PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Instituto de Ciências Exatas e de Informática

# Sistema de Reconhecimento Facial com Microsoft Azure*

Cinthia M. Souza[1]
Douglas Spencer de Oliveira Lucas[2]
Isabela Borlido Barcelos[3]
Sibelius Rodrigues Simões[4]
Wenderson Júnio de Souza[5]

**Resumo**

Os algoritmos de aprendizado de máquina possuem diversas aplicações em sistemas inteligentes, como por exemplo para detecção de objetos, reconhecimento de poses, indexação automática de vídeos, dentre outros. Este trabalho consiste em uma aplicação de reconhecimento facial a partir da API de detecção de faces, disponível pelo Microsoft Azure. O sistema criado é uma proposta da disciplina de Inteligência Artificial, e utiliza a API para extrair atributos das faces, e uma rede *Multi Layer Perceptron* para classificar as faces, de modo a distinguir faces de pessoas autorizadas das não autorizadas. Para treinar o modelo, foi utilizada uma base de dados, disponibilizada na plataforma do Kaggle, denominada *Labelled Faces in the Wild*, que possui imagens previamente tratadas. Apesar do modelo não ter apresentado uma boa acurácia, seu desenvolvimento permitiu aos autores aprimorar seus conhecimentos em aprendizado de máquina, em *web services* e no uso da plataforma Azure Machine Learning Studio.

**Palavras-chave:** Inteligência Artificial. Aprendizado de Máquina. Reconhecimento facial. Microsoft Azure.

*Artigo apresentado à disciplina de Inteligência Artificial da PUC-MG.

[1]Graduanda em Engenharia de Computação pela PUC Minas, Brasil – cinthia.mikaela@sga.pucminas.br.
[2]Graduando em Engenharia de Computação pela PUC Minas, Brasil – douglas.spencer@sga.pucminas.br.
[3]Graduanda em Engenharia de Computação pela PUC Minas, Brasil – isabela.borlido@sga.pucminas.br.
[4]Graduando em Engenharia de Computação pela PUC Minas, Brasil – sibelius.simoes@sga.pucminas.br.
[5]Graduando em Engenharia de Computação pela PUC Minas, Brasil – wenderson.junio@sga.pucminas.br.

# 1 INTRODUÇÃO

A demanda pela resolução de problemas cada vez mais complexos e com alto volume de dados podem ser resolvidos, atualmente, com o uso de técnicas e algoritmos de aprendizado de máquina. São algoritmos que requerem uma alta demanda computacional mas que possibilitam que uma máquina aprenda por indução de hipótese, aproximação de uma funções e/ou experiências passadas, por meio de um processo de inferência de regras, na qual se obtém conclusões genéricas a partir de um dado conjunto de dados de exemplo. Alguns exemplos de aplicações do AM são reconhecimento de falas; predição de presença de tumores; reconhecimento de objetos, etc (FACELI et al., 2011).

Este trabalho apresenta uma aplicação de AM, que tem como objetivo o reconhecimento de faces para autorização de acesso à uma residência. A aplicação desenvolvida foi modelada como um problema de classificação multiclasse. Para o desenvolvimento da aplicação, foi utilizada a API de detecção de faces da Microsoft Azure, denominada Face API, que faz parte dos Serviços Cognitivos do Azure. Esta API é capaz de reconhecer e analisar rostos humanos em imagens, retornando uma série de atributos relacionados à face. A aplicação desenvolvida foi disponibiliza no serviço de nuvem da Azure. Além disso, foi desenvolvido um web service para controle das regras de acesso e uma aplicação que une todo o sistema.

O trabalho está dividido em cinco seções. A Seção 2 apresenta uma breve descrição da ferramenta de detecção facial da Azure. A Seção 3 apresenta a metodologia do trabalho, a Seção 4 apresenta os resultados e análises e a Seção 5 apresenta a conclusão e propostas de trabalhos futuros.

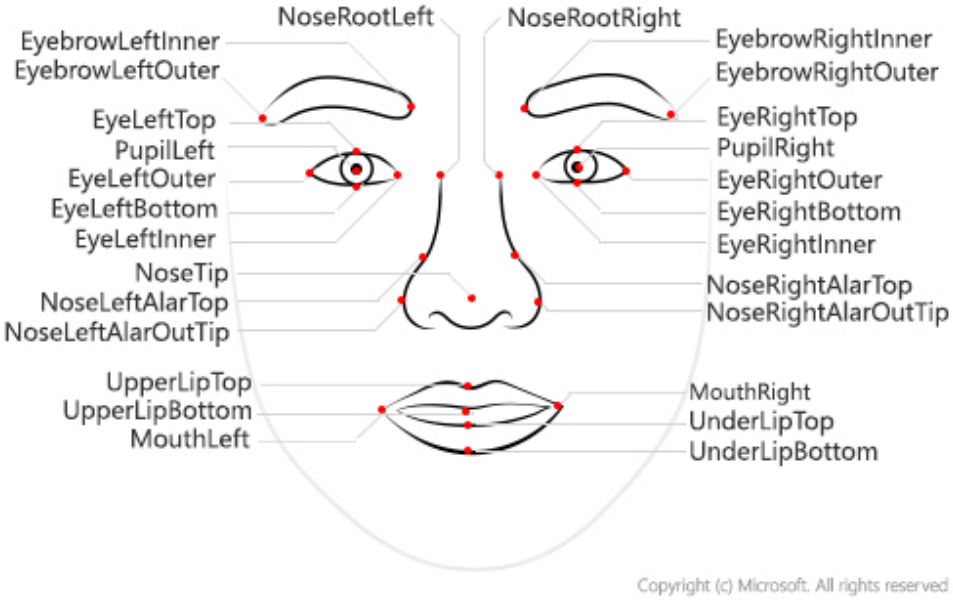
# 2 SERVIÇO DE DETECÇÃO FACIAL DA AZURE

O serviço de detecção facial da Azure é capaz de detectar um rosto em uma imagem e demarcar a posição da cabeça. Para isso, a API extrai características e atributos de uma imagem que contenha pelo menos uma face. A API ainda permite que sejam realizadas ações entre imagens como verificação facial, encontrar rostos semelhantes, agrupamento facial com base na similaridade e identificação pessoal, que compara uma imagem com uma base de imagens, à procura de uma identificação positiva (MICROSOFT AZURE, 2020b).

Dentre as características extraídas pela Face API, destacam-se 27 pontos da face (Figura 1), que consistem em pontos de referência. Cada um dos pontos de referência é recebido como uma coordenada com dois valores (posições x e y), que correspondem à coordenada em pixels daquele ponto. Neste trabalho, além das características indicadas na Figura 1, também foram utilizados os atributos gênero, idade e um *label* com o nome da pessoa na imagem. Para a criação do modelo, é necessário uma base de dados contendo faces. Também foi utilizada a base de dados *Labelled Faces in the Wild* (LFW) [6], disponível na plataforma Kaggle.

[6]Disponível em: https://www.kaggle.com/jessicali9530/lfw-dataset

**Figura 1 – Pontos de referência facial extraídos pela Face API.**




**Fonte: Microsoft Azure (2020a)**

## 3 METODOLOGIA

Nesta subseção são apresentadas as etapas metodológicas utilizadas para criação do sistema de *Home Security*. A metodologia foi dividida em cinco etapas. A primeira etapa consiste na criação da base de dados utilizada para treinamento e validação do modelo. A base de dados utilizada foi criada a partir da base de dados LFW que contém 13200 imagens, com 250x250 pixels cada. As imagens da base LFW são imagens já tratadas com *Open Source Computer Vision Library* (OpenCV), em que foram removidas as distorções nas imagens o cujas faces já estão centralizadas. Devido ao custo computacional e ao tempo para realizar as requisições de extração de características usando a API da Azure, não foi possível utilizar toda a base de imagens para a modelagem do problema. Logo, foram selecionadas as primeiras pessoas da base e algumas manualmente, totalizando 100 pessoas. Destas, foram selecionadas as 10 primeiras imagens para extração de características. A base de dados criada para os experimento, é composta por 10 imagens de 100 diferentes pessoas, totalizando 1000 imagens.

Após a definição das imagens que compõem a base de dados, foi realizada a extração das *features* das imagens. Para isso, foi utilizada a API de Detecção Facial da Azure. Nesse trabalho, foram utilizadas todas as *features* de *face landmarks* e as *features* idade e genêro de *face attributes*. Com isso, foi criada uma base de dados onde as colunas representam as *features* e as linhas representam as imagens, gerando assim, uma matriz com 1000 linhas e 56 colunas. Para treinamento do modelo, é necessário uma base de dados classificada, ou seja, com *labels*. Nesse trabalho, os *labels* são um atributo categórico nominal que representam o nome das pessoas das imagens. Após, a base de dados foi dividida em base de dados de treinamento,

com 80% dos registros, e base de dados de validação, com 20% dos registros. Essa divisão foi realizada considerando cada uma das classes, sendo assim, para cada pessoa há 8 imagens na base de dados de treinamento e 2 imagens na base de dados de validação.

Na segunda etapa, foi realizada a criação e validação do modelo de classificação. O modelo utilizado é uma rede *Multi Layer Perceptron* (MLP). A rede MLP utilizada é uma rede *fully connected*, composta por uma camada de entrada com 56 neurônios, uma camada oculta com 200 neurônios e uma camada de saída com 10 neurônios. Durante o treinamento do classificador, foi utilizada uma taxa de aprendizagem de 0.1, o diâmetro dos pesos iniciais foi definido como 0.1 e a função de normalização usada foi a gaussiana. O modelo foi treinado por 200 épocas. Os hiperparâmetros utilizados foram definidos empiricamente. Para avaliar o desempenho do modelo, foram utilizadas as métricas *Overall accuracy*, *Average accuracy*, *Micro-averaged precision*, *Micro-averaged recall* e *Macro-averaged recall*. O modelo criado foi disponibilizado utilizando o *web service* da Azure.

Na quarta etapa, foi realizada a criação de um *web service* para realizar o controle de acesso. Nesse *web service* são armazenados os nomes das pessoas que possuem autorização no sistema. O *web service* criado possui dois métodos, *get* e *post*. O método *get* retorna uma lista com o nome de todas as pessoas autorizadas, o método *post* permite adicionar pessoas na lista de autorizados, mas remove as que foram adicionadas antes. O *web service* foi hospedado na Azure e o banco de dados foi hospedado no Google Cloud.

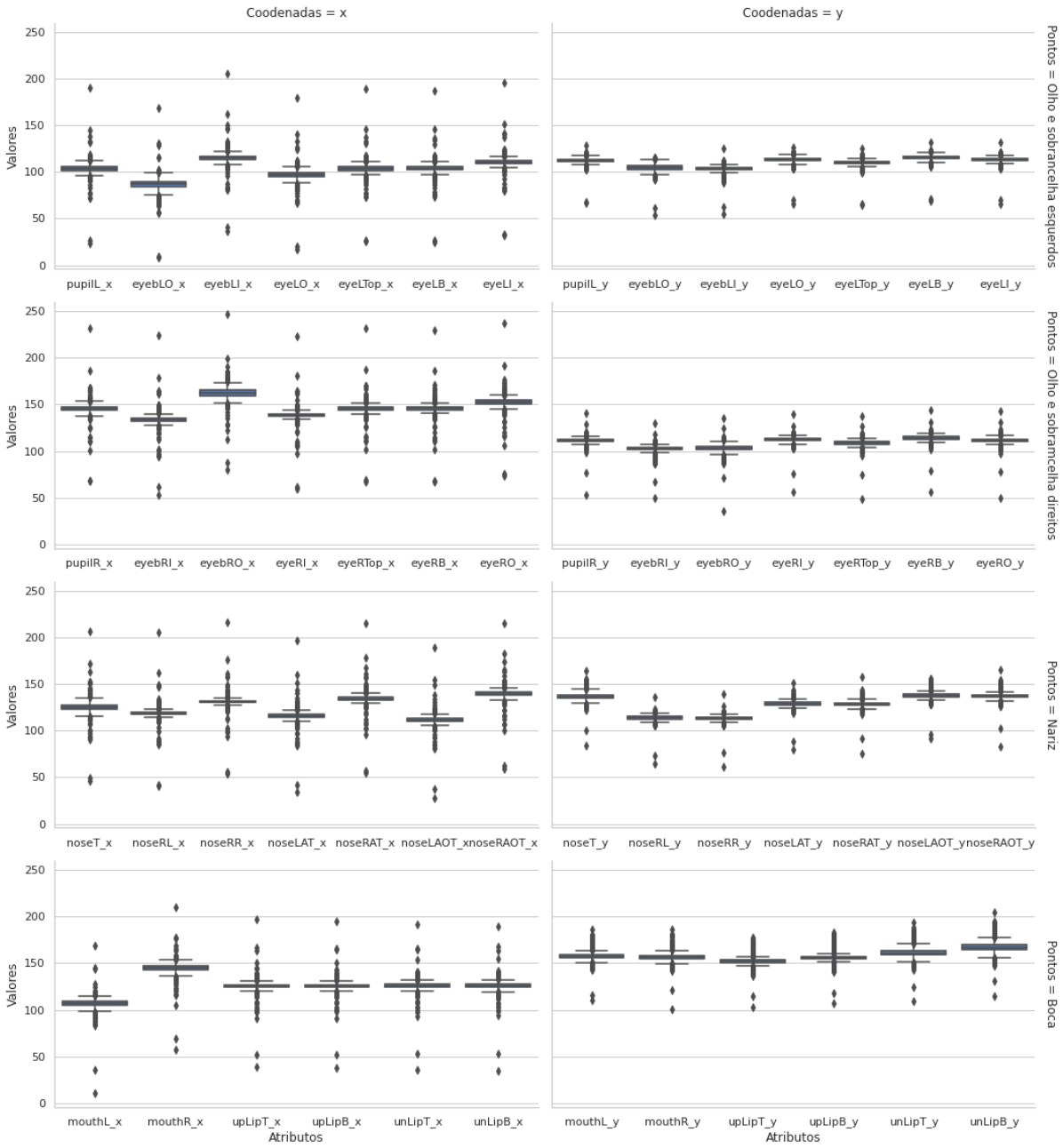
Na quinta etapa, foi criado um sistema, utilizando a linguagem Python, que une todos os processos. Nesse sistema, permite-se verificar se uma pessoa é autorizada e, também, a manutenção do *web service*. Para verificar se a pessoa é autorizada, passa-se o caminho de uma imagem. O sistema, então, extrai as *features* da imagem utilizando a API de Detecção de Faces da Azure. Após, realiza-se uma chamada utilizando a API do modelo de classificação, passando as *features* extraídas como atributo, essa API retorna os dados da classificação juntamente com a classe de maior probabilidade. Após, realiza-se uma chamada para o *web service* de controle de acesso utilizando o método get. O método *get* retorna a lista com o nome das pessoas autorizadas, os valores dessa lista são comparados com o nome retornado pelo modelo de classificação. Se o nome retornado estiver na lista de autorizado o sistema retorna uma mensagem indicado que a pessoa está autorizada a entrar, caso contrário, o sistema retorna acesso negado.

## 4 RESULTADOS E ANÁLISES

Nesta seção são apresentados os resultados obtidos com a aplicação desenvolvida, com base na metodologia descrita na Seção 3. Inicialmente, foi realizada a criação da base de dados. Nesse processo, foi gerada uma base de dados com 1000 linhas e 57 colunas, onde as colunas representam as features e as linhas os registros. Dentre as 57 colunas, uma delas é o atributo *label*, que contém o nome da pessoa associado a cada registro. Os 56 atributos restantes consistem em 27 coordenadas de pontos da face, mostradas na Figura 1, em que cada ponto resulta

em duas *features*, uma para a coordenada daquele ponto no eixo y, e outra no eixo x. Portanto, os pontos de referência facial resultantes da API da azure compõem 54 *feratures*, cujos gráficos são apresentados na Figura2. Além das 54 *features* geradas pelos pontos faciais, também foram utilizadas duas outras *features*, idade e gênero, cujos gráficos são mostrados na Figura 3.

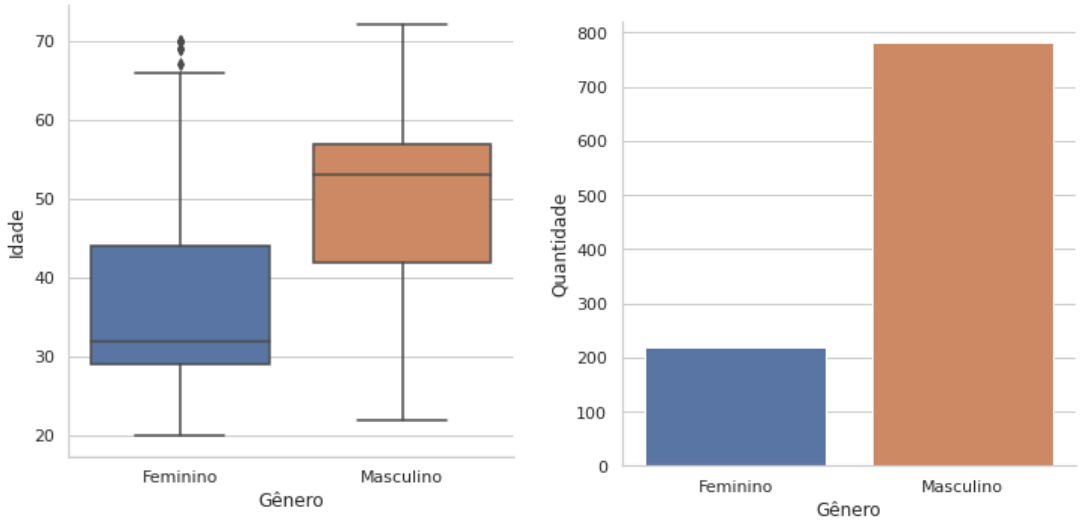
**Figura 2 – Representação boxplot dos pontos da face.**




**Fonte: Elaborada pelos autores.**

Analisando os gráficos boxplot, apresentados na Figura 2, é possível avaliar a distribuição dos dados para cada uma das *features* e identificar os *outliers* presentes nesses dados. Esses *outliers* evidenciam a presença de dados atípicos dentro do conjunto de dados. Os boxplots apresentados na Figura 2 possuem muitos *outliers*, isso mostra que há uma variação entre esses atributos para cada uma das pessoas. Essa diferença é vista como um ponto positivo, já que se todas as pessoas tivessem atributos com valores muito próximos esse atributo poderia não

**Figura 3 – Representação dos Atributos Idade e Gênero**




(a) Distribuição da idade pelo gênero. (b) Quantidade de amostras por gênero.

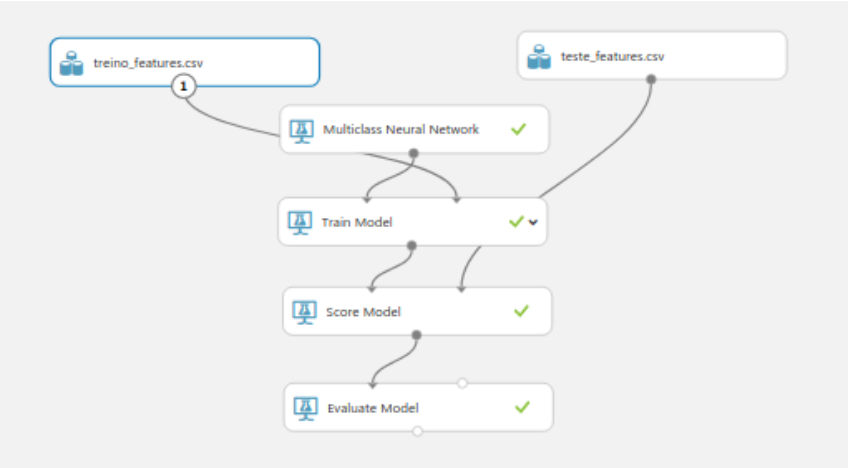
**Fonte: Elaborada pelos autores.**

agregar valor, pois não permite diferenciar essas pessoas. Uma caracteristica evidenciada no boxplot da Figura 3a é a diferença entre as idades de cada gênero. Sendo que, para o gênero feminino as idades variam entre 20 e, aproximadamente, 70 anos, com uma mediana de cerca de 32 anos e para o gênero masculino as idades variam entre, aproximadamente, 22 anos e chega a ultrapassar os 70 anos, tendo um valor de mediana em torno de 53 anos. Embora as faixas etárias estejam próximas, a distribuição desses valores é bem distinta. A partir dos gráficos, é possível ver que a maior parte dos registros do sexo feminino estão associados a pessoas mais novas, enquanto que, a maior parte dos registros do sexo masculino estão associados a pessoas mais velhas. Analisando o gráfico da Figura 3b, verifica-se que a base de dados possui uma maior quantidade de dados de imagens de pessoas do sexo masculino. Vale ressaltar, que essas informações apresentadas não são, necessariamente, assertivas, pois estas foram obtidas por meio da extração de *features* da API da Azure, isso significa que, esses valores têm um valor de acurácia associado.

Após, a base de dados criada foi divida em base de dados de treinamento e teste, sendo que a de treinamento possui 800 registros e a de teste possui 200. A Figura 4 apresenta o diagrama do modelo criado utilizando o Microsoft Azure Machine Learning Studio.

Dentre as ferramentas disponibilizadas na plataforma da Azure, tem-se as métricas de avaliação de modelos. A Tabela 1 apresenta os resultados percentuais obtidos para as métricas *Overall accuracy*, *Average accuracy*, *Micro-averaged precision*, *Micro-averaged recall* e *Macro-averaged recall*. Essas métricas são baseadas no conceito de *Accuracy*, *Precision* e *Recall*. A métrica *accuracy* representa o percentual de registros classificados corretamente. A métrica *precision* verifica quanto dos valores que foram considerados positivos são realmente positivos e a métrica *recall* representa quanto dos valores positivos foram classificados como positivos. Para realizar os cálculos, são utilizados os conceitos de *True Positive* (TP), *False Positive* (FP), *True Negative* (TN) e *False Negative* (FN). Onde TP indica o número de registros

**Figura 4 – Modelo de Classificação usando MLP.**




**Fonte: Elaborada pelos autores.**

da classe positiva classificada como positiva, FP indica o número de registros da classe negativa classificada como positiva, TN indica o número de registros da classe negativa classificada como negativa e FN indica o número de registros da classe negativa classificada como positiva. A partir desses conceitos, podemos definir que as métricas *Micro-averaged* de *recall* e *precision* representam a métrica *precision* e *recall* analisando todas as possíveis classes, a métrica *Macro-averaged recall* representa, também, a métrica *recall* porém para cada classe de modo independente. A métrica *Overall accuracy* e *Average accuracy* representam a métrica *accuracy*, sendo que *Overall accuracy* considera todas as classes e *Average accuracy* é calculada para cada classe de modo independente.

**Tabela 1 – Resultados das Métricas de Avaliação do Modelo**

| Métricas | Valor (%) |
|---|---|
| *Overall accuracy* | 44.44 |
| *Average accuracy* | 98.88 |
| *Micro-averaged precision* | 44.44 |
| *Micro-averaged recall* | 44.44 |
| *Macro-averaged recall* | 44.44 |

**Fonte: Elaborada pelos autores.**

Com base nos resultados apresentados na Tabela 1, verifica-se que o modelo utilizado não apresentou bons resultados. Dentre as hipóteses levantadas para justificar os resultados, temos: a falta de conhecimento sobre o domínio das *features*, o que não permite realizar uma melhor seleção de *features*; e que os modelos disponíveis na Azure não possuem ajustes suficientes. Dentre os modelos testados, a rede MLP foi a que obteve o melhor desempenho, no entanto, o modelo disponibilizado na ferramenta possui um limite de camadas ocultas, o que pode ter limitado a capacidade do modelo.

# 5 CONCLUSÃO

Neste trabalho foi possível aplicar e implementar técnicas de modelagem para um sistema de multivariáveis. Sendo assim, agregou conhecimentos práticos e teóricos sobre AM, preparação de dados, problemas que podem ser encontrados desde a preparação dos dados até a análise dos dados de um problema que envolve a seleção, preparação e treinamento de dados, aos autores deste trabalho. Durante o desenvolvimento do trabalho, alguns problemas foram encontrados. Inicialmente, foi definido pelo grupo a utilização dos *notebooks* disponíveis Kaggle, no entanto, foi necessário mudar para a plataforma do Google Colab, devido a melhor capacidade de compartilhamento e colaboração entre diferentes usuários. Além disso, durante a etapa de extração de *features*, ocorreram erros de requisição, quando essas ultrapassaram um limite de 400 registros e às vezes menos. Diante disso, foi necessário pausar a etapa de extração de *features*, para, no outro dia, retomá-la. Estes erros foram causados, provavelmente, por um impedimento do Colab de retornar as requisições, ainda que a requisição tenha sido realizada com sucesso nos servidores da Azure. Como proposta para trabalhos futuros, pretende-se investigar as principais *features* que distinguem as face das pessoas, aumentar a quantidade de registros referentes a cada pessoa e utilizar modelos de classificação mais robustos.

## REFERÊNCIAS

FACELI, Katti et al. **Inteligência Artificial - Uma abordagem de aprendizado de máquina**. [S.l.]: LTC, 2011.

MICROSOFT AZURE. **Detecção de face e atributos**. 2020. Disponível em: <https://docs.microsoft.com/pt-br/azure/cognitive-services/Face/concepts/face-detection>. Acesso em: 02 de mai. de 2020.

MICROSOFT AZURE. **O que é o serviço de Detecção Facial do Azure?** 2020. Disponível em: <https://docs.microsoft.com/pt-br/azure/cognitive-services/Face/>. Acesso em: 02 de mai. de 2020.