INSTITUTO
UNIVERSITÁRIO
DE LISBOA

Mestrado em Tecnologias Digitais para o Negócio
2024/2025

Integração de Inteligência Artificial e *Machine Learning* em programas de formação nas organizações

Paulo Manuel Martins Bernardino

Orientador
Professor Doutor Ruben Pereira
ISCTE-IUL

Dezembro de 2024

Agradecimentos

À minha família pela compreensão, presença, carinho e confiança…

Ao meu orientador o Prof. Ruben, pelo apoio e acompanhamento que foram essenciais para desenvolver este projeto...

Aos meus colegas de trabalho e amigos, especialmente o Leandro e Jonathan pela disponibilidade e ajuda prestada…

Resumo

As empresas enfrentam desafios relacionados com a competitividade, eficácia e retenção de talento, bem como com a dificuldade em obter informações detalhadas sobre o impacto do investimento realizado na formação dos colaboradores e a sua contribuição para a criação de valor na entidade e assim compreender melhor se os programas formativos fornecidos pelas mesmas estão adequados às funções a desempenhar.

Este trabalho apresenta o desenvolvimento de um sistema de recomendação para apoiar a gestão de formação e o desenvolvimento de competências nas organizações. O sistema foca-se em três funcionalidades principais: antecipação de saídas profissionais, previsão de profissões futuras e recomendação de formações com base nas competências. Utilizando dados de colaboradores e de um catálogo de competências, o sistema combina métodos de *machine learning* e análise de dados para oferecer recomendações personalizadas.

O sistema também integra uma interface amigável, que facilita a navegação e a escolhas pelos gestores. Apesar do sucesso em diversas áreas, foram identificadas limitações, como a precisão moderada na previsão de profissões futuras e a ausência de dados detalhados sobre funções, que abrem caminho para melhorias futuras.

Com este projeto, espera-se contribuir para a eficiência e eficácia dos programas de formação, promovendo a retenção de talentos, o desenvolvimento contínuo dos colaboradores e o fortalecimento da cultura organizacional.

Palavras-chave: Aprendizagem automática, Competência, Recomendação, Previsão

Abstract

Companies face challenges related to competitiveness, effectiveness and talent retention, as well as the difficulty of obtaining detailed information on the impact of the investment made in employee training and its contribution to the creation of value in the entity, and thus better understand whether the training programs provided by them are appropriate to the functions to be performed.

This paper presents the development of a recommendation system to support training management and skills development in organizations. The system focuses on three main functionalities: anticipating career paths, predicting future jobs and recommending training based on skills. Using data from employees and a skills catalog, the system combines machine learning methods and data analysis to offer personalized recommendations.

The system also integrates a user-friendly interface that makes it easy for managers to navigate and make choices. Despite its success in several areas, limitations have been identified, such as moderate accuracy in predicting future professions and the lack of detailed data on functions, which open the way for future improvements.

With this project, it is hoped to contribute to the efficiency and effectiveness of training programs, promoting talent retention, continuous employee development and strengthening the organizational culture.

Keywords: Machine Learning, Competence, Recommendation, Prediction

Índice

Resumo iii
Abstract iv
Índice de Figuras vi
Índice de Tabelas viii
Acrónimos ix
1. Introdução 1
2. Trabalho Relacionado 3
3. Metodologia de Investigação 15
4. Desenvolvimento da arquitetura 19
5. Demonstração 27
6. Avaliação 30
7. Conclusões e Trabalho futuro 32
Bibliografia 33
Anexos 37
1.1. Anexo I 37
1.2. Anexo II 39
1.3. Anexo III 44
1.4. Anexo IV 56
1.5. Anexo V 58
1.6. Anexo VI 59
1.7. Anexo VII 60

Índice de Figuras

Figura 1 - Fluxograma PRISMA para revisão sistemática 11
Figura 2 - Modelo de processo DSRM 15
Figura 3 - Ilustração de objetivos propostos 17
Figura 4 - BPMN - Caso 1 - Profissão futura 19
Figura 5 - BPMN - Caso 2 - Antecipar saídas 19
Figura 6 - BPMN - Caso 3 – Recomendação de cursos 20
Figura 7 - Arquitetura do sistema de recomendação 20
Figura 8 - Conjunto de dados - Lista de competências 21
Figura 9 - Resultados - Futura Profissão 25
Figura 10 - Resultados - Antecipar Saídas 26
Figura 11 - Lista de competências 27
Figura 12 - Antecipar saídas - Colaboradores 27
Figura 13 - Antecipar saídas - Colaboradores por profissão 28
Figura 14 - Futura profissão/função - Profissão 28
Figura 15 - Recomendar formação 29
Figura 16 - Vertex AI - Criar Dataset 39
Figura 17 - Vertex AI - Adicionar Dataset 40
Figura 18 - Vertex AI - Propriedades do dataset 40
Figura 19 - Vertex AI - Treinar o modelo - Método de treino 41
Figura 20 - Vertex AI - Treinar o modelo - Detalhes do modelo 42
Figura 21 - Vertex AI - Treinar o modelo - Opção de treino 42
Figura 22 - Vertex AI - Treinar o modelo - Compilar 43
Figura 23 - Vertex AI - Treinar o modelo - Testar 43
Figura 24 - Script - Serviços GCP 1/2 44
Figura 25 - Script - Serviços GCP 2/2 44
Figura 26 - Script - Dependências 45
Figura 27 - Script - Bibliotecas 45
Figura 28 - Script - Chave - GCP 46
Figura 29 - Script - Inicialização 47
Figura 30 - Script - Caminho para o conjunto de dados 47
Figura 31 - Script - Upload e Processamento - Inicio das conexões 47
Figura 32 - Script - Upload e Processamento - Renomeação das colunas 48
Figura 33 - Script - Upload e Processamento - Coluna da idade 48
Figura 34 - Script - Upload e Processamento - Colunas removidas 49
Figura 35 - Script - Upload e Processamento - Coluna "Proximo_Emprego" 49
Figura 36 - Script - Upload e Processamento - Formato do conjunto de dados 50
Figura 37 - Script - Upload e Processamento - Remoção de instâncias com valores nulos 50
Figura 38 - Script - Upload e Processamento - Número insuficiente de labels 50
Figura 39 - Script - Upload e Processamento - Processo de tradução 51
Figura 40 - Script - Upload e Processamento - Tradução 51
Figura 41 - Script - Upload e Processamento - Atualizar as profissões 51
Figura 42 - Script - Upload e Processamento - Upload do conjunto de dados 52
Figura 43 - Script - Upload e Processamento - Resultado 52
Figura 44 - Script - Criar o Dataset 52

Figura 45 - Script - Criar o modelo 53
Figura 46 - Script - Previsão - Algoritmo 54
Figura 47 - Script - Previsão - Instâncias 54
Figura 48 - Script - Previsão - Apresentação das previsões 54
Figura 49 - Script - Previsão - Resultado das previsões 55
Figura 50 - Script - Previsão - Desativar o endpoint 55
Figura 51 - Avaliação do modelo - Futura Profissão - Métricas 56
Figura 52 - Script - Processamento de dados 58
Figura 53 - Avaliação do modelo - Antecipar Saídas - Métricas 59
Figura 54 - Script - Guardar em CSV - 1/2 60
Figura 55 - Script - Guardar em CSV - 2/2 60
Figura 56 - Script - Guardar em CSV - Resultado 61

Índice de Tabelas

Tabela 1 - Termos pesquisados e filtros aplicados 9
Tabela 2 - Discussão - Sumarizada 13
Tabela 3 - Requisitos identificados 18
Tabela 4 - Conjunto de dados - Futura Profissão 21
Tabela 5 - Conjunto de dados - Antecipar Saídas 21
Tabela 6 - Conjunto de dados original 22
Tabela 7 - Resultados - Campos - Futura profissão 25
Tabela 8 - Resultados - Campos - Antecipar profissão 25
Tabela 9 - Pontos fortes e fracos 30
Tabela 10 - Avaliação dos utilizadores - Resumo 31
Tabela 11 - Legendas da Tabela 10 31
Tabela 12 - Descrição das colunas do Antecipar saídas 61

Acrónimos

Application programming interface	API
Bidirectional Encoder Representations from Transformers	BERT
Comma-separated values, Comma-separated values	CSV
Design Science Research Methodology	DMSR
Google Cloud Plataform	GCP
JavaScript Object Notation	JSON
Learning Managemant System	LMS
Machine Learning	ML
Natural Language Processing	NLP
Preferred Reporting Items for Systematic Reviews and Meta-Analyses	PRISMA

1. Introdução

Nos dias de hoje, o desenvolvimento profissional de cada colaborador é essencial para o sucesso de qualquer empresa, a educação contínua não só melhora as competências pessoais, como também aumenta o desempenho de uma organização [1]. Através de vários testemunhos de clientes e acesso a artigos online [2] [3], concluiu-se que as empresas enfrentam um problema significativo na adequação da formação correta para os seus colaboradores, mesmo quando oferecem formação ao colaborador pode não ser a mais indicada para o presente e futuro profissional do colaborador.

Diante deste cenário, a presente tese propõe o desenvolvimento de um sistema de recomendação, a ser integrado num software de gestão de formação, com o objetivo de otimizar o processo de planeamento e entrega de formação personalizada. O sistema visa identificar as competências em falta de cada colaborador e recomendar programas de formação adequados, promovendo um alinhamento mais eficiente entre as necessidades do colaborador e as metas da organização. Este sistema será projetado para ser adaptável ao perfil e às expectativas de cada colaborador, contribuindo para o seu desenvolvimento contínuo e para a retenção de talentos dentro da empresa.

Os sistemas de recomendação tornaram-se fundamentais nas interações diárias com produtos, serviços e conteúdos, mas a criação de sistemas eficazes pode levantar vários desafios como problemas relacionados com a dispersão, o "*cold start*" e a escalabilidade [4]. Métodos híbridos podem mitigar algumas questões, enquanto o "*overfitting*", "*underfitting*" e a falta de diversidade também afetam a precisão das recomendações [5]. Questões de privacidade, além de preocupações legais associadas a conteúdos nocivos, são problemas adicionais [6]. Para mitigar ou possivelmente colmatar estes desafios, o trabalho deste mestrado tem como objetivo:

- Desenvolver um sistema de recomendação que possa apoiar os recursos humanos na gestão do desenvolvimento das competências;
- Previsões de progressão de carreira;
- Identificação de possíveis saídas de colaboradores, promovendo a retenção e substituição de colaboradores de forma eficiente.

A metodologia adotada para esta pesquisa inclui uma revisão bibliográfica aprofundada sobre os conceitos de gestão da formação e aprendizagem personalizada, seguida pelo desenvolvimento do módulo utilizando técnicas de engenharia de software. A implementação será avaliada através de feedback de colaboradores e da organização, medindo o impacto na eficácia dos programas de formação e no desempenho dos colaboradores.

Com esta tese, espera-se contribuir para o campo da gestão da formação, oferecendo uma ferramenta prática que possa ser utilizada pelas empresas para melhorar a eficiência e eficácia dos seus programas de formação. A longo prazo, este módulo tem o potencial de influenciar positivamente a cultura organizacional e o desempenho empresarial.

Este trabalho está estruturado em cinco capítulos e anexos:

- Capítulo 1, apresenta o contexto geral do projeto, como também a metodologia e a motivação para a pesquisa.
- Capítulo 2, procura discutir pesquisas e trabalhos anteriores relacionados com o tema.
- Capítulo 3, é apresentada uma visão sobre o sistema elaborado, desde a fase de identificação de requisitos até á implementação dos mesmos.

- Capítulo 4, descreve o desempenho do sistema desenvolvido e os casos onde foi empregue.
- Capítulo 5, os resultados obtidos e as principais dificuldades examinadas e como foram superadas.
- Capítulo 6, sintetiza os principais resultados do projeto e discute as hipóteses de desenvolvimentos futuros.

2. Trabalho Relacionado

Nos dias de hoje, o desenvolvimento e a gestão eficaz da formação dos colaboradores são cruciais para o sucesso e competitividade das empresas. A formação contínua não só melhora as competências dos colaborares, como também otimiza o desempenho da organização. Neste contexto, os sistemas de recomendação surgem como ferramentas válidas para auxiliar na definição e planeamento de percursos formativos personalizados e adequados às necessidades individuais de cada colaborador.

Os sistemas de recomendação utilizam técnicas de AI e análise de dados para identificar as melhores oportunidades de formação para cada colaborador, com base em diversos fatores como, histórico de formação, competências e conhecimentos. Ao analisar os dados anteriores, os sistemas de recomendação podem gerar recomendações personalizadas de formação que podem promover o envolvimento e motivação que contribuem para o desenvolvimento profissional contínuo dos colaboradores.

Este capítulo apresenta uma análise do trabalho relacionado com os sistemas de recomendação, com o objetivo de definir uma base teórica e contextual para a sua implementação em empresas. A revisão de literatura abrange estudos de caso que implementaram algum tipo de sistema de recomendação, com o objetivo de identificar as melhores práticas e os resultados obtidos. A conclusão do capítulo sintetiza as informações relevantes obtidas e estabelece a importância destes para o desenvolvimento e avaliação de um sistema de recomendação.

2.1. Contexto

2.1.1. Dimensão Histórica

A evolução dos sistemas de recomendação pode ser dividida em três fases principais: as origens, a expansão comercial e a revolução da inteligência artificial.

O estudo dos sistemas de recomendação começou em 1992, com *Belkin* e *Croft* separando a filtragem de informação da recuperação de informação. O *Tapestry*, *de Goldberg et al*., foi o primeiro sistema a usar filtragem colaborativa, baseado em avaliações humanas. Esse período também viu o surgimento do *GroupLens*, desenvolvido por pesquisadores do MIT e da UMN, que marca o início do uso da filtragem colaborativa utilizador-utilizador. O laboratório *GroupLens* foi pioneiro no estudo de sistemas de recomendação, e lançou o *MovieLens* em 1997, tornando-se um recurso importante para pesquisas futuras.

Com o crescimento do comércio tecnológico, empresas como *Net Perceptions* (1996) começaram a comercializar motores de recomendação, com clientes como *Amazon* e *Best Buy*. Durante esse período, surgiram modelos colaborativos como a filtragem utilizador-utilizador e item-item, que dominaram até 2005. O *Netflix Prize* (2006-2009) impulsionou o desenvolvimento de modelos de factoração e matriz, que se tornaram amplamente utilizados. A partir de 2007, a conferência ACM *RecSys* consolidou a importância acadêmica do campo. Modelos como *Factorization Machines* (2010) ajudaram a refinar as interações entre recursos nos sistemas de recomendação.

A partir de 2016, a inteligência artificial transformou os sistemas de recomendação, com modelos como *Wide&Deep*, *DeepFM* e *YouTubeDNN* sendo adotados pela indústria. Os modelos anteriores permitiram captar interações complexas de utilizadores e as suas

preferências ao longo do tempo. Questões éticas, como privacidade, explicabilidade e justiça, tornaram-se centrais nos debates atuais sobre o futuro dos sistemas de recomendação.

Todas as informações anteriores, foram retiradas do artigo "*A Brief History of Recommender Systems*" por *Zhenhua Dong et al*. [7].

2.1.2. Evolução tecnológica

Os primeiros sistemas de recomendação eram relativamente simples, utilizavam técnicas como a filtragem colaborativa para efetuar recomendações. No entanto, debatiam-se com o problema do "arranque a frio", em que o sistema não conseguia fazer recomendações precisas para novos utilizadores ou itens porque não tinha dados suficientes. Também existia a dificuldade em lidar com elevada dimensionalidade e dispersão dos dados, bem como com os problemas de escalabilidade à medida que o número de utilizadores e itens aumentava.

Com a evolução da inteligência artificial, os sistemas de recomendação tornaram-se mais sofisticados. Os sistemas modernos podem utilizar algoritmos para analisar grandes quantidades de dados, e assim aprendem o comportamento do utilizador para recomendar soluções personalizadas para o mesmo. Os modelos de *Machine Learning (ML),* podem capturar padrões e relações complexas nos dados, levando a recomendações mais precisas e personalizadas. Por exemplo, podem ter em conta o contexto do comportamento do utilizador, como a hora do dia ou a localização do utilizador, para fazer recomendações relevantes.

Outro desenvolvimento é a utilização de sistemas híbridos, que combinam diferentes métodos para ultrapassar certos problemas. Por exemplo, um sistema pode utilizar filtragem colaborativa para fazer recomendações baseadas no comportamento do utilizador e a filtragem baseada no conteúdo para fazer recomendações baseadas nos atributos do item.

Em conclusão, os sistemas de recomendação percorreram um longo caminho desde a sua criação. Desde sistemas simples, que tinham problemas como a falta de dados, para sistemas elaborados que utilizam *ML* para realizar recomendações.

Todas as informações e afirmações anteriores foram retiradas do Web Site "101.school" do módulo "*History and Evolution of Recommender Systems*" [8].

2.2. Conceitos

Nesta secção, são apresentados os conceitos essenciais relacionados com o sistema de recomendação personalizado, como a definição e o a utilização de um sistema de recomendação, os tipos de abordagens ou algoritmos para um sistema de recomendação e a definição de ML.

2.2.1. Sistema de Recomendação

Um sistema de recomendação é um algoritmo de AI, normalmente associado a ML, que utiliza *Big Data* para sugerir ou recomendar produtos adicionais aos consumidores. Estes podem basear-se em vários critérios, incluindo compras anteriores, histórico de pesquisa, informações demográficas e outros fatores. Os sistemas de recomendação são muito úteis, pois ajudam os utilizadores a descobrir produtos e serviços que, de outra forma, não teriam encontrado por si próprios [9].

Os sistemas de recomendação são treinados para compreender as preferências, as decisões anteriores e as características das pessoas e dos produtos, utilizando dados recolhidos

sobre as suas interações. Neste caso de estudo, o sistema de recomendação vai focar-se nas competências, na função atual e na experiência do utilizador. Este vai recomendar cursos ou percursos formativos mais indicados para aquele utilizador específico.

2.2.2. Desafios de um Sistema de Recomendação

Os sistemas de recomendação tornaram-se essenciais nas interações diárias com produtos, serviços e conteúdos, mas a criação de sistemas eficazes pode apresentar vários desafios. Segue-se um resumo dos principais desafios e potenciais soluções:

- Dados espalhados - Os utilizadores interagem apenas com uma pequena fração dos itens disponíveis, o que dificulta a recomendação de novos conteúdos.
- "Cold start" - Quando novos utilizadores ou itens não dispõem de dados suficientes, a filtragem baseada no conteúdo e os métodos híbridos que combinam a filtragem colaborativa com abordagens baseadas no conhecimento são eficazes.
- Escalabilidade - Os grandes conjuntos de dados constituem um desafio para o desempenho em tempo real dos sistemas de recomendação.
- "Overfitting" e "Underfitting" - O Overfitting (ou "sobre ajuste" em português) ocorre quando um modelo é demasiado adaptado aos dados de treino, enquanto o Underfitting (ou "suba juste" em português) conduz a resultados demasiado simples.
- Diversidade - Os sistemas de recomendação concentram-se frequentemente em itens populares, faltando-lhes diversidade. Métricas como entropia e recomendações baseadas em serendipidade ajudam a promover conteúdo inesperado, mas relevante.
- Privacidade - Os sistemas requerem frequentemente dados pessoais dos utilizadores, o que acarreta riscos de privacidade.
- Viés e riscos legais - Os sistemas podem desenvolver preconceitos, como dar prioridade a conteúdos populares ou reforçar comportamentos prejudiciais. Estes preconceitos podem levar a desafios legais, como demonstrado por ações judiciais que ligam os sistemas de recomendação a conteúdos nocivos.

A resolução destes desafios através de algoritmos complexos e estruturas éticas criará sistemas de recomendação mais exatos, justos e fiáveis.

Todos as informações anteriores, é um compilado de vários Web Sites acerca do tema [4] [5] [6].

2.2.3. Tipos de abordagens para um sistema de recomendação

Embora exista um vasto número de algoritmos e técnicas de recomendação, a maior parte deles enquadra-se nas seguintes categorias: Filtragem Colaborativa (*Collaborative Filtering*); Filtragem de Conteúdos (*Context Filtering*); e Filtragem de Contexto (*Context Filtering*) [9].

Na Filtragem Colaborativa, o algoritmo de recomendação recomenda itens (esta é a parte de filtragem) com base em informações de preferência de muitos utilizadores (esta é a parte colaborativa). Esta abordagem utiliza a semelhança do comportamento das preferências do utilizador, tendo em conta as interações anteriores entre os utilizadores e os itens, os algoritmos de recomendação aprendem a prever a interação futura. Estes sistemas de recomendação constroem um modelo a partir do comportamento passado de um utilizador, como os artigos comprados anteriormente ou as classificações atribuídas a esses artigos e

decisões semelhantes de outros utilizadores. A ideia é que, se algumas pessoas tomaram decisões e fizeram compras semelhantes no passado, como a escolha de um filme, então há uma grande probabilidade de concordarem com outras seleções futuras. Por exemplo, se um recomendador de filtragem colaborativa souber que você e outro utilizador partilham gostos semelhantes em termos de filmes, poderá recomendar-lhe um filme que sabe que esse outro utilizador já gosta.

Em contrapartida, a abordagem de conteúdo (Filtragem de Conteúdos) utiliza os atributos ou características de um item (esta é a parte do conteúdo) para recomendar outros itens semelhantes às preferências do utilizador. Esta abordagem baseia-se na semelhança das características do item e do utilizador, tendo em conta as informações sobre um utilizador e os itens com que interagiu (por exemplo, a idade de um utilizador, a categoria da cozinha de um restaurante, a crítica média de um filme), modelar a probabilidade de uma nova interação.

Por exemplo, se um recomendador de filtragem de conteúdos vir que um utilizador gostou dos filmes "*You've Got Mail*" e "*Sleepless in Seattle*", poderá recomendar outro filme com os mesmos géneros e/ou elenco, como "*Joe Versus the Volcano*" ao utilizador.

E por último, a Filtragem por Contexto, inclui as informações contextuais dos utilizadores no processo de recomendação. Esta abordagem utiliza uma sequência de ações contextuais do utilizador, mais o contexto atual, para prever a probabilidade da ação seguinte.
Embora existam várias abordagens e técnicas de recomendação, a mais importante e a que possivelmente vai ser utilizada é a Filtragem Colaborativa.

2.2.4. Machine Learning

ML é um subcampo de AI que se concentra no uso de dados e algoritmos para permitir que a AI imite a maneira como os humanos aprendem, melhorando gradualmente a sua precisão [10].

Em termos simples, a ML permite que os computadores aprendam a partir de dados sem serem explicitamente programados para isso. Isso significa que os algoritmos de ML podem identificar padrões, sem a necessidade de intervenção humana.

Existem diferentes tipos de ML, cada um com os seus próprios pontos fortes e fracos. Alguns dos tipos mais comuns incluem os seguintes.

- Aprendizagem supervisionada (*Supervised learning*): Neste tipo de ML, os algoritmos são treinados com um conjunto de dados previamente rotulados. Por exemplo, um conjunto de gatos e cães pode ser usado para treinar um algoritmo de ML a identificar novos gatos e cães em imagens.
- Aprendizagem não supervisionada (*Unservised learning*): Os algoritmos não recebem um conjunto de dados previamente rotulados. Em vez disso, eles devem identificar padrões e estruturas nos dados por conta própria. Por exemplo, um algoritmo de ML não supervisionado pode ser usado para agrupar clientes em um conjunto de dados de acordo com seus hábitos de compra.
- Aprendizagem por reforço (*Reinforcement learning*): Os algoritmos aprendem por meio de tentativa e erro. Eles interagem com um ambiente e recebem recompensas ou penalidades por suas ações. O objetivo do algoritmo é aprender a tomar ações que maximizem a sua recompensa.

2.2.4.1. Algoritmos de ML

Como já referido anteriormente, existem 3 conjuntos de algoritmos [11]:

- **Supervisionado;**
  - **Classificação**: utiliza um algoritmo para atribuir com precisão dados de teste a categorias especificas. Reconhece entidades específicas no conjunto de dados e tenta tirar algumas conclusões sobre a forma como essas entidades devem ser rotuladas ou definidas. Os algoritmos de classificação mais comuns são *linear classifiers*, *SVM* (*suport vector machines*), *decision trees*, *K-nearest neighbor* e *random forest*.
  - **Regressão:** é usada para compreender a relação entre variáveis dependentes e independentes. Alguns algoritmos de regressão são a regressão linear, a regressão logística e a regressão polinomial. Para ver mais detalhadamente os algoritmos anteriores, consultar o *Anexo I.*
- **Não supervisionado:**
  - *Clustering* que pode identificar padrões nos dados para que estes possam ser agrupados. Estes algoritmos podem ajudar a identificar diferenças entre itens de dados que os humanos não tenham detetado.
  - *Hierarchical clustering* agrupa os dados numa árvore de clusters. Este método começa por tratar cada ponto de dados como um cluster separado. De seguida, executa repetidamente os seguintes passos: Identificar os dois clusters que podem estar mais próximos; fundir os dois clusters máximos comparáveis. Estes passos continuam até que todos os clusters estejam unidos.
  - *K-means clustering* identifica grupos dentro dos dados sem *labels* (rótulos) em diferentes grupos, encontrando grupos de dados semelhantes entre si. O nome "*K-means*" provém dos centroides K que utiliza para definir os agrupamentos. Um ponto é atribuído a um determinado agrupamento se estiver mais próximo do centroide desse agrupamento do que de qualquer outro centroide.
- **Por reforço:**
  - Os algoritmos por reforço são treinados tal como os humanos aprendem através de recompensas e penalizações, que são medidas e seguidas por um agente de aprendizagem por reforço, que tem um conhecimento geral da probabilidade de fazer subir ou descer a pontuação. Através de tentativa e erro, o agente aprende a tomar medidas que conduzem aos resultados mais favoráveis ao longo do tempo. A aprendizagem por reforço é frequentemente utilizada na gestão de recursos, na robótica e nos jogos de vídeo.

Além destes algoritmos, também pode haver um algoritmo que resulta da junção do algoritmo supervisionado com o não supervisionado, chamado semi-supervisionado. Neste caso, a aprendizagem ocorre quando apenas uma parte dos dados de entrada fornecidos são rotulados. Esta abordagem pode combinar o melhor de dois mundos, como a maior precisão associada à aprendizagem supervisionada e a capacidade de utilizar dados não rotulados com uma boa relação custo-eficácia, como no caso da aprendizagem não supervisionada.

2.2.5. Dataset

Um *dataset* é um conjunto de dados utilizados para análise de modelos e normalmente organizados num formato estruturado. Esse formato estruturado pode ser em Excel, um ficheiro CSV, em JSON ou outros formatos. Os dados podem ser organizados de várias formas e criados a partir de uma grande variedade de fontes, como uma sondagem de clientes, uma experiência ou uma base de dados existente. Este pode ser utilizado para muitos fins, incluindo a criação e teste de modelo de ML, a visualização de dados, a investigação ou a análise estatística [12].

2.2.6. Endpoint

Um *endpoint* é um dispositivo físico (ou digital) que conecta a uma rede para trocar informações. Exemplos de *endpoints* incluem dispositivos móveis, máquinas virtuais e servidores. Quando um dispositivo conecta a uma rede, o fluxo de informações entre, por exemplo, um computador e a rede pode ser comparado a uma conversa por telefone entre duas pessoas [13]. Neste contexto, o *endpoint* é utilizado para obter previsões e explicações, com os modelos já em funcionamento [14].

2.3. Revisão de literatura

Para a condução desta revisão de literatura, será utilizada uma metodologia sistemática baseada nas diretrizes do PRISMA (*Preferred Reporting Items for Systematic Reviews and Meta-Analyses*) [15].

A aplicação do método PRISMA inclui uma análise rigorosa e transparente, envolvendo a identificação, triagem, elegibilidade e inclusão dos estudos. Cada etapa do processo será documentada em um fluxograma PRISMA, garantindo a rastreabilidade das decisões tomadas. A análise dos dados extraídos dos estudos incluídos permitirá sintetizar as principais descobertas e identificar lacunas na literatura existente, proporcionando uma base sólida para futuras pesquisas.

2.4. Metodologia de pesquisa

A seleção dos artigos será realizada nas bases de dados *IEEE Xplore* [16], *Web of Science Core Collection* (*WoSCC*) [17] e *Scopus* [18], reconhecidas por sua relevância e abrangência na literatura científica, especialmente nas áreas de engenharia.

Inicialmente, será definida uma estratégia de busca específica para cada base de dados, utilizando palavras-chave que reflita, os tópicos de interesse da revisão. Após a realização de buscas, todos os resultados serão importados para o *Zotero* [19], um gestor de referências bibliográficas. Em seguida, os títulos e resumos dos artigos serão avaliados quanto à sua relevância, e aqueles que atenderem aos critérios de inclusão serão selecionados para leitura completa.

Os termos utilizados na busca foram definidos considerando palavras-chave relacionadas a três dimensões. Na dimensão “Tecnologia”, destacam-se a Inteligência Artificial (IA) e o ML, tecnologias essenciais para o desenvolvimento de sistemas avançados de recomendação inteligente no contexto educativo. Na dimensão “Propósito”, os termos de

busca concentram-se em educação (*Education*) e desenvolvimento profissional (*Professional Development*), com o objetivo de otimizar a definição e o planeamento de percursos formativos adequados ao perfil dos colaboradores, melhorando assim a eficácia dos programas de formação. Já na dimensão "Âmbito", a pesquisa está centrada no contexto de competências (*Competence*) ou nas capacidades (*Capabilities*) e nas recomendações (*Recomendation*).

Outras palavras-chave foram utilizadas e testadas na pesquisa, mas os resultados não foram promissores como os escolhidos. Entre as palavras-chave rejeitadas estão *skills* e *career development*, que foram substituídas por sinónimos como *competence* e *professional development*, respetivamente, devido à maior frequência dessas palavras nos artigos.

A pesquisa abrange apenas publicações em português ou inglês, com um recorte temporal entre 2022 e março de 2024. Este filtro garante a relevância e a atualidade das fontes utilizadas. A Tabela *1*, sumariza os termos que foram usados na pesquisa bem como os filtros aplicados.

Tabela 1 - Termos pesquisados e filtros aplicados

Tecnologia	Propósito	Âmbito	Filtros
AI – Artificial intelligence *ML - Machine Learning*	*Education* *Professional Development*	*Recommendation* *Competence* *Capabilities*	Apenas publicados em português ou inglês, entre 2022 e 2024.

Com base nas três dimensões e termos identificados, foi criada uma *string* de pesquisa que foi usada nas bases de dados de conhecimento mencionadas anteriormente.

((ai OR ml OR "machine learning" OR "artificial intelligence") AND (education OR "professional development") AND (recommendation AND (competence OR capabilities)))

Inicialmente, foram selecionadas publicações de acordo com a metodologia PRISMA [15], utilizando a análise dos títulos e resumos para determinar sua relevância e adequação ao tema em estudo. Em seguida, verificou-se se os artigos completos estavam disponíveis para consulta, priorizando aqueles que permitiam uma análise detalhada.

Além dos artigos científicos encontrados em bases de dados académicos, a revisão da literatura também considerou outras fontes de conhecimento, como recursos online pertinentes e citações de especialistas na área. Após essa fase, procedeu-se à leitura completa dos documentos selecionados para avaliar se atendiam aos critérios de inclusão estabelecidos.

2.5. Resultados

Seguindo a metodologia PRISMA [15] e as premissas estabelecidas para as palavras-chave, o ano de publicação e o idioma, foi efetuada uma pesquisa nas bases de dados selecionadas.

No começo, foram assinalados 229 registos que cumprem os critérios de inclusão estabelecidos. Estes foram sujeitos a uma análise preliminar para analisar a sua relevância e adequação à revisão sistemática em questão.

Após a identificação dos artigos que contêm os termos relacionados às três dimensões analisadas, de acordo com a metodologia descrita no PRISMA, verificou-se que 85 deles eram duplicados e, portanto, foram excluídos do conjunto de estudos.

Posteriormente, os títulos e resumos dos 144 registos restantes foram examinados. Neste processo de triagem, 134 registos não foram considerados relevantes para o objeto de estudo, convergindo em 10.

Foi conduzida uma pesquisa adicional para verificar a acessibilidade dos estudos escolhidos. Nesse processo, 5 publicações não estavam disponíveis para leitura integral e, por esse motivo, foram excluídas da análise. Isso deixou 5 publicações para uma avaliação mais aprofundada.

Após a triagem e seleção dos registos, restaram 5 estudos ao final do processo. Essas publicações compõem o conjunto final de estudos que foram analisados e discutidos na revisão de literatura, com o intuito de fornecer uma base sólida para entender o estado atual do conhecimento sobre o tema.

A *Figura 1* mostra o fluxograma do processo metodológico utilizado na revisão sistemática, conforme as diretrizes da metodologia PRISMA [15]. Este fluxograma detalha as etapas desde a identificação inicial dos estudos até a seleção final dos artigos que foram incluídos na revisão.

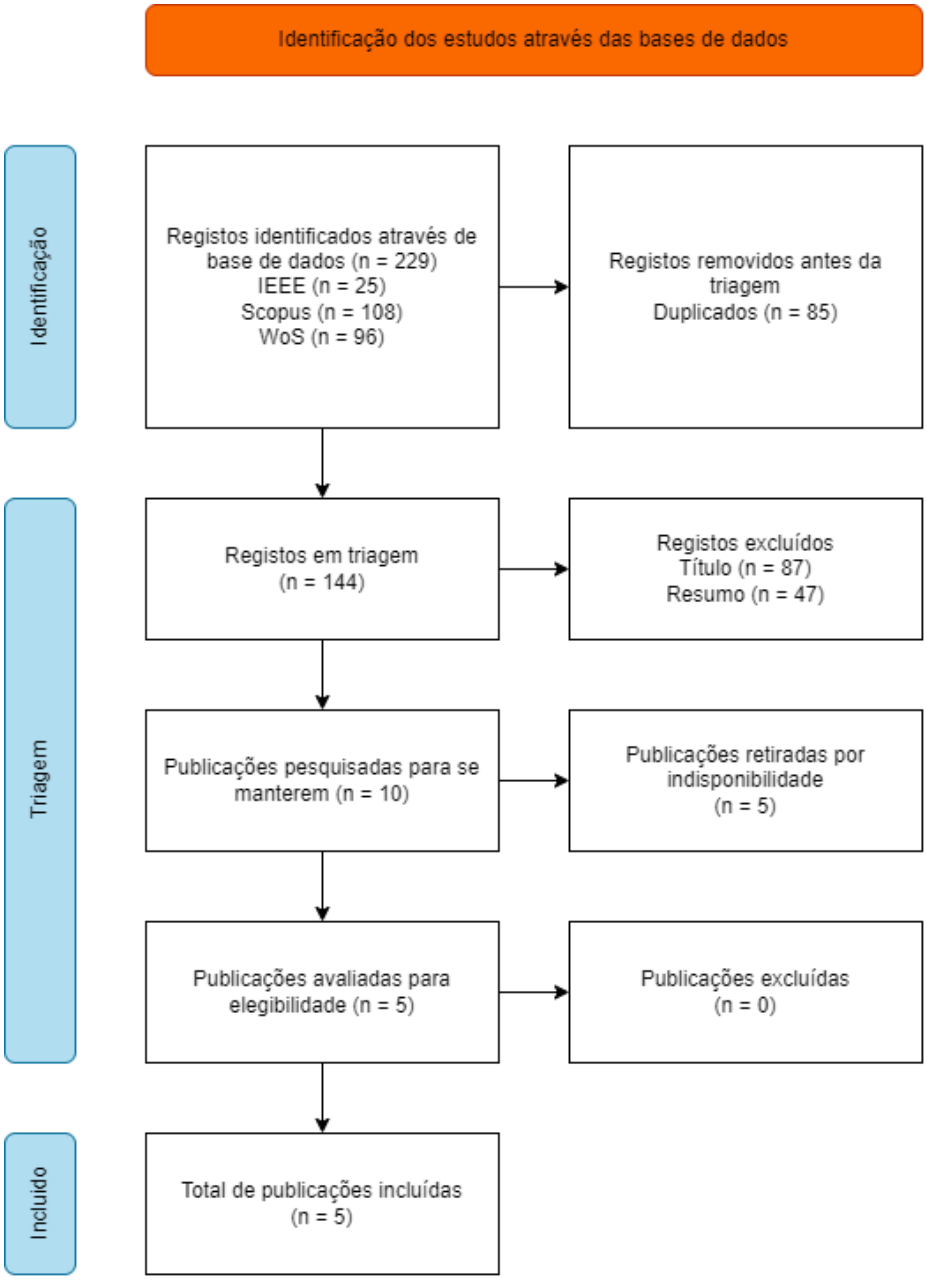

Figura 1 - Fluxograma PRISMA para revisão sistemática

2.6. Discussão

Nesta secção, é apresentada uma análise resumida dos artigos identificados pela metodologia PRISMA, com o objetivo de explorar e interpretar os principais resultados obtidos na revisão sistemática realizada.

Os sistemas de recomendação têm se tornado uma ferramenta valiosa em diversas áreas, incluindo no mercado de trabalho e na educação/formação. Esta secção visa a examinar diferentes abordagens de sistemas de recomendação voltados para a correspondência de carreiras, focando em vários intervenientes.

Em 2022, *Guyard* e *Deriaz* [20] propuseram um sistema de recomendação híbrido que combina filtragem baseada no conteúdo e indexação por semelhança para ligar trabalhadores seniores a empresas. O sistema analisa as competências e experiência dos trabalhadores, fazendo a correspondência com os requisitos das empresas. Isto beneficia tanto os seniores, permitindo-lhes continuar ativos, quanto às empresas, ao identificar candidatos qualificados.
Mais ligado a universidades, em 2023, o *Hang et al*. [21], desenvolveu um sistema de recomendação voltado para recém-licenciados de engenharia. Este sistema propõe corresponder as competências desenvolvidas nas instituições de ensino às profissões mais indicadas. Utiliza algoritmos de ML para fazer recomendações de carreira, recolhendo informações detalhadas dos utilizadores, incluindo dados pessoais (nome, idade, …), histórico pedagógico e profissional, competências técnicas e sociais, e informações retiradas das redes sociais (*LinkedIn, Instagram*, …). A utilização de NLP (*Natural Language Processing*) baseada em *Transformers*, mais especificamente com a utilização do modelo BERT (*Bidirectional Encoder Representations from Transformers*), foi uma abordagem empregada para analisar os dados e fornecer recomendações precisas.

Em 2024, o *Deshpande et al*. [22], desenvolveu um sistema de recomendação, que com a ajuda de um *chatbot*, impulsionado por NLP e algoritmos de ML, como *Decision Tree* e *Clustering*, para fornecer aconselhamento de carreira personalizado focado em estudantes. Através de avaliações de aptidão, personalidade e interesse, o sistema identifica os pontos fortes e fracos dos alunos, gerando recomendações de carreira alinhadas com as suas capacidades e aspirações. O sistema visa a alcançar estudantes em áreas desfavorecidas, capacitando-os a tomar decisões informadas sobre as suas carreiras.

E por fim, também em 2024, o *Gedrimiene et al*. [23], desenvolveu um estudo onde explora as perceções dos utilizadores sobre a transparência e confiabilidade de uma ferramenta de análise para fins de orientação profissional. A pesquisa realizada descobriu que a precisão das recomendações é mais influente na intenção de seguir as recomendações do que a compreensão da origem dessas recomendações. A idade dos utilizadores também afeta a avaliação da transparência. O estudo destaca a importância da precisão e sugere que a confiança dos utilizadores pode ser melhorada com maior transparência das ferramentas de IA.

A revisão da literatura revela uma diversidade de abordagens e sistemas de recomendação para correspondência de carreiras e personalização de educação. Embora não existam artigos específicos que discutam a personalização de formação, neste contexto, podemos utilizar os conceitos e as metodologias apresentados no trabalho de *Irsalireva et al*. [24], para desenvolver um sistema de recomendação personalizado que atende às necessidades de diferentes grupos de utilizadores.

O artigo anterior, aborda a importância progressiva da personalização da educação no ensino superior no Cazaquistão, realçando a necessidade de melhorar os sistemas de informação para programas educativos. Este artigo fala sobre um modelo de educação

personalizado baseado na matriz de competências do aluno, tem como objetivo ajudar os alunos a escolher cursos que caracterizam os seus interesses e o seu nível de conhecimento, usando técnicas baseadas no conhecimento do aluno, no conteúdo do curso e na filtragem colaborativa. O artigo foca no desenvolvimento de percursos educativos individuais que têm em conta as necessidades e os comportamentos dos alunos, utilizando modelos ontológicos para estruturar conteúdos e avaliar o progresso automaticamente. Encerra-se que a personalização da educação, suportada por técnicas de IA e de análise de dados, pode aperfeiçoar a flexibilidade e a eficácia do ensino, ajudando os alunos a estarem mais preparados para o mercado de trabalho e a permitir uma educação mais intensa e relevante. Em seguida, está uma tabela que sumariza toda a informação anterior.

Tabela 2 - Discussão - Sumarizada

Ano	Autores	Objetivo	Metodologia/ Algoritmos Utilizados	Grupo-alvo	Principais Conclusões
2022	Guyard & Driaz	Recomendação de emprego para trabalhadores seniores	Filtragem baseada em conteúdo e indexação por semelhança	Trabalhadores seniores	Facilita a correspondência entre trabalhadores seniores e empresas, permitindo que continuem ativos e ajuda empresas a encontrar profissionais experientes
2023	Hang et al.	Recomendação de carreiras para recém-licenciados	ML e NLP, análise de redes sociais	Recém-licenciados em engenharia	Propõe recomendações com base nas competências adquiridas na universidade e em dados pessoais, melhorando assim a transição para o mercado de trabalho
2024	Deshpande et al.	Aconselhamento de carreira personalizada	NLP, ML (Decision Tree, Clustering), Chatbot	Estudantes	Utiliza avaliações de aptidão, personalidade e interesse para recomendar carreiras, com foco em estudantes de áreas desfavorecidas
2024	Gedrimiene et al.	Estudo sobre a transparência e confiabilidade de sistemas de recomendação de carreiras	Análise de perceções dos utilizadores	Diversos utilizadores	A precisão das recomendações influencia mais a adesão do que a compreensão da origem dos resultados; a idade dos utilizadores afeta a perceção de transparência
2024	Irsalireva et al.	Personalização da educação no ensino superior	Ontologias, Filtragem colaborativa, Modelos baseados non conhecimento do aluno	Estudantes universitários	Propõe um sistema que estrutura percursos educativos individuais, melhorando a adequação dos cursos às necessidades dos alunos e aumenta a sua preparação para o mercado de trabalho

Em suma, os artigos analisados demonstram que o objetivo deste projeto ainda não foi explorado. Embora existam abordagens semelhantes voltadas para diferentes grupos-alvo e

utilizando diversas técnicas, nenhuma delas corresponde exatamente à proposta deste trabalho.

2.7. Conclusão

Este capítulo examinou o estado da arte dos sistemas de recomendação, com ênfase na maneira como podem ser usados para gerir o desenvolvimento e a formação de colaboradores em uma empresa. A análise abordou os conceitos essenciais e métodos de recomendação como filtragem colaborativa, filtragem de conteúdo e filtragem por contexto. Além disso, foi abordado o papel do ML e as diferentes abordagens, enfatizando como essas tecnologias podem ser usadas para criar recomendações eficazes e personalizadas.

A revisão de literatura, conduzida segundo a metodologia PRISMA, permitiu identificar e avaliar estudos relevantes que implementaram sistemas de recomendação em diversos contextos. Os estudos analisados demonstram a eficácia dos sistemas de recomendação em áreas como correspondência de carreiras e personalização da educação. Exemplos notáveis incluem o trabalho de *Guyard* e *Deriaz*, que propuseram um sistema híbrido para conectar trabalhadores seniores a empresas, e o estudo de *Hang et al*., que desenvolveu um sistema de recomendação para recém-licenciados em engenharia, utilizando técnicas de ML e NLP. Diante da ausência de uma solução completa para o problema, torna-se imprescindível o desenvolvimento de uma nova abordagem para resolvê-lo.

Em síntese, a análise realizada neste capítulo fornece uma base teórica e contextual sólida para a implementação de sistemas de recomendação em empresas. A aplicação de técnicas de AI e ML para a personalização da formação pode não apenas melhorar as competências dos colaboradores, mas também otimizar o desempenho de uma organização, promovendo um ambiente de aprendizagem contínuo e adaptativo. A utilização dessas tecnologias avançadas representa um passo significativo em direção à inovação na gestão de formação, capacitando as empresas a serem mais competitivas e eficazes no desenvolvimento a nível profissional do colaborador.

3. Metodologia de Investigação

A metodologia *Design Science Research Methodology* (DSMR), foi usado para o desenvolvimento deste projeto, esta abordagem combina a teoria e a prática para resolver problemas específicos através da criação e avaliação de artefactos. A DSRM [25] é caracterizada por um ciclo iterativo que começa com a identificação do problema, passa pela conceção e construção de uma solução, e culmina na avaliação dessa solução no contexto real.

Como mostrado na *Figura 2 (Traduzido de Peffers et al. (2007))*, a DSRM consiste em diversas etapas interconectadas, que orientam o pesquisador desde a identificação do problema até a avaliação e aprimoramento da solução sugerida.

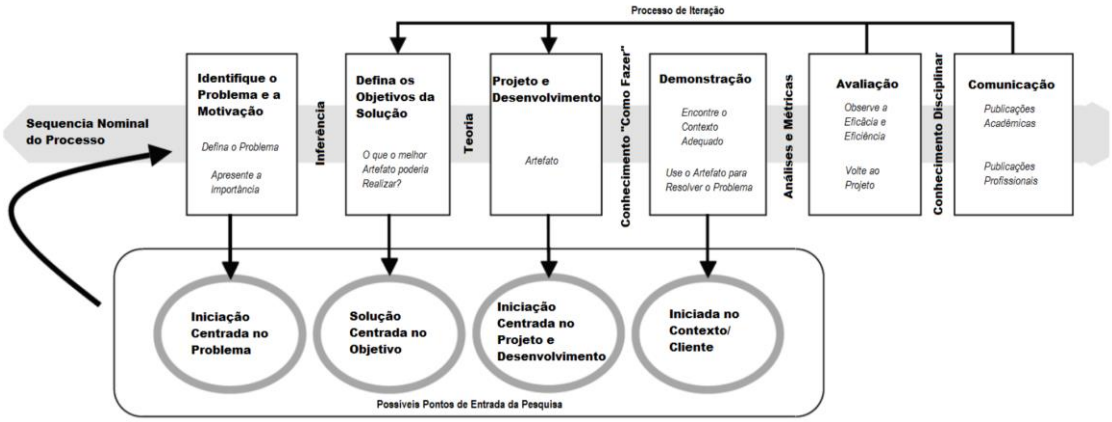

Figura 2 - Modelo de processo DSRM

Na metodologia mencionada, há várias abordagens possíveis, dependendo do objetivo e contexto do problema. Para este trabalho, escolheu-se a abordagem "Iniciação centrada no problema", que prioriza a identificação e compreensão detalhada do problema antes de prosseguir para as fases de design e implementação.

Este projeto foi planeado e desenvolvido conforme as etapas da DSRM, sendo executado em três iterações distintas. Cada iteração contribuiu para o aprimoramento do sistema. Os detalhes dessas iterações e das etapas específicas da DSRM seguidas estão descritos nos Capítulos 3 e 4.

Para garantir que o desenvolvimento do projeto responde às necessidades identificadas (3.1), é fundamental avaliar o grau de cumprimento dos requisitos estabelecidos na primeira etapa do DSRM. A metodologia escolhida para esta avaliação é a escala NLPF da ISO 15504 [26], que foi substituída pela ISO 33002 [27], que se divide em quatro graus são os seguintes [28].

- Não Atingido (N) – O requisito não foi cumprido ou foi atendido de forma muito limitada;
- Parcialmente Atingido (P) – O requisito foi em parte cumprido, mas ainda existem bastantes lacunas ou áreas que necessitam de melhorias;
- Largamente Atingido (L) – O requisito foi amplamente cumprido, com algumas áreas que precisam de melhoria;
- Totalmente Atingido (F) – O requisito foi completamente cumprido, sem lacunas identificadas.

Esta escala proporciona uma análise qualitativa precisa, permitindo uma visão clara de quais requisitos foram plenamente satisfeitos e quais ainda necessitam de ajustes ou melhorias.

O sistema de recomendação tem como objetivo principal fornecer recomendações personalizadas de conteúdos ou produtos a utilizadores. Como resultado, este sistema de recomendação foi criado para ser adicionado como um novo módulo da plataforma gtrainiG, do qual muitas empresas apresentavam essa necessidade. O sistema de recomendação é descrito em detalhes neste capítulo, começando com a identificação dos objetivos e dos requisitos necessários nas secções 3.1, 3.2 e 3.3 respetivamente, a definição da arquitetura, com a recolha e tratamento de dados, além da apresentação de resultados, na secção 3.4, descrição do sistema na secção 3.5 e finalmente as tecnologias utilizadas na secção 3.6.

3.1. Problema e Motivação

A *FPTIC Consulting* é uma software *house* portuguesa com sede na Nazaré, fundada em 2006. A mesma certificada pela DGERT para formação presencial e a distância, presta serviços de formação, consultoria informática e proprietária de um software denominado gtrainiG.co, este último é dedicado a gestão de formação. A FPTIC atua em Portugal e em alguns países da Europa e Africa, possui uma equipa de profissionais experientes e qualificados.

A *FPTIC Consulting* tem vários clientes que podem beneficiar deste sistema, os dados que foram utilizados para esta fase foi dos clientes da FPTIC.

Ao longo dos últimos anos, principalmente após COVID, as entidades ficaram mais disponíveis para aceitar a implementação de tecnologias nas suas empresas.

Alguns dos principais problemas apresentados pelos clientes foram os seguintes.

- Não ter qualquer informação sobre quais os colaborares que devem receber formação com base na profissão / função.
- Muito dificuldade em identificar e recomendar quais os percursos formativos para os colaboradores.
- Incapacidade de prever quem terá capacidade de exercer determinadas funções em casos de saídas ou substituições temporárias.

Estes são os principais problemas apresentados por parte dos clientes e que vão ser abordados nesta tese.

3.2. Objetivo

O objetivo deste projeto de mestrado é desenvolver um sistema de recomendação que apoie os recursos humanos na gestão do desenvolvimento das competências, previsões de progressão de carreira e identificação de possíveis saídas de colaboradores, promovendo a retenção, além de recomendar formações. Para ajudar no desenvolvimento das competências, o sistema deve com base na profissão atual ou futura, verificar se o colaborador contém as competências necessárias para a profissão atual ou futura, se este não possuir, o sistema vai recomendar formações para que o colaborador possa adquirir estas competências. Na previsão de progressão de carreira, o sistema deve analisar dados, relativos ao colaborador, como o histórico profissional (profissão atual e anteriores), idade em que começou o contrato, entidade empresarial e a localidade, para identificar os padrões de progressão de carreira dos colaboradores, ou seja, com base nos dados atuais do colaborador encontrar a sua possível futura profissão. E por fim, a identificação de possíveis saídas de colaboradores, o sistema deve analisar o tempo em que um colaborador está numa função para antecipar possíveis saídas dos mesmos, isto permite à organização planear de forma mais eficaz as ações de retenção ou substituição, relativas aos colaboradores.

A *Figura 3* é a demonstração dos objetivos explicados anteriormente.

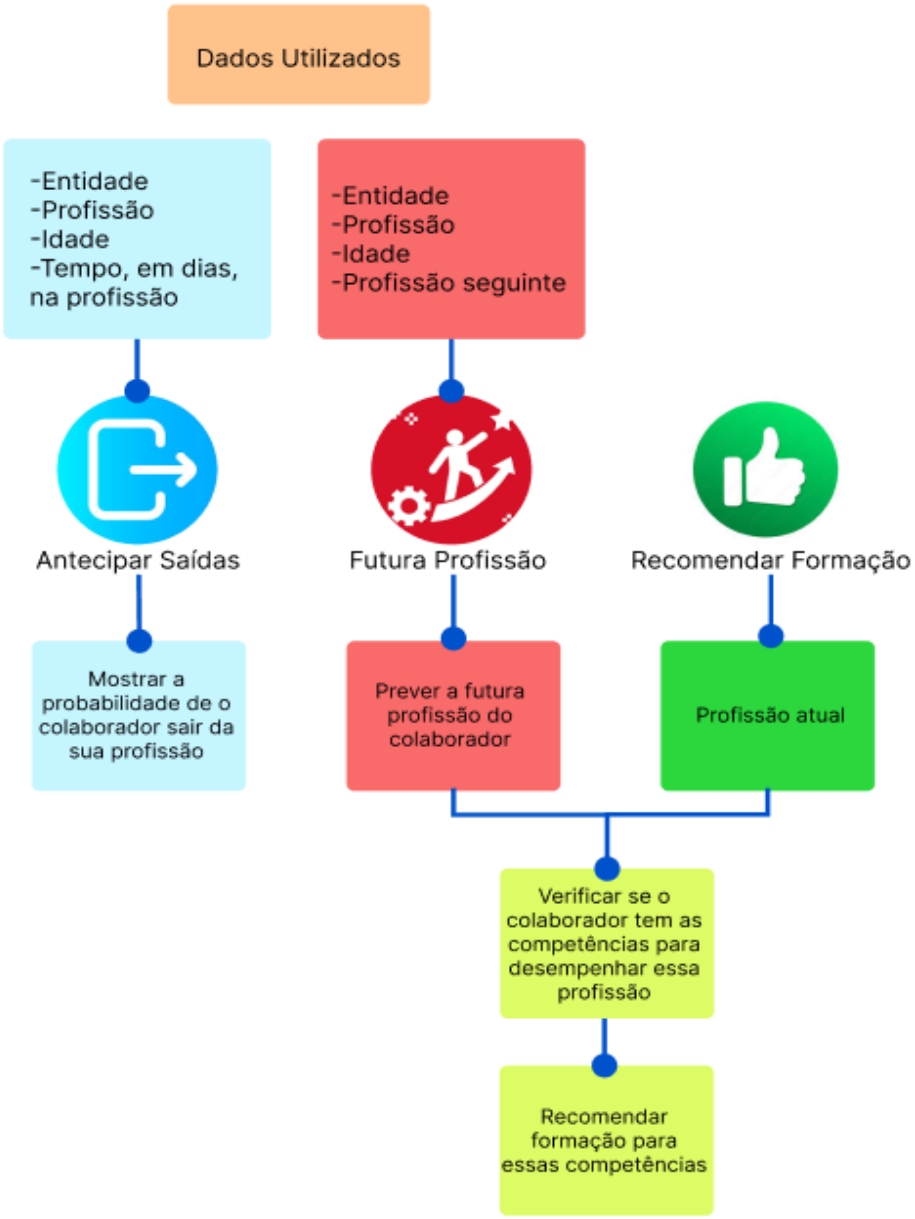

Figura 3 - Ilustração de objetivos propostos

3.3. Identificação de requisitos

Esta secção descreve a fase inicial do processo de desenvolvimento do sistema de recomendação com foco na recomendação de formações ou percursos formativos com base na profissão ou função. A análise detalhada dos requisitos específicos que a solução deve atender é fornecida.

A identificação de requisitos foi realizada por meio de várias reuniões com clientes de diferentes empresas/organizações verificou-se os requisitos essenciais para o projeto, na Tabela *3*.

De acordo com a metodologia DSRM, a identificação dos requisitos é um passo fundamental na criação de um sistema. Na Tabela *3*, vai ser apresentada uma lista de requisitos essencial para o desenvolvimento deste projeto.

Tabela 3 - Requisitos identificados

#	Requisito
1	O sistema deve analisar e prever os padrões de progressão de carreira, com base no histórico profissional, na entidade, da idade que começou a função e na localidade.
2	O sistema deve prever a possíveis saídas de colaboradores, com base no tempo médio de permanência em cada função.
3	O sistema deve ser utilizado *ML* para a construção de um modelo, que possa ajudar a desenvolver os requisitos 1 e 2. Mais especificamente, usar o *Google Cloud Plataform* (Vertex) para a criação e treino desses modelos.
4	O sistema deve identificar quais colaboradores estão com competências em falta para a função atual ou futura. Este deve indicar formações que contenham as competências em falta para que o colaborador consiga adquirir as mesmas.
5	As competências são obtidas através do site "https://catalogo.anqep.gov.pt/ucPesquisa", denominadas competências transversais.

4. Desenvolvimento da arquitetura

Neste tópico, é demonstrado o funcionamento do sistema de recomendação em três casos diferentes relativos aos objetivos "Futura função", "Antecipar possíveis saídas" e "Recomendar Formação", respetivamente. A *Figura 4* e a *Figura 5,* contém quase o mesmo processo, a única diferença é o tipo de dados que o sistema de recomendação vai recolher.

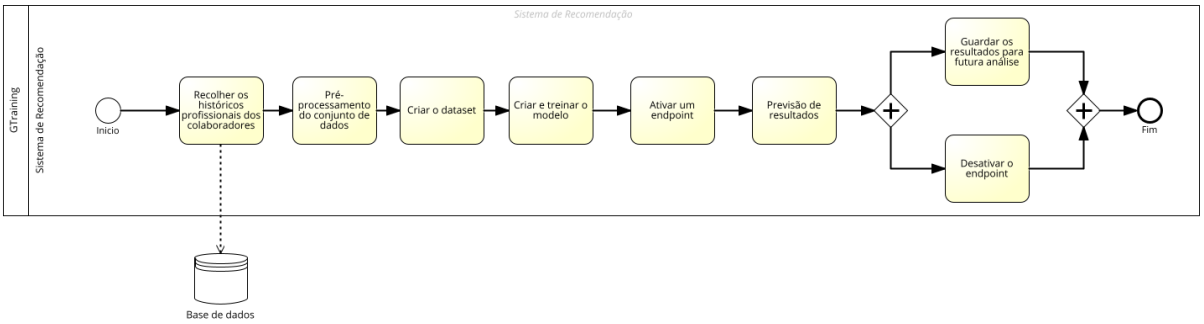

Figura 4 - BPMN - Caso 1 - Profissão futura

No caso da Figura 4, o sistema vai recolher o conjunto de dados relativo ao histórico profissional dos colaboradores, como a profissão, a idade em que começou o contrato, a entidade e a localidade. Depois, é realizado o pré-processamento do conjunto de dados para atender aos requisitos do GCP, como eliminar dados com *strings* vazias ou nulas, formalizar as profissões para uma só linguagem, entre outros. Posteriormente, é criado o *dataset*, criado e treinado o modelo usando o *AutoML* do *Vertex* AI, para facilitar o treino do modelo. Por fim, é necessário ativar um *endpoint* para aceder ao modelo, já treinado, é realizado as previsões de resultados, e no fim, são guardadas essas previsões, como também desativado o *endpoint*, utilizado para aceder ao modelo. O BPMN da *Figura 5* contém o segundo caso.

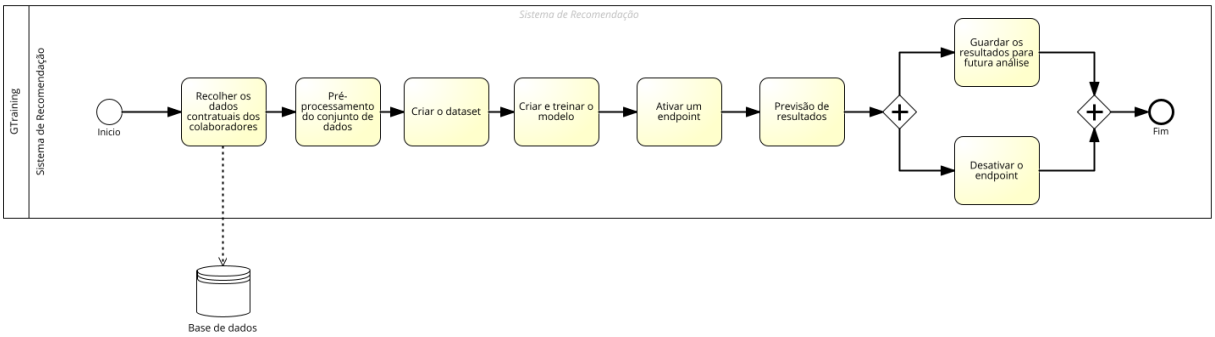

Figura 5 - BPMN - Caso 2 - Antecipar saídas

Neste caso, o processo como referido anteriormente é quase igual ao da Figura 4, a única diferença é os dados que vai recolher, estes são referentes aos dados contratuais do colaborador, mais especificamente a data de início e fim do contrato. E por fim o último caso, relativo ao "Recomendar Formação", na *Figura 6*.

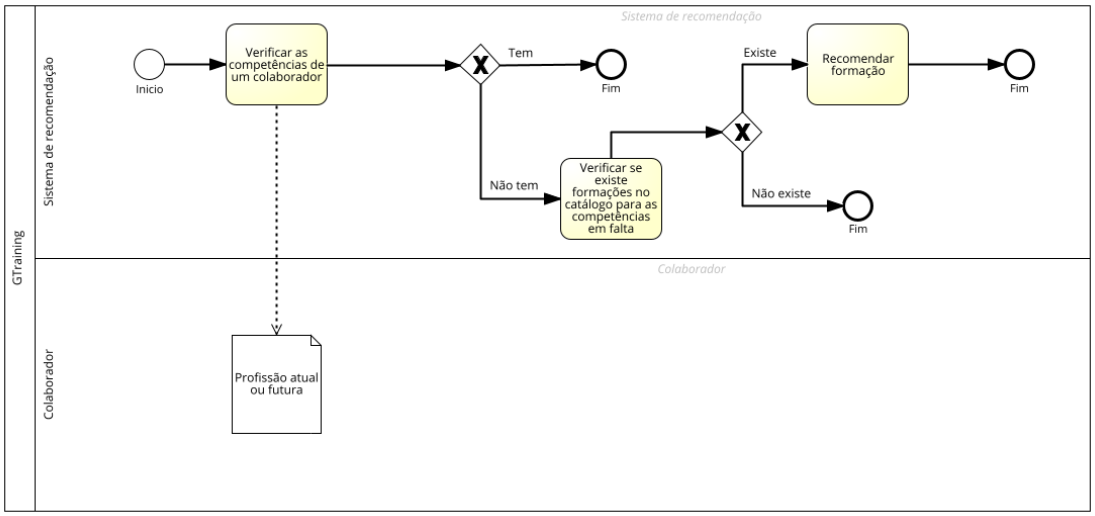

Figura 6 - BPMN - Caso 3 – Recomendação de cursos

O sistema analisa a profissão atual e/ou futura do colaborador e verifica as competências necessárias para desempenhá-la. Se o colaborador já possuir todas as competências exigidas, nenhum curso será recomendado. Caso contrário, o sistema procura no catálogo cursos que abordem as competências em falta e, se encontrados, recomenda-os ao colaborador.

4.1. Desenvolvimento do Sistema

Nesta fase, os conceitos desenvolvidos nas fases anteriores são materializados por meio de código, algoritmos e interfaces de utilizador intuitivas. Esta secção descreve o processo de construção do sistema de recomendação, bem como as decisões tomadas para atingir os objetivos estabelecidos.

A Figura 7, mostra as etapas e módulos do processo de recomendação personalizada. Isso permite uma visão clara da ordem de trabalho, que começa com a recolha de informações sobre os colaboradores e termina na previsão de resultados do modelo.

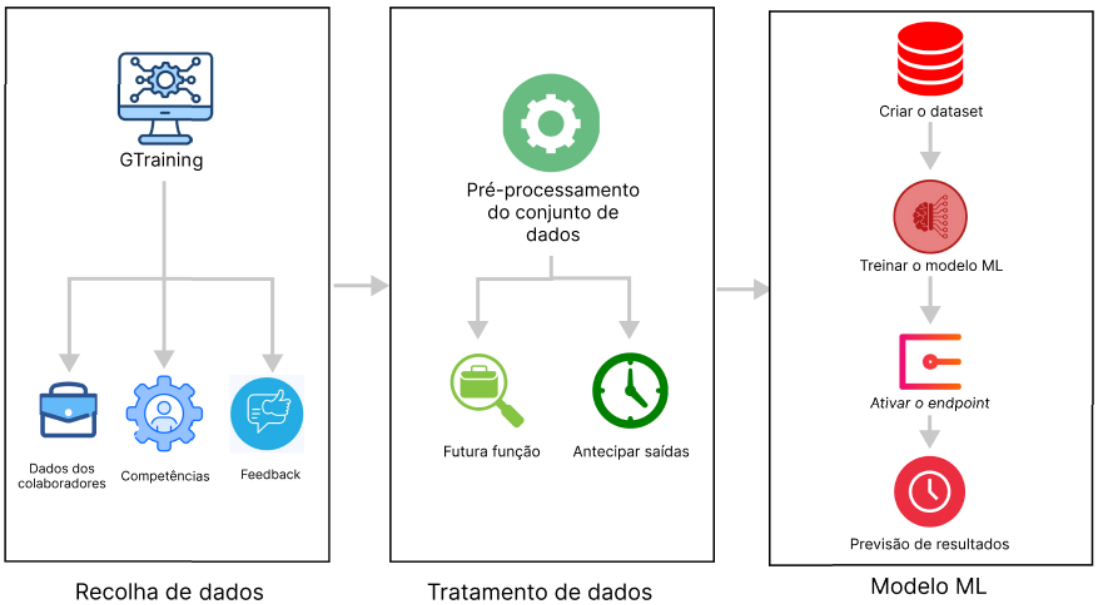

Figura 7 - Arquitetura do sistema de recomendação

4.1.1. Recolha de dados

Esta etapa inicia o processo de recolha de dados, é responsável pela compilação de dados de várias fontes. Isso inclui histórico profissional, competências e feedback relacionados a colaboradores. No histórico profissional serão armazenados dados como a função atual do colaborador e o seu histórico de empregos. Em relação às competências, além das que o colaborador já possui, serão consideradas também as competências adquiridas nas formações que ele frequentou.

Como dispomos apenas dos dados histórico profissional e competências, estes serão os únicos utilizados na fase inicial. Posteriormente, quando houver um volume suficiente de feedback dos colaboradores sobre as formações recomendadas, esses dados poderão ser empregues para treinar o modelo.

Foi utilizado os seguintes conjuntos de dados presentes nas tabelas, Tabela 4 e Tabela 5, para prever futuras possíveis profissões dos colaboradores, como também para a antecipação de possíveis saídas, respetivamente e *Figura 6* para recomendações de cursos.

Tabela 4 - Conjunto de dados - Futura Profissão

Nome da coluna	Descrição	Tipo de dados
Entidade	Entidade empregadora	Categórico
Profissão	Profissão atual do contrato	Categórico
Localidade	Localidade atual	Categórico
Idade	Idade no início do contrato	Numérico
Próximo emprego	Próxima profissão, a seguir á profissão atual do contrato	Categórico

Tabela 5 - Conjunto de dados - Antecipar Saídas

Nome da coluna	Descrição	Tipo de dados
Entidade	Entidade empregadora	Categórico
Profissão	Profissão atual do contrato	Categórico
Localidade	Localidade atual	Categórico
Duração	Duração, em dias, do contrato atual	Numérico
Mudou de Profissão	Se mudou de profissão	Numérico

A *Figura 8*, contém a lista de competências que vão ser usadas para a recomendação personalizada.

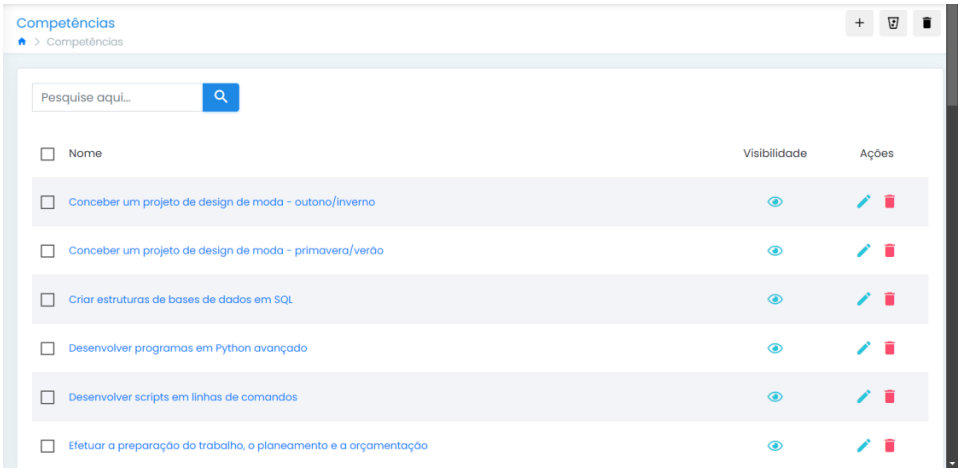

Figura 8 - Conjunto de dados - Lista de competências

Estas competências são a base para desenvolver o projeto no *gtrainingG*. Vão ser associadas a profissões e funções, para completar os 3 objetivos do projeto, no primeiro vai ser usado as competências em falta para uma profissão/função para a recomendação de cursos que tenham essas competências para adquirir. No segundo objetivo, é o mesmo processo que no primeiro, só que em vez de ser para a profissão/função atual, é para a futura profissão/função. E no último objetivo, vai verificar quais colaboradores estão perto de sair na profissão/função, e recomenda cursos para os próximos sucessores, caso não tenham as competências necessárias para essa profissão/função.

4.1.2. Tratamento de dados

Nesta etapa, vai haver o pré-processamento do mesmo conjunto de dados para fins diferentes, um para a futura função e outro para antecipar possíveis saídas. Nos dois conjuntos de dados, é necessário tratar de valores ausentes, inconsistências, além de traduzir palavras para uma só língua (neste caso é o Inglês).

O conjunto de dados seguinte da Tabela 6 é o original, ou seja, os conjuntos de dados das Tabela *4* e Tabela *5*, são resultados do seguinte conjunto de dados.

Tabela 6 - Conjunto de dados original

Nome da coluna	Tipo de dados	Descrição	Valores em falta	Valores distintos
#	Numérico	ID da instância	-	11896
ID	Numérico	ID do colaborador	-	11896
USER_ID(Contrato/Percurso Profissional)	Numérico	-	-	2152
PARENT_ID (Contrato)	Numérico	ID correspondente ao ID do colaborador	-	2200
Nº Col. (Contrato)	Numérico	-	-	2152
Entidade Empregadora (Contrato)	Categórico	Empresa ou organização que empregue o colaborador	-	3
Data Inicial (Contrato)	DateTime (data e hora)	Data inicial do contrato	-	1401
Data Saída (Contrato)	DateTime (data e hora)	Data final do contrato	-	881
Regime Reforma Aplicado (Contrato)	Categórico	-	-	2
Tipo Contrato (Contrato)	Categórico	-	-	3
Departamento/Equipa/Grupo (Percurso Profissional)	Categórico	Departamento ou Equipa ou Grupo	-	123
Unidade de Negócio (Percurso Profissional)	Categórico	-	-	32
Secção (Percurso Profissional)	Categórico	Secção de negócio do colaborador	-	225
Profissão (Percurso Profissional)	Categórico	Profissão do colaborador	-	302
Especialidade (Percurso Profissional)	Categórico	Especialidade do colaborador	-	1
Função (Percurso Profissional)	Categórico	Função do colaborador	-	1
Data Inicial (Percurso Profissional)	DateTime (data e hora)	Data inicial da profissão empregue	-	870
Data Saída (Percurso Profissional)	DateTime (data e hora)	Data final da profissão empregue	-	879
Chefia (Percurso Profissional)	Numérico	-	-	226
User Locality	Categórico	Localidade do colaborador		182
User Birthdate	DateTime (data e hora)	Data de nascimento do colaborador	-	-

Mas para ser usado para o treino dos modelos, tem de ser processado. Nos conjuntos de dados, algumas colunas foram excluídas por conterem valores vazios, valores que apontam para outras tabelas não necessárias, valores considerados inúteis para este modelo e valores que são usados para gerar outras colunas. Depois, ao analisar a documentação do GCP [29] as colunas não podem ter certos caracteres e espaços vazios, logo os nomes das colunas também tiveram de ser alterados.

Depois na coluna "Profissão (Percurso Profissional)" que contém o nome das profissões algumas em inglês e outras em português, então traduziu-se os nomes em português para inglês. "Inicialmente, utilizou-se o Google Tradutor, mas percebeu-se que as traduções não eram completamente precisas. Por isso, optou-se por usar a API do *DeepL* [30] , que oferece um plano gratuito de até 500.000 caracteres por mês, o que é mais que suficiente para este projeto, tendo em conta que um processo de tradução, para este conjunto de dados, gasta 20.000 caracteres.

Em seguida para o fim relativo à futura função, foi criada a coluna alvo [31] ("target column"), denominada "Proximo_Emprego", que, como o nome indica, representa o emprego subsequente ao atual. Esta vai ser obtida através da coluna "Profissao_Percurso_Profissional", como explicado no *Anexo III*.

Relativo ao fim de antecipar função, a coluna alvo vai ser denominada "Saida", esta vai ter os calores 0 ou 1, vai indicar se o colaborador não saiu ou saiu da função que estava a desempenhar.

Além das colunas alvos, foram criadas duas outras colunas, uma que indica a idade no início do contrato e a outra mostra os dias que durou o contrato, relativos á futura profissão e para antecipar saídas, respetivamente. A idade é calculada com os dados "data de nascimento" e "data de início de contrato", e os dados para calcular os dias são a "data de início de contrato" e a "data final de contrato" (se não existir data final é usado a data atual). Estas colunas, usadas para o cálculo, não constam nas tabelas 3 e 4 porque são usadas só para o cálculo da idade e dos dias, não são necessárias para o modelo.

Por fim, basta fazer upload dos conjuntos de dados anteriores para o *Cloud Storage* do GCP, para que posso ser usado para construir o *dataset* e posteriormente para o modelo ML.

4.1.3. *Dataset* e Modelo *ML*

Com pré-processamento do conjunto de dados realizado, chegou á altura de criar o *dataset*. Vai ser usado o *dataset* do tipo "Tabular" que vai ser usado para regressão ou classificação, neste caso é usado para classificação, mas no exemplo do *Anexo II* foi usado para regressão (por se tratar de dados numéricos. Depois, como explicado no *Anexo III*, vai procurar o conjunto de dados, mencionado anteriormente, no *Cloud Storage* e cria o *dataset*. Também é possível visualizar os valores em falta e os valores únicos.

Finalmente, um dos últimos passos é criar e treinar o modelo, para isso é preciso especificar o *dataset* que o modelo vai usar para treinar, o tipo de previsão (neste caso é a classificação), o tipo de dados de cada coluna do dataset (categórico, numérico ou *datetime*) e a percentagem do *dataset* que vai ser usada para o treino, a validação e o teste. Por fim, para poder utilizar o modelo depois do processo de treinamento (neste caso, o treino demora aproximadamente duas horas), tem de adicionar um *endpoint* ao modelo, que é usado para previsão de instâncias (que vão conter dados de colaboradores). Para mais informações consultar o *Anexo III*.

O último passo é a previsão dos resultados, com o endpoint ativo é possível enviar um pedido com as instâncias que vão ser previstas, estas vão ser enviadas em JSON (pode ser enviado uma ou mais instâncias) com os seguintes campos: Entidade empregadora; Profissão atual; Localização; Idade. Como o campo que o modelo quer prever é o da próxima profissão, este não deve constar nos campos enviados para a previsão. Para melhores resultados foi selecionado as três melhores pontuações, como se pode verificar pelo *Anexo III*.

4.2. Tecnologias utilizadas

4.2.1. gtrainiG

Um Sistema de Gestão de Formação (SGF) completo e personalizável, desenvolvido pela *FPTIC Consulting* para auxiliar empresas na gestão dos seus programas de formação e desenvolvimento profissional. As funcionalidades principais desta plataforma são as seguintes [32].

- Gestão de cursos/ações: Criação, edição, organização e publicação de cursos online e presenciais.
- Diagnósticos de necessidades formativas;
- Gestão de formandos/colaboradores: Inscrições, acompanhamento do progresso dos alunos e emissão de certificados.
- Gestão de entidades formadoras/formadores;
- Entrega de conteúdos (LMS).
- Gestão de custos;
- Avaliações e relatórios: Criação e aplicação de testes e avaliações qualitativas e quantitativas, criação de relatórios de desempenho individual e por turma.
- Comunicação e colaboração: Ferramentas para comunicação entre alunos e formadores, fóruns de discussão e partilha de materiais.
- Personalização: Adaptação da plataforma à identidade visual da empresa e às suas necessidades específicas.

4.2.2. Google Cloud

O *Google Cloud Plataform* (GCP) consiste num conjunto de recursos físicos, como computadores e unidades de disco rígido, e recursos virtuais, como máquinas virtuais, localizados nos *data centers* por todo o mundo, também oferece ferramentas avançadas de análise de dados, ML e AI. O GCP permite que empresas e desenvolvedores construam, testem e implementem aplicações de forma rápida e escalável, e utiliza a mesma infraestrutura que sustenta os próprios serviços do Google [33].

Com o GCP, as empresas podem otimizar os seus processos, reduzir custos e impulsionar a inovação. Seja para desenvolver aplicações web e móveis, armazenar grandes volumes de dados, realizar análises complexas ou criar modelos de ML, o GCP oferece as ferramentas e a flexibilidade necessárias para atender às necessidades do cliente [33].

A *Vertex AI* é uma plataforma de ML que permite treinar e implementar modelos de ML e aplicações de *AI, mas contém custo na sua utilização*. A *Vertex AI* combina fluxos de trabalho de engenharia de dados, ciência de dados e engenharia de ML, permitindo a colaboração em equipa utilizando um conjunto de ferramentas comum [29]. A plataforma oferece várias opções para treinar e implementar modelos [29] como os seguintes.

- O *AutoML* permite treinar dados de tabela, de imagem, de texto ou de vídeo sem escrever código ou preparar divisões de dados.
- O treino personalizado oferece controlo total sobre o processo de treino, incluindo o uso da *framework* de ML de preferência.
- O *Model Garden* permite descobrir, testar personalizar e implementar a *Vertex AI*, além de selecionar modelos e recursos de código aberto.
- A AI generativa dá acesso aos grandes modelos de AI generativa do Google para várias modalidades (textos, códigos, imagens, fala).

Para visualizar melhor o funcionamento da *Vertex AI* em relação à preparação de dados e ao desenvolvimento de um modelo personalizado, consultar o *Anexo II* para um exemplo da utilização da mesma.

4.2.3. Resultados do ML

Ao executar os scripts dos *Anexos III, V e VII*, para o objetivo "Futura profissão" é gerado os seguintes campos, da Tabela *7*.

Tabela 7 - Resultados - Campos - Futura profissão

Nome da coluna	Descrição
USER_ID_Contrato_Percurso_Profissional	ID do utilizador
Proxima Profissao	Futura profissão
Probabilidade	Probabilidade da profissão

A *Figura 9,* contém parte dos resultados, relativo ao objetivo "Futura Profissão", da Tabela *7*.

42733	Senior Waiter	0.41779909
42736	2nd cook	0.50818667
42737	Waiter	0.18488798
42738	2nd cook	0.52519433
42739	Chef de Partie	0.50028123
42740	Steward	0.18844537
42742	1st Receptionist	0.2913668
42743	Pastry Senior Cook	0.02647898

Figura 9 - Resultados - Futura Profissão

E para o outro objetivo "Antecipar saídas", os campos que estão presentes na Tabela *8*, vão ser usados para obter as previsões.

Tabela 8 - Resultados - Campos - Antecipar profissão

Nome da coluna	Descrição
USER_ID_Contrato_Percurso_Profissional	ID do utilizador
Saiu	Se mudou de profissão
Probabilidade	0 a 1

E por fim a *Figura 10* contém parte dos resultados do objetivo anterior.

42733	False	0.860854	True	0.139146
42734	False	0.737465	True	0.262535
42736	False	0.920745	True	0.079255
42737	False	0.961461	True	0.038539
42738	False	0.920745	True	0.079255
42739	False	0.732522	True	0.267478
42740	False	0.834701	True	0.165299
42741	False	0.834381	True	0.165619
42742	True	0.957306	False	0.042694

Figura 10 - Resultados - Antecipar Saídas

Ainda na *Figura 10*, os dados estão divididos nas seguintes colunas: Id do colaborador; *True*/*False* 1; Número da probabilidade do 1; *True*/*False* 2; Número da probabilidade do 2. Estes dados estão apresentados desta maneira devido ao script, mas na plataforma irá ser só a probabilidade de sair que é o valor *True*, por exemplo no primeiro registo, a probabilidade de o colaborador 42733 sair é 0.139.

Os *Anexos IV e VI* contém a avaliação dos modelos, com a explicação de cada métrica.

5. Demonstração

Este capítulo apresenta o funcionamento do sistema de recomendação, destacando as suas principais funcionalidades relacionadas ao objetivo definido. Em específico, será abordado as capacidades do sistema de antecipar saídas profissionais, sugerir uma possível futura profissão e recomendar formações com base na profissão atual e/ou futura.

Os dados que vão ser utilizados pertencem a utilizadores reais, de clientes da FPTIC *Consulting,* além dos dados relativos aos cursos e competências.

5.1. Demonstração do sistema

Neste tópico, vai ser demonstrado as funcionalidades implementadas com base no objetivo proposto para este projeto. Começando pela lista de competências, Figura *11*, que é a base para as recomendações.

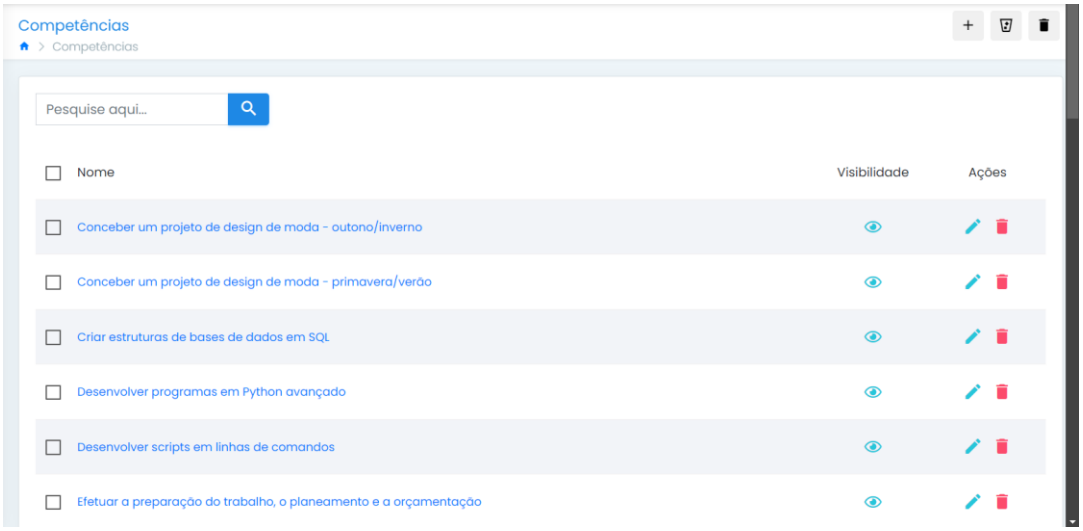

Figura 11 - Lista de competências

A lista anterior retirada do catálogo da ANQEP, contém todas as competências que vão ser usadas para as recomendações de formação. Nesta lista é possível pesquisar, visualizar, editar, criar e remover competências, mas só aos gestores de formação é possível realizar este tipo de funcionalidades.

As figuras, *Figura 12* e *Figura 13*, contêm a exemplificação do objetivo relativo a antecipar possíveis saídas.

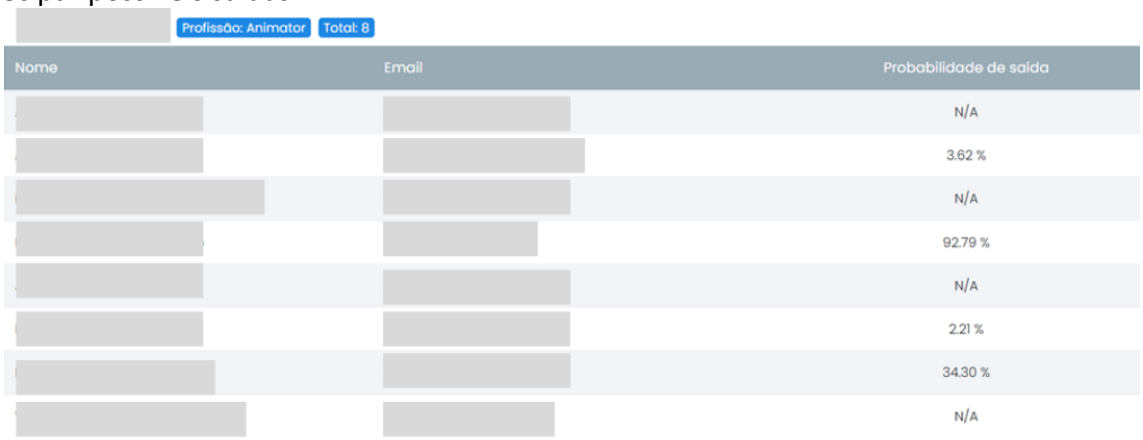

Figura 12 - Antecipar saídas - Colaboradores

É possível verificar pela *Figura 12*, os colaboradores que têm como profissão "*Animator*", possuem uma percentagem que indica a probabilidade de sair dessa profissão (alguns não possuem).

Figura 13 - Antecipar saídas - Colaboradores por profissão

Na *Figura 13*, a profissão "*Animator*" que conta com 8 colaboradores, só 1 é que está perto de sair, isto é, devido ao facto de a probabilidade de saída desse colaborador ser superior a 50%, logo conta para a contagem de colaboradores mais perto de sair. Agora para a futura profissão/função, na *Figura 14*.

Figura 14 - Futura profissão/função - Profissão

Este colaborador tem como futura profissão "*Front Office Agent*" e para realizar esta profissão é necessário adquirir certas competências que este colaborador não contêm, logo é recomendado o curso "Espanhol Técnico - Iniciação" para adquirir essas competências em falta.

E por último, para as recomendações das formações com base nas competências em falta para a profissão e/ou função, a figura exemplifica esse objetivo.

Figura 15 - Recomendar formação

Neste caso, o colaborador com a profissão "*Bell Attendant*" e função "*Bell Function*", tem quatro recomendações de cursos para adquirir as competências necessárias para desempenhar a profissão e função respetivas.

6. Avaliação

A avaliação do sistema de recomendação é essencial para assegurar que este responde aos objetivos propostos. Ao analisar o trabalho realizado, é possível avaliar a sua capacidade de personalização de recomendações, prever possíveis cenários e adaptar-se às necessidades das organizações. Esta análise permitirá identificar pontos fortes, oportunidades de melhoria e consolidar o sistema como uma ferramenta necessária para os recursos humanos e gestores de formação. A *Tabela 9* é um resumo dos pontos fortes e fracos do sistema de recomendação.

Tabela 9 - Pontos fortes e fracos

Pontos Fortes	Pontos Fracos
• Recomendações Personalizadas	• Modelo da Futura Profissão com um baixo grau de precisão
• Modelo do Antecipar Saídas com um elevado grau de precisão	• Falta do feedback dos utilizadores, relativo á formação recomendada.
• Interface amigável	• Qualidade dos dados da profissão ou função
• Previsões atualizadas mensalmente	• Custos de treino de modelos ML

As recomendações de formação são personalizadas, baseando-se em lacunas identificadas entre as competências que o colaborador possui, e as competências que são exigidas para a profissão atual e/ou futura, o sistema sugere formações que não apenas preenchem essas lacunas, mas também fortalecem o desenvolvimento contínuo do colaborador.

O sistema utiliza ML para antecipar potenciais saídas de colaboradores com um elevado grau de precisão. Por meio da análise de variáveis como tempo na função/profissão, histórico profissional, tempo médio do colaborador no cargo, o sistema oferece esta funcionalidade que permite ações proativas. Desta forma, as organizações podem adotar estratégias de retenção ou o início de planos para a sucessão desse colaborador, assegurando assim a continuidade da produção.

Outro diferenciador que o sistema oferece é a forma como apresenta esta informação dita anteriormente. Os utilizadores podem facilmente tomar decisões importantes, por meio de uma interface amigável. Este design prioriza a acessibilidade, facilitando a navegação.

O sistema está planeado para fazer a recolha de dados mês a mês, o que garante a recalibração de novas previsões, o que melhora a capacidade de adaptação para mudanças do mercado de trabalho. Também é necessário realçar, que o *Vertex AI* é uma ferramenta paga e isto pode ser um constrangimento para algumas entidades.

Existem algumas lacunas que o sistema de recomendação pode melhorar como, o modelo utilizado para prever a profissão futura, não está com uma percentagem de precisão elevado (67%), o que indica que o modelo pode não prever a profissão futura correta. Ainda na profissão futura, o sistema estava planeado para ser profissão e função, mas como não havia dados para funções, foi só utilizado a profissão para a previsão, isto é uma funcionalidade que pode ser melhorada.

Outra lacuna que pode ser indicadora que o sistema de recomendação está a funcionar corretamente, é a obtenção do feedback dos utilizadores, neste caso, dos gestores de formação e colaboradores. Com a recolha do feedback dos utilizadores, os modelos podem ser precisos em relação á previsão, o que que melhora o sistema de recomendação, pouco a pouco.

Por fim, uma falha relativa à qualidade de dados das profissões é o nome da mesma, os clientes ao preencherem o nome da profissão, não eram coerentes, há versões em português, em inglês, isto afetou a qualidade das previsões, porque foi usado um tradutor para uniformizar o nome, neste caso, das profissões.

Com base no modelo DSRM, na Figura *2*, o projeto foi largamente atingido (L). Estes resultados foram obtidos através de reuniões semanais com os clientes da *FPTIC Consulting*.

A avaliação da proposta foi avaliada por diferentes perfis de utilizadores da plataforma, tais como gestores da formação, técnicos de recursos humanos, diretores de recursos humanos durante um determinado período já em produção, onde abaixo apresentamos na Tabela 10.

Tabela 10 - Avaliação dos utilizadores - Resumo

Perfil do Utilizador	Nº de respostas	Tópicos avaliados	Avaliação (1-5)	Principais Frases
GF/TRH/DRH	35	Personalização das recomendações	4	"As recomendações são relevantes, mas podem ser aprimoradas com mais feedback dos utilizadores."
GF/TRH/DRH	35	Precisão do modelo de previsão de profissão futura	3	"A precisão do modelo de profissão futura é baixa (67%), pode ser melhorada."
GF/TRH/DRH	35	Precisão do modelo de antecipação de saídas	5	"O modelo de antecipação de saídas tem alta precisão e ajuda na tomada de decisões."
GF/TRH/DRH	35	Interface e usabilidade	5	"A interface é amigável e facilita a tomada de decisão."
GF/TRH/DRH	35	Qualidade dos dados de profissões	3	"Os dados das profissões precisam de melhor padronização e qualidade."

Tabela 11 - Legendas da Tabela 10

Legenda de perfil de utilizador:	
GF	Gestor de Formação
TRH	Técnico de Recursos Humanos
DRH	Diretor de Recursos Humanos

Escala de avaliação:	
1	Pouco Satisfeito
5	Muito Satisfeito

7. Conclusões e Trabalho futuro

7.1. Conclusão

O desenvolvimento do sistema de recomendação apresentado nesta tese demonstrou ser uma ferramenta promissora para a gestão de formação e desenvolvimento de competências nas organizações. Ao integrar a personalização das recomendações com a previsão da futura profissão e a antecipação de possíveis saídas de colaboradores, o sistema responde a desafios críticos enfrentados pelos departamentos de recursos humanos.

Através de funcionalidades como a análise de competências, previsão de profissões futuras e recomendação de formações, foi possível alinhar as necessidades das organizações com o crescimento profissional dos colaboradores. A adoção de técnicas de ML para prever saídas e identificar futuras profissões permitiu criar um ambiente proativo, no qual as organizações podem atuar estrategicamente para a retenção e substituição de talentos.

Embora o sistema tenha demonstrado resultados positivos, pode ainda fornecer recomendações incorretas e inválidas aos colaboradores. Os modelos de previsão encontram-se numa fase de aprendizagem o que pode gerar resultados que não correspondem totalmente às expetativas dos utilizadores. No entanto, com o feedback dos utilizadores, o sistema poderá melhorar significativamente os resultados apresentados. Esta limitação deve-se, em parte, à impossibilidade de realizar testes extensivos para identificar eventuais falhas no sistema, devido ao prazo reduzido para a sua implementação.

Em suma, o objetivo foi alcançado, e o sistema representa um ponto de partida promissor. Apesar de poder apresentar limitações, com as previsões incertas, constitui uma base sólida para expandir e refinar as suas funcionalidades ao longo do tempo. A implementação atual do sistema é suficiente para responder às suas necessidades iniciais.

7.2. Trabalho Futuro

As iterações futuras poderão concentrar-se na melhoria contínua, umas das quais será a introdução do feedback dos utilizadores no momento da apresentação das recomendações dos planos de formação e assim recolher e analisar os resultados e a experiência em geral.

Submeter com periodicidade relatórios baseados neste novo sistema de recomendação às chefias e cargos de direção para avaliarem o impacto da formação e assim contribuírem para apurar a eficácia da formação.

E por fim, melhorar a apresentação dos dados relativos às futuras saídas e previsões de progressão de carreira, com uso de ilustrações ou gráficos.

Bibliografia

[1] T. Marra, “Qual é a importância da formação contínua no mercado de trabalho?,” Gi Group , [Online]. Available: https://www.gigroupholding.com/portugal/insights/importancia-formacao-continua/. [Acedido em 11 Setembro 2024].

[2] M. Karanjia, “Top 9 Corporate Training Challenges & How To Tackle Them in 2024,” IIDE, [Online]. Available: https://iide.co/blog/corporate-training-challenges/. [Acedido em 13 Setembro 2024].

[3] S. Solutions, “7 Challenges in Training and Development and How to Overcome Them,” SVC Solutions, [Online]. Available: https://svcsolutions.co.uk/7-challenges-in-training-and-development/. [Acedido em 13 Setembro 2024].

[4] S. Chatterjee, “Recommendation Systems: Ethical Challenges and the Regulatory Landscape,” Holistic AI, 7 Junho 2023. [Online]. Available: https://www.holisticai.com/blog/recommendation-systems. [Acedido em 11 Setembro 2024].

[5] I. Convergence, “Challenges and Solutions for Building Effective Recommendation Systems,” IT Convergence, 6 Abril 2023. [Online]. Available: https://www.itconvergence.com/blog/challenges-and-solutions-for-building-effective-recommendation-systems/. [Acedido em 11 Setembro 2024].

[6] N. Shukla, “Challenges of the Recommendation system,” Medium, 1 Setembro 2023. [Online]. Available: https://medium.com/@nidhishukla2023/challenges-of-the-recommendation-system-f2406b3f2232. [Acedido em 11 Setembro 2024].

[7] D. Zhenhua, W. Zhe, X. Jun, T. Ruining e W. Jirong, “A Brief History of Recommender Systems,” *A Brief History of Recommender Systems,* pp. 1-2, 5 Setembro 2022.

[8] 101.school, “History and Evolution of Recommender Systems,” 101.school, [Online]. Available: https://101.school/courses/recommendation-systems/modules/1-introduction-to-recommender-systems/units/1-history-and-evolution-of-recommender-systems. [Acedido em 10 Setembro 2024].

[9] R. System, “NVIDIA - Recommendation System,” [Online]. Available: https://www.nvidia.com/en-eu/glossary/recommendation-system/. [Acedido em 8 Julho 2024].

[10] IBM, “What is machine learning,” [Online]. Available: https://www.ibm.com/topics/machine-learning. [Acedido em 10 Julho 2024].

[11] IBM, “Types of machine learning algorithms,” IBM, [Online]. Available: https://www.ibm.com/topics/machine-learning-algorithms. [Acedido em 23 Julho 2024].

[12] Databricks, “What is a Dataset?,” Databricks, [Online]. Available: https://www.databricks.com/glossary/what-is-dataset. [Acedido em 16 Setembro 2024].

[13] Microsoft, “What is an endpoint?,” Microsoft, [Online]. Available: https://www.microsoft.com/en-us/security/business/security-101/what-is-an-endpoint. [Acedido em 5 Setembro 2024].

[14] G. Cloud, “REST Resource: projects.locations.endpoints,” Google Cloud, [Online]. Available: https://cloud.google.com/vertex-ai/docs/reference/rest/v1/projects.locations.endpoints. [Acedido em 5 Setembro 2024].

[15] M. J. P. e. al., “The PRISMA 2020 statement: an updated guideline for reporting systematic reviews,” *BMJ,* vol. 372, p. n71, Março 2021.

[16] “IEEE Xplore - Document search,” [Online]. Available: https://ieeexplore.ieee.org/Xplore/home.jsp. [Acedido em 2 Julho 2024].

[17] “Web of Science Core Collection - Document search,” [Online]. Available: https://www.webofscience.com/wos/woscc/basic-search. [Acedido em 2 Julho 2024].

[18] “Scopus - Document search,” 203. [Online]. Available: https://www.scopus.com/search/form.uri?display=basic#basic. [Acedido em 2 Julho 2024].

[19] “Zotero | Your personal research assistant,” [Online]. Available: https://www.zotero.org/start. [Acedido em 2 Julho 2024].

[20] K. C. Guyard e M. Deriaz, “A scalable recommendation system approach for a companies - seniors matching,” *2022 5TH INTERNATIONAL CONFERENCE ON ARTIFICIAL IN TELLIGENCE FOR INDUSTRIES, AI4I,* pp. 5-9, 2022.

[21] A. Hang, I. Puskas, M. Nitu, L. V. Nemoianu e M.-l. Dascalu, “CareProfSys Recommender for Modern Engineering Roles based on Emergent Technologies,” *13th International Symposium on Advanced Topics in Electrical Engineering, ATEE 2023,* 2023.

[22] A. Deshpande, A. Dubey, A. Dhavale, A. Navatre, U. Gurav e A. K. Chanchal, “Implementation of an NLP-Driven Chatbot and ML Algorithms for Career Counseling,” *7th International Conference on Inventive Computation Technologies, ICICT 2024,* pp. 853-859, 2024.

[23] E. Gedrimiene, I. Celik, K. Makitalo e H. Muukkonen, “Transparency and Trustworthiness in User Intentions to Follow Career Recommendations from a Learning Analytics Tool,” *JOURNAL OF LEARNING ANALYTICS,* 2023.

[24] L. Irsaliyera, G. Bekmanova e S. Aljawarneh, “Development of an intellectual model of personalized learning,” *Proceedings - 2022 International Conference on Engineering and MIS, ICEMIS 2022,* 2022.

[25] K. P. T. T. M. A. R. e S. C. , “A design science research methodology for information systems research,” *Journal of Management Information Systems,* vol. 24, nº 3, pp. 45-77, 2007.

[26] ISO, “ISO/IEC 15504-2:2003,” 2003. [Online]. Available: https://www.iso.org/standard/37458.html. [Acedido em 2 Julho 2024].

[27] ISO, “ISO/IEC 33002:2015,” 2015. [Online]. Available: https://www.iso.org/standard/54176.html. [Acedido em 2 Julho 2024].

[28] K. E. Eman, “The internal consistency of the ISO/IEC 15504 software process capability scale,” em *Proceedings of the Fifth International Software Metrics Symposium*, 1998.

[29] G. Cloud, “Introdução à Vertex AI,” Google Cloud, [Online]. Available: https://cloud.google.com/vertex-ai/docs/start/introduction-unified-platform?hl=pt-br. [Acedido em 30 Julho 2024].

[30] DeepL, “Crie uma experiência multilingue com a API do DeepL,” DeepL, [Online]. Available: https://www.deepl.com/pt-PT/pro-api. [Acedido em 21 Agosto 2024].

[31] H20.ai, “What is a Target Variable in Machine Learning?,” H20.ai, [Online]. Available: https://h2o.ai/wiki/target-variable/. [Acedido em 30 Agosto 2024].

[32] GTraining, GTraining, [Online]. Available: https://gtraining.co/. [Acedido em 17 Julho 2024].

[33] Google, “Visão geral do Google Cloud,” Google, [Online]. Available: https://cloud.google.com/docs/overview?hl=pt-br. [Acedido em 30 Julho 2024].

[34] C. Crawford, “1000 Cameras Dataset,” Kaggle, [Online]. Available: https://www.kaggle.com/datasets/crawford/1000-cameras-dataset. [Acedido em 5 Julho 2024].

[35] G. Colab, “Google Colaboratory,” Google Colab, [Online]. Available: https://colab.google/. [Acedido em 23 Agosto 2024].

[36] G. Colab, “Colab Updated to Python 3.9,” Google Colab, [Online]. Available: https://colab.google/articles/py3.9. [Acedido em 23 Agosto 2024].

[37] C. Storage, “Armazenamento de objetos para empresas de todos os tamanhos,” Google Cloud, [Online]. Available: https://cloud.google.com/storage?hl=pt_br. [Acedido em 27 Agosto 2024].

[38] pandas, “pandas,” pandas, [Online]. Available: https://pandas.pydata.org/. [Acedido em 27 Agosto 2024].

[39] P. Org, “os — Miscellaneous operating system interfaces,” Python, [Online]. Available: https://docs.python.org/pt-br/3/library/os.html. [Acedido em 27 Agosto 2024].

[40] Google, “google.protobuf.json_format,” Google, [Online]. Available: https://googleapis.dev/python/protobuf/latest/google/protobuf/json_format.html. [Acedido em 27 Agosto 2024].

[41] Google, “google.protobuf.struct_pb2,” Google, [Online]. Available: https://googleapis.dev/python/protobuf/latest/google/protobuf/struct_pb2.html. [Acedido em 27 Agosto 2024].

[42] SpaCy, “Industrial-Strength Natural Language Processing,” SpaCy, [Online]. Available: https://spacy.io/. [Acedido em 21 Agosto 2024].

[43] Python, “langdetect 1.0.9,” Python, [Online]. Available: https://pypi.org/project/langdetect/. [Acedido em 27 Agosto 2024].

[44] G. Cloud, “Criar um conjunto de dados para o Cloud Storage tabular,” Google Cloud, [Online]. Available: https://cloud.google.com/vertex-ai/docs/samples/aiplatform-create-dataset-tabular-gcs-sample?hl=pt-br#aiplatform_create_dataset_tabular_gcs_sample-python. [Acedido em 2 Setembro 2024].

[45] G. f. Geeks, “Train a model using Vertex AI and the Python SDK,” Geek for Geeks, 14 Dezembro 2023. [Online]. Available: https://www.geeksforgeeks.org/train-a-model-using-vertex-ai-and-the-python-sdk/. [Acedido em 3 Setembro 2024].

[46] Microsoft, “O que é o aprendizado de máquina automatizado (AutoML)?,” Microsoft , 2 Setembro 2024. [Online]. Available: https://learn.microsoft.com/pt-pt/azure/machine-learning/concept-automated-ml?view=azureml-api-2. [Acedido em 3 Setembro 2024].

[47] R. Duarte, “Métricas de Avaliação em Modelos de Classificação em Machine Learning,” Sigmoidal, [Online]. Available: https://sigmoidal.ai/metricas-de-avaliacao-em-modelos-de-classificacao-em-machine-learning/. [Acedido em 6 Setembro 2024].

[48] “Software de Folha de Cálculo do Microsoft Excel | Microsoft 365,” [Online]. Available: https://www.microsoft.com/pt-pt/microsoft-365/excel. [Acedido em 2 Julho 2024].

[49] NVIDIA, “Recommendation System,” NVIDIA, [Online]. Available: https://www.nvidia.com/en-eu/glossary/recommendation-system/. [Acedido em 8 Julho 2024].

[50] S. Pichai e D. Hassabis, “Introducing Gemini: our largest and most capable AI model,” Gemini, 6 Dezembro 2023. [Online]. Available: https://blog.google/technology/ai/google-gemini-ai/#introducing-gemini. [Acedido em 17 Junho 2024].

[51] JSONSchema, JSON Schema Validation API, [Online]. Available: https://assertible.com/json-schema-validation. [Acedido em 17 Julho 2024].

[52] R. H. Group, “Sobre nós,” Real Hotels Group, [Online]. Available: https://www.realhotelsgroup.com/pt-pt/historia/. [Acedido em 22 Julho 2024].

[53] Python, “About Python,” Python, [Online]. Available: https://www.python.org/about/. [Acedido em 22 Julho 2024].

[54] GeeksforGeeks, “Machine Learning with Python Tutorial,” Geek for Geeks, 7 Novembro 2023. [Online]. Available: https://www.geeksforgeeks.org/machine-learning-with-python/. [Acedido em 22 Julho 2024].

[55] G. C. Plataform, “Documentação da Vertex AI,” Google Cloud Plataform, 5 Julho 2024. [Online]. Available: https://cloud.google.com/vertex-ai/docs?hl=pt-br. [Acedido em 19 Agosto 2024].

[56] DeepL, “The developer-friendly, enterprise-ready translation API,” DeepL, [Online]. Available: https://www.deepl.com/en/pro-api. [Acedido em 27 Agosto 2024].

[57] F. Consulting, “FPTIC Consulting,” FPTIC Consulting, [Online]. Available: https://fptic.com/. [Acedido em 16 Setembro 2024].

[58] R. H. Group, “Sobre Nós,” Real Hotels Group, [Online]. Available: https://www.realhotelsgroup.com/pt-pt/historia/. [Acedido em 16 Setembro 2024].

Anexos

1.1. Anexo I

Para os algoritmos supervisionados temos a seguinte lista [11]:

- *AdaBoost* ou *Grading Boosting* - Esta técnica reforça um algoritmo de regressão com fraco desempenho, combinando-o com outros mais fracos para criar um algoritmo mais forte que resulte em menos erros. O *boosting* combina o poder de previsão de vários estimadores de base.

- *Decision Tree* - Utilizado para prever valores numéricos como para classificar dados em categorias, este método utiliza uma sequência ramificada de decisões ligadas que podem ser representadas através de um diagrama de árvore. Uma das vantagens das *decisions tree* é o facto de serem fáceis de validar e auditar.

- *Dimensionality Reduction* - Quando um conjunto de dados selecionado tem um elevado número de características, tem uma dimensionalidade elevada. A redução da dimensionalidade reduz o número de características, deixando apenas as informações mais significativas.

- *K-Nearest Neighbor* - Também conhecido como KNN, este algoritmo não paramétrico classifica os pontos de dados com base na sua proximidade e associação a outros dados disponíveis. Parte do princípio de que pontos de dados semelhantes podem ser encontrados próximos uns dos outros. Como resultado, procura calcular a distância entre os pontos de dados, normalmente através da distância euclidiana, e depois atribui uma categoria com base na categoria ou média mais frequente.

- Regressão linear - É utilizada para identificar a relação entre uma variável dependente e uma ou mais variáveis independentes e é normalmente utilizada para fazer previsões sobre resultados futuros. Quando existe apenas uma variável independente e uma variável dependente, é conhecida como regressão linear simples.

- Regressão logística - Enquanto a regressão linear é utilizada para variáveis dependentes continuas, a regressão logística é selecionada quando a variável dependente é categórica, o que significa que existem resultados binários, como "verdadeiro" e "falso" ou "sim" e "não". Embora ambos os modelos de regressão procurem compreender as relações entre as entradas de dados, a regressão logística é utilizada principalmente para resolver problemas de classificação binária, como a identificação de spam.

- *Naïve Bayes* - Esta abordagem adota o princípio da independência condicional das classes do Teorema de *Bayes*. Isto significa que a presença de uma caraterística não afeta a presença de outra na probabilidade de um determinado resultado e que cada preditor tem um efeito igual nesse resultado. Existem três tipos de classificadores *Naïve Bayes*: Multinominal *Naïve Bayes*, *Bernoulli Naïve Bayes* e *Gaussian Naïve Bayes*. Esta técnica é utilizada principalmente na classificação de textos, na identificação de spam e em sistemas de recomendação.

- *Random Forest* - O algoritmo prevê um valor ou categoria combinando os resultados de várias árvores de decisão. A "*Forest*" refere-se a árvores de decisão não correlacionadas, que são montadas para reduzi a variância e permitir previsões mais exatas.

- SVM (*Suport Vector Machines*) - Este algoritmo pode ser utilizado tanto para a classificação como para a regressão de dados, mas normalmente para problemas de classificação, construindo um híper plano onde a distância entre duas classes de pontos de dados é máxima. Este híper plano é conhecido como o limite de decisão, separando as classes de pontos de dados (como laranjas vs. maçãs) em ambos os lados do plano.

1.2. Anexo II

Este anexo demonstra, por meio de um exemplo prático, como aproveitar um conjunto de dados simples [34] e utilizá-lo para treinar e avaliar um modelo na plataforma *Vertex AI*. Numa fase mais avançada vai ser utilizado um *dataset* próprio para o caso de uso deste projeto.

Na *Vertex AI* é ser possível fazer três etapas cruciais relacionadas com a criação e desenvolvimento de modelos: Preparar dados; Desenvolvimento de modelos; Implementar e usar. Na etapa de preparar dados é preciso selecionar o tipo de dados e o objetivo para os mesmos, além de introduzir um nome para o conjunto de dados.

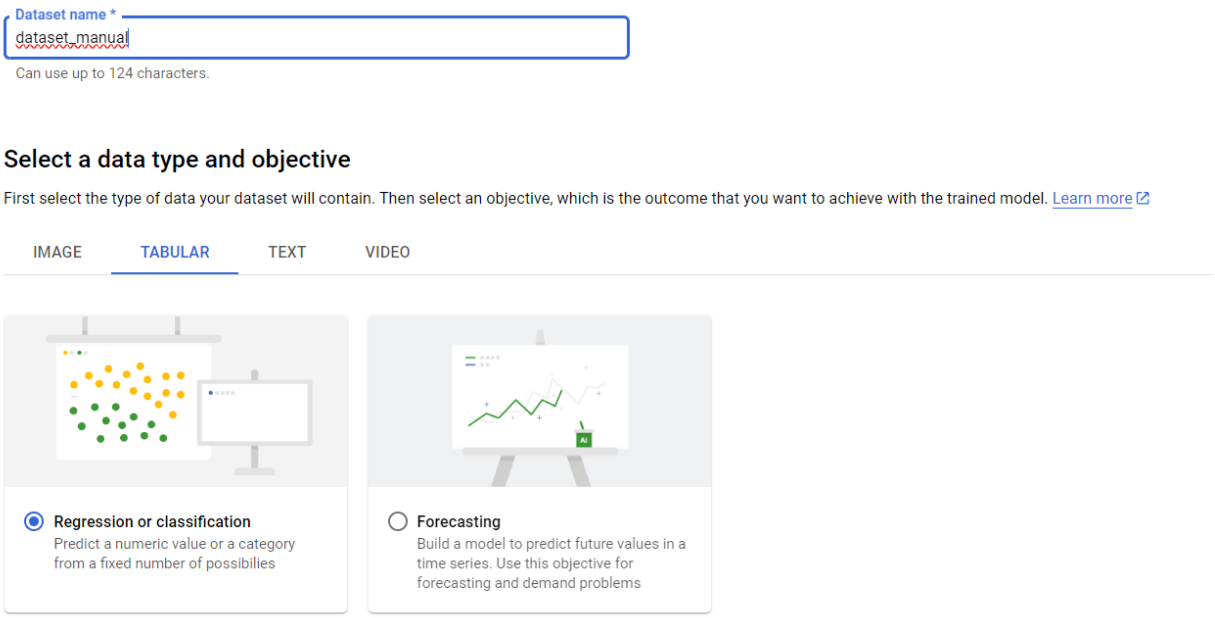

Figura 16 - *Vertex AI* - Criar *Dataset*

Como se pode verificar pela Figura *16*, foi selecionado o tipo de dados "TABULAR" porque neste caso vai ser enviado um ficheiro CSV para treinar e foi selecionado a opção "*Regression or classification*" para prever valores numéricos ou categóricos, no *dataset* de exemplo vai ser utilizado para prever valores numéricos.

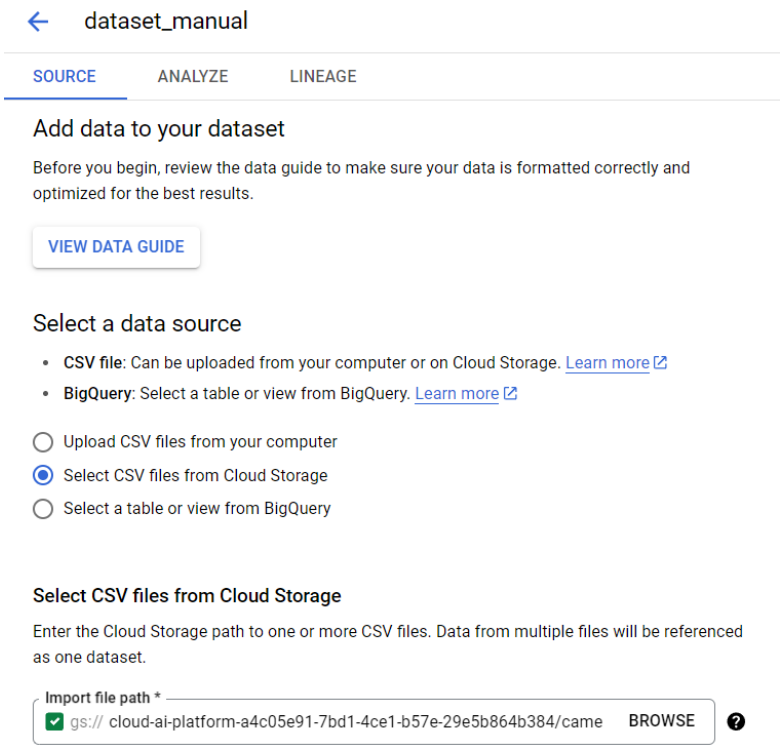

Figura 17 - *Vertex* AI - Adicionar *Dataset*

Na Figura *17*, é realizada a escolha do conjunto de dados, quer o experimental quer o que vai ser usado para o projeto, são em formato CSV. Depois é escolhido o local onde se encontra o conjunto de dados, neste caso foi selecionado o *Cloud Storage*. Este permite armazenar e aceder a dados não estruturados de qualquer tipo e tamanho, como imagens, vídeos, áudios, documentos, entre outros. Na Figura *18*, encontra-se uma visão geral do *dataset* escolhido, como o nome das colunas, valores em falta e destintos.

Column name	Missing % (count)	Distinct values
Dimensions	-	102
Effective_pixels	-	16
Low_resolution	-	70
Macro_focus_range	-	30
Max_resolution	-	99
Model	-	1038
Normal_focus_range	-	32
Price	-	43
Release_date	-	14
Storage_included	-	45
Weight_inc_batteries	-	238
Zoom_tele_T	-	100
Zoom_wide_W	-	25

Figura 18 - *Vertex* AI - Propriedades do *dataset*

Já com o *dataset* criado, a etapa seguinte é criar e treinar um modelo especificamente para este *dataset*.

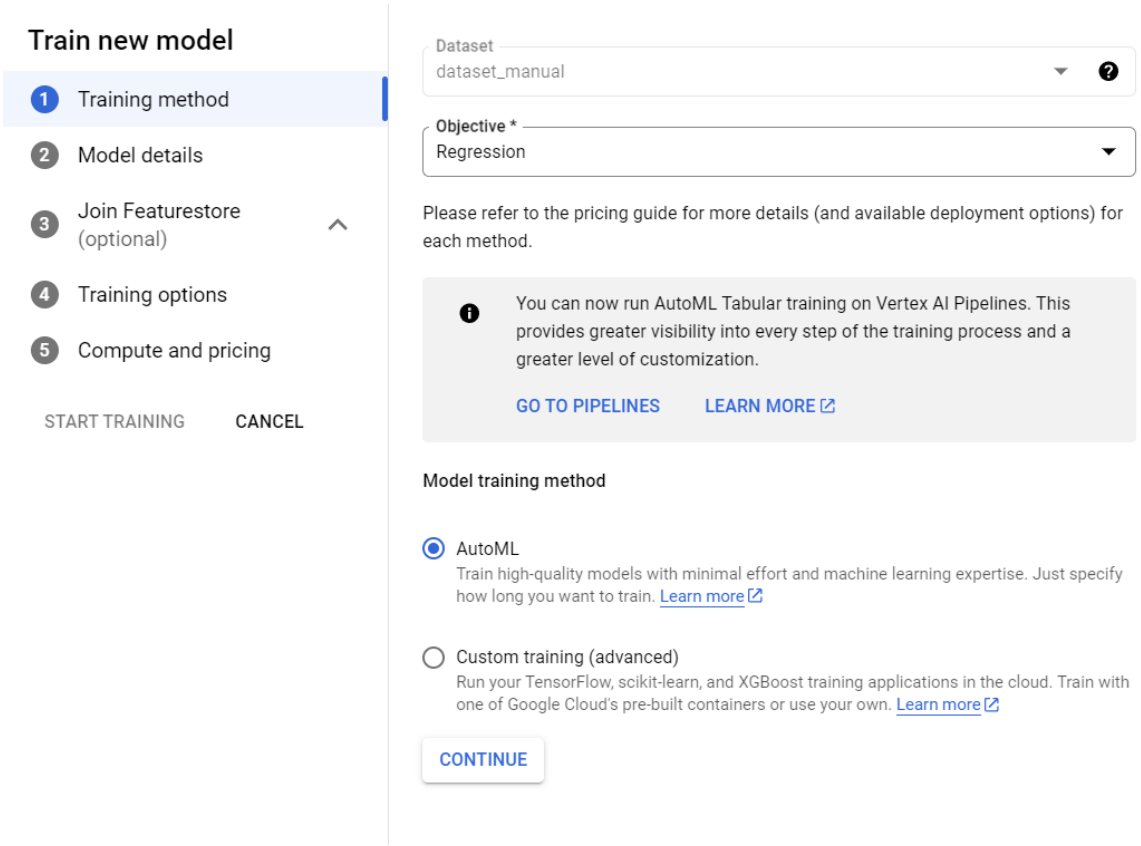

Figura 19 - *Vertex* AI - Treinar o modelo - Método de treino

Na Figura *19*, encontra-se o menu com os passos para a criação de um modelo. Na primeira etapa foi escolhida, primeiramente o *dataset* já referido (está selecionado como padrão), depois escolher o objetivo do modelo, neste caso havia duas opções, classificação ou regressão, para este *dataset* foi escolhido regressão por ser o mais indicado para este tipo de dados numéricos. Por último, o método de treino, foi eleito o método *AutoML* que comparativamente ao outro método, é mais simples de implementar e perceber, porque não é necessário código nem um conhecimento prévio de ML, só é necessário introduzir o tempo que quer que o modelo treine.

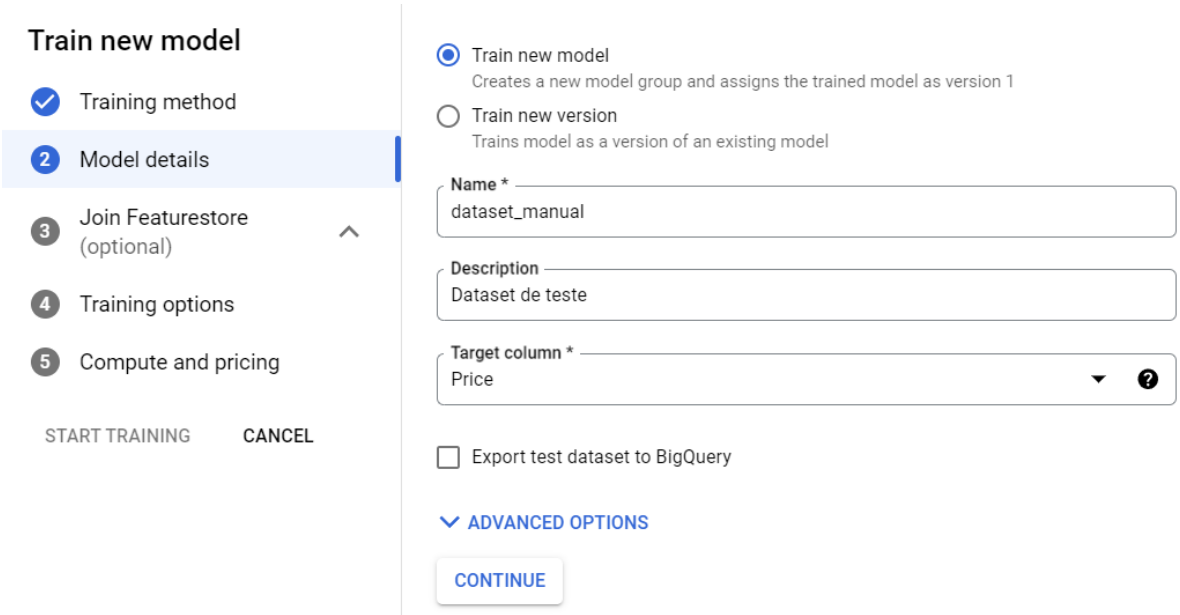

Figura 20 - *Vertex* AI - Treinar o modelo - Detalhes do modelo

Como visualizado na Figura *20*, o próximo passo é detalhar o modelo, a única *label* relevante é escolher o que o modelo vai prever, esta é denominada como "*Target column*" e a coluna que vai ser calculada é a coluna "*Price*", ou seja, neste *dataset* vamos calcular o preço com base no conjunto de dados presentes no *dataset*.

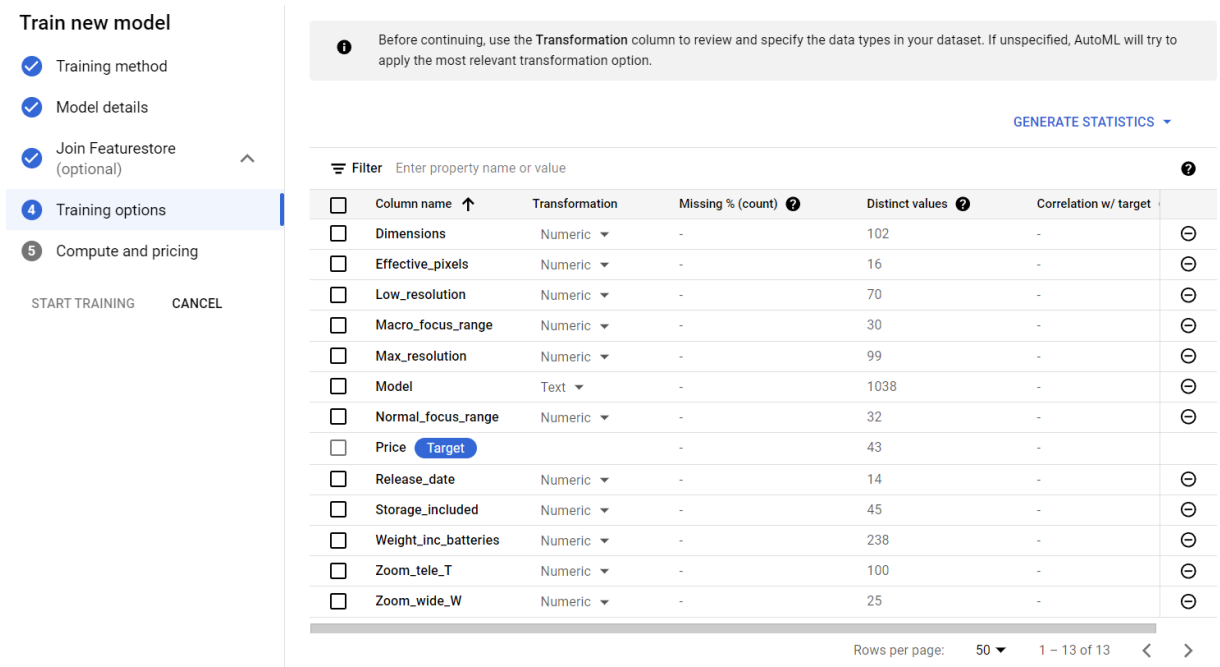

Figura 21 - *Vertex* AI - Treinar o modelo - Opção de treino

Agora na Figura *21*, temos uma visão geral do *dataset* e podemos alterar o tipo de dados de cada coluna, estes podem ser numéricos, texto, tempo e categórico, como também podem ser retirados colunas que sejam desnecessários para treinar o modelo.

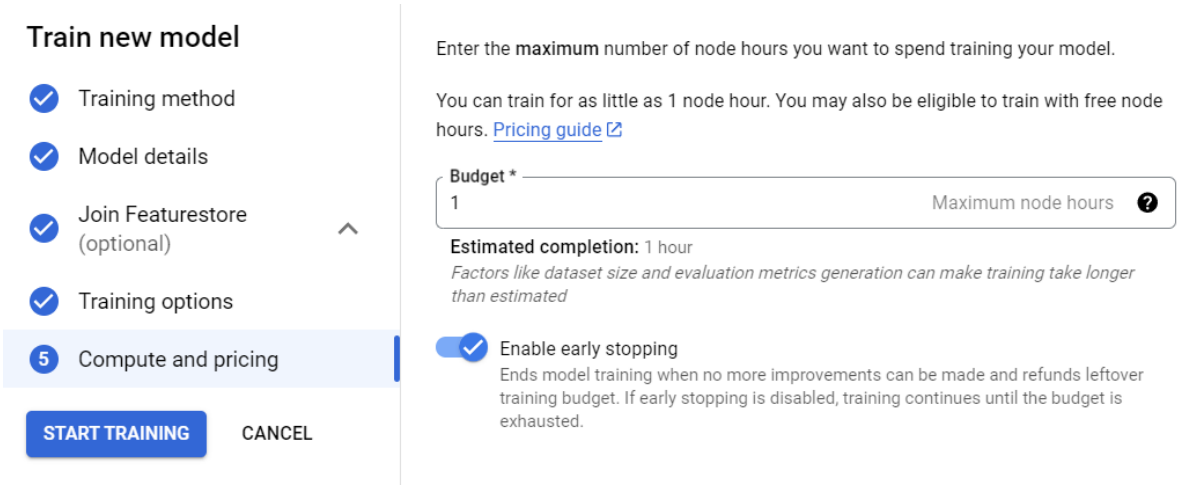

Figura 22 - *Vertex* AI - Treinar o modelo - Compilar

Finalmente, na Figura *22*, é escolhido o número de horas que o modelo vai treinar, como este modelo é de exemplo, foi introduzida o número mínimo de horas, ou seja, um (1). Quanto maior for o número de horas, mais probabilidade de os resultados serem mais precisos.

Como este modelo precisa de consumir tempo de processamento de outras máquinas, neste caso, da Google, para treinar é necessário 20 € ou 20 créditos (no início o utilizador tem ao dispor 280 créditos para gastar) para treinar, neste caso, uma hora. A plataforma avisa, enviando um email, quando o modelo acabou de treinar.

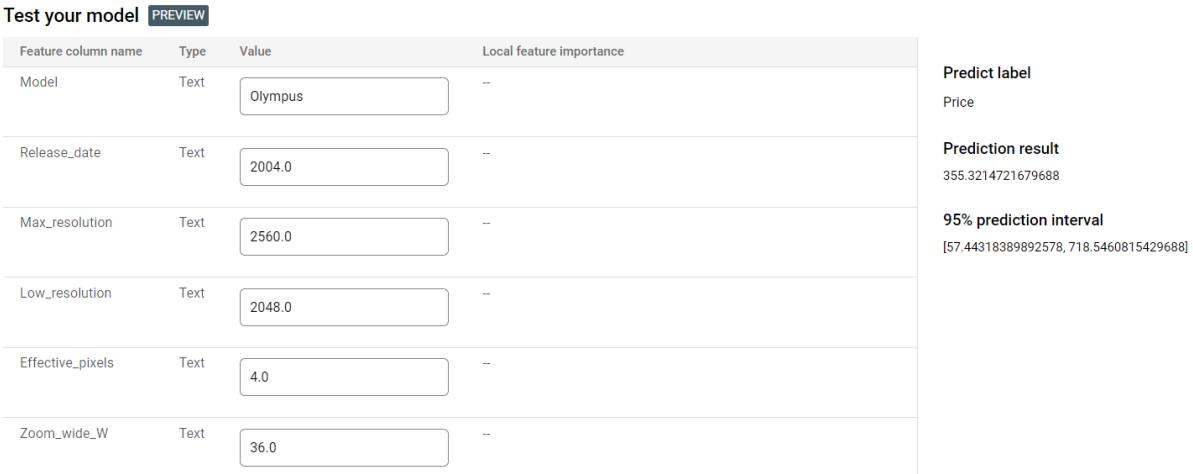

Figura 23 - *Vertex* AI - Treinar o modelo - Testar

E, finalmente, com o treino do modelo concluído, ele pode ser testado. Neste exemplo da Figura *23*, utiliza apenas um valor para o teste, mas no projeto real, utilizará uma quantidade muito maior de dados para avaliar a precisão do modelo. Como se pode verificar pela figura anterior, com os dados introduzidos da esquerda, foi possível prever um preço de "355.3214….", como não há forma de avaliar esta previsão, o valor do preço pode não estar correto.

1.3. Anexo III

Neste anexo, vai ser apresentado e explicado um script em *Python*, que tem como objetivo treinar o modelo e fazer previsões de um certo conjunto de dados, mais especificamente para a futura profissão. Além do objetivo anterior, este script contém o pré-processamento do *dataset* (etapa importante para a construção do modelo), a criação do mesmo e a criação, treinamento e previsão do modelo de ML.

Obs.: Este script foi executado no *Google Colab* [35], que é um serviço do *Jupyter Notebook* que fornece acesso gratuito a recursos de computadores, incluindo *GPUs e TPUs*. O *Colab* é especialmente utilizado para ML, ciência de dados e educação. Este código utiliza a versão 3.9 do *Python* [36]. Portanto, ao executá-lo em outros sistemas ou plataformas, recomenda-se utilizar preferencialmente o *Python* na versão 3.9.

Antes de apresentar o script, no GCP, tem de ser ativado os seguintes serviços (Figura *24* e Figura *25*).

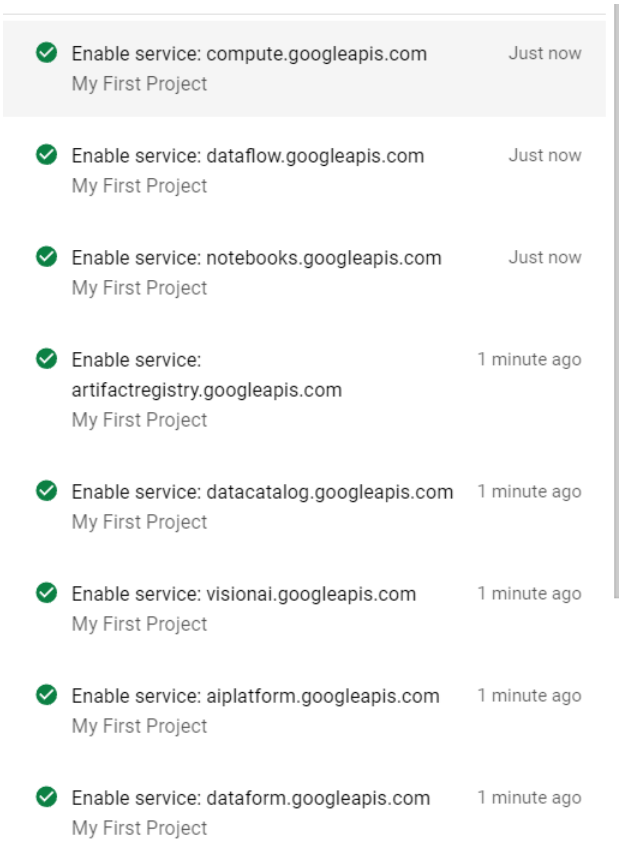

Figura 24 - Script - Serviços GCP 1/2

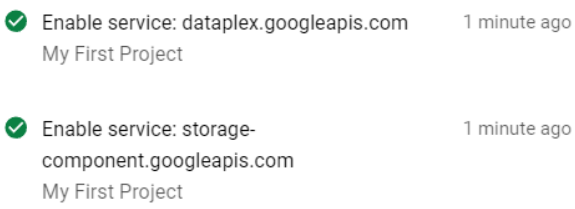

Figura 25 - Script - Serviços GCP 2/2

Estes serviços são necessários para permitir acesso a vários recursos que o GCP oferece através do uso de um script, explicado a seguir.

Os primeiros necessários são instalar as dependências necessárias para o funcionamento do script. A Figura *26* contém as dependências usadas.

```
#Dependências necessárias
!pip install google-cloud-aiplatform #API-GCP
!python -m spacy download en_core_web_md #Model NLP - Simulariedade entre palavras
!pip install --upgrade deepl #API-DeepL
!pip install langdetect #Google Language-Detection
```

Figura 26 - Script - Dependências

As dependências são necessárias para:

- '*!pip install google-cloud-aiplatform*' - Necessária para utilizar os recursos do GCP, desde do upload do conjunto de dados até à previsão do modelo.
- '*!python -m spacy download en_core_web_md*' - Necessária para utilizar o modelo pré-treinado do tipo NLP para encontrar a similaridade entre palavras.
- '*!pip install --upgrade deepl*' - Necessária para utilizar a API do *DeepL*.
- '*!pip install langdetect*' - Necessária para utilizar a API do *Google Language-Detection* para detetar a língua com base na palavra.

A Figura *27*, contém todas as bibliotecas usadas.

```
#Bibliotecas necessárias
from google.cloud import storage,aiplatform
import pandas as pd
import os
from google.protobuf import json_format
from google.protobuf.struct_pb2 import Value
import spacy
from langdetect import detect, DetectorFactory
import deepl
```

Figura 27 - *Script* - Bibliotecas

As bibliotecas são necessárias são:

- *'google.cloud.storage*' - Faz parte do SDK do *Google Cloud* e é usada para interagir com o GCS. Permite que manipular ficheiros armazenados em *buckets* na *cloud* do Google [37].
- *'google.cloud.aiplatform*' - Também faz parte do SDK do *Google Cloud*, é usada para interação com o Vertex AI. Esta permite manipular e implementar modelos de ML, entre outras tarefas relacionados com a AI [33].
- 'pandas' - É uma biblioteca de análise de dados em *Python*. Permite oferecer estruturas de dados como *DataFrames*, que facilitam a manipulação, análise e visualização de dados tabulares [38].
- 'os' - Permite interagir com o sistema operacional. Neste caso, é usada para armazenar *keys* para *APIs* [39].

- *'google.protobuf.json_format*' e 'google.protobuf.struct_pb2.Value' - Fazem parte do *Google Protocol Buffers* (*Protobuf*), que é um método de serialização de dados estruturados. *'json_format*' permite converter entre objetos JSON e mensagens *Protobuf* [40]. *'Value'* é uma estrutura que permite trabalhar com valores dinâmicos dentro de mensagens *Protobuf* [41].
- *'spacy'* - É uma biblioteca avançada de NLP [42].
- *'langdetect.detect*' e *'langdetect.DetectorFactory*' - Usada para detetar automaticamente a linguagem de um texto [43].
- *'deepl'* - Acesso a recursos da API do *DeepL* para o serviço de tradução automática do *DeepL* [30].

Para aceder aos recursos do GCP e do API do *DeepL* são necessárias chaves. Estas são obtidas através das plataformas mencionadas anteriormente. A chave do GCP é em JSON, logo é necessário fazer download do ficheiro (guardar num diretório acessível) e guardar o caminho para o ficheiro numa variável de ambiente, neste caso, foi utilizada uma variável global por se tratar de uma fase inicial do projeto. No entanto, para fazer corretamente, deve ser implementada como uma variável de ambiente. A Figura *28* contém o caminho para o ficheiro JSON do GCP.

```
key = "/content/maximal-emitter-433811-h4-b8f3169a9b41.json"
```

Figura 28 - Script - Chave - GCP

No caso da chave do *DeepL* é apenas um código, que pode ser obtido no site da plataforma. Este código deve ser guardado numa variável de ambiente do sistema operativo.

Depois, para desenvolver um código mais limpo, certos valores vão ser usados muitos vezes ao longo do script, como:

- ID do projeto no *Vertex* AI
- Localização do projeto no *Vertex* AI
- Nome do *Cloud Storage* no *Vertex* AI
- Nome do ficheiro no *Vertex* AI
- URI do *Cloud Storage* no *Vertex* AI
- Caminho para o conjunto de dados no *Cloud Storage*
- API *endpoint* do GCP
- Nome do *dataset*
- Nome do modelo
- ID do *dataset*
- ID do *endpoint*

Para melhorar a segurança, estas variáveis globais têm de ser variáveis de ambiente, porque contém informações sensíveis, mas como referenciado anteriormente, é uma fase inicial do projeto então entendeu-se que não fosse necessário desenvolver já essa parte.

Depois para iniciar as conexões com os serviços do GCP, a API do *DeepL* e o detetor da linguagem, é preciso o seguinte código na Figura *29*.

```
credentials = service_account.Credentials.from_service_account_file(key)

#Inicializar a conexão com a GCP
aiplatform.init(
    project=project_id,
    location=location,
    staging_bucket=gcs_uri,
    credentials=credentials,
)

#Inicializar a conexão com o DeepL
translator = deepl.Translator(key_deepl)

#Inicializar o detector de linguagem
DetectorFactory.seed = 0
```

Figura 29 - Script - Inicialização

Inicialmente, o script vai procurar as credenciais da conta associada no GCP utilizando a chave em JSON, referenciado anteriormente. Depois vai utilizar o ID do projeto, a localização, o URI do *Cloud Storage* no GCP e as credenciais obtidas anteriormente. Ainda na Figura *29*, é iniciada a conexão com a API do *DeepL* através da chave associada à conta da *DeepL*.

A seguir vem a primeira fase do treino, o pré-processamento de um conjunto de dados, apresentado na Tabela 6. Para realizar esta fase, além das variáveis globais, é necessário também o caminho para o ficheiro CSV que contém os dados para processamento e transformação para *dataset*, Figura *30*.

```
upload_data('/content/contracts_professional_path_realhotelsgroup_21082024_1049.csv')
```

Figura 30 - Script - Caminho para o conjunto de dados

A Figura *31*, inicia o processo do upload e processamento do conjunto de dados da Tabela 6.

```
def upload_data(source_file_name):
  storage_client = storage.Client.from_service_account_json(key)
  bucket = storage_client.bucket(bucket_name)
  blob = bucket.blob(destination_file_name)

  df = pd.read_csv(source_file_name, delimiter=';') #Lê o fichei
```

Figura 31 - Script - Upload e Processamento - Inicio das conexões

O script começa por obter acesso ao *bucket* através da chave em JSON, que também foi usada para obter as credenciais anteriormente. Depois vai buscar a referência ao *bucket* através do nome do mesmo e por fim cria uma referência dentro desse *bucket* para poder criar, remover, editar, neste caso um ficheiro. E por fim, o ficheiro CSV vai ser transformado em um *dataframe* para ser transformado em um *dataset* que possa ser devidamente treinado.

A Figura *32* contém a renomeação das colunas, o GCP não permite espaços vazios, caracteres especiais e pontuação, então os espaços vazios foram substituídos por um *underscore* e os caracteres especiais e a pontuação foram retirados ou substituídos, como se pode visualizar pela figura seguinte.

```
df.rename(columns={
      'Especialidade (Percurso Profissional)': 'Especialidade_Percurso_Profissional',
      'Data Saída (Contrato)': 'Data_Saida_Contrato',
      'Data Inicial (Percurso Profissional)': 'Data_Inicial_Percurso_Profissional',
      'Nº Col. (Contrato)': 'Num_Col_Contrato',
      'Departamento/Equipa/Grupo (Percurso Profissional)': 'Departamento_Equipa_Grupo_Percurso_Profissional',
      'Data Inicial (Contrato)': 'Data_Inicial_Contrato',
      'Regime Reforma Aplicado (Contrato)': 'Regime_Reforma_Aplicado_Contrato',
      'Chefia (Percurso Profissional)': 'Chefia_Percurso_Profissional',
      'Entidade Empregadora (Contrato)': 'Entidade_Empregadora_Contrato',
      '#': 'Numero',
      'PARENT_ID (Contrato)': 'PARENT_ID_Contrato',
      'Data Saída (Percurso Profissional)': 'Data_Saida_Percurso_Profissional',
      'Tipo Contrato (Contrato)': 'Tipo_Contrato_Contrato',
      'Profissão (Percurso Profissional)': 'Profissao_Percurso_Profissional',
      'Função (Percurso Profissional)': 'Funcao_Percurso_Profissional',
      'USER_ID (Contrato/Percurso Profissional)': 'USER_ID_Contrato_Percurso_Profissional',
      'Secção (Percurso Profissional)': 'Seccao_Percurso_Profissional',
      'Unidade de Negócio (Percurso Profissional)': 'Unidade_Negocio_Percurso_Profissional',
      'User Locality':'User_Locality',
      'User Birthdate':'User_Birthdate'
  }, inplace=True)
```

Figura 32 - Script - Upload e Processamento - Renomeação das colunas

Agora começa a transformação do conjunto de dados para *dataset*. Primeiro, vai ser criado uma coluna nova, esta vai indicar com que idade é que o colaborador começou o seu contrato, ou seja, começou a trabalhar naquela profissão, isso pode ser um dado importante para o modelo.

```
#Conversão para datetime
df['User_Birthdate'] = pd.to_datetime(df['User_Birthdate'])
df['Data_Inicial_Contrato'] = pd.to_datetime(df['Data_Inicial_Contrato'])

#Calculo da idade no inicio do contrato e criação de uma nova coluna que guarda a idade
df['Idade_No_Inicio_Contrato'] = df.apply(lambda row: (row['Data_Inicial_Contrato'] - row['User_Birthdate']).days // 365, axis=1)
```

Figura 33 - Script - Upload e Processamento - Coluna da idade

Na Figura *33*, é usada as colunas "*User_Birthdate*" e "*Data_Inicial_Contrato*" que tem informações da data de nascimento e a data inicial do contrato, respetivamente, para calcular a idade do colaborador no início do contrato. Inicialmente, é feita a conversão dos dados das colunas para o formato *datetime* para depois ser utilizada para o cálculo da idade, que é a data inicial do contrato menos a data de nascimento, o resultado vai ser o número de dias da diferença de datas, que finalmente é dividida por 365 para descobrir a idade do colaborador. E finalmente é armazenada esse valor numa nova coluna denominada "*Idade_No_Inicio_Contrato*".

Agora, alguns dados não são necessários como dados referentes a outras colunas, ou colunas que têm muitas instâncias vazias ou colunas que foram usadas para calcular outros dados, vão ser removidos do *dataset*.

```
columns_delete = ['ID',
                  'PARENT_ID_Contrato',
                  'Num_Col_Contrato',
                  'Regime_Reforma_Aplicado_Contrato',
                  'Tipo_Contrato_Contrato',
                  'Especialidade_Percurso_Profissional',
                  'Funcao_Percurso_Profissional',
                  'Seccao_Percurso_Profissional',
                  'Unidade_Negocio_Percurso_Profissional',
                  'Chefia_Percurso_Profissional',
                  'Data_Inicial_Contrato',
                  'Data_Saida_Contrato',
                  'Data_Saida_Percurso_Profissional',
                  'Data_Inicial_Percurso_Profissional',
                  'Departamento_Equipa_Grupo_Percurso_Profissional',
                  'Numero',
                  'USER_ID_Contrato_Percurso_Profissional',
                  'User_Birthdate',
                  ]
df = df.drop(columns=columns_delete)
```

Figura 34 - Script - Upload e Processamento - Colunas removidas

Como se pode verificar pela Figura *34*, as colunas removidas são:

- *ID*
- *PARENT_ID_Contrato*
- *Num_Col_Contrato*
- *Regime_Reforma_Aplicado_Contrato*
- *Tipo_Contrato_Contrato*
- *Especialidade_Percurso_Profissional*
- *Funcao_Percurso_Profissional*
- *Seccao_Percurso_Profissional*
- *Unidade_Negocio_Percurso_Profissional*
- *Chefia_Percurso_Profissional*
- *Data_Inicial_Contrato*
- *Data_Saida_Contrato*
- *Data_Saida_Percurso_Profissional*
- *Data_Inicial_Percurso_Profissional*
- *Departamento_Equipa_Grupo_Percurso_Profissional*
- *Numero*
- *USER_ID_Contrato_Percurso_Profissional*
- *User_Birthdate*

A seguinte coluna, da Figura *35*, que vai ser criada é a coluna denominada "*Target column*" [31] (em português "coluna alvo"), esta é a *feature* que o modelo quer descobrir. Na maioria das situações é utilizado para modelos com um algoritmo do tipo "Supervisionado" (que é o caso).

```
df['Proximo_Emprego'] = df['Profissao_Percurso_Profissional'].shift(-1)
df.at[len(df) - 1, 'Proximo_Emprego'] = None
```

Figura 35 - Script - Upload e Processamento - Coluna "*Proximo_Emprego*"

O código da Figura *35*, vai criar uma coluna denominada "*Proximo_Emprego*" que contém o emprego que aparece na linha seguinte para cada linha do conjunto de dados. Para a última linha, onde não há um próximo emprego, o valor é definido como "*None*". Só é possível desenvolver do modo anterior, porque o conjunto de dados contém o seguinte formato, apresentado na Figura *36*.

Segurança Social				
	Human Resources	Shared Services		HR Supervisor ? Talent & Development
	Human Resources	Shared Services		HR Supervisor ? Talent & Development
Segurança Social				
	RBV OPCs	Real Bellavista	RBV OPCs Entertainment	Entertainment Supervisor
			Animação	Chefe de Animação
			Animação	Chefe de Animação
			Animação	Chefe de Animação
	RBV OPCs	Real Bellavista	RBV OPCs Entertainment	Entertainment Supervisor
	RBV OPCs	Real Bellavista	RBV OPCs Entertainment	Entertainment Supervisor
	RBV OPCs	Real Bellavista	RBV OPCs Entertainment	Entertainment Supervisor
	RBV OPCs	Real Bellavista	RBV OPCs Entertainment	Entertainment Supervisor

Figura 36 - Script - Upload e Processamento - Formato do conjunto de dados

Tem o seguinte formato:

- 1º Linha (Regime = "Segurança Social") - Início do colaborador
- 2º e 3º Linhas (Regime = "") - Percurso profissional do colaborador

Então, com base na Figura *36*, o que vai acontecer é:

- Criar uma coluna "*Proximo_Emprego*"
- 1º Linha vai ter ("*None*" e "*HR Supervisor*")
- 2º Linha vai ter ("*HR Supervisor*" e "*HR Supervisor*")
- 3º Linha vai ter ("*HR Supervisor*" e "*None*")

Depois como não se pode treinar um modelo com conjunto de dados com instâncias vazias ou com valores nulos ou vazios, as instâncias que contêm valores nulos ou vazios vão ser removidos.

```
df.dropna(subset=['Profissao_Percurso_Profissional', 'Proximo_Emprego'], inplace=True)
```

Figura 37 - Script - Upload e Processamento - Remoção de instâncias com valores nulos

Como se pode verificar pela Figura *37*, vai ser percorrido as colunas "*Profissao_Percurso_Profissional*" e "*Proximo_Emprego*" para encontrar valores nulos ou vazios, que posteriormente as instâncias serão removidas. Logo o exemplo anterior da Figura *36*, apenas a 2º linha ficará, já que as 1º e 3º linhas contêm valores nulos.

O código seguinte foi adicionado depois da ocorrência de um erro no treinamento do modelo. Este erro impede de treinar o modelo quando o *dataset* não contem no mínimo 10 valores do mesmo tipo, ou seja, se houver 5 valores do tipo "*waiter*" vai ocorrer esse erro, denominado "*Train set should contain all labels*".

```
min_occurrences = 10 #Minimo de ocorrências
class_distribution = df['Proximo_Emprego'].value_counts()
classes_to_keep = class_distribution[class_distribution >= min_occurrences].index
df = df[df['Proximo_Emprego'].isin(classes_to_keep)]
```

Figura 38 - Script - Upload e Processamento - Número insuficiente de labels

Para solucionar este problema, se um valor de uma profissão não existir pelo menos 10 vezes em todo o conjunto de dados, as instâncias relativas a esse valor serão eliminadas.

Outro problema que foi encontrado, foi o nome das profissões não estarem numa linguagem formalizada, algumas estão em inglês outras estão em português. A solução foi traduzir tudo para inglês, foi inicialmente utilizado o google tradutor, mas verificou-se que é ineficaz em algumas traduções, então foi utilizado o *DeepL* que oferece por mês 500.000 caracteres para tradução.

```
unique_values = pd.concat([df['Profissao_Percurso_Profissional'], df['Proximo_Emprego']]).unique()

#Traduz todos as profissões para inglês
translations_professions = translation(unique_values)
```

Figura 39 - Script - Upload e Processamento - Processo de tradução

Na Figura *39*, inicialmente realizada a busca por todos os valores únicos de profissões presentes nas tabelas "*Profissao_Percurso_Profissional*" e "*Proximo_Emprego*" que depois vão ser traduzidos utilizando a seguinte função.

```
#Traduz, neste caso, as profissões
#Retorna um dicionário com a palavra original associada à palavra traduzida
def translation(texts):
  translations = {}
  for text in texts:
    try:
      #language = detect(text)

      result = translator.translate_text(text, target_lang="EN-GB")
      translations[text] = result.text

    except Exception as e:
      print(f'Erro na tradução de {text}: {e}')
      translations[text] = None

  return translations
```

Figura 40 - Script - Upload e Processamento - Tradução

A função da Figura *40*, vai percorrer todos valores únicos da profissão, e vai utilizar o método oferecido pela *DeepL* para traduzir texto e por fim, mete o texto traduzido associado ao texto original (antes da tradução) num dicionário. Caso ocorra alguma exceção onde não foi possível fazer a tradução, o valor da tradução é nulo.

```
df['Profissao_Percurso_Profissional'] = df['Profissao_Percurso_Profissional'].map(translations_professions)
df['Proximo_Emprego'] = df['Proximo_Emprego'].map(translations_professions)
```

Figura 41 - Script - Upload e Processamento - Atualizar as profissões

Na Figura *41*, os valores da profissão que estão nas colunas "*Profissao_Percurso_Profissional*" e "*Proximo_Emprego*" vão ser substituídos pelos valores traduzidos anteriormente. Para o caso de originar valores traduzidos nulos, vai ser utilizado o mesmo método da Figura *37* para a remoção dos mesmos.

Finalmente, o *dataframe* que foi utilizado para fazer todas as alterações anteriores é transformado para um ficheiro CSV, posteriormente irá para o *Cloud Storage* do GCP, como é comprovado na Figura *42*.

```
df.to_csv(source_file_name, index=False)
blob.upload_from_filename(source_file_name)
```

Figura 42 - Script - Upload e Processamento - Upload do conjunto de dados

O resultado deve ser o seguinte (Figura *43*).

Figura 43 - Script - Upload e Processamento - Resultado

Já com o conjunto de dados no *Cloud Storage*, o passo seguinte é a criação do *dataset*. Foi utilizado o código de uma amostra da documentação pertencente ao GCP [44].

```
def create_dataset(project_id, display_name, gcs_uri, location, api_endpoint, timeout: int = 300):
  client_options = {"api_endpoint": api_endpoint}

  # Use service account credentials
  credentials = service_account.Credentials.from_service_account_file(key)

  client = aiplatform_v1.DatasetServiceClient(client_options=client_options, credentials=credentials)
  metadata_dict = {"input_config": {"gcs_source": {"uri": [gcs_uri]}}}
  metada = json_format.ParseDict(metadata_dict, Value())

  dataset = {
      "display_name": display_name,
      "metadata_schema_uri": "gs://google-cloud-aiplatform/schema/dataset/metadata/tabular_1.0.0.yaml",
      "metadata": metada
  }

  parent = f"projects/{project_id}/locations/{location}"
  response = client.create_dataset(parent=parent, dataset=dataset)
  create_dataset_response = response.result(timeout=timeout)
```

Figura 44 - Script - Criar o Dataset

A função anterior destina-se a criar um *dataset* no GCP. Esta usa a chave em JSON para obter as credenciais de acesso á plataforma, depois é criado uma instância do cliente para interagir com o serviço de conjuntos de dados no GCP. Na configuração dos meta dados é construído um dicionário que descreve a configuração de entrada do conjunto de dados, especificamente a fonte de dados no GCP (*'gcs_uri'*). Este dicionário é depois convertido num objeto *'Value*()' do Google *protobuf* para ser utilizado na criação do conjunto de dados.

Na definição do *dataset* é determinado o nome do dataset, o URI que aponta para o esquema de meta dados que descreve o tipo de conjuntos de dados utilizado (neste caso, um conjunto de dados tabular) e os meta dados que descrevem a fonte de dados, que foram construídos no passo anterior.

Depois é construído o identificador completo (*'parent'*) que especifica o projeto e a localização no GCP onde o conjunto de dados será armazenado.

Finalmente, envia o pedido para criar o conjunto de dados. A função *'create_dataset'* (diferente da função *'create_dataset*' principal) do cliente envia o pedido, e *'response.result(timeout=timeout)*' aguarda que a operação seja concluída dentro do tempo especificado. Após a criação do *dataset*, já se pode começar a desenvolver o modelo.

```
def create_model():
  dataset = aiplatform.TabularDataset(dataset_name=dataset_id)

  job = aiplatform.AutoMLTabularTrainingJob(
      display_name="model_automl_upload",
      optimization_prediction_type="classification",
      column_transformations=[
          {"categorical":{"column_name": "Entidade_Empregadora_Contrato"}},
          {"numeric":{"column_name": "Idade_No_Inicio_Contrato"}},
          {"categorical":{"column_name": "Profissao_Percurso_Profissional"}},
          {"categorical":{"column_name": "Proximo_Emprego"}},
          {"categorical":{"column_name": "User_Locality"}},
          ],
      )

  model = job.run(
      dataset = dataset,
      target_column="Proximo_Emprego",
      training_fraction_split=0.8,
      validation_fraction_split=0.1,
      test_fraction_split=0.1,
      model_display_name="model_upload",
      disable_early_stopping=False,
  )

  endpoint = model.deploy(
      machine_type="e2-standard-4",
  )
```

Figura 45 - Script - Criar o modelo

O código anterior da Figura *45*, foi desenvolvido com o auxílio de um tutorial do *Geek For Geeks* [45], nele ajuda a construir o modelo, bem como prever resultados. Em relação ao código, começa por ir procurar o *dataset* que foi criado anteriormente, através do ID. Depois começa a definição do tipo de modelo que vai ser desenvolvido, neste caso é utilizado o tipo de treino "*AutoML*" [46] que "democratiza" o acesso ao ML, permitindo que um público mais amplo crie e implemente modelos sofisticados sem a necessidade de uma compreensão profunda dos detalhes técnicos. Na fase de definição do modelo, é possível especificar o nome do modelo, o tipo de algoritmo (neste caso, de classificação), e o tipo de dados das colunas, que pode ser categórico, numérico, *timestamp* ou *datetime*. Também é necessário definir a coluna alvo e as percentagens de divisão do *dataset* em três partes: treino, validação e teste. Além disso, é importante determinar o tipo de máquina em que o modelo será processado, sendo a "e2-standard-4" a opção padrão. Com essas configurações, o *endpoint* ficará ativo e pronto para uso.

Agora chegou parte final do script, onde é realizada a previsão de resultados. Para obter previsões é preciso especificar o id do *endpoint*, o nome do projeto, a localização do projeto e as instâncias que são efetivamente o que é para prever.

```python
def model_prediction(endpoint_id, project, location, instances):
  endpoint = aiplatform.Endpoint(endpoint_id, project=project, location=location)

  predictions = endpoint.predict(instances=instances)

  return predictions
```

Figura 46 - Script - Previsão - Algoritmo

Como se pode verificar pela Figura *46*, primeiro é obtido o *endpoint* através do id do mesmo, o nome e a localização do projeto. A Figura *47* contém as instâncias que vão ser previstas.

```python
instances = [
    {"Entidade_Empregadora_Contrato": "Real Hotels Group",
     "Profissao_Percurso_Profissional": "Waiter",
     "User_Locality": "RIO DE MOURO",
     "Idade_No_Inicio_Contrato": "40"
     },

    {"Entidade_Empregadora_Contrato": "New Palm",
     "Profissao_Percurso_Profissional": "Steward",
     "User_Locality": "LISBOA",
     "Idade_No_Inicio_Contrato": "34"
     },
]

predictions = model_prediction(endpoint_id, project_id, location, instances)
```

Figura 47 - Script - Previsão - Instâncias

As instâncias anteriores estão em JSON. A seguir na Figura *48* contém as previsões obtidas.

```python
for prediction in predictions.predictions:
  #Combina o nome das classes com os scores associada à classe
  combined = list(zip(prediction['classes'], prediction['scores']))

  #Ordena as previsões pela pontuação (score) em ordem decrescente
  combined_sorted = sorted(combined, key=lambda x: x[1], reverse=True)

  #Seleciona as três melhores previsões
  top_three = combined_sorted[:3]

  print("Top 3 predictions:")
  for cls, score in top_three:
      print(f"Class: {cls}, Score: {score}")
  print("\n")
```

Figura 48 - Script - Previsão - Apresentação das previsões

Para apresentar as previsões das instâncias, é necessário um ciclo (apesar de ser só duas neste caso) depois é criada um objeto do tipo *'list'* para combinar o nome da classe (*eg. waiter, stewart*) e a pontuação dessa classe (de 0 a 1). Depois é ordenado as previsões em ordem

decrescente (maior pontuação para menor pontuação) e é apresentado as três melhores pontuações para cada instância.

```
Top 3 predictions:
Class: Waiter, Score: 0.5734809637069702
Class: HR Executive - Payroll & Benefits, Score: 0.1212331280112267
Class: Senior Waiter, Score: 0.03888529166579247


Top 3 predictions:
Class: Steward, Score: 0.7378037571907043
Class: Waiter, Score: 0.2159344255924225
Class: Front Office Agent, Score: 0.007727602496743202
```

Figura 49 - Script - Previsão - Resultado das previsões

Como se pode verificar pela Figura *49*, a primeira instância que tinha como profissão inicial "*Waiter*", tem como maior probabilidade de ser "*Waiter*" com 0.57, depois vem o "HR *Executive - Payroll & Benefits*" com 0.12 e o "*Senior Waiter*" com 0.03. A outra instância que tem como profissão inicial "*Steward*", a sua maior pontuação é a mesma profissão "*Steward*" com 0.73, depois vem o "*Waiter*" com 0.21 e o "*Front Office Agent*" com 0.007. Pode concluir-se vários aspetos em relação aos resultados anteriores, o primeiro é o colaborador tem mais probabilidade de continuar na mesma profissão do que mudar ou progredir na carreira, a outra é no caso do "*Waiter*" existe uma pequena probabilidade de ele progredir de "*Waiter*" para "*Senior Waiter*", isto pode ser um bom indicador que o modelo está a funcionar corretamente (apesar de não ser a 1 ou 2 profissão com maior pontuação).

Para finalizar, se o *endpoint* estiver sempre ativo gasta 20 créditos por dia, então a solução que está prevista para este projeto é realizar todo o processo anterior deste anexo e em vez de ser apenas duas instâncias de colaboradores, realizar as previsões de todos os colaboradores, guardar e caso os colaboradores queiram ver a previsão, basta ir buscar a previsão que está guardada. Assim, em vez de o *endpoint* estar sempre ativo e a gastar 20 créditos por dia, quando um colaborador quer, basta ir procurar onde foi guardado essa previsão. A Figura *50* mostra como desativar um *endpoint*.

```python
def model_undeploy(endpoint_id):
  endpoint = aiplatform.Endpoint(endpoint_id)

  deployed_models = endpoint.list_models()

  for deployed_model in deployed_models:
    endpoint.undeploy(deployed_model_id=deployed_model.id)

model_undeploy(endpoint_id)
```

Figura 50 - Script - Previsão - Desativar o *endpoint*

Na Figura *50*, para desativar um *endpoint*, basta especificar o id do *endpoint* e com ele, desativa os modelos associados ao *endpoint*.

1.4. Anexo IV

Este anexo apresenta a avaliação do modelo ML, relativo ao conjunto de dados do anexo III criado e treinado no GCP. Na Figura *51*, apresenta as métricas de avaliação para os modelos do tipo de classificação (cada tipo de modelo tem as suas métricas especificas) que são: A Precisão, o *Recall*, o *F1-Score*, a Curva *ROC* e a Exatidão.

PR AUC	0.72
ROC AUC	0.985
Log loss	0.92
Micro-average F1	0.75611967
Macro-average F1	0.47049782
Micro-average precision	78.1%
Micro-average recall	73.3%

Figura 51 - Avaliação do modelo - Futura Profissão - Métricas

Cada métrica avalia uma parte do modelo, como [47]:

- Exatidão ('*Accuracy*' ou PR AUC) - Mede a proporção de previsões corretas feitas pelo modelo.
- Precisão ('*Precision*' ou *Micro-average precision*) - Mede a proporção de previsões positivas feitas pelo modelo que estão corretas.
- *Recall* (*Micro-average recall*) - Mede a proporção de exemplos positivos que foram corretamente identificados pelo modelo.
- F1-Score - A média micro considera o total de verdadeiros positivos, falsos positivos e falsos negativos em todas as classes. A média macro calcula a média de cada classe individualmente, sem ponderação pela frequência das classes.
- Curva ROC (ROC AUC) - Avalia o desempenho de modelos de classificação binária em diferentes limites de decisão. Mede a área sob a curva da Taxa de Verdadeiros Positivos (*Recall*) em função da Taxa de Falsos Positivos. Quanto maior a curva, melhor o modelo está em separar as classes.
- Com base nos indicadores fornecidos, o modelo pode ser analisado da seguinte forma:
- Exatidão (0.72) - Embora ainda seja um valor aceitável, é consideravelmente menor que a Curva ROC. Isto pode sugerir que o modelo não está a manter uma Precisão e *Recall* tão altas quanto indicaria a Curva ROC.
- Precisão (78.1%) - O valor é razoável, indica que a maioria das previsões do modelo estão corretas.
- *Recall* (73.3%) - O valor sugere que o modelo está a identificar cerca de 73% das verdadeiras instâncias positivas, perdendo cerca de 27% das mesmas.
- Micro F1-Score (0.756) - Indica um bom equilíbrio entre precisão e *recall* global do modelo.

- Macro F1-Score (0.470) - Sugere que o desempenho do modelo varia significativamente entre classes, possivelmente indica que está bem nas classes maioritárias, mas com dificuldade nas classes minoritárias.
- Curva ROC (0.985) - O valor extremamente alto indica que o modelo possui uma excelente capacidade de distinguir entre as classes positivas e negativas. Em termos práticos, há uma probabilidade de 98,5% de o modelo classificar corretamente uma amostra positiva em relação a uma negativa escolhida aleatoriamente.

Também existe a métrica *Log Loss* que mede a incerteza das previsões do modelo, um valor mais baixo é desejável, indicando que as probabilidades previstas estão próximas dos valores reais. Um valor de 0.92 pode indicar que, embora o modelo esteja certo na classificação, as previsões não altamente confiantes.

Em conclusão, o modelo demonstra excelente capacidade de discriminação entre classes (curva ROC alta) e uma boa precisão e *recall* geral. A diferença significativa entre as médias F1 micro e macro sugere que o modelo pode estar enfrentando desafios com classes menos representadas, isto indica um desequilíbrio nas classes do conjunto de dados. O modelo apresenta um desempenho sólido em termos de discriminação geral, mas há espaço para melhorias, mais especificamente no tratamento de classes minoritárias e na calibração das previsões probabilísticas.

1.5. Anexo V

Este anexo contém o script usado para o desenvolvimento do objetivo relativo a "Antecipar possíveis saídas". Maior parte do código usado neste script é similar ao do Anexo III, logo não vai ser explicado novamente, só o que difere do código do Anexo III. A Figura *52*, contém o pré-processamento do conjunto de dados, para o objetivo referido anterior.

```
#Valores nulos -> Data atual
df['Data_Saida_Contrato'].fillna(date.today().strftime('%Y-%m-%d'), inplace=True)

#Há valores como "9999-12-31" que não são validos, vão ser substituidos pela data atual
#print(df['Data_Saida_Contrato'][df['Data_Saida_Contrato'].str.contains('9999-12-31')])
df['Data_Saida_Contrato'].replace('9999-12-31', date.today().strftime('%Y-%m-%d'), inplace=True)

#Conversão para datetime
df['Data_Saida_Contrato'] = pd.to_datetime(df['Data_Saida_Contrato'])
df['Data_Inicial_Contrato'] = pd.to_datetime(df['Data_Inicial_Contrato'])

#Data final - Data inicial = x dias = valor de 0 a 1
df['Duracao_Contrato_Dias'] = (df['Data_Saida_Contrato'] - df['Data_Inicial_Contrato']).dt.days
#df['Duracao_Contrato'] = df['Duracao_Contrato_Dias'] / df['Duracao_Contrato_Dias'].max()
#print(df['Duracao_Contrato'])

#Saiu ou não Saiu?
#Vai buscar a proxima profissão
df['Proxima_Profissao'] = df['Profissao_Percurso_Profissional'].shift(-1)
#Verifica se a profissão seguinte é igual a profissão atual 0=nao,1=sim
df['Mudou_Profissao'] = (df['Profissao_Percurso_Profissional'] != df['Proxima_Profissao'])
#print(df[['Profissao_Percurso_Profissional', 'Proxima_Profissao', 'Mudou_Profissao']])

df.dropna(subset=['Profissao_Percurso_Profissional','Proxima_Profissao'], inplace=True)
```

Figura 52 - Script - Processamento de dados

Primeiramente, o script vai preencher os valores vazios ou nulos ou valores como "9999-12-31" da coluna "*Data_Saida_Contrato*" com a data atual, estes valores têm de ser preenchidos porque muitos dos contratos não contêm a data final do contrato ou têm valores inválidos. Em seguida, os valores das colunas *'Data_Saida_Contrato'* e *'Data_Inicial_Contrato'* são convertidos para o formato *datetime*, permitindo o cálculo da diferença entre a data final e a data inicial, a fim de determinar a quantidade de dias em que o contrato esteve vigente e posteriormente é guardado numa coluna denominada "*Duracao_Contrato_Dias*". Por fim, é criado a coluna alvo denominada "*Mudou_Profissao*", que indica se o colaborador mudou de profissão com valores "*True*" ou "*False*", para isso verifica se a profissão atual é igual á profissão seguinte do mesmo colaborador, se não for igual quer dizer que mudou de profissão devolve o valor "*True*" se for igual quer dizer que não mudou logo "False".

O resto do processo desde o processamento até à criação e treino do modelo é igual, à exceção das colunas a serem treinadas: "*Mudou_Profissao*" em vez de "*Proximo_Emprego*" e "*Duracao_Contrato_Dias*" em vez de "*Idade_No_Inicio_Contrato*".

1.6. Anexo VI

Este anexo diz respeito à avaliação do modelo relativo a "Antecipar Saídas". A Figura *53*, apresenta as métricas de avaliação para os modelos do tipo de classificação (cada tipo de modelo tem as suas métricas especificas) que são: A Precisão, o *Recall*, o F1-Score, a Curva ROC e a Exatidão.

PR AUC	0.916
ROC AUC	0.923
Log loss	0.352
Micro-average F1	0.85037035
Macro-average F1	0.7023563
Micro-average precision	85%
Micro-average recall	85%

Figura 53 - Avaliação do modelo - Antecipar Saídas - Métricas

Relativamente às métricas anteriores:

- Exatidão ('*Accuracy*' ou PR AUC) (0.922) - Um valor elevado, indica que o modelo tem um bom desempenho ao lidar com classes não balançadas.
- Curva ROC (ROC AUC) (0.925) - Este valor sugere que o modelo tem um bom desempenho na separação de classes.
- *Log Loss* (0.346) - Valor baixo, sugere que o modelo faz previsões precisas e não se desvia muito das classes verdadeiras.
- *Macro-average* F1 (0.7157) - Indica que algumas classes têm um desempenho inferior comparado com outras, provavelmente as classes minoritárias.
- *Micro-average* F1 (0.8637) - Este valo indica um bom equilíbrio entre precisão e recall para todas as classes.
- *Micro-average precision* (85%) - 85% das precisões estão corretas.
- *Micro-average recall* (85%) - 85% significa que o modelo está a recuperar corretamente 85% das instâncias positivas verdadeiras.

Em geral, o modelo parece ter um desempenho muito bom, com métricas de AUC elevadas e boas médias de precisão e *recall*.

1.7. Anexo VII

Nos anexos V e VII, faltou representar como os dados iriam ser previstos e guardados, neste caso, é do anexo V, mas o do anexo III não difere muito. A Figura *54* e a Figura *55*, contêm esse processo.

```
def save_in_csv(source_file_name,endpoint_id):
  df = pd.read_csv(source_file_name, delimiter=';')

  df = data_processing_prev(df)

  user_ids = df['USER_ID_Contrato_Percurso_Profissional'].tolist()

  df = df.drop(columns=['USER_ID_Contrato_Percurso_Profissional'])

  instances = df.to_dict(orient='records')
  #print(instances)

  for instance in instances:
    instance['Duracao_Contrato_Dias'] = str(instance['Duracao_Contrato_Dias'])

  predictions = model_prediction(endpoint_id, project_id, location, instances)
```

Figura 54 - Script - Guardar em CSV - 1/2

Na Figura *54*, começa por ler o ficheiro CSV que vai ser usado para previsões depois o mesmo vai ser processado. Depois os *IDs* dos utilizadores são guardados numa lista à parte e é eliminado do conjunto de dados. Posteriormente, é calculado as previsões com base no conjunto de dados anteriormente processado.

```
resultados = []

for user_id, prediction in zip(user_ids, predictions.predictions):
  #Combina o nome das classes com os scores associada à classe
  combined = list(zip(prediction['classes'], prediction['scores']))

  #Ordena as previsões pela pontuação (score) em ordem decrescente
  combined_sorted = sorted(combined, key=lambda x: x[1], reverse=True)

  #Seleciona as três melhores previsões
  top_three = combined_sorted[:3]

  resultado = {'USER_ID_Contrato_Percurso_Profissional': user_id}
  for i, (cls, score) in enumerate(top_three, 1):
    resultado[f'Saiu?'] = cls
    resultado[f'Probabilidade'] = score
  resultados.append(resultado)

df_resultados = pd.DataFrame(resultados)
df_resultados.to_csv('/content/resultados2_previsoes.csv', index=False)
```

Figura 55 - Script - Guardar em CSV - 2/2

Por fim na Figura *55*, vai percorrer todas as previsões e guardar as mesmas num ficheiro CSV juntamente com os *IDs* dos utilizadores guardados anteriormente, o resultado é o seguinte.

Tabela 12 - Descrição das colunas do Antecipar saídas

Coluna	Descrição
1	Id do utilizador
2	*True/False* - O que tiver maior probabilidade
3	A maior probabilidade
4	*True/False* - O que tiver menor probabilidade
5	A menor probabilidade

42733	False	0.860854	True	0.139146
42734	False	0.737465	True	0.262535
42736	False	0.920745	True	0.079255
42737	False	0.961461	True	0.038539
42738	False	0.920745	True	0.079255
42739	False	0.732522	True	0.267478
42740	False	0.834701	True	0.165299
42741	False	0.834381	True	0.165619
42742	True	0.957306	False	0.042694

Figura 56 - Script - Guardar em CSV - Resultado