinar o seu modelo preditivo. Além disso, é possível exportar tanto a configuração do pré-processamento como a respetiva *metadata* de pré-processamento. Esta exportação é crucial, pois permitirá definir qual é a predição a ser realizada na aplicação web implementada, assim como os diferentes passos que o processo de pré-processamento no contexto de predição deverá seguir.

A configuração exportada incluirá todos os valores observados por cada transformador de atributos, os mapas de substituição de valores e as instruções de como lidar com valores não observados. Esses valores podem ser substituídos por um valor predefinido (como "Desconhecido/a") ou descartados completamente. Quando a configuração é exportada para um ficheiro *JSON*, ela segue uma estrutura semelhante à ilustrada na **Figura 85**, facilitando a reutilização em futuros processos de pré-processamento de contexto de treino e/ou predição.

```
{
  "columns_rename_map": {
    "DS_INSTITUIÇÃO": "escola",
    "NM_CURSO": "curso",
    "Total ECTS Matriculados": "ects_matriculados",
    "INGRESSO": "tipo_ingresso",
    "SEXO": "sexo",
    "NACIONALIDADE": "nacionalidade",
    "NATURALIDADE": "naturalidade",
    "CD_POSTAL": "codigo_postal",
    "IDADE": "idade",
    "TIPOS_ALUNO": "tipos_aluno",
    "REGIME": "regime_estudo",
    "DS_HABILIT_PAÍ": "habilitacao_academica_pai",
    "DS_HABILIT_MAE": "habilitacao_academica_mae",
    "SIT_PROF_PAÍ": "situacao_profissional_pai",
    "SIT_PROF_MAE": "situacao_profissional_mae",
    "GRUPO_PROF_PAÍ": "grupo_profissional_pai",
    "GRUPO_PROF_MAE": "grupo_profissional_mae",
    "Habilitacal Anterior": "habilitacao_anterior",
    "País Ens. Secund.": "pais_ensino_secundario"
  },
  "fill_na_value": "Desconhecido/a",
  "columns_transformers": [
    {"python_class": "DistrictPostalCodeTransformer"}, {
      "python_class": "BinaryColumnTransformer",
      "col_name": "sexo",
      "new_column_name": "e_feminino",
      "is_class": false,
      "true_value": "F"
    },
    {"python_class": "MultiLabelBinarizeColumnTransformer"}, {
      "python_class": "ReplaceColumnValuesTransformer"}, {
        "python_class": "ReplaceColumnValuesTransformer",
        "col_name": "habilitacao_academica_pai",
        "new_column_name": "habilitacao_academica_pai",
        "is_class": false,
        "values_map": {
          "Outra": 0,
          "Não sabe ler nem escrever": 1,
          "Sabe ler sem possuir a 4ª classe": 2,
          "Ensino básico 1.º ciclo (4º classe)": 3,
          "Ensino básico 2.º ciclo (6º ano)": 4,
          "Ensino básico 3.º ciclo (9º ano)": 5,
          "Ensino Médio (11º ano)": 6,
          "12º ano de escolaridade": 7,
          "CET": 8,
          "CTeSP": 9,
          "Bacharelato": 10,
          "Licenciatura (Pré-Bolonha)": 11,
        }
      }
    }
  ]
}
```

Figura 85 - Parte do ficheiro de configuração resultante da execução do processo de pré-processamento de treino. Esta configuração é posteriormente importada pelo processo de pré-processamento no contexto de predição

Uma vez que a solução é modular, pode-se criar diversos processos de pré-processamento que tratam diferentes conjuntos de atributos. Sendo possível reutilizar os pré-processamentos que resultam sempre no mesmo *output* em momentos temporais diferentes, evita-se a ocorrência de erros relacionados a valores ou atributos desconhecidos quando os dados são fornecidos ao modelo de ML.

Para se poder reutilizar um processo de pré-processamento no contexto de previsões, foram desenvolvidas cinco novas classes que lidam especificamente com esse cenário. O diagrama que ilustra essa arquitetura é ilustrado na **Figura 86**. Destaca-se que as classes responsáveis pelos transformadores de atributos, já apresentadas na **Figura 83** e explicadas na **Tabela 30**, também são utilizadas neste contexto de

predição. Porém, por razões óbvias, não foram incluídas novamente no diagrama nem serão explicadas novamente.

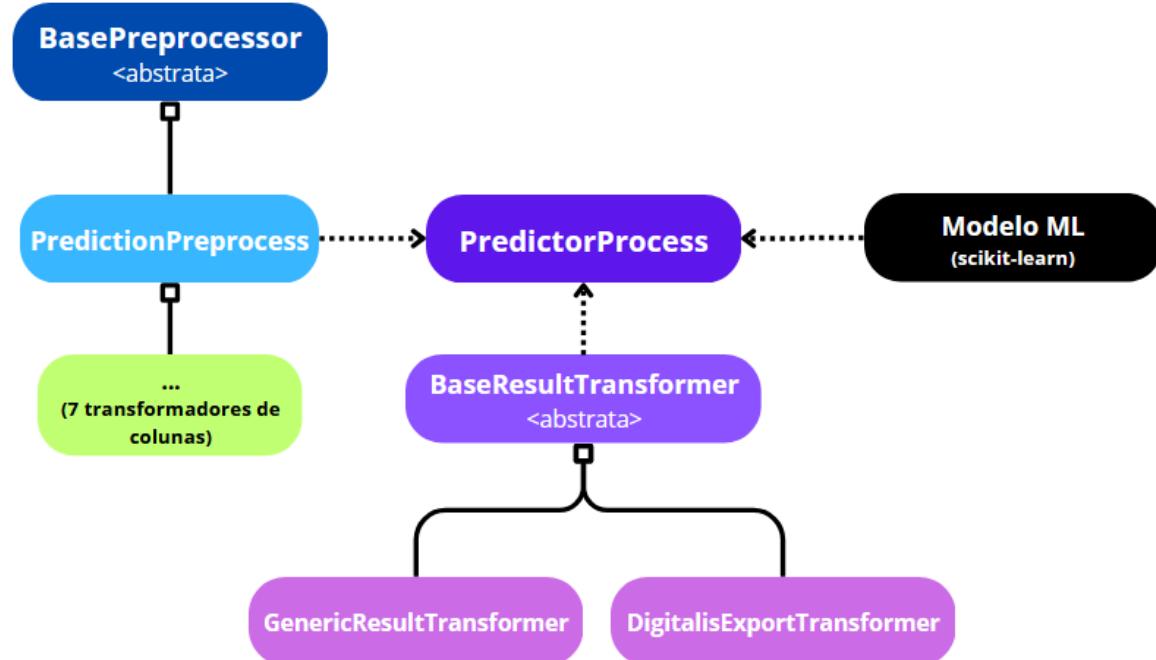


Figura 86 - Mapa de classes para o contexto de pré-processamento de treino de predição e processo de predição

Para que seja possível reutilizar o processo de pré-processamento, deve-se criar uma instância de “*PredictionPreprocess*”, que ao contrário da configuração do processo no contexto de treino (**Figura 84**), requer apenas um parâmetro, sendo este a configuração anteriormente exportada (ver **Figura 85**) num objeto *dict* (dicionário de Python) ou o caminho lógico do ficheiro local. Quando é dado o caminho lógico do ficheiro, a própria inicialização da instância é também responsável por efetuar a leitura do ficheiro de configuração. A **Figura 87** ilustra o código necessário para realizar essa importação.

```
class PredictionPreprocess(BasePreprocessor): 3 usages ± Miguel Magueijo
    required_config_variables = BasePreprocessor.required_config_variables | {"columns_transformers"=}

    def __init__(self, config: dict | str): ± Miguel Magueijo
        if isinstance(config, str):
            with open(config, "r", encoding=DEFAULT_FILE_ENCODING) as config_file:
                config = json_load_file(config_file)

        keys = PredictionPreprocess.required_config_variables - set(config.keys())
        if len(keys) > 0:
            raise InvalidConfig(f"Missing required config keys: {keys}")

        super().__init__(**config)
        self._required_columns = set(config["columns_rename_map"].keys())
        self._columns_transformers = []

        for transformer_config in config["columns_transformers"]:
            match transformer_config["python_class"]:
                case DistrictPostalCodeTransformer.__name__:
                    self._columns_transformers.append(DistrictPostalCodeTransformer.from_config(transformer_config))
                case BinaryColumnTransformer.__name__:
                    self._columns_transformers.append(BinaryColumnTransformer.from_config(transformer_config))
                case ReplaceColumnValuesTransformer.__name__:
                    self._columns_transformers.append(ReplaceColumnValuesTransformer.from_config(transformer_config))
                case MultiLabelBinarizeColumnTransformer.__name__:
                    self._columns_transformers.append(
                        MultilabelBinarizeColumnTransformer.from_config(transformer_config))
                case OneHotEncodeColumnTransformer.__name__:
                    self._columns_transformers.append(OneHotEncodeColumnTransformer.from_config(transformer_config))
                case RemoveColumnsTransformer.__name__:
                    self._columns_transformers.append(RemoveColumnsTransformer.from_config(transformer_config))
                case _:
                    raise InvalidConfig(f"Unknown transformer {transformer_config['python_class']}")
```

Figura 87 - Código de inicialização de um objeto *PredictionPreprocess*

Relativamente à classe “*PredictorProcess*”, esta é a que se responsabiliza pelo processo de uma predição, realizando o pré-processamento dos dados (caso seja necessário) e de os dar ao modelo de ML para efetuar a predição. A inicialização desta classe recebe obrigatoriamente um modelo de ML (instância de qualquer algoritmo de ML da biblioteca *Scikit-Learn*) e opcionalmente um objeto de “*PredictionPreprocess*” quando necessita de realizar pré-processamento e um objeto transformador dos resultados. Após a sua inicialização pode ser invocada a função *predict* que recebe como parâmetro um objeto *DataFrame* (biblioteca Pandas) com todas as instâncias a predizer. O código dessa mesma função é apresentado na **Figura 88**. É possível verificar que para além de ser realizado o pré-processamento (quando necessário) e comunicação com o modelo de ML, são criados meta dados como o tempo que o modelo de ML necessitou para predizer os valores de cada instância, número total de instâncias preditas e qual é a contagem de instâncias para cada valor possível.

```

def predict(self, df: pd.DataFrame, result_transformer: BaseResultTransformer = None, ± Miguel Magueijo
           result_value_map: dict = None) -> pd.DataFrame:
    num_instances_to_predict = len(df)
    if self.preprocess is not None:
        required_columns = list(self.preprocess.file_required_columns)

        to_predict_df = self.preprocess.transform_df(df[required_columns])[self.model.feature_names_in_]
    else:
        to_predict_df = df[self.model.feature_names_in_]

    start_prediction_time = time()
    predicted_values: np.ndarray = self.model.predict(to_predict_df)

    self.metadata["time_to_predict"] = f"{time() - start_prediction_time:.5f}s"
    self.metadata["total_instances_predicted"] = len(predicted_values)
    self.metadata["prediction_done_on"] = start_prediction_time

    if result_value_map is not None:
        result_map_func = np.vectorize(lambda v: result_value_map[v])
        predicted_values = result_map_func(predicted_values)

        self.metadata["class_features_stats"] = {}
        unique_values_count = np.unique(predicted_values, return_counts=True)
        for value, value_count in zip(unique_values_count[0], unique_values_count[1]):
            self.metadata["class_features_stats"][str(value)] = {
                "count": int(value_count),
                "percentage": f"{round(value_count / len(predicted_values), 4)}%"
            }

    if len(predicted_values) != num_instances_to_predict:
        raise Exception("Numbers of predictions is different from number of inputs")

    if self.preprocess is None:
        return pd.DataFrame(predicted_values, columns=[self.result_column_name])

    if result_transformer is None:
        return pd.DataFrame(predicted_values, columns=[self.result_column_name])

    result_transformer.generate_memory(df, required_columns)
    return result_transformer.join_with_results(predicted_values)

```

Figura 88 - Código da função "predict" da classe *PredictorProcess*

Caso seja necessário realizar transformações aos resultados, como mapear valores preditos para outra representação (por exemplo, de numérico para nominal), podem desenvolvidas e utilizadas classes que estendem a classe "*BaseResultTransformer*". Esta classe abstrata define as funções que as classes herdadas devem implementar para realizar suas próprias transformações. Posto isto, "*BaseResultTransformer*" atua de maneira semelhante a uma interface em Java. No contexto deste projeto e até ao momento foram criadas apenas duas classes que herdam de "*BaseResultTransformer*":

1. ***GenericResultTransformer***: permite que colunas do *DataFrame* original sejam transferidas para o *DataFrame* com os resultados após a predição;
2. ***DigitalisExportTransformer***: cria novas colunas no *DataFrame* de resultados, baseando-se em valores das colunas originais, antes de ocorrer o pré-processamento e a predição.

Desta forma, foi possível desenvolver uma solução modular do zero, que não só permite configurar pré-processamentos de *datasets* e reutilizar esse pré-

processamento, como também permite uma interação facilitada com os modelos de ML. A solução foca-se também na garantia de consistência nos pré-processamentos, minimizando o risco de erros originados de valores desconhecidos ou atributos inesperados durante o processo de predição. No entanto, apesar de a solução se encontrar numa versão funcional e ser possível executar as operações pretendidas, é importante destacar que ainda pode apresentar possíveis erros e *bugs* dado que não foram realizados testes extensivos.

Antes de ser apresentada a arquitetura e os ecrãs da aplicação, é relevante mencionar que a mesma está ativa e acessível através do URL <https://revup.dev.miguelmagueijo.pt>. O alojamento foi possível graças ao uso do serviço de hospedagem de servidores virtuais da Oracle, denominado *Oracle Cloud Free Tier* [139], que oferece, de forma gratuita, uma máquina virtual com recursos limitados.

Além disso, como o serviço disponibiliza um servidor virtual com o sistema operativo *Ubuntu Server 22.04*, foi necessário utilizar a tecnologia *NGINX* como servidor *HTTP* para servir a aplicação web. O autor optou também pela utilização da tecnologia *Docker* para criar *containers* (imagens isoladas do sistema operativo) para o servidor de interface gráfica (*SvelteKit*), *REST API (FASTAPI)* e base de dados (*PostgreSQL*).

7.2.1 Arquitetura e ferramentas utilizadas

A arquitetura da segunda aplicação assemelha-se em parte à primeira, particularmente na divisão em três componentes principais: interface gráfica, *REST API* e código *Python* para comunicar com o modelo de ML. No entanto, devido à maior complexidade desta aplicação, foram integradas duas novas vertentes: o sistema de ficheiros do sistema operativo e a base de dados. Para ilustrar toda a arquitetura de forma simples, foi criado o diagrama da **Figura 89**.

O desenvolvimento da segunda aplicação exigiu a utilização de várias bibliotecas e tecnologias sendo elas: *SvelteKit* para a interface gráfica, *FASTAPI* para a implementação da *REST API*, *Scikit-Learn*, *Pandas* e *Skops* para a interação e processamento de modelos de ML, e por último *PostgreSQL* como base de dados. Adicionalmente, como ferramentas foram utilizadas: *PyCharm* para codificação de código *Python* (*REST API*, pré-processamentos e interação com modelos de ML), *WebStorm* para codificação das páginas web (*SvelteKit*, *TypeScript* e *HTML*) e *DataGrip* para um controlo e manipulação da estrutura da base de dados.

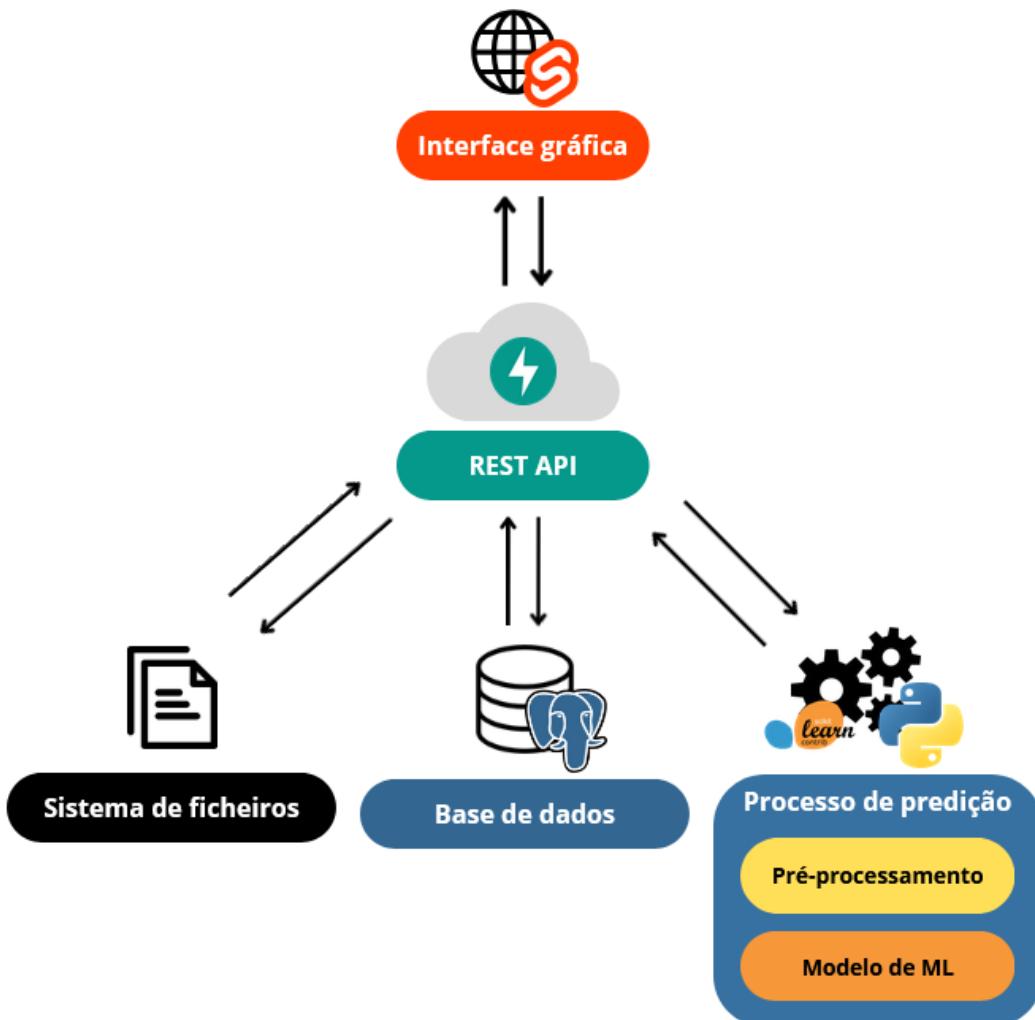


Figura 89 - Representação da arquitetura da segunda aplicação de forma simplificada

Dada a necessidade de serem armazenadas informações referentes a configurações de predições, pré-processamentos, modelos de ML, resultados de predições, utilizadores e sessões (autenticação), optou-se por utilizar uma base de dados *PostgreSQL*. De forma a ser armazenada toda essa informação, ao todo foram criadas 11 tabelas. Além disso, como os dados por sua natureza possuem relações entre si, foi necessário definir essas relações entre as tabelas.

Com o objetivo de facilitar a compreensão da estrutura da base de dados e das suas relações, foi gerado um diagrama (**Figura 90**) utilizando a ferramenta *DataGrip*. Este diagrama fornece uma representação visual que simplifica a interpretação da estrutura, uma vez que a interpretação direta do código seria mais complexa. É importante destacar que, ao longo do desenvolvimento da aplicação, as tabelas e suas relações passaram por alterações significativas, o que impossibilitou a criação de um diagrama entidade-relação fixo. Assim, as tabelas e seus campos ao longo do desenvolvimento foram sendo ajustadas conforme as funcionalidades que eram implementadas. Contudo, algumas funcionalidades foram previstas antecipadamente e não exigiram ajustes contínuos.

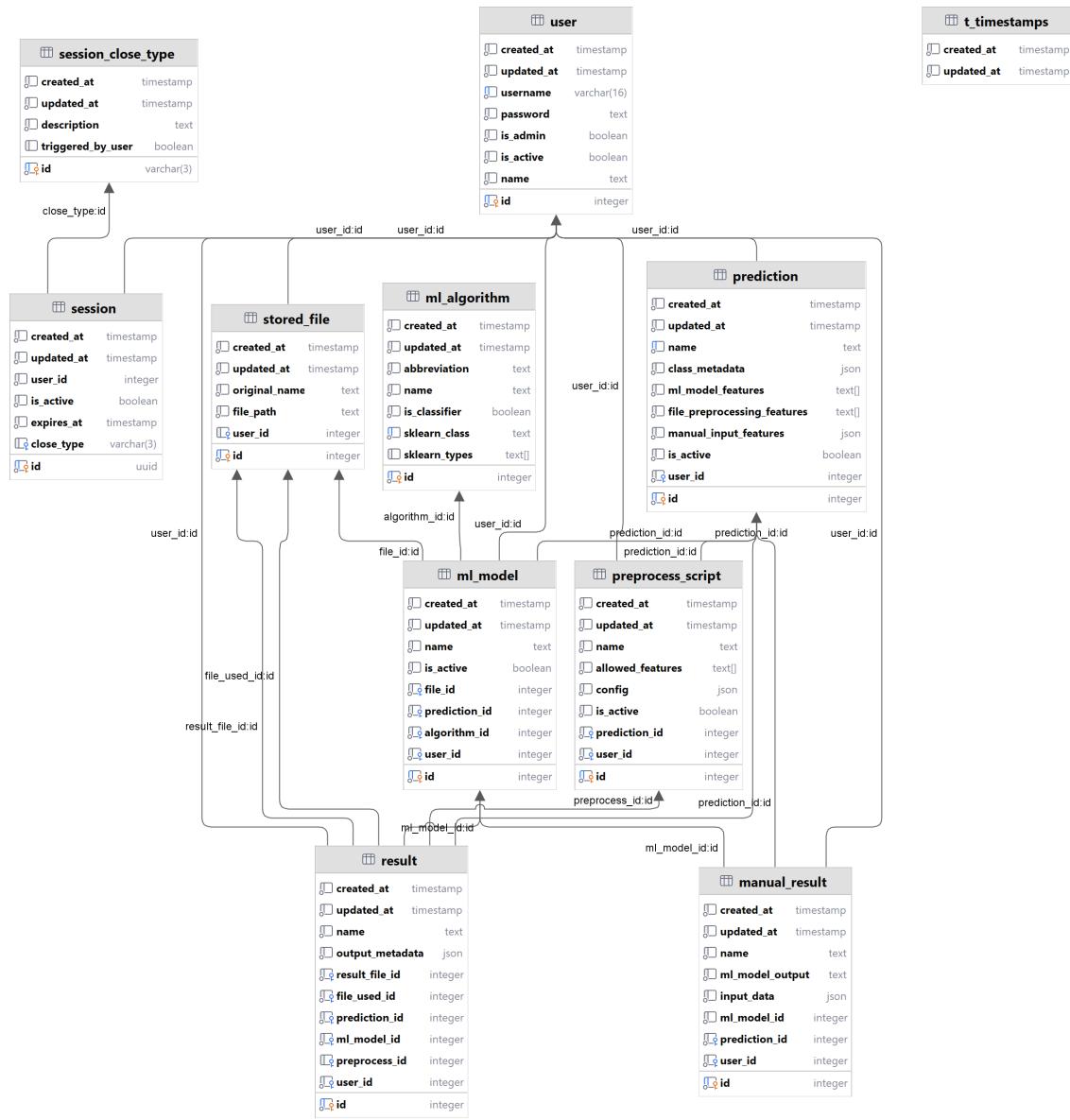


Figura 90 -Estrutura da base de dados a ser utilizada na segunda aplicação desenvolvida

Entre as tabelas criadas, destaca-se a tabela “`t_timestamps`”, que é utilizada exclusivamente para herança. Esta abordagem foi adotada porque todas as tabelas necessitam de duas colunas, “`created_at`” e “`updated_at`”, e ambas com o mesmo tipo de dado (“`TIMESTAMP`”). Em vez de ser duplicada a sua definição em cada tabela a criar, a herança permite centralizar a sua definição. No PostgreSQL, quando uma tabela herda outra tabela, as colunas da tabela mãe são automaticamente criadas na tabela filha, o que evita que quando as colunas comuns sofrem alterações, a sua alteração seja feita unicamente numa única definição. Posto isto, caso seja necessário modificar o tipo de dado de “`TIMESTAMP`” para “`DATE`”, por exemplo, essa alteração poderá ser feita diretamente na tabela “`t_timestamps`”, no entanto é necessário recriar todas as tabelas novamente, pois, uma alteração após a criação das tabelas não é propagada.

Além do uso da base de dados, a aplicação permite o *upload* de diferentes modelos de ML e armazena os ficheiros de dados de predição e aqueles que resultam de uma predição. Dessa forma, foi necessário armazenar esses ficheiros diretamente no sistema operativo. Sempre que um ficheiro é armazenado no sistema, é necessário registar o seu caminho lógico, uma vez que o autor não achou viável guardar ficheiros, em formato *bytes*, diretamente na base de dados.

Para isso, foi criada a tabela "stored_file", que contém um campo que referencia o caminho lógico onde os ficheiros são armazenados. Esta abordagem permite que esse caminho seja fornecido às diversas funções codificadas em Python, que por sua vez carregam o ficheiro correspondente. No entanto, o uso desta abordagem exige uma sincronização constante entre os registo da base de dados e os ficheiros armazenados. Se um registo ou ficheiro for alterado ou removido no sistema de ficheiros sem que essa alteração seja refletida na base de dados (ou vice-versa), ocorrerá um erro quando a função em Python tentar aceder ao ficheiro, devido à inconsistência de informações entre as duas fontes.

Por fim, é importante destacar que embora o seu uso não esteja explicitamente mencionado ou representado no diagrama da **Figura 89**, a arquitetura também inclui o uso das classes responsáveis pelos processos de pré-processamento e predição, apresentadas anteriormente. Sem a integração dessas classes, a implementação desta aplicação não seria possível, uma vez que elas desempenham um papel fundamental no processamento dos dados e na integração dos modelos de ML para as predições.

7.1.2. Ecrãs da aplicação

A segunda aplicação desenvolvida, apresenta uma maior variedade de ecrãs, e muitos destes estão disponíveis apenas para utilizadores com privilégios de administrador. Antes de serem apresentados os ecrãs acessíveis pelos utilizadores autenticados, é crucial apresentar aqueles que podem ser acedidos por utilizadores não autenticados.

Quando um utilizador sem autenticação entra na aplicação (através do URL), ele é direcionado para a página ilustrada na **Figura 91**. Esta página oferece apenas uma breve descrição da aplicação web e contém um botão que redireciona o utilizador para a página de login, onde se poderá autenticar.



Figura 91 - Segunda aplicação, página inicial de um utilizador sem autenticação

Na segunda aplicação, um utilizador não autenticado tem acesso restrito apenas à página inicial e à página de login. Se este tentar aceder a qualquer outra página, será automaticamente redirecionado de volta para a página inicial. Este comportamento de redirecccionamento também se aplica a utilizadores autenticados (ou não) que tentem aceder a páginas com privilégios de administrador sem a devida autorização.

Todo este processo de redirecccionamento é gerido no lado do servidor, recorrendo à utilização de um *hook* (SvelteKit) que valida se a sessão do utilizador (valor armazenado nas *cookies* do *browser*) está ativa e se o utilizador tem os privilégios para aceder a página (verificado em páginas protegidas do administrador). As instruções codificadas no *hook* são executadas todas as vezes que um utilizador tenta aceder a uma página, garantindo que as permissões são sempre verificadas antes de ser enviada a página do servidor para o cliente. Assim, foram codificadas instruções e funções que tratam deste processo de redirecccionamento e que são apresentadas na **Figura 92**. Estas verificações asseguram constantemente que a sessão do utilizador está ativa e válida, impedindo o acesso não autorizado.

```

export const handle: Handle = async ({ event, resolve }) => {
  no usages ± Miguel Magalhães
  event.locals.isUserAdmin = false;
  event.locals.hasSession = false;
  event.locals.isApiAlive = await isApiAlive();

  if (!event.locals.isApiAlive) {
    if (!event.url.pathname.startsWith("/offline")) {
      throw redirect(307, "/offline");
    } else {
      return resolve(event);
    }
  } else if (event.url.pathname.startsWith("/offline")) {
    throw redirect(303, "/");
  }

  const revSession = event.cookies.get("rev_session");

  if (revSession !== undefined) {
    const sessionData = await getSessionData(revSession);

    if (sessionData === null) {
      event.cookies.delete("rev_session", { path: "/" });
    } else {
      event.locals.hasSession = true;
      event.locals.isUserAdmin = sessionData.data.is_user_admin;
      event.locals.userId = sessionData.data.user_id;
      event.locals.userName = sessionData.data.user_name;
      event.locals.userUsername = sessionData.data.user_username;
    }
  } else {
    event.locals.hasSession = false;
  }

  if (isProtectedAdminPath(event.url.pathname) && !event.locals.isUserAdmin) {
    throw redirect(303, "/");
  }

  if (isProtectedPath(event.url.pathname) && !event.locals.hasSession) {
    throw redirect(303, "/");
  }

  return resolve(event);
}

```

Figura 92 - Segunda aplicação, código do hook de SvelteKit com instruções para redirecionar utilizadores para a página inicial quando tentam aceder ecrãs protegidos por autenticação ou páginas de administrador sem os privilégios suficientes

O autor decidiu que os utilizadores não poderão realizar um novo registo na aplicação. Assim, a adição de novos utilizadores só pode ser feita através de um ecrã de administrador ou manualmente (*queries SQL*). Quando um utilizador tenta realizar o login, ele visualizará a página apresentada na **Figura 93**. Se o mesmo tentar realizar login com credenciais inválidas, a página apresentará uma mensagem de erro como a ilustrada na **Figura 94**. Caso a conta do utilizador esteja desativada, será também apresentada uma mensagem específica que informa o utilizador dessa mesma restrição, **Figura 95**.

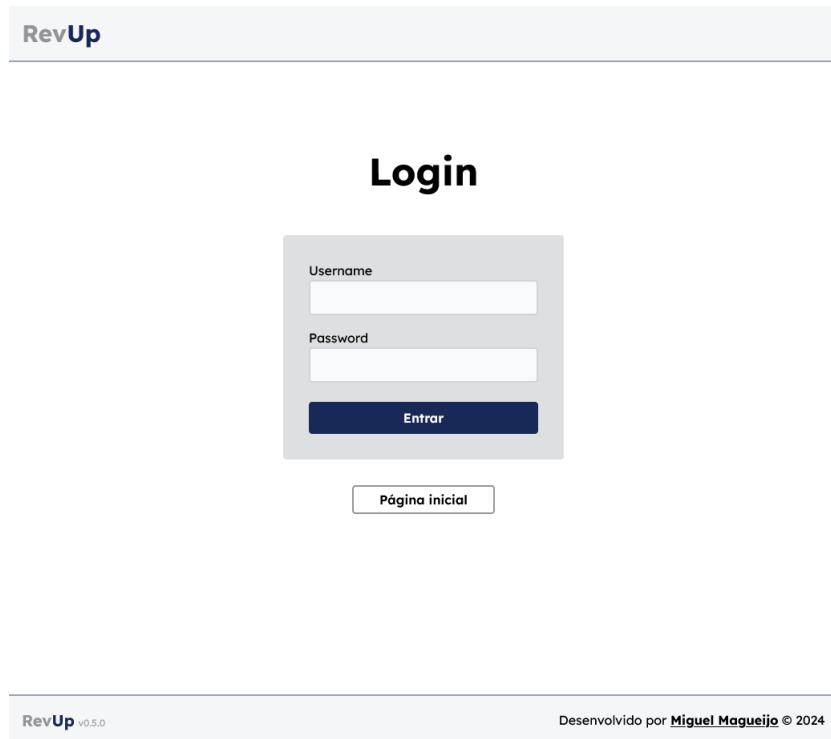


Figura 93 - Segunda aplicação, página de login

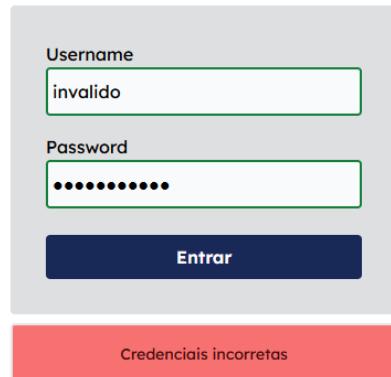


Figura 94 - Segunda aplicação, login com credenciais incorretas

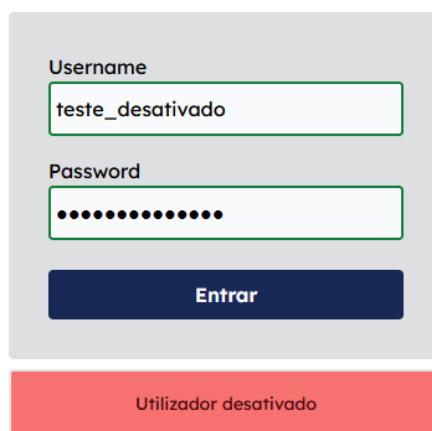


Figura 95 - Segunda aplicação, login de uma conta desativada

Quando o utilizador se autêntica com sucesso poderá visualizar duas versões diferentes da página inicial. Caso tenha privilégios de administrador visualizará a página apresentada pela **Figura 96** e quando não possui esses privilégios visualizará a da **Figura 97**. A única diferença entre as duas versões da página inicial são os botões de navegação presentes na barra de navegação e aqueles apresentados por baixo da descrição da aplicação web. Dado que um utilizador com privilégios de administrador possui uma maior variedade de páginas, este dispõe de um botão de navegação adicional, “Painel de administração”.

The screenshot shows the RevUp application's homepage for an administrator. At the top, there is a navigation bar with the logo "RevUp" and three links: "Painel de admin", "Nova predição", and "Meus resultados". On the far right of the navigation bar is a user profile icon labeled "admin". Below the navigation bar, the main title "Projeto RevUp" is displayed in large, bold letters, followed by the subtitle "Soluções inteligentes para o combate ao abandono e insucesso escolar". A descriptive paragraph states: "O projeto RevUp tem como missão combater o abandono e o insucesso escolar dos alunos do IPCB." Below this, there is a section titled "Para atingir este objetivo, o projeto utiliza técnicas de Machine Learning para desenvolver modelos preditivos inteligentes, capazes de identificar alunos em risco." Another section describes the tool as a means of interaction with models, allowing users to predict the risk of school abandonment based on text input or a form. At the bottom of the page, there are several buttons: "Páginas de administrador" (highlighted in a blue box), "Painel de administração" (highlighted in a blue box), "Páginas" (highlighted in a blue box), "Predizer alunos" (highlighted in an orange box), "Os meus resultados" (highlighted in an orange box), and "Definições da conta" (highlighted in a purple box). The footer contains the text "RevUp v0.5.0" and "Desenvolvido por Miguel Magueijo © 2024".

Figura 96 - Segunda aplicação, página inicial de um administrador



Figura 97- Segunda aplicação, página inicial de um utilizador autenticado sem privilégios de administrador

Estando autenticado o utilizador poderá visualizar o nome de utilizador (*username*) da sua conta no canto superior direito. Ao passar o rato por cima do seu nome de utilizador, é apresentado um menu suspenso (**Figura 98**) com dois botões, um para ser redirecionado até à página de definições de conta e outro para efetuar o *logout*.

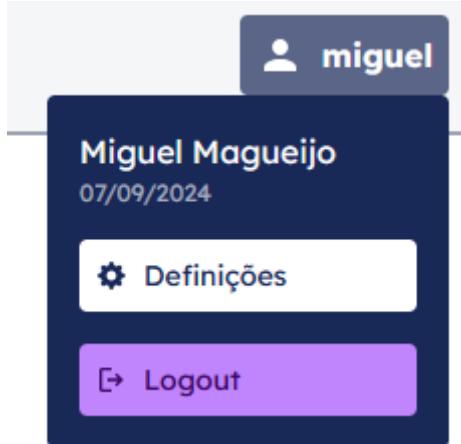
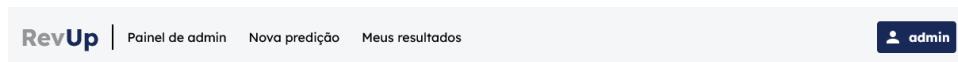


Figura 98 - Segunda aplicação, menu suspenso quando o utilizador passa o rato por cima do seu nome de utilizador na barra de navegação

Na página de definições da sua conta, o utilizador pode visualizar o seu nome e nome de utilizador, além de ter a opção de alterar a sua password, como ilustrado na **Figura 99**. Para garantir a segurança e correto uso deste processo, foram implementadas várias proteções. Uma delas é a verificação da password atual, pois, o

utilizador só pode alterar a password se a atual inserida for a correta, caso contrário, será apresentada uma mensagem de erro, conforme ilustrado na **Figura 100**.

Além disso, existem validações para quando a nova password e a sua confirmação não são iguais ou quando o utilizador tenta alterar a password atual para a mesma que já está em uso. Em ambos estes casos, é também apresentada uma mensagem de erro utilizando a mesma representação da **Figura 100**. Estas validações são feitas tanto no lado do cliente, através de Javascript, quanto no lado do servidor, pois, só assim é assegurado que pedidos inválidos de alteração de password sejam ignorados ou descartados.



 A screenshot of the 'Alterar password' (Change password) form from the RevUp application. The form consists of three input fields: 'Password atual' (Current password), 'Nova password' (New password), and 'Confirmar nova password' (Confirm new password). Each field has a placeholder text indicating character length requirements ('8 a 128 caracteres e/ou dígitos'). Below the fields is a purple button labeled 'Alterar Password' (Change Password).


Figura 99 - Segunda aplicação, página de definições da conta

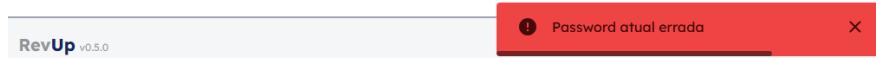
 A screenshot of the 'Alterar password' (Change password) form from the RevUp application. The form fields are identical to Figura 99. A red error message box at the bottom states 'Password atual errada' (Incorrect current password) with an exclamation mark icon. The RevUp logo and version 'v0.5.0' are at the bottom left, and the copyright notice is at the bottom right.


Figura 100 - Segunda aplicação, utilizador tenta alterar password com a atual errada

A segurança do uso correto da aplicação foi um dos pilares mais importantes durante o seu desenvolvimento. Assim, qualquer ação que envolva o utilizador é devidamente validada tanto no lado do cliente (através de *Javascript*), quanto no lado do servidor (*Python*). Esta abordagem garante que qualquer pedido realizado pelo utilizador seja verificado e validado. Porém, existem algumas exceções a esta regra. No caso da predição manual, a validação é realizada exclusivamente no lado do servidor. Isso acontece porque ainda não foi implementada uma funcionalidade capaz de efetuar essa validação, pois, os campos são gerados dinamicamente.

Antes de serem apresentados os menus de predição e resultados, acessíveis por qualquer utilizador autenticado, é necessário introduzir os diferentes ecrãs exclusivos a utilizadores com privilégios de administrador. O primeiro é o painel de administração, ilustrado na **Figura 101**, que atua como um portal para as várias páginas de controlo de conteúdo. Para facilitar a navegação e distinguir as diferentes áreas da aplicação, adotou-se pelo uso de cores diferentes para cada contexto em que: os botões relacionados à gestão de utilizadores são apresentados com a cor azul, enquanto os botões referentes à gestão de conteúdo da aplicação utilizam a cor laranja.

É importante ressaltar que ainda não foi implementada a funcionalidade que permite ao administrador consultar as previsões realizadas pelos utilizadores da aplicação. Esta escolha fundamenta-se no facto dessa funcionalidade necessitar ainda de ser estudada, visto que não se deseja que os administradores tenham a capacidade de alterar os resultados. Posto isso, a implementação dos ecrãs responsáveis por essa funcionalidade foram deixados para uma futura atualização.

The screenshot shows the RevUp Admin Panel interface. At the top, there is a navigation bar with the RevUp logo, a 'Painel de admin' link, and menu items 'Nova predição' and 'Meus resultados'. A user icon labeled 'admin' is also present. Below the navigation, there are several sections:

- Gerir utilizadores**: Includes a 'Utilizadores' section with 'Adicionar' and 'Consultar' buttons, and a 'Sessões' section with a 'Consultar & gerir (brevemente)' button.
- Gerir aplicação**: Includes sections for 'Predições' (with 'Adicionar' and 'Consultar' buttons), 'Pré-processamentos' (with 'Adicionar' and 'Consultar' buttons), and 'Modelos preditivos' (with 'Adicionar' and 'Consultar' buttons).
- Footer**: Shows the 'RevUp v0.5.0' logo and a copyright notice 'Desenvolvido por [Miguel Magueijo](#) © 2024'.

Figura 101 - Segunda aplicação, painel de administração

O administrador possui a capacidade de criar novos utilizadores e, para isso, pode visitar a página de criação de utilizador, ilustrada na **Figura 102**. Nesta página, o administrador necessita de preencher todos os campos obrigatórios, como nome, nome de utilizador (*username*) e password. De forma a garantir a criação de utilizadores com passwords fortes, o administrador pode gerar uma password aleatória, que é automaticamente copiada para a sua área de transferência. Adicionalmente, o administrador pode controlar o tamanho da password gerada, sendo que a mesma deve ter no mínimo 8 caracteres.

RevUp | Painel de admin Nova predição Meus resultados **admin**

[<< Painel de administração](#)

Criar novo utilizador

Formulário

Nome [+ Adicionar](#)

Deve conter 4 a 128 caracteres

Username

Deve conter 4 a 16 caracteres (letras, números e '_', não pode começar nem terminar por '_')

Password

Deve conter no mínimo 8 caracteres

Tamanho da password desejada [Gerar password aleatória](#)

RevUp v0.5.0 Desenvolvido por **Miguel Magueijo** © 2024

Figura 102 - Segunda aplicação, página de criação de um novo utilizador

Após preencher todos os campos obrigatórios, o administrador pode proceder à criação de um novo utilizador. Após ser confirmada a sua criação pela REST API, será apresentada uma mensagem de sucesso no canto inferior direito do ecrã (ver **Figura 103**). Por omissão, quando um utilizador é criado a sua conta já se encontra ativa.

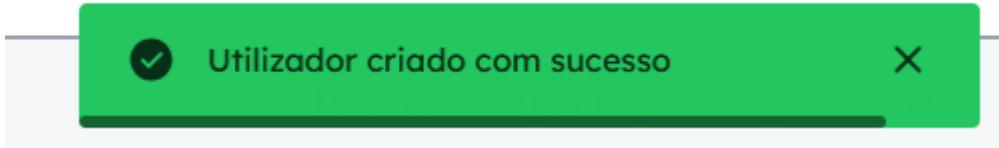


Figura 103 - Segunda aplicação, mensagem de sucesso na criação de um utilizador

O administrador pode também consultar os utilizadores existentes, como ilustrado na **Figura 104**. Enquanto consulta os utilizadores, este poderá remover um utilizador ou desativar a sua conta (impedindo o mesmo de efetuar login). No entanto, ainda não foi implementada a funcionalidade de edição de dados de um utilizador. Esta funcionalidade ainda não foi implementada devido ao tempo necessário de implementação da mesma e será adicionada em versões futuras. Considerando que pode existir um número elevado de utilizadores, foi adicionada a capacidade de filtrar por nome de utilizador, **Figura 105**.

RevUp | Painel de admin Nova predição Meus resultados

admin

<< Painel de administração

Utilizadores

Procurar por nome

ID	Nome	Username	Admin?	Criado a	Atualizado a	Ações
15	Novo utilizador	novo_utilizador	Não	07/09/2024	07/09/2024	Apagar Desativar
14	Teste desativado	teste_desativado	Não	07/09/2024	07/09/2024	Apagar Ativar
13	Daniel Salvado	danisal	Não	04/09/2024	04/09/2024	Apagar Desativar
12	Miguel Magueijo	miguel	Não	01/09/2024	01/09/2024	Apagar Desativar
10	Utilizador desabilitado 4	disabled4	Não	27/08/2024	27/08/2024	Apagar Ativar
9	Utilizador desabilitado 3	disabled3	Não	27/08/2024	27/08/2024	Apagar Ativar
8	Utilizador desabilitado 2	disabled2	Não	27/08/2024	27/08/2024	Apagar Ativar
7	Utilizador desabilitado 1	disabled1	Não	27/08/2024	01/09/2024	Apagar Ativar
...	Apagar Desativar

RevUp v0.5.0 Desenvolvido por Miguel Magueijo © 2024

Figura 104 - Segunda aplicação, página de consulta dos utilizadores

miguel

ID	Nome	Username	Admin?	Criado a	Atualizado a	Ações
12	Miguel Magueijo	miguel	Não	01/09/2024	01/09/2024	Apagar Desativar

Figura 105 - Segunda aplicação, funcionalidade de filtrar utilizadores pelo seu nome ou nome de utilizador (*username*)

Outra capacidade que o administrador possui é criar e controlar os tipos de previsões disponíveis aos utilizadores, bem como os pré-processamentos e modelos de ML disponíveis. A aplicação foi desenvolvida com o objetivo de suportar diferentes tipos de previsão, assim como permitir que seja possível predizer o abandono de um aluno e quais os ECTS realizados. Posto isto, o administrador pode acrescentar novos tipos de previsões ao aceder a página ilustrada na **Figura 106**.

Ao aceder à página de adição de um novo tipo de previsão, o administrador pode visualizar de forma imediata um aviso que a página lhe dá (destacado com um fundo amarelo). Este aviso recomenda que caso o administrador não saiba como funciona o processo de adição de novas previsões, não deve continuar o processo de e deve

comunicar com uma pessoa com conhecimento específico sobre esta ação, neste caso o programador.

The screenshot shows a web-based application interface for managing predictions. At the top, there's a header with the logo 'RevUp' and navigation links: 'Painel de admin', 'Nova predição', and 'Meus resultados'. A user icon labeled 'admin' is also present. Below the header, a breadcrumb navigation shows '<< Painel de administração'. The main title is 'Adicionar nova predição'. On the left, there's a 'Formulário' section with a 'Nome' input field containing 'exemplo: Continua estudos - CNA'. To the right of the input is a button '+ Adicionar nova'. Below this, there's a 'Ficheiro de metadata (pré-processamento)' section with a 'Selecionar ficheiro (.json)' button. A note below says 'Tomei conhecimento do aviso e responsabilizo-me por possíveis riscos: '. A yellow callout box contains an 'Aviso' section with instructions about selecting JSON files generated by the preprocessing script. It also lists requirements for the JSON file, including attributes like 'generated_at', 'model_input_features', 'preprocess_required_columns', 'prediction_features', and 'prediction_classes'. The bottom of the page includes the 'RevUp v0.5.0' logo and a copyright notice 'Desenvolvido por Miguel Magueijo © 2024'.

Figura 106 - Segunda aplicação, página de criação de um novo tipo de predição

Caso o administrador esteja familiarizado com o processo de adição de novos tipos de previsões, este necessita de selecionar um ficheiro JSON que contém as configurações da previsão a ser realizada. Este ficheiro nada mais é do que a *metadata* exportada do pré-processamento de treino, que é gerada pela classe “*TrainingPreprocess*”. Após selecionar um ficheiro válido, as configurações da previsão correspondente serão apresentadas na página, conforme ilustrado na **Figura 107**.

Ao serem apresentadas as configurações, o administrador poderá verificar os atributos necessários para realizar uma previsão manual, as colunas obrigatórias que os ficheiros com dados (instâncias de alunos) a serem usados para previsão precisam conter, além dos atributos que os modelos ML devem possuir. Caso o ficheiro correspondente da previsão inclua múltiplas classes a predizer pelos modelos de ML, o administrador poderá escolher qual a que será usada para o tipo de previsão a ser criada naquele momento.

Após o administrador dar um nome único ao tipo de previsão, o botão que permite efetuar a criação ficará ativo e poderá clicar no mesmo. Caso o nome inserido já exista na base de dados ou selecione um ficheiro de previsão inválido, uma mensagem de erro será apresentada no canto inferior direito do ecrã, conforme mostrado na **Figura 108**.

Em caso de sucesso, uma mensagem confirmando a criação do novo tipo será apresentada.

RevUp | Painel de admin Nova predição Meus resultados

Adicionar nova predição

Formulário

Nome
exemplo: Continua estudos - CNA

Ficheiro de metadata (pré-processamento)

Tomei conhecimento do aviso e responsabilizo-me por possíveis riscos:

Classe a prever:
ects_realizados: float

Informação do ficheiro selecionado
Ficheiro gerado a: 28/08/2024

Atributos para predição manual	Total: 19	Colunas obrigatórias de pré-processamento	Total: 19
ects_matriculados: float		DS_INSTITUIC;	
idade: integer		NM_CURSO;	
tipo_ingresso: one_hot		Total ECTS Matriculados;	
curso: one_hot		INGRESSO;	
grupo_profissional_mae: one_hot		SEXO;	
habilitacao_anterior: one_hot		NACIONALIDADE;	

Input do modelo de ML
Total: 203

ects_matriculados;
idade;
habilitacao_academica_pai;
habilitacao_academica_mae;
e_feminino;
tipo_aluno-Agente_Ensino;

Figura 107 - Segunda aplicação, página de adição de tipo de predição com pré-visualização do conteúdo do ficheiro selecionado e da predição a ser adicionada

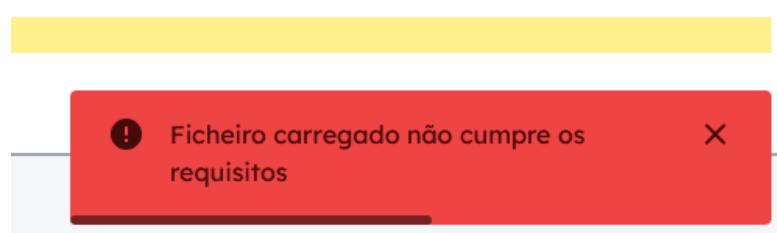


Figura 108 - Segunda aplicação, mensagem de erro apresentada ao administrador quando seleciona um ficheiro de *metadata* de predição inválido

Quando o administrador visita a página de consulta de tipos de predição, ele visualizará a interface ilustrada pela **Figura 109**. Ao contrário da página de consulta de utilizadores (**Figura 104**), onde os existentes são apresentados numa lista, a consulta de tipos de predições requer que o administrador selecione uma predição específica antes de visualizar os seus dados. Optou-se pela apresentação de uma única predição de cada vez dado que cada tipo de predição possui muita informação associada. Assim, após selecionar uma predição, toda a informação referente à mesma é carregada e apresentada ao administrador como ilustrado na **Figura 110**.

The screenshot shows the RevUp application's administration panel. At the top, there is a navigation bar with the logo 'RevUp' and links for 'Painel de admin' (selected), 'Nova predição', and 'Meus resultados'. On the right, there is a user icon labeled 'admin'. Below the navigation, a breadcrumb trail says '<< Painel de administração'. The main title is 'Predições'. A sub-section title 'Predição a visualizar' shows 'Nenhuma selecionada'. A blue information bar at the bottom states: 'Informações da predição selecionada serão apresentadas aqui'.

Figura 109 - Segunda aplicação, página para consultar os tipos de predições existentes

Painel de admin | [Nova predição](#) | [Meus resultados](#)

[**Predições**](#)

Predição a visualizar
19 a 23 - Continua Estudos

ID	6	Nome	19 a 23 - Continua Estudos	Desativar	Eliminar
Atributo classe & tipo	continua_estudos: boolean	Está ativa?	Sim		
Criada em	01/09/2024	Última alteração em	01/09/2024		
Atributos de predição manual		Total: 19	Colunas do pré-processamento		Total: 19
ects_matriculados:	float	DS_INSTITUIC;			
idade:	integer	NM_CURSO;			
tipo_ingresso:	one_hot	Total ECTS Matriculados;			
curso:	one_hot	INGRESSO;			
grupo_profissional_mae:	one_hot	SEXO;			
habilitacao_anterior:	one_hot	NACIONALIDADE;			
Inputs do modelo de ML		Total: 203			
ects_matriculados;					
idade;					
habilitacao_academica_pai;					
habilitacao_academica_mae;					
e_feminino;					
tipo_aluno-Agente_Ensino;					

RevUp v0.5.0 | Desenvolvido por [Miguel Magueijo](#) © 2024

Figura 110 – Segunda aplicação, página de consulta de tipos de predição quando o administrador seleciona uma predição existente

Enquanto o administrador visualiza um tipo de predição, este pode desativá-la, o que impede utilizadores de realizar esse tipo de predição. Adicionalmente, pode também eliminar o registo da base de dados. No entanto, a eliminação de um tipo de predição é considerada uma ação altamente destrutiva, pois, na versão atual da aplicação web, ao ser eliminada uma predição, todos os dados com relações a esse tipo de predição são também permanentemente eliminados. De forma a prevenir eliminações acidentais, todas as ações de eliminação disponíveis na aplicação web são acompanhadas por uma janela (*popup*) de confirmação, conforme apresentado pela **Figura 111**. Assim, é sempre garantido que o utilizador tem sempre a certeza quando deseja realizar uma ação de eliminação de conteúdo.

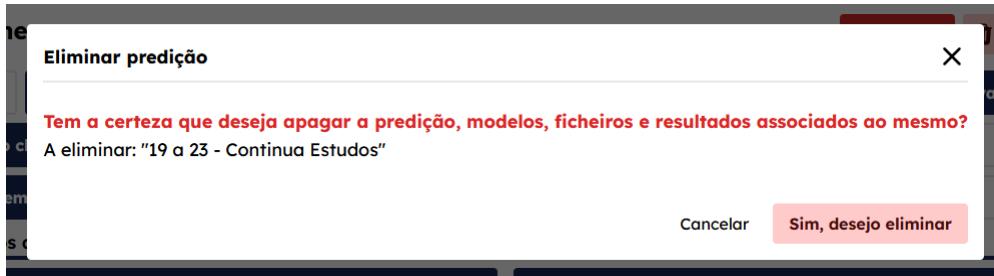


Figura 111 - Segunda aplicação, *popup* de confirmação de eliminação de conteúdo

Após adicionar um novo tipo de predição, o administrador, por norma, deve proceder à inserção de pré-processamentos correspondentes a essa predição. Para isso, deve visitar a página dedicada à adição de pré-processamentos. Quando o administrador entra nessa página, ele visualiza a interface ilustrada na **Figura 112**. Esta página segue uma lógica semelhante à de outros processos de adição, onde o administrador deve preencher um campo com o nome e selecionar o ficheiro de configuração correspondente.

Tal como a página de adição de novo tipo de predição (**Figura 106**), a página de adição de pré-processamentos apresenta também um aviso ao utilizador. Adicionalmente, dado que um pré-processamento está sempre vinculado a um tipo de predição existente, o administrador também deve indicar qual predição a vincular o processo de pré-processamento.

RevUp | Painel de admin Nova predição Meus resultados **admin**

[« Painel de administração](#)

Adicionar novo pré-processamento

Formulário

Nome **+ Adicionar**

Predição associada

Ficheiro de criação do processo

Tomei conhecimento do aviso e responsabilizo-me por possíveis riscos:

Aviso

Só deve selecionar ficheiros que foram devidamente gerados pelo script de pré-processamento.
Em caso de adicionar um pré-processamento mal formado, que no entanto é aceite, estará a comprometer o bom funcionamento da aplicação.
Uma vez que o modelo não conseguirá funcionar corretamente.
A informação do pré-processamento será validada com aquela da predição.

Requisitos

Formato do ficheiro: JSON (.json)
Tamanho máximo do ficheiro: 5MB
Propriedades:

- **columns_rename_map**: mapa (string: string) - as chaves também indicam as colunas obrigatórias a serem validadas com a predição
- **fill_na_value**: string
- **columns_transformers**: array de objetos (python_class, col_name, new_column_name, is_class, **outros) representativos dos transformadores
- * - propriedade opcional

Figura 112 - Segunda aplicação, página de adição de um novo pré-processamento

Assim que o administrador seleciona uma configuração válida de um processo de pré-processamento, os detalhes dessa configuração são apresentados ao mesmo (conforme ilustrado na **Figura 113**). Quando apresentados os detalhes, o administrador pode verificar as colunas obrigatórias que os ficheiros (com as instâncias a predizer) devem conter para serem pré-processados, bem como os transformadores que serão aplicados a cada uma dessas colunas (atributos).

É crucial que o pré-processamento esteja corretamente alinhado com o tipo de predição a ser associado. Ou seja, as colunas que precisam ser pré-processadas devem corresponder às aquelas especificadas no tipo de predição. Esta coerência é essencial para garantir que o *workflow* de predição seja executado sem erros. A verificação dessa mesma correspondência é realizada exclusivamente no lado do servidor e caso seja detetada alguma discrepância entre a configuração do pré-processamento e os requisitos do tipo de predição, o servidor envia uma mensagem de erro ao cliente informando-o dessa inconsistência (mensagem apresentada no canto inferior direito do seu ecrã).

RevUp | Painel de admin Nova predição Meus resultados

Adicionar novo pré-processamento

Formulário

Nome
exemplo: Nominais com one hot

Predição associada
Nenhuma selecionada

Ficheiro de criação do processo
Remover **preprocess_config_REVUP_all_v4.json**

Tomei conhecimento do aviso e responsabilizo-me por possíveis riscos:

Informação do ficheiro selecionado

Valor de substituição de nulos/vazios	Desconhecido/a
Colunas obrigatórias no ficheiro	Total: 19
DS_INSTITUIC;	
NM_CURSO;	
Total ECTS Matriculados;	
INGRESSO;	
SEXO;	
NACIONALIDADE;	

Transformadores	Total: 25
DistrictPostalCodeTransformer (codigo_postal -> distrito);	
BinaryColumnTransformer (sexo -> e_feminino);	
MultiLabelBinarizeColumnTransformer (tipos_aluno);	
ReplaceColumnValuesTransformer (curso);	
ReplaceColumnValuesTransformer (nacionalidade);	
ReplaceColumnValuesTransformer (naturalidade);	

RevUp v0.5.0 Desenvolvido por **Miguel Magueijo** © 2024

Figura 113 - Segunda aplicação, página de adição de um novo pré-processamento após ser selecionado um ficheiro de configuração válido

Seguindo a mesma estrutura de consulta e manipulação de conteúdo da aplicação web, o administrador também dispõe da possibilidade de consultar os processos de pré-processamentos existentes, sendo a interface dessa página apresentada pela **Figura 114**. Nesta página, os processos podem ser filtrados pelo seu nome (como a consulta de utilizadores) e o administrador pode desativar ou ativar a sua utilização, eliminar um processo ou carregar no botão “Ver” que irá apresentar toda a informação do processo de pré-processamento por baixo da lista de processos.

Novamente, a funcionalidade de editar os detalhes de um processo de pré-processamento existente ainda não foi implementada, devido à complexidade e necessidade garantir uma associação válida entre o pré-processamento e o tipo de predição à qual ele está associado. A implementação desta funcionalidade de edição está prevista a ser lançada em futuras atualizações da aplicação, de forma que quando os processos sejam alterados não comprometam o correto funcionamento do *workflow* de predição.

The screenshot shows the RevUp application's 'Pré-processamentos' (Pre-processing) page. At the top, there is a navigation bar with links for 'RevUp', 'Painel de admin' (selected), 'Nova predição', 'Meus resultados', and a user icon labeled 'admin'. Below the navigation is a search bar with the placeholder 'Procurar por nome' and a magnifying glass icon.

ID	Nome	Predição	Ações
4	Normal	[ID: 6] 19 a 23 - Continua Estudos	Apagar Desativar Ver
5	OneHot	[ID: 7] Situacao	Apagar Desativar Ver

Detalhes do pré-processamento selecionado

ID	4	Nome	Normal	Activ? Sim
Predição ID	6	Nome da predição	19 a 23 - Continua Estudos	
Utilizador criador ID	2	Username do utilizador	admin	
Criado a	01/09/2024	Atualizado a	01/09/2024	
Valor a substituir nulos/vazios	Desconhecido/a			

Colunas obrigatórias do ficheiro Total: 19 **Transformadores** Total: 25

DS_INSTITUIC	DistrictPostalCodeTransformer (codigo_postal -> distrito);
NM_CURSO	BinaryColumnTransformer (sexo -> e_feminino);
Total ECTS Matriculados	MultiLabelBinarizeColumnTransformer (tipos_aluno);
INGRESSO	ReplaceColumnValuesTransformer (curso);
SEXO	ReplaceColumnValuesTransformer (nacionalidade);
NACIONALIDADE	ReplaceColumnValuesTransformer (naturalidade);

RevUp v0.5.0

Desenvolvido por Miguel Magueijo © 2024

Figura 114 - Segunda aplicação, página de consulta de processos de pré-processamentos

Após serem adicionados o(s) processo(s) de pré-processamento de um tipo de predição, o administrador deve proceder à adição de modelos de ML, que por sua vez são aqueles que geram os resultados para os tipos de predições previamente criados. O design da página de adição de modelos de ML segue a mesma lógica das outras, como apresentado pela **Figura 115**.

Contudo, o processo de adição de um modelo de ML apresenta desafios adicionais em comparação com os outros componentes (tipo de predição e processo de pré-processamento), já que não é possível extrair de forma dinâmica a predição (classe, valor a predizer pelo modelo) que o modelo realiza. Isto dificulta a validação dinâmica do modelo de ML e a sua validação limitou-se à verificação dos atributos que o modelo recebe para garantir que os mesmos correspondem ao tipo de predição existente e associada. Portanto, este processo de adição de modelo de ML exige uma maior responsabilidade por parte do administrador de que o mesmo está a adicionar um modelo de ML válido para esse tipo de predição.

RevUp | Painel de admin Nova predição Meus resultados

Adicionar novo modelo preditivo

Detalhes do novo modelo

Nome
exemplo: Random Forest - Apenas CNA

Predição associada
Nenhuma selecionada

Algoritmo de Machine Learning do modelo
Nenhum selecionado

Ficheiro do modelo
Selecionar ficheiro (.skops)

Tomei conhecimento do aviso e aceito os riscos:

Aviso
Garanta que o ficheiro a importar é o correto, pois, os atributos do modelo serão validados com aqueles definidos presentes na predição.
Caso desconheça todo o processo, por favor fale com um desenvolvedor primeiro.
O envio de modelos que desconheça as suas métricas de avaliações ou funcionamento podem levar a maus resultados preditivos.

Figura 115 - Segunda aplicação, página de adição de um novo modelo de ML a uma predição existente

É importante destacar que todas as páginas de adição de componentes de predição possuem sempre um aviso que deve ser lido e aceite pelo administrador. Dado que a realização de validações extensivas e completas que garantem que nenhum erro ocorra no *workflow* de predição seria impraticável nesta fase inicial da aplicação web, a responsabilidade e garantia da correta adição de componentes recai sobre os administradores e outras pessoas responsáveis associadas. Caso ocorram problemas, é essencial identificar o utilizador responsável e abordá-lo adequadamente. Dessa forma, todos os dados na base de dados incluem o identificador (ID) do utilizador que efetuou a sua criação, facilitando o rastreamento e a resolução de problemas.

Numa tentativa de garantir que não é feito o *upload* de ficheiros de modelos de ML inválidos, foi implementada uma verificação específica do tipo de algoritmo de ML e do modelo de ML a ser adicionado. Como um modelo de ML exportado é essencialmente uma instância de um algoritmo da biblioteca Scikit-Learn, a REST API com base na utilização da biblioteca Skops é responsável por validar o conteúdo do ficheiro submetido pelo utilizador e assegurar que este corresponde a um algoritmo de ML existente. Caso a instância presente no ficheiro submetido pelo utilizador não corresponda ao tipo esperado, o utilizador será avisado através de uma mensagem de erro (conforme ilustrado na **Figura 116**) e o modelo de ML não será armazenado no servidor e o seu registo adicionado à base de dados.

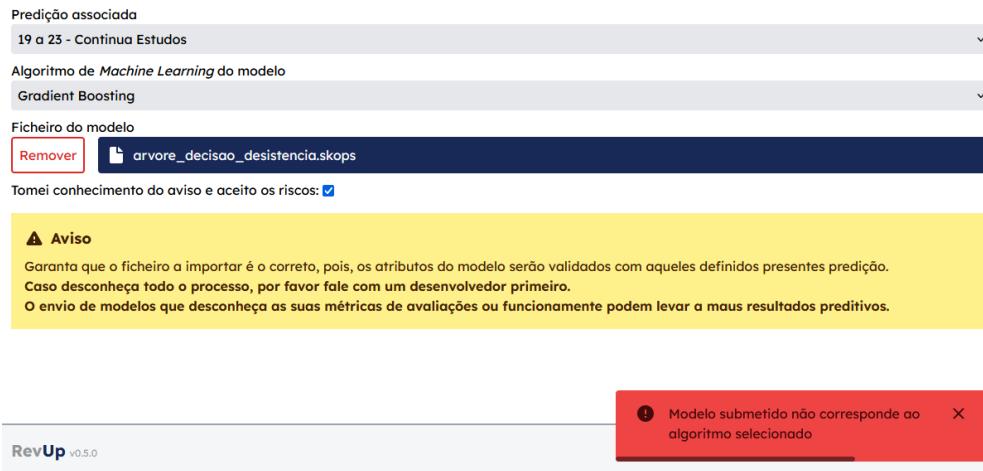


Figura 116 - Segunda aplicação, erro apresentado ao administrador quando submete um modelo de ML diferente do algoritmo selecionado

Quando um administrador visita a página de consulta de modelos de ML existentes, ele visualizará a interface apresentada pela **Figura 117**. Nesta página, o administrador pode remover um modelo existente ou desativar o seu uso. Dado que os modelos de ML são armazenados como ficheiros no sistema de ficheiros do servidor e são instâncias de classes da biblioteca Scikit-Learn, não há informação a apresentar sobre os mesmos, para além do seu nome e predição associada. Portanto, para além da visualização dessa informação, a única funcionalidade disponível ao administrador é a descarga (*download*) do modelo de ML.

ID	Nome	Predição	Algoritmo	Ações
6	Floresta sem melhoria	[ID: 6] 19 a 23 - Contínua Estudos	[RF] Random Forest	Apagar Desativar Download

RevUp v0.5.0

Desenvolvido por Miguel Magueijo © 2024

Figura 117 - Segunda aplicação, página de consulta de modelos de ML existentes

Após o administrador adicionar tipos de previsões, pré-processamentos e modelos de ML, os utilizadores têm a capacidade de realizar previsões com os seus dados (instâncias). Para isso, os utilizadores devem entrar na página de previsão, que pode ser acedida através do botão presente na página inicial ou barra de navegação.

Quando os utilizadores entram na página de previsão, necessitam inicialmente de selecionar o tipo de previsão que desejam realizar. Após esta seleção, são apresentados diferentes detalhes e opções conforme ilustrado na **Figura 118**. Os utilizadores podem verificar qual o valor que será previsto pelo modelo de ML para cada instância a ser dada pela previsão. Além disso, a página disponibiliza um campo de *input* para que o utilizador possa, opcionalmente, atribuir um nome ao resultado da previsão. Adicionalmente, para que o utilizador possa realizar uma previsão, este deve escolher o modelo de ML a ser utilizado, especificar os dados (instâncias) que serão dados ao modelo de ML para prever e opcionalmente o pré-processamento a aplicar caso queira prever instâncias de um ficheiro que não tenha sido pré-processado.

RevUp | Painel de admin [Nova predição](#) [Meus resultados](#)

[**<< Página inicial**](#)

Predizer alunos

Predição a realizar

19 a 23 - Continua Estudos

Valor a predizer: continua_estudos (boolean)

Nome do resultado (opcional): exemplo: Predição de CNA para ano letivo 24_25
Só letras (A a Z, números e espaços), não introduza caracteres especiais como ç e á

Selecionar modelo a utilizar

Filtrar por nome

- Floresta sem melhoria [RF] Random Forest

Total disponíveis: 1

Instâncias a predizer

Ficheiro Manual

Aviso: Deve garantir que o seu ficheiro cumpre os requisitos de input.
[Consultar colunas obrigatórias](#)

Selecionar ficheiro (.xlsx, .xls ou .csv)

Formato do ficheiro com o resultado: CSV (.csv)

Necessita de ser pré-processado:
Colunas não pré-processadas vão para o resultado:

Pré-processamento a aplicar: Normal

Realizar predição

RevUp v0.5.0 Desenvolvido por [Miguel Magueijo](#) © 2024

Figura 118 - Segunda aplicação, página de realização de uma nova predição

O utilizador tem duas opções de realização de predição, através de um ficheiro contendo as várias instâncias ou preenchendo um formulário para predizer uma única instância. Se o utilizador optar por realizar uma predição a partir de um ficheiro, será apresentado um aviso informando-o que o ficheiro deve cumprir certos requisitos de colunas. Caso o utilizador desconheça esse requisito, poderá clicar em “Consultar colunas obrigatórias” que irá apresentar um elemento com as colunas que o ficheiro deve conter, conforme ilustrado na **Figura 119**. Esse elemento é apresentado abaixo da seleção do pré-processamento e pode ser escondido a qualquer momento.

The screenshot shows a user interface for a machine learning application. On the left, there is a large input field labeled 'Total disponíveis: 1'. To the right, there are several configuration options:

- A yellow box at the top right contains the message: "Deve garantir que o seu ficheiro cumpre os requisitos de input." and the button "Consultar colunas obrigatórias".
- An orange button labeled "Selecionar ficheiro (.xlsx, .xls ou .csv)".
- A dropdown menu for "Formato do ficheiro com o resultado" set to "CSV (.csv)".
- A checkbox "Necessita de ser pré-processado: ".
- A checkbox "Colunas não pré-processadas vão para o resultado: ".
- A dropdown menu for "Pré-processamento a aplicar" set to "Normal".
- Two side-by-side lists of columns:
 - Colunas quando é necessário pré-processamento:** DS_INSTITUIC; NM_CURSO; Total ECTS Matriculados; INGRESSO; SEXO; NACIONALIDADE;
 - Colunas para o caso sem pré-processamento:** ects_matriculados; idade; habilaccao_academica_pai; habilaccao_academica_mae; e_feminino; tipo_aluno-Agente_Ensino;
- A grey button at the bottom labeled "Realizar predição".

Figura 119 - Segunda aplicação, apresentação das colunas obrigatórias que o ficheiro com as várias instâncias a predizer deve conter

Quando o utilizador realiza a predição para várias instâncias (ficheiro), deve indicar qual é o formato do ficheiro de resultados a ser gerado, podendo optar entre CSV ou XLSX (Excel). Adicionalmente, o utilizador pode especificar se o ficheiro precisa ser pré-processado e selecionar qual o pré-processamento a aplicar. Existe também a opção “Colunas não pré-processadas vão para o resultado”, que garante que as colunas que não foram transformadas durante o pré-processamento sejam transferidas para o ficheiro de resultado. Após definir todas as opções desejadas, o utilizador pode iniciar o processo de predição ao clicar no botão “Realizar predição”.

Após esperar um momento, durante o qual o modelo de ML realiza as predições, e depois da REST API retornar uma resposta, serão apresentadas as estatísticas do resultado de predição, conforme ilustrado pela **Figura 120**. Para além de visualizar as estatísticas de predição o utilizador poderá descarregar o ficheiro de resultados (com as instâncias e valor predito).

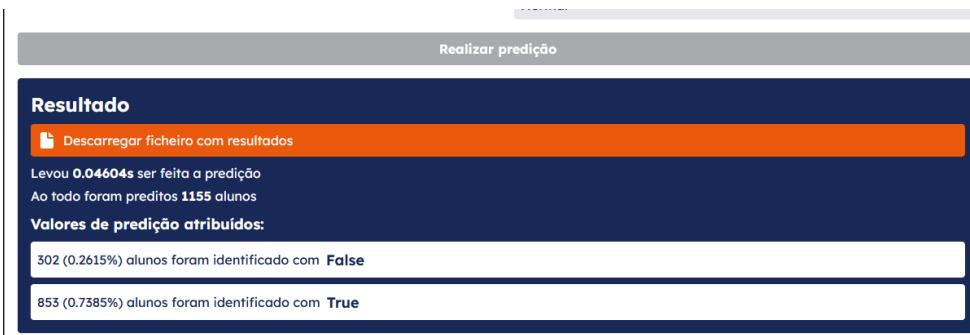


Figura 120 - Segunda aplicação, resultado apresentado depois da realização de uma predição de ficheiro

No entanto, caso o utilizador queira fazer uma única predição, este pode-a fazer através do preenchimento de um formulário. Desta forma, assim que o utilizador carrega no botão “Manual” a sua página irá ser alterada e será apresentado o formulário com os campos que necessita de preencher para realizar a predição, como ilustrado na **Figura 121**. Além do formulário, é também apresentado um aviso conforme os valores numéricos não são validados, pois, essa ainda é uma funcionalidade que ainda não foi implementada. Este aviso realça que a inserção de valores irrealistas irá, por consequente, resultar em predições irrealistas. Posto isto, após preencher todos os campos e clicar no botão “Realizar predição”, a predição resultante para os dados inseridos será apresentada abaixo do botão, conforme apresentado na **Figura 122**.

RevUp | Painel de admin [Nova predição](#) Meus resultados admin

[<< Página inicial](#)

Predizer alunos

Predição a realizar

19 a 25 - Continua Estudos

Valor a predizer continua_estudos (boolean)

Nome do resultado (opcional)
exemplo: Predição de CNA para ano letivo 24_25
Só letras (A a Z, números e espaços), não introduza caracteres especiais como ç e á

Selecionar modelo a utilizar

Filtrar por nome

Floresta sem melhoria
[RF] Random Forest

Instâncias a predizer

Ficheiro Manual

Por favor preencha o formulário que se encontra mais em baixo

Total disponíveis: 1

A Todos os campos são obrigatórios de preencher.
Importante: valores numéricos não são validados, podendo ser inserido valores irrealistas (negativos), no entanto isso leva também a uma predição irrealista

Ects matriculados	Idade
Tipo ingresso	Curso
Grupo profissional mae	Habilitacao anterior
Regime estudo	Escola
Situacao profissional pai	Situacao profissional mae
Distrito	País ensino secundario
Grupo profissional pai	Habilitacao academica mae
Habilitacao academica pai	Naturalidade
Nacionalidade	Tipos aluno
E feminino	

Realizar predição

RevUp v0.5.0 Desenvolvido por [Miguel Magueijo](#) © 2024

Figura 121 - Segunda aplicação, página de predição quando o utilizador pretende realizar uma predição manual (instância única)



Figura 122 - Segunda aplicação, resultado apresentado ao utilizador depois da realização de uma predição manual

Após realizar as predições, o utilizador pode consultar as mesmas (históricas) a qualquer momento. Essa consulta é possível através de visita à página ilustrada na **Figura 123**. Quando este entra na página poderá imediatamente visualizar os resultados de ficheiros e dispõe da opção para visualizar apenas os resultados manuais (instâncias únicas).

The screenshot shows the "Meus resultados" (My Results) page of the RevUp application. The header includes the RevUp logo, a "Painel de admin" link, a "Nova predição" link, and a "Meus resultados" link which is underlined. A "admin" user icon is also present.

Os meus resultados

Two tabs are visible: "Ficheiros" (selected) and "Manuais". Below them are two filter inputs: "Filtrar por nome" and "Filtrar por predição". A dropdown menu shows "Todas" and other options. The main content area displays a single result entry for a "Teste relatorio" (Test report) from 07/09/2024:

- Predição: 19 a 23 - Continua Estudos
- Nome do modelo: Floresta sem melhoria
- Algoritmo do modelo: Random Forest
- Estatísticas**: Levou 0.04604s ser feita a predição. Ao todo foram preditos 1155 alunos.
- Valores de predição atribuídos:**
 - 853 (0.7385%) alunos foram identificado com True
 - 302 (0.2615%) alunos foram identificado com False

At the bottom, there are "RevUp v0.5.0" and "Desenvolvido por Miguel Magueijo © 2024" links.

Figura 123 - Segunda aplicação, página de consulta de resultados das predições realizadas

Conforme ilustrado na **Figura 123**, a página de consulta de resultados permite que os utilizadores filtrem os resultados por tipo, "Ficheiro" ou "Manual". Adicionalmente, em cada tipo, é ainda possível aplicar filtros adicionais por nome e/ou tipo de predição. Posto isto, quando os utilizadores visualizam resultados relacionados a ficheiros (ver **Figura 123**), podem consultar o tipo de predição realizada, o nome do modelo de ML utilizado, o algoritmo de ML do modelo, estatísticas de predição e têm a opção de descarregar o ficheiro com os resultados.

No caso de consultar os resultados manuais (ver **Figura 124**), a página apresenta informações semelhantes, como o tipo de predição, o nome do modelo de ML e o

algoritmo de ML do modelo. No entanto, como só foi realizada a predição para uma única instância, a apresentação do resultado possui o valor predito e um botão que abre uma janela (*popup*) com os valores dos diferentes campos preenchidos pelo utilizador. Porém, numa versão inicial a apresentação dos valores da instância ainda é feita em formato cru (JSON), conforme ilustrado na **Figura 125**.

The screenshot shows the RevUp application interface. At the top, there is a navigation bar with the logo 'RevUp' and links for 'Painel de admin', 'Nova predição', and 'Meus resultados'. A user profile icon indicates the user is 'admin'. Below the navigation bar, the main title 'Os meus resultados' is displayed. Underneath, there are two tabs: 'Ficheiros' (highlighted in orange) and 'Manuais'. A search bar labeled 'Filtrar por nome' is present. Below it, a dropdown menu for 'Filtrar por predição' is set to 'Todas'. Two results are listed:

- 20240907_184721** (Realizada a: 07/09/2024)
 - Predição: 19 a 23 - Continua Estudos
 - Nome do modelo: Floresta sem melhoria
 - Algoritmo do modelo: Random Forest
 - Valor predito pelo modelo: True
- 20240905_173519** (Realizada a: 05/09/2024)
 - Predição: 19 a 23 - Continua Estudos
 - Nome do modelo: Floresta sem melhoria
 - Algoritmo do modelo: Random Forest

At the bottom of the page, there is a footer with the 'RevUp v0.5.0' logo and the text 'Desenvolvido por Miguel Magueijo © 2024'.

Figura 124 - Segunda aplicação, página de consulta de resultados manuais

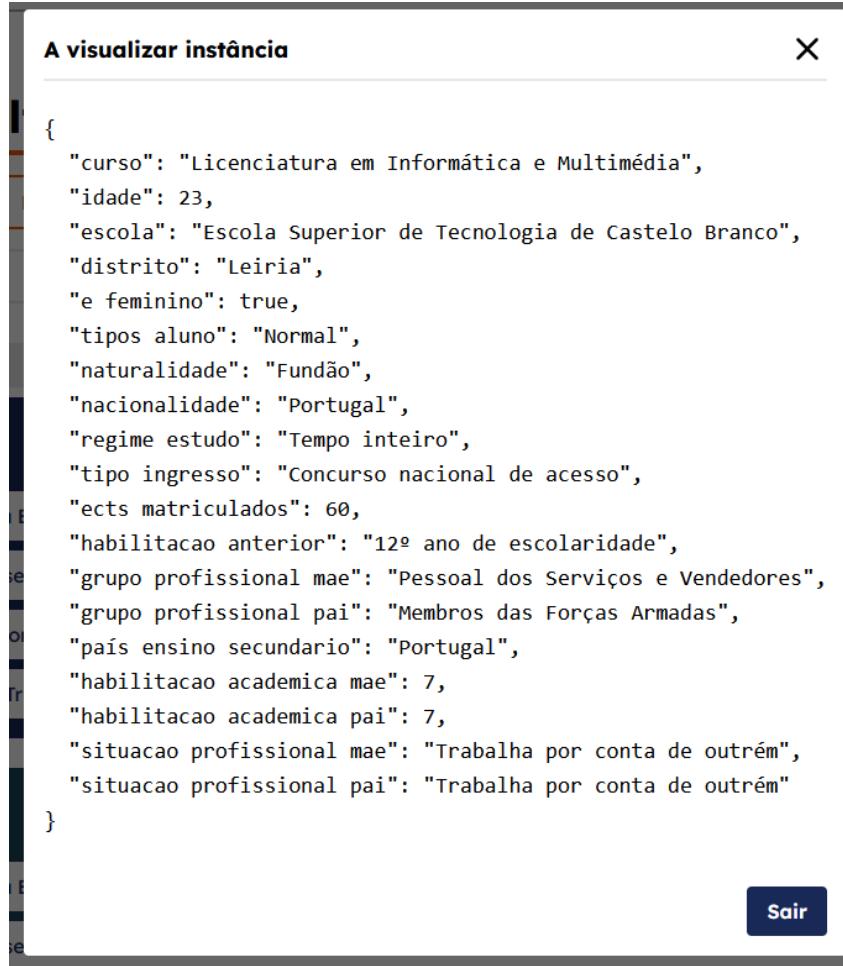


Figura 125 - Segunda aplicação, janela com os valores dos campos preenchidos (da instância predita) pelo utilizador

Depois de apresentados todos os ecrãs da aplicação é importante realçar que muitos detalhes da sua implementação foram omitidos. Esta decisão explica-se pelas limitações temporais de escrita e entrega do relatório, bem como o objetivo de evitar um documento excessivamente extenso.

Adicionalmente, caso este documento esteja a ser utilizado como documentação técnica da aplicação, solicita-se que qualquer questão relacionada ao funcionamento ou à implementação das suas funcionalidades seja feita diretamente ao autor através do email miguelmagueijo@gmail.com.

8. Contribuição externa do autor na Digitalis

Conforme mencionado ao longo do relatório, a empresa Digitalis [140] chegou a um acordo com o IPCB para desenvolver um módulo, denominado por SI.PREVINA, com o objetivo de ser possível acompanhar os alunos identificados com determinado risco. No entanto, após o acordo, surgiu um impasse referente ao uso e integração da componente de ML, já que a Digitalis, até o momento do acordo, não possuía colaboradores especializados nessa área. Dado que este projeto aborda precisamente a componente de ML e o autor possui familiaridade com técnicas de ML, a Digitalis apresentou-lhe uma proposta de emprego, a qual foi aceite.

O autor iniciou as suas funções na Digitalis no dia 1 de Julho de 2024, tendo como um dos objetivos desenvolver uma primeira versão do modulo SI.PREVINA a ser entregue no mês de Setembro de 2024. Posto isto, visto que o autor é um dos principais desenvolvedores e contribuidores (componente de ML) deste novo módulo, o mesmo optou por partilhar de uma forma breve a sua contribuição. Adicionalmente, a escolha de apresentação desta contribuição fundamenta-se em que o produto final (modulo PREVINA) utiliza os diferentes componentes de pré-processamento, treino e avaliação que o autor desenvolveu no contexto deste projeto.

De forma a manter este capítulo breve, optou-se por apresentar o *workflow* do SI.PREVINA no contexto do IPCB e os ecrãs desenvolvidos até ao momento de forma simplificada, uma vez que uma explicação detalhada seria mais adequada para um relatório ou apresentação independente, dada a complexidade do(s) produto(s) da Digitalis. Antes de ser apresentado o *workflow* típico do uso do SI.PREVINA e seus ecrãs, é importante fornecer uma visão geral simples das diferentes tecnologias utilizadas pela Digitalis no desenvolvimento das suas aplicações web focadas em gestão académica.

Atualmente, a Digitalis utiliza uma *framework* interna de desenvolvimento de aplicações web, denominada DIF (Digitalis Framework) [141], inteiramente codificada em Java. Esta *framework* foi totalmente desenvolvida, e atualmente atualizada, pelos próprios colaboradores (desenvolvedores de software) da empresa. Além disso, a Digitalis para o desenvolvimento e ambiente de produção das suas aplicações utiliza a Oracle Database como sistema de base de dados. Com estas duas tecnologias, a Digitalis desenvolveu diversas aplicações web dedicadas, sobretudo, à gestão académica são que amplamente utilizadas por dezenas de instituições de ensino superior. Entre as aplicações mais populares está o NetP@, que é também a plataforma utilizada pelo IPCB e na qual será integrado o SI.PREVINA.

O *workflow* do módulo SI.PREVINA pode ser dividido de uma forma muito simples em oito passos distintos. Para ilustrar esses passos, foi criado o diagrama da **Figura 126**. No entanto, é crucial realçar que este *workflow* não reflete o produto final. Várias funcionalidades, como mentorias, agendamento de sessões mentoria, agendas únicas para cada utilizador, registo de atitude em aula de cada aluno, entre outras, estão

omitidas, pois só estarão disponíveis numa segunda versão, cujo lançamento está previsto para 2025.

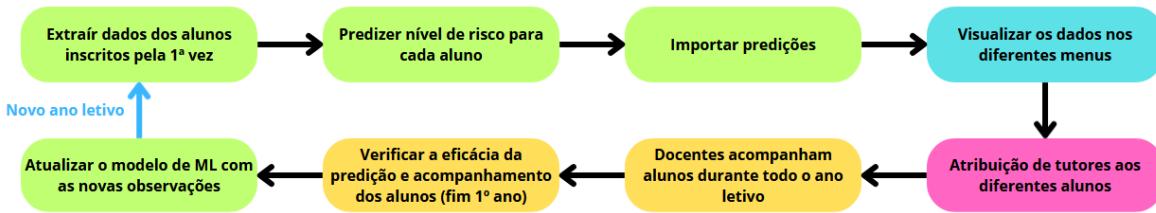


Figura 126 - Exemplo simplificado do *workflow* do módulo SI.PREVINA no contexto do IPCB até ao momento

Como ilustrado na **Figura 126**, o módulo SI.PREVINA, no contexto do IPCB, começa com a extração de dados dos alunos que se inscrevem no ensino superior pela primeira vez. Após a extração, é feito um pedido a um modelo de ML para prever o nível de risco desses novos alunos. Com a predição realizada, os resultados do modelo de ML são importados para o SI.PREVINA, permitindo que os docentes do IPCB (como o conselho pedagógico, diretores de curso e professores) accedam a essas predições através dos diferentes menus disponíveis.

No entanto, logo após as predições, os diretores de curso atribuem tutores (outros docentes, professores) aos alunos sinalizados. Com a atribuição concluída, inicia-se o acompanhamento dos alunos ao longo do ano letivo. Este acompanhamento deve ser feito regularmente por todos os docentes, que poderão consultar os resultados das avaliações e o comportamento dos alunos (assiduidade e atitude em aula) mediante o decorrer do ano letivo. O acompanhamento termina no final do ano letivo e após o seu término é realizado um balanço final.

Nesse balanço final, as partes envolvidas (conselho pedagógico e professores) avaliam se foi possível apoiar os alunos em risco para que não desistassem e continuassem os seus estudos. Adicionalmente, nesta etapa, devem ser comparadas as predições realizadas pelo modelo de ML com as situações finais dos alunos. Isso permite identificar os casos em que o modelo fez predições incorretas e aqueles em que a predição foi precisa e ajudou o aluno. Por fim, após o balanço final, as novas observações (com correção de eventuais predições incorretas) são usadas para treinar novamente o modelo, que será atualizado para o próximo ano letivo.

Relativamente aos ecrãs, são apresentados aqueles que o autor desenvolveu até ao último momento em que este relatório foi escrito (setembro de 2024). A escolha de serem apresentados unicamente os ecrãs desenvolvidos pelo autor vai ao encontro do objetivo deste capítulo, destacar as contribuições externas. O autor, até ao momento, contribuiu com a criação de dois ecrãs distintos que, no entanto, incluem múltiplos submenus e elementos acessíveis apenas por interação.

É importante realçar que os ecrãs apresentados ainda estão em desenvolvimento, ou seja, estão sujeitos a alterações até à sua versão final. Adicionalmente, todos os

dados que compõem os mesmos são de teste, podendo conter valores que, em um cenário de produção, não seriam possíveis ou falte o seu preenchimento.

O primeiro ecrã, **Figura 127**, é aquele que qualquer docente do IPCB visualiza quando entra no modulo SI.PREVINA. Ao entrar, o docente possui a indicação de qual é a análise que se encontra a visualizar, que por omissão é a mais recente, e todos os submenus disponíveis no lado esquerdo do seu ecrã. O docente poderá navegar entre diferentes análises ao clicar em “Selecionar análise” que por sua vez abre um elemento *dialog* com as diferentes análises disponíveis, como exemplificado na **Figura 128**. Após selecionar a análise a visualizar, a página é recarregada com os dados referentes à análise escolhida.

Figura 127 - SI.PREVINA, página principal do docente

Figura 128 - SI.PREVINA, dialog de escolha da análise a ser visualizada

O menu de navegação, **Figura 129**, disponibilizado ao utilizador docente apresenta diferentes opções de acordo com o seu perfil. Existem cinco perfis distintos (excluindo o caso do psicólogo, que ainda está em fase inicial):

1. Presidente do Conselho Pedagógico (Conselho Pedagógico) –visualiza informações a nível da instituição;
2. Diretor (Coordenador) de um ou vários cursos (Diretor de curso) – visualiza informações no contexto dos cursos de que é diretor;
3. Regente (Responsável) de Unidade Curricular (Regência da UC) – visualiza informações das UCs que é regente;
4. Docente (docência, disponível para todos os perfis) – visualiza informações das UCs que leciona, perfil base e sempre presente;
5. Tutor (Tutoria) – visualiza os alunos para os quais lhe foi atribuído a funções de tutor;

Desta forma, o utilizador visualiza apenas as opções que possui privilégios (possui perfil). Por exemplo, um docente que não é presidente do conselho pedagógico, não possui qualquer tipo de regência ou tutoria só poderá ver as opções de "Docência". Já um diretor de curso, além das opções de "Docência", também dispõe das opções de "Diretor de curso". É importante destacar que um docente pode acumular múltiplos perfis, podendo até possuir todos, visualizando nesse caso o menu exemplificado pela **Figura 129**.

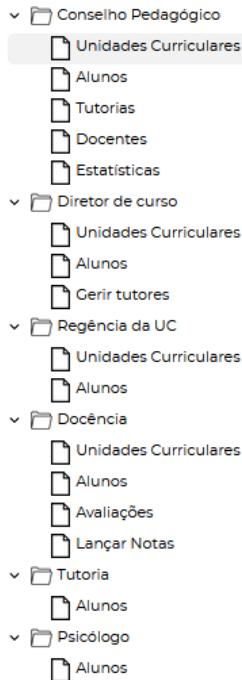


Figura 129 - SI.PREVINA, menu de navegação do docente

Apesar de existirem divisórias dependentes do perfil do utilizador (docente), grande parte das opções existentes acabam por apresentar conteúdos idênticos, que por sua vez são responsáveis por filtrar devidamente os dados com aqueles que o perfil

do utilizar tem acesso. Assim, um docente simples (sem regências, associação ao conselho pedagógico e/ou tutorias) ao visualizar as suas unidades curriculares ou alunos, só vê aquelas que leciona. Em contraste, um diretor de curso pode visualizar todos os alunos e unidades curriculares do curso(s) de que é regente, enquanto o presidente do conselho pedagógico pode ver todos os dados da instituição em que é responsável pelo conselho pedagógico. Desta forma, uma das opções é a consulta de “Unidades Curriculares” que quando selecionada abrirá a lista exemplificada pela **Figura 130**.

CONSELHO PEDAGÓGICO - UNIDADES CURRICULARES

Período	Código	Nome	Regente(s)	Total alunos	
				Inscritos	Sinalizados
Anual	9002356	Análise Matemática		13	1 Ver alunos
Anual	9002263	Análise Matemática		2	Ver alunos
Anual	9002003	Arquitectura Inf Beta		8	Ver alunos
Anual	9002359	Base de Dados		17	Ver alunos
Anual	9002352	Base de dados		6	Ver alunos
Anual	9002357	Controlo de Qualidade			
Anual	10001004	Espanhol			
Anual	12	Estrutura de Dados e Técnicas de Programação			
Anual	9002353	Gestão de dados		13	Ver alunos
Anual	9002364	Gestão Financeira	Abel Luís Costa Fernandes	1	Ver alunos
Anual	2	HISTÓRIA DO DIREITO d'orey		2	Ver alunos
Anual	805	Informática		3	Ver alunos
Anual	23	Inglês		11	Ver alunos
Anual	9002002	Inglês Inf. Beta			
Anual	1	INTRODUÇÃO AO ESTUDO DO DIREITO	Abel Luís Costa Fernandes	6	Ver alunos
Anual	9002361	Métodos Qualitativos para a Ciência Política	Beta Gestão Rafael dos Santos Pereira	2	1 Ver alunos
Anual	9002360	Princípios Gerais do Direito	Carolina CSP Rafael dos Santos Pereira	2	1 Ver alunos
Anual	9002358	Programação Avançada		3	Ver alunos
Anual	9002355	Programação I		10	Ver alunos
Anual	9002354	Tecnologias da Informação	Bianca Artes	9	Ver alunos

Figura 130 - SI.PREVINA, lista apresentada quando o utilizador seleciona a opção “Unidades Curriculares”

Quando o utilizador se encontra a visualizar as diferentes UCs que o seu perfil permite, este pode visualizar informações básicas de cada UC, como o período de lecionação, código interno, nome, regentes, número de alunos inscritos e número de alunos sinalizados. Além disso, cada UC da lista dispõe da opção “Ver alunos” que por sua vez abre um *dialog* com todos os alunos dessa UC, conforme ilustrado na **Figura 131**.

No *dialog* de consulta dos alunos da UC, o utilizador pode filtrar os alunos por “Todos”, “Sinalizados” ou “Não sinalizados”. Dentro deste diálogo, o utilizador pode consultar as informações básicas de cada aluno, nomeadamente o nível de sinalização, e ainda dispõe de outras consultas, as avaliações e presenças. As avaliações podem ser consultadas através de um *dialog* específico (**Figura 132**) ou através de uma outra lista adicional (**Figura 133**), acessível na barra de navegação incorporada no *dialog* dos alunos (**Figura 131**).

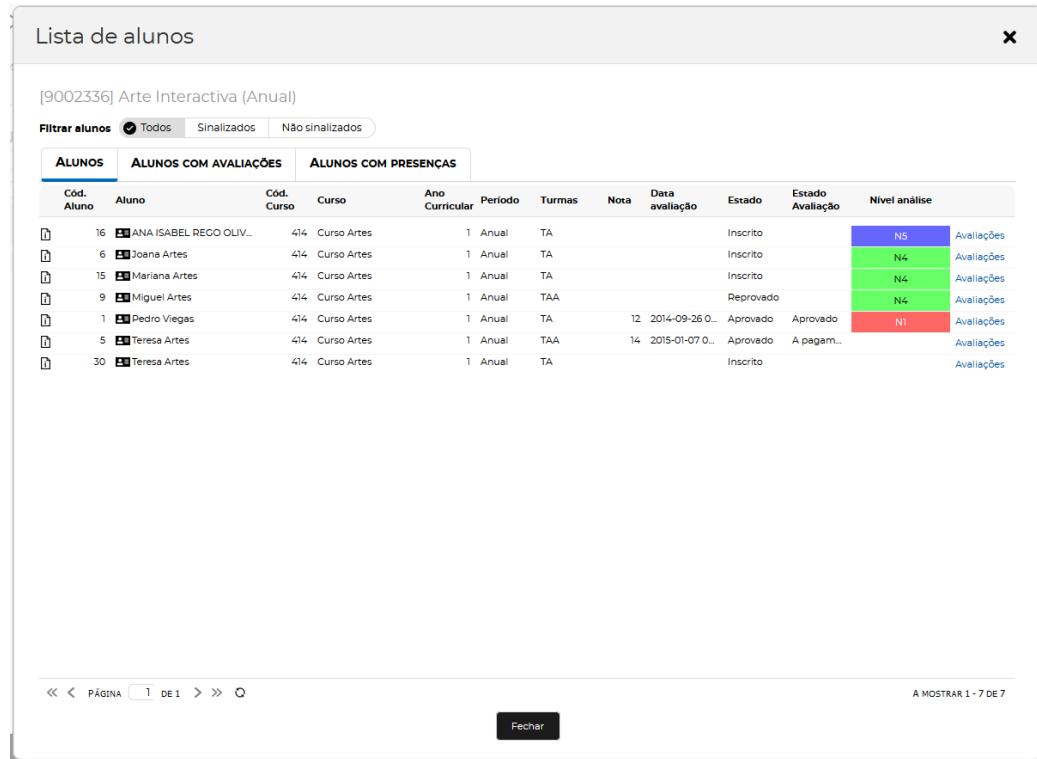


Figura 131 - SI.PREVINA, dialog com os alunos da UC selecionada

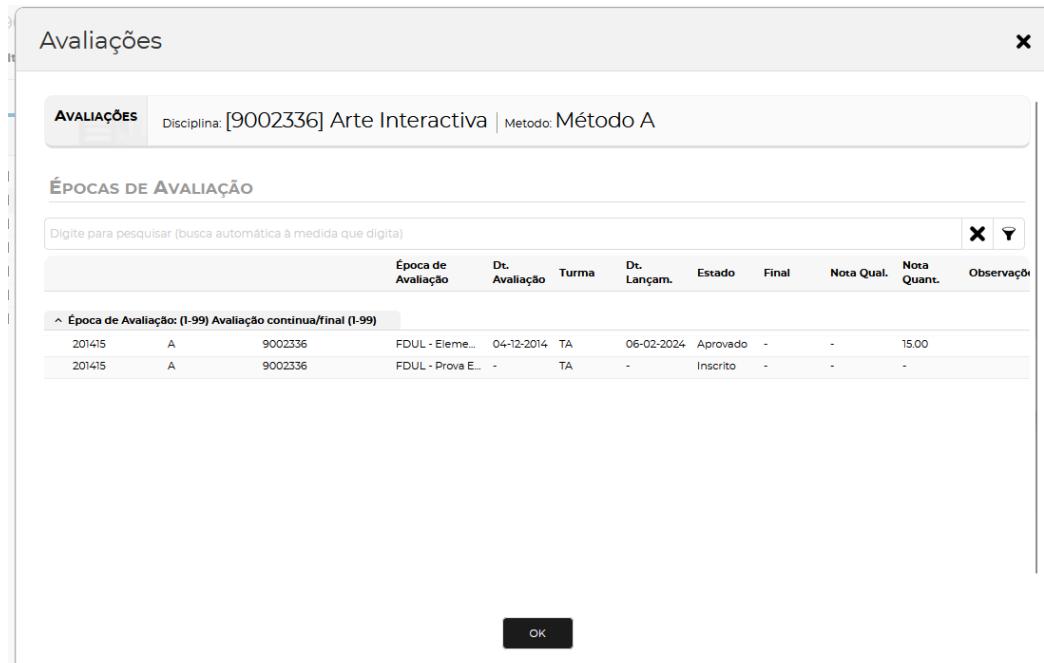


Figura 132 - SI.PREVINA, consulta das avaliações de um só aluno quando selecionada a opção "Avaliações" do dialog com os alunos de uma UC

[9002336] Arte Interactiva (Anual)

Filtrar alunos Todos Sinalizados Não sinalizados

ALUNOS	ALUNOS COM AVALIAÇÕES	ALUNOS COM PRESENÇAS								
Cód. Aluno	Aluno	Nota	Data avaliação	Estado	Estado Avaliação	Nível análise	Avaliação continua/final (1-99)	Avaliação nº2	Avaliação nº3	Avaliação nº4
16	ANA ISABEL REGO OLIV...			Inscrito		N5				
6	Joana Artes			Inscrito		N4		15	Inscrito	
15	Mariana Artes			Inscrito		N4				
9	Miguel Artes			Reprovado		N4				
1	Pedro Viegas	12	2014-09-26 0...	Aprovado	Aprovado	N1			12	
5	Teresa Artes	14	2015-01-07 0...	Aprovado	A pagam...					14
30	Teresa Artes			Inscrito					Inscrito	

« < PÁGINA 1 DE 1 > »

A MOSTRAR 1 - 7 DE 7

[Fechar](#)

Figura 133 - SI.PREVINA, consulta das diferentes épocas de avaliação de todos alunos de uma determinada UC

O docente também pode visualizar informações sobre a assiduidade e a atitude em aula dos alunos referentes às últimas quatro semanas, **Figura 134**. No entanto, se essas semanas não coincidirem com semanas do período do ano letivo da análise a ser visualizada (caso esteja a ser visualizada uma análise antiga), serão apresentadas as informações das últimas quatro semanas do período letivo.

Lista de alunos										
[9002336] Arte Interactiva (Anual)										
Filtrar alunos										
ALUNOS			ALUNOS COM AVALIAÇÕES		ALUNOS COM PRESENÇAS					
Cód. Aluno	Aluno	Nível análise	Semana 31/8 a 6/9		Semana 7/9 a 13/9		Semana 14/9 a 20/9		Semana 21/9 a 27/9	
			Assiduidade	Atitude	Assiduidade	Atitude	Assiduidade	Atitude	Assiduidade	Atitude
16	ANA ISABEL REGO OLIV...	N5					2 (67%)	Atento	0 (0%)	
6	Joana Artes	N4					0 (0%)		0 (0%)	
15	Mariana Artes	N4					0 (0%)		0 (0%)	
9	Miguel Artes	N4					0 (0%)		0 (0%)	
1	Pedro Viegas	NI					0 (0%)		0 (0%)	
5	Teresa Artes						0 (0%)		0 (0%)	
30	Teresa Artes						0 (0%)		0 (0%)	

Figura 134 - PREVINA, consulta da assiduidade e atitude média das últimas quatro semanas de cada aluno de uma determinada UC

Quando o utilizador seleciona a opção "Alunos" no menu de navegação principal, **Figura 129**, poderá visualizar todos os alunos que o seu perfil permite (ver **Figura 135**). Além disso, os alunos podem ser filtrados por "Todos", "Sinalizados" ou "Não sinalizados". De forma inversa ao menu de unidades curriculares, o utilizador pode visualizar as UCs de um aluno ao selecionar a ação "Ver UCs". Esta ação abrirá um diálogo com a informação das unidades curriculares em que o aluno se inscreveu, bem como a sua nota final (ver **Figura 136**) caso já tenha sido lançada.

DOCÊNCIA - ALUNOS						
Filtrar alunos		Todos	Sinalizados	Não sinalizados		
Digite para pesquisar (necessita selecionar o botão Pesquisar após digitar)						
Cód. Aluno	Aluno	Cód. Curso	Curso	Ano Curricular	Nível análise	
16	[6] ANA ISABEL REGO OLIVEIRA	414	Curso Artes	1	N5	Ver UCs
12	[6] Carla Artes Saint	414	Curso Artes	1	N5	Ver UCs
23	[6] Catarina Artes	414	Curso Artes	1	N5	Ver UCs
6	[6] Joana Artes	414	Curso Artes	1	N4	Ver UCs
15	[6] Mariana Artes	414	Curso Artes	1	N4	Ver UCs
9	[6] Miguel Artes	414	Curso Artes	1	N4	Ver UCs
22	[6] Nuno Artes	414	Curso Artes	1	N2	Ver UCs
20	[6] Nuria Artes	414	Curso Artes	1	N2	Ver UCs
1	[6] Pedro Viegas	414	Curso Artes	1	N1	Ver UCs
2	[6] Bruno Marketing	412	Curso de Marketing	1		Ver UCs
77	[6] Cátia Artes	412	Curso de Marketing	1		Ver UCs
21	[6] Pedro Marketing	412	Curso de Marketing	2		Ver UCs
32	[6] Pedroso	414	Curso Artes	1		Ver UCs
27	[6] Rodrigo Artes	414	Curso Artes	1		Ver UCs
33	[6] Stevie Wonder	414	Curso Artes	1		Ver UCs
5	[6] Teresa Artes	414	Curso Artes	1		Ver UCs
30	[6] Teresa Artes	414	Curso Artes	1		Ver UCs
4	[6] Tiago Artes	414	Curso Artes	1		Ver UCs

Figura 135 - SI.PREVINA, lista apresentada quando o utilizador seleciona a opção “Alunos”

Lista de UCs do aluno							
[6] Joana Artes							
Curso: Curso Artes							
Cód. Disciplina	Nome	Período	Estado	Turmas	Estado Avaliação	Data avaliação	Nota
9002240	Estágio A	Anual	Aprovado		Aprovado	29/09/2014	14
9002241	Estágio B	Anual	Aprovado		Aprovado	29/09/2014	14
9002242	Estágio C	Anual	Aprovado		Aprovado	29/09/2014	14
9002335	Artes Plásticas	Anual	Inscrito	TA	Inscrito		
9002336	Arte Interactiva	Anual	Inscrito	TA	Inscrito		
9002338	História de Artes	Anual	Aprovado		Aprovado	29/09/2014	10

Figura 136 - SI.PREVINA, dialog de lista de UCs de um determinado aluno

O utilizador dispõe de submenus adicionais de perfil, como "Avaliações", "Lançar notas", "Tutorias", "Docentes" e "Estatísticas". No entanto, como esses submenus foram desenvolvidos por outro colaborador da Digitalis, esses não serão apresentados neste relatório.

Quanto ao segundo ecrã desenvolvido, **Figura 137**, este é acessível quando um funcionário da instituição entra no módulo SI.PREVINA. O ecrã segue a mesma estrutura do ecrã principal do docente, mas não possui divisórias para diferentes perfis, uma vez que o funcionário tem um único perfil. Assim, neste contexto de funcionário, o mesmo pode visualizar todos os alunos e unidades curriculares de todas as instituições. Devido ao facto do ecrã ser muito semelhante ao do docente, apenas é

apresentada a visualização específica do funcionário, conforme ilustrado na **Figura 137**.

Cód. Aluno	Aluno	Cód. Curso	Curso	Ano Curricular	Nível análise
8	Carina IC	444	Informática de Gestão	1	N5
20091233	Joaquim Fernandes Teixeira	9939	Gestão de Sistemas e Computação	5	N5
11233472	Feste aluno ativo em 2 cursos em ano letivo diferentes	1	BIOTECNOLOGIA DAS COISAS	1	N5
6	Ivana IC	444	Informática de Gestão	1	N5
20053202	Lourenço de Mêdiç	8	GESTÃO EMP-TURIS. HOTELEIRAS	1	N5
30	Leanne Dino Zecarão,	9118	Curso Testes	3	N2
20053202	Lucrecia Borgia	8	GESTAO EMP-TURIS. HOTELEIRAS	1	N2
236	Maria Direito	8	GESTAO EMP-TURIS. HOTELEIRAS	1	N1
20053202	ADRIANA ISABEL RAPOSO SILVESTRE	80	CIÉNCIAS DOCUMENTAIS	1	N1
10	Afonso IG	444	Informática de Gestão	1	Ver UCs
20053207	Albert Einstein	8	GESTAO EMP-TURIS. HOTELEIRAS	1	Ver UCs
5	Alexandre IG	444	Informática de Gestão	1	Ver UCs
20	Alice Bio	451	Engenharia Biomedica	1	Ver UCs
23	Alice CCC	446	Ciências e Comunicação	1	Ver UCs
36	Alice CE	442	Gestão de Empresas	1	Ver UCs
9	Alice IG	506	Curso Tecnologias - Mobilidade	1	Ver UCs
14	Alice RI	443	Relações Internacionais	1	Ver UCs
14	Alice RI	445	Relações Internacionais I	1	Ver UCs
1361	ALBINO PAULO PARREIRÃO E GOMES	24	CIÉNCIAS DA COMUNICAÇÃO E DA CULTURA	1	Ver UCs

Figura 137 - SI.PREVINA, ecrã de acompanhamento do utilizador Funcionário

Os dois ecrãs apresentados, **Figura 127** e **Figura 137**, foram todos os que o autor desenvolveu no módulo SI.PREVINA até à data, excluindo outras pequenas contribuições de excertos de código para a aplicação geral. É importante destacar que a apresentação aqui realizada não reflete completamente a complexidade dos ecrãs. Embora possam parecer simples, eles apresentam uma complexidade consideravelmente elevada, devido aos dados que necessitam de ser processados e apresentados. Isso ocorre porque os ecrãs não apenas suportam a realidade do IPCB, mas também são projetados para suportar outras instituições de ensino superior que possam estar interessadas em adquirir e utilizar o módulo SI.PREVINA.

Por fim, é relevante mencionar que o autor, inicialmente, passou por todo um processo de integração e formação no uso da *framework* interna (DIF) e até ao momento continua a aprender sobre todas as funcionalidades e ferramentas que esta tem a oferecer. Ao contrário de *frameworks* bem estabelecidas, como Laravel ou ASP.NET, a DIF não possui comunidades de desenvolvedores dedicadas a responder perguntas frequentes ou com contribuições de código, pois, é uma *framework* desenvolvida e gerida internamente.

9. Conclusões

O trabalho aqui documentado teve como objetivo investigar, explorar e implementar uma solução, com base no uso de técnicas de ML, capaz de sinalizar o nível de risco de abandono e sucesso escolar dos alunos que se matriculam no IPCB pela primeira vez. Perante um aumento da taxa de desistências no ensino superior, sobretudo ao fim do primeiro ano letivo, as instituições começaram a procurar soluções para combater o abandono e insucesso escolar. Nesse âmbito, o IPCB, em 2024, iniciou o desenvolvimento/aplicação do projeto REVUP que tem como objetivo promover o sucesso escolar e combater o abandono ao sinalizar e apoiar, de uma forma mais dedicada, os alunos que enfrentem maior risco de incorrer numa destas situações. Desse modo, o projeto propõe a utilização de modelos de ML para predição deste tipo de situações, uma vez que esta área tem vindo a ter um crescimento bastante grande nos últimos tempos com aplicações a diferentes áreas da sociedade e com resultados significativos.

O trabalho desenvolvido no âmbito deste projeto permitiu investigar o desempenho de modelos de ML para a sinalização de situações de abandono e sucesso escolar combinadas. Além disso, serviu como uma base e primeira versão do desenvolvimento e implementação de um sistema inteligente, baseado em ML, capaz de ser integrado na ferramenta de acompanhamento que está a ser desenvolvida para o IPCB. A partir deste trabalho, várias linhas de investigação poderão ser seguidas e implementadas ao longo dos próximos anos, com o objetivo em melhorar e garantir predições mais precisas.

Numa primeira fase, que resultou do trabalho realizado na UC de Projeto I, foram apresentados os tipos de aprendizagem computacional (ML) existentes, confirmando-se novamente que o problema abordado neste trabalho se enquadra na categoria de classificação, requerendo uma aprendizagem supervisionada. Além disso, foi também mostrada a importância dos dados e a necessidade de serem utilizadas métricas corretas para avaliar o desempenho dos modelos de ML.

A elaboração do estado da arte permitiu com que fossem analisados 11 artigos distintos. Esta análise confirmou que é possível utilizar técnicas de ML para a predição do abandono e/ou insucesso escolar, sendo que todos os estudos demonstraram obter resultados positivos nas predições investigadas. Além disso, observou-se que as predições resultam em melhores resultados quando realizadas no final do primeiro semestre ou ano letivo. Em contraste, as predições, para o sucesso escolar, feitas no momento da matrícula apresentaram resultados preditivos inferiores, devido à falta de dados relevantes, como as notas de testes intercalares ou presenças do aluno.

Posteriormente, com base nos dados fornecidos pelo IPCB para o treino e avaliação dos modelos de ML, foram identificadas seis classes diferentes de predição. Uma análise do número de instâncias por valor em cada uma dessas classes de problemas de classificação revelou um desequilíbrio na representação de cada valor possível. Além disso, constatou-se que a maior parte dos alunos com risco mais elevado (maior

probabilidade desistir e/ou reprovar) ou que acabam por desistir ao fim do primeiro ano, são alunos estrangeiros, podendo-se associar a dificuldade de integração como uma das causas que podem justificar esta observação.

Após a análise dos dados, foram treinados e avaliados os modelos de ML utilizando as técnicas de treino: *holdout* e validação cruzada. Num primeiro trabalho de investigação, observou-se que, para as classes de predição relacionadas com o nível de risco (excluindo o binário), os resultados ficaram aquém das expectativas. Os modelos de ML apresentaram uma média de 70% para a métrica exatidão e 50% para *F1-Score*. Por outro lado, a predição do abandono escolar ou risco binário demonstraram ser a predição mais viável, pois, os diferentes modelos de ML alcançaram uma exatidão próxima de 80% e um F1-Score de 70%, que por sua vez são próximos dos resultados obtidos por outros estudos. É importante salientar que o momento de predição investigado neste trabalho é o ato da matrícula. Tal como foi observado na análise do estado da arte, este é um momento particularmente difícil de aprendizagem por parte dos modelos de ML, pois, apresenta os piores resultados analisados.

Além disso, a análise dos resultados obtidos no treino dos modelos de ML revelou que, para os dados fornecidos, o balanceamento das instâncias do conjunto de treino não resultou em melhorias no desempenho dos vários modelos de ML treinados. Esta observação sugere a necessidade de serem investigados o uso de outros atributos relativos aos dados dos alunos, de forma verificar se o que impacta o desempenho é a qualidade dos dados ou o momento de predição.

Através da análise do estado da arte e os resultados obtidos é possível concluir que os docentes não se devem basear exclusivamente nas predições realizadas pelos modelos de ML. Em vez disso, devem validar as predições, ao estarem atentos ao comportamento e o desempenho do aluno ao longo do ano letivo. Isto porque, como foi evidenciado, os modelos não são perfeitos nas suas predições. Podendo existir alunos que tenham sido mal classificados ou por terem sido considerados alunos de risco, sem o ser (*outliers*), ou porque tenham sido considerados sem risco quando durante um período de validação se verifica necessitarem de um acompanhamento mais dedicado.

Relativamente aos objetivos delineados no início do projeto, é possível afirmar que foram concluídos com sucesso. Adicionalmente, foram alcançados outros objetivos adicionais durante o desenvolvimento deste trabalho, incluindo:

- **Reutilização de pré-processamentos:** foi possível implementar uma solução completa de pré-processamento modular que permite exportar todos os passos realizados e valores aceites. Possibilitando reutilizar o pré-processamento aos dados utilizados numa predição posterior ao treino, evitando erros de atributos ou valores desconhecidos pelo modelo de ML;
- **Analizar o desempenho de modelos de ML para diferentes tipos de predição:** foi possível analisar o desempenho para seis classes de predição distintas e, além disso, foi possível avaliar um algoritmo de sinalização de risco que utiliza dois modelos de ML para efetuar a sua predição;

- **Aplicação web modular:** desenvolveu-se uma aplicação web complexa, capaz de se adaptar a diferentes tipos de predição e contextos. Adicionalmente, permite a adição de novos tipos de predição e modelos de ML através da interface gráfica;
- **Contribuir para o desenvolvimento do modulo SI.PREVINA:** além do trabalho desenvolvido no contexto escolar, foi oferecido um emprego antecipado e a oportunidade de contribuir para o desenvolvimento do novo modulo da aplicação NetP@.

Por último, é possível concluir que, embora os resultados de sinalização do nível de risco tenham ficado aquém do pretendido, para o objetivo delineado pela submedida da DGES [7], cujo se centra na predição do abandono, os resultados foram positivos, sendo observada uma exatidão de 80% para vários modelos de ML, com destaque para o algoritmo *Gradient Boosting*. Além disso, é importante realçar, que caso seja explorado o uso do sistema de sinalização de risco proposto, as suas predições devem ser devidamente validadas pelos docentes do IPCB, face aos resultados obtidos.

9.1. Trabalho futuro

Perante o trabalho desenvolvido, reconhece-se que este se apresenta como um ponto de partida, existindo vários caminhos possíveis para a melhoria do sistema de sinalização de risco dos alunos do IPCB aqui apresentado. Uma sugestão de melhoria é a realização de uma análise mais detalhada dos dados existentes, com o objetivo de refinar as codificações dos atributos e verificar se existem representações que possam levar a melhores resultados. Adicionalmente, destaca-se que pode ser explorada a componente de combinação de atributos (*feature engineering*). Além disso, interessa realçar que com o passar dos anos e a recolha de mais dados de treino, será possível fornecer mais exemplos aos modelos de ML, permitindo-lhes reforçar padrões existentes ou identificar novos padrões.

Complementarmente, está planeado concluir o desenvolvimento da aplicação web. Apesar de estar apta a ambientes de produção, ainda falta a implementação de determinadas funcionalidades, como a edição de dados. Além disso, pretende-se documentar a aplicação de forma a facilitar a sua adoção e modificação por outros investigadores e/ou programadores. O objetivo final é disponibilizar a aplicação num repositório de código aberto, com intuito de contribuir para a comunidade de código aberto.

Um complemento futuro ao trabalho desenvolvido, e um desejo expresso pela professora orientadora Ana Paula Silva durante a realização deste projeto, é a possibilidade de serem extraídas as regras de predição dos modelos de ML. A extração de regras dos modelos de ML poderá ser fundamental para uma melhor compreensão dos dados, pois, permitirá verificar como os modelos de ML chegam às suas predições ao observar que padrões os mesmos aprenderam. Isto porque, os modelos de ML têm

a capacidade de aprender padrões que os humanos têm tendência a ignorar ou desconhecer.

Adicionalmente, é importante salientar que, após o desenvolvimento deste trabalho, o autor expressa o desejo e a vontade de continuar a colaborar com o IPCB, em particular com os docentes responsáveis pelo projeto REVUP, com o objetivo de continuar a contribuir para uma melhoria contínua do sistema de sinalização de alunos em risco de abandono e/ou insucesso escolar. Por fim, destaca-se que o autor continuará a contribuir para o desenvolvimento do módulo SI.PREVINA da aplicação NetP@, dado que é um dos integrantes da equipa de desenvolvimento da Digitalis. Essa colaboração irá prolongar-se, no mínimo, até ao segundo semestre do ano letivo de 2024/2025, altura em que está prevista a segunda entrega de funcionalidades do módulo.

Referências

- [1] “Abandono está a aumentar no ensino superior público, sobretudo no interior.” Accessed: Sep. 13, 2024. [Online]. Available: <https://www.dn.pt/2117681773/abandono-esta-a-aumentar-no-ensino-superior-publico-sobretudo-no-interior/>
- [2] “Taxa de desistência do ensino superior é a mais alta em oito anos – ECO.” Accessed: Sep. 13, 2024. [Online]. Available: <https://eco.sapo.pt/2024/06/20/taxa-de-desistencia-do-ensino-superior-e-a-mais-alta-em-oito-anos/>
- [3] “Abandono aumenta no Ensino Superior e já chega aos 26,9% nos cursos de dois anos.” Accessed: Sep. 13, 2024. [Online]. Available: <https://www.jn.pt/5518427853/abandono-aumenta-no-ensino-superior-e-ja-chega-aos-269-nos-cursos-de-dois-anos/>
- [4] “Dados e Estatísticas de Cursos Superiores.” Accessed: Sep. 13, 2024. [Online]. Available: <https://infocursos.pt/>
- [5] “IPCB promove sessão de acolhimento e integração de estudantes internacionais | Instituto Politécnico de Castelo Branco.” Accessed: Sep. 15, 2024. [Online]. Available: <https://www.ipcb.pt/ipcb-promove-sessao-de-acolhimento-e-integracao-de-estudantes-internacionais>
- [6] “Estudantes abandonam cada vez mais o ensino superior público após o primeiro ano – Observador.” Accessed: Sep. 13, 2024. [Online]. Available: <https://observador.pt/2024/03/03/estudantes-abandonam-cada-vez-mais-o-ensino-superior-publico-apos-o-primeiro-ano/>
- [7] “05. Inovação e Modernização Pedagógica no Ensino Superior - Programa de promoção de sucesso e redução de abandono no ensino superior (Aviso de Abertura de Concurso N.º 05/C06-i07/2023) - PRR - Recuperar Portugal.” Accessed: Sep. 15, 2024. [Online]. Available: <https://recuperarportugal.gov.pt/candidatura/05-inovacao-e-modernizacao-pedagogica-no-ensino-superior-programa-de-promocao-de-sucesso-e-reducao-de-abandono-no-ensino-superior-aviso-de-abertura-de-concurso-n-o-05-c06-i07-2023/>
- [8] “IPCB com 300.000€ para combate ao abandono e insucesso | Instituto Politécnico de Castelo Branco.” Accessed: Sep. 13, 2024. [Online]. Available: <https://www.ipcb.pt/ipcb-com-300000eu-para-combate-ao-abandono-e-insucesso>
- [9] Aurélien Géron, “Hands-on machine learning with Scikit-Learn, Keras and TensorFlow: concepts, tools, and techniques to build intelligent systems,” *O'Reilly Media*, p. 851, 2019, Accessed: Nov. 29, 2023. [Online]. Available:

<https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/>

- [10] A. M. Turing, "COMPUTING MACHINERY AND INTELLIGENCE," 1950. Accessed: Nov. 29, 2023. [Online]. Available: <https://redirect.cs.umbc.edu/courses/471/papers/turing.pdf>
- [11] S. J. (Stuart J. Russell, "Artificial intelligence : a modern approach / Stuart J. Russell and Peter Norvig ; contributing writers: Ming-Wei Chang [and 8 others].," *Artificial intelligence : a modern approach*, pp. 1–1115, 2020, Accessed: Nov. 29, 2023. [Online]. Available: <https://aima.cs.berkeley.edu/>
- [12] "A Very Short History Of Artificial Intelligence (AI)." Accessed: Nov. 29, 2023. [Online]. Available: <https://www.forbes.com/sites/gilpress/2016/12/30/a-very-short-history-of-artificial-intelligence-ai/>
- [13] "Artificial Intelligence (AI) Coined at Dartmouth | Dartmouth." Accessed: Nov. 29, 2023. [Online]. Available: <https://home.dartmouth.edu/about/artificial-intelligence-ai-coined-dartmouth>
- [14] "What is Artificial Intelligence (AI) ? | IBM." Accessed: Nov. 29, 2023. [Online]. Available: <https://www.ibm.com/topics/artificial-intelligence>
- [15] "What is Artificial Intelligence (AI)? | Oracle." Accessed: Nov. 29, 2023. [Online]. Available: <https://www.oracle.com/artificial-intelligence/what-is-ai/>
- [16] "6 Major Sub-Fields of Artificial Intelligence | by Rancho Labs | Medium." Accessed: Nov. 29, 2023. [Online]. Available: <https://rancholabs.medium.com/6-major-sub-fields-of-artificial-intelligence-77f6a5b28109>
- [17] "AI Myth Busters: Who You Gonna Call? Umbrella, Of Course | AppFutura." Accessed: Nov. 29, 2023. [Online]. Available: <https://www.appfutura.com/blog/ai-myth-busters-who-you-gonna-call-umbrella-of-course/>
- [18] "How Companies Are Already Using AI." Accessed: Nov. 29, 2023. [Online]. Available: <https://hbr.org/2017/04/how-companies-are-already-using-ai>
- [19] "Understanding the four types of AI, from reactive robots to self-aware beings." Accessed: Nov. 29, 2023. [Online]. Available: <https://theconversation.com/understanding-the-four-types-of-ai-from-reactive-robots-to-self-aware-beings-67616>
- [20] "Kasparov vs. Deep Blue | The Match That Changed History - Chess.com." Accessed: Nov. 29, 2023. [Online]. Available: <https://www.chess.com/article/view/deep-blue-kasparov-chess#kasparov-deep-blue-1997-rematch>

- [21] Y. C. Goh, X. Q. Cai, W. Theseira, G. Ko, and K. A. Khor, "Evaluating human versus machine learning performance in classifying research abstracts," *Scientometrics*, vol. 125, no. 2, pp. 1197–1212, Nov. 2020, doi: 10.1007/S11192-020-03614-2/TABLES/5.
- [22] "Netflix Research." Accessed: Nov. 29, 2023. [Online]. Available: <https://research.netflix.com/research-area/machine-learning>
- [23] "How does DeepL work?" Accessed: Nov. 29, 2023. [Online]. Available: <https://www.deepl.com/en/blog/how-does-deepl-work>
- [24] "Giving Lens New Reading Capabilities in Google Go – Google Research Blog." Accessed: Nov. 29, 2023. [Online]. Available: <https://blog.research.google/2019/09/giving-lens-new-reading-capabilities-in.html>
- [25] "A primer on machine learning for fraud detection." Accessed: Nov. 29, 2023. [Online]. Available: <https://stripe.com/blog/a-primer-on-machine-learning-for-fraud-detection>
- [26] "6 ways Gmail uses AI features to help you save time." Accessed: Nov. 29, 2023. [Online]. Available: <https://blog.google/products/gmail/gmail-ai-features/>
- [27] "How Does Facebook Use Machine Learning to Deliver Ads? | Meta for Business." Accessed: Nov. 29, 2023. [Online]. Available: <https://www.facebook.com/business/news/good-questions-real-answers-how-does-facebook-use-machine-learning-to-deliver-ads>
- [28] "A guide to the types of machine learning algorithms | SAS UK." Accessed: Nov. 29, 2023. [Online]. Available: https://www.sas.com/en_gb/insights/articles/analytics/machine-learning-algorithms.html
- [29] "A Beginner's Guide to Supervised Machine Learning Algorithms | by Soner Yıldırım | Towards Data Science." Accessed: Jan. 27, 2024. [Online]. Available: <https://towardsdatascience.com/a-beginners-guide-to-supervised-machine-learning-algorithms-6e7cd9f177d5>
- [30] "Distance metrics and K-Nearest Neighbor (KNN) | by Luigi Fiori | Medium." Accessed: Jan. 27, 2024. [Online]. Available: <https://medium.com/@luigi.fiori.lf0303/distance-metrics-and-k-nearest-neighbor-knn-1b840969c0f4>
- [31] "Multilayer Perceptron Explained with a Real-Life Example and Python Code: Sentiment Analysis | by Carolina Bento | Towards Data Science." Accessed: Jan. 27, 2024. [Online]. Available: <https://towardsdatascience.com/multilayer-perceptron-explained-with-a-real-life-example-and-python-code-sentiment-analysis-cb408ee93141>

- [32] “Types of Machine Learning - Javatpoint.” Accessed: Nov. 29, 2023. [Online]. Available: <https://www.javatpoint.com/types-of-machine-learning>
- [33] “3 Types of Machine Learning You Should Know | Coursera.” Accessed: Nov. 29, 2023. [Online]. Available: <https://www.coursera.org/articles/types-of-machine-learning>
- [34] “What is Unsupervised Learning? | IBM.” Accessed: Nov. 29, 2023. [Online]. Available: <https://www.ibm.com/topics/unsupervised-learning>
- [35] “Machine Learning Guide Podcast.” Accessed: Nov. 29, 2023. [Online]. Available: <https://ocdevel.com/mlg>
- [36] “What is reinforcement learning? - University of York.” Accessed: Nov. 29, 2023. [Online]. Available: <https://online.york.ac.uk/what-is-reinforcement-learning/>
- [37] “Reinforcement Learning 101. Learn the essentials of Reinforcement... | by Shweta Bhatt | Towards Data Science.” Accessed: Nov. 29, 2023. [Online]. Available: <https://towardsdatascience.com/reinforcement-learning-101-e24b50e1d292>
- [38] M. L. Littman, “Markov Decision Processes,” *International Encyclopedia of the Social & Behavioral Sciences*, pp. 9240–9242, 2001, doi: 10.1016/B0-08-043076-7/00614-8.
- [39] “An Introduction to Q-Learning: A Tutorial For Beginners | DataCamp.” Accessed: Nov. 29, 2023. [Online]. Available: <https://www.datacamp.com/tutorial/introduction-q-learning-beginner-tutorial>
- [40] A. Glassner, “Volume 1 DEEP LEARNING: From Basics to Practice,” 2018. [Online]. Available: www.glassner.com
- [41] F. Chollet, “Machine learning 분야 소개 및 주요 방법론 학습 기본 machine learning 알고리즘에 대한 이해 및 응용 관련 최신 연구 동향 습득,” *Mach Learn*, vol. 45, no. 13, pp. 40–48, 2017, Accessed: Nov. 29, 2023. [Online]. Available: <https://www.manning.com/books/deep-learning-with-python>
- [42] “What is Deep Learning? | IBM.” Accessed: Nov. 29, 2023. [Online]. Available: <https://www.ibm.com/topics/deep-learning>
- [43] J. Howard and S. Gugger, “Tabular Modeling Deep Dive,” *Deep Learning for Coders with fastai and PyTorch*, 2020, Accessed: Nov. 29, 2023. [Online]. Available: <https://www.oreilly.com/library/view/deep-learning-for/9781492045519/>
- [44] “data noun - Definition, pictures, pronunciation and usage notes | Oxford Advanced Learner’s Dictionary at OxfordLearnersDictionaries.com.” Accessed: Nov. 29, 2023. [Online]. Available: <https://www.oxfordlearnersdictionaries.com/definition/english/data>

- [45] R. Wilka, R. Landy, and S. McKinney, "How Machines Learn: Where Do Companies Get Data for Machine Learning and What Licenses Do They Need?," *Washington Journal of Law, Technology & Arts*, vol. 13, no. 3, Apr. 2018, Accessed: Nov. 29, 2023. [Online]. Available: <https://digitalcommons.law.uw.edu/wjltv/vol13/iss3/2>
- [46] "Kaggle: Your Machine Learning and Data Science Community." Accessed: Nov. 29, 2023. [Online]. Available: <https://www.kaggle.com/>
- [47] "Guide to Data Collection for Machine Learning." Accessed: Nov. 29, 2023. [Online]. Available: <https://www.altextsoft.com/blog/data-collection-machine-learning/>
- [48] "Etapa 1: coletar dados | Machine Learning | Google for Developers." Accessed: Nov. 29, 2023. [Online]. Available: <https://developers.google.com/machine-learning/guides/text-classification/step-1?hl=pt-br>
- [49] "ML | Introduction to Data in Machine Learning - GeeksforGeeks." Accessed: Nov. 29, 2023. [Online]. Available: <https://www.geeksforgeeks.org/ml-introduction-data-machine-learning/>
- [50] "Problems in Machine Learning Models? Check your Data First | by Dhruv Sharma | Towards Data Science." Accessed: Nov. 29, 2023. [Online]. Available: <https://towardsdatascience.com/problems-in-machine-learning-models-check-your-data-first-f6c2c88c5ec2>
- [51] "Fantastic Data Quality Issues and Where to Find Them | Towards Data Science." Accessed: Nov. 29, 2023. [Online]. Available: <https://towardsdatascience.com/data-quality-issues-that-kill-your-machine-learning-models-961591340b40>
- [52] "What are the key privacy concerns associated with machine learning?" Accessed: Nov. 29, 2023. [Online]. Available: <https://www.linkedin.com/pulse/what-key-privacy-concerns-associated-machine->
- [53] F. E. Grubbs, "Procedures for detecting outlying observations in samples," *Technometrics*, vol. 11, no. 1, pp. 1–21, 1969, doi: 10.1080/00401706.1969.10490657.
- [54] "7 Major Challenges Faced By Machine Learning Professionals - GeeksforGeeks." Accessed: Nov. 29, 2023. [Online]. Available: <https://www.geeksforgeeks.org/7-major-challenges-faced-by-machine-learning-professionals/>
- [55] "A Guide to Data Splitting in Machine Learning | by Data Science Wizards | Medium." Accessed: Nov. 29, 2023. [Online]. Available: <https://medium.com/@datasciencewizards/a-guide-to-data-splitting-in-machine-learning-49a959c95fa1>

- [56] "How to Evaluate your Machine Learning Model. | Analytics Vidhya." Accessed: Nov. 29, 2023. [Online]. Available: <https://medium.com/analytics-vidhya/how-to-evaluate-your-machine-learning-model-76a7671e9f2e>
- [57] "Evaluating a machine learning model." Accessed: Nov. 29, 2023. [Online]. Available: <https://www.jeremyjordan.me/evaluating-a-machine-learning-model/>
- [58] "Confusion Matrix." Accessed: Nov. 29, 2023. [Online]. Available: <https://devopedia.org/confusion-matrix>
- [59] "Confusion Matrix: Performance Evaluator of Classifier | by Simran Panthi | FAUN—Developer Community 🌐." Accessed: Nov. 29, 2023. [Online]. Available: <https://faun.pub/confusion-matrix-performance-evaluator-of-classifier-ac60325c88bb>
- [60] "Machine Learning Model Evaluation - GeeksforGeeks." Accessed: Nov. 29, 2023. [Online]. Available: <https://www.geeksforgeeks.org/machine-learning-model-evaluation/>
- [61] "PRISMA." Accessed: Nov. 29, 2023. [Online]. Available: <http://prisma-statement.org/>
- [62] "Scopus preview - Scopus - Welcome to Scopus." Accessed: Aug. 06, 2024. [Online]. Available: <https://www.scopus.com/home.uri>
- [63] "ACM partnerships with indexing services." Accessed: Aug. 06, 2024. [Online]. Available: <https://authors.acm.org/journals/journals-indexing-list>
- [64] "Abstracting & Indexing (A&I) Databases - IEEE Author Center Journals." Accessed: Aug. 06, 2024. [Online]. Available: <https://journals.ieeeauthorcenter.ieee.org/when-your-article-is-published/abstracting-indexing-ai-databases/>
- [65] "Bibliographic databases." Accessed: Aug. 06, 2024. [Online]. Available: <https://www.springeropen.com/get-published/indexing-archiving-and-access-to-data/new-content-item>
- [66] "b-on." Accessed: Nov. 29, 2023. [Online]. Available: <https://www.b-on.pt/>
- [67] H. Zeineddine, U. Braendle, and A. Farah, "Enhancing prediction of student success: Automated machine learning approach," *Computers and Electrical Engineering*, vol. 89, Jan. 2021, doi: 10.1016/J.COMPELECENG.2020.106903.
- [68] M. Yağcı, "Educational data mining: prediction of students' academic performance using machine learning algorithms," *Smart Learning Environments*, vol. 9, no. 1, Dec. 2022, doi: 10.1186/S40561-022-00192-Z.
- [69] C. C. Gray and D. Perkins, "Utilizing early engagement and machine learning to predict student outcomes," *Comput Educ*, vol. 131, pp. 22–32, Apr. 2019, doi: 10.1016/J.COMPEDU.2018.12.006.

- [70] J. Y. Chung and S. Lee, "Dropout early warning systems for high school students using machine learning," *Child Youth Serv Rev*, vol. 96, pp. 346–353, Jan. 2019, doi: 10.1016/J.CHILDYOUTH.2018.11.030.
- [71] A. S. Hoffait and M. Schyns, "Early detection of university students with potential difficulties," *Decis Support Syst*, vol. 101, pp. 1–11, Sep. 2017, doi: 10.1016/J.DSS.2017.05.003.
- [72] S. N. Liao, D. Zingaro, K. Thai, C. Alvarado, W. G. Griswold, and L. Porter, "A robust machine learning technique to predict low-performing students," *ACM Transactions on Computing Education*, vol. 19, no. 3, Jan. 2019, doi: 10.1145/3277569.
- [73] S. C. Tsai, C. H. Chen, Y. T. Shiao, J. S. Ciou, and T. N. Wu, "Precision education with statistical learning and deep learning: a case study in Taiwan," *International Journal of Educational Technology in Higher Education*, vol. 17, no. 1, Dec. 2020, doi: 10.1186/S41239-020-00186-2.
- [74] A. Nabil, M. Seyam, and A. Abou-Elfetouh, "Prediction of Students' Academic Performance Based on Courses' Grades Using Deep Neural Networks," *IEEE Access*, vol. 9, pp. 140731–140746, 2021, doi: 10.1109/ACCESS.2021.3119596.
- [75] J. Niyogisubizo, L. Liao, E. Nziyumva, E. Murwanashyaka, and P. C. Nshimyumukiza, "Predicting student's dropout in university classes using two-layer ensemble machine learning approach: A novel stacked generalization," *Computers and Education: Artificial Intelligence*, vol. 3, Jan. 2022, doi: 10.1016/J.CAEAI.2022.100066.
- [76] A. J. Fernandez-Garcia, J. C. Preciado, F. Melchor, R. Rodriguez-Echeverria, J. M. Conejero, and F. Sanchez-Figueroa, "A real-life machine learning experience for predicting university dropout at different stages using academic data," *IEEE Access*, vol. 9, pp. 133076–133090, 2021, doi: 10.1109/ACCESS.2021.3115851.
- [77] V. L. Miguéis, A. Freitas, P. J. V. Garcia, and A. Silva, "Early segmentation of students according to their academic performance: A predictive modelling approach," *Decis Support Syst*, vol. 115, pp. 36–51, Nov. 2018, doi: 10.1016/J.DSS.2018.09.001.
- [78] M. Ouahi, S. Khoulji, and M. L. Kerkeb, "Advancing Sustainable Learning Environments: A Literature Review on Data Encoding Techniques for Student Performance Prediction using Deep Learning Models in Education", doi: 10.1051/e3sconf/202447700074.
- [79] "What is Python? Executive Summary | Python.org." Accessed: Dec. 22, 2023. [Online]. Available: <https://www.python.org/doc/essays/blurb/>
- [80] "Why Python for Machine Learning? - Python Tutorial." Accessed: Dec. 22, 2023. [Online]. Available: <https://pythonbasics.org/why-python-for-machine-learning/>

- [81] “4 Reasons Why is Python Used for Machine Learning | Inoxoft.” Accessed: Dec. 22, 2023. [Online]. Available: <https://inoxoft.com/blog/why-use-python-for-machine-learning/>
- [82] “Style guide — numpydoc v1.7.0rc0.dev0 Manual.” Accessed: Dec. 22, 2023. [Online]. Available: <https://numpydoc.readthedocs.io/en/latest/format.html>
- [83] “What is Jupyter Notebook? | Domino Data Lab.” Accessed: Dec. 22, 2023. [Online]. Available: <https://domino.ai/data-science-dictionary/jupyter-notebook>
- [84] “scikit-learn: machine learning in Python — scikit-learn 1.3.2 documentation.” Accessed: Dec. 22, 2023. [Online]. Available: <https://scikit-learn.org/stable/>
- [85] “pandas - Python Data Analysis Library.” Accessed: Dec. 22, 2023. [Online]. Available: <https://pandas.pydata.org/about/index.html>
- [86] “pandas documentation — pandas 2.1.4 documentation.” Accessed: Dec. 22, 2023. [Online]. Available: <https://pandas.pydata.org/docs/>
- [87] “What is NumPy? — NumPy v1.26 Manual.” Accessed: Dec. 22, 2023. [Online]. Available: <https://numpy.org/doc/stable/user/whatisnumpy.html>
- [88] “NumPy documentation — NumPy v1.26 Manual.” Accessed: Dec. 22, 2023. [Online]. Available: <https://numpy.org/doc/stable/>
- [89] “Matplotlib — Visualization with Python.” Accessed: Dec. 23, 2023. [Online]. Available: <https://matplotlib.org/>
- [90] “Plot types — Matplotlib 3.8.2 documentation.” Accessed: Dec. 23, 2023. [Online]. Available: https://matplotlib.org/stable/plot_types/index
- [91] “Matplotlib documentation — Matplotlib 3.8.2 documentation.” Accessed: Dec. 23, 2023. [Online]. Available: <https://matplotlib.org/stable/>
- [92] “An introduction to seaborn — seaborn 0.13.0 documentation.” Accessed: Dec. 22, 2023. [Online]. Available: <https://seaborn.pydata.org/tutorial/introduction.html>
- [93] “Matplotlib vs. seaborn vs. Plotly vs. MATLAB vs. ggplot2 vs. pandas - Ritzo Articles.” Accessed: Dec. 22, 2023. [Online]. Available: <https://ritzo.co/articles/matplotlib-vs-seaborn-vs-plotly-vs-MATLAB-vs-ggplot2-vs-pandas/>
- [94] “API reference — seaborn 0.13.0 documentation.” Accessed: Dec. 22, 2023. [Online]. Available: <https://seaborn.pydata.org/api.html>
- [95] “PyCharm: the Python IDE for Professional Developers by JetBrains.” Accessed: Dec. 21, 2023. [Online]. Available: <https://www.jetbrains.com/pycharm/>
- [96] “PyCharm Solutions for Web Development & Data Science.” Accessed: Dec. 21, 2023. [Online]. Available: <https://www.jetbrains.com/pycharm/use-cases/>

- [97] "Scientific & Data Science Tools - Features | PyCharm." Accessed: Dec. 21, 2023. [Online]. Available: https://www.jetbrains.com/pycharm/features/scientific_tools.html
- [98] "PyCharm IDE Integrations | JetBrains." Accessed: Dec. 21, 2023. [Online]. Available: <https://www.jetbrains.com/pycharm/integrations/>
- [99] "WebStorm: The JavaScript and TypeScript IDE, by JetBrains." Accessed: Aug. 06, 2024. [Online]. Available: <https://www.jetbrains.com/webstorm/>
- [100] "WebStorm: Features." Accessed: Aug. 06, 2024. [Online]. Available: <https://www.jetbrains.com/webstorm/features/>
- [101] "All Developer Tools and Products by JetBrains." Accessed: Aug. 06, 2024. [Online]. Available: <https://www.jetbrains.com/products/>
- [102] "Olá, Mundo - GitHub Docs." Accessed: Dec. 21, 2023. [Online]. Available: <https://docs.github.com/pt/get-started/quickstart/hello-world>
- [103] "Weka – Graphical User Interference Way To Learn Machine Learning." Accessed: Aug. 06, 2024. [Online]. Available: <https://www.analyticsvidhya.com/learning-paths-data-science-business-analytics-business-intelligence-big-data/weka-gui-learn-machine-learning/>
- [104] "Weka (software) - Wikipedia." Accessed: Aug. 06, 2024. [Online]. Available: [https://en.wikipedia.org/wiki/Weka_\(software\)](https://en.wikipedia.org/wiki/Weka_(software))
- [105] "SvelteKit • Web development, streamlined." Accessed: Aug. 06, 2024. [Online]. Available: <https://kit.svelte.dev/>
- [106] "Introduction • Docs • SvelteKit." Accessed: Aug. 06, 2024. [Online]. Available: <https://kit.svelte.dev/docs/introduction>
- [107] "TypeScript: JavaScript With Syntax For Types." Accessed: Aug. 06, 2024. [Online]. Available: <https://www.typescriptlang.org/>
- [108] "TypeScript - MDN Web Docs Glossary: Definitions of Web-related terms | MDN." Accessed: Aug. 06, 2024. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Glossary/TypeScript>
- [109] "TypeScript Introduction." Accessed: Aug. 06, 2024. [Online]. Available: https://www.w3schools.com/typescript/typescript_intro.php
- [110] "FastAPI - FastAPI." Accessed: Aug. 06, 2024. [Online]. Available: <https://fastapi.tiangolo.com/>
- [111] "Flask-RESTful — Flask-RESTful 0.3.10 documentation." Accessed: Aug. 06, 2024. [Online]. Available: <https://flask-restful.readthedocs.io/en/latest/>
- [112] "Comparing Django, Flask, and FastAPI as Python Web Frameworks - Search My Expert Blog." Accessed: Aug. 06, 2024. [Online]. Available: <https://blog.searchmyexpert.com/python-web-frameworks-comparison/>

- [113] “PostgreSQL: About.” Accessed: Aug. 06, 2024. [Online]. Available: <https://www.postgresql.org/about/>
- [114] “historical trend of the popularity ranking of database management systems.” Accessed: Aug. 06, 2024. [Online]. Available: https://db-engines.com/en/ranking_trend
- [115] “Stack Overflow Developer Survey 2023.” Accessed: Aug. 06, 2024. [Online]. Available: <https://survey.stackoverflow.co/2023/#section-most-popular-technologies-databases>
- [116] “Python Package Introduction — xgboost 2.0.3 documentation.” Accessed: Jan. 26, 2024. [Online]. Available: https://xgboost.readthedocs.io/en/stable/python/python_intro.html
- [117] “Welcome to LightGBM’s documentation! — LightGBM 4.0.0 documentation.” Accessed: Jan. 26, 2024. [Online]. Available: <https://lightgbm.readthedocs.io/en/stable/>
- [118] “imbalanced-learn documentation — Version 0.12.3.” Accessed: Sep. 12, 2024. [Online]. Available: <https://imbalanced-learn.org/stable/>
- [119] “Welcome to skops’s documentation! — skops 0.9 documentation.” Accessed: Jan. 26, 2024. [Online]. Available: <https://skops.readthedocs.io/en/stable/>
- [120] “The Psycopg 3 project — Psycopg.” Accessed: Aug. 06, 2024. [Online]. Available: <https://www.psycopg.org/psycopg3/>
- [121] “Tailwind CSS - Rapidly build modern websites without ever leaving your HTML.” Accessed: Aug. 06, 2024. [Online]. Available: <https://tailwindcss.com/>
- [122] “Imbalanced-Learn module in Python - GeeksforGeeks.” Accessed: Sep. 12, 2024. [Online]. Available: <https://www.geeksforgeeks.org/imbalanced-learn-module-in-python/>
- [123] “9. Model persistence — scikit-learn 1.4.0 documentation.” Accessed: Jan. 29, 2024. [Online]. Available: https://scikit-learn.org/stable/model_persistence.html
- [124] “Configuration - Tailwind CSS.” Accessed: Aug. 06, 2024. [Online]. Available: <https://tailwindcss.com/docs/configuration>
- [125] “Optimizing for Production - Tailwind CSS.” Accessed: Aug. 06, 2024. [Online]. Available: <https://tailwindcss.com/docs/optimizing-for-production>
- [126] “centraldedados/codigos_postais: Códigos postais em Portugal.” Accessed: Aug. 15, 2024. [Online]. Available: https://github.com/centraldedados/codigos_postais
- [127] “miguelmagueijo/Portugal-Postal-Codes: Extract Portugal postal codes from a old database of CTT. Based on:

- [https://github.com/centraldedados/codigos_postais.](https://github.com/centraldedados/codigos_postais)" Accessed: Aug. 15, 2024. [Online]. Available: <https://github.com/miguelmagueijo/Portugal-Postal-Codes>
- [128] "What is the Difference between OrdinalEncoder and LabelEncoder - GeeksforGeeks." Accessed: Aug. 16, 2024. [Online]. Available: <https://www.geeksforgeeks.org/what-is-the-difference-between-ordinalencoder-and-labelencoder/>
- [129] "Quadro de Qualificações | DGES." Accessed: Aug. 16, 2024. [Online]. Available: https://www.dges.gov.pt/pt/quadro_qualificacoes
- [130] "Beyond One-Hot: an exploration of categorical variables - KDnuggets." Accessed: Aug. 16, 2024. [Online]. Available: <https://www.kdnuggets.com/2015/12/beyond-one-hot-exploration-categorical-variables.html>
- [131] "OneHotEncoder — scikit-learn 1.5.1 documentation." Accessed: Sep. 03, 2024. [Online]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html>
- [132] "K-Fold Cross Validation - Ultralytics YOLOv8 Docs." Accessed: Jan. 26, 2024. [Online]. Available: <https://docs.ultralytics.com/pt/guides/kfold-cross-validation/>
- [133] "2. Over-sampling — Version 0.12.3." Accessed: Sep. 12, 2024. [Online]. Available: https://imbalanced-learn.org/stable/over_sampling.html#from-random-over-sampling-to-smote-and-adasyn
- [134] "Smote for Imbalanced Classification with Python, Technique." Accessed: Sep. 12, 2024. [Online]. Available: <https://www.analyticsvidhya.com/blog/2020/10/overcoming-class-imbalance-using-smote-techniques/>
- [135] "Overcoming Class Imbalance with SMOTE: How to Tackle Imbalanced Datasets in Machine Learning - Train in Data's Blog." Accessed: Sep. 12, 2024. [Online]. Available: <https://www.blog.trainindata.com/overcoming-class-imbalance-with-smote/>
- [136] "regression - MAD vs RMSE vs MAE vs MSLE vs R²: When to use which? - Data Science Stack Exchange." Accessed: Sep. 14, 2024. [Online]. Available: <https://datascience.stackexchange.com/questions/42760/mad-vs-rmse-vs-mae-vs-msle-vs-r%2B2-when-to-use-which>
- [137] "Regression Model Evaluation Metrics: R-Squared, Adjusted R-Squared, MSE, RMSE, and MAE | by Brandon Wohlwend | Medium." Accessed: Sep. 14, 2024. [Online]. Available: <https://medium.com/@brandon93.w/regression-model-evaluation-metrics-r-squared-adjusted-r-squared-mse-rmse-and-mae-24dcc0e4cbd3>

- [138] “r2_score — scikit-learn 1.5.2 documentation.” Accessed: Sep. 14, 2024. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html
- [139] “Oracle Cloud Free Tier | Oracle.” Accessed: Sep. 08, 2024. [Online]. Available: <https://www.oracle.com/cloud/free/>
- [140] “Home.” Accessed: Sep. 04, 2024. [Online]. Available: <https://digitalis.pt/>
- [141] “Digitalis Framework - DIF.” Accessed: Sep. 04, 2024. [Online]. Available: https://ensino.digitalis.pt/index.php?option=com_content&view=article&id=170:digitalis-framework-dif&catid=101:home-artigos

Anexos

A. Resultados obtidos

Último acesso realizado a: 12/09/2024.



ResultadosObtidos.zip