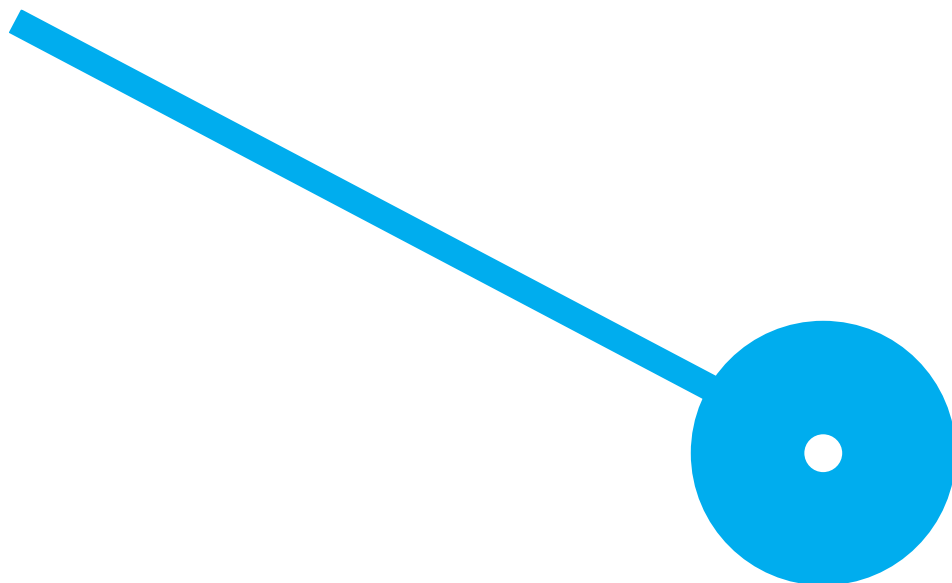
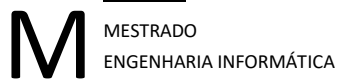


Edu-Code: uma ferramenta para avaliar o desempenho do estudante na programação

Paulo Pinto Da Cunha

12/2023





Edu-Code: uma ferramenta para avaliar o desempenho do estudante na programação

Paulo Pinto Da Cunha

8170257

Orientador

Prof. Doutor João Ricardo Martins Ramos

Dissertação apresentado para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Informática pela Escola Superior de Tecnologia e Gestão do Instituto Politécnico do Porto.

Declaração de Integridade

Eu, Paulo Pinto Da Cunha, estudante nº 8170257, do Mestrado de Engenharia Informática, da Escola Superior de Tecnologia e Gestão do Instituto Politécnico do Porto, declaro que não fiz plágio nem auto-plágio, pelo que o trabalho intitulado “Edu-Code: uma ferramenta para avaliar o desempenho do estudante na programação” é original e da minha autoria, não tendo sido usado previamente para qualquer outro fim. Mais declaro que todas as fontes usadas estão citadas, no texto e na bibliografia final, segundo as regras de referência adotadas na instituição.

Agradecimentos

A dissertação é, de todos os trabalhos académicos, um dos mais desafiantes e solitário, talvez o último, quem sabe. É um trabalho que foi construído, durante longos meses, até cá chegar e tenho a confessar que foram longos meses de desespero, ora tinha bastante informação e mal inicio a escrita sumia-se a poucas alíneas.

Queria deixar aqui um pequeno registo de agradecimentos às principais pessoas que me ajudaram neste longo percurso.

Expresso a minha profunda gratidão ao Professor Doutor João Ricardo Martins Ramos, que desde o primeiro momento me acompanhou com uma infinita disponibilidade, sugestões muitíssimo pertinentes para o meu melhor desempenho, apoio emocional, tudo isto, contribui de uma forma decisiva, para não ter desistido e hoje cá me encontro a realizar um sonho de menino, defesa da minha dissertação.

Escola Superior de Tecnologia e Gestão, que me acolheu e que me permitiu crescer na minha vida pessoal e profissional. Não poderia deixar de mencionar também todo o corpo docente e não docente pela conjugação dos conhecimentos adquiridos ao longo destes anos de estudo e investigação.

Não poderia deixar de agradecer aos meus colegas e amigos, que foram extremamente importantes ao longo de toda a minha jornada e nunca me deixaram desanimar, um sincero obrigado a todos eles, sem exceção.

Em particular, gostaria de expressar a minha gratidão a Raquel, meu braço direito na vida pessoal. Genuinamente grato por tudo, não foram meses fáceis, para ambos devido a

um susto de saúde mas mesmo assim sempre me apoiaste de uma forma incansável, grato.

Por último, gostaria de agradecer à minha família, sempre me motivaram a prosseguir os meus estudos, apoio fundamental em todas as vertentes, realçando o campo financeiro, emocional e moral. Estou lisonjeado por toda ajuda que recebi da vossa parte.

A Todos, e sem exceção, o meu sincero e profundo OBRIGADO!

Resumo

O ensino deve, geralmente, abordar várias temáticas, não se restringindo a uma área muito específica. No Ensino Superior, em particular em Engenharia Informática, são abordados diversos temas de áreas distintas com ênfase no seu relacionamento, incluindo gestão e linguagens de programação. Para que os conceitos sejam mais apelativos e de fácil compreensão, a criação de um ambiente interativo e intuitivo possui um conjunto alargado e comprovado de benefícios, uma vez que os conceitos transmitidos se adequam aos interesses dos estudantes. No entanto, novas tecnologias nem sempre são sinónimo de uma aprendizagem efetiva ou garantem que os conteúdos são adquiridos pela totalidade dos estudantes. Assim, há carência de *feedback* dos estudantes para o Professor de modo a assegurar esta missão. De modo a suprimir esta dificuldade pretende-se desenvolver uma plataforma *web*, cujo objetivo é identificar a maturidade do utilizador no desenvolvimento de código na linguagem de programação em *Java* e, simultaneamente, fornecer *feedback* sobre possíveis erros que os estudantes estejam a cometer e determinar o seu desempenho usufruindo os conceitos de *Machine Learning*. Desta forma, a ferramenta permite a captação de métricas para determinar se o utilizador encontra-se num estado inicial, médio ou experiente na programação.

Palavras-chaves: *machine learning*, avaliação remoto, *skills* de programação, análise qualitativa e quantitativa.

Conteúdo

1	Introdução	1
1.1	Motivação	1
1.2	Psicologia e Educação	2
1.3	Linguagens de Programação e Atualidade	3
1.4	Questão de Investigação	4
1.5	Objetivos	5
1.6	Metodologia de Investigação	6
1.7	Estrutura do Documento	7
2	Educação, Psicologia, Programação	9
3	Métodos de Aprendizagem	17
3.1	Aprendizagem Supervisionada	18
3.1.1	Regressão Linear e Logística	18
3.1.2	Árvore de Decisão e <i>Random Forest</i>	20
3.1.3	Redes Neurais Artificiais	24
3.1.4	<i>Support Vector Machines</i>	27
3.2	Aprendizagem Não Supervisionada	28
3.3	Aprendizagem por Reforço	30
3.4	Discussão	30
4	Edu-Code: Visão Geral	33
4.1	Validação da necessidade da ferramenta	33

4.2	Arquitetura Conceptual	40
4.3	Recolha de Dados	46
4.4	Análise de Dados	49
5	Edu-Code: Solução Proposta	53
5.1	Aplicação Desenvolvida	53
5.1.1	Perspetiva do estudante	55
5.1.2	Perspetiva do Professor	58
5.2	Aplicação de Métodos de Raciocínio	61
5.3	Discussão de Resultados	66
6	Conclusões e Trabalho Futuro	73
A	Dados de treino dos modelos de ML	79

Lista de Figuras

1.1	Representação Gráfica do Diagrama de Gantt	8
3.1	Regressão Linear e Logística	20
3.2	Árvore de Decisão Overview	21
3.3	Compra de Telemóvel - Árvore de Decisão	22
3.4	Funcionamento Prático <i>Random Forest</i>	24
3.5	Redes Neurais Conceito	26
3.6	Perceptron Output	27
3.7	Apresentação Gráfica SVM	28
3.8	Aprendizagem Não Supervisionada	29
4.1	Distribuição por Faixa Etária	34
4.2	Curso Frequentado no Ensino Secundário	35
4.3	Frequência de Cursos de CTeSP	36
4.4	Classificações de Acesso ao Ensino Superior	37
4.5	Contacto Prévio com Programação	38
4.6	Concordância da Aquisição dos Conhecimentos	38
4.7	Média das Funcionalidades Requeridas	39
4.8	Número Médio de Horas de Estudo de Programação	40
4.9	Arquitetura do Projeto com Integração SonarQube	46
4.10	Workflow do Processo de Tratamento de Dados	51
5.1	Estrutura do Projeto da Calculadora	54

5.2	Página Inicial da Interação com um Exercício	55
5.3	Demonstração do <i>alert</i> ao Guardar um Ficheiro	56
5.4	Resultado da Compilação do Exercício	56
5.5	Demonstração de um Erro de Compilação	57
5.6	Submissão do Exercício Incompleto	58
5.7	Submissão do Exercício Completo	58
5.8	Visão Geral das Métricas	59
5.9	Gráfico dos Resultados da Compilação e Submissão	59
5.10	Gráfico da Evolução da Compilação e Submissão ao Longo do Tempo . . .	60
5.11	Gráfico da Utilização do CPU e Memória na Compilação e Submissão . . .	61

Lista de Tabelas

1.1	Descrição do Planeamento de Atividades	7
4.1	Lista dos Atributos e Tipo de Dados Simplificado	47
4.2	Lista dos Atributos e Tipo de Dados Complexo	47
4.3	Distinção dos Estudantes	49
4.4	Informação Detalhada dos Datasets	49
4.5	Data Engineering - Atributos Removidos	50
4.6	Dataset1 - Estatísticas	51
4.7	Dataset2 - Estatísticas	52
4.8	Dataset3 - Estatísticas	52
4.9	Dataset4 - Estatísticas	52
5.1	Configurações Utilizados para a Regressão Logística	62
5.2	Configurações Utilizados para a Rede Neuronal	63
5.3	Configurações Utilizados para a <i>Random Forest</i>	64
5.4	Configurações Utilizadas para a <i>Support Vector Machine</i>	65
5.5	Resultados de Treino e Teste - Redes Neurais - Configuração 3 - Dataset2	70
5.6	Resultados de Treino e Teste - <i>Random Forest</i> - Configuração 2 - Dataset4	71
A.1	Resultados de Treino e Teste - Logística - Configuração 1 - Dataset1	80
A.2	Resultados de Treino e Teste - Logística - Configuração 1 - Dataset2	81
A.3	Resultados de Treino e Teste - Logística - Configuração 1 - Dataset3	82
A.4	Resultados de Treino e Teste - Logística - Configuração 1 - Dataset4	83

A.5	Resultados de Treino e Teste - Redes Neurais - Configuração 1 - Dataset1	84
A.6	Resultados de Treino e Teste - Redes Neurais - Configuração 2 - Dataset1	85
A.7	Resultados de Treino e Teste - Redes Neurais - Configuração 3 - Dataset1	86
A.8	Resultados de Treino e Teste - Redes Neurais - Configuração 4 - Dataset1	87
A.9	Resultados de Treino e Teste - Redes Neurais - Configuração 5 - Dataset1	88
A.10	Resultados de Treino e Teste - Redes Neurais - Configuração 6 - Dataset1	89
A.11	Resultados de Treino e Teste - Redes Neurais - Configuração 1 - Dataset2	90
A.12	Resultados de Treino e Teste - Redes Neurais - Configuração 2 - Dataset2	91
A.13	Resultados de Treino e Teste - Redes Neurais - Configuração 4 - Dataset2	92
A.14	Resultados de Treino e Teste - Redes Neurais - Configuração 5 - Dataset2	93
A.15	Resultados de Treino e Teste - Redes Neurais - Configuração 6 - Dataset2	94
A.16	Resultados de Treino e Teste - Redes Neurais - Configuração 1 - Dataset3	95
A.17	Resultados de Treino e Teste - Redes Neurais - Configuração 2 - Dataset3	96
A.18	Resultados de Treino e Teste - Redes Neurais - Configuração 3 - Dataset3	97
A.19	Resultados de Treino e Teste - Redes Neurais - Configuração 4 - Dataset3	98
A.20	Resultados de Treino e Teste - Redes Neurais - Configuração 5 - Dataset3	99
A.21	Resultados de Treino e Teste - Redes Neurais - Configuração 6 - Dataset3	100
A.22	Resultados de Treino e Teste - Redes Neurais - Configuração 1 - Dataset4	101
A.23	Resultados de Treino e Teste - Redes Neurais - Configuração 2 - Dataset4	102
A.24	Resultados de Treino e Teste - Redes Neurais - Configuração 3 - Dataset4	103
A.25	Resultados de Treino e Teste - Redes Neurais - Configuração 4 - Dataset4	104
A.26	Resultados de Treino e Teste - Redes Neurais - Configuração 5 - Dataset4	105
A.27	Resultados de Treino e Teste - Redes Neurais - Configuração 6 - Dataset4	106
A.28	Resultados de Treino e Teste - <i>Random Forest</i> - Configuração 1 - Dataset1	107
A.29	Resultados de Treino e Teste - <i>Random Forest</i> - Configuração 2 - Dataset1	108
A.30	Resultados de Treino e Teste - <i>Random Forest</i> - Configuração 3 - Dataset1	109
A.31	Resultados de Treino e Teste - <i>Random Forest</i> - Configuração 4 - Dataset1	110
A.32	Resultados de Treino e Teste - <i>Random Forest</i> - Configuração 5 - Dataset1	111
A.33	Resultados de Treino e Teste - <i>Random Forest</i> - Configuração 6 - Dataset1	112

A.34 Resultados de Treino e Teste - <i>Random Forest</i> - Configuração 1 - Dataset2	113
A.35 Resultados de Treino e Teste - <i>Random Forest</i> - Configuração 2 - Dataset2	114
A.36 Resultados de Treino e Teste - <i>Random Forest</i> - Configuração 3 - Dataset2	115
A.37 Resultados de Treino e Teste - <i>Random Forest</i> - Configuração 4 - Dataset2	116
A.38 Resultados de Treino e Teste - <i>Random Forest</i> - Configuração 5 - Dataset2	117
A.39 Resultados de Treino e Teste - <i>Random Forest</i> - Configuração 6 - Dataset2	118
A.40 Resultados de Treino e Teste - <i>Random Forest</i> - Configuração 1 - Dataset3	119
A.41 Resultados de Treino e Teste - <i>Random Forest</i> - Configuração 2 - Dataset3	120
A.42 Resultados de Treino e Teste - <i>Random Forest</i> - Configuração 3 - Dataset3	121
A.43 Resultados de Treino e Teste - <i>Random Forest</i> - Configuração 4 - Dataset3	122
A.44 Resultados de Treino e Teste - <i>Random Forest</i> - Configuração 5 - Dataset3	123
A.45 Resultados de Treino e Teste - <i>Random Forest</i> - Configuração 6 - Dataset3	124
A.46 Resultados de Treino e Teste - <i>Random Forest</i> - Configuração 1 - Dataset4	125
A.47 Resultados de Treino e Teste - <i>Random Forest</i> - Configuração 3 - Dataset4	126
A.48 Resultados de Treino e Teste - <i>Random Forest</i> - Configuração 4 - Dataset4	127
A.49 Resultados de Treino e Teste - <i>Random Forest</i> - Configuração 5 - Dataset4	128
A.50 Resultados de Treino e Teste - <i>Random Forest</i> - Configuração 6 - Dataset4	129
A.51 Resultados de Treino e Teste - <i>Support Vector Machines</i> - Configuração 1 - Dataset1	130
A.52 Resultados de Treino e Teste - <i>Support Vector Machines</i> - Configuração 1 - Dataset2	131
A.53 Resultados de Treino e Teste - <i>Support Vector Machines</i> - Configuração 1 - Dataset3	132
A.54 Resultados de Treino e Teste - <i>Support Vector Machines</i> - Configuração 1 - Dataset4	133

Acrónimos

CD *Continuous Delivery*. 41

CI *Continuous Integration*. 41

CNN *Convolutional Neural Network*. 26

CPU *Central Processing Unit*. 42, 61, 65

ICT *Information and Communications Technology*. 1, 7

IDE *Integrated Development Environment*. 2

ITE *Information Technology Education*. 10, 53

JSON *JavaScript Object Notation*. 47

MLP *Multi Layer Perceptron*. 26

ML *Machine Learning*. 6–8, 17–19, 27, 40, 46, 48–50, 53, 63, 66, 73, 74

NLP *Natural Language Processing*. 11

RF *Random Forest*. 22, 23, 27, 61

RNN *Recurrent Neural Network*. 27

SNS *Social Network Service*. 10

SVM *Support Vector Machine*. 27, 28, 62

AD *Árvore de Decisão*. 20–23

AMOT *Amotivação*. 3

BD Base de Dados. 44–46

EMOT Motivação Extrínseca. 3

IA Inteligência Artificial. 8, 11, 17, 18, 25

IMOT Motivação Intrínseca. 3

LP Linguagens de Programação. 2, 3, 7, 54

PLN Processamento de Linguagem Natural. 10, 11

Reg. Lin. Regressão Linear. 18, 19

Reg. Log. Regressão Logística. 19, 61

RNA Rede Neuronal Artificial. 24–27, 61, 63

TIC Tecnologias da Informação e Comunicação. 4

Capítulo 1

Introdução

No Capítulo de introdução, abordam-se diversos tópicos fundamentais. Começando com a motivação por detrás do estudo, destacando a importância de compreender a interseção entre a psicologia e a educação. Explora-se a relevância das linguagens de programação na atualidade, bem como as questões de investigação que irão orientar o trabalho desenvolvido. Os objetivos da pesquisa são delineados, juntamente com a metodologia que será utilizada. Por fim, apresenta-se uma visão geral da estrutura do documento, fornecendo um guia ao leitor sobre o que esperar.

1.1 Motivação

O ensino tem, nos últimos anos, sofrido grandes reformulações. A introdução do computador como elemento útil de auxílio à aprendizagem em comunhão com outras tecnologias de informação (*Information and Communications Technology (ICT)*) permitiu uma interação mais rápida e facilitada entre estudantes e professores, sendo também uma tecnologia útil na transmissão de material.

O recente cenário da pandemia provocou um aumento da utilização destas ferramentas e tecnologias, uma vez que parte do ensino foi realizado remotamente, ou seja, tanto estudantes como professores estavam confinados nas suas casas e o contacto era facilitado pela Internet e por ferramentas de transmissão de áudio/vídeo.

No ensino de novas Linguagens de Programação (LP) existem ferramentas como, por exemplo, o *plugin* desenvolvido pela JetBrains para o IntelliJ *Integrated Development Environment (IDE)*. Este é utilizado para aprender ou melhorar os conhecimentos em determinados conceitos de uma LP. Esta ferramenta, Edu-tools¹, permite a criação de vários cursos onde cada um contém um conjunto de exercícios dedicados a um determinado conceito da LP. Porém, esta ferramenta apresenta uma limitação, uma vez que é direcionada para quem adquire os conceitos, não se focando em quem leciona estes conceitos. Desta forma, o professor/instrutor pode criar um conjunto de exercícios (e indicar o método de resolução) e os estudantes conseguem, de uma forma autónoma, resolver e ter uma percepção das possíveis dificuldades encontradas durante a realização dos exercícios propostos. No entanto, o professor não consegue perceber se os estudantes solucionaram de forma correta todo o material disponibilizado, ou seja, esta ferramenta não fornece *feedback*.

Deste modo, e para proporcionar uma aprendizagem mais eficaz da linguagem de Programação *Java*, pretende-se desenvolver um sistema que simule um *IDE* e, desta forma, possibilitar ao estudante inserir as resoluções do(s) exercício(s) propostos pelo professor/instrutor. Por outro lado, o professor terá a possibilidade de obter *feedback* em tempo real do desenvolvimento de cada estudante. O sistema, com auxílio de uma ferramenta externa, deve integrar um modelo de aprendizagem para automatizar os processos de validação e de verificação das proficiências dos estudantes na linguagem de programação *Java*.

1.2 Psicologia e Educação

A transmissão de conhecimentos é realizada com o esforço tanto do estudante como do professor, ou seja, o professor realiza toda a descrição e resolução detalhada dos conceitos/exercícios, enquanto que o estudante pode, por tentativa e erro, resolvê-los. Neste método, é provável que não exista uma comunicação estudante/professor como ocorre num cenário de ensino presencial. Desta forma, caso o estudante não questione o professor, não

¹Disponível em <https://blog.jetbrains.com/education/2022/11/24/edutools-plugin-2022-11-is-available/>. Último acesso a 15/10/2023

é possível assegurar uma correta aprendizagem. Por outro lado, a falta de comunicação também pode afetar a motivação do estudante para a realização dos exercícios propostos pelo professor. Em [1], Vallerand et al. definiram três tipos de motivação: Motivação Intrínseca (IMOT), Motivação Extrínseca (EMOT) e Amotivação (AMOT). IMOT é o desejo de atingir a meta por si mesmo, EMOT é o interesse relacionado em atingir a meta por uma fonte externa, enquanto AMOT é o desinteresse em atingir a meta estabelecida. De modo a identificar e a provar o efeito potencial que a motivação terá no processo de aprendizagem, em [2] demonstra-se que, no caso particular do EMOT, os psicólogos motivacionais consideram que as questões focais devem ser pessoalmente significativas para os estudantes, ou seja, relacionadas às suas experiências culturais, objetivos e interesses. Utilizando esta escala de interesse, evidencia-se que deve existir um esforço tanto por parte dos professores para apresentarem conteúdos que vão de acordo aos interesses dos ouvintes quanto por parte dos estudantes para atingirem os seus objetivos pessoais de forma a evitar a amotivação.

1.3 Linguagens de Programação e Atualidade

Nos dias de hoje, a programação desempenha um papel crucial na sociedade e na economia. A capacidade de escrever código não é mais restrita a uma elite de especialistas em tecnologia, mas tornou-se uma habilidade cada vez mais valorizada e acessível a uma ampla gama de pessoas.

A procura de programadores está em constante crescimento. Nos últimos anos, tem-se assistido a um aumento exponencial no número de programadores ativos em todo o mundo. Esta tendência é impulsionada pela digitalização de muitos aspetos da vida quotidiana, desde aplicações móveis e *websites* até à automação industrial e à inteligência artificial. Cada vez mais, as empresas de diversos setores reconhecem a importância de ter uma equipa de desenvolvimento de *software*, e capacitam as mesmas para se manterem competitivas no mercado global.

Uma das razões para o crescimento da programação é a diversidade de LP disponíveis.

Cada linguagem possui as suas próprias características e é adequada para tarefas específicas. Linguagens populares como Python, *JavaScript*, *Java*, C++, e outras têm comunidades de programadores ativas e são amplamente utilizadas em diversas aplicações. Além disso, a diversificação das linguagens permite que iniciantes e profissionais experientes consigam selecionar aquela que melhor se adequa às suas necessidades e interesses ou, no sentido inverso, que possam selecionar a linguagem de programação na qual tenham maior facilidade para utilizar/aprender.

A programação também desempenha um papel crucial na economia global. Tecnologias da Informação e Comunicação (TIC) têm-se tornado de uma força motriz por detrás do crescimento económico em muitos países. *Startups* inovadoras estão a revolucionar vários setores inteiros, enquanto empresas estabelecidas dependem da tecnologia para melhorar a eficiência e criar novos produtos e serviços. A capacidade de programar não é apenas ampliar as oportunidades de carreira, mas também permite a participação ativa na construção do futuro digital.

A acessibilidade das plataformas para aprender sobre programação é outro fator crucial para a sua crescente popularidade. A Internet está repleta de recursos gratuitos, tutoriais e cursos *online* que permitem que qualquer pessoa, independentemente da sua formação, adquira habilidades de programação. Além disso, as escolas e universidades estão a incorporar, cada vez mais, a programação nos seus currículos, devido à sua relevância para o mundo moderno.

1.4 Questão de Investigação

Um das maiores preocupações de um professor ao lecionar determinados conceitos/matérias refere-se à transmissão dos conhecimentos e à aquisição destes pelos seus estudantes.

Como é demonstrado na Secção 1.2 é crucial existir um esforço contínuo por parte do professor na apresentação dos conteúdos como por parte do estudante para atingir os seus objetivos pessoais.

No entanto, as principais questões que se colocam perante a perspectiva do professor é se efetivamente é possível motivar o estudante adquirir os conceitos de programação ou se é possível ao professor perceber se o estudante se encontra com dificuldade na interpretação dos conceitos teóricos e/ou práticos.

Do lado do estudante, perante o desafio proposto pelo docente, que pode ser em formato de questões ou exercícios, a aquisição dos conceitos pode gerar dúvidas. Por consequência, os estudantes colocam as suas dúvidas quando são lecionados os conceitos ou em horas destinadas ao esclarecimento de dúvidas? Em alternativa tentam adquirir estes conhecimentos de outra forma (vídeos, livros e outros)?

Diante as duas perspectivas, poder-se-á realmente avaliar o desempenho do estudante na programação?

Desta forma, a principal questão de investigação que se coloca é:

É possível avaliar (classificar) de forma automática o nível de aprendizagem de um estudante na linguagem de programação *Java*?

1.5 Objetivos

O desenvolvimento de um projeto requer a definição de um ou mais objetivos. Estes serão o pilar para garantir que todo o desenvolvimento irá ser realizado de modo a satisfazer os objetivos inicialmente definidos. Para este projeto, podem ser identificados os seguintes objetivos principais:

- Desenvolvimento de uma plataforma *web* que permita a criação de conjuntos de exercícios pelo Professor/Docente;
- Criação de métodos para os estudantes criarem/submeterem código;
- Validação do código submetido através de compilação e execução do código no servidor;
- Criação de métricas para verificar o progresso dos estudantes, com transmissão em

tempo real de *feedback* para o Professor;

- Desenvolvimento de modelos de *Machine Learning (ML)* para atestar o nível de maturidade do estudante na resolução dos exercícios propostos.

Os objetivos identificados nos 4 primeiros pontos tiveram um desenvolvimento inicial anterior ao presente documento. Neste projeto pretende-se dar continuidade ao desenvolvimento, dando mais ênfase aos últimos dois pontos. De modo a satisfazer os objetivos definidos é necessário concretizar o seguinte sub-objetivo, ou seja, a implementação de métodos de treino de acordo com os dados guardados e disponibilização dos resultados pós-treino para validação da maturidade dos estudantes.

1.6 Metodologia de Investigação

A área do ensino encontra-se em constante desenvolvimento. Desta forma, para que um novo projeto seja iniciado é impreterível a utilização de um método de investigação ativo, do inglês, *action research*.

Neste trabalho será utilizado este método de forma a que seja atribuído uma grande ênfase a projetos existentes, verificando o que ainda é necessário aperfeiçoar ou sistemas que ainda não foram desenvolvidos e que por vezes são apontados como necessidades imediatas. Pretende-se, desta forma, solucionar um problema existente relativamente à avaliação do desempenho do estudante na programação.

Na área de ensino, é cada vez mais frequente o uso das novas tecnologias para enriquecer o ensino. O método de investigação ativo surge quando existe a necessidade de resolver um problema concreto, quer este se reflita na implementação de novos sistemas, quer represente a continuação de projetos de outrém.

Para o desenvolvimento deste projeto, foi efetuada uma planificação dividida em diferentes etapas (tarefas) que se expõe de seguida:

1. Acrescentar novas métricas do *client-side* como *Mouse Dynamics*;

2. Escolha das ferramentas adequadas;
3. Integrar a ferramenta SonarQube com a ferramenta em desenvolvimento para adicionar novas métricas;
4. Escolha e Aplicação dos modelos de *ML*;
5. Avaliação do desempenho dos modelos utilizados;
6. Redação da dissertação.

Na Tabela 1.1 identificam-se as tarefas anteriormente mencionadas com os seus respetivos números, com uma breve descrição e atribuindo-lhe uma duração em meses. De forma a facilitar a visualização da distribuição das diversas etapas apresenta-se na Figura 1.1 um diagrama de Gantt.

Tabela 1.1: Descrição do Planeamento de Atividades

Nº da Tarefa	Descrição	Duração (meses)
1	Acrescentar novas métricas do <i>client-side Mouse Dynamics</i>	2
2	Escolha das ferramentas adequadas	2
3	Integrar a ferramenta SonarQube com o Edu-Code	3
4	Escolha e Aplicação dos modelos de <i>ML</i>	4
5	Avaliação do desempenho dos modelos utilizados	1
6	Redação da dissertação	8

1.7 Estrutura do Documento

A presente dissertação retrata a área do ensino e de uso de tecnologias em ambiente *web*, estando dividida em 6 Capítulos. No primeiro Capítulo é efetuada uma introdução do tema, definindo o conceito das *ICT* com a sua importância nos tempos de hoje na área de ensino. Elabora-se um breve estudo da psicologia, educação e LP diante da atualidade, finalizando com os objetivos do trabalho desenvolvido e inclusão ao método de investigação utilizado.

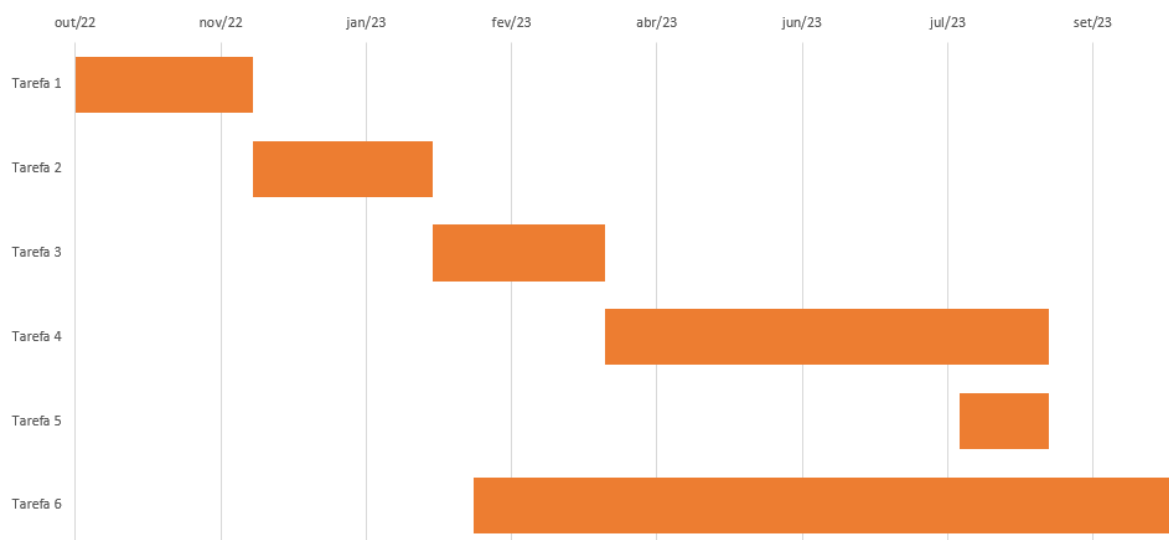


Figura 1.1: Representação Gráfica do Diagrama de Gantt

No Capítulo 2 são mencionados os conceitos teóricos mais aprofundados da educação, psicologia e programação.

De seguida, no Capítulo 3, é fundamentado o conceito da Inteligência Artificial (IA) explorando um dos sub-componentes desta área que são os algoritmos de *ML* aplicados na dissertação, apresentando o seu funcionamento e que tipos de projetos foram utilizados no passado.

Como é um projeto desenvolvido para ambiente *web*, no Capítulo 4 apresenta-se uma visão geral da ferramenta desenvolvida, detalhando o seu propósito, a sua relevância como a arquitetura conceptual e a recolha de dados efetuado para utilizar nos algoritmos de *ML*.

No Capítulo 5, é apresentada a solução proposta para a perspetiva do estudante a resolver o exercício previamente proposto pelo professor e a perspetiva do Professor na obtenção de *feedback* da resolução em *realtime* exercido pelo estudante com a discussão dos resultados obtidos.

Por ultimo, no Capítulo 6 são divulgadas as conclusões e possíveis melhorias a serem implementadas no futuro.

Capítulo 2

Educação, Psicologia, Programação

A educação é uma forma de transmitir conhecimento, facilitar a aprendizagem e inspirar a inovação, sendo essencial para o desenvolvimento individual e social [3]. Há muito tempo, foi considerada uma importante força para o desenvolvimento económico e social, uma forma eficaz de reduzir a pobreza e promover a prosperidade, e foi referido como uma das prioridades em várias agendas de desenvolvimento global, como refletem os objetivos de desenvolvimento sustentável das Nações Unidas da Agenda 2030 para o desenvolvimento Sustentável [4].

No início dos anos 90 surgiu um novo conceito de aprendizagem nomeada pelo processo de Bolonha [5] que foi aplicado em toda a Europa. Trata-se de introduzir um sistema de ensino superior que é dividido em três ciclos, o mesmo consiste em estudos de licenciatura, mestrado e doutoramento. Este novo conceito pretende também assegurar o reconhecimento mútuo das qualificações e dos períodos de aprendizagem no estrangeiro, concluídos noutras universidades e aplicar um sistema de garantia da qualidade, a fim de reforçar a qualidade e a relevância da aprendizagem e do ensino. Ao aplicar este processo, como exemplo, foi originada uma nova formação e apoio à educação nomeada por Erasmus+. Num caso de estudo realizado por Asoodar et al. [6] foi realizada uma monitorização aos estudantes antes, durante e no fim desta formação utilizando como metodologia uma abordagem qualitativa e quantitativa para verificar, através de um questionário, se esta

nova experiência foi um sucesso. Como resultado os autores concluíram que se verificou uma melhoria das habilidades de comunicação do idioma local, motivação através de outras culturas e aprendizagem pessoal do estudante (autoaprendizagem).

Mais recentemente, novas abordagens de ensino foram criadas com recurso a *Information Technology Education (ITE)* com o propósito de facilitar a aprendizagem e compreensão de novos conceitos. Por exemplo em Shizuoka no Japão foram desenvolvidos materiais de ensino para ajudar os estudantes do primeiro ao terceiro ano a ter noção do que devem publicar no *Social Network Service (SNS)*. Este material contém um sistema baseado em imagens e compara-a a uma temática específica [7], no exemplo do Twitter [8] conclui-se que é considerado como uma ferramenta útil para obter informações sobre catástrofes locais com base na sua atualidade, realidade e especificidade. Apesar da sua utilidade, este não é o primeiro projeto a utilizar esta abordagem de ensino. Sakai, Shiota e Eguchi [9] utilizaram a mesma abordagem com diferença de acrescentar um método de classificação que permite que as crianças entendam melhor os problemas de comunicação do *SNS*.

Considerando a mesma temática (aprendizagem), mas com foco no ensino de linguagens de programação, o plugin Edu-Tools do IntelliJ IDE da JetBrains [10] permite a criação de um curso completo ou de simples exercícios de programação interativa em que o estudante pode obter dicas (ajudas) que o auxiliam na resolução dos exercícios. O professor pode também incluir algumas validações automáticas de código com base em testes unitários e fornecer um *feedback* padrão de acordo com o tipo de erros. A resolução dos exercícios de programação é baseada no resultado de, por exemplo, funções *Java*, permitindo ao utilizador desenvolver livremente o algoritmo para atingir o objetivo.

Para uma educação com sucesso não é apenas necessário realizar a apresentação de conteúdos, mas considerar de igual modo como é que a área de psicologia afeta ao ensino. Esta presença em cursos de Engenharia como, por exemplo Engenharia Informática, pode trazer vários benefícios. No entanto é necessário considerar cuidadosamente a integração desta disciplina para rentabilizar a aprendizagem tornando-a o mais produtivo possível.

Um exemplo presente na atualidade e no quotidiano das pessoas é o Processamento de Lin-

guagem Natural (PLN), do inglês *Natural Language Processing (NLP)*. Este é um campo de estudo multidisciplinar que envolve processos de IA, Informática, Estatística e a área da linguística e que permite, através de diversos processos computacionais, manipular e interpretar a linguagem humana. Os assistentes virtuais como a Microsoft Cortana, Apple Siri, Amazon Alexa e Google Home são exemplos ativos da utilização deste tipo de tecnologia que permite ao utilizador interagir com a máquina através da comunicação verbal [11]. Além da comunicação verbal, o PLN permite também que a máquina (computador) tenha capacidade de interpretar textos escritos.

O desenvolvimento de uma aplicação informática implica, por um lado, a capacidade adquirida de programação e, por outro, a interação com outros intervenientes como, por exemplo, elementos da equipa de desenvolvimento, clientes finais, e outros elementos com participação ativa como, por exemplo, gestores de projeto. Assim, torna-se imprescindível que além de se reconhecer a transmissão prévia de conhecimentos de programação se deva considerar a área da psicologia como uma área de desenvolvimento ativa durante o processo de aprendizagem.

A psicologia verifica-se em diferentes áreas como:

- **Compreensão das Interações Humanas:** o desenvolvimento de um sistema informático é realizado para suprimir uma necessidade de um cliente (pessoa). A correta compreensão e interação entre o desenvolvedor e o utilizador final é fundamental para o sucesso de um projeto, uma vez que poderá permitir uma melhor interpretação das necessidades e expectativas do utilizador final;
- **Integração em Equipas Interdisciplinares:** a realização de um projeto em Engenharia Informática requer, por vezes, a interação entre elementos com competências e conhecimentos distintos. Numa equipa multidisciplinar é crucial garantir que a mensagem seja transmitida corretamente e que a linguagem seja adaptada ao recetor da informação, garantindo que a aplicação da psicologia neste tipo de comunicação revela-se útil, uma vez que permite uma maior e melhor colaboração entre os vários intervenientes;

A ligação entre a educação, psicologia e a programação consegue proporcionar uma perceção única sobre como os casos se processam de aprendizagem e desenvolvimento humano. Como referido no subtítulo neste decorrer de texto são abordado três temas que se interligam e completam, refletindo uma combinação de campos bastante harmoniosa para a contribuição da evolução da educação moderna.

A Educação Holística pretende restabelecer estas conexões em todas as esferas da vida e em todos os tipos de relações, tais como refere Yus [12]:

- Relações entre o pensamento linear e intuição;
- Relações entre a mente e o corpo;
- Relações entre o domínio de conhecimento (Abordagens interdisciplinares);
- Relações entre “eu” e a comunidade (Desenvolvimento de habilidades interpessoais, de serviço comunitário e de ação social);
- Relações entre o “eu” e o “Eu” (parte mais profunda de cada pessoa, tendo como veículos de conexão a música, a dança, a poesia, a pintura, o drama, etc).

Quando se entra no ramo da educação e em todos os seus fundamentos, pode alegar-se que é um dos pilares fundamentais para o desenvolvimento humano. A psicologia, que estuda o comportamento humano e desempenha o papel fundamental para que as pessoas consigam ter um maior aproveitamento na aprendizagem, alia-se e bem à educação. Os fundamentos da educação passam um pouco pelas bases dadas ao aprendiz.

“A principal meta da educação é criar homens que sejam capazes de fazer coisas novas, não simplesmente repetir o que outras gerações já fizeram. Homens que sejam criadores, inventores, descobridores. A segunda meta da educação é formar mentes que estejam em condições de criticar, verificar e não aceitar tudo que a elas se propõe.”, como referido por Jean Piaget.

Com base nesta citação pode alegar-se que a educação enfatiza a necessidade de novas abordagens até chegar ao ponto da individualidade. Sim individualidade, transpondo para a vertente teórica/prática, é uma nova forma de ensinar os estudantes com dados

de aprendizagem cognitivos diferentes com ritmos de processamento, baseados em dados fornecidos pelos psicólogos. Os psicólogos em equipe com os educadores adaptam o melhor método de ensino, criando desta mesma forma necessidades únicas para cada estudante. Em suma desta mesma forma cada estudante consegue adquirir o máximo de conhecimento possível com um método adaptado às suas dificuldades/necessidades e deste mesmo modo consegue otimizar-se a aprendizagem do estudante e o seu desenvolvimento.

A programação é uma ferramenta educacional bastante poderosa. A implantação educacional com a introdução nas escolas tem vindo a ser um grande pilar no desenvolvimento de várias habilidades cognitivas, tais como: forma de pensar; agir; como agilizar/resolver um problema, entre outros. Tudo isto ajuda a contribuir para o sucesso dos estudantes de hoje em dia a serem melhores profissionais amanhã. Desta forma foi-lhes ensinado a “trabalharam” o cérebro em várias situações e, assim, conseguiram adquirir de uma forma abrangente mais sabedoria; cultura em acréscimo ao tema educacional fundamentado no modulo base da escola.

Martha Nussbaum (2010) denuncia que os países com sucesso económico renunciam as competências essenciais para a sobrevivência das democracias. Se esta tendência persistir, em breve, produzir-se-ão, pelo mundo inteiro, gerações de máquinas úteis, dóceis e tecnicamente qualificadas, em vez de cidadãos, capazes de pensarem por si próprios, de porem em causa a tradição, e de compreenderem o sentido do sofrimento e das realizações dos outros [13].

A educação holística não é atribuído a uma pessoa/figura em particular que a tenha feito “surgir”. Por sua vez, foi o resultado de uma evolução gradual das ideias e da sua filosofia educacional ao longo deste tempo. Várias pessoas e pensadores contribuíram para o desenvolvimento e a promoção de educação holística ao longo da história. Algumas das influências e figuras notáveis associadas a essa mesma abordagem são:

- Rudolf Steiner (1861-1925): Fundador da pedagogia woldorf, Rudolf Steiner teve um papel significativo na promoção da educação holística. Desenvolveu um sistema educacional que enfatiza o desenvolvimento equilibrado do ser humano, incluindo

aspectos emocionais, sociais e espirituais.

- Maria Montessori (1870-1952): Médica Italiana, desenvolveu um método Montessori, que também abraça uma abordagem holística da educação, dando ênfase à autonomia, à auto-educação e ao desenvolvimento integral da criança.
- Fritjof Capra (nascido em 1939): Físico teórico e escritor conhecido por explorar a interconexão entre a ciência, a espiritualidade e a ecologia.
- Ken Wilber (nascido em 1949): Ken Wilber é um filósofo e escritor que desenvolveu uma Teoria Integral que procura integrar diferentes dimensões da experiência humana.

Rudolf Steiner acreditava na importância de conciliar a razão e a espiritualidade do indivíduo. O método Waldorf baseia-se num processo de ensino-aprendizagem lúdico e inclusivo, onde os estudantes têm a oportunidade de aprender através de experiências individuais ou em grupos. Além disso, o método tem um foco claro nas manifestações artísticas. Ao contrário dos métodos tradicionais de ensino, o método Waldorf não se concentra na memorização de conceitos nem em testes constantes. Em vez disso, explora diferentes formas de aprendizagem, como a arte ou a música, por exemplo, que muitas vezes são ignoradas pelos métodos convencionais [14]. Portanto, neste método de ensino, a educação é interdisciplinar e estimula a imaginação, trazendo resultados positivos para o desenvolvimento cognitivo das crianças.

O método de Montessori é um método de educação que se baseia na atividade auto dirigida, aprendizagem prática e encadeia a rede colaborativa. Nas salas de aula que seguem os métodos Montessori, as crianças têm a oportunidade de fazerem escolhas criativas visando a aprendizagem. Criado pela médica Maria Montessori, este mesmo método tem como metodologia a ideia de que criança tem um potencial único e deve ser incentivada a desenvolvê-lo de forma individual [15] através de materiais pedagógicos específicos e estratégias pedagógicas. Esta abordagem educacional tem conquistado cada vez mais adeptos nos vários pontos do mundo inteiro, embora seja criticado por outros filósofos de outras áreas parecidas.

Fritjof Capra é um renomado físico teórico e escritor que teve um pico de destaque no parâmetro da exploração dos conceitos ligados à Educação Holística. Este veio trazer um olhar interdisciplinar e bem mais aprofundado sobre a causa de quando ocorreu o surgimento das transformações/evoluções a níveis da educação, no qual se espelhou através dos desenvolvimentos da sociedade no campo da contemporânea. Uma das suas maiores obras é “O ponto de mutação”. Através deste livro consegue interligar grandes temas do mundo, tais como: educação; espiritualidade e ciência [16]. É uma obra que aborda a educação holística através do meio de várias situações e parâmetros, tais como: Interconexão entre disciplinas; Aprendizagem contextual; Abordagem ecológica; Pensamento sistêmico; Espiritualidade e consciência e mudança de paradigma.

O método de Ken Wilber, conhecido como “Teoria Integral”, destaca a importância de compreender e abordar o desenvolvimento completo do indivíduo, abrangendo aspectos físicos, emocionais, mentais e espirituais [17]. Esta abordagem visa proporcionar uma visão mais abrangente do ser humano e do mundo, reconhecendo que todos estes aspectos estão interligados e desempenham papéis fundamentais na formação do indivíduo. A teoria integral procura integrar diferentes perspectivas, disciplinas e dimensões da experiência humana para criar uma compreensão mais completa e integrada da realidade.

Em suma, verifica-se que ensinar é “uma arte” que envolve várias etapas e que deve ter vários parâmetros em consideração caso se pretenda maximizar o seu sucesso.

Capítulo 3

Métodos de Aprendizagem

Nos últimos anos, a IA tem estado mais presente para ajudar a otimizar os processos com o intuito de responder às necessidades da sociedade de hoje em dia e nas suas diferentes áreas. O grande volume de informação que os sistemas digitais têm gerado permitem que as respostas geradas por estes métodos sejam de qualidade, auxiliando o utilizador na sua tomada de decisão.

Um dos primeiros programas a ser desenvolvido nesta área, foi realizado na década de 1950 por Allen Newell e Herbert A. Simon [18]. Esta ferramenta foi criada com o intuito de resolver teoremas matemáticos. Newell e Simon submeteram ao seu sistema 52 teoremas apresentados no livro “Principia Mathematica” de Russel e Whitehead. Esta obra, constituída por três volumes, é conhecida pelos seus fundamentos de matemática. O programa teve capacidade para resolver os primeiros 38 problemas tendo, em alguns casos, encontrado provas mais elegantes às propostas pelos autores.

Poucos anos depois, em 1956, Arthur Samuel [19] desenvolveu um programa designado “Checkers”, na IBM 704, capaz de determinar as melhores jogadas num jogo de xadrez tendo em consideração a disposição das peças no tabuleiro e os movimentos do adversário. Este *software* foi desenvolvido recorrendo a métodos e técnicas que atualmente integram a IA, mais concretamente a área de Aprendizagem Máquina, do inglês *ML*. Com base no *software* desenvolvido e no aperfeiçoamento das diversas técnicas de *ML* utilizadas, a

11 Maio de 1997 o campeão de xadrez Garry Kasparov perdeu o seu sexto jogo contra a máquina *Deep Blue* [20] em 19 jogadas. Considerado um momento de revolução da IA, este tipo aprendizagem automática é atualmente amplamente utilizado por jogadores que xadrez que pretendem melhorar e progredir neste jogo.

A IA tem tido uma evolução constante ao longo dos últimos anos, existindo uma ampla variedade de modelos. Estes são maioritariamente subdivididos em modelos de aprendizagem supervisionada (Secção 3.1), não supervisionada (Secção 3.2) e por reforço (Secção 3.3). Os Capítulos subsequentes apresentam, de forma não exaustiva, um conjunto de técnicas utilizadas na área de *ML*. A seleção destes métodos tem também em consideração a sua possível utilização no sistema desenvolvido (apresentado na Secção 4). No final da presente secção é realizada uma discussão sobre os métodos de raciocínio apresentados.

3.1 Aprendizagem Supervisionada

A aprendizagem supervisionada, do inglês *Supervised Learning*, implica a pré-existência de um conjunto de dados (*dataset*) no qual se conhece o resultado final para um conjunto de dados. Desta forma, os algoritmos que se enquadram neste tipo de aprendizagem têm como objetivo identificar a melhor ponderação para cada uma das variáveis independente de modo a que o resultado estimado (variável dependente) seja o mais próximo possível do real, minimizando o erro cometido [21].

3.1.1 Regressão Linear e Logística

A regressão que se considera mais simples e que é amplamente utilizada é a Regressão Linear (Reg. Lin.). A sua representação mais básica equipara-se a uma reta onde existe uma variável independente (x) e uma variável dependente y . O objetivo desta regressão é determinar a melhor ponderação para x e o valor da constante de modo a obter o menor erro possível na previsão de y . Como resultado de execução, obtém-se uma reta do tipo $y = mx + b$. Numa representação mais genérica, considera-se a existência de várias variáveis independentes e o objetivo é determinar o nível de ponderação de cada variável,

ou seja, se considerar a função genérica $y = \beta_0 + \beta_1x_1 + \dots + \beta_nx_n$ onde n representa o número de atributos o objetivo é determinar os melhores valores para β_i . Este tipo de regressão tem como resultado um valor contínuo, sendo utilizado para previsão de precipitação e temperatura diárias [22] ou previsão exata do preço do BitCoin [23].

A Regressão Logística (Reg. Log.) é um algoritmo de *ML* supervisionado utilizado para problemas de classificação. Essa nomenclatura está relacionada com o fato da Reg. Log. ser construída a partir da aplicação de uma transformação/função (denominada função logística ou sigmóide) sobre a Reg. Lin.. Esta transformação converte o resultado contínuo que varia no espaço de \mathcal{R} num intervalo entre 0 e 1. Considerando-se um valor de *threshold* de, por exemplo, 0.5 então ao resultado obtido realiza-se o arredondamento para 0 ou 1. Com um resultado binário obtém-se um mecanismo de classificação onde 1 indica a que grupo pertence (caso positivo) e 0 caso contrário. Este tipo de sistemas de classificação é utilizado na detecção de doenças crónicas [24]. O resultado binomial (0 ou 1) pode também ser estendido para 3 ou mais categorias (previsão multicategórica), sendo aplicado na detecção do tipo de cancro [25] ou previsão de diabetes mellitus [26].

Na figura 3.1 é apresentado uma visão geral da regressão, indicando o tipo de regressão, objetivo final e tipo de resposta obtido como variável dependente.

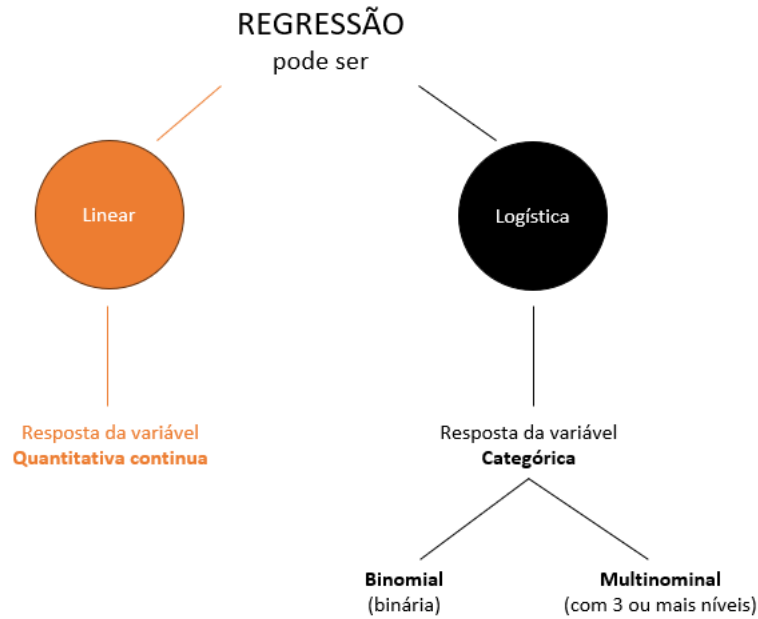


Figura 3.1: Regressão Linear e Logística

3.1.2 Árvore de Decisão e *Random Forest*

Uma Árvore de Decisão (AD) é constituída por três componentes: nós de decisão, nós de folha e um nó de raiz. Um algoritmo de AD divide o conjunto de dados em ramos, que por sua vez se sub-dividem noutros ramos. Esta sequência continua até ser atingido um nó folha. O nó folha não pode ser mais segregado.

Os nós da árvore de decisão representam atributos que são utilizados para prever o resultado e geram a separação dos dados em função dessa decisão. Os nós de decisão fornecem uma ligação para as folhas. De uma forma visual apresenta-se o diagrama da Figura 3.2 que mostra os três tipos de nós de uma AD.

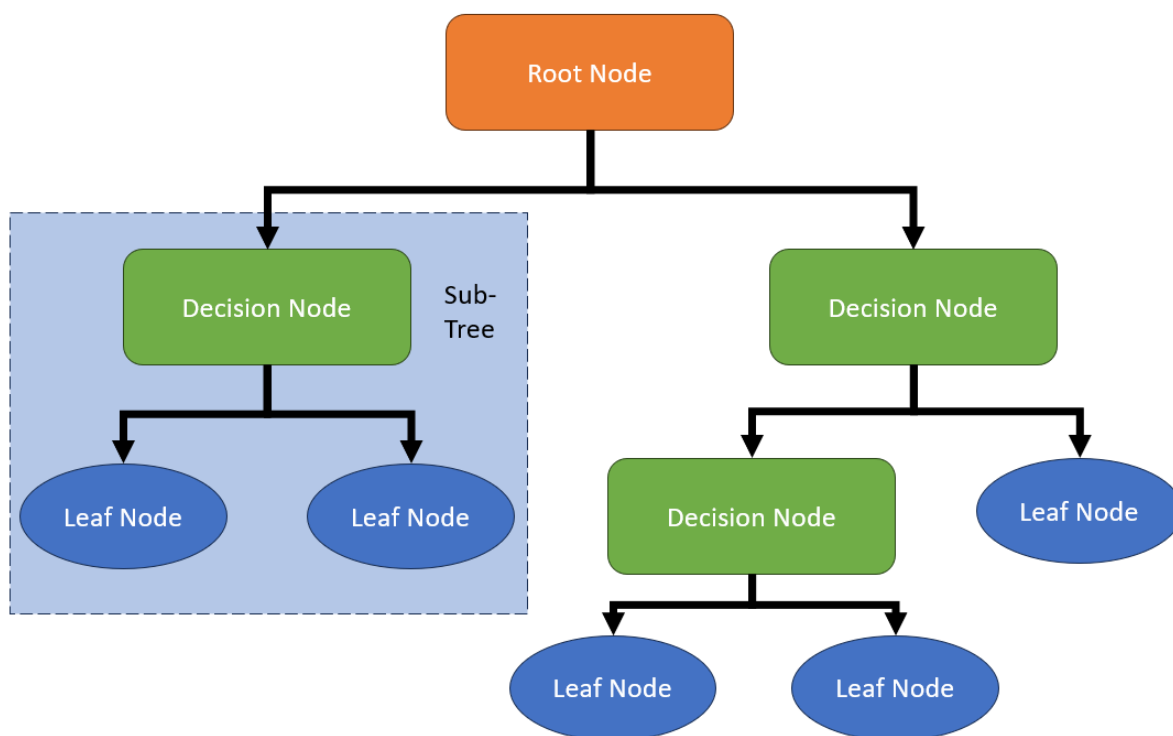


Figura 3.2: Árvore de Decisão Overview

Para demonstrar o funcionamento de uma AD, tome-se como exemplo o caso de identificar se uma pessoa poderá adquirir ou não um novo telemóvel. As características do telemóvel constituem a base da sua decisão. O nó raiz e os nós de decisão representam as características do telemóvel (por exemplo, preço final do dispositivo, capacidade de memória, entre outros). O nó folha representa o resultado final, comprar ou não comprar. As principais características que determinam a escolha incluem o preço, o armazenamento interno e a memória de acesso aleatório (RAM). A AD é representada em formato de árvore e, para o exemplo atual, pode definir-se como representado na Figura 3.3.

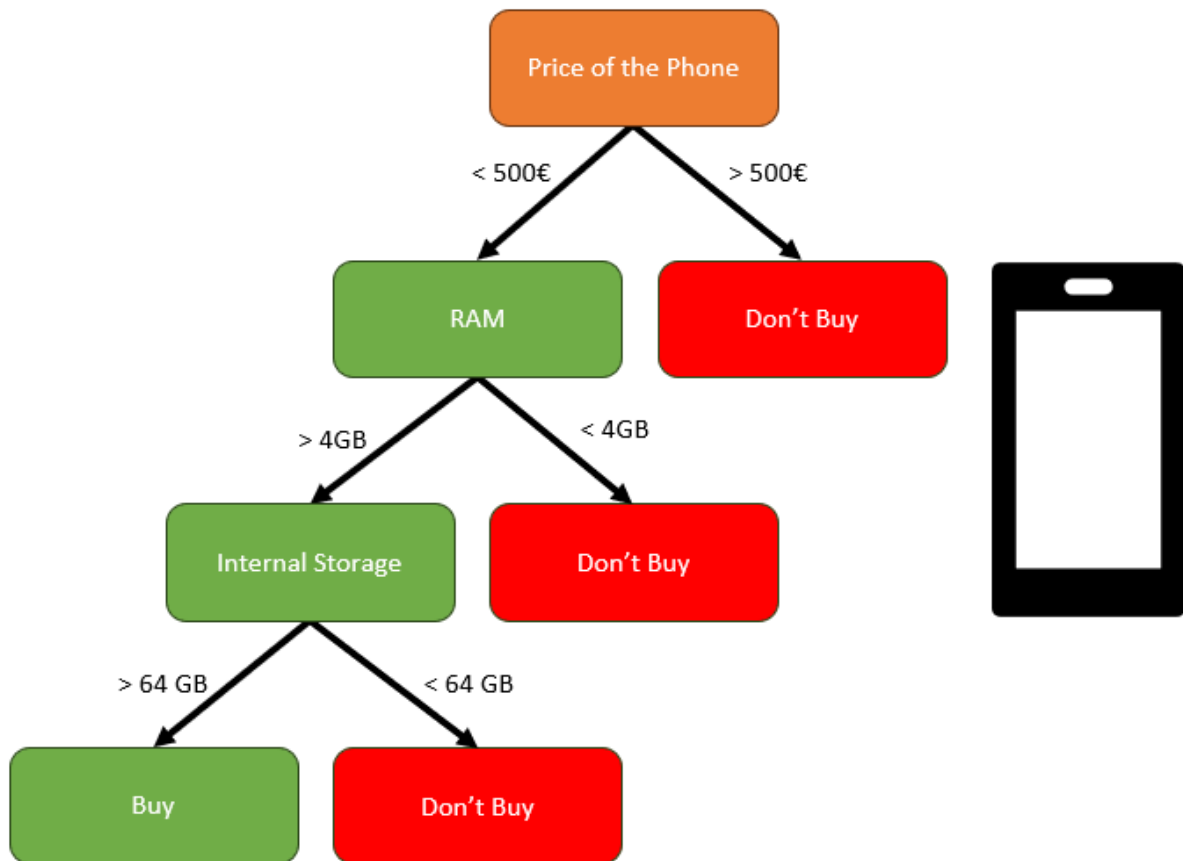


Figura 3.3: Compra de Telemóvel - Árvore de Decisão

Neste caso cada nó de decisão define uma resposta binária e a sub-árvore que se cria debaixo desse nó assume as restrições definidas nos nós de decisão que lhe precedem.

A *Random Forest (RF)* é outro algoritmo de aprendizagem supervisionada que tem por base a estrutura da AD. No entanto, a *RF* utiliza uma aprendizagem em conjunto, que é uma técnica que combina muitos classificadores para fornecer soluções para problemas complexos. Desta forma, é gerado um conjunto de árvores de classificação. Cada uma dessas árvores possui a sua estrutura interna própria, mas garante o mesmo tipo de *output*. Para se obter o resultado final da *RF* é efetuada uma média dos resultados obtidos em cada AD. O aumento do número de árvores (e o respetivo número de níveis de decisão) aumenta normalmente a precisão do resultado, no entanto, requer um processamento maior a nível da máquina para poder treinar e executar o modelo.

O exemplo anterior, cujo objetivo é prever se um cliente iria ou não adquirir um telemóvel, pode ser utilizado para demonstrar o funcionamento da *RF*. Em vez de ter uma única AD,

a *RF* tem um conjunto de AD. A título de exemplo, considere-se que se utilizam quatro AD. Nesse caso, os dados de treino, que incluem as observações e os recursos do telefone, serão divididos em n nós raiz. Os nós de raiz podem representar n características que podem influenciar a escolha do cliente, nomeadamente preço, armazenamento interno e RAM. A *RF* divide os nós para escolher as características de forma aleatória. A previsão final será selecionada com base no resultado das n árvores, ou seja, o resultado indicado pela maioria das árvores será utilizado como resultado final da *RF*.

O método de cálculo do resultado final de uma *RF* é dependente do tipo de resultado que se pretender obter. No exemplo anterior, como o resultado final é do tipo categórico (comprar ou não comprar), recorre-se à moda dos vários valores obtidos para se identificar o valor final. No caso de se pretender obter um resultado numérico, é realizada a média dos resultados obtidos em cada AD.

De modo a permitir uma visualização gráfica de uma *RF* tome-se o exemplo da Figura 3.4. Neste caso o conjunto de dados é constituído por vários frutos, como bananas, maçãs, ananases e mangas. A *RF* divide este conjunto de dados em subconjuntos com parte dos dados de cada um. Cada um dos subconjuntos é utilizado para a criação de uma AD no sistema de *RF*. Cada AD produz o seu resultado específico. No exemplo da figura verifica-se que as árvores 1 e 2 previram maçã enquanto que a árvore 3 previu banana. O classificador da *RF* recolhe a votação da maioria para fornecer a previsão final. Dado que a maioria das AD previram o resultado como maçã, o classificador seleciona esta opção como resultado final da *RF*.

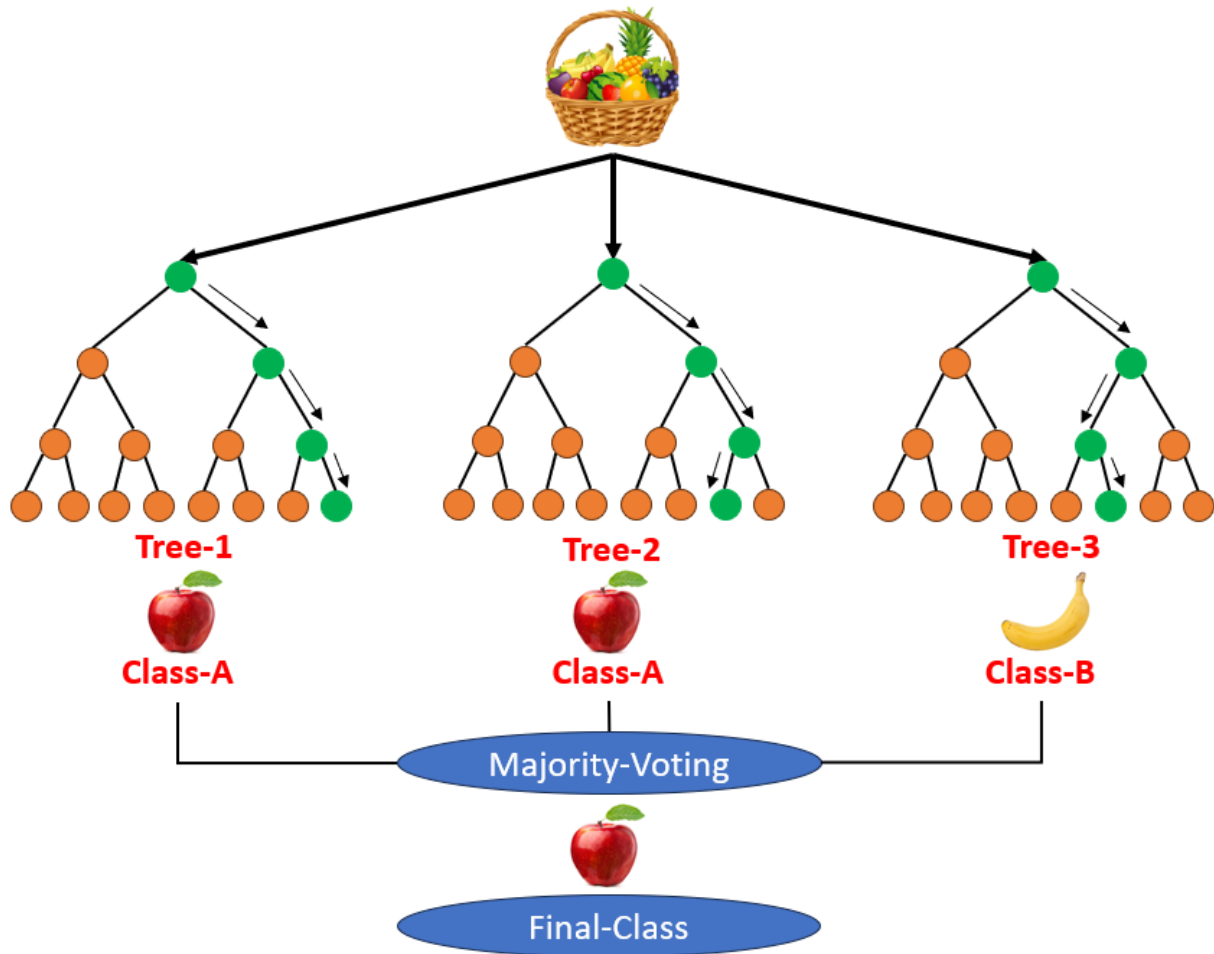


Figura 3.4: Funcionamento Pratico *Random Forest*

3.1.3 Redes Neurais Artificiais

A Rede Neuronal Artificial (RNA) é um algoritmo de aprendizagem que se assemelha ao cérebro humano. Neste a informação é enviada através de vários axónios (sinapses) onde cada neurónio pode, em função de sinapses anteriores, ser ativado e retransmitir a informação para o neurónio seguinte. Uma RNA é, de forma similar, constituída por um conjunto de nós que, caso sejam ativos, transmitem informação para os nós da camada seguinte. Uma RNA é, desta forma, constituída por várias camadas de nós, existindo três tipos de camadas distintos: camada de entrada (*input layer*), camada de saída (*output layer*) e uma ou mais camadas ocultas (*hidden layers*). A camada de entrada representa os dados iniciais (equivalente ao estímulo inicial que ocorre no cérebro humano) e a camada de saída o resultado que se pretende prever (Figura 3.5). Cada nó de uma camada, com exceção da última camada, conecta-se a todos os neurónios da camada seguinte. Cada

ligação entre nós possui um determinado peso e a conjugação dos valores recebidos com ponderação dos respectivos pesos pode ativar o nó de modo a que este transmita informação para os nós subsequentes. A ativação de um nó é representada por uma função (função de ativação) e, de acordo com o tipo de função selecionado, poderá ou não existir um *output* diferente de 0.

As RNA contam com dados de treino para aprender e melhorar sua precisão ao longo do tempo, ou seja, ajustar os valores presentes nos pesos das ligações entre nós de modo a diminuir o erro entre o valor estimado e o valor real. Dado que esses algoritmos de aprendizagem são ajustados para aumentar a precisão, estes tornam-se ferramentas poderosas na área de ciência da computação e de IA, permitindo classificar e agrupar dados a uma alta velocidade. As tarefas de reconhecimento de fala ou reconhecimento de imagem podem levar minutos em vez horas, quando comparadas com a identificação manual feita por especialistas humanos. Uma das RNA's mais conhecidas é o algoritmo de procura do Google [27] que aumenta a Compreensão da linguagem humana pelo motor de busca *Google BERT*.

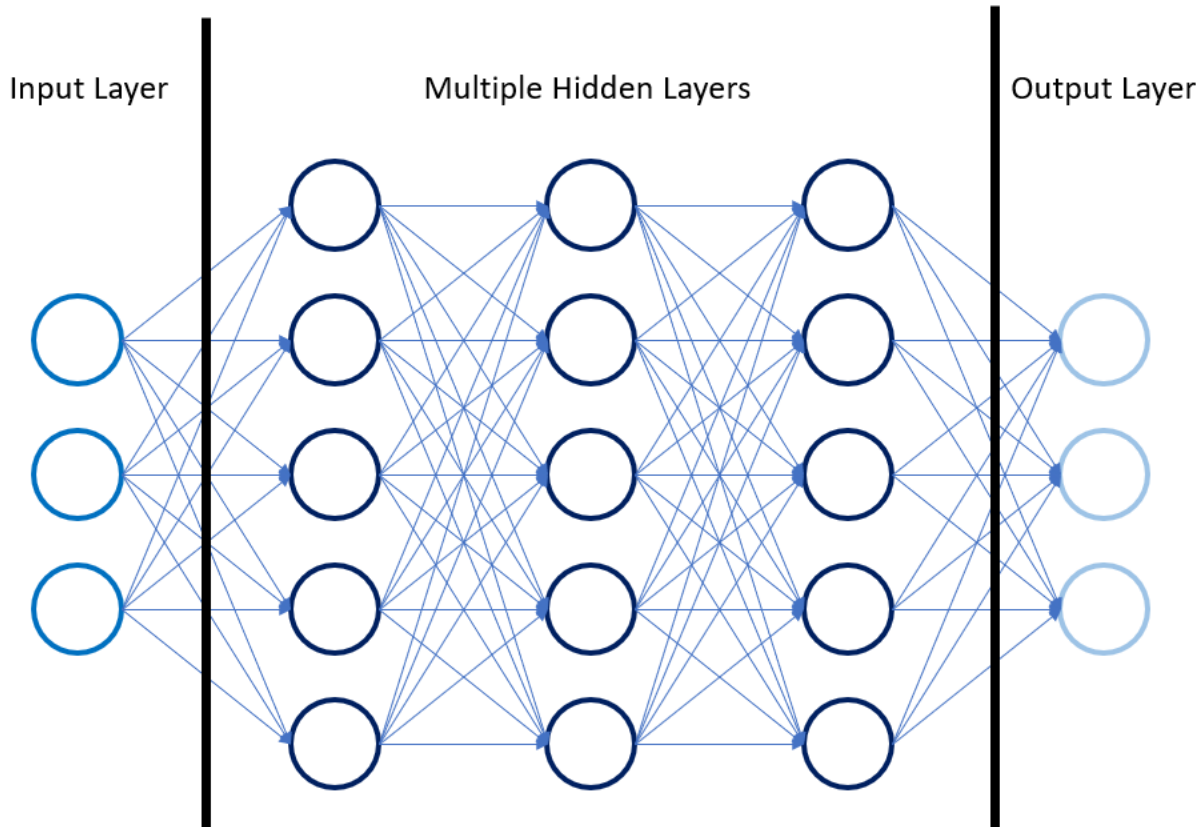


Figura 3.5: Redes Neurais Conceito

As RNA's podem ser classificadas em diferentes tipos, as quais são usadas para diferentes propósitos. Embora esta não seja uma lista abrangente apresenta-se, de seguida, os tipos mais comuns de RNA:

- O *perceptron*, representado na Figura 3.6, é a rede neuronal mais antiga, criada por Frank Rosenblatt em 1958. Um *perceptron* é um modelo matemático que recebe varias entradas x_1, x_2, \dots, x_n e produz uma única saída num formato binário.
- A RNA *feedforward*, ou *Multi Layer Perceptron (MLP)* é composta por uma camada de *input*, uma ou mais camadas ocultas e uma camada de *output*. Este tipo de RNA é, na verdade, composta por neurónios sigmóides e não *perceptrons*, já que a maioria dos problemas do mundo real é não linear. Este tipo de modelo é frequentemente utilizado para visão por computador, processamento de linguagem natural, entre outros.
- A RNA de convolução, *Convolutional Neural Network (CNN)*, é semelhante à rede

feedforward, mas são habitualmente usadas para reconhecimento de imagem, reconhecimento de padrões e/ou de visão por computador. Este tipo de rede aproveita os princípios de álgebra linear, particularmente a multiplicação de matrizes, para identificar padrões dentro de uma imagem.

- As RNA recorrentes, *Recurrent Neural Network (RNN)*, são constituídas por ligações de *feedback*, ou seja, as ligações entre camadas não seguem um padrão linear podendo existir ligações para camadas anteriores. Este algoritmo de aprendizagem é tipicamente selecionado para tratamento de séries temporais para fazer previsões sobre resultados futuros, como previsões de mercado de ações ou previsão de vendas.



Figura 3.6: Perceptron Output

O conceito de uma *RF* é bastante complexo e requer alguma capacidade computacional da máquina para poder executar este algoritmo. Cada vez mais, existe a necessidade de aplicar algoritmos mais complexos de *ML* para algumas previsões, uma delas, tem o projeto de prever a detecção de deslizamentos de terra [28] ou a previsão de stock do Mercado [29].

3.1.4 *Support Vector Machines*

Uma *Support Vector Machine (SVM)* é um algoritmo de aprendizagem utilizado tanto para classificação como para regressão. Essencialmente, o *SVM* é responsável por encontrar a melhor fronteira de separação entre classes/rótulos para um dado conjunto de dados que sejam linearmente separáveis. Para a *SVM*, as diversas fronteiras de separação possíveis que são capazes de separar completamente as classes são chamadas de hiperplanos. Dessa forma, a *SVM* encontra o melhor hiperplano para um dado *dataset* cujas classes

são linearmente separáveis.

Considere-se o exemplo apresentado na Figura 3.7. No exemplo existem duas classes representadas pelos diferentes sinais (o hífen a preto e a cruz a vermelho). O objetivo de se executar a *SVM* é determinar o melhor hiperplano que separe as classes. Este hiperplano corresponde aquele que maximize a distância entre classes.

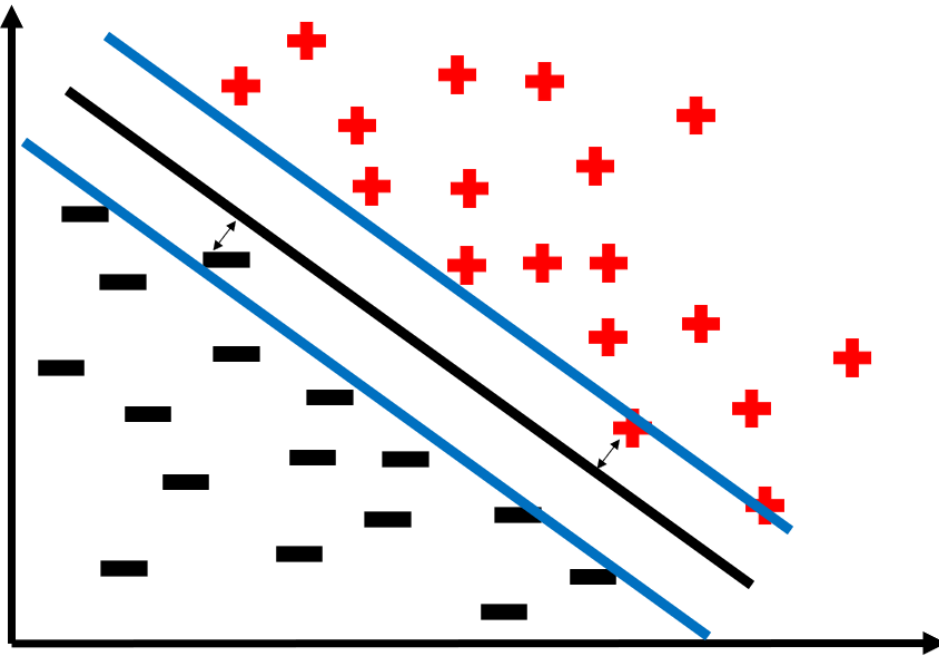


Figura 3.7: Apresentação Gráfica SVM

Este modelo de aprendizagem foi utilizado, por exemplo, para prever os riscos durante a construção de poços de fundações profundas [30] ou para estimar os custos em projetos rodoviários [31].

3.2 Aprendizagem Não Supervisionada

A aprendizagem não supervisionada, do inglês *Unsupervised Learning*, é relativamente similar ao processo de aprendizagem supervisionada, ou seja, é necessário treinar um modelo para depois se aplicar a novos dados. A principal diferença reside no facto de neste tipo de aprendizagem não existirem *labels* ou categorias para os dados. O objetivo é criar a categoria ou *label* com base nos padrões encontrados nos novos dados sem envolver a intervenção humana [32].

A Figura 3.8 demonstra um exemplo de execução de um algoritmo de *clustering* cujo objetivo é identificar o tipo de *tweet* que é realizado na rede social. Após criação de *clusters* é necessário realizar uma tarefa de pós-processamento de modo a identificar o *cluster* como contendo mensagens positivas ou negativas. De uma forma genérica considere-se que se possui um *dataset* constituído por diferentes tipos de *tweets*. O algoritmo é de aprendizagem não supervisionada recebe esse conjunto de dados, não possuindo qualquer informação sobre o tipo de categorias a que pertence cada amostra. A tarefa do algoritmo de aprendizagem não supervisionada é identificar o estado de espírito do utilizador tendo por base o seu texto escrito. Os algoritmos de aprendizagem não supervisionada realizam esta tarefa agrupando o conjunto de dados de *tweets* em grupos de acordo com as semelhanças entre os *tweets*.

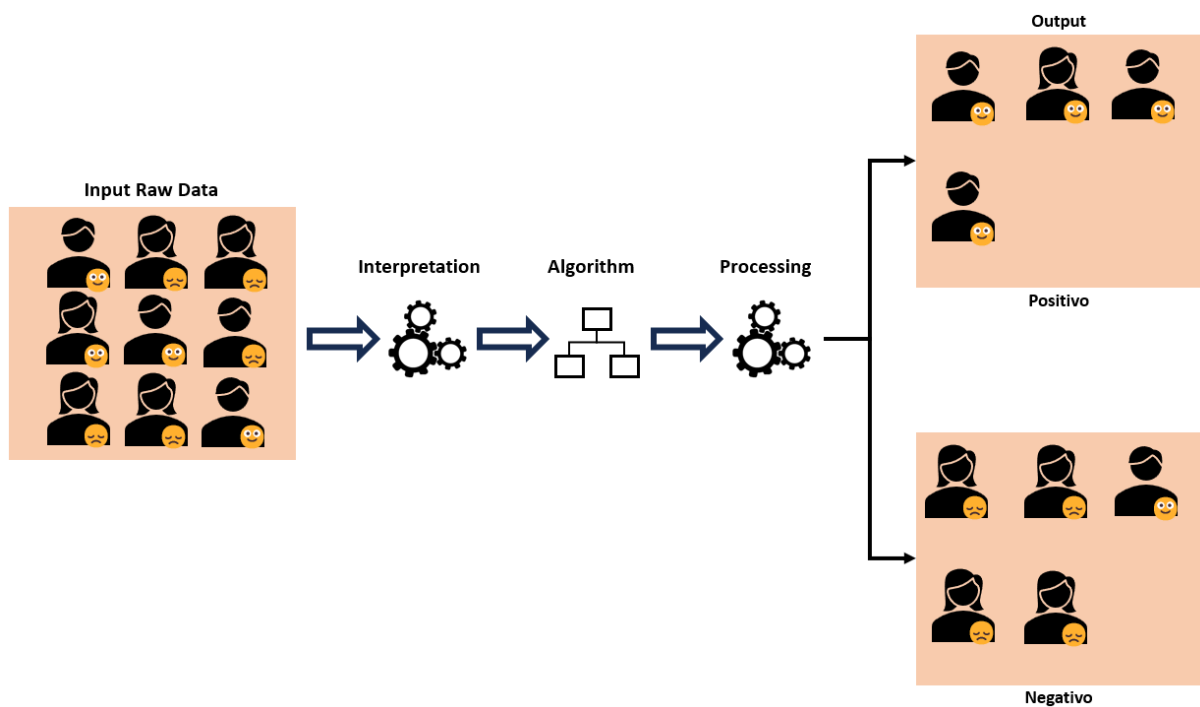


Figura 3.8: Aprendizagem Não Supervisionada

Este tipo de aprendizagem revela-se ser relevante quando não existe uma noção dos possíveis *outputs* que se querem prever para o contexto ou quando não é possível definir *à-priori* o tipo de categoria com base num conjunto genérico de dados. Por exemplo, Li, Shepperd, e Guo [33] recorreram a este tipo de aprendizagem para a previsão de defeitos de *software*, no qual concluíram que, em 14 famílias de modelos não supervisionados, o

Fuzzy CMeans (FCM) e o Fuzzy SOMs (FSOMs) têm o melhor desempenho. No entanto, reforçam a importância da qualidade dos dados uma vez que existiu um impacto nos resultados por falta de cuidado e por *reports* incompletos.

3.3 Aprendizagem por Reforço

A aprendizagem por reforço (*Reinforcement Learning*) é um método de raciocínio que se distancia dos modelos de aprendizagem supervisionada e não supervisionada. Neste tipo de aprendizagem não existe, numa fase inicial, um conjunto de dados que indique a melhor ação perante uma determinada situação. Nesta abordagem, o modelo terá de aprender numa base de tentativa e erro onde as ações mais corretas são reforçadas de forma positiva e as ações menos boas são penalizadas.

Os modelos de aprendizagem por reforço baseiam-se no *feedback* que lhes é proporcionado através de uma função de *reward*, equiparando-se a um modelo de auto-aprendizagem. O algoritmo é exposto a um ambiente dinâmico com um objetivo específico. O mesmo usa um processo de tentativa e erro onde o objetivo é maximizar a capacidade de acertar e minimizar o erro, maximizando assim a sua recompensa final.

Sallab *et al.* [34] aplicaram este tipo de aprendizagem para condução autónoma no simulador *TORCS Racing Car Simulator*, que apresentou uma adaptação adequada perante a pista de corrida.

3.4 Discussão

Perante os modelos de aprendizagem apresentados nos Capítulos anteriores, é importante clarificar as vantagens e desvantagens na utilização de cada um e o tipo de necessidade que conseguem satisfazer. Desta forma:

- **Aprendizagem Supervisionada:** A Aprendizagem Supervisionada é um dos modelos de aprendizagem mais comuns. Neste paradigma é necessário um *dataset* em que os algoritmos são treinados com um conjunto de dados rotulados, ou seja, o mo-

delo recebe dados de entradas juntamente com as suas saídas desejadas (rótulos). A principal utilidade da Aprendizagem Supervisionada reside na capacidade de realizar tarefas de previsão e classificação com base nos exemplos fornecidos durante a fase de treino. Isto é altamente eficaz em cenários onde se deseja fazer previsões com base em dados históricos.

- **Aprendizagem Não Supervisionada:** A Aprendizagem Não Supervisionada é um paradigma de aprendizagem onde os algoritmos são treinados com dados que não possuem rótulos ou categorias predefinidas. A sua utilidade está em descobrir padrões intrínsecos nos dados. Isto é particularmente valioso para a segmentação de dados, redução de dimensionalidade e deteção de anomalias. A Aprendizagem Não Supervisionada é frequentemente usada na análise exploratória de dados e na geração de *insights* a partir de grandes conjuntos de dados não estruturados, porém, como é demonstrado, necessita-se de uma análise dos dados mais aprofundados no *dataset* a utilizar.
- **Aprendizagem por Reforço:** A Aprendizagem por Reforço é um paradigma de aprendizagem no qual os algoritmos aprendem a tomar decisões através da interação com um ambiente. Eles recebem recompensas ou penalizações com base nas ações que executam. A principal utilidade da Aprendizagem por Reforço reside em situações nas quais se procura otimizar um processo sequencial, como jogar jogos, controlar robôs e tomar decisões em tempo real. Esta abordagem é especialmente apropriada para problemas nos quais as ações afetam o ambiente e onde o agente deve aprender a tomar decisões com base na experiência.

Em resumo, a Aprendizagem Supervisionada é apropriada para tarefas de previsão e classificação com dados rotulados, a Aprendizagem Não Supervisionada é usada para descobrir padrões em dados não rotulados e a Aprendizagem por Reforço é adequada para a tomada de decisões sequenciais em ambientes interativos.

Dos modelos apresentados para o propósito do projeto Edu-Code, considera-se que a aplicação de modelos de aprendizagem supervisionada será a escolha primordial, uma vez que

se pretende lidar com um problema de classificação. Os algoritmos não supervisionados poderiam ser utilizados caso se pretende-se, por exemplo, segmentar grupos de estudantes. Este propósito está, no entanto, fora dos objetivos do projeto atual.

Capítulo 4

Edu-Code: Visão Geral

No presente Capítulo aborda-se a motivação por trás do projeto, que inclui a aplicação de um questionário e a posterior análise dos dados recolhidos. É também apresentada a arquitetura conceptual da ferramenta e detalhado o processo de recolha e análise de dados para o desenvolvimento da ferramenta.

4.1 Validação da necessidade da ferramenta

Para efetuar um estudo de opinião sobre a utilização de uma ferramenta que auxilia a aprendizagem e estudo de linguagens de programação (ferramenta Edu-Code) foi desenvolvido um questionário destinado aos estudantes do ensino superior a frequentar uma licenciatura na área de informática no Instituto Politécnico do Porto, campus de Felgueiras (público-alvo). Para perceber o comportamento do público-alvo foi efetuada uma amostragem por conveniência, ou seja, foram selecionados elementos da população com base na frequência dos estudantes nas unidades curriculares de programação. Apesar da divulgação, as taxas efetivas de resposta foram baixas, tendo sido recolhida uma amostra com 44 respostas válidas.

A amostra é maioritariamente do sexo masculino, sendo apenas 4 elementos do sexo feminino, ou seja, representam 9% da amostra. Como é possível observar na Figura 4.1 uma grande parte dos estudantes da amostra que frequentam uma licenciatura na área de in-

formática no Instituto Politécnico do Porto, campus de Felgueiras, tem entre 18 e 20 anos, totalizando uma percentagem de 52,27%. Estudantes entre os 22 e 24 anos representam 22,73% da amostra. Com mais de 25 anos são 13,64% dos estudantes, representando uma parte menor da amostra.

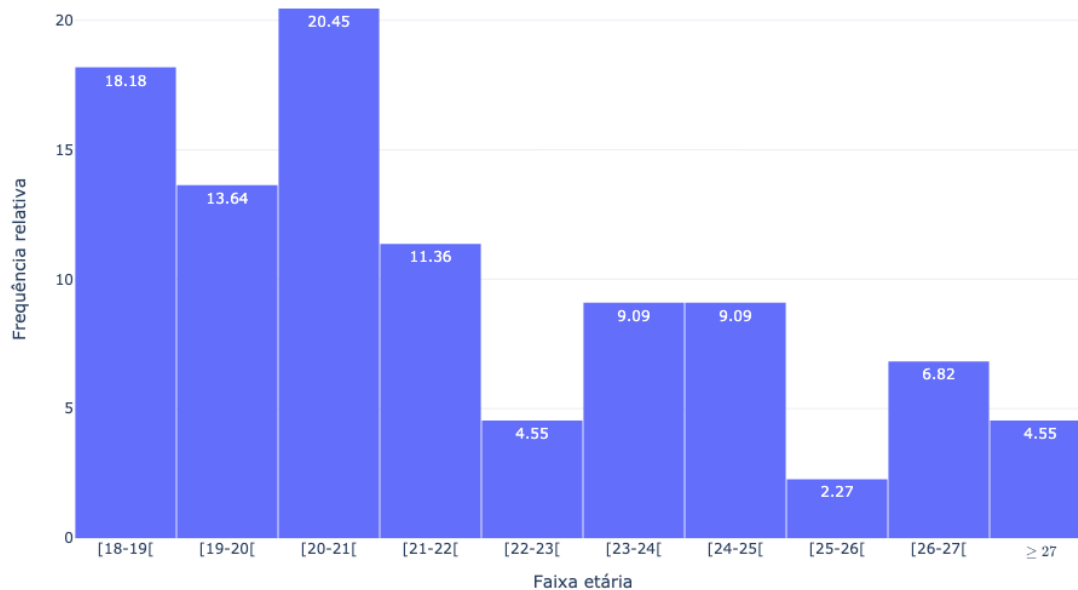


Figura 4.1: Distribuição por Faixa Etária

De acordo com a Figura 4.2 verifica-se que a maior parte dos estudantes ingressou no ensino superior através do curso de ciências e tecnologia efetuado no ensino secundário (61,1%). Os restantes estudantes fizeram um curso profissional (36,4%) e apenas uma pequena percentagem ingressou através de outro curso, nesta caso em concreto o curso de ciências sócio económicas (2,27%).

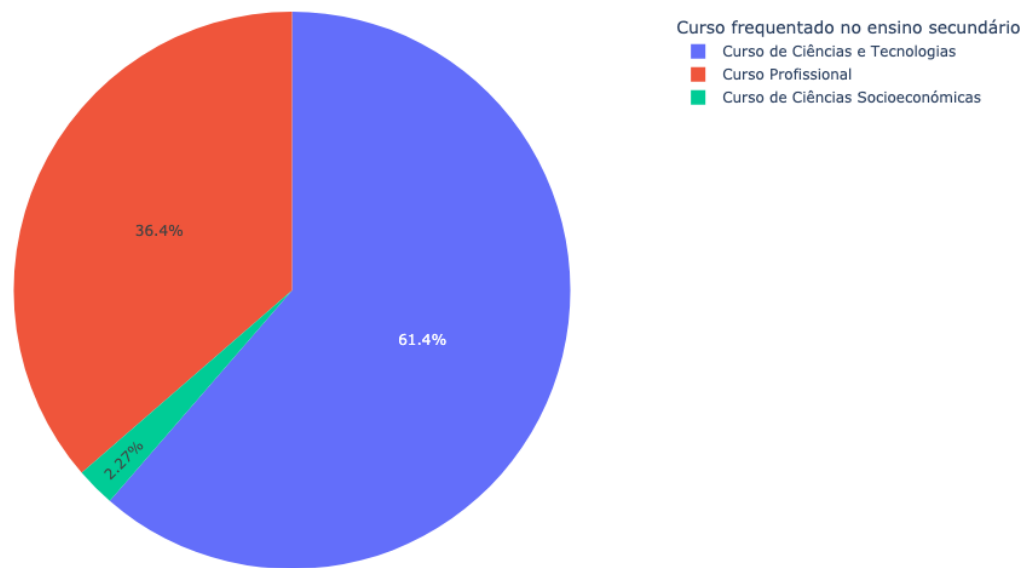


Figura 4.2: Curso Frequentado no Ensino Secundário

Pode ainda verificar-se (Figura 4.3) que um reduzido número de estudantes que ingressaram pelo curso de ciências e tecnologias frequentou um CTesP (Curso Técnico Superior Profissional), evidenciando-se uma maior frequência de estudantes num CTesP quando este realizou um curso profissional.

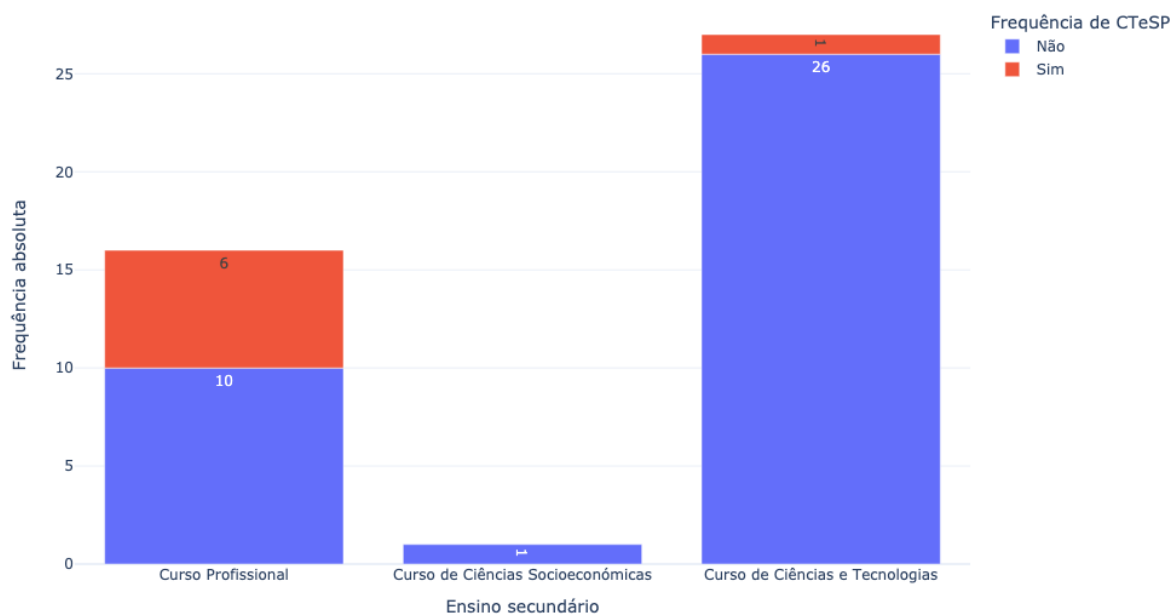


Figura 4.3: Frequência de Cursos de CTeSP

Através do histograma apresentado na Figura 4.4 pode observar-se que o intervalo de classificações (de média de entrada no curso) mais frequente (moda) é entre 14 e 16 valores exclusive com 23 observações, que corresponde a 52,3% dos estudantes. Pode ainda verificar-se que as classificações seguem uma distribuição aproximada da normal (Distribuição de Gauss) com 8 ocorrências nos intervalos imediatamente antes (entre 12 e 14 valores exclusive) e depois (entre 16 e 18 valores exclusive) do intervalo modal. Com classificações mais baixas (intervalo entre 10 e 12 valores exclusive) existem apenas 3 ocorrências. No intervalo entre 18 e 20 valores inclusive, representando as classificações mais elevadas, existem apenas 2 ocorrências.

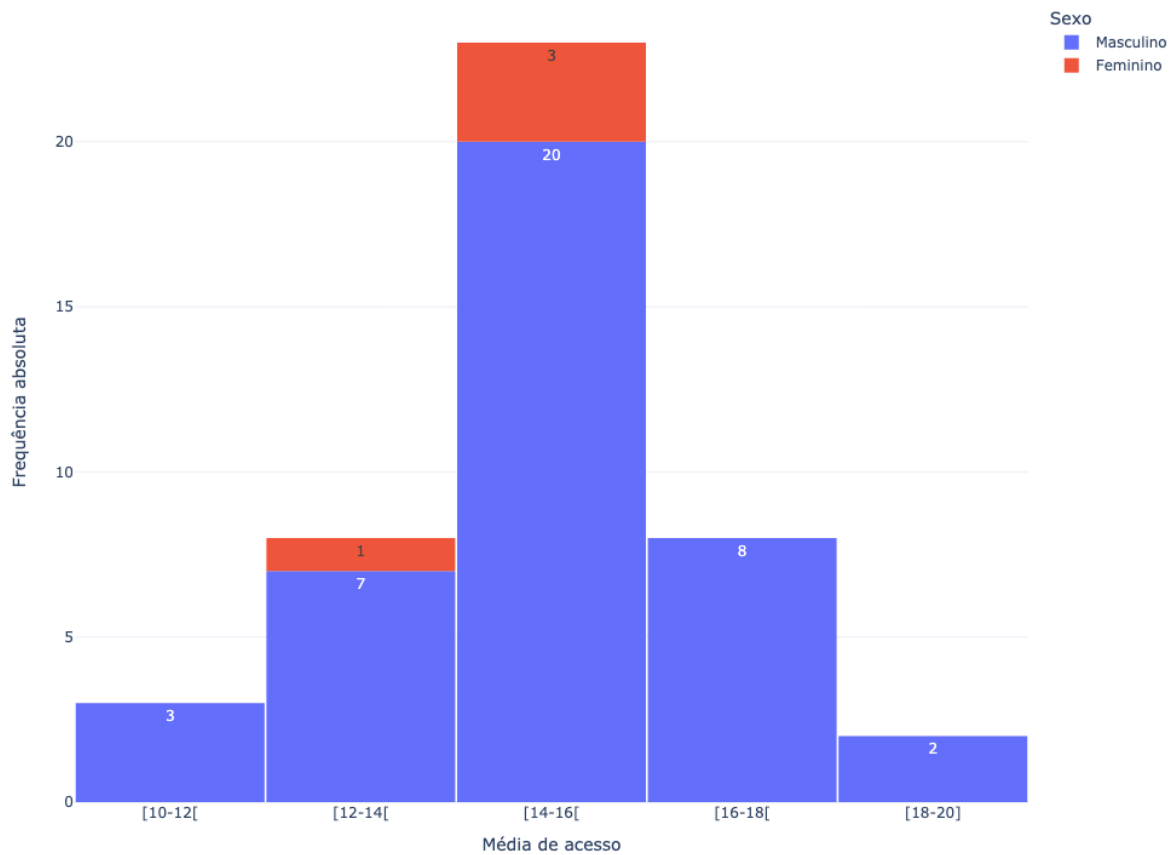


Figura 4.4: Classificações de Acesso ao Ensino Superior

Como se pode observar na Figura 4.5, 24 estudantes nunca tiveram um contacto anterior com linguagens de programação no ensino secundário (54,5%). Os estudantes que revelaram ter contacto com linguagens de programação (45,5%) estudaram maioritariamente num curso profissional (12) ou em ciências e tecnologias (8).

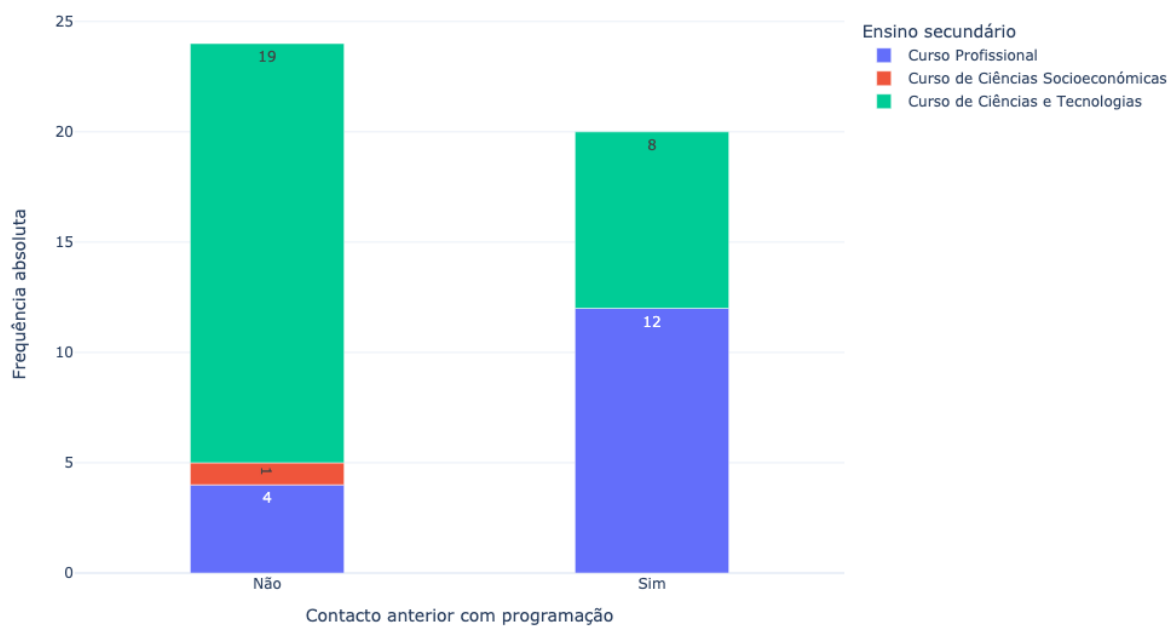


Figura 4.5: Contacto Prévio com Programação

Verifica-se, através da análise do gráfico de barras da Figura 4.6, que a maior parte dos estudantes concorda que os conhecimentos adquiridos em aulas teóricas são essenciais como base de aprendizagem, mas concordam totalmente com a importância da auto-aprendizagem, referindo que as aulas teórico-práticas são essenciais para consolidar as aprendizagens e ter sucesso às unidades curriculares que abordam a aprendizagem de linguagens de programação.

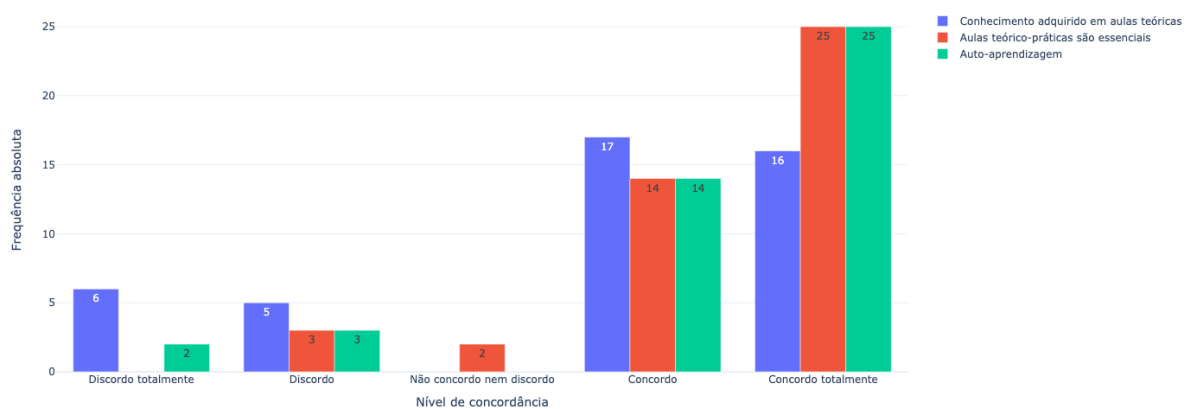


Figura 4.6: Concordância da Aquisição dos Conhecimentos

Após apresentação da proposta de ferramenta Edu-Code todos os estudantes responderam que estariam interessados em ter acesso à ferramenta (100%). Os estudantes destacaram como ponto preferencial a possibilidade de realizar testes unitários com uma média de 4,27. O “Feedback do Professor”, a “Qualidade do Código”, o “Sistema de ranking” e as “Sugestões (*hints*) de resolução” são consideradas funcionalidades importantes uma vez que apresentam uma média próxima do nível de importância 4, no entanto não existem diferenças muito relevantes entre estas (Figura 4.7).

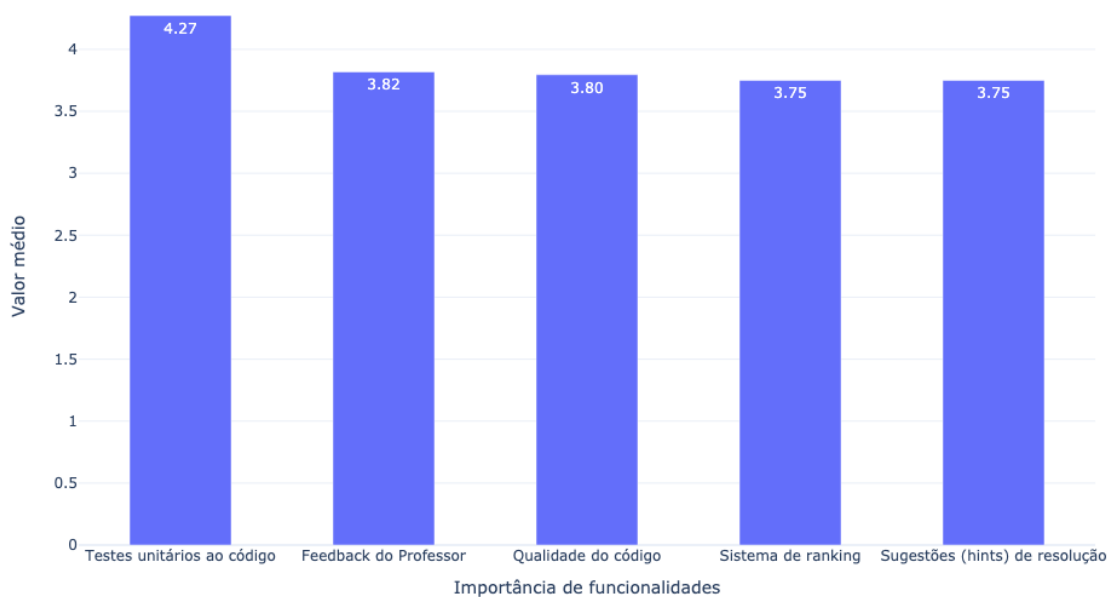


Figura 4.7: Média das Funcionalidades Requeridas

Para finalizar, os estudantes foram questionados sobre o tempo médio, em horas, que dedicavam ao estudo por semana sem acesso à ferramenta e foi ainda pedida a projeção do número de horas que iriam estudar se tivessem a ferramenta. Como se pode verificar no gráfico de barras da Figura 4.8, quem estuda menos horas, entre 0 e 2 exclusive, e entre 6 e 8 horas exclusive, iria aumentar o tempo dedicado ao estudo. Quem estuda mais de 8 horas semanais continuaria a estudar o mesmo número de horas com a ferramenta. Os estudantes que dedicam entre 2 e 6 horas exclusive consideram que iriam perder menos tempo na análise de erros. No geral, pode verificar-se que a diferença média entre o número de horas de estudo antes e a previsão após utilização da ferramenta é de 0,1 horas, o que

parece não ser significativo do ponto de vista da análise de dados, correspondendo a 5,3 e 5,4 horas, respetivamente. Se a ferramenta permitir que o tempo efetivo de estudo seja mais produtivo e menos dedicado a encontrar erros de codificação, no mesmo tempo de estudo é esperado que sejam obtidos melhores resultados. No entanto, esta questão deve ser melhor explorada após implementação prática da ferramenta numa amostra de estudantes.

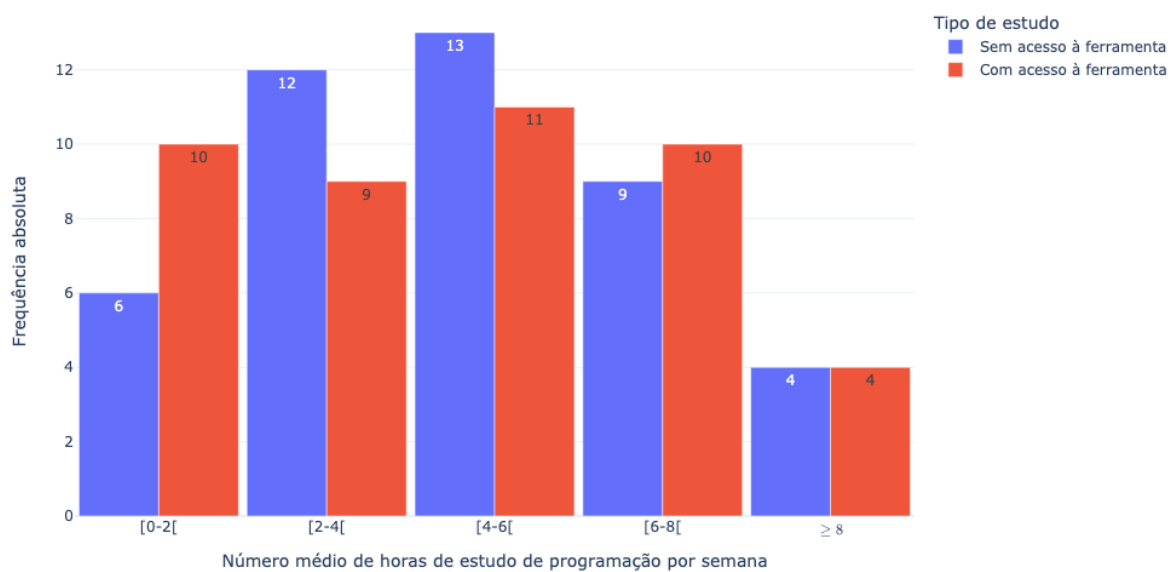


Figura 4.8: Número Médio de Horas de Estudo de Programação

4.2 Arquitetura Conceptual

O projeto **Edu-Code** teve um início prévio à presente dissertação, tendo sido realizada a publicação “A Tool to Assess Student’s Performance In Programming” [35]. Durante a continuação do seu desenvolvimento sentiu-se a necessidade de reforçar as métricas existentes no projeto para se melhorar os resultados dos modelos de *ML*, aplicando também conceitos de Engenharia de *Software* para garantir uma melhor escolha das métricas recolhidas.

Deste modo, foi necessário incorporar na arquitetura do projeto uma ferramenta externa

designada SonarQube¹. Esta ajuda a monitorizar o projeto/exercício desenvolvido ao longo do tempo pelo estudante dando um conjunto de métricas significativas.

O SonarQube é uma ferramenta de análise estática de código que desempenha um papel fundamental na garantia da qualidade do *software*. Desenvolvida pela SonarSource, esta ferramenta tem como objetivo identificar e corrigir problemas no código-fonte que podem comprometer a segurança, a confiabilidade e a manutenção de um projeto de *software*.

O SonarQube oferece uma variedade de recursos que permite às equipas de desenvolvimento melhorar a qualidade do seu código de forma contínua. Algumas das suas funcionalidades mais destacadas incluem:

- Análise Automática: O SonarQube automatiza a análise estática do código, examinando-o de forma a detetar vulnerabilidades, *bugs*, códigos duplicados e outras questões de qualidade. Esta análise poupa tempo aos desenvolvedores, uma vez que não precisam realizar tais verificações manualmente;
- Métricas e Relatórios: A ferramenta fornece uma variedade de métricas e relatórios que ajudam as equipas a entender a qualidade do seu código. Isso inclui informações sobre a cobertura de código, complexidade, conformidade com padrões de código, entre outros;
- Integração com CI/CD: O SonarQube pode ser integrado nas *pipelines* de *Continuous Integration (CI)* e *Continuous Delivery (CD)*, o que permite a deteção precoce de problemas de código e a correção imediata;
- Suporte a Múltiplas Linguagens: A ferramenta suporta uma ampla gama de linguagens de programação, tornando-a adequada para projetos heterogêneos;
- Personalização: É possível definir regras de qualidade personalizadas de acordo com as necessidades do projeto, permitindo que as equipas adotem padrões de código específicos.

Desta forma, o SonarQube é uma ferramenta essencial para qualquer equipa de desen-

¹Disponível em <https://www.sonarsource.com/products/sonarqube/>. Último acesso a 13/10/2023

volvimento que valorize a qualidade do código. Ao automatizar a detecção de problemas, fornecer métricas detalhadas e facilitar a correção de erros, o SonarQube contribui para o desenvolvimento de *software* mais seguro, eficiente e sustentável. É uma ferramenta valiosa para aperfeiçoar o processo de desenvolvimento de *software* e garantir que os projetos atinjam os mais altos padrões de qualidade.

Tendo em consideração todas as métricas/informações disponibilizadas pelo SonarQube, verifica-se que a integração desta ferramenta no desenvolvimento do projeto é uma mais-valia, uma vez que se pretende determinar/avaliar a maturidade de um estudante no desenvolvimento de aplicações/programas na linguagem de programação *Java*. Esta avaliação pretende-se que seja automática, sem intervenção humana no processo, ou seja, sem intervenção do Professor.

Para avaliar as habilidades de programação do estudante e disponibilizar a informação ao professor com informações significativas, são extraídas as seguintes métricas:

- *CPU Usage*: número inteiro que indica o tempo total de utilização da *Central Processing Unit (CPU)* para executar o código enviado;
- *Memory Usage*: O uso de memória necessária, em bytes;
- Número de compilação: O número total de compilação bem/mal sucedida do código enviado;
- Datas das submissões: Os intervalos de tempo (dia e hora) entre as submissões realizadas pelo estudante;
- Datas das compilações: Os intervalos de tempos (dia e hora) no momento da compilação pelo estudante;
- Tempo de resolução: O tempo decorrido em que o exercício é iniciado pelo estudante até o seu envio, para um determinado exercício e estudante;
- Número de submissão total: O número total de submissões feitas pelo estudante na resolução de um determinado exercício;

- Número de compilação total: O número de vezes que o estudante compilou o seu código num determinado exercício.
- Duração do rato no ecrã no exercício: O tempo decorrido em que o rato se encontra no exercício, em milissegundos;
- Duração do rato fora do ecrã do exercício: O tempo decorrido em que o rato se encontra fora do exercício, em milissegundos;
- Histórico das teclas pressionados do estudante: histórico das teclas pressionados pelo estudante na resolução do exercício. Nesta métrica é desconsiderada a tecla em si, pretendendo-se determinar, por exemplo, o intervalo que decorre entre teclas pressionadas.
- Complexidade: A complexidade refere-se à complexidade ciclomática, uma métrica quantitativa utilizada para calcular o número de caminhos através do código. Sempre que o fluxo de controlo de uma função se divide, o contador de complexidade é incrementado em um. Cada função tem uma complexidade mínima de 1. Este cálculo varia ligeiramente consoante a linguagem, porque as palavras-chave e as funcionalidades também o fazem;
- Complexidade Cognitiva: Uma métrica exclusiva do SonarQube formulada para medir com maior exatidão a compreensibilidade relativa dos métodos.
- *Duplicated Blocks*: O número de blocos de linhas duplicados.
- *Duplicated lines*: O número de linhas de código duplicados.
- *Vulnerabilities*: O número de problemas de vulnerabilidade.
- *Security Rating*: atribui uma classificação de acordo com as vulnerabilidades identificadas de acordo com a seguinte lista:
 - A - 0 vulnerabilidades
 - B - pelo menos uma vulnerabilidade do tipo *Minor*
 - C - pelo menos uma vulnerabilidade do tipo *Major*

- D - pelo menos uma vulnerabilidade do tipo *Critical*
 - E - pelo menos uma vulnerabilidade do tipo *Blocker*
- *Security Remediation Effort*: O esforço para corrigir todos os problemas de vulnerabilidade. A medida é armazenada em minutos na Base de Dados (BD). Assume-se um dia de 8 horas quando os valores são apresentados em dias;
- *Security Review Rating*: A classificação da revisão de segurança é uma nota com uma letra baseada na percentagem de Hotspots de segurança revistos. É necessário considerar que os pontos de acesso de segurança são considerados revistos se estiverem marcados como Reconhecidos, Fixos ou Seguros. A classificação é atribuída da seguinte forma:

$$\cdot A: \geq 80\%$$

$$\cdot B: \geq 70\% \text{ e } < 80\%$$

$$\cdot C: \geq 50\% \text{ e } < 70\%$$

$$\cdot D: \geq 30\% \text{ e } < 50\%$$

$$\cdot E: < 30\%$$

- *Security Hotspots Reviewed*: A percentagem de pontos críticos de segurança analisados. Esta percentagem é determinada de acordo com a equação 4.1

$$\text{Rácio} = \frac{\text{Número de pontos de acesso revistos} \times 100}{\text{Pontos de acesso a rever} + \text{pontos de acesso revistos}} \quad (4.1)$$

- *Classes*: O número de classes (incluindo classes aninhadas, interfaces, enums e anotações);
- *Comment Lines*: O número de linhas que contêm comentários ou código comentado. As linhas de comentário não significativas (linhas de comentário vazias, linhas de comentário contendo apenas caracteres especiais, entre outros) não aumentam o número de linhas de comentário;

- *Comments (%)*: Esta métrica é determinada através da equação 4.2:

$$\frac{\text{comment lines}}{\text{lines of code} + \text{comment lines}} \times 100 \quad (4.2)$$

Com base na equação apresentada conclui-se que:

- 50% significa que o número de linhas de código é igual ao número de linhas de comentário.
 - 100% significa que o ficheiro apenas contém linhas de comentário;
- Funções: O número de funções. Dependendo da linguagem de programação, uma função é definida como uma função, um método ou um parágrafo. No caso do *Java*, os métodos anónimos são ignorados (*void method*).

A arquitetura do projeto é apresentada na Figura 4.9, onde se verifica a integração do SonarQube. O sistema encontra-se dividido em duas partes, *frontend* e *backend*. A primeira permite que o utilizador, Estudante ou Professor, possa interagir com a ferramenta desenvolvida através de uma interface gráfica. Para o Estudante é possível verificar os exercícios a resolver e submeter a sua resolução obtendo posteriormente uma análise ao código introduzido. Para o Professor pretende-se a criação dos exercícios e de testes unitários com respetiva indicação/sugestão de correção. No *backend* encontra-se a BD, responsável por armazenar toda a informação necessária para a correta execução da aplicação e o SonarQube que analisa e extrai as métricas previamente definidas.

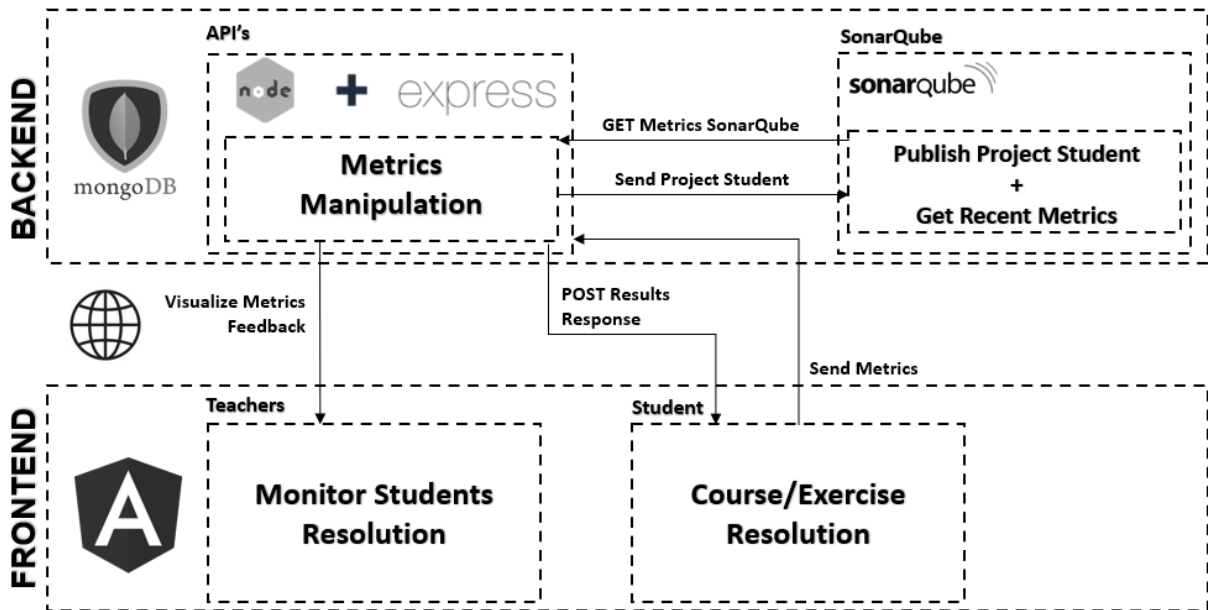


Figura 4.9: Arquitetura do Projeto com Integração SonarQube

Quando o estudante submete o exercício, é enviada a informação das alterações efetuados para o servidor que contém o seu projeto. Ao executar os testes unitários (previamente definidos pelo professor) é possível validar se o exercício se encontra corretamente resolvido. De forma paralela, o código submetido pelo estudante é também enviado, em forma de projeto *Java*, para a ferramenta SonarQube para que esta possa extrair as métricas necessárias, importantes para futura utilização nos modelos de *ML*.

4.3 Recolha de Dados

Antes de efetuar a recolha de dados, é fundamental identificar e definir os dados a recolher e o seu respetivo tipo, permitindo assim um correto armazenamento na BD. A BD, definida em MongoDB, armazena todas as ações do estudante perante o exercício disponibilizado. A definição dos atributos a serem utilizados encontra-se apresentada na Tabela 4.1.

Tabela 4.1: Lista dos Atributos e Tipo de Dados Simplificado

Atributos	Descrição	Tipo
_idStudent	Id do Estudante	mongoose.ObjectId
_idCourseExercise	Id do Exercise	mongoose.ObjectId
isOnline	O Estudante está a resolver a ficha?	Booleano
files	Ficheiros utilizados para o exercício	Lista de Objeto
timeStudentMouseDownExercise	Tempo do rato no exercício em milissegundos	Lista Numérico
timeStudentMouseLeaveExercise	Tempo do rato fora do exercício em milissegundos	Lista Numérico
keyUserTyping	Histórico das teclas pressionados	Lista de Texto
cpuUsageCompilation	Utilização de CPU na compilação	Lista de Objeto
memoryUsageCompilation	Utilização de Memória na compilação	Lista de Objeto
cpuUsageSubmission	Utilização de CPU na submissão	Lista de Objeto
memoryUsageSubmission	Utilização de Memória na submissão	Lista de Objeto
sonarqubeMetrics	Métricas do SonarQube na submissão	Lista de Objeto
numCompilation	Número de compilação com sucesso ou não	Lista Booleano
numSubmission	Número de submissão com sucesso ou não	Lista Booleano
dateCompilation	Data que o estudante compilou	Lista de Datas
dateSubmission	Data que o estudante submeteu	Lista de Datas
startWorksheet	Tempo que decorreu ao resolver o exercício em milissegundos	Numérico
timeResolution	Tempo de resolução do exercício em milissegundos	Numérico

Dos atributos apresentados existem alguns que são do tipo “Lista de Objeto” uma vez que o MongoDB tem a particularidade de armazenar os dados em coleções. Essas coleções representam como um objeto *JavaScript Object Notation (JSON)*, definido como chave valor, que apresenta um conjunto de informações mais detalhadas sobre os mesmos. Estes objetos foram definidos de acordo com os objetivos inicialmente estabelecidos para o desenvolvimento do projeto atual. Na Tabela 4.2 apresenta-se em detalhe o conteúdo para os atributos definidos como Objeto.

Tabela 4.2: Lista dos Atributos e Tipo de Dados Complexo

Objeto	Atributos	Descrição	Tipo
cpuUsage	user	Tempo decorrido pelo utilizador	Numérico
cpuUsage	system	Tempo decorrido pelo sistema	Numérico
memoryUsage	rss	Quantidade de espaço ocupado na memória principal	Numérico
memoryUsage	heapTotal	Utilização da Memória V8	Numérico
memoryUsage	heapUsed	Utilização da Memória V8	Numérico
memoryUsage	external	Utilização da Memória gerido pelo V8	Numérico
memoryUsage	arrayBuffers	Memória atribuída para ArrayBuffers, SharedArrayBuffers e Buffers Node.js	Numérico
sonarqubeMetrics	comment_lines_density	O número de linhas comentados	Numérico
sonarqubeMetrics	complexity	Complexidade ciclomática	Numérico
sonarqubeMetrics	comment_lines	O número de linhas comentados em %	Numérico
sonarqubeMetrics	vulnerabilities	O número de problemas na vulnerabilidade	Numérico
sonarqubeMetrics	functions	Número de funções	Numérico
sonarqubeMetrics	duplicated_lines	O número de linhas duplicados	Numérico
sonarqubeMetrics	duplicated_blocks	O número de blocos duplicados	Numérico
sonarqubeMetrics	duplicated_lines_density	O número de linhas duplicados em %	Numérico
sonarqubeMetrics	cognitive_complexity	Compreensibilidade relativa dos métodos	Numérico
sonarqubeMetrics	classes	Número de classes	Numérico
sonarqubeMetrics	security_remediation_effort	Nível de esforço para corrigir os problemas	Numérico
sonarqubeMetrics	reliability_rating	Classificação de Fiabilidade	Numérico
sonarqubeMetrics	security_review_rating	Classificação da Segurança	Numérico
sonarqubeMetrics	sqale_rating	Classificação do Technical Debt Ratio	Numérico
sonarqubeMetrics	security_rating	Classificação da revisão de segurança	Numérico
sonarqubeMetrics	datetime	Data da submissão	Data

Para se demonstrar o funcionamento da ferramenta desenvolvida, e de modo a poste-

riormente validar os modelos de *ML* selecionados, foi necessário proceder ao fabrico de dados de teste. Este desenvolvimento teve como objetivo realizar uma pré-validação do sistema desenvolvido antes de se tentar introduzir o mesmo num cenário de teste em ambiente real, permitindo determinar antecipadamente possíveis erros ou melhorias a realizar. Desta forma, foram criados 6 perfis de estudantes do seguinte modo:

(A) 2 Estudantes Principiantes

(B) 2 Estudantes Médios

(C) 2 Estudantes Experientes

Cada um destes perfis tem um propósito diferente em níveis de avaliação de desempenho, realçando que o modelo de exercício avaliativo é igual para todos. A simulação de resolução do exercício proposto teve em consideração o perfil identificado para cada estudante (principiante, médio, experiente) considerando fatores como, por exemplo:

- Nível de Aprendizagem;
- Agilidade na resolução do problema;
- Advertência na resolução, por início de estudo;
- Nível de captação e aplicação dos conhecimentos adquiridos anteriormente.

Os estudantes experientes refletiram o seu grau de conhecimento na linguagem de programação *Java*, solucionando o problema de força célere. Por outro lado, estudantes iniciantes nesta linguagem de programação são mais propensos a erros e, conseqüentemente, maiores tempos de resolução. Outra variável considerada relevante neste estudo é o número de compilação e de submissão. Através de alguma observação direta, verificou-se que os estudantes, em particular os iniciantes, quando confrontados com um novo problema de programação tendem, por vezes, a solucioná-lo por tentativa e erro. Este processo demonstra, por exemplo, falta de aquisição de competências, desconcentração e provoca um aumento no tempo de resolução derivado, por exemplo, pela ocorrência de erros mínimos (por exemplo, ponto e virgular no fim de uma linha de código).

Tendo em consideração estas características de cada perfil de estudante definiram-se as condições médias para cada tipo de perfil como demonstrado na Tabela 4.3.

Tabela 4.3: Distinção dos Estudantes

	Principiante	Médio	Experiente
Tempo de Resolução	[5min.-10min.]	[3min.-5min.]	[2min.-3min.]
Nº Compilação	[2-10]	[2-10]	[2-4]
Nº Submissão	[2-3]	[2-3]	[1]

Com base nestes critérios mencionados foram fabricados 4 *datasets* (em formato excel) aos quais foram atribuídos estudantes diferentes. Isto possibilitou a obtenção de uma maior variabilidade de dados para avaliação nos modelos de *ML*. Os *datasets* possuem um número de estudantes e um número de amostras bastante distinto como se observa na Tabela 4.4.

Tabela 4.4: Informação Detalhada dos Datasets

Nome do Dataset	Nº Estudante	Total de Linhas
Dataset1	50	219
Dataset2	100	353
Dataset3	200	783
Dataset4	500	2056

4.4 Análise de Dados

No caso do projeto Edu-Code, é fundamental identificar que tipo de tratamento de dados é necessário aplicar para o propósito do problema. Este processo tem como objetivo garantir uma melhor previsão nos resultados esperados. Pretende-se, entre outros objetivos, verificar se o estudante se encontra num nível principiante, médio ou experiente, ou seja, o objetivo é realizar uma classificação dado um conjunto de variáveis. Com o modelo de *ML* que será aplicado tem-se como objetivo determinar os melhores valores para resolver da função apresentada em 4.3.

$$f(x_1, x_2, \dots, x_n) = [y_1, y_2, y_3] \quad (4.3)$$

Desta forma, dados os valores dos vários atributos x_1, \dots, x_n pretende-se determinar o

melhor valor de y_1 , y_2 e y_3 que indica se o estudante é considerado iniciante, médio ou experiente, respetivamente. O valor de y_i toma um valor entre 0 e 1, indicando a probabilidade de um estudante pertencer a um determinado grupo.

Ao estabelecer os *outputs* desejados, é preciso aplicar os conceitos de *data engineering*. Este é um processo fundamental no desenvolvimento e na implementação bem-sucedidos de modelos de *ML*. Esta etapa inclui a preparação, transformação e armazenamento dos dados necessários para treinar e fazer inferências com modelos de *ML*. Numa fase inicial devem remover-se as colunas que não possuem informação relevante para o modelo, nomeadamente os atributos apresentados na Tabela 4.5.

Tabela 4.5: Data Engineering - Atributos Removidos

Atributos	Descrição	Tipo
isOnline	O Estudante está a resolver a ficha?	Booleano
_idStudent	Id do Estudante	mongoose.ObjectId
_idCourseExercise	Id do Exercise	mongoose.ObjectId
files	Ficheiros utilizados para o exercício	Lista de Objeto
updated	Ultima Atualização	Data
startWorksheet	Tempo que decorreu ao resolver o exercício em milissegundos	Numérico
__v	Versão do Registo	Numérico

De seguida, foi executado uma ordenação das fases de compilação e submissão pelas suas respetivas datas para cada Estudante. Este passo foi necessário dado que estes atributos são armazenados em dois atributos (objetos) separados, sendo possível atribuir os seguintes valores/atributos *numCompilation*, *numSubmission*, *dateCompilation*, *dateSubmission* num só objeto. Desta forma, os dados são uniformizados, sendo este processo necessário visto que não foi considerado esta abordagem numa fase inicial do projeto desenvolvido.

Após esta etapa, foi necessário fabricar novos dados com base no perfil dos 6 estudantes iniciais, simulando cada um dos níveis de programação de programação. Com base nestas informações, utilizou-se o mínimo e máximo valor de cada atributo para gerar os novos Estudantes a resolver o exercício.

Por último, criou-se uma nova coluna designada “ExperienceLevel” que identifica que Estudantes são principiantes, médios ou experientes. Esta coluna foi preenchida de acordo com o tempo de resolução, número de compilações e submissões para classificar os Estudantes gerados. Todo o procedimento é demonstrado na Figura 4.10:

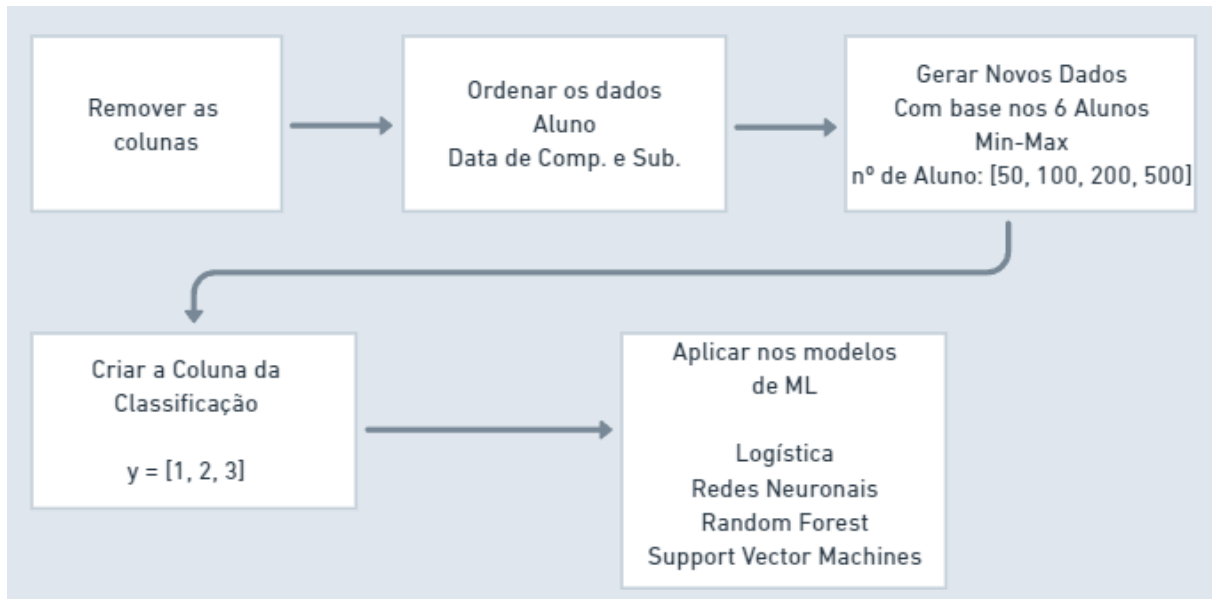


Figura 4.10: Workflow do Processo de Tratamento de Dados

Para perceber a importância dos dados gerados e descritos no parágrafo anterior, é demonstrado nas Tabelas 4.6-4.9 as medidas de tendência central e de dispersão dos dados de cada *dataset* gerado.

Tabela 4.6: Dataset1 - Estatísticas

Nome Métrica / Estatísticas	<i>mean</i>	<i>std</i>	<i>min</i>	25%	50%	75%	<i>max</i>
TimeResolution	299442.42	109361.96	43000	211448.75	306785.5	380628.5	464000
SonarqubeMetric_comment_lines_density	11.021	8.05	0	2.80	10.92	18.56	24.6
SonarqubeMetric_complexity	6.89	4.88	0	2.75	7	11	16
SonarqubeMetric_comment_lines	4.74	3.16	0	2.00	5	7	10
SonarqubeMetric_vulnerabilities	10.87	7.18	0	5.03	10	17.60	23.5
SonarqubeMetric_functions	4.49	3.27	0	1	5	7	10
SonarqubeMetric_duplicated_lines	7.40	5.03	0	3	7	11	16
SonarqubeMetric_duplicated_blocks	13.09	8.31	0	5.60	12.68	20.05	27.90
SonarqubeMetric_duplicated_lines_density	11.43	7.58	0	5.78	10.58	19.05	23.50
SonarqubeMetric_cognitive_complexity	10.33	7.40	0	4.13	8.50	16.75	24.60
SonarqubeMetric_classes	10.76	6.82	0	5.028	11.15	15.88	23.50
SonarqubeMetric_security_remediation_effort	12.32	7.73	0	4.98	13.45	19.13	24.60
SonarqubeMetric_reliability_rating	3.42	2.15	0	1.25	3.00	5.53	7.00
SonarqubeMetric_security_review_rating	11.72	7.42	0	5.42	10.66	19.40	23.50
SonarqubeMetric_sqale_rating	9.85	6.88	0	3.88	9.15	15.60	23.50
SonarqubeMetric_security_rating	0.83	0.86	0	0	1	2	2

Tabela 4.7: Dataset2 - Estatísticas

Nome Métrica / Estatísticas	<i>mean</i>	<i>std</i>	<i>min</i>	25%	50%	75%	<i>max</i>
<i>TimeResolution</i>	299570	106309.93	43000	211492.5	305899.5	395306	464000
<i>SonarqubeMetric_comment_lines_density</i>	11.70	7.69	0	4.60	11.51	18.27	24.6
<i>SonarqubeMetric_complexity</i>	7.78	5.17	0	3	8	12	16
<i>SonarqubeMetric_comment_lines</i>	4.73	3.21	0	2	5	7	10
<i>SonarqubeMetric_vulnerabilities</i>	10.81	6.96	0	5	10.52	16.79	23.5
<i>SonarqubeMetric_functions</i>	4.85	3.28	0	2	5	8	10
<i>SonarqubeMetric_duplicated_lines</i>	6.91	4.88	0	2.75	7	11	16
<i>SonarqubeMetric_duplicated_blocks</i>	13.46	8.22	0	7.24	13.66	20.14	27.90
<i>SonarqubeMetric_duplicated_lines_density</i>	11.47	7.25	0	4.62	11.59	18.26	23.50
<i>SonarqubeMetric_cognitive_complexity</i>	11.54	6.93	0	6.44	11.57	17.02	24.6
<i>SonarqubeMetric_classes</i>	10.913	7.01	0	5.03	9.55	17.27	23.5
<i>SonarqubeMetric_security_remediation_effort</i>	11.67	7.38	0	5.02	11.61	17.80	24.6
<i>SonarqubeMetric_reliability_rating</i>	3.09	1.98	0	1.35	2.89	4.51	7
<i>SonarqubeMetric_security_review_rating</i>	11.06	6.78	0	4.57	11.29	16.33	23.5
<i>SonarqubeMetric_sqale_rating</i>	11.94	7	0	6.18	12.38	17.61	23.5
<i>SonarqubeMetric_security_rating</i>	1	0.84	0	0	1	2	2

Tabela 4.8: Dataset3 - Estatísticas

Nome Métrica / Estatísticas	<i>mean</i>	<i>std</i>	<i>min</i>	25%	50%	75%	<i>max</i>
<i>TimeResolution</i>	287693.89	100683.98	43000	203268	287043	372693	464000
<i>SonarqubeMetric_comment_lines_density</i>	11.76	7.30	0	5.42	11.52	17.98	24.6
<i>SonarqubeMetric_complexity</i>	7.77	4.93	0	4	7	12	16
<i>SonarqubeMetric_comment_lines</i>	4.76	3.28	0	2	5	7	10
<i>SonarqubeMetric_vulnerabilities</i>	11.64	7.01	0	5.75	12	17.98	23.5
<i>SonarqubeMetric_functions</i>	4.99	3.26	0	2	5	8	10
<i>SonarqubeMetric_duplicated_lines</i>	7.74	5.14	0	3	8	12	16
<i>SonarqubeMetric_duplicated_blocks</i>	14.25	8.29	0	7.01	14.06	21.36	27.9
<i>SonarqubeMetric_duplicated_lines_density</i>	11.65	6.94	0	6.13	11.43	17.64	23.5
<i>SonarqubeMetric_cognitive_complexity</i>	11.75	7.17	0	5.84	11.68	18.05	24.6
<i>SonarqubeMetric_classes</i>	11.41	6.9	0	5.78	10.69	17.56	23.5
<i>SonarqubeMetric_security_remediation_effort</i>	12.35	7.39	0	5.82	12.6	18.79	24.6
<i>SonarqubeMetric_reliability_rating</i>	3.44	1.96	0	1.8	3.45	5.09	7
<i>SonarqubeMetric_security_review_rating</i>	11.72	6.82	0	6.08	11.62	17.23	23.5
<i>SonarqubeMetric_sqale_rating</i>	11.65	7.02	0	4.97	11.92	18.06	23.5
<i>SonarqubeMetric_security_rating</i>	1.04	0.85	0	0	1	2	2

Tabela 4.9: Dataset4 - Estatísticas

Nome Métrica / Estatísticas	<i>mean</i>	<i>std</i>	<i>min</i>	25%	50%	75%	<i>max</i>
<i>TimeResolution</i>	295457.13	100400.9	43000	207904	304196.5	381666.5	464000
<i>SonarqubeMetric_comment_lines_density</i>	12.35	7.3	0	5.84	12.4	18.95	24.6
<i>SonarqubeMetric_complexity</i>	7.90	4.87	0	4	8	12	16
<i>SonarqubeMetric_comment_lines</i>	5.03	3.21	0	2	5	8	10
<i>SonarqubeMetric_vulnerabilities</i>	11.92	6.9	0	5.94	11.96	17.81	23.5
<i>SonarqubeMetric_functions</i>	4.96	3.16	0	2	5	8	10
<i>SonarqubeMetric_duplicated_lines</i>	7.83	4.96	0	3	8	12	16
<i>SonarqubeMetric_duplicated_blocks</i>	13.9	8.12	0	6.69	14.06	20.76	27.9
<i>SonarqubeMetric_duplicated_lines_density</i>	11.62	6.88	0	5.625	11.685	17.31	23.5
<i>SonarqubeMetric_cognitive_complexity</i>	12.49	7.16	0	6.29	12.54	18.99	24.6
<i>SonarqubeMetric_classes</i>	12.25	6.82	0	6.39	12.78	17.83	23.5
<i>SonarqubeMetric_security_remediation_effort</i>	12.04	7.13	0	5.705	11.79	18.32	24.6
<i>SonarqubeMetric_reliability_rating</i>	3.47	2	0	1.81	3.39	5.15	7
<i>SonarqubeMetric_security_review_rating</i>	11.74	6.96	0	5.62	11.55	17.59	23.5
<i>SonarqubeMetric_sqale_rating</i>	11.48	6.81	0	5.60	11.13	17.56	23.5
<i>SonarqubeMetric_security_rating</i>	1.01	0.84	0	0	1	2	2

Capítulo 5

Edu-Code: Solução Proposta

Neste Capítulo são apresentados os principais elementos da aplicação desenvolvida, destacando a perspectiva tanto do estudante quanto do professor. Por último, é apresentado as diferentes configurações utilizados para os modelos de *ML* e as diversas conclusões extraídas dos resultados obtidos.

5.1 Aplicação Desenvolvida

O projeto Edu-Code é uma plataforma *web* onde se aplica o conceito de *ITE*, ou seja, o principal objetivo da solução é permitir que o professor possua ferramentas para, por exemplo, num cenário de ensino remoto, avaliar com maior precisão a qualidade da transmissão dos conhecimentos, percebendo se os estudantes estão de forma efetiva a adquirir as competências necessárias para aprender uma linguagem de programação. Desta forma, durante a realização de um exercício por um estudante, são extraídas várias métricas que se disponibilizam ao professor (em tempo real) e, desta forma, é possível verificar se o estudante é mais ou menos proficiente em programação. As métricas como, por exemplo, número de erros de compilação ocorridos para um determinado exercício possibilitam que o professor possa verificar que estudantes podem necessitar de mais suporte ou garantir que todos os estudantes estão a assimilar os conteúdos lecionados. A interação estudante-professor é melhorada e estimulada por meio de um sistema que atua como *middleware*

entre eles, diminuindo situações onde, por vergonha, o estudante não questiona o professor e, conseqüentemente, mantém as dúvidas existentes. Com recurso ao sistema desenvolvido, o professor pode questionar diretamente o estudante sobre um determinado exercício que, *à priori*, tem conhecimento que existem dificuldades.

Antes de apresentar a perspectiva dos diferentes utilizadores, por uma questão de compreensão das figuras que serão mencionadas nos próximos subtópicos, é explicado o cenário do exercício escolhido para demonstração da ferramenta. Desta forma, considera-se que existe um curso na plataforma denominado “PPCurso” que contém uma ficha prática com apenas um exercício. O exercício é baseado na criação de uma calculadora muito simples na LP *Java* com o objetivo do estudante implementar as 4 operações aritméticas básicas (somar, subtrair, multiplicar e dividir). O estudante terá de implementar os quatro métodos na classe “Calculator.java” em que cada um destes métodos representa uma dessas operações. Após os cenários feitos, serão apresentadas as métricas da simulação do estudante.

O exercício utilizado como demonstração contém a estrutura de projeto apresentada na Figura 5.1.

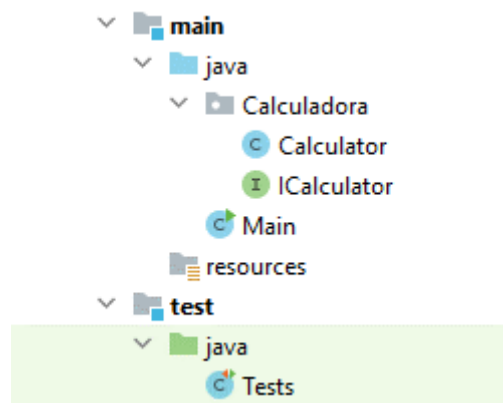


Figura 5.1: Estrutura do Projeto da Calculadora

5.1.1 Perspetiva do estudante

O estudante, após iniciar sessão na plataforma, inicia o exercício da calculadora, obtendo o projeto base previamente existente no servidor (Figura 5.2).

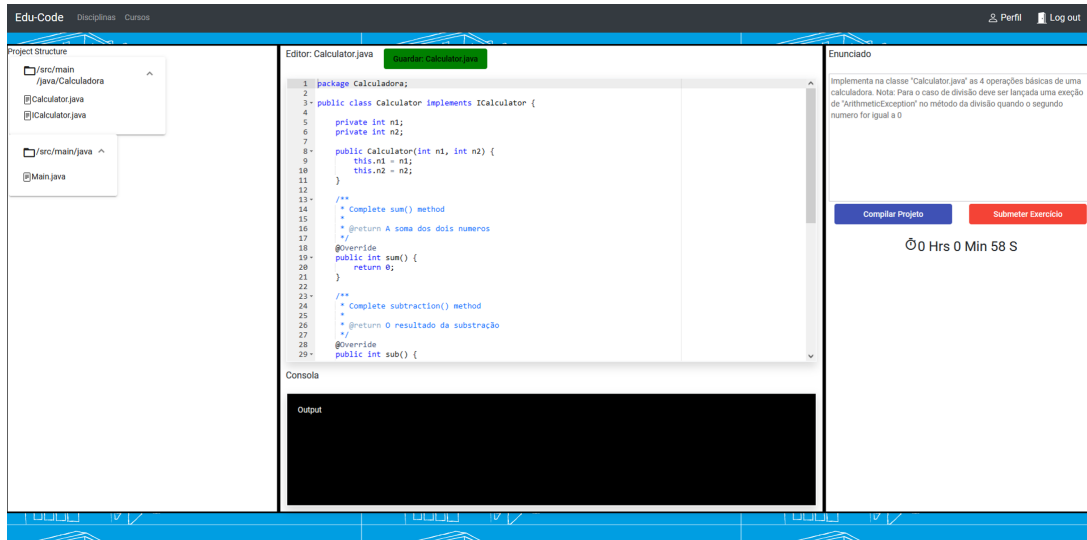


Figura 5.2: Página Inicial da Interação com um Exercício

No lado esquerdo da figura apresenta-se uma visão mais simplificada da estrutura do projeto com os seus conteúdos, sem dar conhecimento ao estudante, que existem testes unitários no projeto. No centro é apresentado o conteúdo do ficheiro “Calculator.java” no qual o estudante pode proceder à sua edição (inserção do código implementado) e guardar as alterações. Por baixo desta janela existe uma simulação de consola que indica os *outputs* da compilação e submissão. No lado direito é possível visualizar o enunciado a designar a tarefa a realizar no exercício. É ainda possível escolher se se pretende apenas compilar o projeto ou submeter o exercício com um tempo a decorrer na interação do exercício. Quando o estudante guardar as mudanças de um determinado ficheiro será dado um *alert* do sucesso dessa alteração, demonstrado na Figura 5.3.

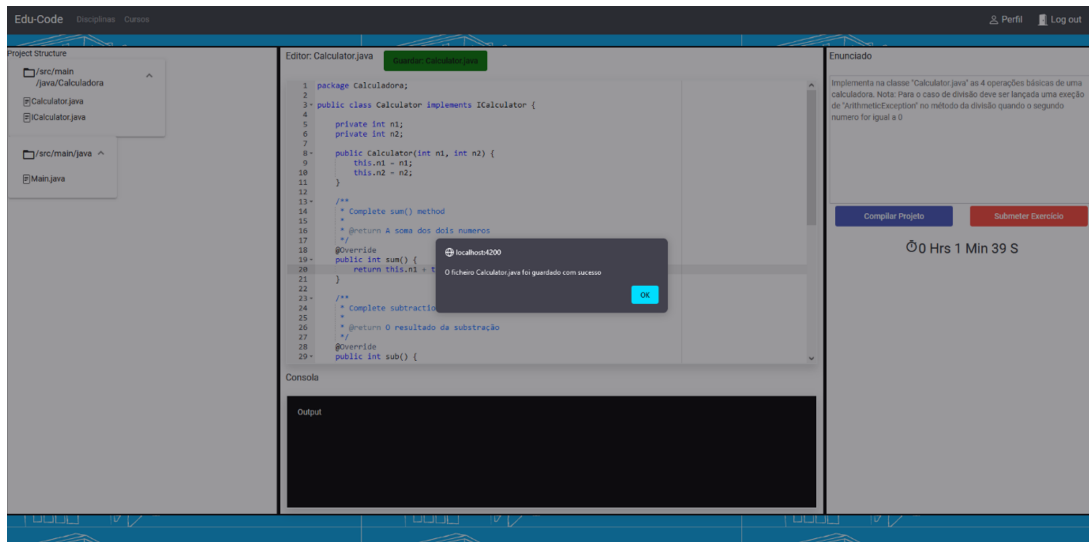


Figura 5.3: Demonstração do *alert* ao Guardar um Ficheiro

Após inserção do código que solucionará o problema proposto, o estudante terá de compilar o seu projeto para verificar se efetivamente o método que acabou de editar apresenta o resultado previsto. Assim, o estudante deverá carregar no botão “Compilar Projeto” para verificar o *output*, como demonstrado na Figura 5.4.

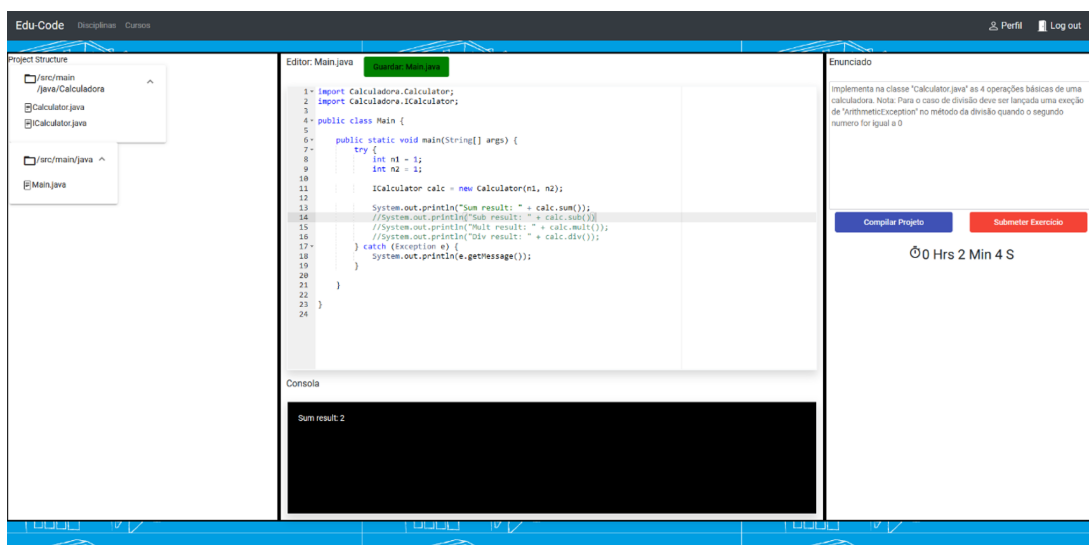


Figura 5.4: Resultado da Compilação do Exercício

De acordo com o código inserido na Figura 5.4, o resultado esperado da soma foi bem-sucedida. De modo a demonstrar uma submissão sem sucesso, apresenta-se na Figura 5.5 um cenário onde o estudante não cumpre com os objetivos do exercício. Desta forma, como previsto devido ao erro de sintaxe ocorrido (falta de ‘;’ no final da instrução) surge uma mensagem de erro de compilação a identificar o erro.

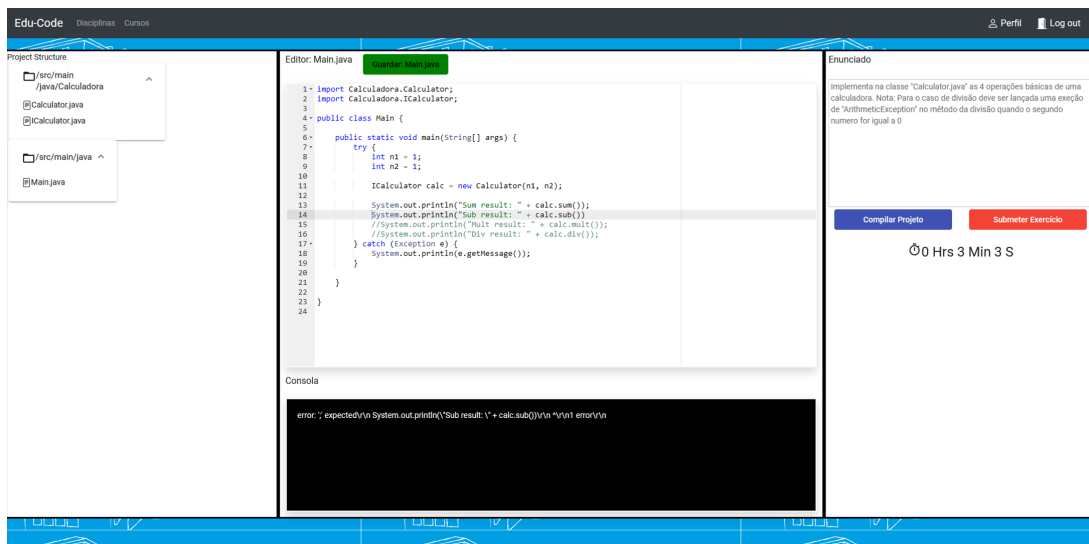


Figura 5.5: Demonstração de um Erro de Compilação

Após compilação do projeto e correção de erros de sintaxe, o estudante estará em condições de submeter o projeto para validação final. Para demonstrar os possíveis cenários considere-se que o estudante já inseriu todas as instruções necessárias ao cumprimento dos requisitos do enunciado. No entanto, na operação da divisão não considerou o caso particular da divisão por zero. Durante a execução do projeto neste estado irá surgir a exceção “ArithmeticException”, como demonstrado na Figura 5.6.

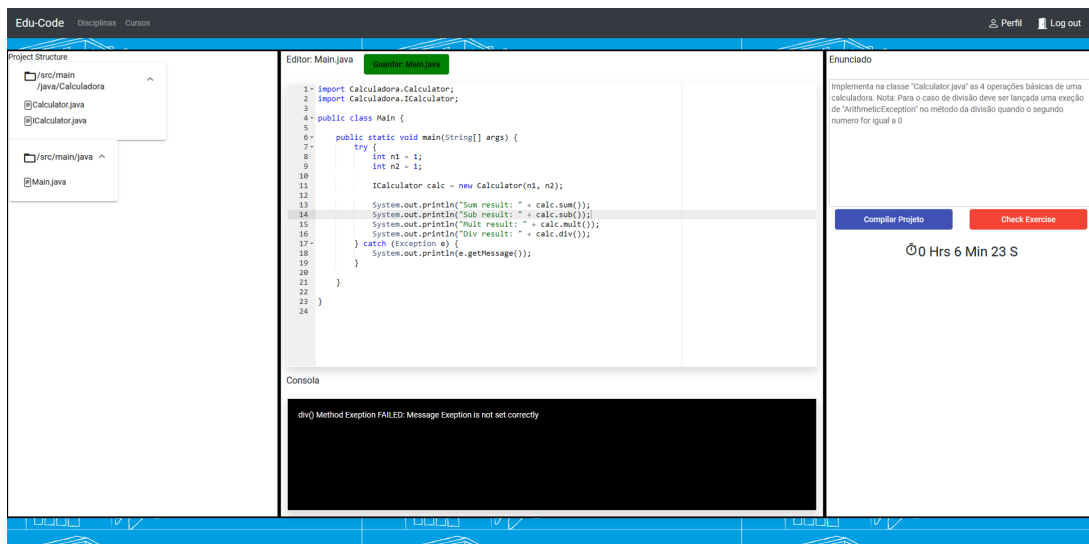


Figura 5.6: Submissão do Exercício Incompleto

O erro ocorrido aparece no *output* na consola e identifica o método que falta completar com uma mensagem personalizada. Essa mensagem é originada pelos testes unitários previamente feitos pelo professor que criou o exercício. A mensagem de erro permite direcionar o estudante para a correta implementação de código e, após submissão e validação de todos os testes, é indicado ao estudante a correta execução do exercício (Figura 5.7).

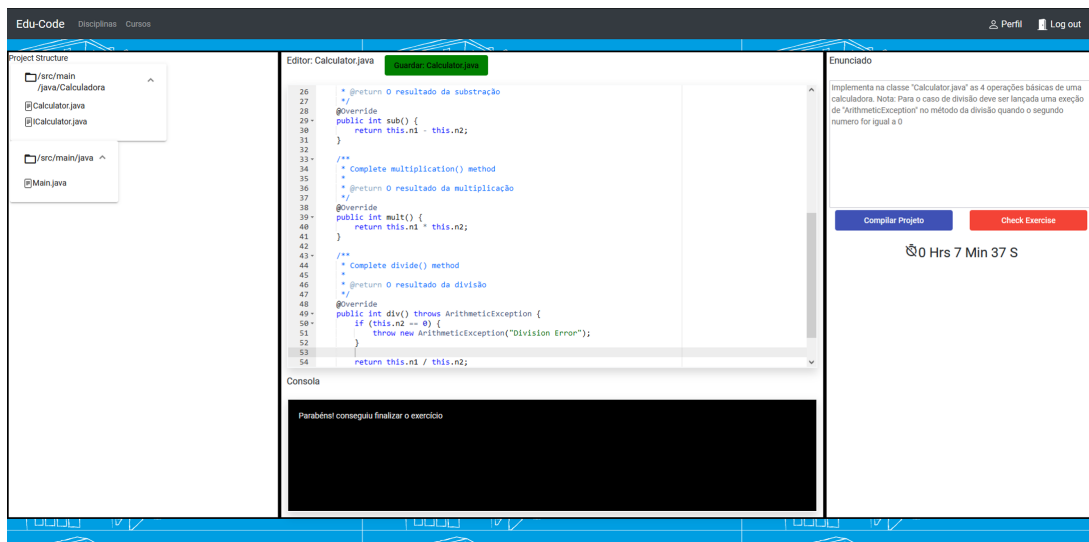


Figura 5.7: Submissão do Exercício Completo

5.1.2 Perspetiva do Professor

Após o professor iniciar a sessão e escolher a opção de supervisionar a resolução de um determinado exercício, consegue identificar que estudantes estão, naquele momento, a

resolver a ficha e visualizar algumas métricas como, por exemplo, número de submissões com/sem sucesso (Figura 5.8).

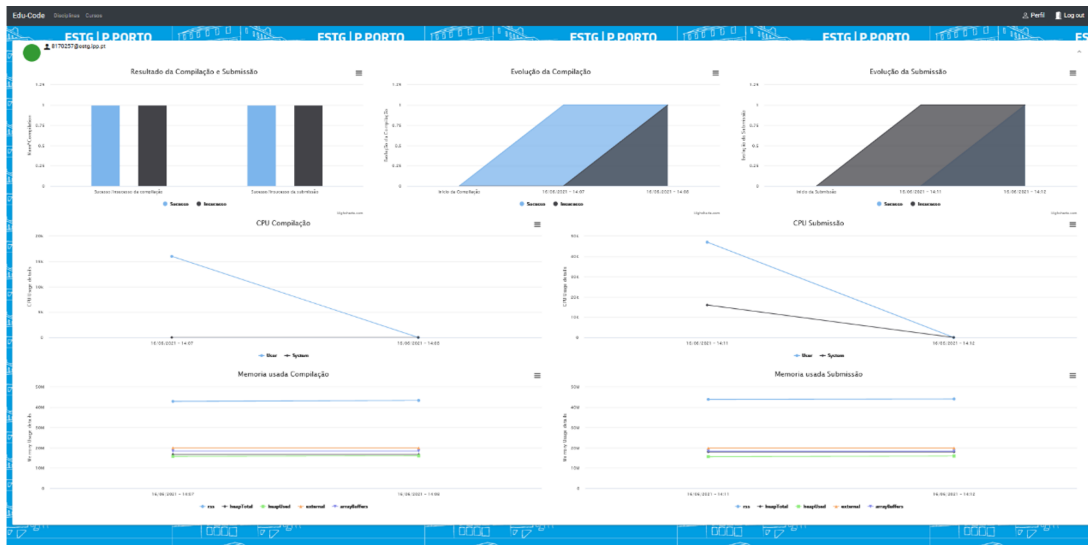


Figura 5.8: Visão Geral das Métricas

Para cada estudante que irá interagir com o exercício, e dependendo das suas ações, são apresentadas as métricas nos diferentes gráficos. Cada um dos gráficos tem o propósito de ajudar o professor a perceber se o estudante consegue resolver o exercício ou se está com dificuldade.

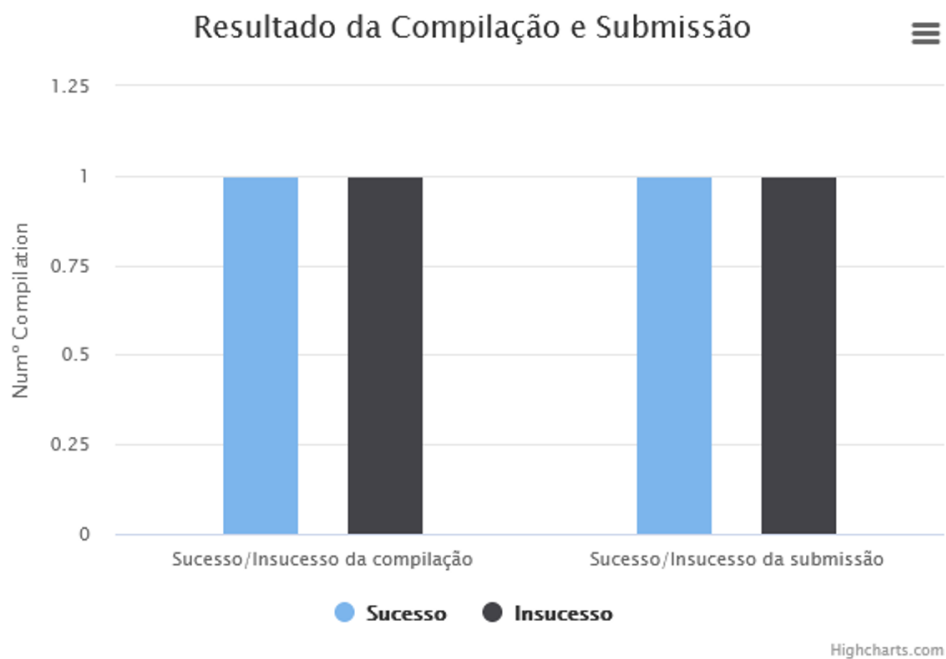


Figura 5.9: Gráfico dos Resultados da Compilação e Submissão

Na Figura 5.9 pode visualizar-se quantas vezes um determinado estudante compilou ou submeteu o exercício e, dessas tentativas, quantas foram bem sucedidas.

Com este gráfico é possível perceber se o estudante está a dar muita importância à compilação ou submissão. Se o estudante estiver só a compilar o seu exercício, independentemente de compilar com ou sem sucesso, ou se existir um elevado número de submissões com insucesso, poder-se-á assumir que o estudante não consegue resolver o exercício e necessita de uma explicação personalizada. Também é possível verificar se a resolução foi muito à base de tentativa e erro. Um indicador deste cenário será a existência de um número maior de insucessos tanto na compilação como na submissão, o que significa que o estudante não está a aplicar bem os conceitos ou estará a utilizar uma abordagem menos correta para resolver o exercício. A indicação de sucesso na submissão permite identificar os estudantes que conseguiram finalizar os exercícios propostos.

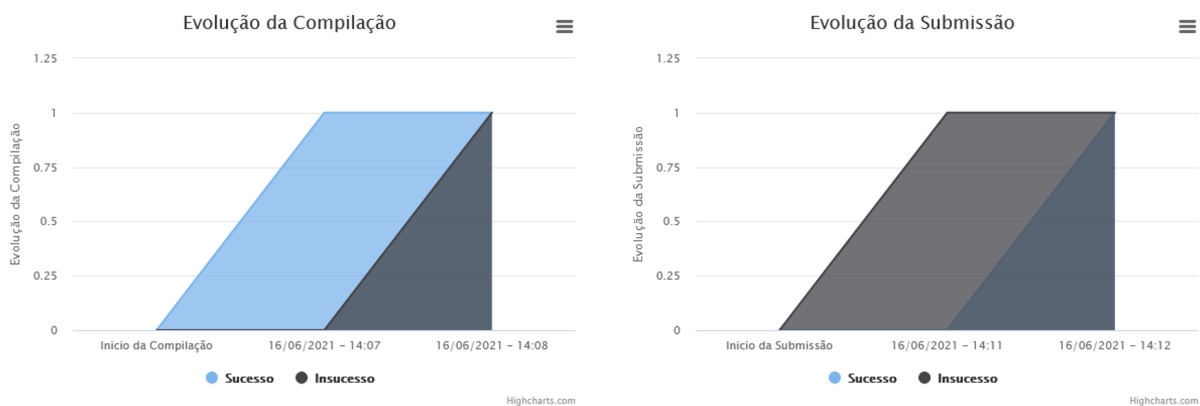


Figura 5.10: Gráfico da Evolução da Compilação e Submissão ao Longo do Tempo

Na Figura 5.10 é possível obter a informação da evolução da compilação e submissão ao longo do tempo. Caso se trate de uma submissão por tentativa e erro, verifica-se que o acumular de submissões inválidas vai aumentando até culminar numa submissão válida.

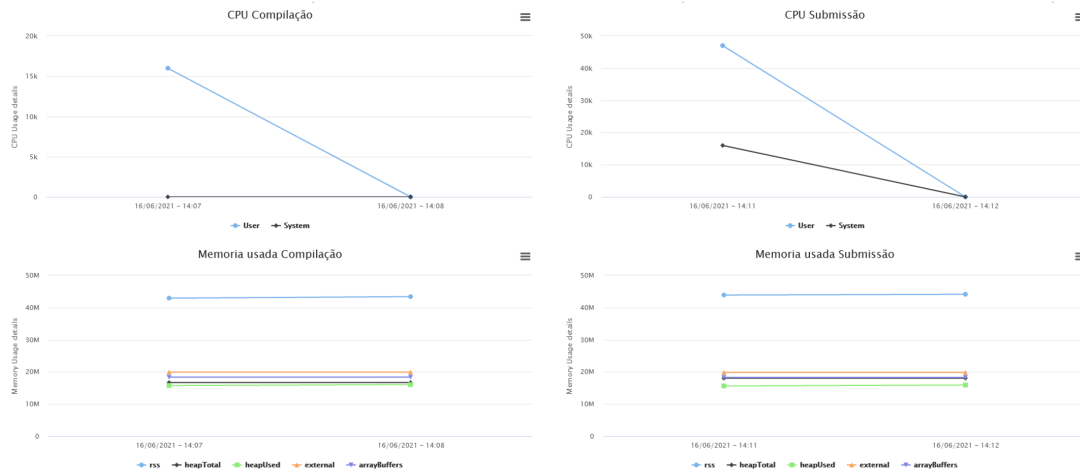


Figura 5.11: Gráfico da Utilização do CPU e Memória na Compilação e Submissão

Na Figura 5.11 apresentam-se os gráficos temporais relativos aos recursos utilizados pelo estudante, ou seja, necessidade de memória e de *CPU* durante todas as fases de compilação e submissão. Com recurso a estes gráficos é possível verificar se o estudante está a utilizar demasiados recursos desnecessários para a resolução do exercício. Um programador experiente deverá desenvolver código otimizado e, desta forma, utilizar o mínimo possível dos recursos computacionais disponíveis. Caso um problema que deveria ser de resolução simples necessite de bastante memória, então a forma de programação que o estudante está a realizar poderá não estar correta e, desta forma, ser necessário auxílio para demonstrar boas práticas de programação.

5.2 Aplicação de Métodos de Raciocínio

Para a classificação dos estudantes de acordo com a sua proficiência na linguagem de programação *Java* foram selecionados alguns modelos de aprendizagem supervisionada. Esta seleção deveu-se aos factos já mencionados na Secção 3.4. Assim, foram selecionados os seguintes modelos em conjunto com todos os dados mencionados no Capítulo 4:

1. Reg. Log.;
2. RNA;
3. *RF*;

4. SVM.

Para cada modelo selecionado existe um conjunto de parâmetros que é necessário selecionar e, eventualmente, ajustar de modo a melhorar a qualidade do modelo treinado. Desta forma obtiveram-se várias configurações que são apresentadas nas Tabelas 5.1-5.4.

Para o modelo de regressão logística obteve-se a configuração final apresentada na Tabela 5.1.

Tabela 5.1: Configurações Utilizados para a Regressão Logística

Nome do Modelo	Logística		
	penalty	C	random_state
Nº da Configuração	1	2	0

Os vários atributos da configuração foram:

1. **Penalty** (Regularização): O parâmetro '*penalty*' é um hiperparâmetro da regressão logística que controla o tipo de regularização aplicada ao modelo. Existem dois valores principais para '*penalty*':
 - (a) **L1**: Utiliza a regularização L1 (*Lasso*), que adiciona o valor absoluto dos coeficientes ao custo da função de perda. Isso promove a seleção de características, fazendo com que alguns coeficientes sejam exatamente zero.
 - (b) **L2**: Utiliza a regularização L2 (*Ridge*), que adiciona o quadrado dos coeficientes ao custo da função de perda. Isso impede que os coeficientes sejam muito grandes, evitando *overfitting*.
2. **C** (Inverso da Força de Regularização): O hiperparâmetro '*C*' é o inverso da força de regularização. Quanto menor o valor de '*C*', maior será a regularização aplicada ao modelo. Isso significa que um valor menor de '*C*' irá aumentar a penalização sobre os coeficientes do modelo, tornando-o mais resistente ao *overfitting*. Por outro lado, um valor maior de '*C*' permite que o modelo se ajuste mais aos dados de treino, mas pode ser mais suscetível ao *overfitting*.
3. **Random_state** (Estado Aleatório): O '*random_state*' não é um hiperparâmetro da regressão logística em si, mas é um parâmetro comum em muitas bibliotecas de

ML, como o *Scikit-Learn*. Este parâmetro permite a definição de uma semente (*seed*) para a geração de números aleatórios, o que torna os resultados reproduzíveis. Caso se defina um valor específico para `'random_state'`, serão sempre obtidos os mesmos resultados de cada vez que se executar o modelo.

Para o modelo de RNA foram definidas as configurações apresentadas na Tabela 5.2.

Tabela 5.2: Configurações Utilizados para a Rede Neuronal

Nome do Modelo	Redes Neurais			
	<code>hidden_layer_sizes</code>	<code>activation</code>	<code>max_iter</code>	<code>solver</code>
1	500	logistic	200	adam
2	500	logistic	200	lbfgs
3	500	tanh	200	sgd
4	500	tanh	200	adam
5	500	tanh	200	lbfgs
6	500	tanh	200	sgd

De um modo mais específico considera-se que:

1. **Hidden_layer_sizes** (Tamanhos das Camadas Ocultas): este parâmetro define a arquitetura da rede neuronal, especificando o número de neurónios em cada camada oculta. Pode passar um tuplo indicando o número de neurónios em cada camada. Por exemplo, o tuplo (5, 2) especifica uma rede neuronal com uma camada oculta de 5 neurónios e outra camada oculta de 2 neurónios. A escolha dos tamanhos das camadas ocultas é importante e afeta a capacidade de aprendizagem da rede.
2. **Activation** (Ativação): O parâmetro `'activation'` define a função de ativação a ser usada em cada camada da rede neuronal. Podem ser escolhidas várias funções de ativação, incluindo:
 - (a) **relu**: Função de ativação Rectified Linear Unit (ReLU);
 - (b) **logistic**: Função logística (sigmóide).
 - (c) **tanh**: Tangente hiperbólica;
 - (d) **identity**: Função de identidade (não aplica nenhuma transformação).
3. **Max_iter** (Número Máximo de Iterações): `'max_iter'` é o número máximo de

iterações (épocas) durante o treino da rede neuronal. Define quantas vezes a rede passará pelo conjunto de treino durante o processo de aprendizagem. O treino pode parar antes de atingir o número máximo de iterações se a convergência ocorrer antes disso.

4. **Solver** (Otimizador): O parâmetro *'solver'* especifica o algoritmo de otimização a ser usado para ajustar os pesos da rede durante o treino. Alguns dos valores possíveis incluem:
- (a) **lbfgs**: Algoritmo de otimização baseado em Quasi-Newton (apropriado para conjuntos de dados mais pequenos).
 - (b) **sgd**: Gradiente Descendente Estocástico.
 - (c) **adam**: Algoritmo de otimização baseado no método de momentos adaptativos.

Para o modelo de *Random Forest* foram definidas seis diferentes combinações de parâmetros como referido na Tabela 5.3.

Tabela 5.3: Configurações Utilizados para a *Random Forest*

Nome do Modelo	<i>Random Forest</i>	
Nº da Configuração	n_estimators	min_samples_leaf
1	200	1
2	200	2
3	300	1
4	300	2
5	500	1
6	500	2

Cada parâmetro indicado possui o seguinte significado:

1. **n_estimators** (Número de Estimadores): O parâmetro *'n_estimators'* especifica o número de árvores na floresta aleatória. Cada árvore na floresta é construída a partir de uma amostra aleatória do conjunto de dados de treino. Um valor maior para *'n_estimators'* geralmente resulta num modelo mais robusto, mas também aumenta o tempo de treino. É importante escolher um valor adequado para obter um equilíbrio entre desempenho e eficiência computacional.

2. **min_samples_leaf** (Número Mínimo de Amostras na Folha): este parâmetro define o número mínimo de amostras necessárias para criar uma folha (nó final) numa árvore de decisão na floresta. Valores maiores resultam em árvores mais profundas com menos divisões e são úteis para evitar o *overfitting*. A escolha do valor depende da complexidade do problema e do conjunto de dados.
3. **random_state** (Estado Aleatório): O 'random_state' é um valor que permite controlar a aleatoriedade no processo de construção da floresta aleatória. Definir um valor específico para 'random_state' garante que os resultados sejam reproduzíveis, o que é útil para fins de depuração e comparação de resultados entre diferentes execuções.
4. **n_jobs** (Número de Processadores Paralelos): O parâmetro 'n_jobs' especifica o número de processadores paralelos a serem usados para treinar as árvores na floresta. Isso pode acelerar significativamente o treino em máquinas com vários núcleos de *CPU*. Um valor de -1 indica que todos os núcleos disponíveis devem ser usados, enquanto um valor de 1 significa treino sequencial.

Para o último modelo de aprendizagem supervisionada, *Support Vector Machine*, foi realizada a configuração apresentada na Tabela 5.4.

Tabela 5.4: Configurações Utilizadas para a *Support Vector Machine*

Nome do Modelo	<i>Support Vector Machine</i>		
Nº da Configuração	kernel	C	random_state
1	linear	1	0

Cada parâmetro possui o seguinte significado:

1. **Kernel** (Núcleo): O parâmetro 'kernel' define o tipo de função de núcleo a ser usada no algoritmo de máquina de vetores de suporte. O kernel determina a forma como os dados são transformados para encontrar um hiperplano de separação. Alguns dos valores possíveis incluem:
 - (a) *linear*: Utiliza um hiperplano linear para separação.
 - (b) *rbf* (Radial Basis Function): Utiliza uma função de núcleo radial que é útil

para dados não linearmente separáveis.

(c) **poly**: Utiliza uma função de núcleo polinomial para separação não linear.

(d) **sigmoid**: Utiliza uma função de núcleo sigmoide para separação.

2. **C** (Parâmetro de Regularização): O hiperparâmetro 'C' controla o *trade-off* entre maximizar a margem e minimizar a classificação incorreta. Um valor maior de 'C' permite que o modelo se ajuste mais aos dados de treino, mas pode resultar em *overfitting*. Um valor menor de 'C' promove uma margem mais ampla, tornando o modelo mais robusto, mas pode levar a classificações incorretas.

3. **Random_state** (Estado Aleatório): O 'random_state' permite controlar a aleatoriedade no processo de treino do modelo. Definir um valor específico para 'random_state' garante a reprodutibilidade dos resultados.

Considerando todos os modelos selecionados e dado serem modelos de aprendizagem supervisionada, foi necessário fornecer um conjunto de dados para a realização do respectivo treino. No entanto, após a fase de treino é necessário garantir que o modelo não sofre de, por exemplo, *overfitting*, ou seja, o modelo fica demasiado ajustado aos dados e perde a capacidade de generalização, obtendo resultados mais fracos quando são apresentados novos dados. Por este motivo é habitual realizar uma divisão dos dados de modo a utilizar parte destes na fase de treino e a restante parte para teste de modo a validar a qualidade do modelo quando este é confrontado com novos dados. A divisão foi realizada da seguinte forma:

- Dados de treino: 70%
- Dados de teste: 30%

5.3 Discussão de Resultados

Para que seja possível existir uma distinção dos piores ou melhores resultados obtidos perante os modelos de *ML* utilizados e das suas configurações, existe um conjunto de

métricas para avaliar a qualidade dos modelos de aprendizagem supervisionada para classificação. As métricas utilizadas são as seguintes:

1. **Matriz de Confusão:** A matriz de confusão é uma tabela que resume o desempenho de um modelo de classificação. É essencial para perceber onde o modelo está a cometer erros e o tipo de erro que é mais comum. Esta inclui quatro elementos:
 - (a) **Verdadeiros Positivos (VP):** Número de exemplos corretamente classificados como positivos.
 - (b) **Verdadeiros Negativos (VN):** Número de exemplos corretamente classificados como negativos.
 - (c) **Falsos Positivos (FP):** Número de exemplos erroneamente classificados como positivos.
 - (d) **Falsos Negativos (FN):** Número de exemplos erroneamente classificados como negativos.
2. **Precisão:** A precisão é a proporção de verdadeiros positivos em relação ao número total de previsões positivas feitas pelo modelo. A fórmula de cálculo é apresentada na Equação 5.1.

$$\text{Precisão} = \frac{VP}{VP + FP} \quad (5.1)$$

Esta avalia a capacidade do modelo em fazer previsões positivas corretas. É importante quando se quer minimizar o número de falsos positivos ou quando se valoriza previsões precisas.

3. **Recall (Sensibilidade):** O *recall* é a proporção de verdadeiros positivos em relação ao número total de exemplos positivos na amostra (Equação 5.2).

$$\text{Recall} = \frac{VP}{VP + FN} \quad (5.2)$$

Esta avalia a capacidade do modelo em identificar corretamente todos os exemplos positivos. É importante quando os falsos negativos têm um alto custo, como em diagnósticos médicos.

4. **F1-Score:** O F1-Score é uma métrica que combina precisão e recall num único número. É a média harmônica dessas duas métricas, dando mais peso à menor delas. É útil quando se procura um equilíbrio entre precisão e recall. Ajuda a encontrar um compromisso entre minimizar falsos positivos e falsos negativos.
5. **Accuracy de Treino e Teste:** A *accuracy* é a proporção de previsões corretas em relação ao número total de exemplos no conjunto de dados. É uma métrica geral que fornece uma visão global do desempenho do modelo. A *accuracy* de treino mede o quão bem o modelo se ajusta aos dados de treino, enquanto a *accuracy* de teste avalia o desempenho do modelo em dados não vistos. Uma diferença significativa entre a *accuracy* de treino e teste pode indicar que o modelo está a ajustar-se demasiado aos dados de treino (*overfitting*).

De acordo com os dados apresentados da Tabela A.1-A.54 com a informação detalhada de cada uma das métricas mencionados acima, no geral, poder-se-á concluir que os modelos não apresentam os melhores resultados esperados, uma vez que, muito destes apresentam uma *accuracy* de 100% nos dados de treino, demonstrando estar em *overfitting*. Outro exemplo, com a mesma métrica, é existir uma diferença significativa entre os dados de treino e de teste, na qual os dados de teste não ultrapassam 70% de *accuracy*. Outra análise que se pode realizar é que os modelos possuem maior dificuldade em prever o caso onde o estudante se insere numa fase mais inicial de programação. Esta situação, embora em menor escala, reflete-se também para estudantes considerados como nível médio de programação. Mesmo para *datasets* de maiores dimensões como, por exemplo, o último *dataset* que possui uma simulação de 500 estudantes, os resultados são melhores, mas apresentam alguma tendência para *overfitting*.

Os melhores resultados obtidos são apresentados na Tabela 5.5 e Tabela 5.6, para a configuração 3 com Redes Neurais e *dataset* 2 e para a configuração 2 com *Random*

Forest e *dataset 4*, respectivamente. No caso do modelo de rede neuronal, o modelo apresenta uma *accuracy* moderada no conjunto de teste, com uma taxa de acerto de 64,79%. O F1-score variou entre as classes, sendo a classe 3 (Estudante Experiente) a melhor classificada.

Para o outro caso de treino, configuração 2 com *Random Forest* e *dataset 4*, o modelo teve um desempenho quase perfeito no treino, mas obteve uma taxa de acerto significativamente menor no teste, o que pode indicar a existência de *overfitting*. A *accuracy* no teste é de apenas de 56,23% e o F1-score apresenta a mesma situação que o modelo anterior.

Tabela 5.5: Resultados de Treino e Teste - Redes Neurais - Configuração 3 - Dataset2

Modelo: Redes Neurais**Nº Configuração:** 3**Nome do Dataset:** Dataset2**Train Accuracy:** 62,28%**Train Confusion Matrix:**

3	5	30
2	33	48
1	20	139

Classification Report:

	precision	recall	f1-score	support
1	0.50	0.08	0.14	38
2	0.57	0.40	0.47	83
3	0.64	0.87	0.74	160

accuracy			0.62	281
macro avg	0.57	0.45	0.45	281
weighted avg	0.60	0.62	0.58	281

Test Accuracy: 64,79%**Test Confusion Matrix:**

1	0	6
1	3	10
0	8	42

Classification Report:

	precision	recall	f1-score	support
1	0.50	0.14	0.22	7
2	0.27	0.21	0.24	14
3	0.72	0.84	0.78	50

accuracy			0.65	71
macro avg	0.50	0.40	0.41	71
weighted avg	0.61	0.65	0.62	71

Tabela 5.6: Resultados de Treino e Teste - *Random Forest* - Configuração 2 - Dataset4**Modelo:** *Random Forest***Nº Configuração:** 2**Nome do Dataset:** Dataset4**Train Accuracy:** 99,24%**Train Confusion Matrix:**

243	1	8
0	440	2
0	0	753

Classification Report:

	precision	recall	f1-score	support
1	1.00	0.96	0.98	243
2	1.00	1.00	1.00	442
3	0.99	1.00	0.99	753

accuracy			0.99	1438
macro avg	0.99	0.99	0.99	1438
weighted avg	0.99	0.99	0.99	1438

Test Accuracy: 56,23%**Test Confusion Matrix:**

2	6	97
0	38	139
0	28	307

Classification Report:

	precision	recall	f1-score	support
1	1.00	0.02	0.04	105
2	0.53	0.21	0.31	177
3	0.57	0.92	0.70	335

accuracy			0.56	617
macro avg	0.70	0.38	0.35	617
weighted avg	0.63	0.56	0.47	617

Capítulo 6

Conclusões e Trabalho Futuro

A melhoria realizada com a integração e uso do SonarQube para o intuito do projeto ajudou o desenvolvimento do mesmo através do reforço das métricas existentes, permitindo também obter uma maior percepção da qualidade do código desenvolvido pelo estudante ao interagir com os exercícios disponibilizado pelo professor.

Tendo em consideração os objetivos estabelecidos para o desenvolvimento do projeto, foi apresentada uma solução que, com recurso a algoritmos de *ML*, permite indicar e classificar o desempenho dos estudantes de acordo com o seu nível de proficiência na linguagem de programação Java. Esta avaliação automática é também um auxílio ao Professor e proporciona um ambiente onde se estimula a autonomia do estudante, uma vez que o Professor tem a possibilidade de verificar, de forma remota, a aquisição de conhecimento por parte dos estudantes e verificar, em tempo real, os respetivos resultados/dificuldades. Deste modo, e de acordo com os vários conceitos apresentados no Capítulo 2 pretende-se não só estimular a autonomia do estudante como a sua vontade em aprender. A classificação automática com recurso aos algoritmos de *ML* permitiu obter uma maior percepção da aprendizagem do estudante.

Considera-se, por este motivo, que todos os objetivos inicialmente propostos foram atingidos com sucesso demonstrados não só pelo desenvolvimento da plataforma *web*, como por todas as funcionalidades implementadas e, por último, pelos algoritmos de classificação

aplicados.

No entanto, as abordagens carecem ainda de maior exploração futura de modo a se tentar melhorar os resultados (*performance*) obtida. Considerando esta lacuna, como trabalho futuro, com o intuito de melhorar a *performance* dos modelos de *ML*, pretende-se a criação de *clusters* para realizar uma pré-segmentação dos estudantes e, sobre cada grupo gerado aplicar uma rede neuronal. Desta forma, julga-se que se conseguirá obter melhores resultados dado que os estudantes já foram previamente selecionados para pertencer a um determinado grupo e a rede neuronal torna-se mais específica para esses elementos.

Outro desenvolvimento futuro é a disponibilização da ferramenta criada num cenário real de sala de aula para realizar o tratamento com dados reais. Os dados fabricados permitiram testar e validar a ferramenta num caso de estudo simulado, mas torna-se de máxima importância realizar testes em cenários reais.

Bibliografia

- [1] R. J. Vallerand, L. G. Pelletier, M. R. Blais, N. M. Briere, C. Senecal, and E. F. Valieres, “The academic motivation scale: A measure of intrinsic, extrinsic, and amotivation in education,” *Educational and Psychological Measurement*, vol. 52, no. 4, 1992.
- [2] J. R. Albrecht and S. A. Karabenick, “Relevance for Learning and Motivation in Education,” 2018.
- [3] L. Guo, J. Huang, and Y. Zhang, “Education development in China: Education return, quality, and equity,” 2019.
- [4] D. S. Kamalam, ““Transforming Our World: The 2030 Agenda for Sustainable Development”,” *Pondicherry Journal of Nursing*, vol. 11, no. 2, 2017.
- [5] M. Murtonen, H. Gruber, and E. Lehtinen, “The return of behaviourist epistemology: A review of learning outcomes studies,” *Educational Research Review*, vol. 22, 2017.
- [6] M. Asoodar, M. R. Atai, and L. Baten, “Successful Erasmus Experience: Analysing Perceptions before, during and after Erasmus,” *Journal of Research in International Education*, vol. 16, no. 1, 2017.
- [7] M. Hashizume, T. Kono, and S. Shiota, “Developing and evaluating teaching material for information technology moral education for high school students with special needs,” *International Journal of Information and Education Technology*, vol. 10, no. 4, 2020.

- [8] K. Jitsuiki, H. Ohsaka, J. Shitara, M. Ishibashi, M. Suzuki, Y. Nozawa, and Y. Yanagawa, “Reaction of juntendo shizuoka hospital at izu peninsula to typhoon hagibis (2019) and an analysis of twitter concerning izunokuni city,” *Disaster Medicine and Public Health Preparedness*, vol. 16, no. 1, pp. 147–152, 2022.
- [9] S. S. Misaki Hashizume, Takeshi Kono, “Development and Evaluation of a Information Ethics Class for Promoting Awareness of Behavior which Leads to Problem,” *Japan Journal of Educational Technology*, vol. 39, no. Suppl, 2016.
- [10] Jetbrains, “Edu tools - Plugin,” 2021.
- [11] V. Kepuska and G. Bohouta, “Next-generation of virtual personal assistants (microsoft cortana, apple siri, amazon alexa and google home),” in *2018 IEEE 8th annual computing and communication workshop and conference (CCWC)*, pp. 99–103, IEEE, 2018.
- [12] R. Yus, *Educação integral: uma educação holística para o século XXI*. Artmed, 2002.
- [13] M. C. Nussbaum, *Not for profit: Why democracy needs the humanities*. Princeton university press, 2016.
- [14] G. Childs, “Steiner education in theory and practice,” 1991.
- [15] M. S. P. Costa, “Maria montessori e seu método,” *Linhas críticas*, vol. 7, no. 13, pp. 305–320, 2001.
- [16] F. Capra, *O ponto de mutação*. editora Cultrix, 2012.
- [17] K. Wilber, “Introduction to integral theory and practice,” *AQAL: Journal of Integral Theory and Practice*, vol. 1, no. 1, pp. 2–38, 2005.
- [18] P. Langley, A. Newell, and H. A. Simon, “Symbols and search in humans and machines,” 1976.
- [19] A. L. Samuel, “Some studies in machine learning using the game of checkers,” *IBM Journal of research and development*, vol. 44, no. 1.2, pp. 206–226, 2000.

- [20] F.-h. Hsu, "Ibm's deep blue chess grandmaster chips," *IEEE micro*, vol. 19, no. 2, pp. 70–81, 1999.
- [21] P. Cunningham, M. Cord, and S. J. Delany, "Supervised learning," in *Machine learning techniques for multimedia: case studies on organization and retrieval*, pp. 21–49, Springer, 2008.
- [22] D. M. Jose, A. M. Vincent, and G. S. Dwarakish, "Improving multiple model ensemble predictions of daily precipitation and temperature through machine learning techniques," *Scientific Reports*, vol. 12, no. 1, p. 4678, 2022.
- [23] H. K. Andi, "An accurate bitcoin price prediction using logistic regression with lstm machine learning model," *Journal of Soft Computing Paradigm*, vol. 3, no. 3, pp. 205–217, 2021.
- [24] S. Nusinovici, Y. C. Tham, M. Y. C. Yan, D. S. W. Ting, J. Li, C. Sabanayagam, T. Y. Wong, and C.-Y. Cheng, "Logistic regression was as good as machine learning for predicting major chronic diseases," *Journal of clinical epidemiology*, vol. 122, pp. 56–69, 2020.
- [25] H. Nankani, S. Gupta, S. Singh, and S. S. Ramesh, "Detection analysis of various types of cancer by logistic regression using machine learning," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 9, no. 1, pp. 99–104, 2019.
- [26] M. Alghamdi, M. Al-Mallah, S. Keteyian, C. Brawner, J. Ehrman, and S. Sakr, "Predicting diabetes mellitus using smote and ensemble machine learning approach: The henry ford exercise testing (fit) project," *PloS one*, vol. 12, no. 7, p. e0179805, 2017.
- [27] S. Ravichandiran, *Getting Started with Google BERT: Build and train state-of-the-art natural language processing models using BERT*. Packt Publishing Ltd, 2021.
- [28] O. Ghorbanzadeh, T. Blaschke, K. Gholamnia, S. R. Meena, D. Tiede, and J. Aryal, "Evaluation of different machine learning methods and deep-learning convolutional neural networks for landslide detection," *Remote Sensing*, vol. 11, no. 2, p. 196, 2019.

- [29] D. W. Jorgenson, M. L. Weitzman, Y. X. ZXhang, Y. M. Haxo, and Y. X. Mat, “Can neural networks predict stock market?,” *AC Investment Research Journal*, vol. 220, no. 44, 2023.
- [30] Y. Zhou, W. Su, L. Ding, H. Luo, and P. E. Love, “Predicting safety risks in deep foundation pits in subway infrastructure projects: support vector machine approach,” *Journal of Computing in Civil Engineering*, vol. 31, no. 5, p. 04017052, 2017.
- [31] N. I. El-Sawalhi, “Support vector machine cost estimation model for road projects,” *Journal of Civil Engineering and Architecture*, vol. 9, no. 9, pp. 846–859, 2015.
- [32] K. K. Hiran, R. K. Jain, K. Lakhwani, and R. Doshi, *Machine Learning: Master Supervised and Unsupervised Learning Algorithms with Real Examples (English Edition)*. BPB Publications, 2021.
- [33] N. Li, M. Shepperd, and Y. Guo, “A systematic review of unsupervised learning techniques for software defect prediction,” *Information and Software Technology*, vol. 122, p. 106287, 2020.
- [34] A. E. Sallab, M. Abdou, E. Perot, and S. Yogamani, “Deep reinforcement learning framework for autonomous driving,” *arXiv preprint arXiv:1704.02532*, 2017.
- [35] J. Ramos, P. Cunha, and F. Silva, “A Tool to Assess Students’ Performance in Programming,” *Lecture Notes in Networks and Systems*, vol. 326, pp. 166–174, 2022.

Apêndice A

Dados de treino dos modelos de ML

O presente capítulo apresenta os vários resultados de treino e de teste de acordo com as configurações apresentadas na Secção 5.2. Na legenda de cada tabela faz-se referência ao modelo, configuração e *dataset* utilizado.

Tabela A.1: Resultados de Treino e Teste - Logística - Configuração 1 - Dataset1

Modelo: Logística**Nº Configuração:** 1**Nome do Dataset:** Dataset1**Train Accuracy:** 65,79%**Train Confusion Matrix:**

8	3	6
1	27	27
1	14	65

Classification Report:

	precision	recall	f1-score	support
1	0.80	0.47	0.59	17
2	0.61	0.49	0.55	55
3	0.66	0.81	0.73	80

accuracy			0.66	152
macro avg	0.69	0.59	0.62	152
weighted avg	0.66	0.66	0.65	152

Test Accuracy: 53,03%**Test Confusion Matrix:**

0	1	7
2	5	5
4	12	30

Classification Report:

	precision	recall	f1-score	support
1	0	0	0	8
2	0,28	0,42	0,33	12
3	0,71	0,65	0,68	46

accuracy			0.53	66
macro avg	0.33	0.36	0.34	66
weighted avg	0.55	0.53	0.54	66

Tabela A.2: Resultados de Treino e Teste - Logística - Configuração 1 - Dataset2

Modelo: Logística**Nº Configuração:** 1**Nome do Dataset:** Dataset2**Train Accuracy:** 69,11%**Train Confusion Matrix:**

3	2	24
1	28	33
1	15	139

Classification Report:

	precision	recall	f1-score	support
1	0.60	0.10	0.18	29
2	0.62	0.45	0.52	62
3	0.71	0.90	0.79	155

accuracy			0.69	246
macro avg	0.64	0.48	0.50	246
weighted avg	0.67	0.69	0.65	246

Test Accuracy: 46,23%**Test Confusion Matrix:**

0	1	15
1	2	32
1	7	47

Classification Report:

	precision	recall	f1-score	support
1	0	0	0	16
2	0.20	0.06	0.09	35
3	0.50	0.85	0.63	55

accuracy			0.46	106
macro avg	0.23	0.30	0.24	106
weighted avg	0.33	0.46	0.36	106

Tabela A.3: Resultados de Treino e Teste - Logística - Configuração 1 - Dataset3

Modelo: Logística**Nº Configuração:** 1**Nome do Dataset:** Dataset3**Train Accuracy:** 50,09%**Train Confusion Matrix:**

2	50	34
1	122	97
0	91	150

Classification Report:

	precision	recall	f1-score	support
1	0.67	0.02	0.04	86
2	0.46	0.55	0.51	220
3	0.53	0.62	0.57	241

accuracy			0.50	547
macro avg	0.55	0.40	0.37	547
weighted avg	0.53	0.50	0.46	547

Test Accuracy: 42,13%**Test Confusion Matrix:**

0	17	23
0	50	49
0	47	49

Classification Report:

	precision	recall	f1-score	support
1	0	0	0	40
2	0.44	0.51	0.47	99
3	0.40	0.51	0.45	96

accuracy			0.42	235
macro avg	0.28	0.34	0.31	235
weighted avg	0.35	0.42	0.38	235

Tabela A.4: Resultados de Treino e Teste - Logística - Configuração 1 - Dataset4

Modelo: Logística**Nº Configuração:** 1**Nome do Dataset:** Dataset4**Train Accuracy:** 52,65%**Train Confusion Matrix:**

0	7	236
0	22	420
0	18	735

Classification Report:

	precision	recall	f1-score	support
1	0.00	0.00	0.00	243
2	0.47	0.05	0.09	442
3	0.53	0.98	0.69	753

accuracy			0.53	1438
macro avg	0.33	0.34	0.26	1438
weighted avg	0.42	0.53	0.39	1438

Test Accuracy: 53,32%**Test Confusion Matrix:**

0	5	100
0	8	169
0	14	321

Classification Report:

	precision	recall	f1-score	support
1	0	0	0	105
2	0.30	0.05	0.08	177
3	0.54	0.96	0.69	335

accuracy			0.53	617
macro avg	0.28	0.33	0.26	617
weighted avg	0.38	0.53	0.40	617

Tabela A.5: Resultados de Treino e Teste - Redes Neurais - Configuração 1 - Dataset1

Modelo: Redes Neurais
Nº Configuração: 1
Nome do Dataset: Dataset1
Train Accuracy: 66.66%
Train Confusion Matrix:

8	3	10
2	23	30
2	11	85

Classification Report:

	precision	recall	f1-score	support
1	0.67	0.38	0.48	21
2	0.46	0.55	0.51	55
3	0.53	0.62	0.57	98

accuracy			0.67	174
macro avg	0.66	0.56	0.58	174
weighted avg	0.66	0.67	0.65	174

Test Accuracy: 52,27%

Test Confusion Matrix:

0	1	3
0	3	9
4	4	20

Classification Report:

	precision	recall	f1-score	support
1	0	0	0	4
2	0.38	0.25	0.30	12
3	0.62	0.71	0.67	28

accuracy			0.52	44
macro avg	0.33	0.32	0.32	44
weighted avg	0.50	0.52	0.51	44

Tabela A.6: Resultados de Treino e Teste - Redes Neurais - Configuração 2 - Dataset1

Modelo: Redes Neurais**Nº Configuração:** 2**Nome do Dataset:** Dataset1**Train Accuracy:** 100%**Train Confusion Matrix:**

21	0	0
0	55	0
0	0	98

Classification Report:

	precision	recall	f1-score	support
1	1.00	1.00	1.00	21
2	1.00	1.00	1.00	55
3	1.00	1.00	1.00	98

accuracy			1.00	174
macro avg	1.00	1.00	1.00	174
weighted avg	1.00	1.00	1.00	174

Test Accuracy: 52,27%**Test Confusion Matrix:**

0	1	3
0	5	7
2	8	18

Classification Report:

	precision	recall	f1-score	support
1	0	0	0	4
2	0.36	0.42	0.38	12
3	0.64	0.64	0.64	28

accuracy			0.52	44
macro avg	0.33	0.35	0.34	44
weighted avg	0.51	0.52	0.51	44

Tabela A.7: Resultados de Treino e Teste - Redes Neurais - Configuração 3 - Dataset1

Modelo: Redes Neurais
Nº Configuração: 3
Nome do Dataset: Dataset1
Train Accuracy: 61.49%
Train Confusion Matrix:

3	4	14
0	17	38
0	11	87

Classification Report:

	precision	recall	f1-score	support
1	1.00	0.14	0.25	21
2	0.53	0.31	0.39	55
3	0.63	0.89	0.73	98

accuracy			0.61	174
macro avg	0.72	0.45	0.46	174
weighted avg	0.64	0.61	0.57	174

Test Accuracy: 63,64%

Test Confusion Matrix:

0	0	4
0	4	8
1	3	24

Classification Report:

	precision	recall	f1-score	support
1	0	0	0	4
2	0.57	0.33	0.42	12
3	0.67	0.86	0.75	28

accuracy			0.64	44
macro avg	0.41	0.40	0.39	44
weighted avg	0.58	0.64	0.59	44

Tabela A.8: Resultados de Treino e Teste - Redes Neurais - Configuração 4 - Dataset1

Modelo: Redes Neurais**Nº Configuração:** 4**Nome do Dataset:** Dataset1**Train Accuracy:** 94,83%**Train Confusion Matrix:**

17	0	4
0	51	4
0	1	97

Classification Report:

	precision	recall	f1-score	support
1	1.00	0.81	0.89	21
2	0.98	0.93	0.95	55
3	0.92	0.99	0.96	98

accuracy			0.95	174
macro avg	0.97	0.91	0.93	174
weighted avg	0.95	0.95	0.95	174

Test Accuracy: 47,73%**Test Confusion Matrix:**

0	1	3
0	5	7
3	9	16

Classification Report:

	precision	recall	f1-score	support
1	0	0	0	4
2	0.33	0.42	0.37	12
3	0.62	0.57	0.59	28

accuracy			0.48	44
macro avg	0.32	0.33	0.32	44
weighted avg	0.48	0.48	0.48	44

Tabela A.9: Resultados de Treino e Teste - Redes Neurais - Configuração 5 - Dataset1

Modelo: Redes Neurais
Nº Configuração: 5
Nome do Dataset: Dataset1
Train Accuracy: 100%
Train Confusion Matrix:

21	0	0
0	55	0
0	0	98

Classification Report:

	precision	recall	f1-score	support
1	1.00	1.00	1.00	21
2	1.00	1.00	1.00	55
3	1.00	1.00	1.00	98

accuracy			1.00	174
macro avg	1.00	1.00	1.00	174
weighted avg	1.00	1.00	1.00	174

Test Accuracy: 52,27%

Test Confusion Matrix:

0	1	3
1	6	5
2	9	17

Classification Report:

	precision	recall	f1-score	support
1	0	0	0	4
2	0.38	0.50	0.43	12
3	0.68	0.61	0.64	28

accuracy			0.52	44
macro avg	0.35	0.37	0.36	44
weighted avg	0.54	0.52	0.53	44

Tabela A.10: Resultados de Treino e Teste - Redes Neurais - Configuração 6 - Dataset1

Modelo: Redes Neurais**Nº Configuração:** 6**Nome do Dataset:** Dataset1**Train Accuracy:** 63,21%**Train Confusion Matrix:**

5	4	12
0	18	37
1	10	87

Classification Report:

	precision	recall	f1-score	support
1	0.83	0.24	0.37	21
2	0.56	0.33	0.41	55
3	0.64	0.89	0.74	98

accuracy			0.63	174
macro avg	0.68	0.48	0.51	174
weighted avg	0.64	0.63	0.59	174

Test Accuracy: 61,36%**Test Confusion Matrix:**

0	0	4
0	4	8
1	4	16

Classification Report:

	precision	recall	f1-score	support
1	0	0	0	4
2	0.50	0.33	0.40	12
3	0.66	0.82	0.73	28

accuracy			0.61	44
macro avg	0.39	0.38	0.38	44
weighted avg	0.55	0.61	0.57	44

Tabela A.11: Resultados de Treino e Teste - Redes Neurais - Configuração 1 - Dataset2

Modelo: Redes Neurais**Nº Configuração:** 1**Nome do Dataset:** Dataset2**Train Accuracy:** 61,92%**Train Confusion Matrix:**

8	4	26
2	35	46
2	27	131

Classification Report:

	precision	recall	f1-score	support
1	0.67	0.21	0.32	38
2	0.53	0.42	0.47	83
3	0.65	0.82	0.72	160

accuracy			0.62	281
macro avg	0.61	0.48	0.50	281
weighted avg	0.61	0.62	0.59	281

Test Accuracy: 64,78%**Test Confusion Matrix:**

2	0	5
1	3	10
1	8	41

Classification Report:

	precision	recall	f1-score	support
1	0.50	0.29	0.36	7
2	0.27	0.21	0.24	14
3	0.68	0.61	0.64	50

accuracy			0.65	71
macro avg	0.50	0.44	0.46	71
weighted avg	0.62	0.65	0.63	71

Tabela A.12: Resultados de Treino e Teste - Redes Neurais - Configuração 2 - Dataset2

Modelo: Redes Neurais**Nº Configuração:** 2**Nome do Dataset:** Dataset2**Train Accuracy:** 100%**Train Confusion Matrix:**

38	0	0
0	83	0
0	0	160

Classification Report:

	precision	recall	f1-score	support
1	1.00	1.00	1.00	38
2	1.00	1.00	1.00	83
3	1.00	1.00	1.00	160

accuracy			1.00	281
macro avg	1.00	1.00	1.00	281
weighted avg	1.00	1.00	1.00	281

Test Accuracy: 46,48%**Test Confusion Matrix:**

3	3	1
0	4	10
9	15	26

Classification Report:

	precision	recall	f1-score	support
1	0.25	0.43	0.32	7
2	0.27	0.29	0.22	14
3	0.70	0.52	0.60	50

accuracy			0.46	71
macro avg	0.38	0.41	0.38	71
weighted avg	0.56	0.46	0.50	71

Tabela A.13: Resultados de Treino e Teste - Redes Neurais - Configuração 4 - Dataset2

Modelo: Redes Neurais**Nº Configuração:** 4**Nome do Dataset:** Dataset2**Train Accuracy:** 80,78%**Train Confusion Matrix:**

22	0	16
1	51	31
0	6	154

Classification Report:

	precision	recall	f1-score	support
1	0.96	0.58	0.72	38
2	0.89	0.61	0.73	83
3	0.77	0.96	0.85	160

accuracy			0.81	281
macro avg	0.87	0.72	0.77	281
weighted avg	0.83	0.81	0.80	281

Test Accuracy: 63,38%**Test Confusion Matrix:**

2	1	4
2	2	10
4	5	41

Classification Report:

	precision	recall	f1-score	support
1	0.25	0.29	0.27	7
2	0.25	0.14	0.18	14
3	0.75	0.82	0.78	50

accuracy			0.63	71
macro avg	0.42	0.42	0.41	71
weighted avg	0.60	0.63	0.61	71

Tabela A.14: Resultados de Treino e Teste - Redes Neurais - Configuração 5 - Dataset2

Modelo: Redes Neurais**Nº Configuração:** 5**Nome do Dataset:** Dataset2**Train Accuracy:** 100%**Train Confusion Matrix:**

38	0	0
0	83	0
0	0	160

Classification Report:

	precision	recall	f1-score	support
1	1.00	1.00	1.00	38
2	1.00	1.00	1.00	83
3	1.00	1.00	1.00	160

accuracy			1.00	281
macro avg	1.00	1.00	1.00	281
weighted avg	1.00	1.00	1.00	281

Test Accuracy: 47,89%**Test Confusion Matrix:**

3	2	2
3	6	5
12	13	25

Classification Report:

	precision	recall	f1-score	support
1	0.17	0.43	0.24	7
2	0.29	0.43	0.34	14
3	0.78	0.50	0.61	50

accuracy			0.48	71
macro avg	0.41	0.45	0.40	71
weighted avg	0.62	0.48	0.52	71

Tabela A.15: Resultados de Treino e Teste - Redes Neurais - Configuração 6 - Dataset2

Modelo: Redes Neurais**Nº Configuração:** 6**Nome do Dataset:** Dataset2**Train Accuracy:** 60,85%**Train Confusion Matrix:**

3	4	31
1	30	52
1	21	138

Classification Report:

	precision	recall	f1-score	support
1	0.60	0.08	0.14	38
2	0.55	0.36	0.43	83
3	0.62	0.86	0.72	160

accuracy			0.61	281
macro avg	0.59	0.43	0.43	281
weighted avg	0.60	0.61	0.56	281

Test Accuracy: 64,79%**Test Confusion Matrix:**

1	0	6
1	3	10
0	8	42

Classification Report:

	precision	recall	f1-score	support
1	0.50	0.14	0.22	7
2	0.27	0.21	0.24	14
3	0.72	0.84	0.78	50

accuracy			0.65	71
macro avg	0.50	0.40	0.41	71
weighted avg	0.61	0.65	0.62	71

Tabela A.16: Resultados de Treino e Teste - Redes Neurais - Configuração 1 - Dataset3

Modelo: Redes Neurais**Nº Configuração:** 1**Nome do Dataset:** Dataset3**Train Accuracy:** 49,12%**Train Confusion Matrix:**

0	46	61
0	101	153
0	58	206

Classification Report:

	precision	recall	f1-score	support
1	0.00	0.00	0.00	107
2	0.49	0.36	0.44	254
3	0.62	0.78	0.60	264

accuracy			0.49	625
macro avg	0.33	0.39	0.35	625
weighted avg	0.41	0.49	0.43	625

Test Accuracy: 49,04%**Test Confusion Matrix:**

0	7	12
0	20	45
0	16	57

Classification Report:

	precision	recall	f1-score	support
1	0.00	0.00	0.00	19
2	0.47	0.31	0.37	65
3	0.50	0.78	0.61	73

accuracy			0.49	157
macro avg	0.32	0.40	0.33	157
weighted avg	0.43	0.49	0.44	157

Tabela A.17: Resultados de Treino e Teste - Redes Neurais - Configuração 2 - Dataset3

Modelo: Redes Neurais**Nº Configuração:** 2**Nome do Dataset:** Dataset3**Train Accuracy:** 47,52%**Train Confusion Matrix:**

0	65	42
0	147	107
1	113	150

Classification Report:

	precision	recall	f1-score	support
1	0.00	0.00	0.00	107
2	0.45	0.58	0.51	254
3	0.50	0.57	0.53	264

accuracy			0.48	625
macro avg	0.32	0.38	0.35	625
weighted avg	0.40	0.48	0.43	625

Test Accuracy: 51,59%**Test Confusion Matrix:**

0	12	7
0	36	29
0	28	45

Classification Report:

	precision	recall	f1-score	support
1	0.00	0.00	0.00	19
2	0.47	0.55	0.51	65
3	0.56	0.62	0.58	73

accuracy			0.52	157
macro avg	0.34	0.39	0.37	157
weighted avg	0.45	0.52	0.48	157

Tabela A.18: Resultados de Treino e Teste - Redes Neurais - Configuração 3 - Dataset3

Modelo: Redes Neurais**Nº Configuração:** 3**Nome do Dataset:** Dataset3**Train Accuracy:** 46,72%**Train Confusion Matrix:**

0	54	53
0	136	118
1	107	156

Classification Report:

	precision	recall	f1-score	support
1	0.00	0.00	0.00	107
2	0.46	0.54	0.49	254
3	0.48	0.59	0.53	264

accuracy			0.47	625
macro avg	0.31	0.38	0.34	625
weighted avg	0.39	0.47	0.42	625

Test Accuracy: 50,32%**Test Confusion Matrix:**

0	8	11
0	30	35
0	24	49

Classification Report:

	precision	recall	f1-score	support
1	0.00	0.00	0.00	19
2	0.48	0.46	0.47	65
3	0.78	0.67	0.58	73

accuracy			0.50	157
macro avg	0.33	0.38	0.35	157
weighted avg	0.44	0.50	0.47	157

Tabela A.19: Resultados de Treino e Teste - Redes Neurais - Configuração 4 - Dataset3

Modelo: Redes Neurais**Nº Configuração:** 4**Nome do Dataset:** Dataset3**Train Accuracy:** 70,72%**Train Confusion Matrix:**

41	33	33
0	178	76
0	41	223

Classification Report:

	precision	recall	f1-score	support
1	1.00	0.38	0.55	107
2	0.71	0.70	0.70	254
3	0.67	0.84	0.75	264

accuracy			0.71	625
macro avg	0.79	0.64	0.67	625
weighted avg	0.74	0.71	0.70	625

Test Accuracy: 45,86%**Test Confusion Matrix:**

0	9	10
0	28	37
0	27	44

Classification Report:

	precision	recall	f1-score	support
1	0.00	0.00	0.00	19
2	0.44	0.43	0.43	65
3	0.48	0.60	0.54	73

accuracy			0.46	157
macro avg	0.31	0.34	0.32	157
weighted avg	0.41	0.46	0.43	157

Tabela A.20: Resultados de Treino e Teste - Redes Neurais - Configuração 5 - Dataset3

Modelo: Redes Neurais**Nº Configuração:** 5**Nome do Dataset:** Dataset3**Train Accuracy:** 100%**Train Confusion Matrix:**

107	0	0
0	254	0
0	0	264

Classification Report:

	precision	recall	f1-score	support
1	1.00	1.00	1.00	107
2	1.00	1.00	1.00	254
3	1.00	1.00	1.00	264

accuracy			1.00	625
macro avg	1.00	1.00	1.00	625
weighted avg	1.00	1.00	1.00	625

Test Accuracy: 43,31%**Test Confusion Matrix:**

6	8	5
14	27	24
10	28	35

Classification Report:

	precision	recall	f1-score	support
1	0.20	0.32	0.24	19
2	0.43	0.42	0.42	65
3	0.55	0.48	0.51	73

accuracy			0.43	157
macro avg	0.39	0.40	0.39	157
weighted avg	0.46	0.43	0.44	157

Tabela A.21: Resultados de Treino e Teste - Redes Neurais - Configuração 6 - Dataset3

Modelo: Redes Neurais**Nº Configuração:** 6**Nome do Dataset:** Dataset3**Train Accuracy:** 47,36%**Train Confusion Matrix:**

0	63	44
1	151	102
1	118	145

Classification Report:

	precision	recall	f1-score	support
1	0.00	0.00	0.00	107
2	0.45	0.59	0.52	254
3	0.50	0.55	0.52	264

accuracy			0.47	625
macro avg	0.32	0.38	0.35	625
weighted avg	0.40	0.47	0.43	625

Test Accuracy: 50,96%**Test Confusion Matrix:**

0	12	7
0	38	27
0	31	42

Classification Report:

	precision	recall	f1-score	support
1	0.00	0.00	0.00	19
2	0.47	0.58	0.52	65
3	0.55	0.58	0.56	73

accuracy			0.51	157
macro avg	0.34	0.39	0.36	157
weighted avg	0.45	0.51	0.48	157

Tabela A.22: Resultados de Treino e Teste - Redes Neurais - Configuração 1 - Dataset4

Modelo: Redes Neurais**Nº Configuração:** 1**Nome do Dataset:** Dataset4**Train Accuracy:** 52,98%**Train Confusion Matrix:**

0	7	279
0	16	469
1	17	855

Classification Report:

	precision	recall	f1-score	support
1	0.00	0.00	0.00	286
2	0.40	0.03	0.06	485
3	0.53	0.98	0.69	873

accuracy			0.53	1644
macro avg	0.31	0.34	0.25	1644
weighted avg	0.40	0.53	0.38	1644

Test Accuracy: 51,34%**Test Confusion Matrix:**

0	0	62
0	0	134
0	4	211

Classification Report:

	precision	recall	f1-score	support
1	0.00	0.00	0.00	62
2	0.00	0.00	0.00	134
3	0.52	0.98	0.68	215

accuracy			0.51	411
macro avg	0.17	0.33	0.23	411
weighted avg	0.27	0.51	0.35	411

Tabela A.23: Resultados de Treino e Teste - Redes Neurais - Configuração 2 - Dataset4

Modelo: Redes Neurais**Nº Configuração:** 2**Nome do Dataset:** Dataset4**Train Accuracy:** 52,98%**Train Confusion Matrix:**

0	10	276
0	18	467
1	19	853

Classification Report:

	precision	recall	f1-score	support
1	0.00	0.00	0.00	286
2	0.38	0.04	0.07	485
3	0.53	0.98	0.69	873

accuracy			0.53	1644
macro avg	0.31	0.34	0.25	1644
weighted avg	0.40	0.53	0.39	1644

Test Accuracy: 52,07%**Test Confusion Matrix:**

0	0	62
0	1	133
0	2	213

Classification Report:

	precision	recall	f1-score	support
1	0.00	0.00	0.00	62
2	0.33	0.01	0.01	134
3	0.52	0.99	0.68	215

accuracy			0.52	411
macro avg	0.29	0.33	0.23	411
weighted avg	0.38	0.52	0.36	411

Tabela A.24: Resultados de Treino e Teste - Redes Neurais - Configuração 3 - Dataset4

Modelo: Redes Neurais**Nº Configuração:** 3**Nome do Dataset:** Dataset4**Train Accuracy:** 53,10%**Train Confusion Matrix:**

0	8	278
0	21	464
1	20	852

Classification Report:

	precision	recall	f1-score	support
1	0.00	0.00	0.00	286
2	0.43	0.04	0.08	485
3	0.53	0.98	0.69	873

accuracy			0.53	1644
macro avg	0.32	0.34	0.26	1644
weighted avg	0.41	0.53	0.39	1644

Test Accuracy: 52,07%**Test Confusion Matrix:**

0	0	62
0	0	134
0	1	214

Classification Report:

	precision	recall	f1-score	support
1	0.00	0.00	0.00	62
2	0.00	0.00	0.00	134
3	0.52	1.00	0.68	215

accuracy			0.52	411
macro avg	0.17	0.33	0.23	411
weighted avg	0.27	0.52	0.36	411

Tabela A.25: Resultados de Treino e Teste - Redes Neurais - Configuração 4 - Dataset4

Modelo: Redes Neurais**Nº Configuração:** 4**Nome do Dataset:** Dataset4**Train Accuracy:** 80,23%**Train Confusion Matrix:**

164	20	102
4	331	150
4	45	824

Classification Report:

	precision	recall	f1-score	support
1	0.95	0.57	0.72	286
2	0.84	0.68	0.75	485
3	0.77	0.94	0.85	873

accuracy			0.80	1644
macro avg	0.85	0.73	0.77	1644
weighted avg	0.82	0.80	0.80	1644

Test Accuracy: 42,82%**Test Confusion Matrix:**

2	16	44
6	21	107
14	48	153

Classification Report:

	precision	recall	f1-score	support
1	0.09	0.03	0.05	62
2	0.25	0.16	0.19	134
3	0.50	0.71	0.59	215

accuracy			0.43	411
macro avg	0.28	0.30	0.28	411
weighted avg	0.36	0.43	0.38	411

Tabela A.26: Resultados de Treino e Teste - Redes Neurais - Configuração 5 - Dataset4

Modelo: Redes Neurais**Nº Configuração:** 5**Nome do Dataset:** Dataset4**Train Accuracy:** 100%**Train Confusion Matrix:**

286	8	278
0	485	464
1	20	873

Classification Report:

	precision	recall	f1-score	support
1	1.00	1.00	1.00	286
2	1.00	1.00	1.00	485
3	1.00	1.00	1.00	873

accuracy			1.00	1644
macro avg	1.00	1.00	1.00	1644
weighted avg	1.00	1.00	1.00	1644

Test Accuracy: 39,17%**Test Confusion Matrix:**

8	12	42
24	37	73
38	61	116

Classification Report:

	precision	recall	f1-score	support
1	0.11	0.13	0.12	62
2	0.34	0.28	0.30	134
3	0.50	0.54	0.52	215

accuracy			0.39	411
macro avg	0.32	0.31	0.31	411
weighted avg	0.39	0.39	0.39	411

Tabela A.27: Resultados de Treino e Teste - Redes Neurais - Configuração 6 - Dataset4

Modelo: Redes Neurais**Nº Configuração:** 6**Nome do Dataset:** Dataset4**Train Accuracy:** 52,92%**Train Confusion Matrix:**

0	10	276
1	22	462
4	25	848

Classification Report:

	precision	recall	f1-score	support
1	0.00	0.00	0.00	286
2	0.39	0.05	0.08	485
3	0.53	0.97	0.69	873

accuracy			0.53	1644
macro avg	0.31	0.34	0.26	1644
weighted avg	0.40	0.53	0.39	1644

Test Accuracy: 51,58%**Test Confusion Matrix:**

0	0	62
0	0	134
0	3	212

Classification Report:

	precision	recall	f1-score	support
1	0.00	0.00	0.00	62
2	0.00	0.00	0.00	134
3	0.52	0.99	0.68	215

accuracy			0.52	411
macro avg	0.17	0.33	0.23	411
weighted avg	0.27	0.52	0.36	411

Tabela A.28: Resultados de Treino e Teste - *Random Forest* - Configuração 1 - Dataset1**Modelo:** *Random Forest***Nº Configuração:** 1**Nome do Dataset:** Dataset1**Train Accuracy:** 100%**Train Confusion Matrix:**

17	0	0
0	55	0
0	0	80

Classification Report:

	precision	recall	f1-score	support
1	1.00	1.00	1.00	17
2	1.00	1.00	1.00	55
3	1.00	1.00	1.00	80

accuracy			1.00	152
macro avg	1.00	1.00	1.00	152
weighted avg	1.00	1.00	1.00	152

Test Accuracy: 60,60%**Test Confusion Matrix:**

1	4	3
0	7	5
0	14	32

Classification Report:

	precision	recall	f1-score	support
1	1.00	0.12	0.22	8
2	0.28	0.58	0.38	12
3	0.80	0.70	0.74	46

accuracy			0.61	66
macro avg	0.69	0.47	0.45	66
weighted avg	0.73	0.61	0.61	66

Tabela A.29: Resultados de Treino e Teste - *Random Forest* - Configuração 2 - Dataset1**Modelo:** *Random Forest***Nº Configuração:** 2**Nome do Dataset:** Dataset1**Train Accuracy:** 100%**Train Confusion Matrix:**

17	0	0
0	55	0
0	0	80

Classification Report:

	precision	recall	f1-score	support
1	1.00	1.00	1.00	17
2	1.00	1.00	1.00	55
3	1.00	1.00	1.00	80

accuracy			1.00	152
macro avg	1.00	1.00	1.00	152
weighted avg	1.00	1.00	1.00	152

Test Accuracy: 60,60%**Test Confusion Matrix:**

0	4	4
0	7	5
0	13	33

Classification Report:

	precision	recall	f1-score	support
1	1.00	0.12	0.22	8
2	0.29	0.58	0.39	12
3	0.79	0.70	0.75	46

accuracy			0.61	66
macro avg	0.36	0.43	0.38	66
weighted avg	0.60	0.61	0.59	66

Tabela A.30: Resultados de Treino e Teste - *Random Forest* - Configuração 3 - Dataset1**Modelo:** *Random Forest***Nº Configuração:** 3**Nome do Dataset:** Dataset1**Train Accuracy:** 100%**Train Confusion Matrix:**

17	0	0
0	55	0
0	0	80

Classification Report:

	precision	recall	f1-score	support
1	1.00	1.00	1.00	17
2	1.00	1.00	1.00	55
3	1.00	1.00	1.00	80

accuracy			1.00	152
macro avg	1.00	1.00	1.00	152
weighted avg	1.00	1.00	1.00	152

Test Accuracy: 60,60%**Test Confusion Matrix:**

1	4	3
0	7	5
0	14	32

Classification Report:

	precision	recall	f1-score	support
1	1.00	0.12	0.22	8
2	0.28	0.58	0.38	12
3	0.80	0.70	0.74	46

accuracy			0.61	66
macro avg	0.69	0.47	0.45	66
weighted avg	0.73	0.61	0.61	66

Tabela A.31: Resultados de Treino e Teste - *Random Forest* - Configuração 4 - Dataset1**Modelo:** *Random Forest***Nº Configuração:** 2**Nome do Dataset:** Dataset1**Train Accuracy:** 100%**Train Confusion Matrix:**

17	0	0
0	55	0
0	0	80

Classification Report:

	precision	recall	f1-score	support
1	1.00	1.00	1.00	17
2	1.00	1.00	1.00	55
3	1.00	1.00	1.00	80

accuracy			1.00	152
macro avg	1.00	1.00	1.00	152
weighted avg	1.00	1.00	1.00	152

Test Accuracy: 60,60%**Test Confusion Matrix:**

0	4	4
0	7	5
0	13	33

Classification Report:

	precision	recall	f1-score	support
1	1.00	0.12	0.22	8
2	0.29	0.58	0.39	12
3	0.79	0.70	0.75	46

accuracy			0.61	66
macro avg	0.36	0.43	0.38	66
weighted avg	0.60	0.61	0.59	66

Tabela A.32: Resultados de Treino e Teste - *Random Forest* - Configuração 5 - Dataset1**Modelo:** *Random Forest***Nº Configuração:** 5**Nome do Dataset:** Dataset1**Train Accuracy:** 100%**Train Confusion Matrix:**

17	0	0
0	55	0
0	0	80

Classification Report:

	precision	recall	f1-score	support
1	1.00	1.00	1.00	17
2	1.00	1.00	1.00	55
3	1.00	1.00	1.00	80

accuracy			1.00	152
macro avg	1.00	1.00	1.00	152
weighted avg	1.00	1.00	1.00	152

Test Accuracy: 60,60%**Test Confusion Matrix:**

0	3	5
0	7	5
0	13	33

Classification Report:

	precision	recall	f1-score	support
1	0.00	0.00	0.00	8
2	0.30	0.58	0.40	12
3	0.77	0.72	0.74	46

accuracy			0.61	66
macro avg	0.36	0.43	0.38	66
weighted avg	0.59	0.61	0.59	66

Tabela A.33: Resultados de Treino e Teste - *Random Forest* - Configuração 6 - Dataset1**Modelo:** *Random Forest***Nº Configuração:** 6**Nome do Dataset:** Dataset1**Train Accuracy:** 100%**Train Confusion Matrix:**

17	0	0
0	55	0
0	0	80

Classification Report:

	precision	recall	f1-score	support
1	1.00	1.00	1.00	17
2	1.00	1.00	1.00	55
3	1.00	1.00	1.00	80

accuracy			1.00	152
macro avg	1.00	1.00	1.00	152
weighted avg	1.00	1.00	1.00	152

Test Accuracy: 59,09%**Test Confusion Matrix:**

0	4	4
0	7	5
0	14	32

Classification Report:

	precision	recall	f1-score	support
1	0.00	0.00	0.00	8
2	0.28	0.58	0.38	12
3	0.78	0.70	0.74	46

accuracy			0.61	66
macro avg	0.35	0.43	0.37	66
weighted avg	0.59	0.59	0.58	66

Tabela A.34: Resultados de Treino e Teste - *Random Forest* - Configuração 1 - Dataset2**Modelo:** *Random Forest***Nº Configuração:** 1**Nome do Dataset:** Dataset2**Train Accuracy:** 100%**Train Confusion Matrix:**

29	0	0
0	62	0
0	0	155

Classification Report:

	precision	recall	f1-score	support
1	1.00	1.00	1.00	29
2	1.00	1.00	1.00	62
3	1.00	1.00	1.00	155

accuracy			1.00	246
macro avg	1.00	1.00	1.00	246
weighted avg	1.00	1.00	1.00	246

Test Accuracy: 57,55%**Test Confusion Matrix:**

2	1	13
0	6	29
0	2	53

Classification Report:

	precision	recall	f1-score	support
1	1.00	0.12	0.22	16
2	0.67	0.17	0.27	35
3	0.56	0.96	0.71	55

accuracy			0.58	106
macro avg	0.74	0.42	0.40	106
weighted avg	0.66	0.58	0.49	106

Tabela A.35: Resultados de Treino e Teste - *Random Forest* - Configuração 2 - Dataset2**Modelo:** *Random Forest***Nº Configuração:** 2**Nome do Dataset:** Dataset2**Train Accuracy:** 97,56%**Train Confusion Matrix:**

25	0	4
0	60	2
0	0	155

Classification Report:

	precision	recall	f1-score	support
1	1.00	0.86	0.93	29
2	1.00	0.97	0.98	62
3	0.96	1.00	0.98	155

accuracy			0.98	246
macro avg	0.99	0.94	0.96	246
weighted avg	0.98	0.98	0.98	246

Test Accuracy: 56,60%**Test Confusion Matrix:**

0	0	16
0	6	29
0	1	54

Classification Report:

	precision	recall	f1-score	support
1	0.00	0.00	0.00	16
2	0.86	0.17	0.29	35
3	0.55	0.98	0.70	55

accuracy			0.57	106
macro avg	0.47	0.38	0.33	106
weighted avg	0.57	0.57	0.46	106

Tabela A.36: Resultados de Treino e Teste - *Random Forest* - Configuração 3 - Dataset2**Modelo:** *Random Forest***Nº Configuração:** 3**Nome do Dataset:** Dataset2**Train Accuracy:** 100%**Train Confusion Matrix:**

29	0	0
0	62	0
0	0	155

Classification Report:

	precision	recall	f1-score	support
1	1.00	1.00	1.00	29
2	1.00	1.00	1.00	62
3	1.00	1.00	1.00	155

accuracy			1.00	246
macro avg	1.00	1.00	1.00	246
weighted avg	1.00	1.00	1.00	246

Test Accuracy: 57,55%**Test Confusion Matrix:**

2	0	14
0	5	30
0	1	54

Classification Report:

	precision	recall	f1-score	support
1	1.00	0.12	0.22	16
2	0.83	0.14	0.24	35
3	0.55	0.98	0.71	55

accuracy			0.58	106
macro avg	0.79	0.42	0.39	106
weighted avg	0.71	0.58	0.48	106

Tabela A.37: Resultados de Treino e Teste - *Random Forest* - Configuração 4 - Dataset2**Modelo:** *Random Forest***Nº Configuração:** 4**Nome do Dataset:** Dataset2**Train Accuracy:** 97,97%**Train Confusion Matrix:**

25	0	4
0	61	1
0	0	155

Classification Report:

	precision	recall	f1-score	support
1	1.00	0.86	0.93	29
2	1.00	0.98	0.99	62
3	0.97	1.00	0.98	155

accuracy			0.98	246
macro avg	0.99	0.95	0.97	246
weighted avg	0.98	0.98	0.98	246

Test Accuracy: 55,66%**Test Confusion Matrix:**

0	0	16
0	5	30
0	1	54

Classification Report:

	precision	recall	f1-score	support
1	0.00	0.00	0.00	16
2	0.83	0.14	0.24	35
3	0.54	0.98	0.70	55

accuracy			0.56	106
macro avg	0.46	0.37	0.31	106
weighted avg	0.56	0.56	0.44	106

Tabela A.38: Resultados de Treino e Teste - *Random Forest* - Configuração 5 - Dataset2**Modelo:** *Random Forest***Nº Configuração:** 5**Nome do Dataset:** Dataset2**Train Accuracy:** 100%**Train Confusion Matrix:**

29	0	0
0	62	0
0	0	155

Classification Report:

	precision	recall	f1-score	support
1	1.00	1.00	1.00	29
2	1.00	1.00	1.00	62
3	1.00	1.00	1.00	155

accuracy			1.00	246
macro avg	1.00	1.00	1.00	246
weighted avg	1.00	1.00	1.00	246

Test Accuracy: 56,60%**Test Confusion Matrix:**

1	0	15
0	5	30
0	1	54

Classification Report:

	precision	recall	f1-score	support
1	1.00	0.06	0.12	16
2	0.83	0.14	0.24	35
3	0.55	0.98	0.70	55

accuracy			0.57	106
macro avg	0.79	0.40	0.35	106
weighted avg	0.71	0.57	0.46	106

Tabela A.39: Resultados de Treino e Teste - *Random Forest* - Configuração 6 - Dataset2**Modelo:** *Random Forest***Nº Configuração:** 6**Nome do Dataset:** Dataset2**Train Accuracy:** 98,37%**Train Confusion Matrix:**

26	0	3
0	61	1
0	0	155

Classification Report:

	precision	recall	f1-score	support
1	1.00	0.90	0.95	29
2	1.00	0.98	0.99	62
3	0.97	1.00	0.99	155

accuracy			0.98	246
macro avg	0.99	0.96	0.97	246
weighted avg	0.98	0.98	0.98	246

Test Accuracy: 57,55%**Test Confusion Matrix:**

1	0	15
0	6	29
0	1	54

Classification Report:

	precision	recall	f1-score	support
1	1.00	0.06	0.12	16
2	0.86	0.17	0.29	35
3	0.55	0.98	0.71	55

accuracy			0.58	106
macro avg	0.80	0.41	0.37	106
weighted avg	0.72	0.58	0.48	106

Tabela A.40: Resultados de Treino e Teste - *Random Forest* - Configuração 1 - Dataset3**Modelo:** *Random Forest***Nº Configuração:** 1**Nome do Dataset:** Dataset3**Train Accuracy:** 100%**Train Confusion Matrix:**

86	0	0
0	220	0
0	0	241

Classification Report:

	precision	recall	f1-score	support
1	1.00	1.00	1.00	86
2	1.00	1.00	1.00	220
3	1.00	1.00	1.00	241

accuracy			1.00	547
macro avg	1.00	1.00	1.00	547
weighted avg	1.00	1.00	1.00	547

Test Accuracy: 51,91%**Test Confusion Matrix:**

2	20	18
0	55	44
0	31	65

Classification Report:

	precision	recall	f1-score	support
1	1.00	0.05	0.10	40
2	0.83	0.14	0.24	99
3	0.55	0.98	0.70	96

accuracy			0.52	235
macro avg	0.68	0.43	0.40	235
weighted avg	0.60	0.52	0.48	235

Tabela A.41: Resultados de Treino e Teste - *Random Forest* - Configuração 2 - Dataset3**Modelo:** *Random Forest***Nº Configuração:** 2**Nome do Dataset:** Dataset3**Train Accuracy:** 100%**Train Confusion Matrix:**

86	0	0
0	220	0
0	0	241

Classification Report:

	precision	recall	f1-score	support
1	1.00	1.00	1.00	86
2	1.00	1.00	1.00	220
3	1.00	1.00	1.00	241

accuracy			1.00	547
macro avg	1.00	1.00	1.00	547
weighted avg	1.00	1.00	1.00	547

Test Accuracy: 55,31%**Test Confusion Matrix:**

0	17	23
0	64	35
0	30	66

Classification Report:

	precision	recall	f1-score	support
1	0.00	0.00	0.00	40
2	0.58	0.65	0.61	99
3	0.53	0.69	0.60	96

accuracy			0.55	235
macro avg	0.37	0.44	0.40	235
weighted avg	0.46	0.55	0.50	235

Tabela A.42: Resultados de Treino e Teste - *Random Forest* - Configuração 3 - Dataset3**Modelo:** *Random Forest***Nº Configuração:** 3**Nome do Dataset:** Dataset3**Train Accuracy:** 100%**Train Confusion Matrix:**

86	0	0
0	220	0
0	0	241

Classification Report:

	precision	recall	f1-score	support
1	1.00	1.00	1.00	86
2	1.00	1.00	1.00	220
3	1.00	1.00	1.00	241

accuracy			1.00	547
macro avg	1.00	1.00	1.00	547
weighted avg	1.00	1.00	1.00	547

Test Accuracy: 53,19%**Test Confusion Matrix:**

2	20	18
0	60	39
0	33	63

Classification Report:

	precision	recall	f1-score	support
1	1.00	0.05	0.10	40
2	0.53	0.61	0.57	99
3	0.53	0.66	0.58	96

accuracy			0.53	235
macro avg	0.69	0.44	0.41	235
weighted avg	0.61	0.53	0.49	235

Tabela A.43: Resultados de Treino e Teste - *Random Forest* - Configuração 4 - Dataset3**Modelo:** *Random Forest***Nº Configuração:** 4**Nome do Dataset:** Dataset3**Train Accuracy:** 100%**Train Confusion Matrix:**

86	0	0
0	220	0
0	0	241

Classification Report:

	precision	recall	f1-score	support
1	1.00	1.00	1.00	86
2	1.00	1.00	1.00	220
3	1.00	1.00	1.00	241

accuracy			1.00	547
macro avg	1.00	1.00	1.00	547
weighted avg	1.00	1.00	1.00	547

Test Accuracy: 56,17%**Test Confusion Matrix:**

0	18	22
0	64	35
0	28	68

Classification Report:

	precision	recall	f1-score	support
1	0.00	0.00	0.00	40
2	0.58	0.65	0.61	99
3	0.54	0.71	0.62	96

accuracy			0.56	235
macro avg	0.38	0.45	0.41	235
weighted avg	0.47	0.56	0.51	235

Tabela A.44: Resultados de Treino e Teste - *Random Forest* - Configuração 5 - Dataset3**Modelo:** *Random Forest***Nº Configuração:** 5**Nome do Dataset:** Dataset3**Train Accuracy:** 100%**Train Confusion Matrix:**

86	0	0
0	220	0
0	0	241

Classification Report:

	precision	recall	f1-score	support
1	1.00	1.00	1.00	86
2	1.00	1.00	1.00	220
3	1.00	1.00	1.00	241

accuracy			1.00	547
macro avg	1.00	1.00	1.00	547
weighted avg	1.00	1.00	1.00	547

Test Accuracy: 52,77%**Test Confusion Matrix:**

1	23	16
0	59	40
0	32	64

Classification Report:

	precision	recall	f1-score	support
1	1.00	0.03	0.05	40
2	0.52	0.60	0.55	99
3	0.53	0.67	0.59	96

accuracy			0.53	235
macro avg	0.68	0.43	0.40	235
weighted avg	0.61	0.53	0.48	235

Tabela A.45: Resultados de Treino e Teste - *Random Forest* - Configuração 6 - Dataset3**Modelo:** *Random Forest***Nº Configuração:** 6**Nome do Dataset:** Dataset3**Train Accuracy:** 100%**Train Confusion Matrix:**

86	0	0
0	220	0
0	0	241

Classification Report:

	precision	recall	f1-score	support
1	1.00	1.00	1.00	86
2	1.00	1.00	1.00	220
3	1.00	1.00	1.00	241

accuracy			1.00	547
macro avg	1.00	1.00	1.00	547
weighted avg	1.00	1.00	1.00	547

Test Accuracy: 54,89%**Test Confusion Matrix:**

0	18	22
0	65	34
0	32	64

Classification Report:

	precision	recall	f1-score	support
1	0.00	0.00	0.00	40
2	0.57	0.66	0.61	99
3	0.53	0.67	0.59	96

accuracy			0.55	235
macro avg	0.37	0.44	0.40	235
weighted avg	0.46	0.55	0.50	235

Tabela A.46: Resultados de Treino e Teste - *Random Forest* - Configuração 1 - Dataset4**Modelo:** *Random Forest***Nº Configuração:** 1**Nome do Dataset:** Dataset4**Train Accuracy:** 100%**Train Confusion Matrix:**

243	0	0
0	442	0
0	0	753

Classification Report:

	precision	recall	f1-score	support
1	1.00	1.00	1.00	243
2	1.00	1.00	1.00	442
3	1.00	1.00	1.00	753

accuracy			1.00	1438
macro avg	1.00	1.00	1.00	1438
weighted avg	1.00	1.00	1.00	1438

Test Accuracy: 55,10%**Test Confusion Matrix:**

2	6	97
0	37	140
0	34	301

Classification Report:

	precision	recall	f1-score	support
1	1.00	0.02	0.04	105
2	0.48	0.21	0.29	177
3	0.56	0.90	0.69	335

accuracy			0.55	617
macro avg	0.68	0.38	0.34	617
weighted avg	0.61	0.55	0.46	617

Tabela A.47: Resultados de Treino e Teste - *Random Forest* - Configuração 3 - Dataset4**Modelo:** *Random Forest***Nº Configuração:** 3**Nome do Dataset:** Dataset4**Train Accuracy:** 100%**Train Confusion Matrix:**

243	0	0
0	442	0
0	0	753

Classification Report:

	precision	recall	f1-score	support
1	1.00	1.00	1.00	243
2	1.00	1.00	1.00	442
3	1.00	1.00	1.00	753

accuracy			1.00	1438
macro avg	1.00	1.00	1.00	1438
weighted avg	1.00	1.00	1.00	1438

Test Accuracy: 54,94%**Test Confusion Matrix:**

2	6	97
0	32	145
0	30	305

Classification Report:

	precision	recall	f1-score	support
1	1.00	0.02	0.04	105
2	0.47	0.18	0.26	177
3	0.56	0.91	0.69	335

accuracy			0.55	617
macro avg	0.68	0.37	0.33	617
weighted avg	0.61	0.55	0.46	617

Tabela A.48: Resultados de Treino e Teste - *Random Forest* - Configuração 4 - Dataset4**Modelo:** *Random Forest***Nº Configuração:** 4**Nome do Dataset:** Dataset4**Train Accuracy:** 99,30%**Train Confusion Matrix:**

236	1	6
0	439	3
0	0	753

Classification Report:

	precision	recall	f1-score	support
1	1.00	0.97	0.99	243
2	1.00	0.99	1.00	442
3	0.99	1.00	0.99	753

accuracy			0.99	1438
macro avg	1.00	0.99	0.99	1438
weighted avg	0.99	0.99	0.99	1438

Test Accuracy: 55,11%**Test Confusion Matrix:**

2	8	95
0	34	143
0	31	304

Classification Report:

	precision	recall	f1-score	support
1	1.00	0.02	0.04	105
2	0.47	0.19	0.27	177
3	0.56	0.91	0.69	335

accuracy			0.55	617
macro avg	0.68	0.37	0.33	617
weighted avg	0.61	0.55	0.46	617

Tabela A.49: Resultados de Treino e Teste - *Random Forest* - Configuração 5 - Dataset4**Modelo:** *Random Forest***Nº Configuração:** 5**Nome do Dataset:** Dataset4**Train Accuracy:** 100%**Train Confusion Matrix:**

243	0	0
0	442	0
0	0	753

Classification Report:

	precision	recall	f1-score	support
1	1.00	1.00	1.00	243
2	1.00	1.00	1.00	442
3	1.00	1.00	1.00	753

accuracy			1.00	1438
macro avg	1.00	1.00	1.00	1438
weighted avg	1.00	1.00	1.00	1438

Test Accuracy: 54,62%**Test Confusion Matrix:**

2	8	95
0	34	143
0	34	301

Classification Report:

	precision	recall	f1-score	support
1	1.00	0.02	0.04	105
2	0.45	0.19	0.27	177
3	0.56	0.90	0.69	335

accuracy			0.55	617
macro avg	0.67	0.37	0.33	617
weighted avg	0.60	0.55	0.46	617

Tabela A.50: Resultados de Treino e Teste - *Random Forest* - Configuração 6 - Dataset4**Modelo:** *Random Forest***Nº Configuração:** 6**Nome do Dataset:** Dataset4**Train Accuracy:** 99,51%**Train Confusion Matrix:**

237	1	5
0	441	1
0	0	753

Classification Report:

	precision	recall	f1-score	support
1	1.00	0.98	0.99	243
2	1.00	1.00	1.00	442
3	0.99	1.00	1.00	753

accuracy			1.00	1438
macro avg	1.00	0.99	0.99	1438
weighted avg	1.00	1.00	1.00	1438

Test Accuracy: 55,27%**Test Confusion Matrix:**

1	8	96
0	33	144
0	28	307

Classification Report:

	precision	recall	f1-score	support
1	1.00	0.01	0.02	105
2	0.48	0.19	0.27	177
3	0.56	0.92	0.70	335

accuracy			0.55	617
macro avg	0.68	0.37	0.33	617
weighted avg	0.61	0.55	0.46	617

Tabela A.51: Resultados de Treino e Teste - *Support Vector Machines* - Configuração 1 - Dataset1**Modelo:** *Support Vector Machines***Nº Configuração:** 1**Nome do Dataset:** Dataset1**Train Accuracy:** 68,42%**Train Confusion Matrix:**

7	3	7
0	26	29
1	8	71

Classification Report:

	precision	recall	f1-score	support
1	0.88	0.41	0.56	17
2	0.70	0.47	0.57	55
3	0.66	0.89	0.76	80

accuracy			0.68	152
macro avg	0.75	0.59	0.63	152
weighted avg	0.70	0.68	0.67	152

Test Accuracy: 56,06%**Test Confusion Matrix:**

0	1	7
1	5	6
2	12	32

Classification Report:

	precision	recall	f1-score	support
1	0.00	0.00	0.00	8
2	0.28	0.42	0.33	12
3	0.56	0.90	0.69	46

accuracy			0.56	66
macro avg	0.33	0.37	0.35	66
weighted avg	0.55	0.56	0.55	66

Tabela A.52: Resultados de Treino e Teste - *Support Vector Machines* - Configuração 1 - Dataset2**Modelo:** *Support Vector Machines***Nº Configuração:** 1**Nome do Dataset:** Dataset2**Train Accuracy:** 68,86%**Train Confusion Matrix:**

3	1	25
0	20	42
0	9	146

Classification Report:

	precision	recall	f1-score	support
1	1.00	0.10	0.19	29
2	0.70	0.47	0.57	62
3	0.66	0.89	0.76	155

accuracy			0.69	246
macro avg	0.78	0.46	0.47	246
weighted avg	0.72	0.69	0.63	246

Test Accuracy: 45,28%**Test Confusion Matrix:**

1	0	15
0	0	35
2	6	47

Classification Report:

	precision	recall	f1-score	support
1	0.33	0.06	0.11	16
2	0.00	0.00	0.00	35
3	0.48	0.85	0.62	55

accuracy			0.45	106
macro avg	0.27	0.31	0.24	106
weighted avg	0.30	0.45	0.34	106

Tabela A.53: Resultados de Treino e Teste - *Support Vector Machines* - Configuração 1 - Dataset3**Modelo:** *Support Vector Machines***Nº Configuração:** 1**Nome do Dataset:** Dataset3**Train Accuracy:** 49,73%**Train Confusion Matrix:**

0	44	42
0	123	97
0	92	149

Classification Report:

	precision	recall	f1-score	support
1	0.00	0.00	0.00	86
2	0.70	0.47	0.57	220
3	0.66	0.89	0.76	241

accuracy			0.50	547
macro avg	0.33	0.39	0.36	547
weighted avg	0.42	0.50	0.45	547

Test Accuracy: 42,98%**Test Confusion Matrix:**

0	16	24
0	54	45
0	49	47

Classification Report:

	precision	recall	f1-score	support
1	0.00	0.00	0.00	40
2	0.45	0.55	0.50	99
3	0.56	0.90	0.69	96

accuracy			0.43	235
macro avg	0.29	0.35	0.31	235
weighted avg	0.36	0.43	0.39	235

Tabela A.54: Resultados de Treino e Teste - *Support Vector Machines* - Configuração 1 - Dataset4**Modelo:** *Support Vector Machines***Nº Configuração:** 1**Nome do Dataset:** Dataset4**Train Accuracy:** 68,86%**Train Confusion Matrix:**

0	0	243
0	0	442
0	0	753

Classification Report:

	precision	recall	f1-score	support
1	0.00	0.00	0.00	243
2	0.00	0.00	0.00	442
3	0.52	1.00	0.69	753

accuracy			0.52	1438
macro avg	0.17	0.33	0.23	1438
weighted avg	0.27	0.52	0.36	1438

Test Accuracy: 54,29%**Test Confusion Matrix:**

0	0	105
0	0	177
0	0	335

Classification Report:

	precision	recall	f1-score	support
1	0.00	0.00	0.00	105
2	0.00	0.00	0.00	177
3	0.54	1.00	0.70	335

accuracy			0.54	617
macro avg	0.18	0.33	0.23	617
weighted avg	0.29	0.54	0.38	617