



CRIAÇÃO DE MODELOS DE ML UTILIZANDO AUTOML



Thayná A. Rodrigues de Almeida – Engenheira I

Autor(es) da apostila

Thayná A. Rodrigues de Almeida – Engenheira I

Instrutor(a) do curso

Luana Dias Pena Forte – Assistente de Suporte Pedagógico

Revisão da apostila

Fit Instituto de Tecnologia

Sorocaba, Janeiro de 2025

Autora



Thayná A. Rodrigues de Almeida

Engenheira

Técnica em Mecatrônica e Engenheira Mecatrônica pela Facens, atualmente cursando pós-graduação em Análise de Dados. FIT desde julho/2021, desenvolvendo firmware, hardware e serviços em nuvem junto ao time de P&D do Sketch to Scale.



APRESENTAÇÃO

A visão computacional, um dos promissores campos da inteligência artificial, possibilita as máquinas a interpretar e processar informações visuais de maneira semelhante à percepção humana. O Google Vertex AI, por sua vez, emerge como uma plataforma avançada que facilita o desenvolvimento e implementação de modelos de visão computacional de forma eficiente. Integrando-se com o Google Cloud, o Vertex AI oferece uma série de ferramentas e recursos que simplificam o treinamento, o gerenciamento e a implantação de modelos de visão computacional em grande escala. Ao fornecer uma infraestrutura escalável e serviços como AutoML, ele torna mais acessível para as organizações incorporarem recursos avançados de visão computacional em suas aplicações, impulsionando inovações em áreas como reconhecimento de objetos, classificação de imagens e análise de vídeo.

A presente apostila é um instrumento teórico que complementa o curso de capacitação em AutoML, promovido pelo Fit - Instituto de Tecnologia, para colaboradores do Fit Instituto de Tecnologia, Flextronics e comunidade. Nela, veremos um breve resumo dos serviços voltados para aprendizado de máquina que estão disponíveis no AutoML, além da criação e categorização de conjuntos de dados e treinamento de modelos. É extremamente recomendável ao aluno que, ao final da leitura de cada seção, realize os exercícios propostos e acesse os materiais indicados nas referências bibliográficas, para aprofundar a leitura desse material e complementar o que foi lido aqui.

Esta apostila está dividida em 10 seções, onde veremos todo o fluxo de criação de um modelo de aprendizado de máquina automatizado com o AutoML, desde a criação e categorização dos conjuntos de dados, o treinamento realizado pela plataforma e como avaliar as métricas de qualidade e realizar inferências de validação do modelo utilizando TF Lite.

Desejo a você, prezado aluno, que tenha um excelente curso!

Boa Leitura!



Sumário

| | | |
|-------|---|----|
| 1 | Preparação de ambiente | 7 |
| 1.1 | Acesso à Google Cloud Platform | 7 |
| 1.2 | Clone do repositório | 7 |
| 1.3 | Instalação de ferramentas | 8 |
| 1.3.1 | Instalação do Visual Studio Code | 8 |
| 1.3.2 | Instalação do Miniforge | 9 |
| 1.4 | Instalação de dependências | 10 |
| 2 | O que é o AutoML? | 13 |
| 2.1 | Treinamentos com dados tabulares | 14 |
| 2.2 | Treinamentos com dados textuais | 15 |
| 2.3 | Treinamentos com dados de vídeo | 17 |
| 2.4 | Treinamentos com dados de imagem | 18 |
| 2.4.1 | Como escolher entre os tipos de modelo disponíveis? | 21 |
| 2.4.2 | Reforçando a teoria da aplicação de modelos de imagem .. | 30 |
| 3 | Introdução ao Cloud Storage | 34 |
| 3.1 | Utilizando o Cloud Storage na prática | 35 |
| 4 | Construção de modelos com o AutoML | 40 |
| 4.1 | Preparação de conjuntos de dados | 40 |
| 4.2 | Criação de datasets no AutoML | 43 |
| 4.3 | Categorização do dataset..... | 51 |
| 4.3.1 | Categorização de modelos de classificação single-label | 52 |
| 4.3.2 | Categorização de modelos de classificação multi-label | 56 |
| 4.3.3 | Categorização de modelos de deteção de objetos | 58 |
| 4.4 | Criação de datasets categorizados com arquivos de carga | 61 |
| 4.5 | Utilizando o template de geração de arquivo de carga | 65 |
| 4.6 | Treinamento de modelos Edge | 70 |



| | | |
|-------|--|-----|
| 4.6.1 | Visualização dos modelos criados | 76 |
| 5 | Análise de métricas de qualidade | 78 |
| 5.1 | Interpretação da saída de modelos | 78 |
| 5.2 | Limite de corte..... | 79 |
| 5.3 | Matriz de confusão | 80 |
| 5.4 | Desempenho dos modelos..... | 82 |
| 5.4.1 | Acurácia | 83 |
| 5.4.2 | Precisão | 83 |
| 5.4.3 | Recall, Revogação ou Sensibilidade..... | 84 |
| 5.4.4 | F1-Score | 85 |
| 5.4.5 | Otimização entre precisão e recall | 86 |
| 5.5 | Reforçando a teoria para análise de métricas de qualidade..... | 89 |
| 5.6 | Análise de métricas de desempenho no AutoML | 91 |
| 6 | Exportação de modelos Edge | 95 |
| 7 | Manipulação do ambiente de implementação..... | 99 |
| 7.1 | Bancada e periféricos..... | 99 |
| 8 | Construção de aplicações próprias em bancada | 104 |
| 8.1 | Treinamento dos modelos | 107 |
| 9 | Verificação dos modelos | 108 |
| 9.1 | Exportação dos modelos | 109 |
| 10 | Familiarização com os templates de software | 111 |
| 10.1 | Interface gráfica com Gradio | 111 |
| 10.2 | Utilização dos templates com modelos treinados | 111 |
| 10.3 | Ensaios e pequenas alterações | 113 |
| 11 | Atividades | 114 |
| | Conclusão..... | 115 |
| | Referências..... | 116 |



CONTROLE DE REVISÃO DO DOCUMENTO / DOCUMENT REVISION CONTROL

..... 116

1 Preparação de ambiente

A preparação do ambiente é uma etapa fundamental para o desenvolvimento de projetos e, nesta seção, serão configuradas as ferramentas necessárias, incluindo a instalação de bibliotecas e extensões, instalação de dependências, organização do ambiente de desenvolvimento e ajustes de software para otimizar o desempenho no decorrer do curso.

1.1 Acesso à Google Cloud Platform

O instrutor irá solicitar um e-mail do aluno para utilização da plataforma no período do curso. O e-mail pode ser pessoal ou corporativo, desde que esteja associado a uma conta Google.

Note que quaisquer extensões de e-mail podem ser associadas a uma conta Google seguindo o [tutorial](#) disponibilizado pela própria empresa. Além disto, contas de e-mail já providas pelo Google (como o Gmail) não precisam realizar este procedimento.

1.2 Clone do repositório

Aqui faremos o clone do repositório contendo todos os scripts e templates de software que serão utilizados no decorrer do curso. Para seguir os passos abaixo, é necessário possuir a ferramenta GIT instalada na máquina.

- Link do repositório: https://github.com/talmeida96/google_cloud_automl

Siga os passos acompanhando a Figura 1:

1. No diretório onde deseja-se clonar o repositório, clique com o botão direito do mouse e procure a opção “Open Git Bash Here”. Isso irá abrir uma janela de execução da ferramenta GIT.
2. No link do Github informado, navegue até o botão “Code” e na seção “HTTPS” copie o url exibido.

3. Na janela do GIT aberta, digite “git clone” + cole o url copiado do Github.

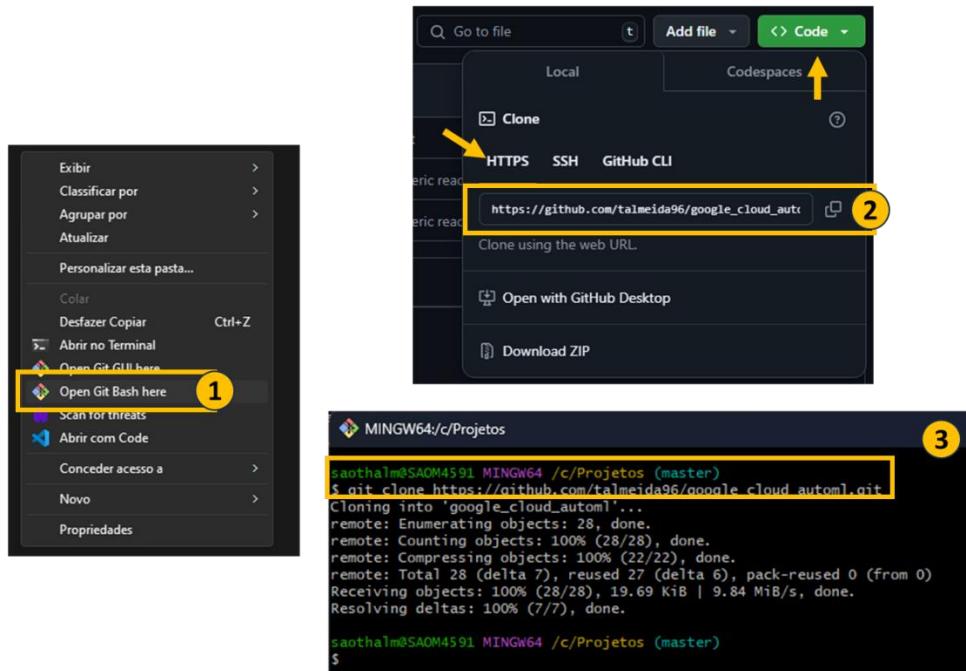


Figura 1 - Clone do repositório localmente. Fonte: autoral.

1.3 Instalação de ferramentas

A instalação das ferramentas necessárias para o pleno acompanhamento das atividades práticas propostas nesta apostila pode ser consultada nos tópicos à seguir.

1.3.1 *Instalação do Visual Studio Code*

Siga os passos descritos abaixo para realizar a instalação inicial do Visual Studio Code e as extensões requeridas.

1. Baixe o instalador no site oficial do Visual Studio através do link <https://code.visualstudio.com/>.
2. Clique no botão “Download for Windows” e aguarde a finalização do download.

3. Navegue até a pasta onde o arquivo foi salvo e execute o instalador através de um “duplo-clique”.
4. Aceite todas as recomendações padrão, selecionando o botão “Next” em todas as telas, e finalmente, selecionando o botão “Install” na última tela.

Após finalizada a instalação, abra o Visual Studio Code e siga os passos abaixo para realizar a instalação das extensões necessárias.

1. Na barra de navegação da lateral esquerda, selecione o símbolo do menu de Extensões (ou utilize o atalho “Ctrl+Shift+x”).
2. Escreva “python” na caixa de busca superior e selecione o primeiro resultado com o provedor verificado Microsoft, conforme Figura 2.
3. Clique em “Install”.

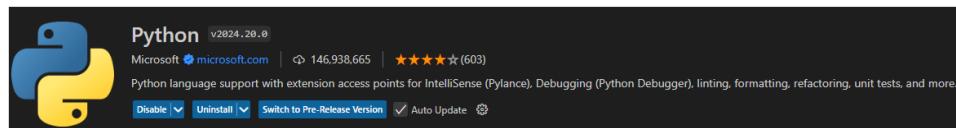


Figura 2 - Extensão Python para VSC. Fonte: autoral.

4. Após instalada a extensão do Python, escreva “Jupyter” na caixa de busca superior e selecione o primeiro resultado com o provedor verificado Microsoft, conforme Figura 3.
5. Clique em “Install”.

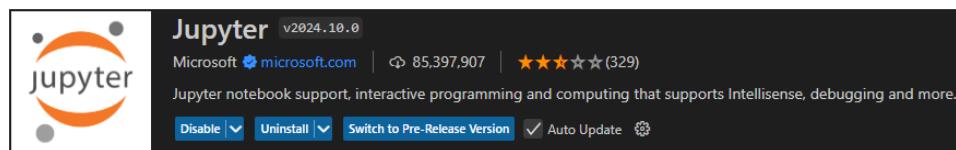


Figura 3 - Extensão Jupyter para VSC. Fonte: autoral.

1.3.2 Instalação do Miniforge

Siga os passos descritos abaixo para realizar a instalação da ferramenta Miniforge.

1. Navegue até o repositório oficial através do link <https://github.com/conda-forge/miniforge/releases>.
2. Baixe o instalador da seguinte versão para Windows: **Miniforge3-24.11.3-0-Windows-x86_64.exe**.
3. Navegue até a pasta onde o arquivo foi salvo e execute o instalador através de um “duplo-clique”.
4. Quando solicitado, ative a caixa de seleção que incluirá o miniforge ao PATH do Windows.
5. Aceite todas as recomendações padrão, selecionando o botão “Next” em todas as telas e, finalmente, selecionando o botão “Install” na última tela.

Após concluir a instalação, ao pesquisar pela ferramenta Miniforge no Windows o aplicativo deve ser retornado conforme Figura 4

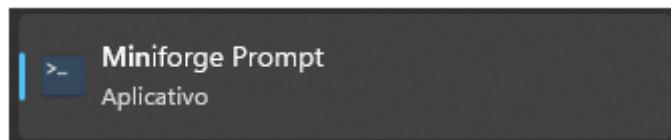


Figura 4 - Miniforge Prompt. Fonte: autoral.

1.4 Instalação de dependências

Com o Miniforge instalado, podemos agora criar o ambiente virtual que será utilizado para nossas validações práticas de inferência de modelos de imagem. Para isso, execute as etapas abaixo:

1. Abra a ferramente Miniforge Prompt
2. Execute o comando para verificação da versão de instalação:
 - a. `conda --version`
3. Copie o comando seguinte e altere-o para navegar até a pasta onde o repositório do curso foi clonado em sua máquina, conforme a seção anterior:
 - a. `cd C:\Projetos\fta_automl`

```
(base) C:\Users\saothalm>cd C:\Projetos\automl_2025
(base) C:\Projetos\automl_2025>
```

Figura 5 - Diretório alterado no Miniforge. Fonte: autoral.

4. Dentro do novo diretório, execute o comando seguinte para criar o ambiente virtual:

a. `conda env create -n vertex_course -f environment.yml`

```
(base) C:\Projetos\automl_2025>conda env create -n vertex_course -f environment.yml
Retrieving notices: done
Channels:
- defaults
- conda-forge
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
      current version: 24.11.3
      latest version: 25.1.1

Please update conda by running

$ conda update -n base -c conda-forge conda

Downloading and Extracting Packages:

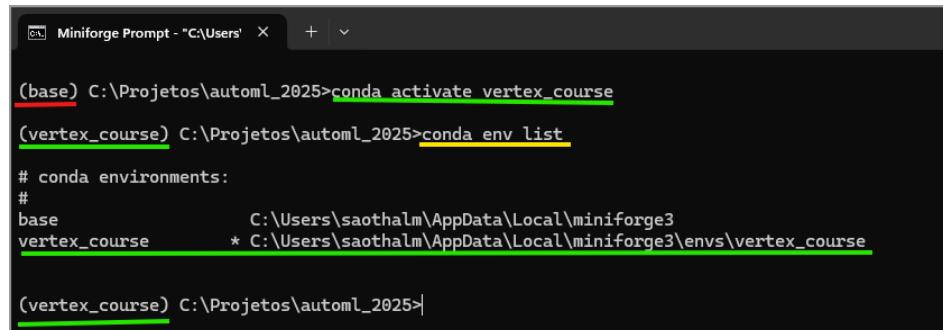
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

Figura 6 - Início da criação de ambiente virtual. Fonte: autoral.

```
Attempting uninstall: typing_extensions
Found existing installation: typing_extensions 4.11.0
Uninstalling typing_extensions-4.11.0:
  Successfully uninstalled typing_extensions-4.11.0
Successfully installed absl-py-2.1.0 aiofiles-23.2.1 aiohttp-3.9.5 aiosignal-1.3.1 annotated-types-0.7.0 anyio-4.3.0 ass4
done
#
# To activate this environment, use
#
# $ conda activate vertex_course
#
# To deactivate an active environment, use
#
# $ conda deactivate
```

Figura 7 - Finalização com sucesso da criação do ambiente virtual. Fonte: autoral.

5. Você pode utilizar o comando abaixo para verificar se o ambiente criado se encontra na lista de ambientes:
 - a. `conda env list`
6. O ambiente ativo no momento encontra-se com o * em frente ao nome. Para ativar um ambiente diferente do “base”, use o comando à seguir:
 - a. `conda activate vertex_course`



```
(base) C:\Projetos\automl_2025>conda activate vertex_course
(vertex_course) C:\Projetos\automl_2025>conda env list

# conda environments:
#
base                  C:\Users\saothalm\AppData\Local\miniforge3
vertex_course          * C:\Users\saothalm\AppData\Local\miniforge3\envs\vertex_course

(vertex_course) C:\Projetos\automl_2025>
```

Figura 8 - Ambiente virtual ativo. Fonte: autoral.

7. Quando necessário desativar ou remover um ambiente virtual criado, utilize os comandos abaixo, respectivamente:
 - a. `conda deactivate vertex_course`
 - b. `conda env remove -n vertex_course`

2 O que é o AutoML?

A Google Cloud, por meio da plataforma Vertex AI, oferece uma ampla variedade de serviços, produtos e ferramentas construídos para auxiliar na exploração do potencial da IA e do ML. Essas soluções permitem aprimorar produtos, otimizar processos e impulsionar tanto a eficiência quanto a inovação.

Os serviços disponíveis podem ser classificados nas seguintes categorias: BigQuery ML, APIs Pré-construídas, AutoML e Treinamento Personalizado. Cada uma dessas categorias oferece níveis distintos de personalização e requer diferentes graus de familiaridade com as tecnologias envolvidas. A Tabela 1 apresenta essa classificação de forma detalhada.

| Categoria | Treinamento Personalizado | BigQuery ML | API Pré-construída | AutoML |
|------------------------------|--------------------------------|-------------|--------------------------------|--|
| Tipo | Tabular, imagem, texto e vídeo | Tabular | Tabular, imagem, texto e vídeo | Tabular, imagem , texto e vídeo |
| Quantidade | Média/Alta | Média/Alta | - | Baixa/Média |
| Experiência (ML/Code) | Alta | Média | Baixa | Baixa/Média |
| Flexibilidade | Alta | Média | Nenhuma | Média |
| Tempo | Alto | Médio | Nenhum | Médio |

Tabela 1 – Categorias de serviços de ML no GCP. Fonte: autoral.

O AutoML – Automate Machine Learning – é uma ferramenta que automatiza o treinamento de modelos de ML e permite que os usuários com pouca ou nenhuma experiência em programação criem modelos personalizados para objetivos específicos em tarefas de classificação de imagens, detecção de objetos, processamento de linguagem natural, entre outras possibilidades. Sua interface amigável com fluxos guiados elimina a necessidade de ajustes manuais complexos e, ao mesmo tempo, entrega modelos otimizados e escaláveis.

Neste curso nos restringiremos ao uso da ferramenta AutoML utilizando dados de imagens estáticas, mas veremos de forma superficial seus demais métodos de treinamento.

2.1 Treinamentos com dados tabulares

Dados tabulares são dados estruturados e provenientes de tabelas, que podem estar armazenadas em planilhas, bancos de dados etc. No AutoML existem quatro opções de treinamento de modelos com este tipo de dados.

- **Classificação Binária:** este tipo de modelo prevê um resultado binário, ou seja, que tenha uma entre duas classes. Esse tipo de modelo é adequado para perguntas com resposta “sim” ou “não”. Por exemplo: (a) para prever se um cliente estaria disposto (ou não) a adquirir uma assinatura; (b) para prever se um paciente será diagnosticado com uma condição médica específica (ou não) com base em exames laboratoriais e histórico clínico. Em circunstâncias iguais, um problema de classificação binária requer menos dados do que outros tipos de modelo.
- **Classificação multiclasse:** este tipo de modelo prevê um resultado que tenha uma entre três ou mais classes distintas. Esse tipo de modelo é adequado para categorizações. Por exemplo: (a) um varejista pode criar um modelo de classificação multiclasse para segmentar os clientes em perfis diferentes; (b) identificar o tipo de fruta (maçã, laranja ou banana) com base em características como peso, cor e tamanho coletados em um conjunto de dados agrícola.
- **Previsão:** este tipo de modelo prevê uma sequência de valores. Por exemplo: (a) como varejista, você pode prever a demanda diária dos seus produtos nos próximos três meses para poder estocá-los adequadamente com antecedência; (b) antecipar as vendas diárias de ingressos para eventos esportivos em diferentes localidades, considerando sazonalidade e datas de feriados.
- **Régressão:** este tipo de modelo prevê um resultado com valor contínuo. Por exemplo: (a) para o varejista, é conveniente criar um modelo de regressão para prever quanto um cliente gastará no próximo mês; (b) estimar o tempo médio de entrega de um pedido baseado em fatores como distância, tráfego e horário de envio.

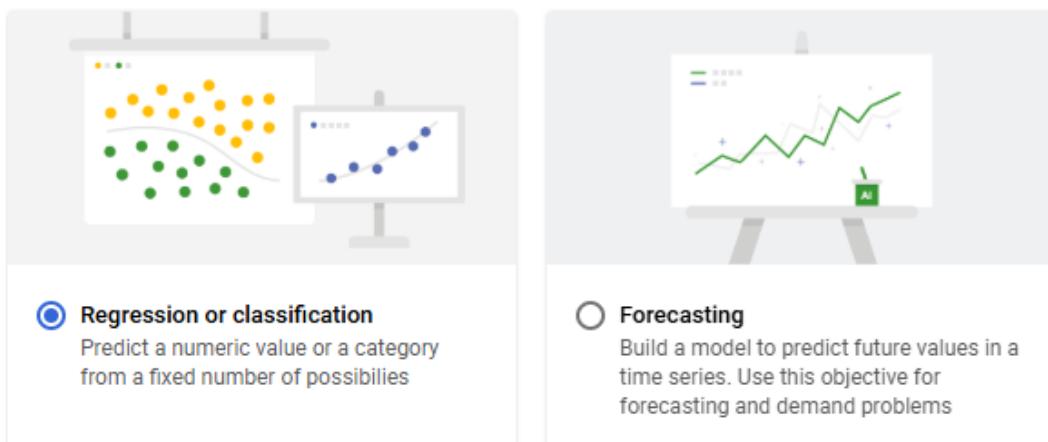


Figura 9 - Métodos de treinamento de modelos tabulares. Fonte: Google Cloud

2.2 Treinamentos com dados textuais

O AutoML analisa a estrutura e o significado de dados de texto. É possível treinar modelos para classificar e extrair informações de textos e até mesmo compreender o sentimento dos autores no momento da escrita.

- Classificação de textos:** um modelo de classificação analisa dados de texto e retorna uma lista de categorias que se aplicam ao texto encontrado nos dados. A Vertex AI oferece modelos de classificação de texto com um ou vários rótulos (binário ou multiclasse). Exemplo de uso: categorizar automaticamente artigos de notícias em temas como "Esportes", "Política", "Tecnologia" ou "Entretenimento", facilitando a organização em portais de mídia.

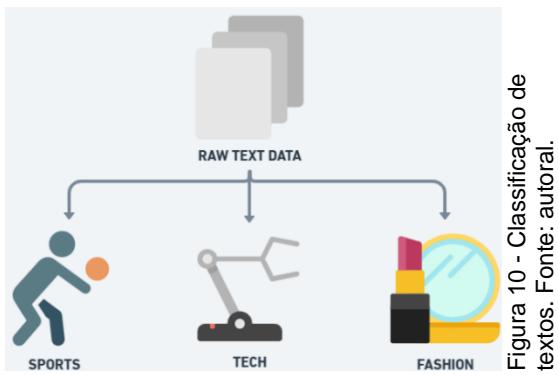


Figura 10 - Classificação de textos. Fonte: autoral.

- Extração de entidade:** um modelo de extração de entidade inspeciona os dados de texto de entidades conhecidas citadas nos dados e rotula essas entidades no texto. Exemplo de uso: identificar nomes de medicamentos, dosagens e instruções de uso em

prontuários médicos digitais para apoiar sistemas de saúde automatizados.

- **Análise de sentimento:** um modelo de análise de sentimento inspeciona dados de texto e identifica o estado emocional predominante neles, principalmente para determinar a atitude de um escritor como positiva, negativa ou neutra. Exemplo de uso: analisar avaliações de clientes em um site de e-commerce para determinar se os comentários são positivos, negativos ou neutros, auxiliando no aprimoramento de produtos e serviços.

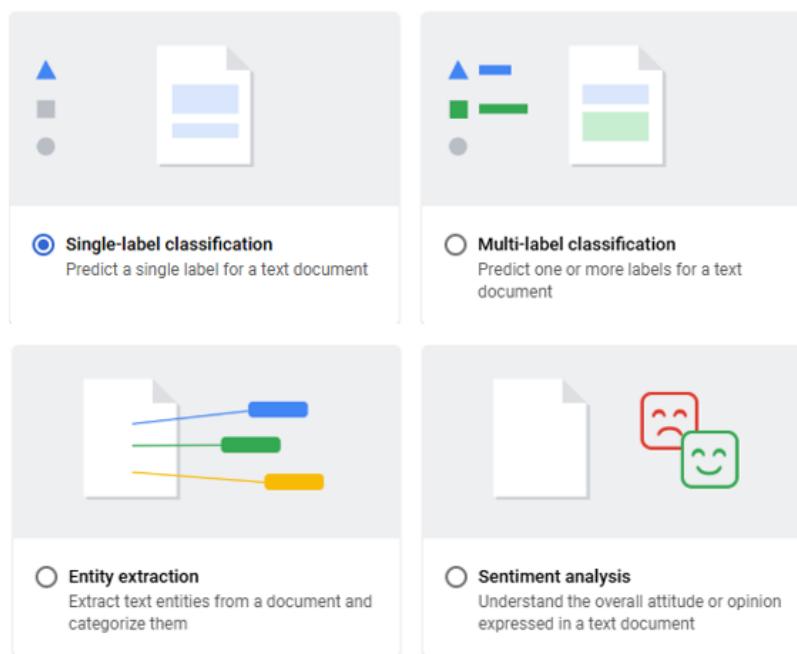


Figura 11 - Métodos de treinamento de modelos textuais. Fonte: Google Cloud.

Atualmente, a [documentação](#) oficial da Google Cloud sobre o desenvolvimento e métodos de uso do AutoML para dados textuais possui um comunicado sobre as transferências de suas funções da atual ferramenta AutoML para a ferramenta Vertex AI Gemini. As funções fornecidas estarão disponíveis no serviço AutoML até 15 de junho de 2025, conforme Figura 12.

! A partir de 15 de setembro de 2024, só será possível personalizar a classificação, a extração de entidades, e os objetivos de análise de sentimento por meio da migração para os comandos e os ajustes do Vertex AI Gemini. O treinamento ou atualização de modelos do AutoML na Vertex AI para classificação de texto, extração de entidades e objetivos de análise de sentimento não estarão mais disponíveis. É possível continuar usando os modelos de AutoML Text na Vertex AI até 15 de junho de 2025. Para uma comparação entre o AutoML Text e o Gemini, consulte [Gemini para usuários de texto do AutoML](#). Para mais informações sobre como o Gemini oferece uma experiência do usuário aprimorada com recursos aprimorados de comandos, consulte [Introdução aos ajustes](#). Para começar a ajustar, consulte [Ajuste de modelos de texto do Gemini](#)

Figura 12 - Nota de transferência de função entre ferramentas. Fonte: Google Cloud

2.3 Treinamentos com dados de vídeo

Analisa dados de vídeos para classificar tomadas e segmentos ou detectar e rastrear objetos que aparecem nos mesmos.

- **Reconhecimento de ações:** reconheça ações categorizadas em tomadas de vídeo. Exemplos de uso: (a) encontre ações como fazer um gol, causar uma falta, cobrar um pênalti. Útil para treinadores que estudam os pontos fortes e fracos da equipe; (b) identificar automaticamente ações como "acidente de trânsito", "dirigindo em alta velocidade" ou "parada no semáforo" em imagens de câmeras de monitoramento urbano, auxiliando na análise de segurança viária.

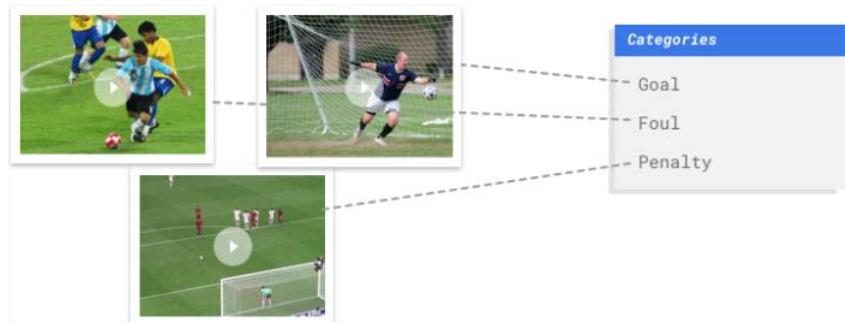


Figura 13 - AutoML: Reconhecimento de ações. Fonte: Google Cloud

- **Classificação:** classifique cada cena do vídeo analisado. Exemplo de uso: (a) categorizar “primeiro/segundo tempo” de um jogo, visualização do “jogo”, visualização do “público”, visualização do “treinador”; (b) categorizar cenas de vídeo em “propaganda”, “trecho do programa”, ou “intervalo comercial” em transmissões televisivas, facilitando a análise de tempo de exibição para anunciantes.

- **Rastreamento de objetos:** prevê caixas delimitadoras e rastreia o início/fim de objetos dentro de vídeos. Exemplos de uso: (a) rastreie a bola ou os jogadores em uma partida de futebol para receber estatísticas dos jogadores, como mapa de calor no campo e passadas bem-sucedidas; (b) rastrear o movimento de carrinhos de compras em vídeos de supermercados para analisar padrões de compra e otimizar a disposição dos produtos nas prateleiras.

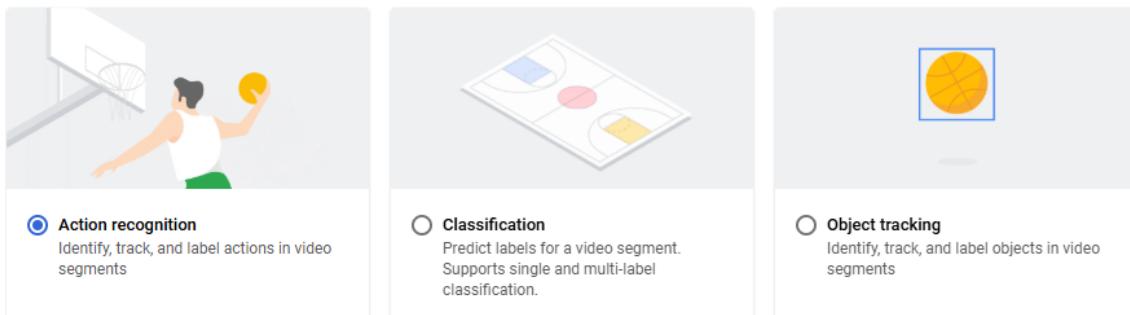


Figura 14 - Métodos de treinamento de modelos de vídeo. Fonte: Google Cloud

2.4 Treinamentos com dados de imagem

São treinamentos que utilizam imagens como entradas para o modelo. Os modelos podem ser do tipo Classificação, Detecção de Objetos ou Segmentação.



Image Classification

Object Detection

Image Segmentation

Figura 15 - Diferenças entre os tipos de modelo de imagem. Fonte: LinkedIn.

- **Classificação:** um modelo de classificação de imagens analisa dados de uma imagem e retorna uma lista de categorias de conteúdo que se

aplicam a ela. Por exemplo, é possível treinar um modelo para classificar imagens de acordo com a presença ou não de um cachorro, ou treinar um modelo para classificar imagens de cães de acordo com a raça. O AutoML oferece modelos de classificação de imagem com um ou vários rótulos de saída (binário ou multiclasse).



Figura 16 - AutoML: Classificação de imagens. Fonte: Google Cloud

- Detecção de objetos:** um modelo de detecção de objetos analisa os dados da imagem e retorna anotações para todos os objetos encontrados em uma imagem, que consiste em um rótulo e um local da caixa delimitadora para cada objeto. Por exemplo, é possível treinar um modelo para encontrar a localização de animais em dados de imagem.

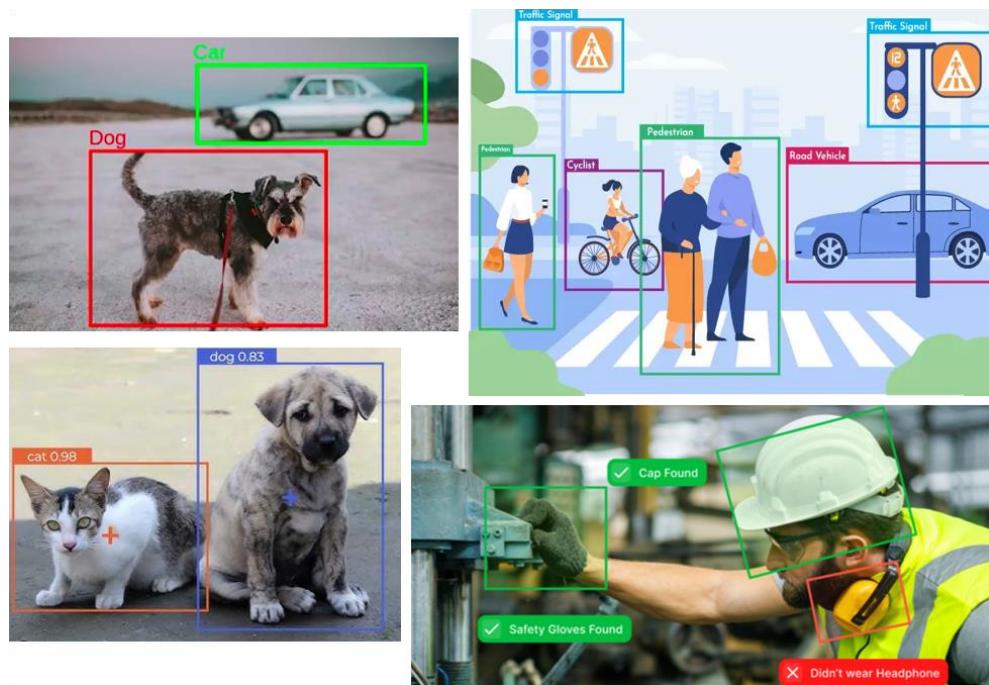


Figura 17 - Exemplos de detecção de objetos. Fonte: autoral.

- **Segmentação:** um modelo de segmentação de imagem analisa os dados de imagem e atribui um rótulo a cada pixel da imagem, permitindo a identificação precisa das diferentes regiões ou objetos. Por exemplo, é possível treinar um modelo para segmentar áreas de tumores em imagens de ressonância magnética ou segmentar diferentes tipos de terreno em imagens de satélite. Existem dois tipos de segmentação: (i) segmentação de instâncias, onde objetos da mesma classe são tratados como diferentes portanto, todos os objetos são coloridos com cores diferentes, mesmo que pertençam à mesma classe; e (ii) segmentação semântica, onde todos os objetos da mesma classe formam uma única classificação, portanto, todos os objetos da mesma classe são coloridos pela mesma cor.

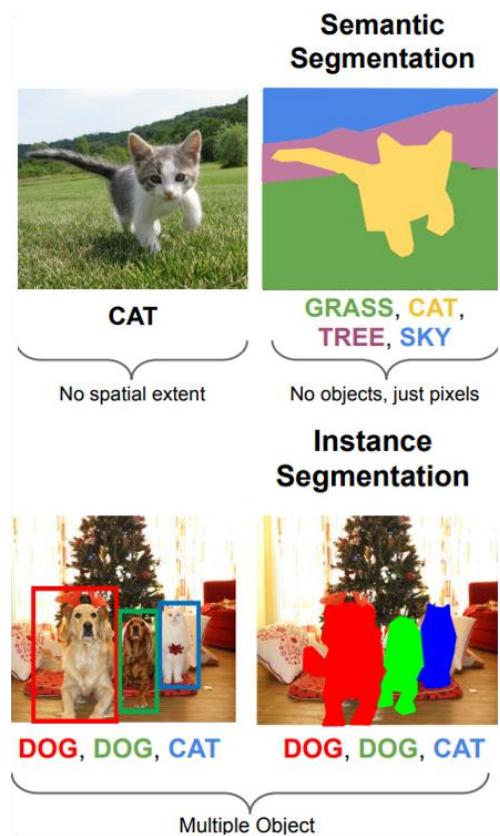


Figura 18 - Segmentação de instâncias e semântica. Fonte: cs231n.stanford.edu



Figura 19 - Diferença entre segmentação semântica e de instâncias. Fonte: livebook.manning.com

2.4.1 Como escolher entre os tipos de modelo disponíveis?

Apesar de todos os tipos de modelo apresentados no início do tópico 2.4 serem aplicáveis para entradas com dados de imagem, é necessário avaliar qual modelo é mais viável para seu caso de uso específico.

Em resumo e exemplificado na Figura 20, um modelo de classificação binária requer que a amostra possua somente um rótulo (classe), enquanto um modelo de classificação multirrótulos pode detectar mais de um rótulo (classes) para cada amostra.

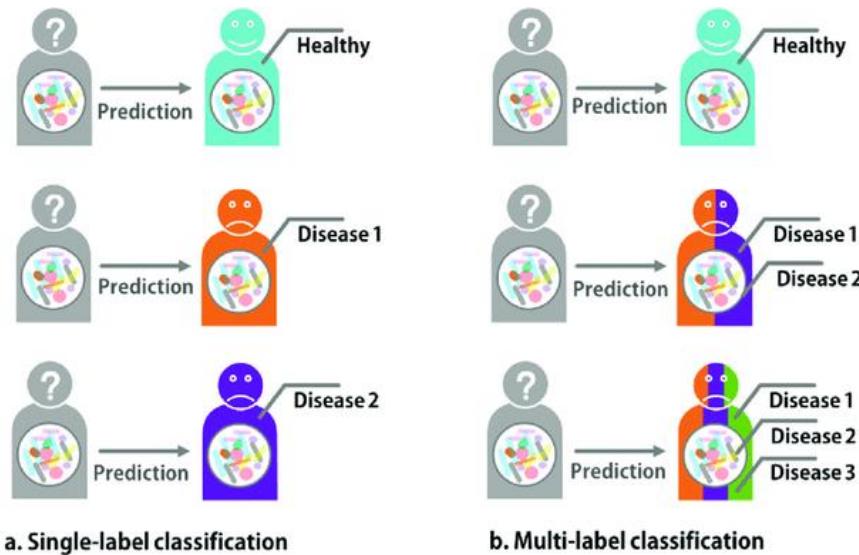


Figura 20 - Comparaçāo entre modelos single-label e multi-label. Fonte: Research Gate.

Existem três questões técnicas relevantes no uso do modelo de classificação multirrótulo que devem ser consideradas:

- Quando maior a quantidade de rótulos nos dados de treinamento, maior o custo computacional.
- Rótulos ausentes reduzem a sensibilidade das detecções.
- Rótulos ambíguos introduzem resultados “falso-positivos”.

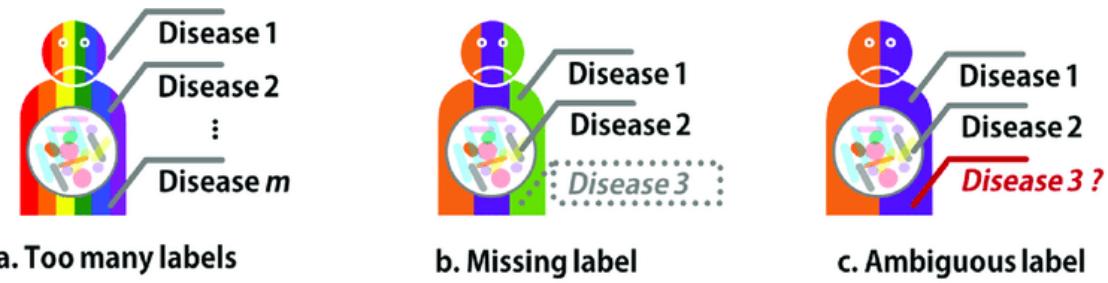


Figura 21 - Questões técnicas no uso de modelos multirrótulos. Fonte: Research Gate.

Veremos abaixo os casos de uso indicados para cada um dos modelos existentes, com seus benefícios e limitações, a fim de escolher o melhor tipo de treinamento para futuras implementações.

1. CLASSIFICAÇÃO DE RÓTULO ÚNICO (SINGLE-LABEL)

Os modelos de classificação de rótulo único (single-label) são indicados para situações em que cada imagem conterá apenas um objeto ou conceito principal que dominará a composição da imagem. Este tipo de modelo funciona melhor quando as seguintes condições estão presentes:

- **Objetivo único predominante:** a imagem possui um único elemento de interesse e este ocupa a maior parte do espaço na mesma.
- **Fundo uniforme ou irrelevante:** o plano de fundo da imagem é simples, consistente ou insignificante para a classificação.
- **Exclusividade de classes:** cada imagem pertence exclusivamente a uma única categoria, sem possibilidade de sobreposição entre classes.

Alguns exemplos de aplicação para modelos de classificação de imagem com rótulo único são:

- **Identificação de objetos individuais:** classificar imagens de frutas (como maçãs, laranjas ou abacaxis); classificar imagens de animais (como cães, gatos ou pássaros). Nestes casos, cada imagem contém apenas um objeto isolado em fundo uniforme (como um fundo branco ou de cor sólida).
- **Classificação de ambientes:** determinar o tipo de cômodo representado em uma foto (como sala de estar, cozinha ou quarto); determinar o tipo de construção representada em uma foto (como casas, prédios ou cabanas). Nestes casos, o foco é o ambiente como um todo, sem considerar a presença de múltiplos elementos distintos.



CAT

Figura 22 – Imagem de gato.
Fonte: google imagem.

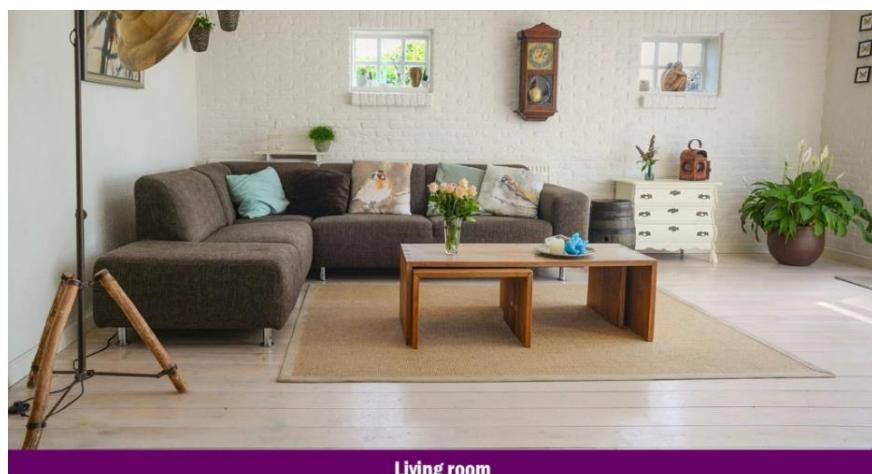


Figura 23 - Exemplo de classificação de ambiente. Fonte: sentisight.ai

A Figura 24 exemplifica como seria a visualização gráfica de um modelo:



Figura 24 - Exemplo gráfico de "entrada" e "saída" de modelos de classificação única.
Fonte: autoral.

Como todos os métodos de treinamento, existem alguns benefícios e limitações ao se optar pelo modelo de classificação de imagem com rótulo único, são eles:

- **Benefícios:**

- Simplicidade no treinamento e na interpretação dos resultados.
- Requer menos processamento para cada imagem.
- Resultados assertivos com maior rapidez.

- **Limitações:**

- Não é ideal para imagens com múltiplos objetos ou conceitos coexistindo, pois se restringe a identificar apenas uma classe por vez.

Desta forma, concluímos que modelos de rótulo único são ótimas ferramentas para cenários onde as classes são mutuamente exclusivas e o foco principal é um único elemento por imagem. Contudo, para cenários mais

complexos, como imagens com múltiplos elementos, é recomendável optar por técnicas de classificação multiclass ou detecção de objetos.

2. CLASSIFICAÇÃO DE MULTIRRÓTULOS (MULTI-LABEL)

Os modelos de classificação multirrótulos são projetados para reconhecer múltiplas classes em uma única imagem. Esses modelos analisam a imagem como um todo e atribuem rótulos independentes para diferentes conceitos ou objetos presentes.

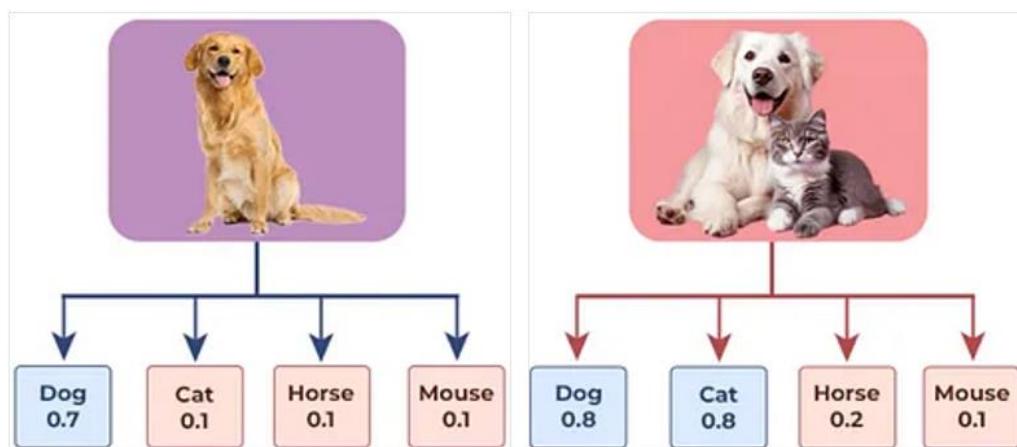


Figura 25 - Exemplo de classificações de modelo multirrótulo. Fonte: Medium.

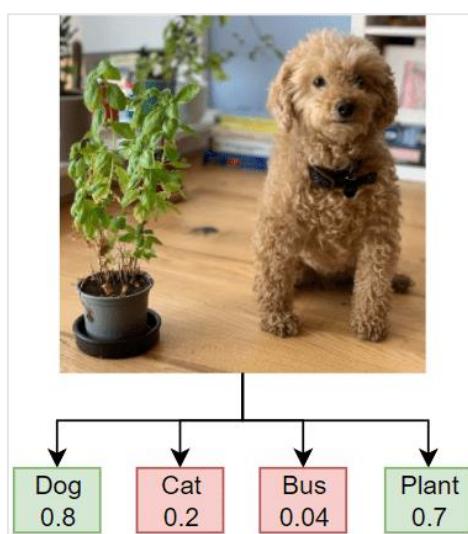


Figura 26 - Exemplo de classificação multirrótulo. Fonte: MathWorks.

Abaixo estão algumas vantagens deste tipo de modelo:

- **Eficiência no processamento:** permite a detecção de diversas classes simultaneamente.
- **Reconhecimento de conceitos abstratos:** ideais para identificar características ou cenários que não representam objetos físicos específicos, mas sim conceitos mais subjetivos, como:
 - Estações do ano: primavera, verão, outono e inverno.
 - Localização ou contexto: interior ou exterior.
 - Condições climáticas: nublado, ensolarado ou chuvoso.

Embora os modelos de Detecção de Objetos sejam adequados para localizar e identificar instâncias específicas de objetos em uma imagem (como "uma pessoa" ou "um carro"), eles não são eficazes para classificar conceitos abstratos que não possuem uma forma física definida. Nesse contexto, os modelos multirrótulo se destacam:

- **Detecção de conceitos não espaciais:** Eles podem atribuir rótulos como "primavera" e "ensolarado" à mesma imagem, sem a necessidade de localizar objetos específicos.
- **Flexibilidade no uso:** Funcionam bem tanto em cenários abstratos quanto em contextos mais específicos, onde múltiplos objetos coexistem na imagem (como mais de uma espécie de animal).

A Figura 27 exemplifica como seria a visualização gráfica de um modelo:

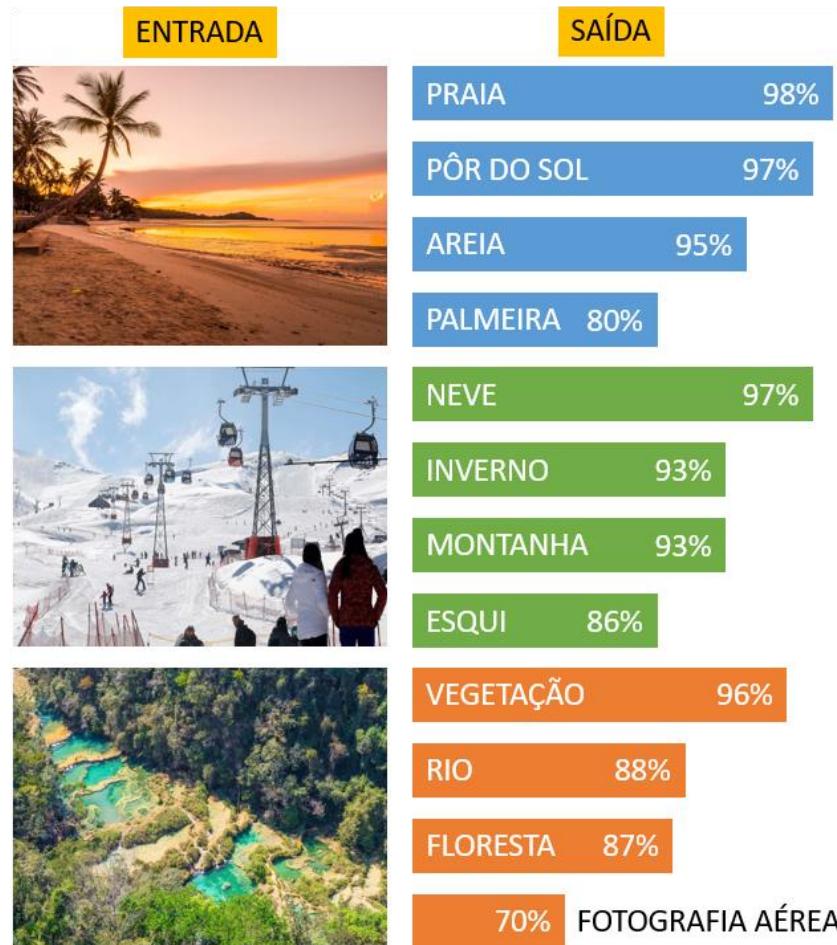


Figura 27 - Exemplo gráfico de "entrada" e "saída" de modelos de classificação multirrótulos.
Fonte: autoral.

Desta forma, concluímos que os modelos de classificação multirrótulos são excelentes treinamentos para cenários onde múltiplos conceitos ou características precisam ser reconhecidos em uma amostra única. Sua flexibilidade o torna ideal para classificações amplas e para contextos abstratos.

3. DETECÇÃO DE OBJETOS

Os modelos de Detecção de Objetos têm uma diferença significativa entre os modelos de classificação de rótulo único ou multirrótulos, pois eles não apenas identificam a presença das classes específicas em uma imagem, mas também localizam e demarcam os objetos de interesse dentro da cena. Essa capacidade torna o modelo essencial para aplicações que demandam precisão espacial e contagem de instâncias de objetos.

As principais características deste tipo de modelo são:

- **Localização dos objetos:** além de identificar as classes existentes na imagem, o modelo desenha caixas delimitadoras (as “bounding boxes”) em torno dos objetos de interesse, ou seja, ele reconhece as classes e determina suas posições na imagem.
- **Contexto e detalhes:** a detecção de objetos é ideal para imagens complexas com muitos elementos de fundo ou onde os objetos de interesse sejam pequenos e não dominem a composição da imagem.
- **Capacidade de contagem:** um recurso que se destaca dos modelos de classificação é a habilidade de contar instâncias de objetos, como números de carros em uma rua ou quantidade de pessoas em um ambiente.

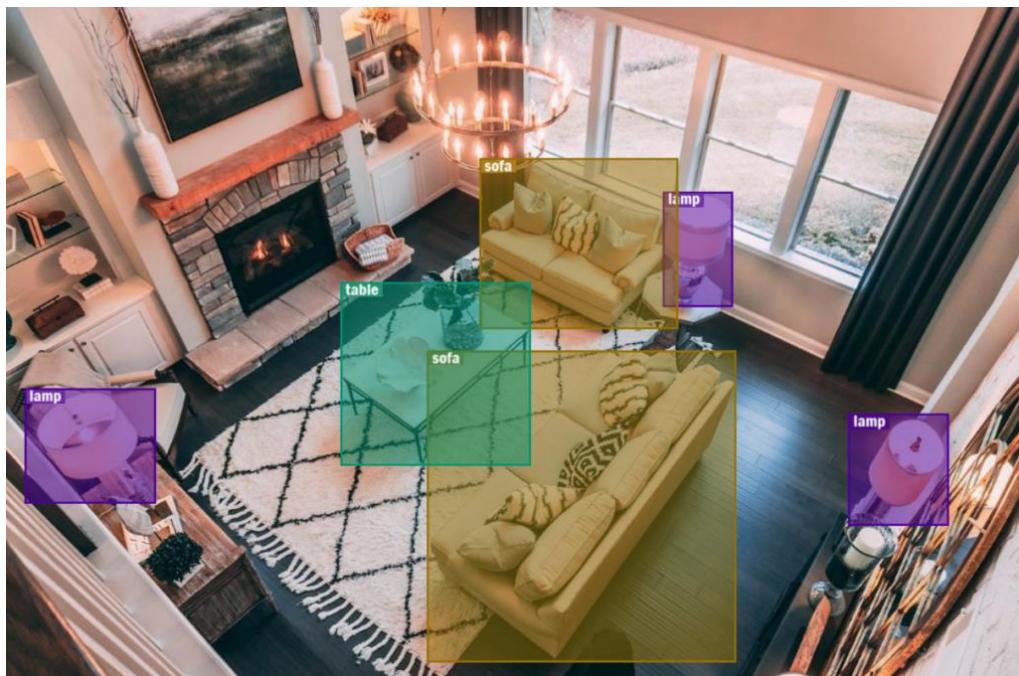


Figura 28 - Exemplo de detecção de objetos. Fonte: sentisight.ai

Existem alguns desafios e considerações a se fazer para a escolha de um treinamento deste tipo:

- **Rotulagem das imagens:** o processo de categorização é mais trabalhoso do que para a classificação, pois exige que cada objeto de interesse seja demarcado manualmente com uma caixa delimitadora. Existem algumas ferramentas para auxiliar nesta tarefa (como o

LabelImg e o Roboflow), mas neste curso, utilizaremos um script produzido para gerar também o arquivo de carga na plataforma AutoML.

- **Tempo de treinamento:** este tipo de modelo requer um tempo de treinamento significativamente maior do que modelos de classificação, devido sua complexidade de aprender tanto a identificação quanto a localização dos objetos.
- **Velocidade de inferência:** o tempo de inferência pode ser comparado ao de modelos de classificação multirrótulos, porém, a detecção de objetos pode se tornar mais lenta a depender do número de classes e da quantidade de objetos presente na imagem.

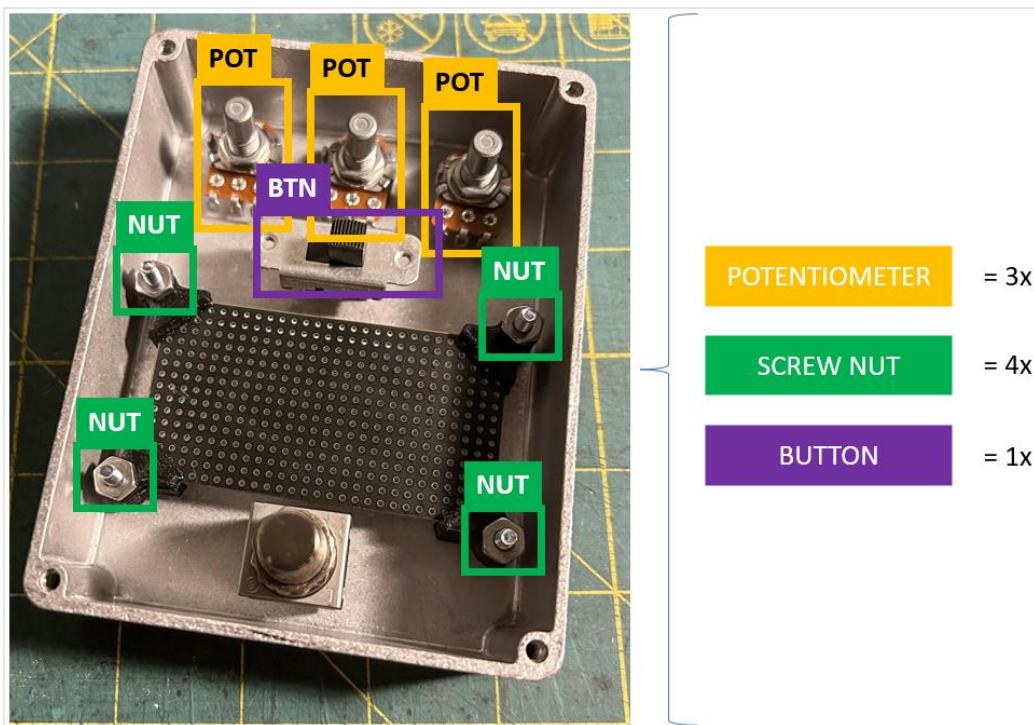


Figura 29 - Exemplo gráfico de "entrada" e "saída" de modelos de detecção de objetos. Fonte: autoral.

Os modelos de detecção de objetos são ferramentas cruciais, que vão além da simples classificação, fornecendo detalhes espaciais e quantitativos sobre os elementos presentes na imagem. Apesar de trazer maior complexidade e custos computacionais, é um modelo indispensável em aplicações que exigem

precisão e localização dos objetos, como segurança, diagnósticos e automação industrial.

2.4.2 Reforçando a teoria da aplicação de modelos de imagem

A seguir, temos algumas situações para reforçar o pleno entendimento sobre as diferenças entre os tipos de modelo de classificação – rótulo único ou multirrótulos – e de detecção de objetos.

1. Para classificar imagens de frutas em categorias exclusivas (como “maçã”, “banana” ou “laranja”), qual tipo de modelo você usaria?
 Single-label
 Multi-label
 Object detection
 Instance segmentation

2. Para identificar a quantidade de tipos diferentes de flores em uma imagem de jardim, qual tipo de modelo você usaria?
 Single-label
 Multi-label
 Object detection
 Instance segmentation

3. Para um sistema que analisa fotos de residências e determina se a fachada tem "janelas", "portas", "varandas", etc., mas sem precisar saber onde cada item está localizado, qual modelo seria suficiente?
 Single-label
 Multi-label
 Object detection
 Instance segmentation

4. Para identificar em uma foto alguns objetos de interesse, qual seria o tipo de modelo mais adequado?

- Single-label
 - Multi-label
 - Object detection
 - Instance segmentation
5. Um site imobiliário precisa analisar imagens de cômodos e classificá-las de acordo com seu uso (“quarto”, “cozinha”, “banheiro”, etc.). Qual modelo seria mais adequado?
- Single-label
 - Multi-label
 - Object detection
 - Instance segmentation
6. Na última estação de uma linha de produção é necessário verificar se todas as etiquetas necessárias foram coladas na embalagem. Qual modelo você usaria?
- Single-label
 - Multi-label
 - Object detection
 - Instance segmentation
7. A quais tipos de modelo pertencem as imagens abaixo?

a)



A

- Biking
- Running
- Swimming

- single

- multi
- instance segmentation
- obj detection

b)



B

- | | |
|--|--|
| <input checked="" type="checkbox"/> Bike | <input checked="" type="checkbox"/> Tree |
| <input checked="" type="checkbox"/> Person | <input type="checkbox"/> Car |
| <input type="checkbox"/> Boat | <input type="checkbox"/> House |

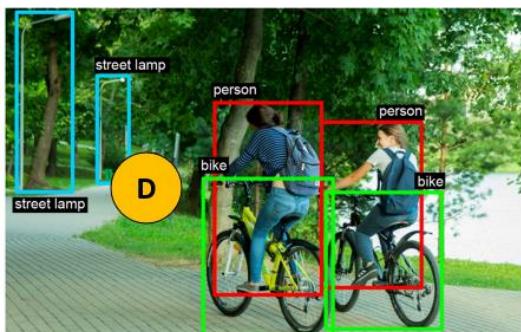
- single
- multi
- instance segmentation
- obj detection

c)



- single
- multi
- instance segmentation
- obj detection

d)



- single
- multi
- instance segmentation
- obj detection

3 Introdução ao Cloud Storage

O Google Cloud Storage é um serviço gerenciado oferecido para armazenamento de dados não estruturados, ou seja, dados que não possuem um formato fixo ou organização pré-definida, como arquivos de áudio, vídeo, imagens e documentos. É possível armazenar qualquer quantidade de dados e recuperá-los quantas vezes forem necessárias, com custo sob demanda.

Diferença entre Tipos de Dados

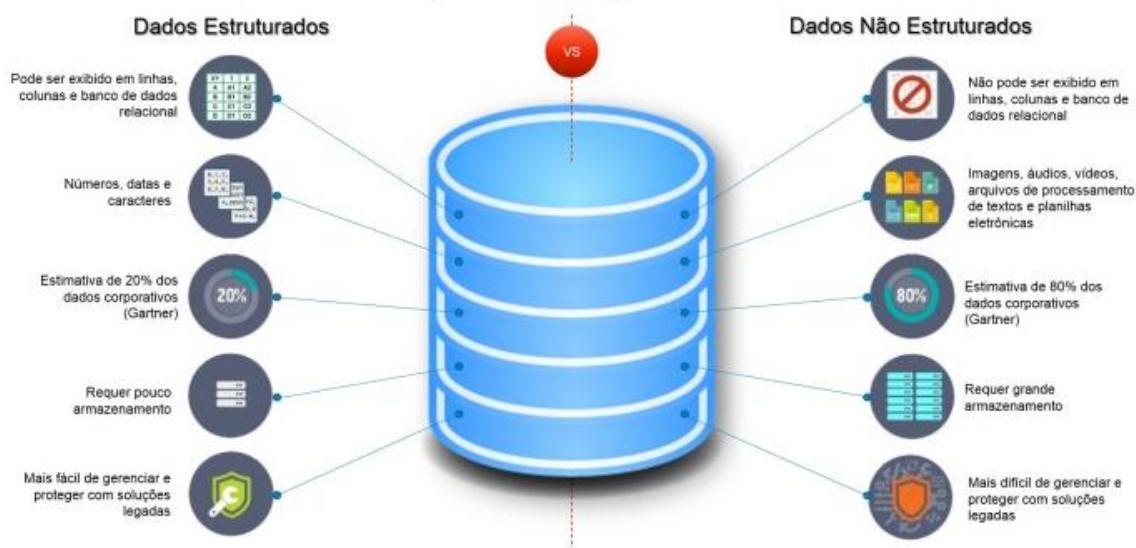


Figura 30 - Diferença entre dados estruturados e não-estruturados. Fonte: LinkedIn.

Existem diferentes categorias de armazenamento no Cloud Storage, que podem ser vistas na Tabela 2. Essas diferentes opções servem para atender as necessidades de uso e de otimização de custos, a depender da frequência de acesso aos dados e da finalidade do armazenamento.

| TIPO | DESCRÍÇÃO | FINALIDADE |
|----------|--|--|
| STANDARD | Dados que são frequentemente acessados e/ou armazenados por breves períodos . | Dados acessados com frequência , como sites, streaming de vídeos e apps mobile. |
| NEARLINE | Altamente durável e de baixo custo para dados acessados com pouca frequência . | Dados que podem ser armazenados por 30 dias . |
| COLDLINE | Altamente durável e de baixo custo para dados pouco acessados. | Dados que podem ser armazenados por 90 dias . |
| ARCHIVE | O menor custo e alta durabilidade para arquivamento de dados, backup on-line e recuperação de desastres. | Dados que podem ser armazenados por 365 dias . |

Tabela 2 - Categorias de armazenamento do Cloud Storage. Fonte: Adaptação Google Cloud.

A Tabela 3 nos ajuda a entender casos reais de uso adequado das categorias de armazenamento disponíveis no Cloud Storage.

| TIPO | CASO DE USO |
|-----------------|---|
| STANDARD | Aplicações em tempo real, como sites dinâmicos e bancos de dados. |
| NEARLINE | Backups mensais e arquivos menos utilizados que ainda precisam estar disponíveis rapidamente. |
| COLDLINE | Arquivamento de longo prazo, como registros históricos ou dados para auditorias. |
| ARCHIVE | Dados de conformidade, arquivos de décadas ou informações que precisam ser armazenadas apenas para consulta legal ou regulatória. |

Tabela 3 - Casos de uso comuns. Fonte: autoral.

3.1 Utilizando o Cloud Storage na prática

O Google Cloud Storage pode ser acessado através da barra de menu principal da plataforma, navegando até o menu “All Products” e selecionando a categoria “Storage” no menu lateral, conforme a Figura 31.

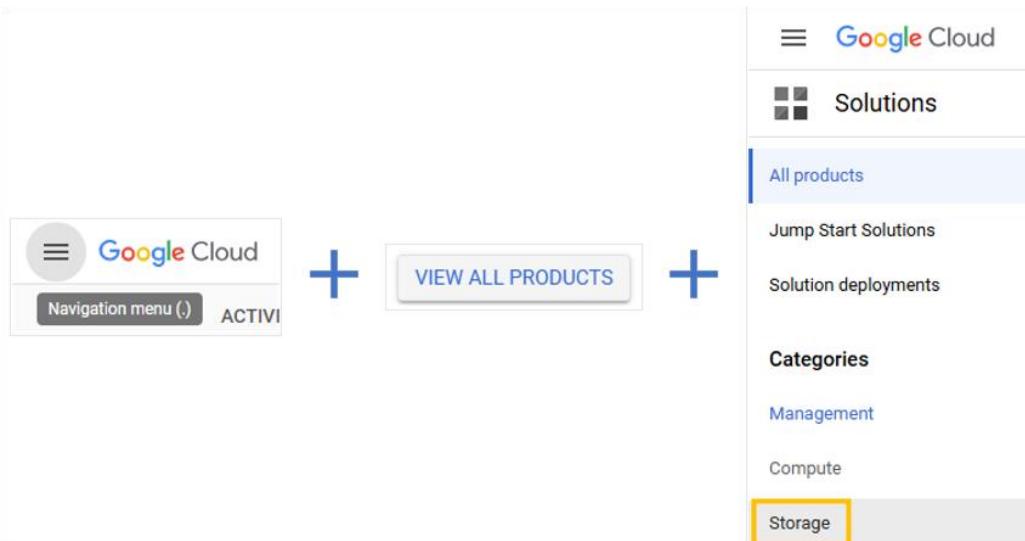
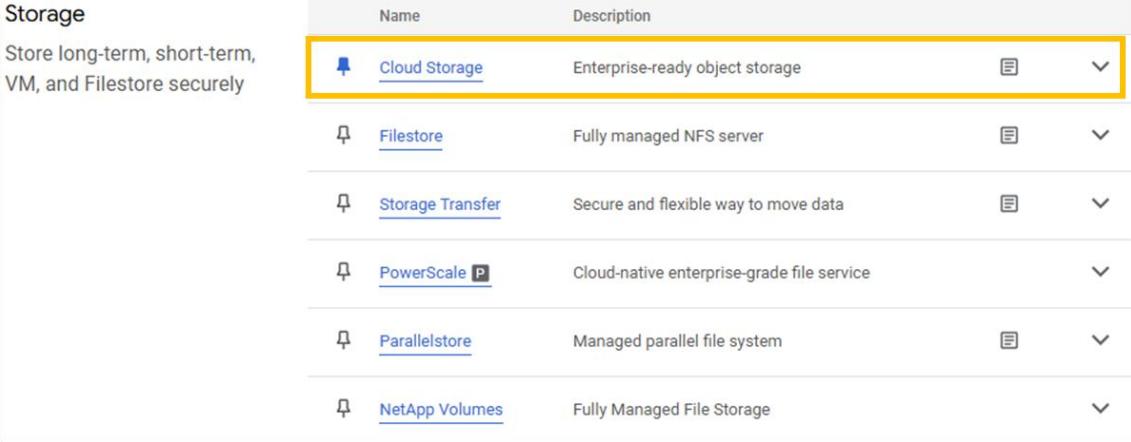


Figura 31 - Navegação até o menu Storage. Fonte: autoral.

Após isso, é possível ver todas as opções de ferramentas de armazenamento disponíveis pela Google Cloud na área de exibição à direita, como mostrado na imagem da Figura 32, inclusive a Cloud Storage, que utilizaremos no decorrer deste curso.



| Storage | Name | Description | Actions |
|---|----------------------------------|--|---------|
| Store long-term, short-term, VM, and Filestore securely | Cloud Storage | Enterprise-ready object storage | |
| | Filestore | Fully managed NFS server | |
| | Storage Transfer | Secure and flexible way to move data | |
| | PowerScale | Cloud-native enterprise-grade file service | |
| | Parallelstore | Managed parallel file system | |
| | NetApp Volumes | Fully Managed File Storage | |

Figura 32 - Acesso a ferramenta Cloud Storage. Fonte: autoral.

Ao clicarmos na ferramenta Cloud Storage, somos redirecionados para a página de uso da mesma, onde podemos ver as opções de acesso ao “buckets”, “monitoring” e “settings”, além é claro, da página de “overview”.

A página “overview” nos apresenta um atalho para as pastas fixadas, os últimos arquivos acessados e as atividades recentes de transferências de arquivos, mas para uso no decorrer deste curso, faremos sempre uso da opção “buckets”.

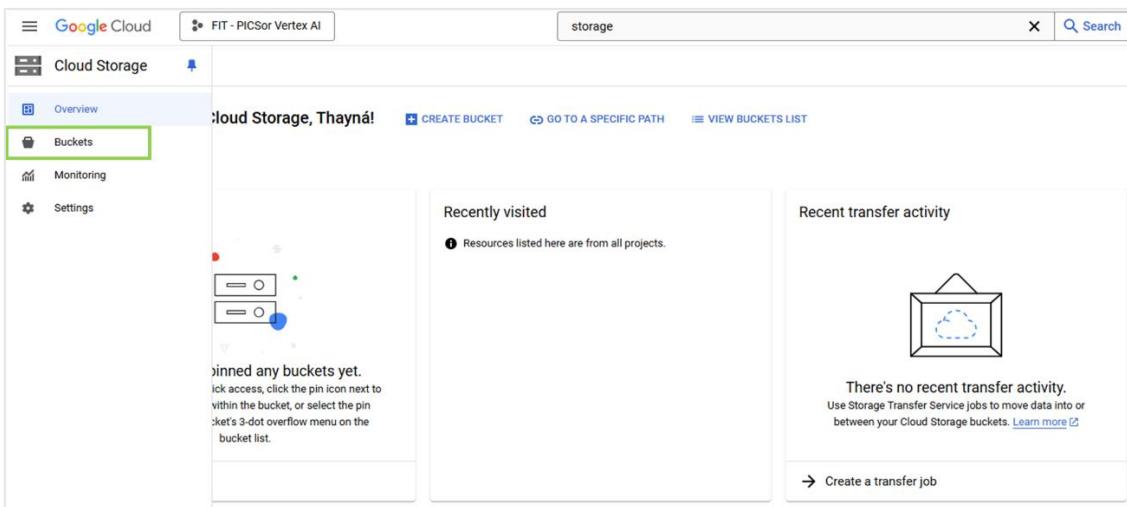


Figura 33 - Menu da ferramenta Cloud Storage. Fonte: autoral.

Os “buckets” são as pastas que podemos criar para a organização dos nossos arquivos na plataforma. Para criar um “bucket”, basta somente clicar no botão “create” na parte superior da tela, acima das pastas já existentes.

A screenshot of the Google Cloud Storage Buckets list interface. At the top left, there's a 'Buckets' header, a 'CREATE' button with a plus sign, and a 'REFRESH' button. Below the header is a 'Filter' section with a dropdown menu and a 'buckets' placeholder. The main area shows a table with four columns: Name, Created, Location type, and Location. The table contains five rows of bucket information, each with a checkbox in the first column. The columns are: Name ↑, Created, Location type, Location, Default storage class, Last modified, Public access, and Access control.

| | Name ↑ | Created | Location type | Location | Default storage class | Last modified | Public access | Access control |
|--------------------------|---|---------------------------|---------------|-------------|-----------------------|---------------------------|------------------------|----------------|
| <input type="checkbox"/> | cloud-ai-platform-dbc28cc5-ca6b-4ab6-9... | Dec 4, 2023, 11:27:41 AM | Region | us-central1 | Regional | Dec 4, 2023, 11:27:41 AM | Subject to object ACLs | Fine-grained |
| <input type="checkbox"/> | estevao_aula_vertex2 | Oct 28, 2024, 10:08:08 AM | Region | us-central1 | Standard | Oct 28, 2024, 10:08:08 AM | Not public | Uniform |
| <input type="checkbox"/> | t0_thayna_main | Apr 24, 2024, 9:38:22 AM | Region | us-central1 | Standard | Apr 24, 2024, 9:38:22 AM | Not public | Uniform |
| <input type="checkbox"/> | vertex_mod_2 | May 27, 2024, 3:08:58 PM | Region | us-central1 | Standard | May 27, 2024, 3:08:58 PM | Not public | Uniform |

Figura 34 - Localização do botão "create" para novos "buckets". Fonte: autoral.

Após clicar em “create”, é necessário preencher algumas informações de configuração da nova pasta, conforme Figura 35.

A screenshot of the 'Create a bucket' wizard. The process is divided into five steps, each highlighted with a yellow circle containing a number:

- Step 1: Get Started** (Top left): Pick a globally unique, permanent name. A text input field contains 'Ex. 'example', 'example_bucket-1', or 'example.com''. A tip below says 'Tip: Don't include any sensitive information'. Below the input field are sections for 'Optimize storage for data-intensive workloads' and 'Labels (optional)'. A 'CONTINUE' button is at the bottom.
- Step 2: Choose where to store your data** (Bottom left): Shows 'Location: us (multiple regions in United States)' and 'Location type: Multi-region'.
- Step 3: Choose a storage class for your data** (Bottom center): Shows 'Default storage class: Standard'.
- Step 4: Choose how to control access to objects** (Bottom left): Shows 'Public access prevention: On' and 'Access control: Uniform'.
- Step 5: Choose how to protect object data** (Bottom center): Shows options for 'Soft delete policy', 'Object versioning', 'Bucket retention policy', 'Object retention', and 'Encryption type: Google-managed'.

To the right of the steps is a 'Good to know' sidebar with sections for 'Location pricing' (with a link to 'Pricing details'), 'Current configuration: Multi-region / Standard', and a table showing monthly costs for different storage configurations. At the bottom of the sidebar is a 'ESTIMATE YOUR MONTHLY COST' button.

Figura 35 - Configurações na criação de um "bucket". Fonte: autoral.

- Get Started:** é onde escolhemos o Nome Único Global e Permanente da nossa pasta. Durante o curso, para fins de organização e padronização, pede-se que se utilize o seguinte padrão: **número-da-turma_sigla-do-site_nome-do-aluno**. Exemplo: **t5_sor_thayna** (turma 5 de Sorocaba, aluna Thayná).



2. **Choose where to store your data:** aqui é onde definimos em qual região de datacenters os dados ficarão armazenados. É importante salientar que, para fins do curso, é imprescindível que a região do “bucket” seja a mesma região de criação dos nossos modelos de ML, e por padrão, utilizaremos a região us-central1 (Iowa).
3. **Choose a storage class for your data:** aqui é onde definimos qual categoria de armazenamento, já vistas na Tabela 2, utilizaremos para o “bucket”. Para o curso, a categoria correta a se escolher é a “Standard”, visto que acessaremos os dados armazenados ali com frequência.
4. **Choose how to control access to objects:** aqui é onde é possível definir o nível de acesso necessário para conseguir ver e manipular os dados armazenados na pasta. Para fins do curso, deixamos as opções padrão.
5. **Choose how to protect object data:** mantemos as opções padrão.

Ao lado das configurações, é possível verificar a mudança da simulação de custos do “bucket” de acordo com as opções de localização escolhidas. Ao finalizar as configurações, basta clicar no botão azul “create” no canto inferior esquerdo da tela.

Após criado o “bucket”, você será redirecionado para dentro da pasta de detalhes do mesmo, onde é possível ver as subpastas, as configurações de criação, permissão e proteção do “bucket”, realizar o upload de arquivos e criar novas subpastas.

Figura 36 - Detalhes do "bucket". Fonte: autoral.

É importante manter a organização também dentro das subpastas, a fim de facilitar a manipulação dos dados no momento de criação dos datasets e categorização dos mesmos. A Figura 37 exemplifica a criação de uma pasta que

será utilizada para criação de um modelo de classificação de rótulo único (single-label) que será utilizado para classificar imagens de placas de modelo ESP32 entre TOP e BOTTOM. Sendo assim, foi criada uma subpasta de nome “**single_esp32**” com outras subpastas chamadas “**top**” e “**bottom**”, separando as imagens conforme suas representações.

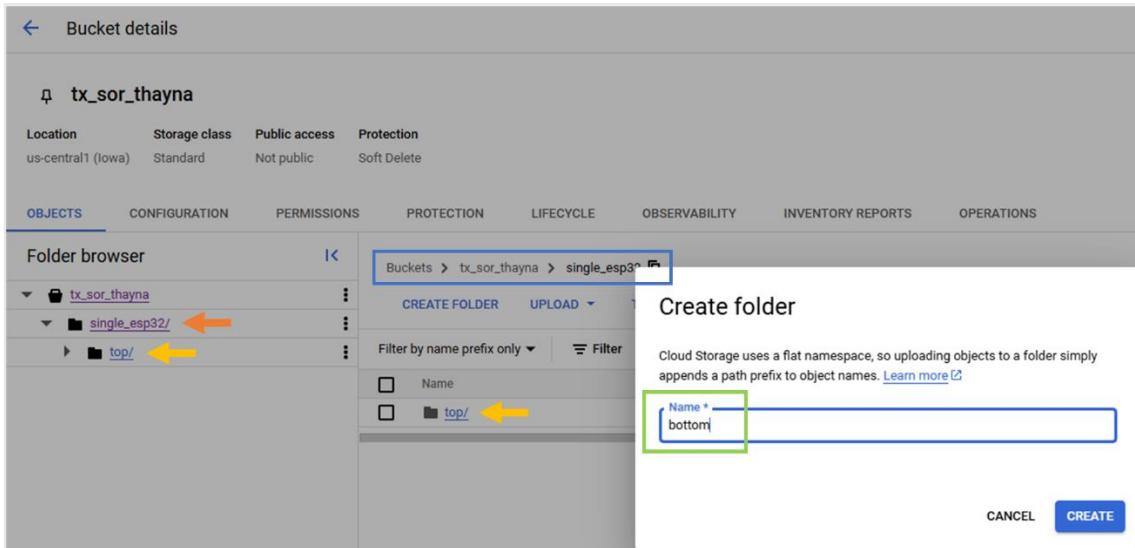


Figura 37 - Criação de subpastas. Fonte: autoral.

Também é no Cloud Storage onde os modelos “edge” criados pelo AutoML são exportados para download local, portanto, manter a organização e intuitividade no nome das pastas e subpastas é importante para facilitar a vida do desenvolvedor e cooperar para atividades realizadas com maior qualidade.

4 Construção de modelos com o AutoML

A construção de modelos de machine learning através da ferramenta AutoML envolve algumas etapas, sendo as etapas que antecedem o treinamento em si, as mais importantes. Para que o seu modelo obtenha qualidade e bons resultados, é necessário empenho na preparação e categorização dos dados, para que após o treinamento finalizado, o modelo obtenha boas notas de métricas de qualidade e possa ser utilizado para instalação final. A Figura 38 exibe o fluxo das etapas no AutoML.



Figura 38 - Etapas para criação de modelos de imagem no AutoML. Fonte: Autoral.

4.1 Preparação de conjuntos de dados

A criação de modelos no AutoML demanda a criação de “datasets” – ou conjuntos de dados – que podem ser definidos como **uma coleção organizada de dados**. Estes dados podem vir de diversas fontes e podem ser utilizados para análises estatísticas, machine learning, pesquisa científica, entre outros propósitos.

A palavra “dataset” é comumente utilizada para ciência de dados e ML para se referir a um conjunto específico de dados que está sendo analisado ou utilizado para treinar modelos. Esses conjuntos podem variar em tamanho e complexidade, e a qualidade dos resultados de análises ou modelos muitas

vezes depende da qualidade e representatividade do conjunto de dados utilizado.

É essencial que as imagens sejam representativas das classes que o modelo precisa identificar. Além disso, garantir a diversidade dos dados melhora a capacidade do modelo de lidar com variações no mundo real. Dicas para otimizar a qualidade do conjunto de dados:

- Inclua imagens de **diferentes condições** (iluminação, ângulos, planos de fundo e resoluções), quando aplicável.
- Certifique-se de que as classes estejam balanceadas para evitar viés.
- Realize uma curadoria nos dados para remover imagens redundantes ou irrelevantes.

Quando necessário, também é possível se utilizar da técnica “data augmentation”, que é uma técnica utilizada para aumentar a diversidade dos dados de treinamento de um modelo de aprendizado de máquina, evitando o sobreajuste. Esse processo consiste em aplicar transformações aleatórias nas imagens originais, como redimensionamento, rotação, espelhamento e ajustes de brilho ou contraste. Dessa forma, o modelo se torna mais robusto, aprendendo características gerais dos objetos em diferentes condições.

No exemplo da Figura 39, a técnica envolve escalonamento aleatório de $\pm 20\%$ e espelhamento horizontal em 50% das imagens durante cada época de treinamento. Isso permite que o modelo reconheça os objetos independentemente de sua orientação ou proximidade em relação à câmera.



Figura 39 - Exemplo da técnica de aumento de dados. Fonte: ResearchGate

CATEGORIAS DE TREINAMENTO

No AutoML, o conjunto de dados é dividido em três categorias: treinamento, validação e teste:

- **Conjunto de Treinamento:** Este conjunto é utilizado para treinar o modelo. Os dados aqui são fundamentais para ensinar o modelo a reconhecer padrões e estabelecer os pesos das conexões dentro da rede neural.
- **Conjunto de Validação:** Usado para ajustar os hiper parâmetros do modelo enquanto o treinamento está em andamento. Ele ajuda a evitar o “overfitting”, que ocorre quando o modelo se ajusta excessivamente aos dados de treinamento, comprometendo sua capacidade de generalizar para novos dados.
- **Conjunto de Teste:** Este conjunto é reservado exclusivamente para avaliar o desempenho do modelo após o treinamento ser concluído. Ele fornece uma medida de como o modelo se comporta com dados novos e não vistos.

A separação clara dos conjuntos de treinamento, validação e teste garante que o modelo seja avaliado de forma imparcial. Dados de validação e

teste nunca devem se sobrepor ao conjunto de treinamento para evitar métricas de desempenho irreais.

Por padrão, a divisão é distribuída com 80% dos dados destinados à treinamento, 10% destinados para validação e 10% para testes, mas esses pesos podem ser ajustados pelo usuário. Os dados podem ser distribuídos aleatoriamente ou então através da determinação do usuário no momento de carregamento dos dados na criação de datasets.

4.2 Criação de datasets no AutoML

Para partir do início, devemos navegar até a ferramenta dentro da Google Cloud Platform. Através da barra de pesquisa fixa superior, pesquise por “Vertex AI” e clique no resultado exatamente igual a busca, conforme a Figura 40.

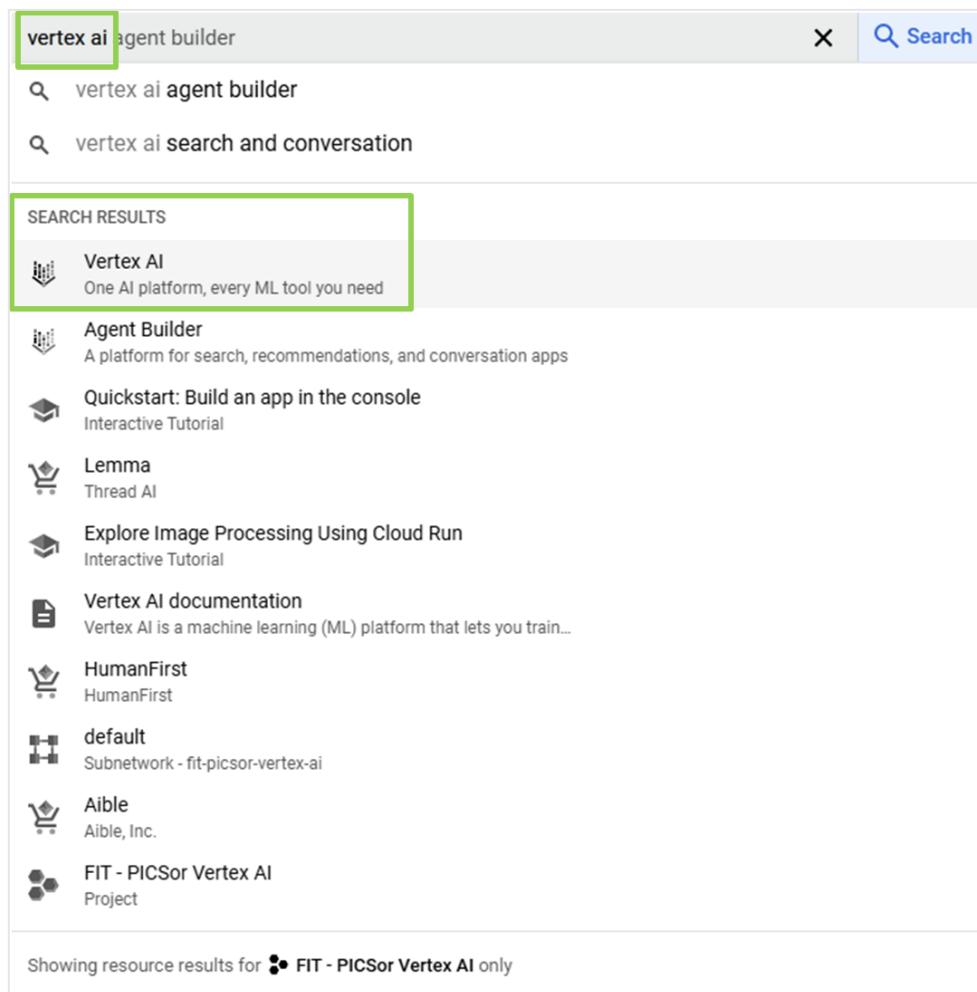


Figura 40 - Navegação até o Vertex AI. Fonte: Autoral.

Existem dois caminhos para a tela de criação dos datasets:

- Rolar a página principal para baixo, até encontrar o cartão “prepare data”, onde há um link de atalho para a criação de datasets.
- Acessar o menu lateral da ferramenta e rolar até a seção “data” e então clicar em “datasets” – esta opção levará até o menu geral de datasets, onde além de poder criar datasets também pode-se ver os já existentes. Os caminhos citados podem ser vistos na Figura 41.

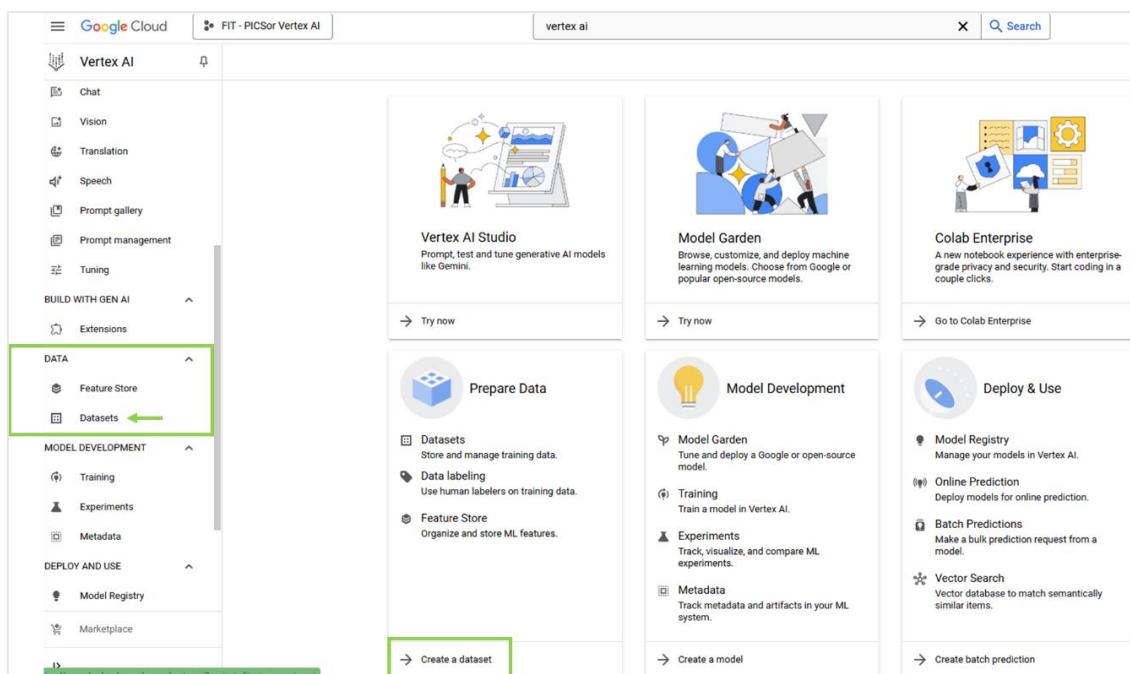


Figura 41 - Navegação para o menu "datasets". Fonte: autoral.

Para iniciar a criação de um novo dataset na plataforma, basta clicar no botão “create”, destacado na Figura 41.

| Managed datasets contain data used to train a machine learning model. Learn more | | | | | | |
|--|---------------------|--------|-------------|-------|--------------|--|
| Region us-central1 (Iowa) | | | | | | |
| Filter Enter a property name | | | | | | |
| Name | ID | Status | Region | Type | Items | |
| teste_thayna | 2668853919949520896 | Ready | us-central1 | Image | 10 images | |
| vertexmod2_rasp_heatsink_identifier | 8834737557389443072 | Ready | us-central1 | Image | 1,000 images | |
| vertexmod2_raspberry_identifier | 2624273671245529088 | Ready | us-central1 | Image | 1,500 images | |
| vertexmod2_verifone_topcase_inspection | 6298648017226432512 | Ready | us-central1 | Image | 1,726 images | |
| t0_thayna_pilot_pens | 942733364283047936 | Ready | us-central1 | Image | 104 images | |
| t0_thayna_single_esp | 6434310159907946496 | Ready | us-central1 | Image | 149 images | |

Figura 42 - Menu de datasets. Fonte: Autoral.

Ao clicar no “create”, a tela mostrada na Figura 43 será exibida, com alguns campos de preenchimento necessário para sequência na atividade:

- **DATASET NAME:** é o nome que será dado ao conjunto de dados. No curso adotaremos um padrão para fins de organização, como o exemplo: **t5_thayna_single_esp32** (turma 5 + aluna Thayná + modelo single-label + referência intuitiva ao objetivo do modelo).
- **SELECT A DATA TYPE AND OBJECTIVE:** aqui é onde escolhemos com qual tipo de dados trabalhados pelo AutoML iremos criar nosso modelo (imagem, tabular, texto ou vídeo) e onde escolhemos o objetivo do nosso treinamento (classificação de rótulo único ou multirrótulos e detecção de objetos, no caso dos treinamentos de imagem). Ambos os parâmetros foram previamente detalhados no Tópico 2 desta apostila. É importante salientar que essas escolhas **não podem** ser editadas após a criação do dataset.
- **REGION:** é a região onde será instanciado o seu conjunto de dados e obrigatoriamente deve ser a mesma região de criação dos buckets de armazenamento do conteúdo do conjunto. Como informado previamente, no decorrer do curso utilizaremos a região **us-central-1 (Iowa)**.

Para avançar para a etapa de carregamento dos dados preparados, onde iremos realizar o upload das imagens do nosso conjunto de dados, basta clicar novamente no botão azul “create”, no canto inferior esquerdo da tela.

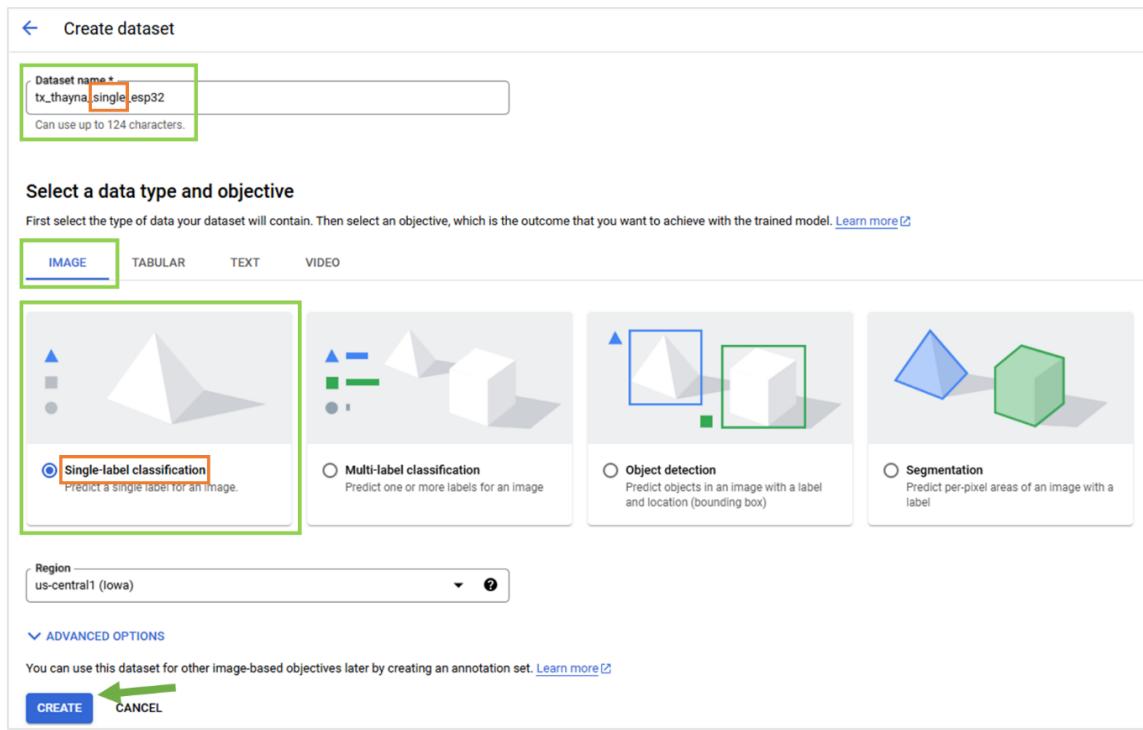


Figura 43 - Criação de dataset. Fonte: Autoral.

Ao ser criado o dataset, você será direcionado para a página de importação dos arquivos, conforme Figura 44**Erro! Fonte de referência não encontrada..**

Add images to your dataset

Before you begin, review the data guide to make sure your data is formatted correctly and optimized for the best results. Supported image file formats: JPEG, PNG, GIF, BMP, TIFF, WebP, ICO

VIEW DATA GUIDE **1**

Select an import method

- Upload images: Recommended if you don't have labels yet
- Import files: Recommended if you already have labels. An import file is a list of Cloud Storage URIs to your images and optional data, like labels. [Learn how to create an import file](#)

Upload images from your computer

Upload import files from your computer

Select import files from Cloud Storage

Summary

CUMULUS
CIRRUS
STRATUS

Image classification models predict one (or many) labels for an image. For example, identifying types of clouds from images of the sky.

Instead of creating a custom model, try Google's Vision API to detect generic objects, faces, and text. [Learn more](#)

SELECT FILES **2** **3**

Figura 44 - Adicionando imagens ao dataset. Fonte: Autoral.

Esta página possui 3 etapas: o Guia de Preparação dos Dados, a seleção do método de importação e o upload dos arquivos de importação.

- **GUIA DE PREPARAÇÃO DOS DADOS:** Traz informações bastante relevantes e pode ser utilizado como um passo-a-passo para a importação dos dados, seu conteúdo na íntegra pode ser visto na Figura 45.



The screenshot shows a step-by-step guide for preparing data. Step 1, 'Data requirements', details the need for 10 images per label, specific formats like JPEG, PNG, GIF, BMP, TIF, WebP, or ICO, and a maximum size of 1.5 MB. Step 2, 'Add labels', explains that labels are needed for training, with at least 100 images per label for few labels and at least 200 for many labels. Step 3, 'Upload images', instructs users to store images in a Cloud Storage bucket, providing steps to do so. Step 4, 'Create an import file', describes how an import file turns raw data into a dataset, containing storage paths and labels, with options to create a CSV or JSON import file. Navigation buttons 'NEXT' and 'PREVIOUS' are visible between steps.

Figura 45 - Data preparation guide. Fonte: Autoral.

- **SELEÇÃO DO MÉTODO DE IMPORTAÇÃO:** Existem 3 métodos de importar seus arquivos para o dataset:
 - **Upload de imagens do computador:** É indicado em casos em que você ainda não possui a categorização das suas imagens, e funciona de forma similar ao upload de imagens em qualquer ferramenta de armazenamento em nuvem. Podem ser carregadas até 500 imagens de uma única vez.
 - **Upload de arquivo de carga do computador:** É o upload à partir do seu computador de um arquivo de carga, que já contém as anotações de categorização, dentre outras informações que serão vistas nos tópicos seguintes deste curso.
 - **Selecionar arquivo de carga da Cloud Storage:** É o mesmo arquivo de carga com as anotações relevantes, porém, já existente em seu bucket no Cloud Storage.
- **SELEÇÃO DE ARQUIVOS PARA UPLOAD:** Esta seção tem uma interface diferente, a depender do método de importação escolhido na etapa anterior, como visto na Figura 46.



Upload images from your computer

Add up to 500 images per upload. Images will be preprocessed and stored in Cloud Storage.

[SELECT FILES](#)

Upload import files from your computer

Images referenced in the import files will be preprocessed and stored in a new Cloud Storage bucket.

[SELECT FILES](#)

Select import files from Cloud Storage

Images referenced in the import files will be preprocessed and stored in a new Cloud Storage bucket ([charges apply](#)).

Import file path * gs:// tx_sor_thayna/single_esp32/import_file/lo: [BROWSE](#) [?](#)

Data split [?](#)

[ADD ANOTHER FILE](#)

Figura 46 - Seleção de arquivos. Fonte: Autoral.

Ao selecionar os métodos de upload à partir do computador, é necessário informar o caminho de um bucket no Cloud Storage para armazenamento das imagens.

WIN_20250103_08_53_45_Pro.jpg 1 file [X](#) [?](#)

WIN_20250103_08_53_50_Pro.jpg 1 file [X](#) [?](#)

[SELECT FILES](#)

Select a Cloud Storage path

After the images are preprocessed, they'll be stored in a new Cloud Storage bucket ([charges apply](#)).

Cloud Storage path * gs:// tx_sor_thayna/single_esp32/ [BROWSE](#) [?](#)

Figura 47 - Upload de imagens do computador. Fonte: Autoral.

O campo “data split” que pode ser visto ao lado do nome dos arquivos a serem carregados na Figura 47, são relativos às categorias de divisão de dados para treinamento, previamente vistas na subseção 4.1 desta apostila. Ao deixar o campo como “default”, os dados serão distribuídos aleatoriamente pelo sistema, de acordo com a porcentagem de distribuição escolhida no momento do treinamento.

Após terminar as configurações para carregamentos dos dados, basta clicar no botão azul “continue” no canto inferior esquerdo da tela.

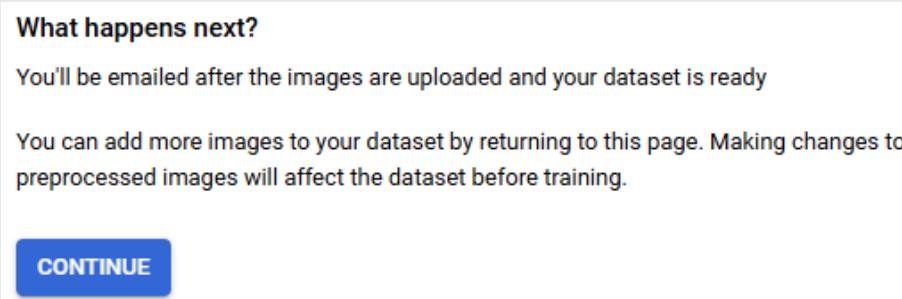


Figura 48 - Confirmação de criação de dataset. Fonte: Autoral.

O carregamento será iniciado e quando a informação “import in progress” aparecer na tela, como o exemplo da Figura 49, é possível navegar para outras seções do AutoML ou outras ferramentas do Google Cloud sem impactar o processo.

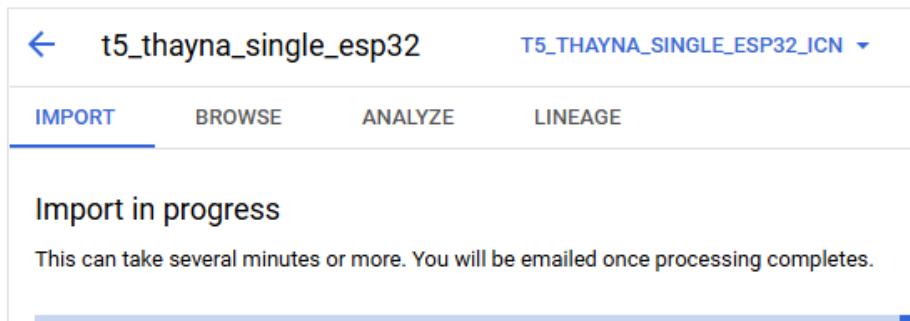


Figura 49 - Import in progress. Fonte: Autoral.

Quando a importação dos arquivos for finalizada, um e-mail é enviado ao usuário que iniciou o carregamento. O e-mail pode informar a conclusão com sucesso do processo de importação, como exemplo da Figura 50, ou também

pode indicar falhas, com uma breve mensagem sobre o problema ocorrido, para que o usuário possa analisar em mais detalhes dentro da plataforma.

Vertex AI <noreply-vertexai@google.com>
To: Thayna Almeida
Retention Policy: Inbox - 6 Months (6 months)
(i) We removed extra line breaks from this message.

This email was sent from someone outside of Flex. Do not click on any links, reply or open attachments unless you recognize the sender and know the content is safe. If suspicious, report it to spamsample@flex.com.

Hello Vertex AI Customer,

Vertex AI finished importing data into dataset "t5_thayna_single_esp32".
Additional Details:
Operation State: Succeeded ←
Resource Name:
[projects/447283747792/locations/us-central1/datasets/9000404822637150208](https://console.cloud.google.com/vertex-ai/locations/us-central1/datasets/projects/447283747792/locations/us-central1/datasets/9000404822637150208)

To continue your progress, go back to your dataset using https://urldefense.com/v3/_https://console.cloud.google.com/vertex-ai/locations/us-central1/datasets/projects/447283747792/locations/us-central1/datasets/9000404822637150208

Sincerely,
The Google Cloud AI Team

Figura 50 - E-mail de conclusão da importação. Fonte: Autoral.

Após a finalização da importação, para casos em que os dados foram importados sem utilizar do arquivo de carga com as anotações específicas, é necessário realizar a categorização dos mesmos. Veremos como categorizar/rotular as imagens na sequência.

4.3 Categorização do dataset

O processo de categorização (ou rotulação) de dados é uma etapa fundamental na criação de modelos de machine learning, especialmente na área de visão computacional, que envolve a classificação e a detecção de objetos em imagens. É o ato de atribuir etiquetas ou categorias a imagens com base em seu conteúdo.

Esse processo é essencial para treinar os modelos, pois fornece os dados necessários para que o modelo aprenda a reconhecer padrões e características específicas, pois os modelos aprendem a partir de exemplos e cada imagem rotulada serve como um exemplo do que o modelo deve aprender.

A qualidade das rotulações impacta diretamente na precisão do modelo. Rotulações incorretas ou inconsistentes podem levar a um desempenho insatisfatório, pois o modelo pode aprender padrões errados.

Existem algumas diferenças quanto à categorização de cada tipo de modelo, e veremos nos tópicos à seguir.

4.3.1 Categorização de modelos de classificação single-label

Os modelos de classificação de rótulo único precisam ter, obrigatoriamente, ao menos duas categorias para rotulação. É possível cadastrar n categorias – sempre lembrando do comportamento padrão deste modelo, visto nos tópicos iniciais desta apostila –, sendo que o modelo identificará somente uma categoria na imagem por vez.

A Figura 51 exemplifica a divisão de 2035 imagens em três categorias distintas. Perceba que, para cada imagem, o modelo reconhece apenas uma única categoria, cuja qual obteve maior pontuação no momento de inferência. Observe que somente o objeto de nominação está em destaque, contribuindo assim para o melhor aprendizado do modelo e assertividade nos resultados futuros.



Figura 51 - Divisão de categorias e exemplos de rótulo único. Fonte: Autoral.

A PRÁTICA NO AUTOML

Na ferramenta AutoML, ao finalizar a importação dos arquivos para o dataset, a página será redirecionada para a tela de categorização. Nesta tela, seguindo as indicações da Figura 52, temos:

1. **Quantidades de imagens:** aqui é possível ver a quantidade total de imagens, a quantidade total de imagens **já rotuladas** e a quantidade total de imagens **não-rotuladas**.
2. **Categorias:** aqui ficam dispostas todas as categorias criadas com o número total de imagens rotuladas com ela em frente ao seu nome.
3. **Imagens:** aqui são todas as imagens importadas para o dataset. É possível filtrar por categoria, alterar o método de visualização e definir o número de imagens visualizados por página no rodapé.
4. **Atalho para treinamento:** é possível enviar um modelo para treinamento por aqui, porém, somente se já houver ao menos duas categorias criadas com pelo menos dez imagens rotuladas em cada uma.

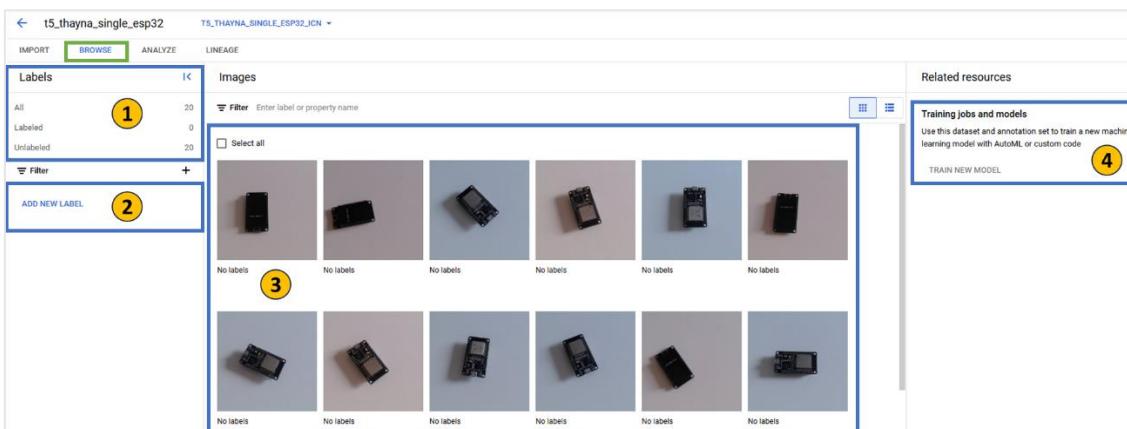


Figura 52 - Tela de categorização. Fonte: Autoral.

Para iniciar o processo de categorização, devemos primeiro criar as categorias necessárias, seguindo os passos 1 e 2 da Figura 53.

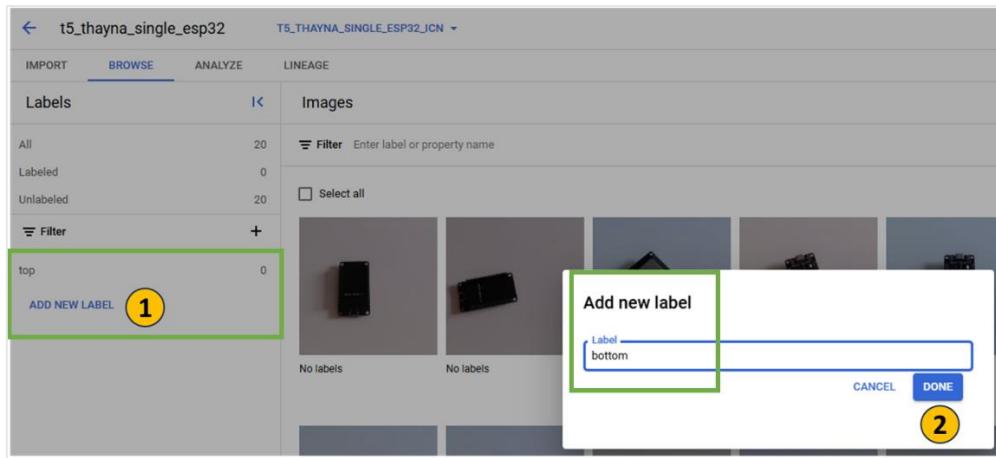


Figura 53 - Criação de categorias. Fonte: Autoral.

Após criadas as categorias, seguimos para a seleção das imagens para categorização. Indica-se filtrar somente pelas imagens não-rotuladas, para evitar sobreescriver imagens já rotuladas com categorias erradas.

É de suma importância a atenção em selecionar somente imagens que correspondam à categoria em processo para não causar incoerências no treinamento do modelo.

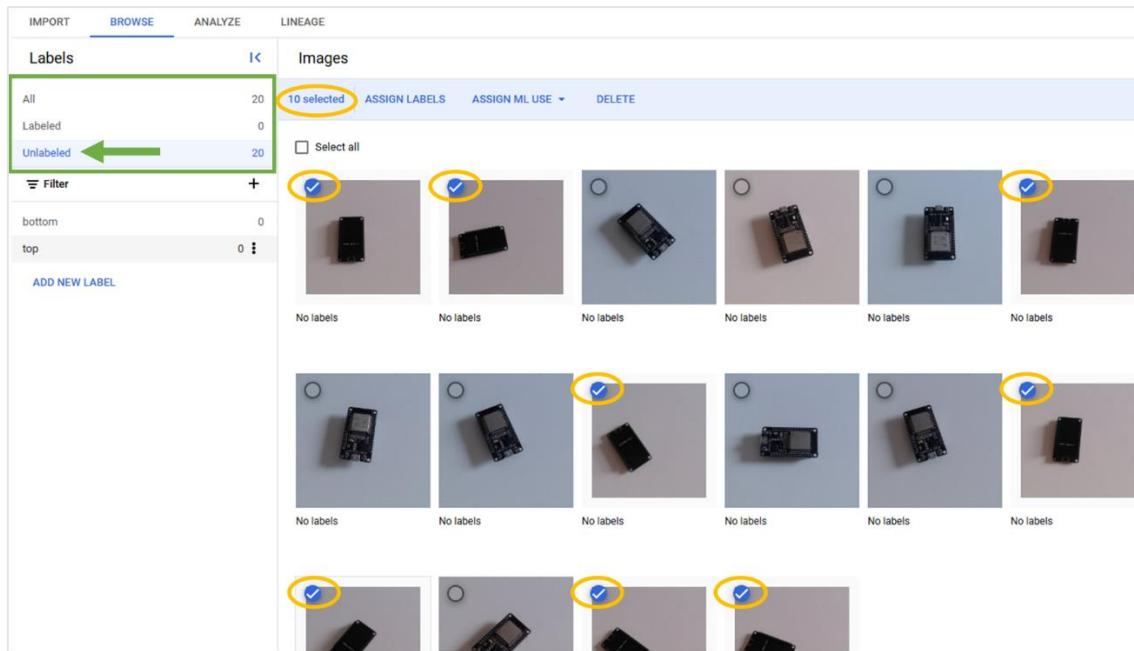


Figura 54 - Seleção de imagens não-rotuladas. Fonte: Autoral.

Após selecionadas as imagens, a opção “assign labels” vista no destaque 1 da Figura 55 ficará disponível. Clicando nesta opção, as categorias criadas

surgirão como no destaque 2, onde basta selecionar a categoria correta e clicar no botão azul “save”.

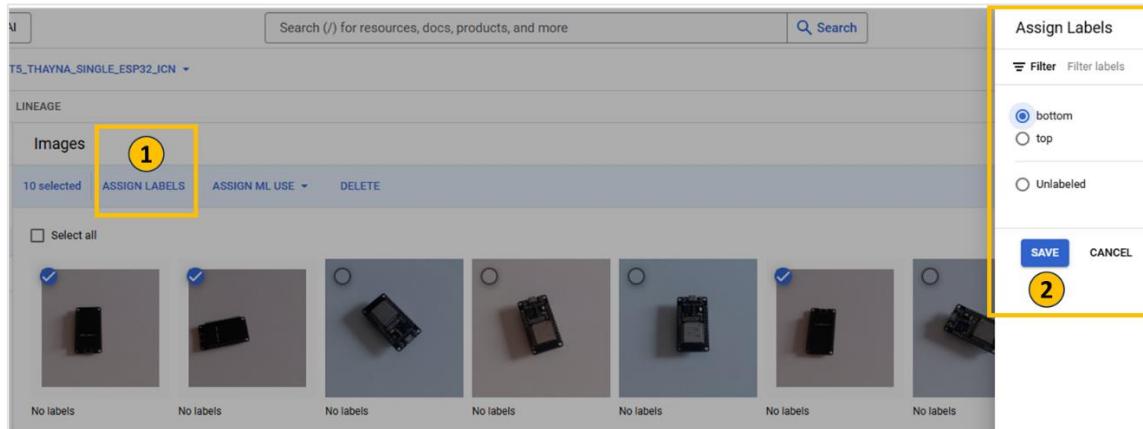


Figura 55 - Seleção de categorias únicas para imagens selecionadas. Fonte: Autoral.

A tela ficará conforme Figura 56 após todas as imagens terem sido rotuladas. É possível ver que 20 imagens de um total de 20 foram categorizadas, e que cada categoria possui o total de 10 imagens cada.

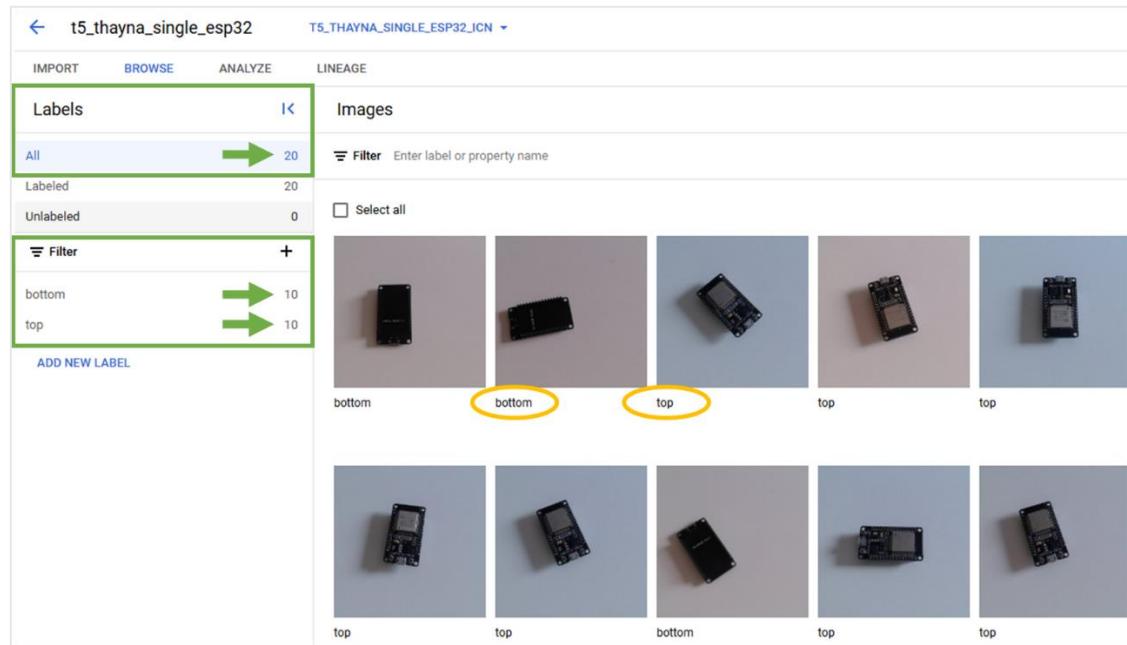


Figura 56 - Imagens para modelo single-label rotuladas. Fonte: Autoral.

O próximo tópico detalhará o processo de categorização de modelos de classificação multi-label.

4.3.2 Categorização de modelos de classificação multi-label

Os modelos de classificação multirrótulos também precisam ter, obrigatoriamente, ao menos duas categorias para rotulação. É possível cadastrar n categorias – sempre lembrando do comportamento padrão deste modelo, visto nos tópicos iniciais desta apostila –, sendo que o modelo poderá identificar mais de uma categoria na imagem por inferência.

A Figura 57 exemplifica a divisão de 1414 imagens em três categorias distintas. Note que, para cada imagem, o modelo pode identificar uma ou mais categorias, de acordo com as categorias de seu conhecimento e os objetos vistos nas figuras.

| CATEGORIA | ESCALADORES | ROCHAS | EQUIPAMENTOS | ROCHAS | EQUIPAMENTOS | ROCHAS | EQUIPAMENTOS |
|-----------|-------------|--------|--------------|--------|--------------|--------|--------------|
| EXEMPLOS | | | | | | | |
| | | | | | | | |
| | | | | | | | |

Figura 57 - Divisão de categorias e exemplos de multirrótulo. Fonte: Autoral.

A PRÁTICA NO AUTOML

Na ferramenta AutoML, ao finalizar a importação dos arquivos para o dataset, a página será redirecionada para a tela de categorização, já explicada no tópico anterior através da Figura 52. A criação de categorias também é idêntica à já detalhada pela Figura 53.

Após criadas as categorias, seguimos para a seleção das imagens para categorização. Indica-se filtrar somente pelas imagens não-rotuladas, para evitar sobre escrever imagens já rotuladas com categorias erradas.

É de suma importância a atenção em selecionar somente imagens que correspondam às categorias em processo para não causar incoerências no treinamento do modelo.

Após selecionadas as imagens, a opção “assign labels” vista no destaque 1 da Figura 58 ficará disponível. Clicando nesta opção, as categorias criadas surgirão como no destaque 2, onde basta selecionar as categorias corretas (podendo neste caso, ser mais de uma) e clicar no botão azul “save”.

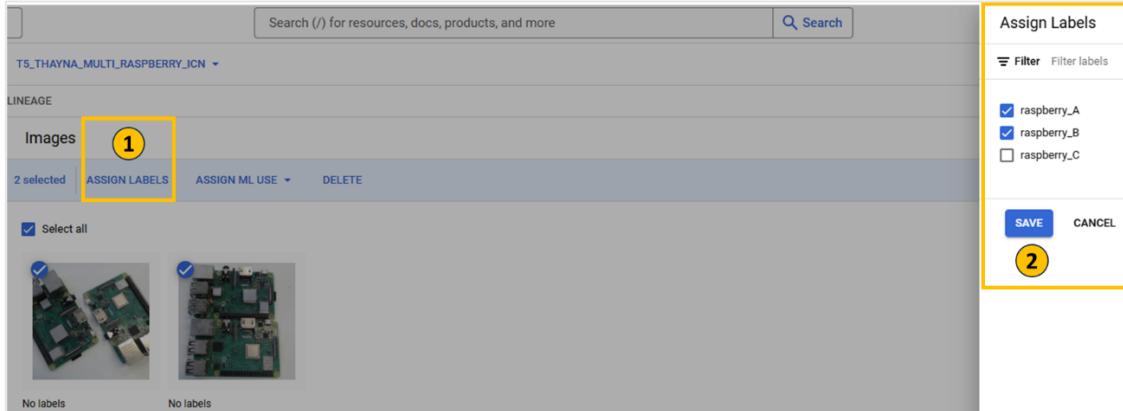


Figura 58 - Seleção de multi categorias para imagens selecionadas. Fonte: Autoral.

A tela ficará conforme Figura 59 após todas as imagens terem sido rotuladas. É possível ver que 42 imagens de um total de 42 foram categorizadas, onde 2 categorias possuem 18 imagens cada, e 1 categoria possui 19 imagens. Também é possível observar no quadrante das imagens, imagens com etiquetas individuais e imagens com multi-etiquetas.

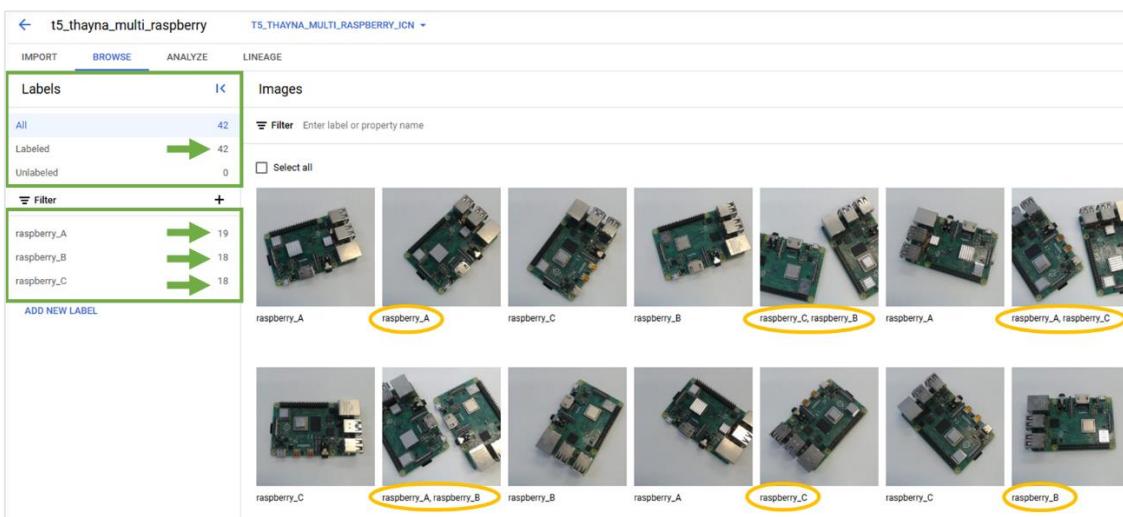


Figura 59 - Imagens para modelo multi-label rotuladas. Fonte: Autoral.

O próximo tópico detalhará o processo de categorização de modelos de detecção de objetos.

4.3.3 Categorização de modelos de detecção de objetos

Em modelos de detecção de objetos, as imagens são rotuladas não somente com a categoria do objeto, mas também com a sua localização na imagem (com um retângulo delimitador ao redor do objeto de interesse). Ele também precisa ter, obrigatoriamente, ao menos duas categorias para rotulação.

É possível cadastrar n categorias – sempre lembrando do comportamento padrão deste modelo, visto nos tópicos iniciais desta apostila –, sendo que o modelo poderá identificar e delimitar mais de uma categoria na imagem por inferência.

A PRÁTICA NO AUTOML

Na ferramenta AutoML, ao finalizar a importação dos arquivos para o dataset, a página será redirecionada para a tela de categorização, já explicada no tópico anterior através da Figura 52. A criação de categorias também é idêntica à já detalhada pela Figura 53.

Após criadas as categorias, seguimos para a etapa de categorização. Indica-se filtrar somente pelas imagens não-rotuladas, para evitar sobreescriver imagens já rotuladas com informações incorretas.

O método de categorização de modelos de detecção de objetos é bastante distinto dos modelos de classificação de imagens uma vez que, além de rotular a categoria dos objetos, também precisamos delimitá-lo nas imagens.

Para iniciar o processo de categorização, deve-se clicar em uma imagem qualquer e aguardar redirecionamento para a tela mostrada na Figura 60, onde temos os seguintes pontos a se considerar:

- 1. Propriedades da imagem:** informações como caminho do armazenamento no Cloud Storage, formato da imagem e dimensões.

2. **Área de manejo da imagem:** com opções como ajuste de brilho e contraste, rotação da imagem, e o ícone para adição de uma caixa delimitadora (ou “bounding box”).
3. **Botão “add label”:** serve para criação de categorias. Indica-se criar as categorias previamente, na tela já vista pela Figura 53.
4. **Seleção de “data split”:** aqui é possível designar a categoria de uso da imagem (entre treinamento, validação, teste ou “default”).
5. **Área de delimitação de objetos:** é onde se desenha as caixas delimitadoras ao redor dos objetos de interesse.
6. **Botão “save”:** botão para salvar as alterações realizadas na categorização da imagem.

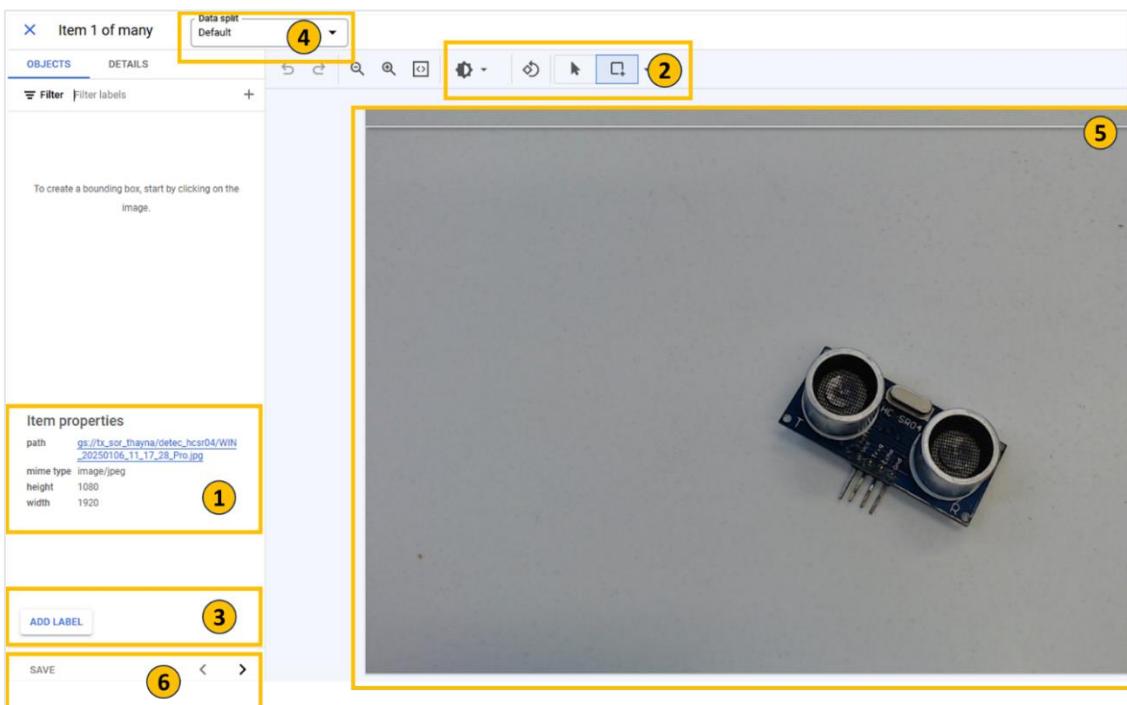


Figura 60 - Tela de rotulação para modelo de detecção de objetos. Fonte: Autoral.

Para criar a área demarcada na figura, basta posicionar o mouse sobre o local e clicar sobre a tela para arrastar a caixa delimitadora. Após soltar o clique do mouse, uma caixa com as opções de categorias previamente criadas aparecerá para escolha. Caso a categoria do objeto ainda não esteja criada, há um atalho para criação. O processo descrito pode ser visualizado na Figura 61.

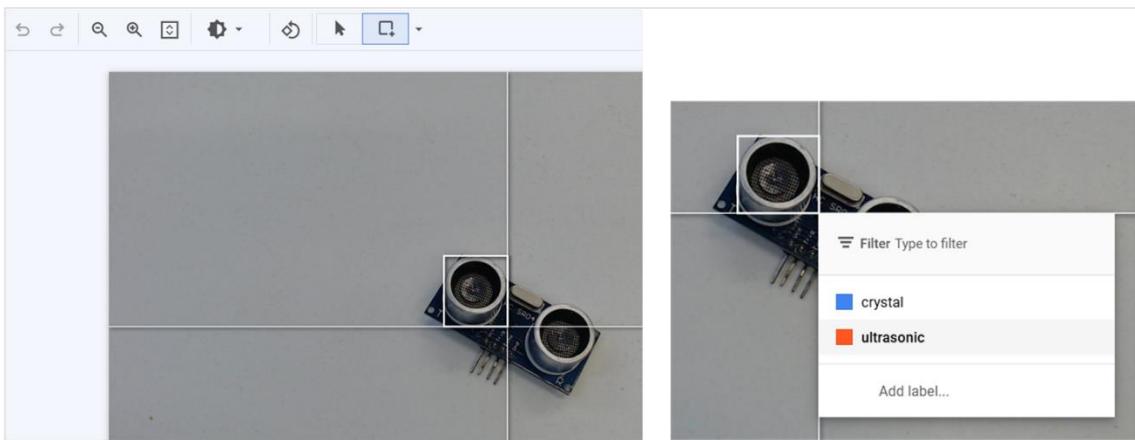


Figura 61 - Criando caixas delimitadoras. Fonte: Autoral.

Quando um novo objeto é demarcado, ele é contabilizado na aba de categorias daquela imagem, como mostra o destaque 1 da Figura 62. É importante selecionar corretamente a categoria da imagem demarcada, e se atentar a demarcação extremamente ajustada do objeto, evitando cortar pedaços do mesmo, ou incluir margens ao redor do objeto. Caso haja caixas delimitadoras se cruzando, como o destaque 2 da imagem, não há problemas.

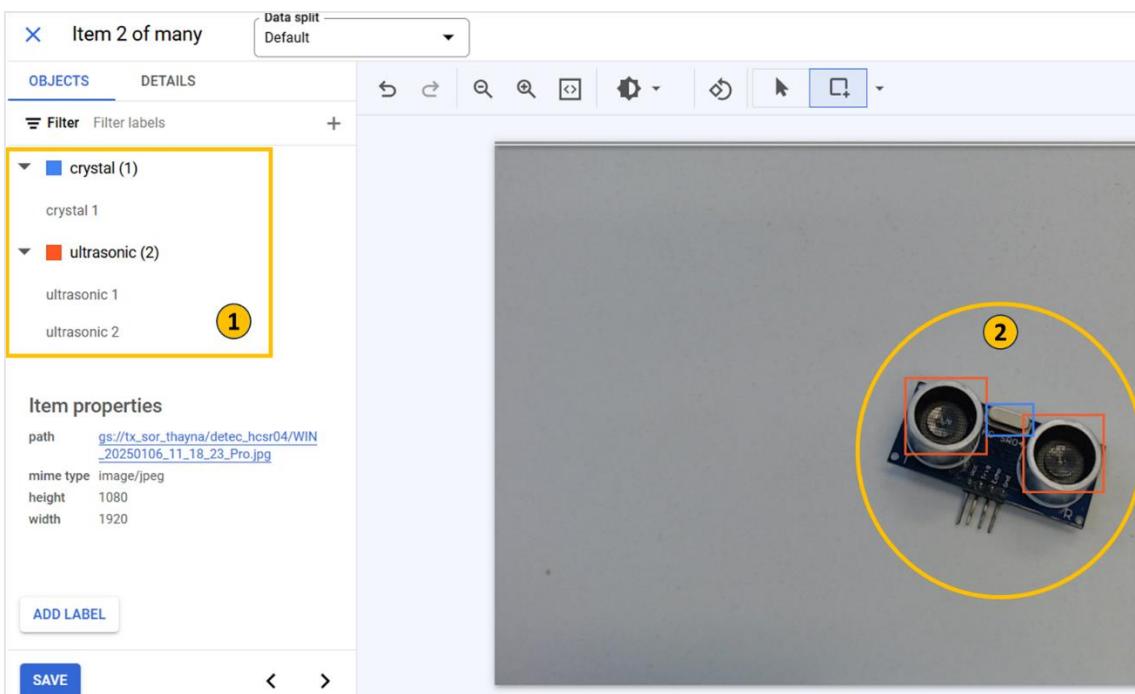


Figura 62 - Imagem pós delimitação de objeto. Fonte: Autoral.

A tela ficará conforme Figura 63 após todas as imagens terem sido rotuladas. É possível ver que 20 imagens de um total de 21 foram categorizadas, onde 2 categorias possuem 20 imagens cada. Também é possível observar no quadrante das imagens a quantidade de categorias que cada uma delas possui, conforme as demarcações realizadas.

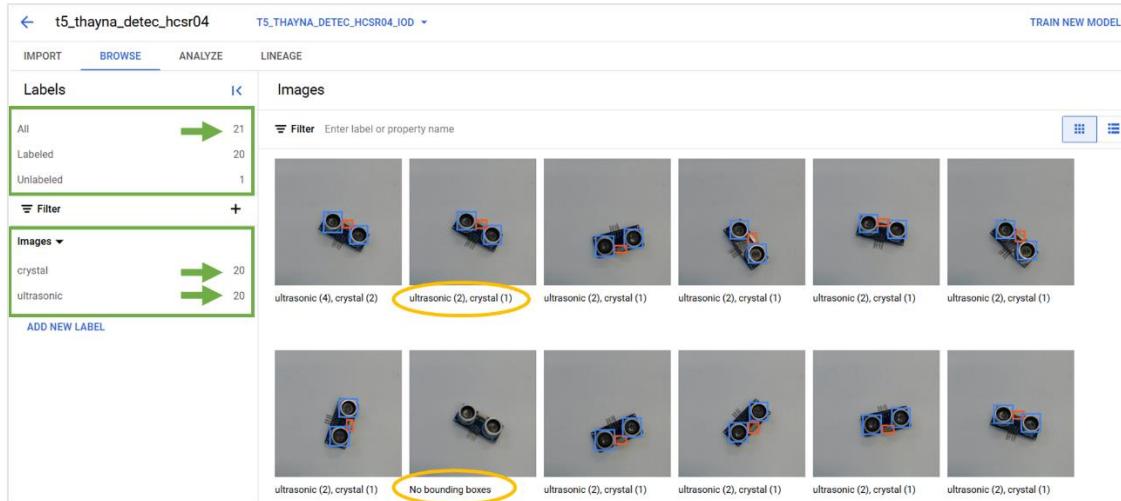


Figura 63 - Imagens para modelo de detecção de objetos rotuladas. Fonte: Autoral.

O processo de demarcação de objetos em imagens manualmente pode ser muito moroso, cansativo e sujeito a falhas humanas e, para contornar isso, podemos utilizar a opção de criação de datasets através dos já citados arquivos de carga. A operação de uso dos mesmos será detalhada no tópico à seguir.

4.4 Criação de datasets categorizados com arquivos de carga

Na subseção anterior vimos como realizar anotações dos objetos manualmente, após o carregamento do dataset no AutoML, porém, podemos fazer nossas anotações um arquivo de carga e realizar o upload deste arquivo para a plataforma. Este arquivo pode ser feito no formato CSV (Comma Separated Values) ou em formato JSON (Javascript Object Notation).

O formato CSV é semelhante à uma tabela com os campos de cada anotação em colunas, ao qual pode ser feito no Excel, mas também se recomenda o uso de scripts em alguma linguagem de programação (como Python) para facilitar e agilizar a criação do arquivo. Para o curso, será provido

um template de script preparado para capturar imagens e realizar suas notações em um arquivo CSV de forma automática.

The screenshot shows a user interface for importing files from a computer or cloud storage. The main area displays a list of files with their paths:

- gs://vertex_examples/img_1719584978.jpg, Ethernet, 0.16250000, 0.42708333, 0.35468750, 0.42708333, 0.35468750, 0.76666667, 0.16250000, 0.76666667
- gs://vertex_examples/img_1719584978.jpg, USB2, 0.61718750, 0.48541667, 0.55937500, 0.48541667, 0.55937500, 0.78125000, 0.39687500, 0.78125000
- gs://vertex_examples/img_1719584981.jpg, Ethernet, 0.16250000, 0.42708333, 0.35468750, 0.42708333, 0.35468750, 0.76666667, 0.16250000, 0.76666667
- gs://vertex_examples/img_1719584981.jpg, USB2, 0.61718750, 0.48541667, 0.55937500, 0.48541667, 0.55937500, 0.78125000, 0.39687500, 0.78125000
- gs://vertex_examples/img_1719584981.jpg, USB2, 0.61718750, 0.48541667, 0.77968750, 0.48541667, 0.77968750, 0.78125000, 0.61718750, 0.78125000
- gs://vertex_examples/img_1719584985.jpg, USB1, 0.16250000, 0.42708333, 0.35468750, 0.42708333, 0.35468750, 0.76666667, 0.16250000, 0.76666667
- gs://vertex_examples/img_1719584985.jpg, USB1, 0.39687500, 0.48541667, 0.55937500, 0.48541667, 0.55937500, 0.78125000, 0.39687500, 0.78125000
- gs://vertex_examples/img_1719584988.jpg, USB1, 0.39687500, 0.48541667, 0.55937500, 0.48541667, 0.55937500, 0.78125000, 0.39687500, 0.78125000
- gs://vertex_examples/img_1719584988.jpg, USB2, 0.61718750, 0.48541667, 0.55937500, 0.48541667, 0.55937500, 0.78125000, 0.61718750, 0.78125000
- gs://vertex_examples/img_1719584988.jpg, USB2, 0.61718750, 0.48541667, 0.77968750, 0.48541667, 0.77968750, 0.78125000, 0.61718750, 0.78125000
- gs://vertex_examples/img_1719584991.jpg, USB1, 0.39687500, 0.48541667, 0.55937500, 0.48541667, 0.55937500, 0.78125000, 0.39687500, 0.78125000
- gs://vertex_examples/img_1719584991.jpg, USB2, 0.61718750, 0.48541667, 0.55937500, 0.48541667, 0.55937500, 0.78125000, 0.61718750, 0.78125000
- gs://vertex_examples/img_1719584994.jpg, Ethernet, 0.16250000, 0.42708333, 0.35468750, 0.42708333, 0.35468750, 0.76666667, 0.16250000, 0.76666667
- gs://vertex_examples/img_1719584994.jpg, USB1, 0.39687500, 0.48541667, 0.55937500, 0.48541667, 0.55937500, 0.78125000, 0.39687500, 0.78125000
- gs://vertex_examples/img_1719584994.jpg, USB2, 0.61718750, 0.48541667, 0.77968750, 0.48541667, 0.77968750, 0.78125000, 0.61718750, 0.78125000
- gs://vertex_examples/img_1719838294.jpg, teste2, 0.15625000, 0.20833333, 0.50625000, 0.20833333, 0.50625000, 0.67500000, 0.15625000, 0.67500000
- gs://vertex_examples/img_1719838294.jpg, teste2, 0.15625000, 0.20833333, 0.50625000, 0.20833333, 0.50625000, 0.67500000, 0.15625000, 0.67500000
- gs://vertex_examples/img_1719838903.jpg, , 0.15625000, 0.20833333, 0.50625000, 0.20833333, 0.50625000, 0.67500000, 0.15625000, 0.67500000
- gs://vertex_examples/img_1719840875.jpg, A, 0.15625000, 0.20833333, 0.50625000, 0.20833333, 0.50625000, 0.67500000, 0.15625000, 0.67500000
- gs://vertex_examples/img_1719840875.jpg, B, 0.15625000, 0.20833333, 0.50625000, 0.20833333, 0.50625000, 0.67500000, 0.15625000, 0.67500000
- gs://vertex_examples/img_1719840878.jpg, A, 0.15625000, 0.20833333, 0.50625000, 0.20833333, 0.50625000, 0.67500000, 0.15625000, 0.67500000
- gs://vertex_examples/img_1719840878.jpg, B, 0.15625000, 0.20833333, 0.50625000, 0.20833333, 0.50625000, 0.67500000, 0.15625000, 0.67500000

Figura 64 - Exemplificação de arquivo de carga. Fonte: Autoral.

As seguintes informações são requeridas no arquivo de carga:

[ML_USE], **[GCS_FILE_PATH]**, **[LABEL]**, **[BOUNDING_BOX]**

Onde:

- **ML_USE**: É a finalidade da divisão dos dados ao treinar um modelo – as categorias de treinamento vistas no Tópico 4. Use **TRAINING**, **TEST** ou **VALIDATION** (campo opcional).

Exemplos:

```
test,gs://bucket/filename1.jpeg, Tomato, 0.3, 0.3,, 0.7, 0.6,,  
training,gs://bucket/filename2.gif, Tomato, 0.8, 0.2,, 1.0, 0.4,,  
gs://bucket/filename2.gif  
gs://bucket/filename3.png, Baked, 0.5, 0.7, 0.8, 0.7, 0.8, 0.8, 0.5, 0.8  
validation,gs://bucket/filename4.tif, Salad, 0.1, 0.2,, 0.8, 0.9,,
```

- **GCS_FILE_PATH**: É o caminho URI do diretório no Cloud Storage de armazenamento das imagens.

Exemplos:

```
test,gs://bucket/filename1.jpeg, Tomato, 0.3, 0.3,, 0.7, 0.6,,  
training,gs://bucket/filename2.gif, Tomato, 0.8, 0.2,, 1.0, 0.4,,  
gs://bucket/filename2.gif  
gs://bucket/filename3.png, Baked, 0.5, 0.7, 0.8, 0.7, 0.8, 0.8, 0.5, 0.8
```

validation,gs://bucket/filename4.tiff,Salad,0.1,0.2,,0.8,0.9,,

- **LABEL:** Os rótulos devem começar com uma letra e conter apenas letras, números e underscores.

Exemplos:

test,gs://bucket/filename1.jpeg,Tomato,0.3,0.3,,0.7,0.6,,
 training,gs://bucket/filename2.gif,Tomato,0.8,0.2,,1.0,0.4,,
 gs://bucket/filename2.gif
 gs://bucket/filename3.png,Baked,0.5,0.7,0.8,0.7,0.8,0.8,0.5,0.8
 validation,gs://bucket/filename4.tiff,Salad,0.1,0.2,,0.8,0.9,,

- **BOUNDING_BOX:** Uma caixa delimitadora para um objeto na imagem (valores entre 0 e 1, indicando as coordenadas da anotação).

Exemplos:

test,gs://bucket/filename1.jpeg,Tomato,**0.3,0.3,,0.7,0.6,,**
 training,gs://bucket/filename2.gif,Tomato,**0.8,0.2,,1.0,0.4,,**
 gs://bucket/filename2.gif
 gs://bucket/filename3.png,Baked,**0.5,0.7,0.8,0.7,0.8,0.8,0.5,0.8**
 validation,gs://bucket/filename4.tiff,Salad,**0.1,0.2,,0.8,0.9,,**

A coluna **BOUNDING_BOX** deve ter o conjunto de coordenadas de todos os objetos de interesse na imagem. Na Figura 65 vemos a representação dos 4 vértices que compõe a caixa delimitadora.

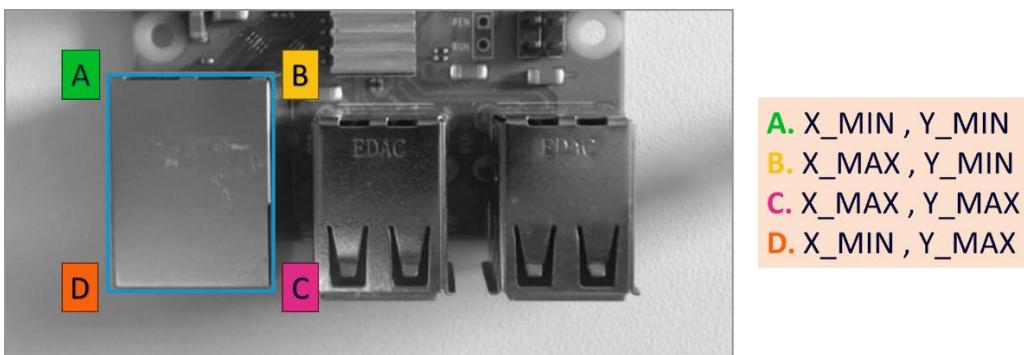


Figura 65 - Coordenadas de anotação. Fonte: Autoral.

Cada vértice é definido pelos valores de X e Y, e estes valores são equivalentes às coordenadas normalizadas entre [0, 1], onde [0,0] é o X_MIN ou Y_MIN e [1,1] é o X_MAX ou Y_MAX.

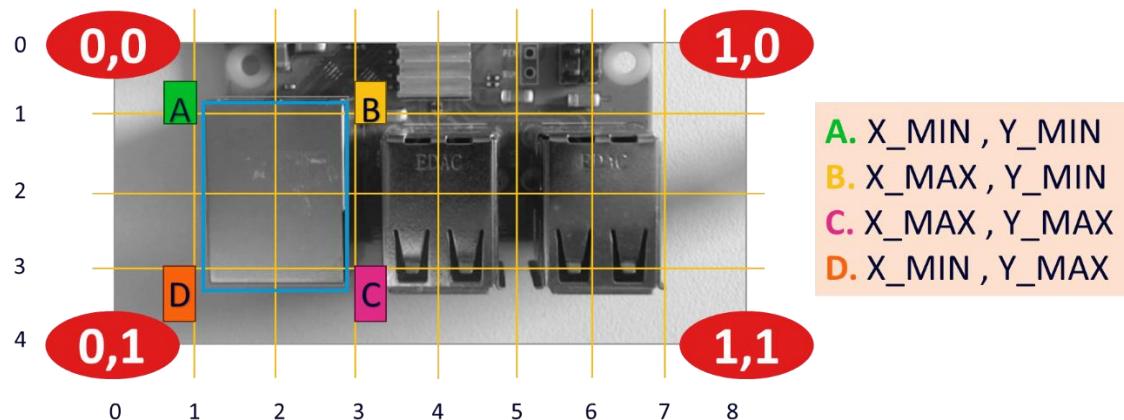


Figura 66 - Exemplo com coordenadas normalizadas. Fonte: Autoral.

A caixa delimitadora também pode ser especificada com somente 2 vértices diagonalmente opostos do retângulo (vértices A e C, representados na Figura 67). O AutoML irá automaticamente especificar os vértices ocultos que complementam o retângulo.

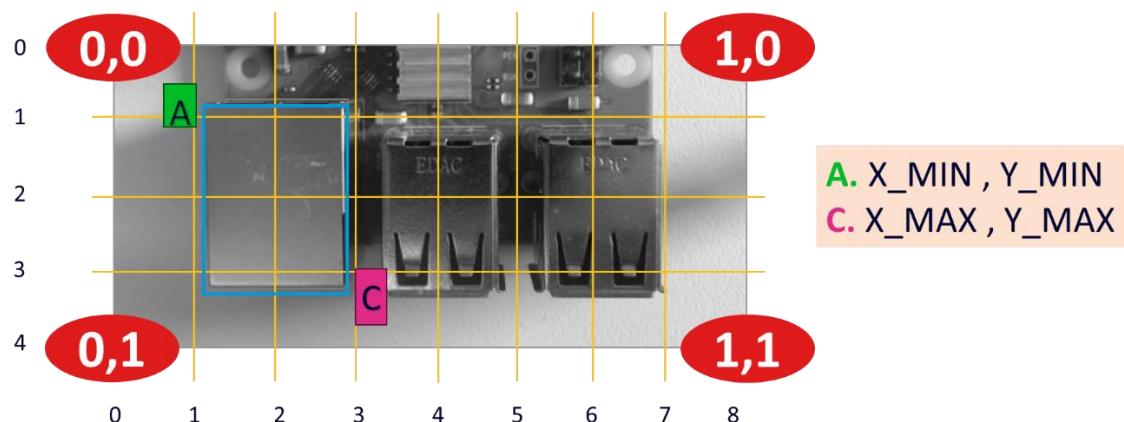


Figura 67 - Exemplo anotação com 2 coordenadas. Fonte: Autoral.

Abaixo temos 2 exemplos de anotação no arquivo de carga, sendo um deles utilizando 2 vértices e o outro utilizando 4 vértices. Note que, apesar dos vértices B e D estarem ocultos no primeiro exemplo, as vírgulas de delimitação de suas colunas no arquivo de carga devem estar presentes.

2 VÉRTICES = training,gs://bucket/filename2.gif,Tomato,**0.8,0.2,,1.0,0.4,,**

4 VÉRTICES = gs://bucket/filename3.png,Baked,**0.5,0.7,0.8,0.7,0.8,0.8,0.5,0.8**

4.5 Utilizando o template de geração de arquivo de carga

Para o período do curso, foi provido o template de um script em Python, para auxiliar o aluno na criação de datasets para o modelo de detecção de objetos utilizando arquivos de carga. A interface gráfica do script pode ser vista na Figura 68.

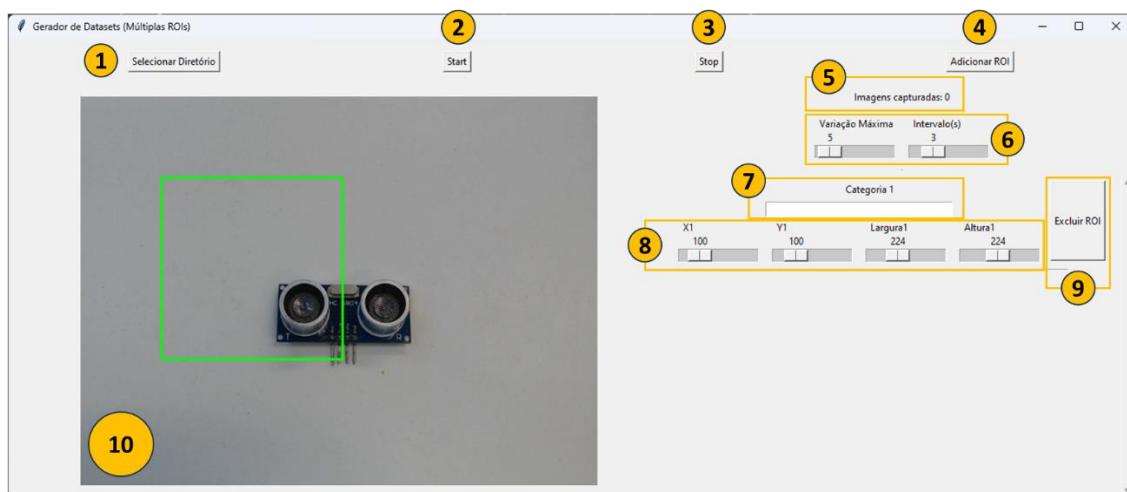


Figura 68 - Interface gráfica do script de geração de dataset. Fonte: Autoral.

Abaixo estão detalhadas as propriedades da interface gráfica:

1. **Selecionar Diretório:** irá abrir uma janela para que seja escolhido o diretório de armazenamento das imagens capturadas. O programa iniciará com erros se a pasta não for previamente selecionada.
2. **Start:** é onde se dá início ao processo de captura das imagens e criação do arquivo de carga, após configurados todos os detalhes prévios.
3. **Stop:** é onde se finaliza o processo de captura das imagens e criação do arquivo de carga.
4. **Adicionar ROI:** é onde se cria mais caixas delimitadoras de regiões de interesse. O programa sempre inicia com 1 ROI, conforme a Figura 68, e é possível adicionar e excluir ROIs conforme necessário. A Figura 69 demonstra um exemplo com mais de uma região de interesse.

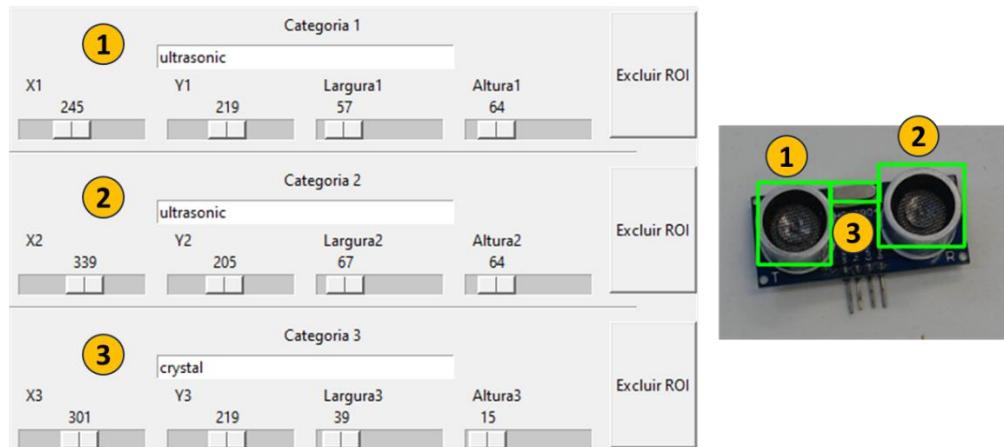


Figura 69 - Imagem com múltiplas regiões de interesse. Fonte: Autoral.

5. **Imagens capturadas:** onde é exibida a quantidade de imagens que já foram capturadas e armazenadas no diretório definido.
6. **Variação Máxima e Intervalo(seg):** a variação máxima é utilizada para evitar o problema do sobreajuste, já mencionado na subseção 4.1. O valor definido neste campo representa a variação das coordenadas desenhadas para as caixas delimitadoras na imagem, para que as anotações não sejam repetidas e viciem o modelo. O intervalo(seg) é o tempo definido entre uma captura de imagem e a próxima.
7. **Categoria:** é onde deve ser informado o nome do rótulo/categoria da caixa delimitadora correspondente.
8. **Definição das coordenadas do ROI:** é onde são definidos os pontos das coordenadas da caixa delimitadora. Sempre inicie definindo os pontos X e Y iniciais, e somente após isso modifique os parâmetros de altura e largura do retângulo.

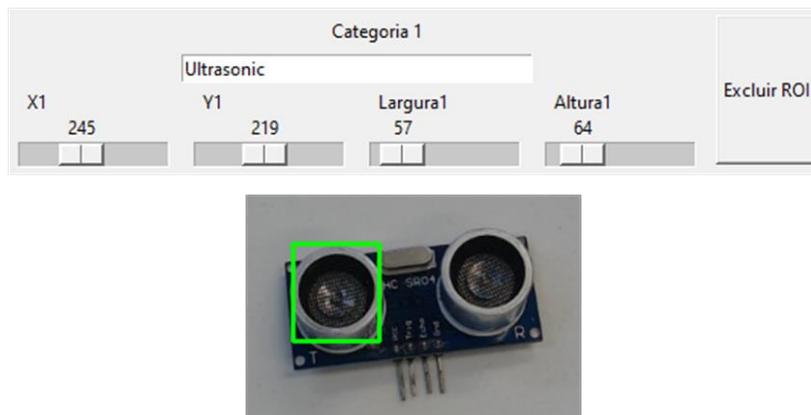


Figura 70 - Coordenadas do bounding box e categoria modificados. Fonte: Autoral.

9. **Excluir ROI:** exclui novas caixas de delimitação criadas e que não serão utilizadas.
10. **Visualização da câmera:** onde é exibida a imagem sendo vista pela câmera e as caixas delimitadoras existentes em seus locais atuais.

EXEMPLO PRÁTICO

O exemplo à seguir visa realizar a demarcação dos componentes “transmissor ultrassônico”, “receptor ultrassônico” e “cristal de oscilação” do sensor de modelo HC-SR04.

Primeiramente, foi selecionado o diretório de armazenamento das imagens capturadas – no mesmo repositório do código, foi criada a pasta “loadfile” para armazenamento.

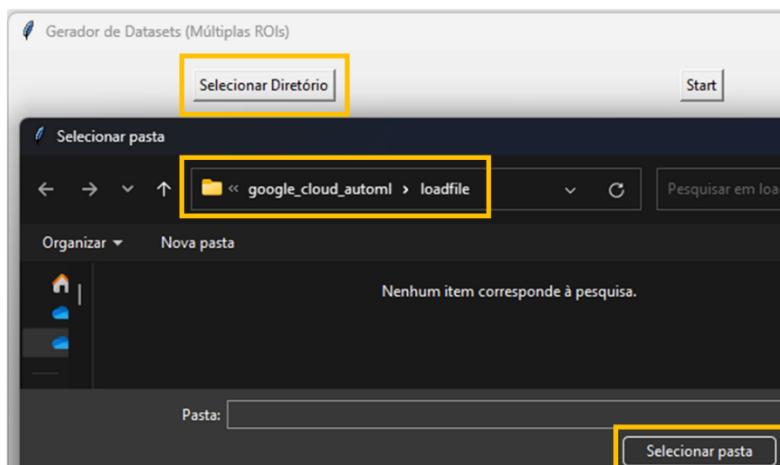


Figura 71 - Seleção de diretório de armazenamento. Fonte: Autoral.

Após, foram criadas 3 regiões de interesse – 2 para os transmissores ultrassônicos e 1 para o cristal oscilador –, e alinhadas as caixas delimitadoras de forma ajustada aos componentes na visualização da câmera. E por fim, ajustada a variação máxima para 3 e o intervalo de captura para 2 segundos.

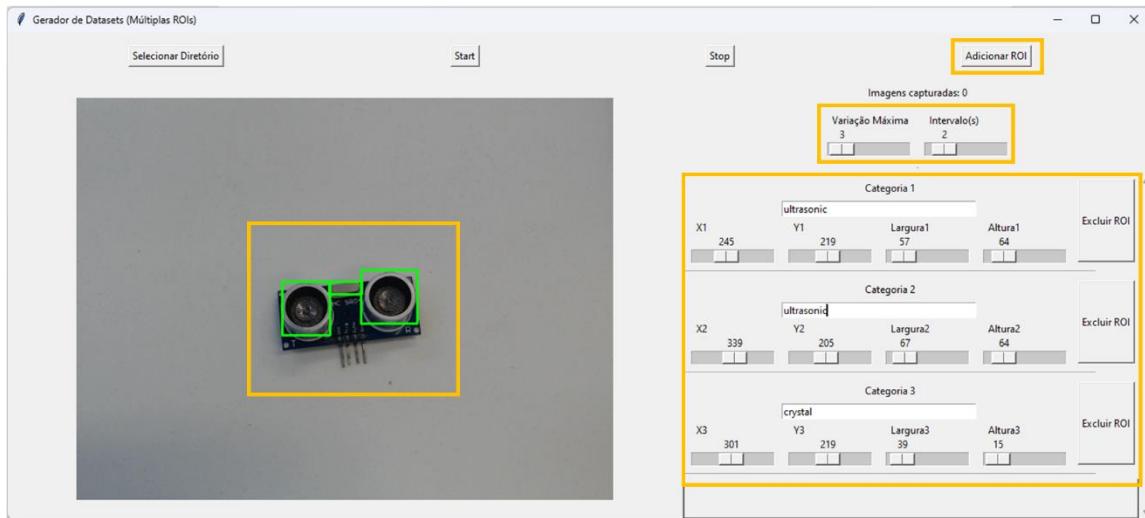


Figura 72 - Exemplo prático com o script de demarcação. Fonte: Autoral.

Foram capturadas 20 imagens com o sensor na posição configurada, e após isso, o programa foi parado através do botão “stop”. Em seguida, o sensor foi reposicionado e novamente ajustadas as caixas delimitadoras, para captura de novas 20 imagens, totalizando 40 imagens capturadas e 120 linhas no arquivo de carga. Note que, ao parar um programa com o botão “stop” e recomeçar com o botão “start”, o mesmo arquivo de carga continuará sendo preenchido, portanto, sempre se lembre de renomear o arquivo de carga antes de iniciar um dataset diferente, ou excluir o arquivo antigo do diretório para evitar misturar dados e corromper a integridade do seu arquivo.

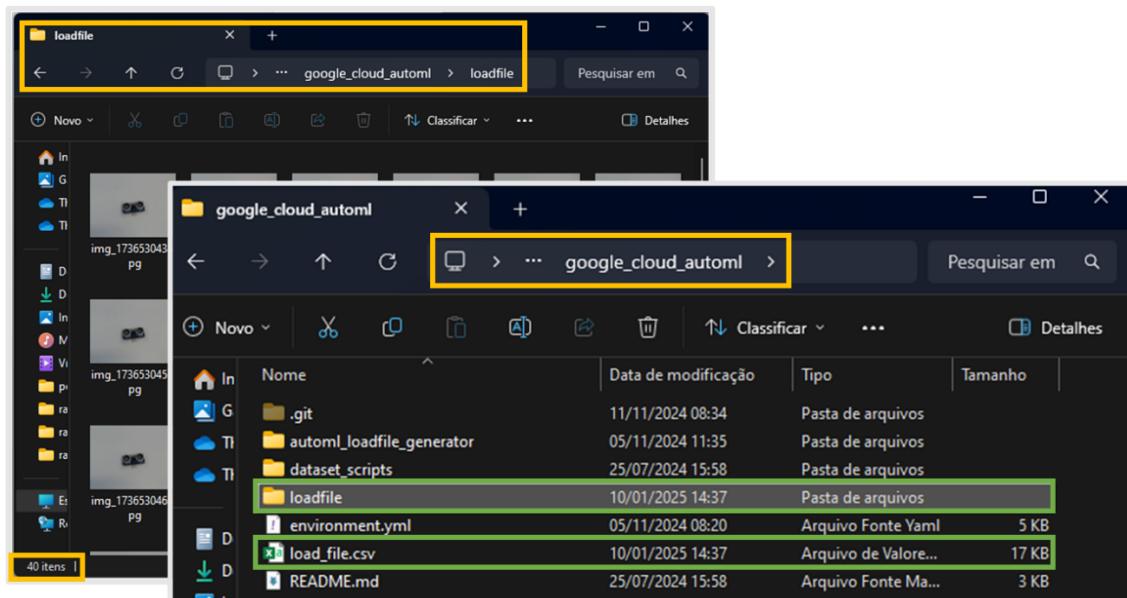


Figura 73 - Organização de arquivos e pastas no diretório. Fonte: Autoral.

A Figura 74 exibe o resultado do arquivo de carga. Note que cada imagem possui 3 linhas no arquivo, pois neste exemplo, cada imagem possuía 3 caixas delimitadoras.

| | |
|-----|--|
| 115 | gs://t6_thayna/detec/img_1736530667.jpg.ultrasonic,0.39687500,0.41250000,0.48593750,0.41250000,0.48593750,0.54583333,0.39687500,0.54583333 |
| 116 | gs://t6_thayna/detec/img_1736530667.jpg.ultrasonic,0.54375000,0.39791667,0.64843750,0.39791667,0.64843750,0.53125000,0.54375000,0.53125000 |
| 117 | gs://t6_thayna/detec/img_1736530667.jpg.crystal,0.48593750,0.51458333,0.54687500,0.51458333,0.54687500,0.54583333,0.48593750,0.54583333 |
| 118 | gs://t6_thayna/detec/img_1736530669.jpg.ultrasonic,0.39687500,0.41250000,0.48593750,0.41250000,0.48593750,0.54583333,0.39687500,0.54583333 |
| 119 | gs://t6_thayna/detec/img_1736530669.jpg.ultrasonic,0.54375000,0.39791667,0.64843750,0.39791667,0.64843750,0.53125000,0.54375000,0.53125000 |
| 120 | gs://t6_thayna/detec/img_1736530669.jpg.crystal,0.48593750,0.51458333,0.54687500,0.51458333,0.54687500,0.54583333,0.48593750,0.54583333 |
| 121 | |

Figura 74 - Exemplo de arquivo de carga. Fonte: Autoral.

Lembre-se, antes de subir seu arquivo de carga em uma importação na criação de datasets, é necessário antes realizar o upload das imagens capturadas em um diretório do Cloud Storage. Caso contrário, o arquivo não encontrará as imagens para realizar as anotações na plataforma e resultará em erros.

Outro ponto de atenção para evitar erros é assegurar que o nome do diretório do Cloud Storage está corretamente configurado no script Python. O nome deve estar exatamente igual em ambos os locais – script e Cloud Storage –, e não se deve editar o que foi escrito no arquivo posteriormente, para evitar corromper o mesmo.

Figura 75 - Relevância do path para o Cloud Storage. Fonte: Autoral.

Na próxima seção, veremos como realizar o treinamento de modelos de classificação e detecção de objetos.

4.6 Treinamento de modelos Edge

Concluídas as etapas de criação do conjunto de dados e categorização do dataset na plataforma AutoML, podemos então seguir para a parametrização do treinamento que será realizado pela plataforma. Daremos preferência ao uso do menu específico de treinamento de modelos, embora haja um atalho após a finalização da categorização do dataset para o mesmo. A página de treinamentos está na seção “Model Development”, sob o nome “training”, conforme Figura 76.

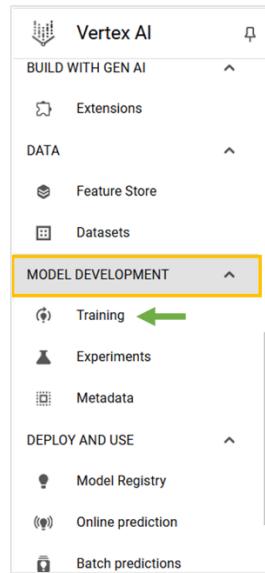


Figura 76 - Navegação para a página de treinamentos. Fonte: Autoral.

Para o curso, utilizaremos somente a aba “training pipelines”, que é onde podemos ver todos os treinamentos já realizados e podemos criar um novo treinamento à partir de um dataset já categorizado.

| Name | ID | Status | Job type | Model type | Duration | Last updated | Created | Ended | Labels |
|--------------------|---------------------|----------|-------------------|-------------------------------------|---------------|--------------------------|-------------------------|--------------------------|--------|
| t6_sabrina_multi | 5038434346915069952 | Finished | Training pipeline | Image classification (Multi-label) | 58 min 39 sec | Nov 5, 2024, 3:25:33 PM | Nov 5, 2024, 2:26:53 PM | Nov 5, 2024, 3:25:33 PM | — |
| t6_welinton_single | 1950653852399173632 | Finished | Training pipeline | Image classification (Single-label) | 58 min 39 sec | Nov 5, 2024, 3:20:02 PM | Nov 5, 2024, 2:21:22 PM | Nov 5, 2024, 3:20:02 PM | — |
| t6_nicolly_multi | 2969030318138327040 | Finished | Training pipeline | Image classification (Multi-label) | 48 min 36 sec | Nov 5, 2024, 3:16:54 PM | Nov 5, 2024, 2:28:17 PM | Nov 5, 2024, 3:16:54 PM | — |
| t5_gabriel_single | 28328470413049856 | Finished | Training pipeline | Image classification (Single-label) | 1 hr 3 min | Oct 2, 2024, 10:46:56 AM | Oct 2, 2024, 9:43:19 AM | Oct 2, 2024, 10:46:56 AM | — |

Figura 77 - Página "training" na seção "training pipelines". Fonte: Autoral.

Para iniciar a criação de um novo modelo, basta clicarmos no botão “train new model” no canto superior esquerdo da tela. Uma aba surgirá na tela para preenchimento dos parâmetros da primeira parte da configuração do treinamento, conforme Figura 78.

Train new model

- 1** Training method
- 2** Model details
- 3** Training options
- 4** Compute and pricing

Dataset * tx_thayna_detec_hcsr04 (40 images) **1**

Annotation set * tx_thayna_detec_hcsr04_iod **2**

Objective Image object detection **3**

Please refer to the pricing guide for more details (and available deployment options) for each method.

Model training method **4**

AutoML
Train high-quality models with minimal effort and machine learning expertise. Just specify how long you want to train. [Learn more](#)

Custom training (advanced)
Run your TensorFlow, scikit-learn, and XGBoost training applications in the cloud. Train with one of Google Cloud's pre-built containers or use your own. [Learn more](#)

Choose where to use the model **5**

Cloud
Deploy to an endpoint for online predictions or use for batch predictions.

Edge
Export for on-prem and on-device use. Typically has lower accuracy.

Vertex AI Vision
Export to Vertex AI Vision to analyze streaming image or video data.

CONTINUE

Figura 78 - Treinamento de novo modelo: parte 1. Fonte: Autoral.

Os parâmetros iniciais são os seguintes:

1. **Dataset:** conforme já mencionado, para realizar o treinamento de um modelo é preciso primeiro criar um conjunto de dados e categorizá-lo. Neste campo, é onde selecionamos o dataset que será utilizado para treinamento do modelo.
2. **Annotation set:** as categorias são salvas em coleções chamadas “annotation sets” e é possível alterar o “annotation sets” para aplicar um diferente grupo de categorias para um mesmo dataset. Isso permite que sejam criados diferentes tipos de modelo utilizando o mesmo dataset.
3. **Objective:** é o tipo de modelo que será treinado. Essa opção já vem preenchida automaticamente.
4. **Model training method:** é possível utilizar um treinamento mais avançado e customizado, que exige mais experiência do usuário, ou utilizar o AutoML que gerará um treinamento de qualidade com o mínimo de esforço por parte do usuário.

5. **Choose Where to use the model:** o tipo de deploy do modelo deve ser informado antes de seu treinamento, e existem as opções “Cloud” (onde a plataforma irá gerar um endpoint para realização das predições), “Edge” (onde será possível exportar o modelo para uso offline) e “Vertex AI Vision” (somente para modelos de detecção de objetos, onde o modelo será treinado para uma melhor performance em predições em streaming). No decorrer do curso, sempre utilizaremos a opção Edge por ser a mais adequada para implementações em chão de fábrica.

Clicando em “continue”, teremos então os parâmetros da Figura 79 para configurar:

1. **Train new model ou Train new version:** é possível escolher entre criar um modelo do zero, ou retreinar um modelo já existente e gerar uma nova versão dele. Posteriormente é possível navegar entre versões no menu de deploy.
2. **Name:** o nome do modelo será preenchido automaticamente de acordo com o nome do dataset selecionado para criação do mesmo. É possível adicionar uma breve descrição para facilitar o entendimento do objetivo do treinamento.
3. **Data split:** é onde são definidas as categorias de treinamento de cada imagem do conjunto de dados, conforme explicado na subseção 4.1. Existem 2 modalidades de divisão:
 - i. **Random:** o dataset será dividido aleatoriamente entre “treinamento”, “validação” e “teste” de acordo com as porcentagens definidas. O padrão é 80% para treinamento, 10% para validação e 10% para teste, mas esses valores podem ser alterados mantendo o total de 100%.
 - ii. **Manual (advanced):** o dataset será dividido manualmente utilizando as definições realizadas no momento de carregamento e/ou categorização das imagens.
4. **Encryption:** é a opção de definição das chaves de segurança. No decorrer do curso, utilizamos o padrão que é gerenciado pela Google.

5. **Model ID:** é possível definir um ID customizado para o modelo ou permitir que o google gere um ID único aleatório.

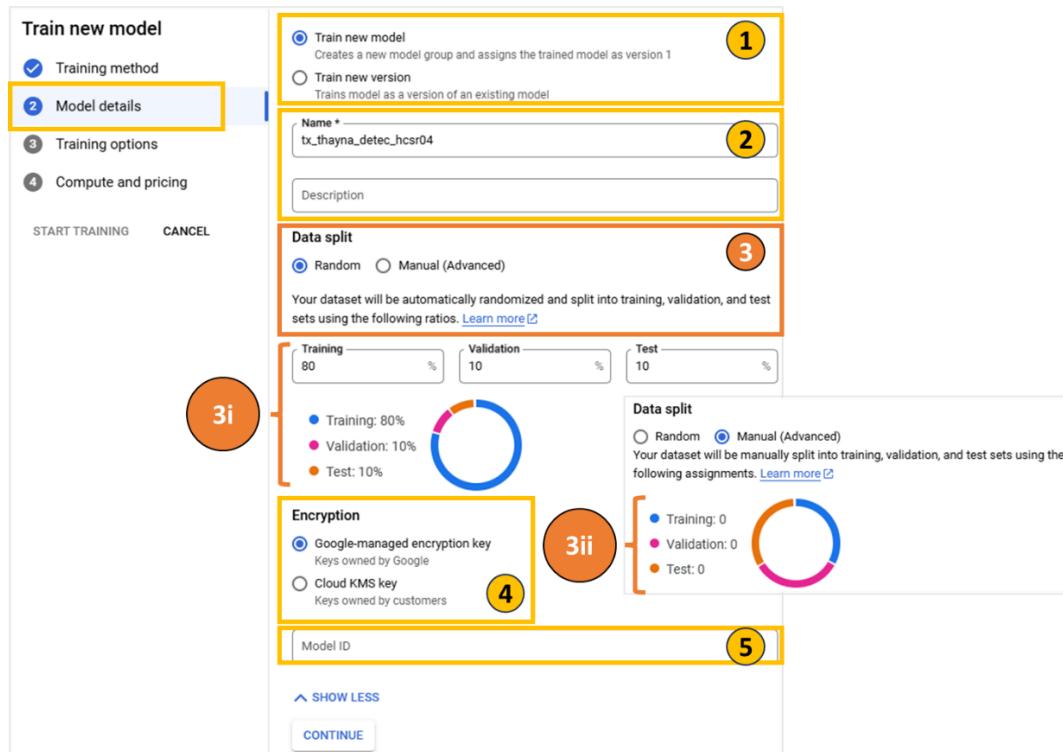


Figura 79 - Treinamento de novo modelo: parte 2. Fonte: Autoral.

Clicando em “continue”, teremos então os parâmetros apresentados na Figura 80 para configurar:

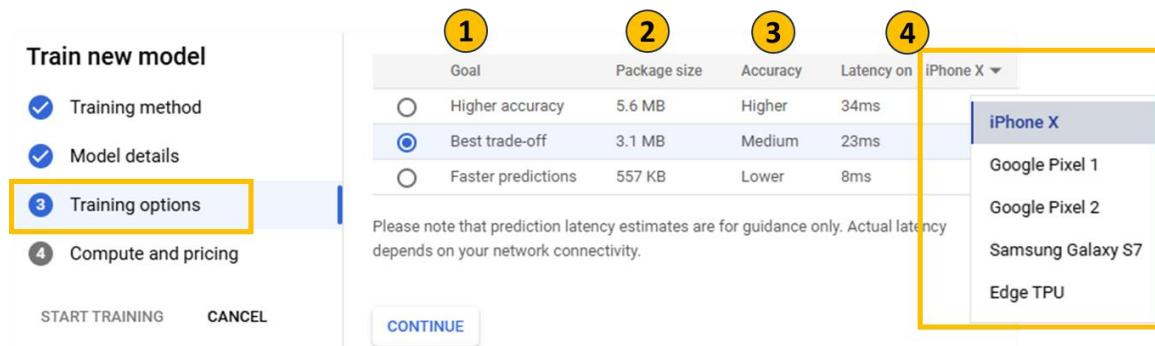


Figura 80 - Treinamento de novo modelo: parte 3. Fonte: Autoral.

1. **Goal:** é onde se escolhe sua prioridade com este treinamento. Existe a opção com maior acurácia nos resultados, a opção de previsões mais rápidas e uma opção intermediária, que é um equilíbrio entre precisão e

eficiência de processamento. A opção utilizada no decorrer do curso é a “best trade-off”.

2. **Package size:** é o tamanho esperado do arquivo do modelo para cada categoria.
3. **Accuracy:** é o nível de acurácia/precisão do modelo que se espera para cada categoria.
4. **Latency:** é o tempo em milissegundos que se espera que o modelo leve para realizar as previsões. É possível alterar o dispositivo de estimativa da latência.

Clicando em “continue”, teremos então os parâmetros apresentados na Figura 81 para configurar:

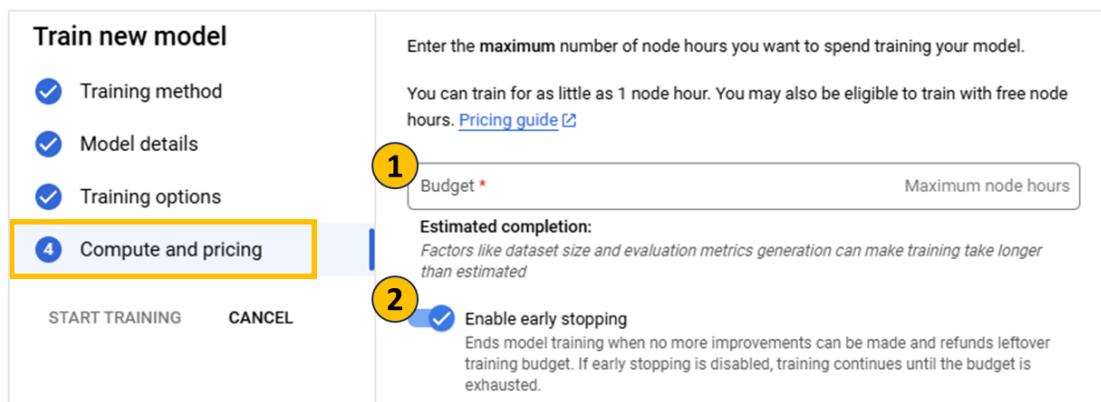


Figura 81 - Treinamento de novo modelo: parte 4. Fonte: Autoral.

1. **Budget:** neste campo especificamos qual a quantidade máxima de “node hours” deseja-se gastar no treinamento deste modelo. É importante se atentar que o tamanho do dataset e as métricas de geração do treinamento podem impactar neste tempo e fazê-lo ser maior que o planejado. Para o curso, utiliza-se **1 para treinamentos de classificação** e **2 para treinamentos de detecção** de objetos.
2. **Enable early stopping:** finaliza o treinamento do modelo caso não haja mais melhorias a serem feitas e reembolsa o tempo economizado ao usuário.

Um “node hour” representa o tempo em que uma máquina virtual executará o seu modelo de previsão. Por exemplo: uma execução que utiliza 1



máquina virtual pelo tempo de 1 hora gerará um custo de 1 “node hour”. Porém, se um endpoint for executado com 4 máquinas virtuais ativas por 24 horas, isso gerará um custo de 96 “node hours”. Esse exemplo é aplicado em casos em que o deploy é feito em Cloud.

Após o preenchimento de todos os parâmetros de todas as 4 etapas, basta clicar no botão azul “start training” para dar início ao treinamento automatizado.

4.6.1 Visualização dos modelos criados

Os modelos existentes – em processo de treinamento ou já finalizados – podem ser vistos na página “Training”, conforme já demonstrado na Figura 77. Abaixo veremos os detalhes de visualização da página de modelos, conforme destaque numerados na Figura 82.

| Name | ID | Status | Job type | Model type | Duration | Last updated | Created | Ended | Labels |
|-----------------------|---------------------|----------|-------------------|------------------------|--------------|--------------------------|--------------------------|-------|--------|
| tx_thyra_detec_hcon04 | 5342742194404982784 | Training | Training pipeline | Image object detection | 1 min 51 sec | Jan 14, 2025, 11:44:15AM | Jan 14, 2025, 11:44:15AM | — | — |

Figura 82 - Detalhes da página "trainings". Fonte: Autoral.

1. **Name:** é o nome do modelo que foi treinado.
2. **ID:** é o ID único criado automaticamente pelo AutoML para identificação do modelo.
3. **Status:** é o status atual do modelo. Pode ser “finished”, “stopped” ou “failed”.
4. **Job type:** é o processo realizado com o modelo.
5. **Model type:** é o tipo de modelo que foi treinado.
6. **Duration:** é o tempo total em que o modelo esteve em treinamento.
7. **Last update:** é a data e horário da última atualização do modelo.
8. **Created:** é a data e horário de criação do treinamento do modelo.
9. **Ended:** é a data e horário de finalização do treinamento do modelo.



10. Neste atalho é possível pausar treinamentos em execução ou deletar treinamentos finalizados.

Após o treinamento de um modelo estar com status “finished”, é possível avaliar suas métricas de desempenho e exportá-lo para deploy offline.

5 Análise de métricas de qualidade

Ao ser finalizado o treinamento do modelo, podemos acessar seus detalhes e métricas de qualidade clicando no nome do modelo dentro da tela “trainings” da seção “model development” ou navegar até a seção “deploy and use” onde a tela “model registry” exibirá uma lista com todos os modelos já treinados dentro da organização da plataforma.

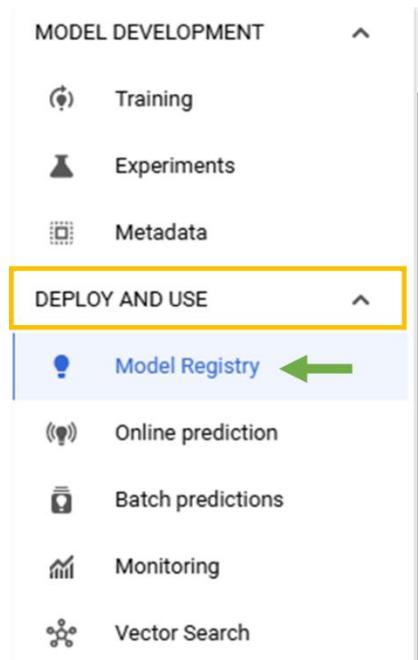


Figura 83 - Navegação para a página de modelos treinados. Fonte: Autoral.

O método de validação irá depender do tipo de dados e do objetivo do treinamento, nesse tutorial será apresentado apenas a validação no conceito de classificação de imagens. Nos tópicos seguintes veremos como avaliar a qualidade final do modelo treinado, verificando sua capacidade de gerar previsões corretas e usar exemplos de teste e métricas padronizadas que nos ajudam a entender sua performance.

5.1 Interpretação da saída de modelos

Para cada imagem que o modelo analisa, ele produz valores que nos indicam qual o nível de confiança do modelo no resultado da predição de

categoria/rótulo obtida. Se o número for alto, significa que o modelo tem muita confiança de que o rótulo se aplica bem à imagem.

Por exemplo, se o modelo analisa uma foto de um cachorro e atribui um número alto ao rótulo "cachorro", isso significa que o modelo está muito certo de que há um cachorro na imagem.

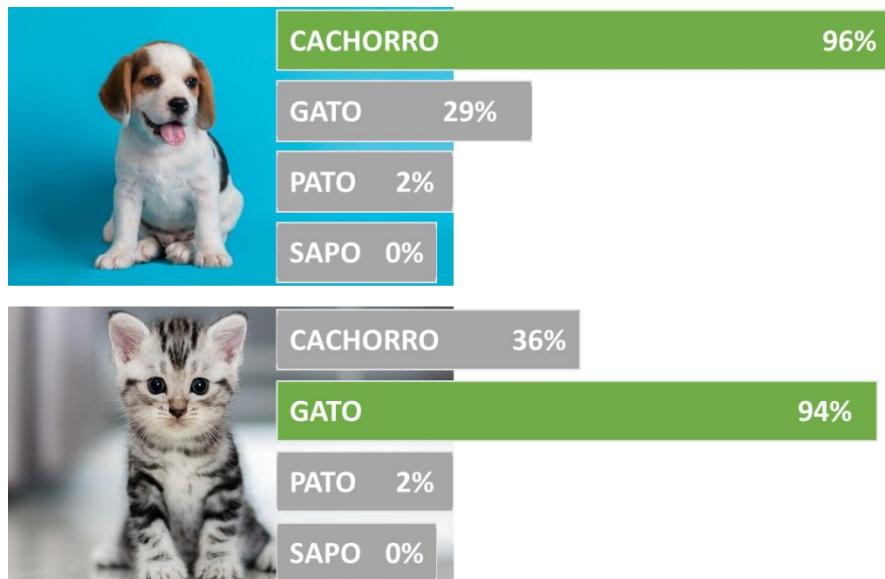


Figura 84 - Exemplo de predição de modelo de classificação. Fonte: Autoral.

Veremos este tipo de pontuação dos modelos e como gerá-las nos tópicos de implementação seguintes.

5.2 Limite de corte

Podemos simplificar essas probabilidades transformando-as em respostas "sim" ou "não" com base em um limite de corte, ou "threshold". Esse limite de corte representa o nível de confiança que o modelo precisa ter para dizer se algo pertence a uma categoria ou não.

O controle deslizante de ponto de corte na plataforma da Google é uma ferramenta visual que ajuda a testar como diferentes limites de corte afetam a classificação das categorias em um conjunto de dados. Se o limite de corte for baixo, o modelo dirá "sim" para mais itens, mas pode errar algumas vezes. Se o limite de corte for alto, o modelo será mais cauteloso e dirá "sim" para menos itens, diminuindo as chances de erro, porém, podendo não incluir resultados

positivos algumas vezes. A Figura 85 é um exemplo prático de como o limite de corte pode influenciar nos resultados.

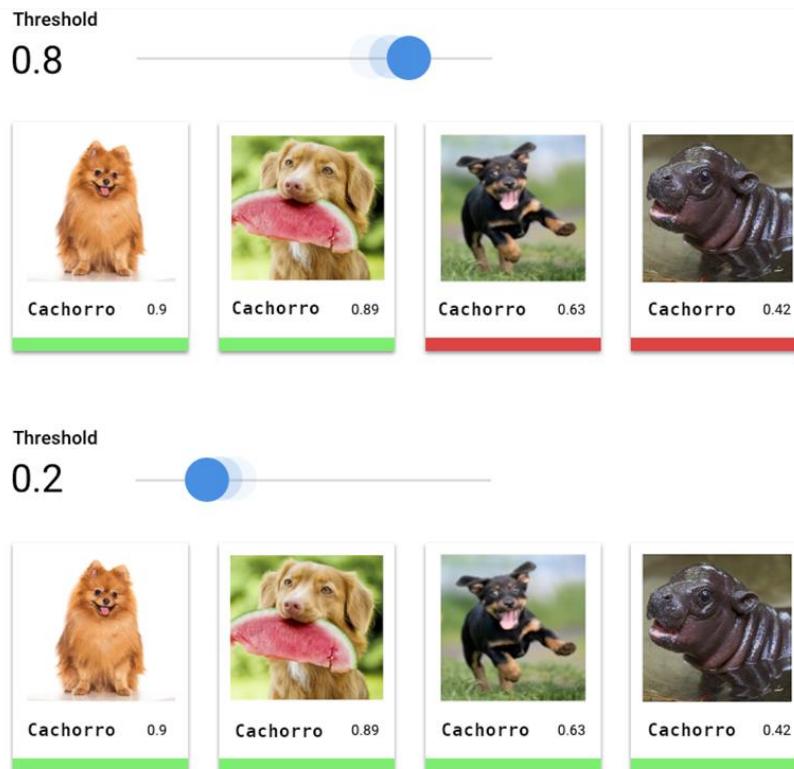


Figura 85 - Exemplos de resultados com limite de corte alterados. Fonte: adaptação da Google Cloud.

No console do Google Cloud, você pode ajustar esses pontos de corte para cada categoria e ver como isso afeta o desempenho. No entanto, quando você usar o modelo em situações reais, é importante definir pontos de corte que funcionem melhor para sua aplicação.

5.3 Matriz de confusão

A matriz de confusão é uma ferramenta fundamental na avaliação do desempenho de modelos de aprendizado de máquina, especialmente em tarefas de classificação de imagens. Ela nos permite entender como o modelo está se saindo em relação às classes que deve prever. Nesta subseção, vamos explorar o que é uma matriz de confusão, como interpretá-la e como utilizá-la no contexto de modelos de imagens.

Uma matriz de confusão é uma tabela – como a Figura 86 –, que descreve o desempenho de um modelo de classificação. Ela compara as previsões do modelo com os rótulos reais (verdadeiros) dos dados. Para um problema de classificação binária, a matriz é organizada da seguinte forma:

| | Previsão: Sim | Previsão: Não |
|----------------|---------------------|---------------------|
| Realidade: Sim | Positivo Verdadeiro | Falso Negativo |
| Realidade: Não | Falso Positivo | Negativo Verdadeiro |

Figura 86 - Matriz de confusão. Fonte: mariofilho.com

- **Verdadeiro Positivo (VP)**: casos em que o modelo **previu corretamente** a classe positiva.
- **Falso Negativo (FN)**: casos em que o modelo não conseguiu identificar a classe positiva (**previu incorretamente** a negativa).
- **Falso Positivo (FP)**: casos em que o modelo **previu incorretamente** a classe como positiva, mas o rótulo real era negativo.
- **Verdadeiro Negativo (VN)**: casos em que o modelo **previu corretamente** classe negativa.

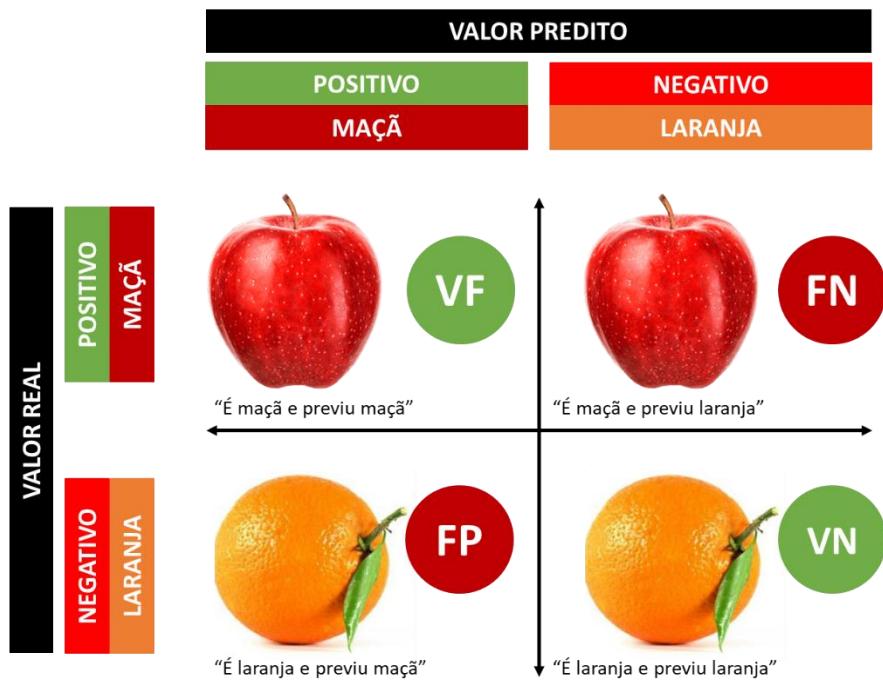


Figura 87 - Exemplo de matriz de confusão. Fonte: Autoral.

A matriz de confusão nos fornece informações valiosas sobre o desempenho do modelo, as quais veremos nas seções seguintes.

5.4 Desempenho dos modelos

Abordaremos no curso 4 métricas de desempenho que podem ser obtidas através do uso da matriz de confusão: precisão, revogação (também conhecida como sensibilidade ou “recall”), F1-score e acurácia.

Para os exemplos utilizados à seguir, iremos considerar os seguintes valores de desempenho sobre a Figura 87:

- **Verdadeiro Positivo (VP) = 6**
Foi previsto como positivo e é verdade! O modelo previu que é uma maçã e realmente é.
- **Verdadeiro Negativo (VN) = 11**
Foi previsto como negativo e é verdade! O modelo previu que não é uma maçã e realmente não é.
- **Falso Negativo (FN) = 2**

Foi previsto como negativo, mas é verdade! O modelo previu que não é uma maçã, mas é uma maçã.

- **False Positivo (FP) = 1**

Foi previsto como negativo e é positivo! O modelo previu que é uma maçã, mas na verdade não é uma maçã.

5.4.1 Acurácia

A acurácia nos mostra a porcentagem de acertos do nosso modelo, mas não mostra os erros e acertos por classe. Para dados平衡ados, é uma escolha válida, porém, deve ser evitada em caso de dados desbalanceados. Para o exemplo da Figura 87 utilizando os valores informados nesta subseção, temos 80% de acurácia.

$$ACURÁCIA = \frac{VP + VN}{VP + VN + FP + FN}$$

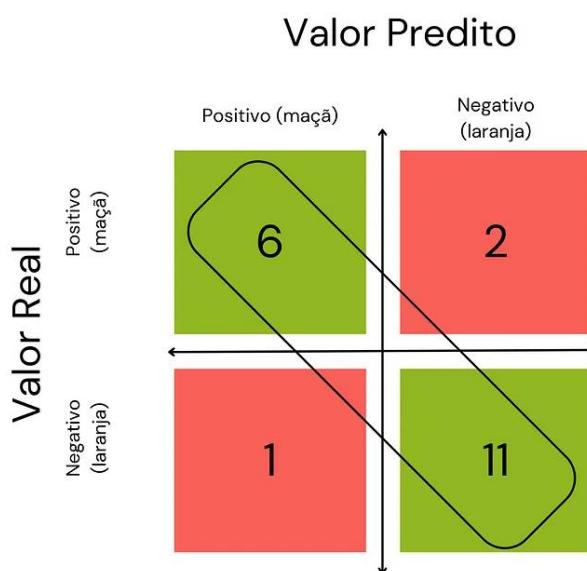


Figura 88 - Matriz de confusão para cálculo de acurácia. Fonte: Ingo Reichert Junior.

5.4.2 Precisão

A precisão mede a proporção de previsões positivas corretas em relação ao total de previsões positivas feitas pelo modelo, ou seja, avalia do total previsões positivas retornadas pelo modelo, a porcentagem de vezes em que ele

realmente estava correto. A fórmula para cálculo da precisão é a divisão da quantidade total de “verdadeiros positivos” pela soma do total de “verdadeiros positivos” somados ao total de “falsos positivos”.

$$PRECISÃO = \frac{VP}{(VP + FP)}$$

Se um modelo previu corretamente 6 imagens e 1 imagem como um falso positivo, a precisão deste modelo será de 0,857, ou seja, 86%.

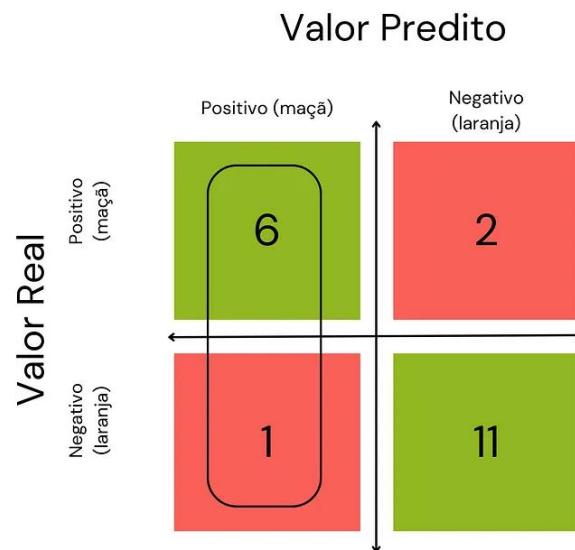


Figura 89 - Matriz de confusão para cálculo de precisão. Fonte: Ingo Reichert Junior.

É uma métrica muito válida nos casos em que os falsos positivos são uma preocupação maior do que os falsos negativos.

5.4.3 Recall, Revogação ou Sensibilidade

O “recall” mede a proporção de verdadeiros positivos em relação ao total de casos positivos reais. A fórmula para cálculo do “recall” é a divisão da quantidade total de “verdadeiros positivos” pela soma total de “verdadeiros positivos” somados ao total de “falsos negativos”.

$$RECALL = \frac{VP}{(VP + FN)}$$

Se um de um total de 8 imagens positivas, o modelo somente previu 6 imagens como positivas, a precisão deste modelo será de 0,75, ou seja, 75%.

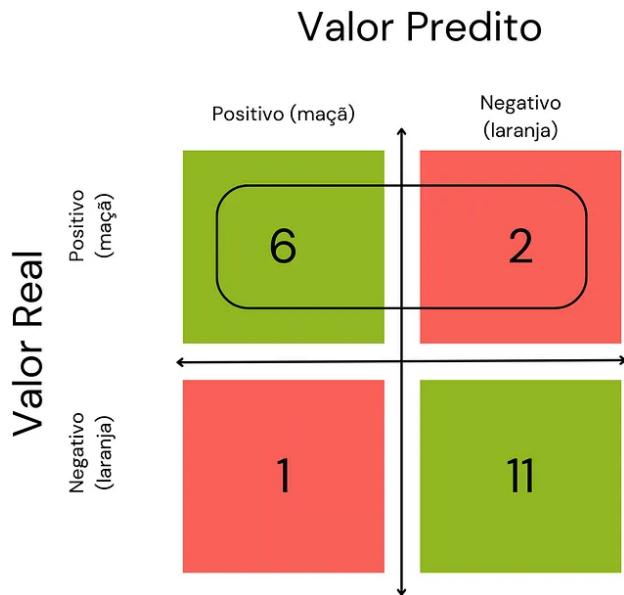


Figura 90 - Matriz de confusão para cálculo de recall. Fonte: Ingo Reichert Junior.

5.4.4 F1-Score

É a média harmônica entre precisão e revogação. Um bom F1-score indica um bom equilíbrio entre precisão e revogação. Sua fórmula é uma média harmônica das duas métricas.

$$F1\ Score = \frac{2 * (\text{precisão}) * (\text{recall})}{(\text{precisão} + \text{recall})}$$

O F1 score é usado quando tanto precisão quanto recall forem igualmente importantes para o problema que você busca resolver. No exemplo que estamos seguindo, temos:

$$F1\ Score = \frac{2 * 0,857 * 0,75}{0,857 + 0,75} = 0,80 = 80\%$$

5.4.5 Otimização entre precisão e recall

Dependendo do caso de uso, é preciso otimizar a precisão ou o recall. Veja nas imagens da Figura 91 e Figura 92 a simulação das diferenças de comportamento de um modelo quando a precisão ou o recall são priorizados.

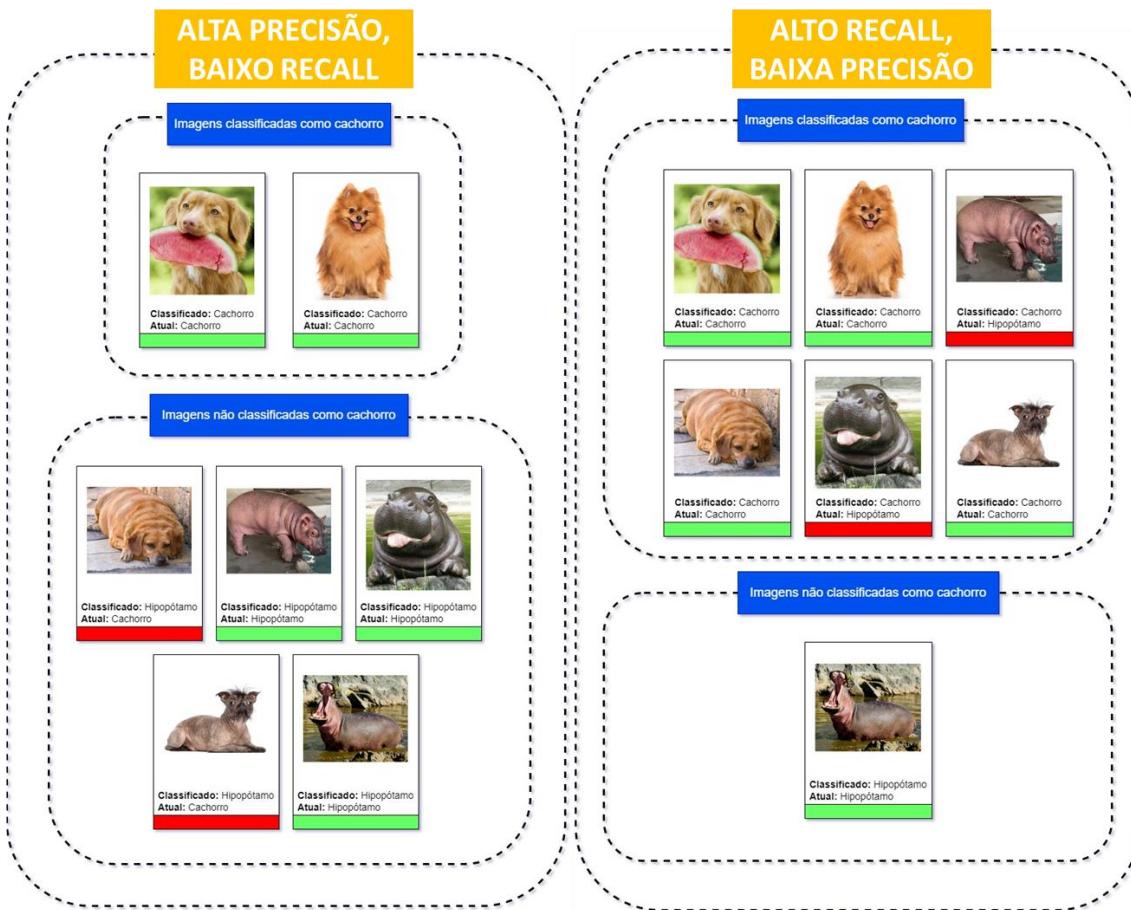


Figura 91 - Comparação entre precisão e recall (1). Fonte: Adaptação Google Cloud.

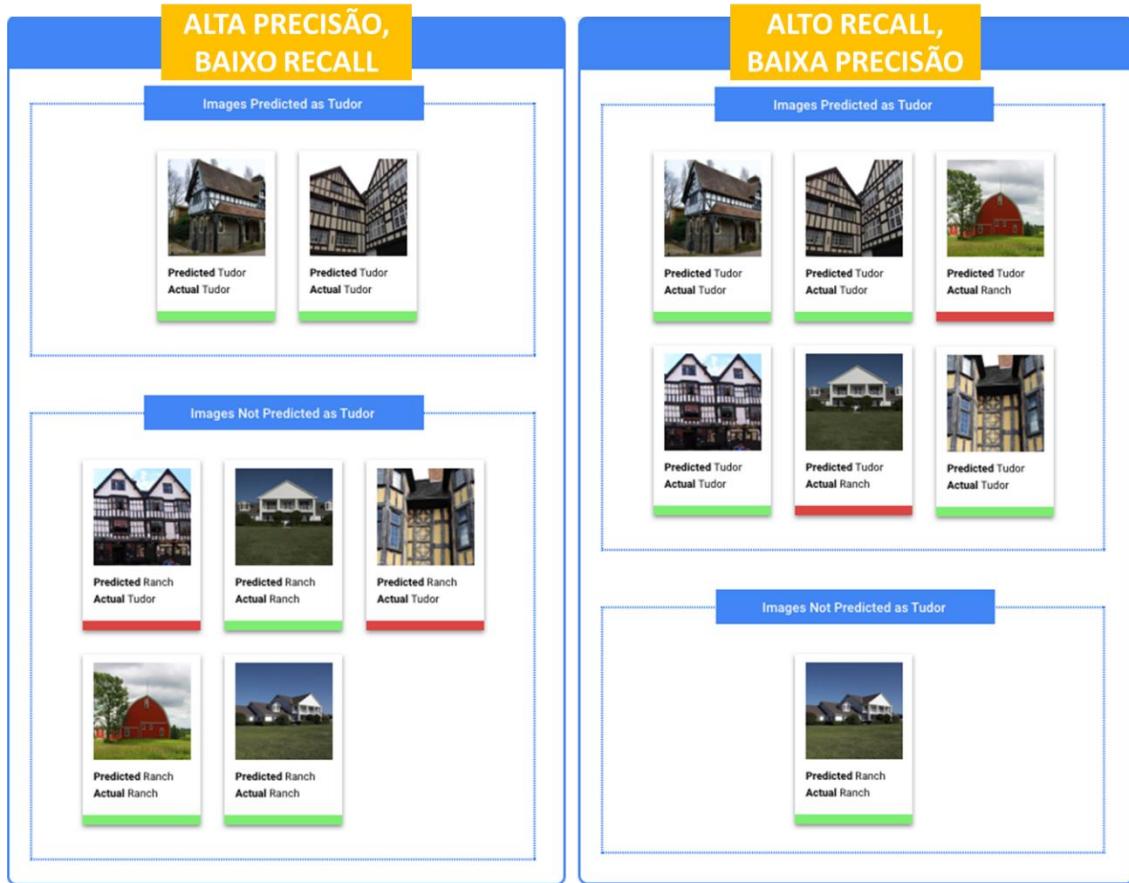


Figura 92 - Comparação entre precisão e recall (2). Fonte: Adaptação Google Cloud.

Considere os dois casos de uso a seguir ao decidir qual abordagem funciona melhor para você.

PRIVACIDADE DE IMAGENS

Imagine que você está construindo um sistema que deve encontrar e esconder automaticamente informações importantes em fotos, como números de passaporte, para proteger a privacidade das pessoas.

Agora, pense nos "falsos negativos" como elementos que deveriam ser escondidos, mas o sistema não escondeu. Por exemplo, se um número de passaporte não for escondido, isso pode ser um problema sério de segurança, pois alguém pode usar essas informações para roubo de identidade.

Nesse caso, é mais importante se concentrar no "recall". O recall nos diz quantas dessas informações importantes estão sendo deixadas de fora em relação a todas as vezes que o sistema deveria ter encontrado e escondido algo.

Um alto recall significa que o sistema está identificando a maioria dos itens importantes, mesmo que alguns não sejam tão relevantes. Isso é útil quando você tem poucos exemplos de treinamento para ensinar o sistema, e quer ter certeza de que ele não perca informações sensíveis.



Figura 93 - Documento de identificação com ocultação de informação. Fonte: Google Imagens.

PESQUISA EM BANCO DE FOTO

Imagine que você está construindo um sistema que encontra a melhor foto em um grande banco de imagens quando você faz uma busca usando uma palavra-chave.

Aqui, um "falso positivo" seria se o sistema mostrasse uma foto que não tem nada a ver com a palavra-chave que você procurou. Isso é um problema porque o sistema deveria encontrar apenas fotos relevantes.

Já um "falso negativo" seria se o sistema não mostrasse uma foto que realmente era relevante para a palavra-chave que você procurou. Isso pode acontecer quando há muitas fotos para escolher, mas o sistema deixou de mostrar uma que você realmente queria ver.

Nesse caso, é mais importante focar na "precisão". A precisão nos diz o quanto as previsões do sistema estão corretas em relação a todas as previsões feitas. Se o sistema tem alta precisão, isso significa que ele mostra principalmente fotos que realmente têm a ver com a palavra-chave que você pesquisou, o que é útil quando essa palavra-chave é comum nas fotos disponíveis.

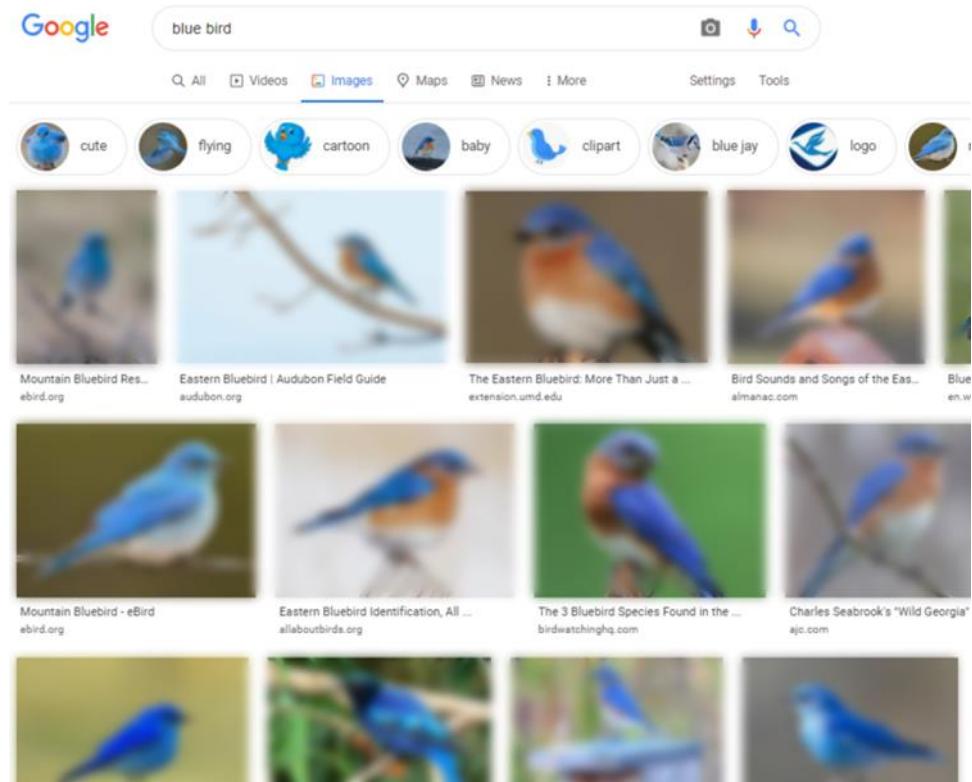


Figura 94 - Busca por pássaro azul. Fonte: Google Imagens.

5.5 Reforçando a teoria para análise de métricas de qualidade

À seguir temos algumas questões para reforçar o pleno entendimento sobre as diferenças entre os tipos de métricas de qualidade e como projetar um modelo baseado em suas representações.

1. O que é precisão?

() A proporção de verdadeiros positivos em relação ao total de previsões feitas.

() A proporção de verdadeiros positivos em relação ao total de casos positivos reais.

() A média de erros de classificação do modelo.

() A diferença entre verdadeiros positivos e falsos negativos.

2. Em qual situação é mais crítico e ideal se ter um alto recall?

() Filtragem de spam em e-mails.

() Diagnósticos de doenças graves.

- () Classificação de produtos em e-commerce.
() Análise de sentimentos em redes sociais.
3. Qual é a principal consequência de um modelo com alta precisão, mas baixo recall?
- () O modelo classifica muitos positivos corretamente, mas perde muitos casos reais positivos.
() O modelo classifica corretamente todos os casos positivos.
() O modelo gera muitos falsos positivos, mas poucos falsos negativos.
() O modelo não faz previsões erradas.
4. Em um sistema de detecção de fraudes, o que deve ser priorizado?
- () Precisão, para minimizar falsos positivos.
() Recall, para garantir que a maioria das fraudes seja detectada.
() Acurácia, para obter um desempenho geral.
() F1-Score, para equilibrar precisão e recall.
5. Qual é a fórmula correta para calcular o recall?
- () Recall = $(VP) / (VP + VP)$
() Recall = $(VP) / (VP + FP)$
() Recall = $(VP) / (VP + FN)$
() Recall = $(VP + VN) / Total$
6. Se um modelo tem uma alta taxa de falsos positivos, isso significa que:
- () O modelo está classificando muitos casos negativos como positivos.
() O modelo está classificando todos os casos corretamente.
() O modelo está perdendo muitos casos positivos.
() O modelo está classificando muitos casos positivos como negativos.
7. Em um projeto onde o custo de um falso negativo é alto, você deve priorizar:
- () Precisão.
() Recall.
() Acurácia.

() F1-Score.

8. Qual métrica você deve observar se a maioria dos falsos positivos for inaceitável em seu projeto?
- () Recall.
 () Precisão.
 () Acurácia.
 () F1-Score.

5.6 Análise de métricas de desempenho no AutoML

Para acessar a página com os detalhes das métricas, siga os passos da Figura 95.

1. Navegue até a seção “deploy and use” e então selecione “model registry”.
2. Na página “model registry” todos os modelos com treinamento finalizado podem ser vistos. Clique em cima do nome do modelo de interesse.
3. Clique no ID gerado pelo AutoML para seu modelo abaixo da seção “pointwise evaluations”.

| Name | Default version | Deployment status |
|------------------------|-----------------|-------------------|
| tx_thayna_detec_hcsr04 | 1 | - |

| Name | Status |
|-----------------------------|-----------|
| untitled_527364474336706560 | Succeeded |

Figura 95 - Acesso às métricas de qualidade de um modelo treinado. Fonte: Autoral.

A tela “evaluate” terá as seguintes informações relevantes para o curso, numeradas conforme Figura 96:

1. Nome e versão do modelo que está sendo analisado.
2. Categorias cadastradas para o modelo.
3. Limite de corte alterável pela plataforma para testes.
4. Métricas de precisão e recall, além das quantidades de imagens para cada categoria de treinamento.
5. Gráficos para análise da curva de relação entre precisão e recall.
6. Tabelas da matriz de confusão.

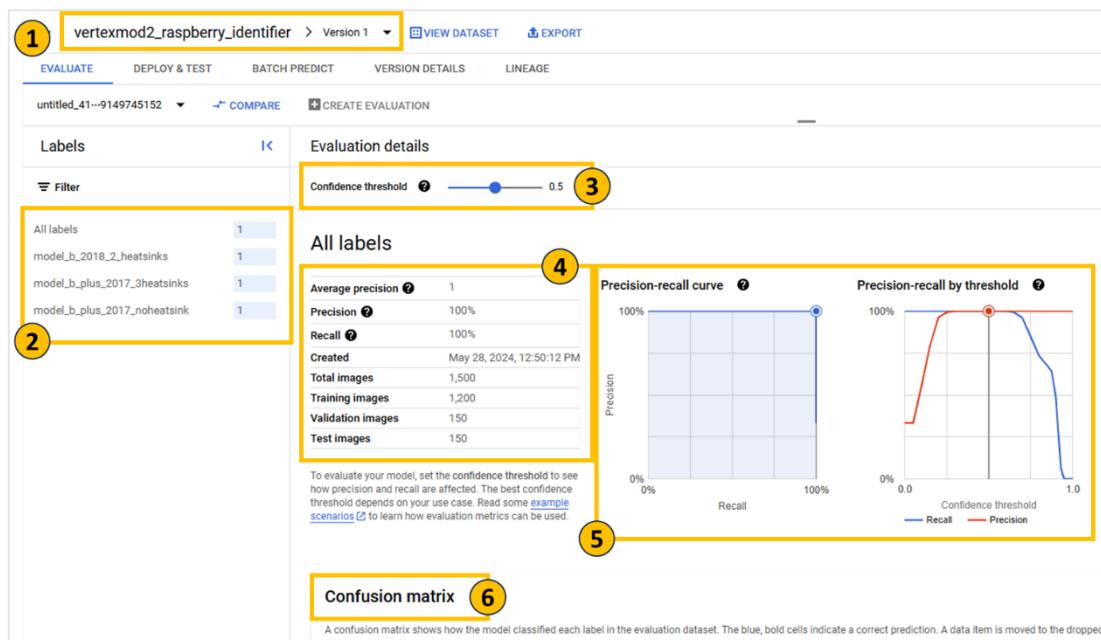


Figura 96 - Tela "evaluate". Fonte: Autoral.

PRECISION-RECALL

Ao alterarmos o nível do limite de corte através da barra “confidence threshold”, podemos verificar a mudança dos gráficos “precision-recall curve” e “precision-recall by threshold”.

- **PRECISION-RECALL CURVE:** Mostra o “trade-off” entre precisão e recall em diferentes limites de confiança. Um limite mais baixo resulta em recall mais alto, mas tipicamente em precisão mais baixa, enquanto um limite mais alto resulta em recall mais baixo, mas tipicamente com precisão mais alta.

- **PRECISION-RECALL BY THRESHOLD:** Mostra como seu modelo se sai no rótulo com pontuação mais alta ao longo de toda a faixa de valores de limite de confiança. Um limite de confiança mais alto produz menos falsos positivos, o que aumenta a precisão. Um limite de confiança mais baixo produz menos falsos negativos, o que aumenta a sensibilidade.

CONFUSION MATRIX

A matriz de confusão também pode se alterar ao modificarmos o valor do limite de corte do nosso modelo. A avaliação da matriz é feita da seguinte forma:

- Se os números na linha diagonal estão altos, significa que o programa está acertando na maioria das vezes.

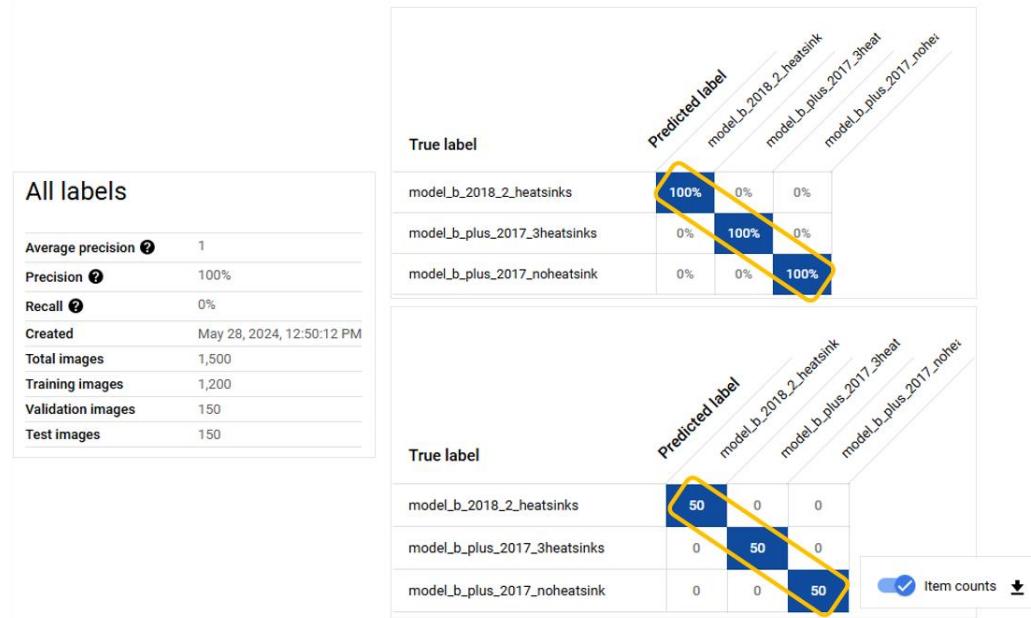


Figura 97 - Avaliação de matriz de confusão no AutoML. Fonte: Autoral.

- Se os números fora da linha diagonal são altos, isso significa que o programa está errando, ou seja, fazendo previsões incorretas. Isso pode ser um sinal de que algo não está indo bem no treinamento do programa.

| | | GOOD | | | |
|-----------|--|-----------|--------|-------|-----------|
| | | Predicted | | | |
| True | | Tudor | Modern | Ranch | Victorian |
| Tudor | | 73% | 9% | 12% | 6% |
| Modern | | — | 100% | — | — |
| Ranch | | 6% | 2% | 82% | 10% |
| Victorian | | 21% | 14% | 2% | 63% |

| | | BAD | | | |
|-----------|--|-----------|--------|-------|-----------|
| | | Predicted | | | |
| True | | Tudor | Modern | Ranch | Victorian |
| Tudor | | 12% | 9% | 72% | 6% |
| Modern | | — | 100% | — | — |
| Ranch | | 82% | 2% | 6% | 10% |
| Victorian | | 21% | 63% | 2% | 14% |

Figura 98 - Avaliação de matriz de confusão no AutoML. Fonte: Adaptação Google Cloud.

6 Exportação de modelos Edge

Existem dois caminhos para encontrar e exportar os modelos treinados. O primeiro deles vimos na Figura 76 – através do menu lateral, na seção “model development” e menu “training”. Nela é possível ver se o treinamento dos modelos já foi concluído, e caso sim, optar por exportá-lo. O outro caminho é através da seção “deploy and use”, na opção “model registry”, também já visto na Figura 83 – onde vemos somente os modelos que já finalizaram a etapa de treinamento.

Ao optarmos pelo menu “model registry” e selecionarmos o modelo de interesse, veremos novamente a tela “evaluate”, já explicada com a Figura 96 na seção anterior.

As opções de exportação do modelo estão na aba “deploy & test”, detalhada na Figura 99:

1. **Nome e versão** do modelo selecionado.
2. Seção de opções para **deploy e testes**.
3. Opções de formato de **exportação** do modelo (varia de acordo com o tipo de modelo). Veremos cada um dos formatos em mais detalhes à seguir.
4. **Use your edge-optimized model**: exibe as opções de formato para exportação de modelo treinado para uso em borda de forma otimizadas.
5. Opção de **deploy** do modelo na **Cloud**, através de um “endpoint” provido pela plataforma. “Endpoints” são modelos de machine learning disponibilizados para solicitações de previsão on-line. São úteis para previsões oportunas de muitos usuários (por exemplo, em resposta a uma solicitação de aplicativo). Você também pode solicitar previsões em lote se não precisar de resultados imediatos.
6. **Teste** o seu modelo, desde que seu deploy tenha sido feito via Cloud.

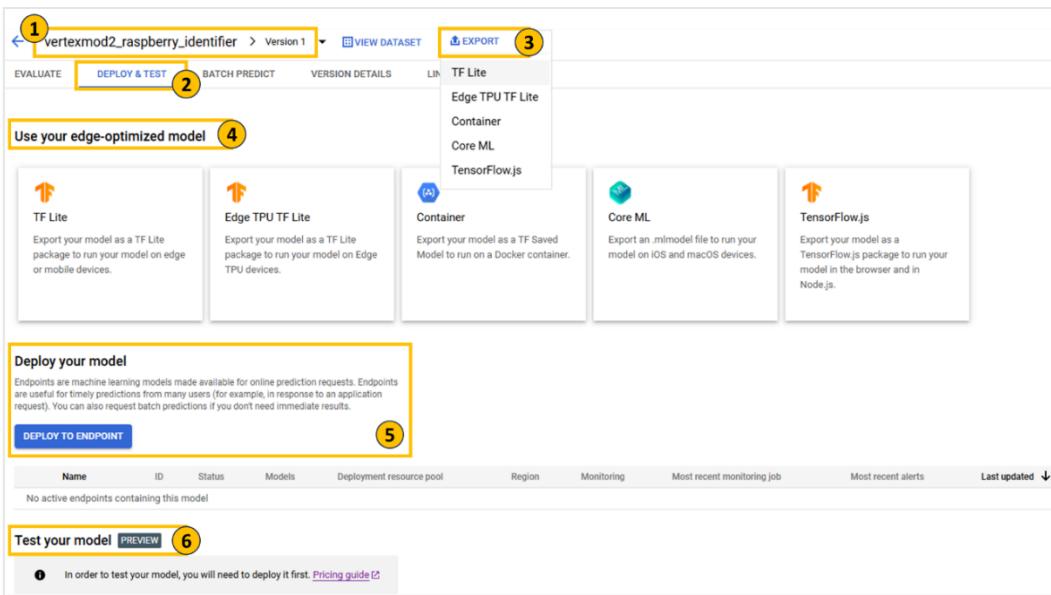


Figura 99 - Tela "evaluate", seção "deploy & test". Fonte: Autoral.

Os formatos disponíveis para exportação destacados nos pontos 3 e 4 são:

- **TF Lite:** É uma versão otimizada do TensorFlow para dispositivos móveis e de borda (edge). Exportar seu modelo como um pacote TF Lite permite que você execute o modelo em dispositivos com recursos limitados, como smartphones e dispositivos IoT. Essa opção é ideal para aplicações que necessitam de inferência rápida e eficiente em tempo real.
- **Edge TPU TF Lite:** esta opção permite exportar o modelo como um pacote TF Lite específico para dispositivos Edge TPU. O Edge TPU é um acelerador de hardware projetado para executar modelos de aprendizado de máquina de forma rápida e eficiente. A exportação do modelo para essa plataforma permite que você aproveite a potência do Edge TPU para obter desempenho superior em inferências.
- **Container:** exportar seu modelo como um TF Saved Model permite que você o execute em um contêiner Docker. Essa abordagem é útil para implantação em ambientes de produção, onde você pode precisar de isolamento e escalabilidade. O uso de contêineres facilita a distribuição e execução do modelo em diferentes plataformas, mantendo a consistência do ambiente.

- **Core ML:** ao exportar um modelo como um arquivo “.mlmodel”, você pode executá-lo em dispositivos iOS e macOS. O Core ML é uma estrutura da Apple que permite integrar modelos de aprendizado de máquina em aplicativos da Apple, proporcionando uma experiência otimizada para os usuários destes dispositivos. Essa opção é ideal para desenvolvedores que desejam incorporar inteligência artificial em seus aplicativos móveis.
- **TensorFlow.js:** esta opção permite exportar seu modelo como um pacote TensorFlow.js, que pode ser executado diretamente no navegador ou em um ambiente Node.js. Isso possibilita que você crie aplicações web interativas que utilizam aprendizado de máquina em tempo real, sem a necessidade de enviar dados para um servidor para inferência.

Note que, todos os 5 formatos explicados estão disponíveis para modelos de Classificação de Imagens “single” e “multi-label”, entretanto, para modelos de Detecção de Objetos somente os formatos “TF Lite”, “Container” e “TensorFlow.js” são disponibilizados.

Ao clicar em uma das opções, a tela da Figura 100 surgirá e então basta selecionar a pasta de destino do modelo exportado no Cloud Storage para armazenamento e clicar no botão “export”. Atente-se em manter a organização das pastas e nomeá-las de forma intuitiva pois o modelo é exportado usando códigos de identificação padronizados pelo AutoML e, exportá-los fora de pastas nomeadas pode tornar complicada a tarefa de encontrá-lo para posterior download local.

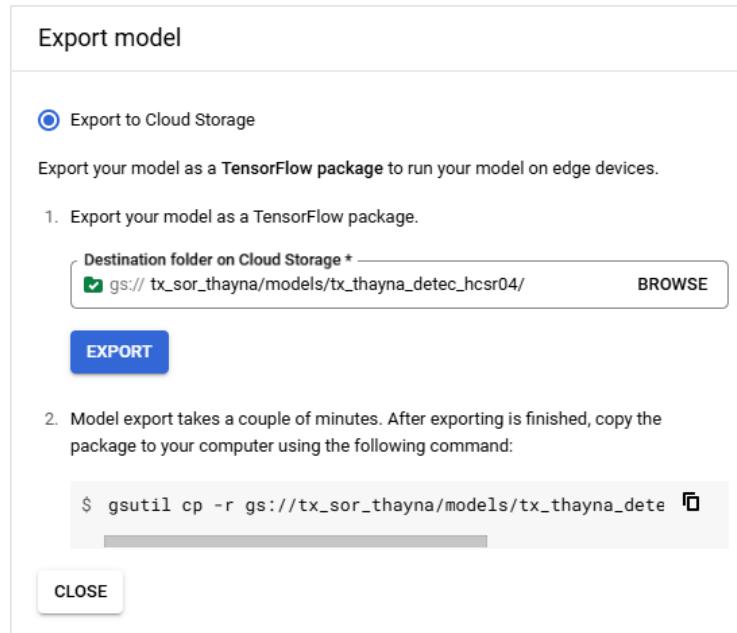


Figura 100 - Export model. Fonte: Autoral.

Após exportado, o modelo pode ser encontrado no diretório de armazenamento indicado. O AutoML cria automaticamente 3 pastas e, dentro do último nível, pode ser visto o arquivo “saved_model.pb” (nome padrão utilizado). Selecione o modelo e clique em download, conforme detalhe 5 da Figura 101.

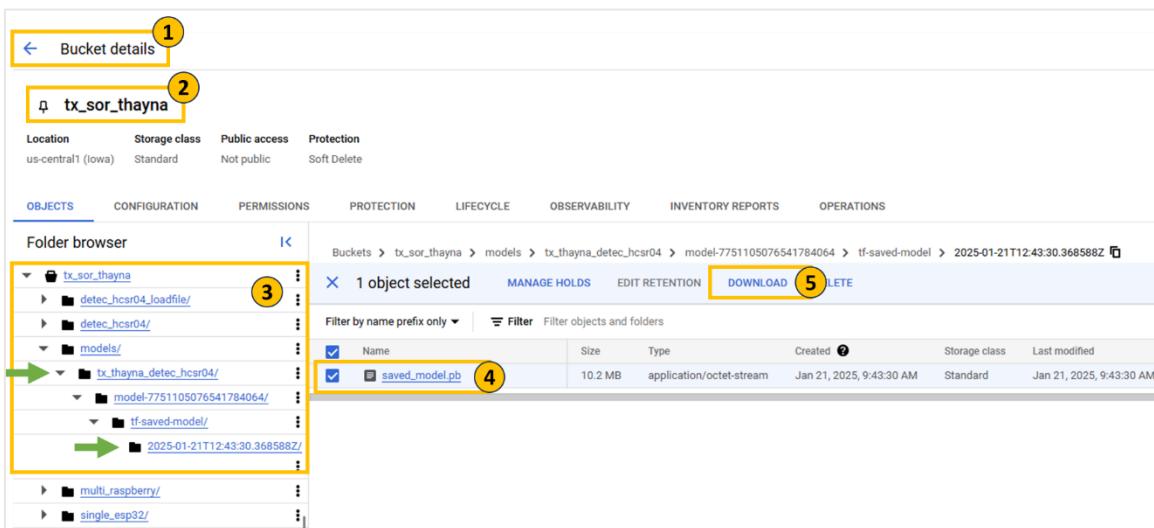


Figura 101 - Download do modelo exportado. Fonte: Autoral.

Lembre-se de renomear seu modelo localmente para facilitar o manuseio e implementação.

7 Manipulação do ambiente de implementação

Para maior compatibilidade e assimilação com os ambientes de produção que frequentamos nos sites da Flex, foi replicada uma bancada utilizada em produção e construída pelo Moonshine para desenvolvimento e execução da parte prática do conteúdo deste curso. Além da bancada, alguns periféricos utilizados para maior comodidade do operador e melhor performance das aplicações foram incluídos e serão descritos na sequência.

7.1 Bancada e periféricos

Abaixo estão descritos alguns itens utilizados para composição da bancada.

- **Bancada:** a bancada é uma réplica de um modelo encontrado em linhas de produção do prédio 09 da Flex Sorocaba. Conta com uma mesa na altura de 1 metro, com suporte para scanner leitor. Abaixo da mesa existe uma gaveta para apoio do mouse e teclado, liberando a mesa do uso de periféricos. Aproximadamente à altura média de 1,40 metros está o monitor para visualização das aplicações e inferências. O braço articulado está instalado em um suporte lateral, paralelo ao monitor e a luminária de led se encontra aproximadamente à 1,70 metro, centralizada à mesa.

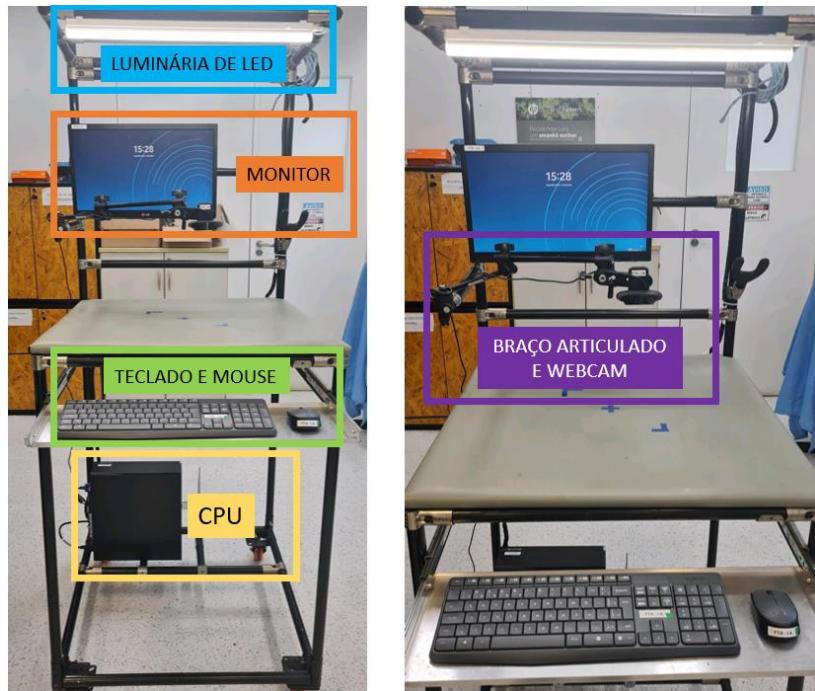


Figura 102 - Bancada instalada em Sorocaba. Fonte: autoral.

- **Luminária de led:** foi utilizado o modelo de luminária led Ledvance com luz branca, com 8w e 6500k. Este modelo dispõe de um botão on/off e pode ser acoplado nos tubos de infraestrutura da bancada, sendo alimentado por energia elétrica. A luminária é um item que visa manter a luminosidade padrão sobre a bancada em qualquer ambiente, evitando alterações de performance do modelo devido à modificações de luz/brilho com relação à quando o modelo foi treinado.



Figura 103 - Luminária de led. Fonte: autoral.

- **Suporte articulável:** foi utilizado o modelo de braço articulado Manfrotto 196B-2 143BKT. Este braço possui cada seção fixada separadamente e com comprimento de 60cm, além de possuir um suporte de câmera removível (143BKT). O suporte é utilizado para suporte da câmera e pode

ser acoplado em diferentes regiões da bancada, além de ajustado para diferentes posições e ângulos, a depender da aplicação de utilização.



Figura 104 - Suporte articulável desmontado. Fonte: autoral.

- **Câmera:** foi utilizado como padrão o modelo C920s de webcam full HD da Logitech. A resolução máxima de captura deste modelo é 1080p e sua conexão é realizada via cabo USB. A webcam foi definida como padrão por se tratar de uma boa opção em relação ao custo-benefício, mas outros modelos de câmera também podem ser acoplados ao suporte articulado para uso na bancada.



Figura 105 - Webcam Logitech C920s. Fonte: autoral.

- **Kit:** teclado e mouse sem fio: foi utilizado o modelo Logitech MK235 com conexão USB de receptor único para ambos os itens. Foi optado por um combo sem fio para facilitar o manuseio pelo operador além de deixar o ambiente de manejo da bancada mais clean, com menos fios roteados.

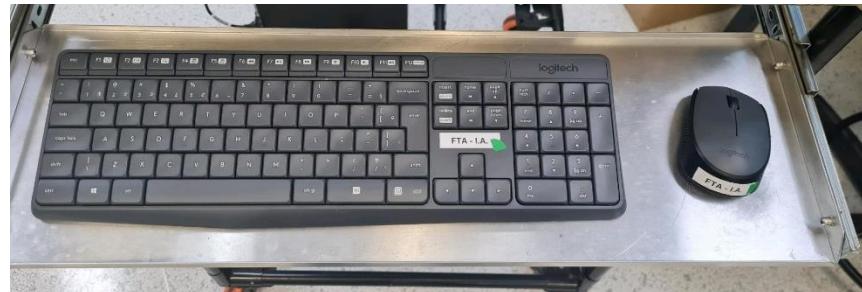


Figura 106 - Kit com teclado e mouse sem fio. Fonte: autoral.

- **Monitor:** foi utilizado um monitor de 20" para instalação à altura de visão do operador.



Figura 107 - Monitor. Fonte: autoral.

- **CPU:** foi utilizado uma máquina formatada com imagem Flex para instalação do executável da aplicação criada.



Figura 108 - CPU com imagem Flex. Fonte: autoral.

Lembre-se que, ao utilizar a bancada para construção de seu dataset, é preciso se atentar para que no momento de uso do modelo gerado por ele, todas as variáveis possíveis estejam de acordo com o cenário da criação do dataset, evitando assim resultados afetados pela alteração do ambiente.

8 Construção de aplicações próprias em bancada

Agora que já foram repassados os conceitos de treinamentos com dados de imagens para modelos de classificação de rótulo único e multirrótulos, além de modelos de detecção de objeto, e métodos e boas práticas para criação de conjuntos de dadas, vamos para o início das atividades práticas deste curso.

Ao finalizar todas as questões a seguir, realizar a entrega de arquivos relacionados à atividade prática executada em aula em arquivo .zip na plataforma de educação.

1. Construção e categorização de dataset para modelo de Classificação Single-Label.

- a. Crie um conjunto de dados para uma aplicação onde o uso de um modelo de classificação single-label agregará valor ao processo.
 - b. Categorize todas as imagens de acordo com suas categorias.
- Requisitos:
 - Duas categorias para identificação das imagens.
 - Mínimo de 50 imagens* por categoria; ou
 - Mínimo de 25 imagens por posicionamento distinto do objeto.
 - Resposta dissertativa contendo a problemática cujo modelo tem objetivo de resolver, e como a implementação deste tende a resolvê-la.
 - Diretório no Cloud Storage organizado.

*a quantidade ideal de imagens está correlacionada com a representatividade atrelada ao seu processo, conforme já estudado.

- Entrega: a entrega deverá ser realizada através de um arquivo compactado contendo os dados abaixo (com os nomes dos arquivos devidamente renomeados) na plataforma do fiteducação (edX):
 - Todas as imagens geradas para o conjunto de dados.
 - Um print da organização das pastas no Cloud Storage.



- Um print da tela de categorização, exibindo a quantidade total de imagens e as quantidades totais de imagens classificadas em cada categoria.
- Um arquivo .txt contendo a resposta dissertativa solicitada.

2. Construção e categorização de dataset para modelo de Classificação Multi-Label.

- a. Crie um conjunto de dados para uma aplicação onde o uso de um modelo de classificação multi-label agregará valor ao processo.
 - b. Categorize todas as imagens de acordo com suas categorias.
-
- Requisitos:
 - Três ou mais categorias para identificação das imagens.
 - Mínimo de 60 imagens* por categoria; ou
 - Mínimo de 30 imagens por posicionamento distinto do objeto.
 - Resposta dissertativa contendo a problemática cujo modelo tem objetivo de resolver, e como a implementação deste tende a resolvê-la.
 - Diretório no Cloud Storage organizado.

*a quantidade ideal de imagens está correlacionada com a representatividade atrelada ao seu processo, conforme já estudado.

- Entrega: a entrega deverá ser realizada através de um arquivo compactado contendo os dados abaixo (com os nomes dos arquivos devidamente renomeados) no edX:
 - Todas as imagens geradas para o conjunto de dados.
 - Um print da organização das pastas no Cloud Storage.
 - Um print da tela de categorização, exibindo a quantidade total de imagens e as quantidades totais de imagens classificadas em cada categoria.
 - Um arquivo .txt contendo a resposta dissertativa solicitada.

3. Construção e categorização de dataset para modelo de Detecção de Objetos utilizando o script detalhado na seção 4.5.

- a. Crie um conjunto de dados para uma aplicação onde o uso de um modelo de detecção de objetos agrega valor ao processo.
 - b. Importe o dataset para o AutoML utilizando um arquivo de carga.
- Requisitos:
 - Mínimo de duas categorias para detecção nas imagens.
 - Mínimo de 60 imagens* por categoria; ou
 - Mínimo de 30 imagens por posicionamento distinto do objeto.
 - Resposta dissertativa contendo a problemática cujo modelo tem objetivo de resolver, e como a implementação deste tende a resolvê-la.
 - Diretório no Cloud Storage organizado.

*a quantidade ideal de imagens está correlacionada com a representatividade atrelada ao seu processo, conforme já estudado.

- Entrega: a entrega deverá ser realizada através de um arquivo compactado contendo os dados abaixo (com os nomes dos arquivos devidamente renomeados) no edX:
 - Todas as imagens geradas para o conjunto de dados.
 - Um print da organização das pastas no Cloud Storage.
 - Um print da tela de categorização, exibindo a quantidade total de imagens e as quantidades totais de imagens classificadas em cada categoria.
 - Um arquivo .txt contendo a resposta dissertativa solicitada.
 - O arquivo .csv gerado pelo script solicitado.



8.1 Treinamento dos modelos

Após realizar a criação e a categorização de todos os datasets das atividades anteriores, além do envio dos arquivos compactados para seu perfil pessoal na plataforma do fiteducação (edX), **solicite que o instrutor** avalie e aprove a sequência da atividade, onde o dataset será enviado para treinamento seguindo as instruções realizadas na seção 4.6.

9 Verificação dos modelos

Seguindo as etapas demonstradas na seção 5 e especialmente 5.6 desta apostila, navegue até seus modelos treinados e realize a avaliação de suas métricas de qualidade. Siga as instruções abaixo para gerar evidências avaliativas.

Ao finalizar todas as questões a seguir, realizar a entrega de arquivos relacionados à atividade prática executada em aula em arquivo .zip na plataforma de educação.

1. Modelo 1: Classificação Single-Label

- a. Tire um print da tela “evaluate”, de forma que as informações de pontuação das labels e as descrições e gráficos de “evaluation details” apareçam.
- b. Tire um print similar, desta vez alterando o valor do “confidence threshold” de forma que o “recall” seja priorizado.
- c. Tire um print da “Confusion matrix”.
- d. Escolha uma das categorias e, tire um print das informações exibidas para os parâmetros de “verdadeiro negativo”, “verdadeiro positivo”, “falso negativo” e “falso positivo”.

Entrega: a entrega deverá ser realizada através de um arquivo compactado contendo todos os dados solicitados acima com os nomes dos arquivos devidamente renomeados.

2. Modelo 2: Classificação Multi-Label

- a. Tire um print da tela “evaluate”, de forma que as informações de pontuação das labels e as descrições e gráficos de “evaluation details” apareçam.
- b. Tire um print similar, desta vez alterando o valor do “confidence threshold” de forma que a “precisão” seja priorizada.
- c. Tire um print da “Confusion matrix”.

- d. Escolha uma das categorias e, tire um print das informações exibidas para os parâmetros de “verdadeiro negativo”, “verdadeiro positivo”, “falso negativo” e “falso positivo”.

Entrega: a entrega deverá ser realizada através de um arquivo compactado contendo todos os dados solicitados acima com os nomes dos arquivos devidamente renomeados.

3. Modelo 3: Detecção de Objetos

- a. Tire um print da tela “evaluate”, de forma que as informações de pontuação das labels e as descrições e gráficos de “evaluation details” apareçam.
- b. Tire um print similar, desta vez alterando o valor do “confidence threshold” de forma que a “precisão” seja priorizada.
- c. Escolha uma das categorias e, tire um print das informações exibidas para os parâmetros de “verdadeiro negativo”, “verdadeiro positivo”, “falso negativo” e “falso positivo”.

Entrega: a entrega deverá ser realizada através de um arquivo compactado contendo todos os dados solicitados acima com os nomes dos arquivos devidamente renomeados.

9.1 Exportação dos modelos

Seguindo as etapas demonstradas na seção 6 desta apostila, navegue até seus modelos treinados e realize a exportação em formato TF Lite dos mesmos. Siga as instruções abaixo para gerar evidências avaliativas.

1. Exporte os 3 modelos treinados (classificação single e multi-label e detecção de objetos) no formato TF Lite. Lembre-se de organizar o diretório de armazenamento na Cloud Storage e renomear seu modelo assim que realizar seu download para a máquina local.



Entrega: a entrega deverá ser realizada através de um arquivo compactado contendo todos os 3 modelos devidamente renomeados e 1 print da organização final dos diretórios da Cloud Storage (todos expandidos) na plataforma do fiteducação (edX).

10 Familiarização com os templates de software

Para fins de simplificação do processo de deploy de um modelo treinado pelo AutoML e exportado com o formato TF Lite, foram criados templates para utilização do aluno. Estes templates fazem uso da biblioteca Gradio para construção da interface gráfica e da biblioteca TensorFlow Lite para execução eficiente de modelos de aprendizado de máquina em dispositivos móveis e dispositivos de borda.

10.1 Interface gráfica com Gradio

Gradio é uma biblioteca python que simplifica a criação e o compartilhamento de interfaces de usuário para modelos de Machine Learning. Com o Gradio, é possível criar interfaces interativas com poucas linhas de código, permitindo que os usuários interajam com os modelos através de uma interface simples e intuitiva. A biblioteca oferece suporte para uma variedade de tipos de entrada, como texto, imagem e áudio, e permite a personalização da aparência e do comportamento da interface. Utilizaremos o Gradio para possibilitar a visualização das inferências dos modelos desse curso.

10.2 Utilização dos templates com modelos treinados

Agora que já temos nossos modelos treinados e suas métricas avaliadas, podemos iniciar a avaliação gráfica dos modelos. Para isso, utilizaremos os arquivos contidos na pasta (3) “models_validation”, presente no repositório clonado no início do curso. A Figura 109 detalha as pastas.

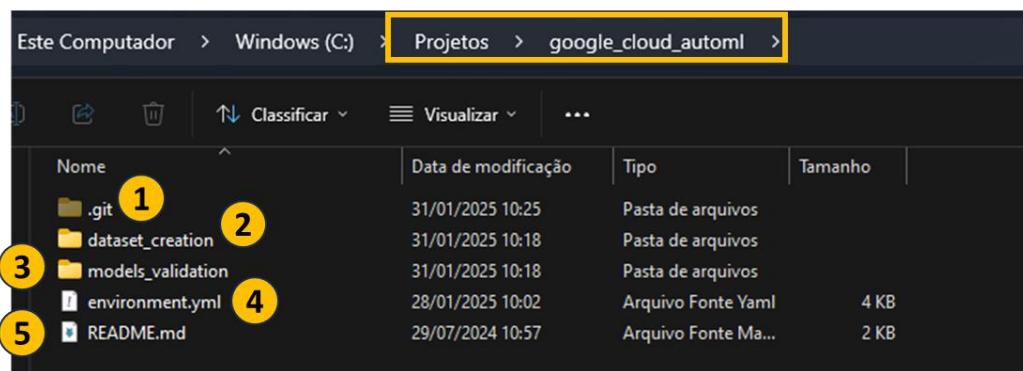


Figura 109 - Detalhes do repositório clonado. Fonte: autoral.

1. **Arquivo “.git”**: é um diretório oculto automaticamente criado na raiz do projeto e que contém todos os dados necessários para o controle de versão dele.
2. **Pasta “dataset_creation”**: contém os scripts para auxiliar na criação do conjunto de dados e arquivo de carga para modelos de detecção de objetos.
3. **Pasta “models_validation”**: contém os notebooks e scripts para criação da interface gráfica e validação dos modelos treinados.
4. **Arquivo “environment.yml”**: arquivo utilizado para criação do ambiente virtual com todas as bibliotecas e dependências necessárias para execução dos templates providos.
5. **Arquivo “README.md”**: contém explicações sobre os diretórios e arquivos do repositório.

Ao navegarmos para a pasta “models_validation”, veremos 3 pastas principais. Abaixo está o detalhamento, conforme a Figura 110.

1. **data**: pasta onde os arquivos de teste e modelos deverão ser armazenados.
 - a. **imgs**: pasta designada para armazenagem de imagens para teste.
 - b. **models**: pasta designada para armazenagem de modelos treinados.
 - i. **detec**: subpasta para modelos de detecção de objetos.
 - ii. **multi**: subpasta para modelos de classificação multi-label.
 - iii. **single**: subpasta para modelos de classificação single-label.
2. **notebooks**: pasta onde os arquivos de execução utilizando o jupyter notebooks estão armazenados.
3. **scripts**: pasta onde os scripts de execução única dos modelos estão armazenados.

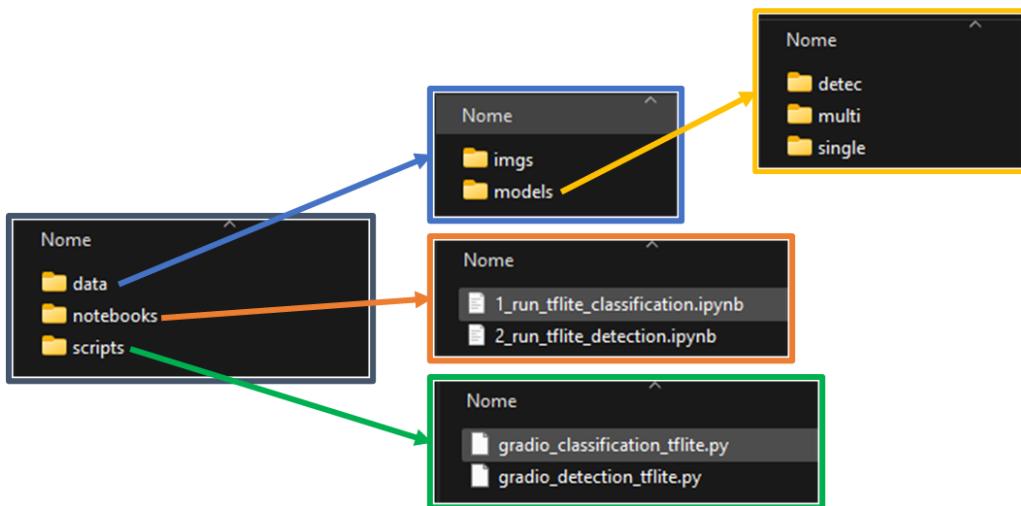


Figura 110 - Organização da pasta "models_validation". Fonte: autoral.

O professor dará instruções no decorrer das aulas sobre como fazer o correto uso e execução dos scripts disponibilizados.

10.3 Ensaios e pequenas alterações

Para fins didáticos, a fim de averiguar a influência do ambiente de uso da aplicação finalizada em comparação com o ambiente onde os dados de treinamento foram coletados, podemos realizar pequenas alterações no setup e verificar a influência das mesmas nos resultados das inferências previamente realizadas. Exemplos de alterações são: modificar a luminosidade no local de instalação da bancada (aumentar ou diminuir o brilho/luz que se projeta sobre o local onde o objeto é visto pela câmera); mudar o fundo onde o objeto é repousado (se as imagens foram captadas em fundo branco, colocar o objeto sobre um fundo preto); alterar o ângulo de visão da câmera com relação ao objeto (se as imagens foram captadas em 90°, alterar para outro ângulo); alterar a posição do objeto abaixo da câmera (se o objeto foi captado posicionado na horizontal, altere para vertical).



11 Atividades

No decorrer do curso foram realizadas atividades de reforço ao aprendizado teórico e prático desenvolvidos. Como última avaliação, analise o que se pede abaixo:

1. Dê uma sugestão de implementação de um modelo treinado pelo AutoML considerando o ambiente fabril da Flex.
 - a. Escolha um dos 3 modelos estudados.
 - b. Descreva o problema e/ou processo existente.
 - c. Descreva como o uso do modelo escolhido poderia solucionar este problema e/ou melhorar a eficiência deste processo.
 - d. Descreva resumidamente os passos necessários para implementação deste modelo no problema/processo escolhido.

Realizar a entrega de arquivos relacionados à atividade prática executada em aula em arquivo .zip na plataforma de educação.



Conclusão

A abordagem desta apostila permite ao leitor uma interação ativa com as ferramentas disponibilizadas por plataformas de computação em nuvem, em específico voltadas para inteligência artificial e aprendizado de máquina, utilizando serviços *codeless*. Ao ensinar como preparar os dados, treinar os modelos e aplicá-los de forma a verificar sua eficácia, espera-se garantir ao aluno autonomia, estímulo, senso crítico e contribuir para uma aprendizagem mais efetiva, onde ele possa aplicar o que aprendeu em aula em outras áreas.

Nesta apostila, foi possível conhecer e entender um pouco mais sobre a ferramenta AutoML, disponibilizada pela Google Cloud através da plataforma Vertex AI. Foi demonstrado como utilizar o AutoML para treinamento de modelos de aprendizado sem requerer experiência prévia em programação ou análise de dados, sendo possível reproduzir o conteúdo explicado em aula para criação de *datasets*, treinamento de modelos, análise do resultado destes treinamentos e possíveis maneiras de implementá-lo em um sistema integrado.

Assim, o conteúdo do curso “*Criação de Modelos de ML utilizando o AutoML*” para construção de modelos *codeless*, familiariza o leitor com o mundo tecnológico, onde a automatização dos processos ganha, cada vez mais, espaço e agrada as empresas por reduzir custos e aumentar as margens de lucro. Obrigada por fazer parte do curso e ter realizado a leitura desta apostila. Espero que o interesse pela inovação tecnológica proporcionado por ferramentas disponibilizadas em nuvem apresentados neste curso esteja aguçado, para que você pratique e conheça ainda mais as maravilhas da inteligência artificial e aprendizado de máquina, e por que não, de forma *codeless*.

Referências

- GOOGLE CLOUD. *AutoML*. Disponível em: <https://cloud.google.com/automl?hl=en>. Acesso em: 10 dez. 2024.
- GOOGLE CLOUD. Google Cloud online storage products. Disponível em: <https://cloud.google.com/products/storage?hl=en>. Acesso em: 10 dez. 2024.
- MATHWORKS. *Multilabel Image Classification Using Deep Learning*. Disponível em: <https://www.mathworks.com/help/deeplearning/ug/multilabel-image-classification-using-deep-learning.html>. Acesso em: 15 dez. 2024.
- SENSIIGHT AI. *Image recognition: Choosing the right AI model for your project*. Disponível em: <https://www.sentisight.ai/image-recognition-choosing-the-right-ai-model-for-your-project/#:~:text=A%20single%2Dlabel%20classification%20model,model%20offered%20by%20SentiSight.ai>. Acesso em: 18 dez. 2024.

CONTROLE DE REVISÃO DO DOCUMENTO / DOCUMENT REVISION CONTROL

| Revisão | Descrição | Razão | Autor | Data |
|---------|--|--------------------------------|--------------------------------|------------|
| A | Elaboração e revisão inicial da apostila | Elaboração/ Revisão inicial | Thayna Almeida/ Luana Forte | 31/01/2025 |



BOM CURSO!