UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO DE INFORMÁTICA
ESPECIALIZAÇÃO EM CIÊNCIA DE DADOS E SUAS APLICAÇÕES

VINÍCIUS DA SILVA MORAES

APRENDIZADO INCREMENTAL COM INTERAÇÃO PARA CLASSIFICAÇÃO DE PERSONAS

MONOGRAFIA

CURITIBA
2021

VINÍCIUS DA SILVA MORAES

APRENDIZADO INCREMENTAL COM INTERAÇÃO PARA CLASSIFICAÇÃO DE PERSONAS

INTERACTIVE INCREMENTAL LEARNING FOR CLASSIFYING PERSONAS

Monografia apresentada como requisito para obtenção do título de Especialista em Ciência de Dados e Suas Aplicações da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador: Prof. Dr. Luiz Celso Gomes Jr.

CURITIBA

2021

Ministério da Educação
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
UTFPR - CAMPUS CURITIBA
DIRETORIA-GERAL - CAMPUS CURITIBA
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO - CAMPUS CURITIBA
DEPARTAMENTO DE APOIO DAS ESPECIALIZAÇÕES LATO-SENSU DOS CURSOS DE INFORMÁTICA - CAMPUS CURITIBA
CURSO DE ESPECIALIZAÇÃO EM CIÊNCIA DE DADOS E SUAS APLICAÇÕES

TERMO DE APROVAÇÃO

APRENDIZADO INCREMENTAL COM INTERAÇÃO PARA CLASSIFICAÇÃO DE PERSONAS

por

Vinicius Da Silva Moraes

Este Trabalho de Conclusão de Curso foi apresentado às 19h00 do dia 14 de julho de 2021 por videoconferência como requisito parcial à obtenção do grau de Especialista em Ciência de Dados e suas Aplicações na Universidade Tecnológica Federal do Paraná - UTFPR - Campus Curitiba. O aluno foi arguido pela Banca de Avaliação abaixo assinados. Após deliberação, a Banca de Avaliação considerou o trabalho aprovado.

_____

Prof. Dr. Luiz Celso Gomes Junior (Presidente/Orientador – DAINF-CT/ UTFPR-CT)

_____

Prof. Dr. Marcelo de Oliveira Rosa (Avaliador 1– DA-DAELT/ UTFPR-CT)

_____

Profa. Dra. Rita Cristina Galarraga Berardi (Avaliadora 2 – DAINF-CT/ UTFPR-CT)

O Termo de Aprovação assinado encontra-se no sistema SEI- Processo nº 23064.029972/2021-70

Referência: Processo nº 23064.029972/2021-70 SEI nº 2173098

RESUMO

DA SILVA MORAES, Vinícius. **Aprendizado incremental com interação para classificação de personas**. 2021. 33 f. Monografia (Especialização em Ciência de Dados e Suas Aplicações) — Universidade Tecnológica Federal do Paraná, Curitiba, 2021.

A utilização de personas para identificação de público ajuda na tomada de decisão e oferta de serviços. A empresa analisada nesta monografia realizou uma pesquisa interna para determinar suas personas dentro dos perfis de funcionários e desenvolveu um sistema de questionário para classificá-los. O objetivo dessa classificação é de melhorar a distribuição de hardware para funcionários, oferecendo opções ideais para cada perfil. Com o crescimento gradual do mapeamento desses funcionários, o sistema de regras de questionário pode se tornar obsoleto e, devido isso, o objetivo desta monografia é analisar e propor um modelo de aprendizado incremental para classificação dos funcionários da empresa, trazendo a possibilidade da substituição do sistema de questionário no futuro. A técnica utilizada para o desenvolvimento do modelo proposto foi o aprendizado incremental para fluxo de dados, com o uso dos classificadores *Adaptive Random Forests* (ARF) e *Support Vector Machines* (SVM). Além disso, o modelo proposto possui adaptação para o recebimento de *feedback* do resultado apresentado do administrador do sistema, fazendo com que o *feedback* seja incorporado no processo de aprendizagem.

Palavras-chave: Aprendizado incremental. Fluxo de dados. Classificação de personas.

ABSTRACT

DA SILVA MORAES, Vinícius. **Interactive Incremental learning for classifying personas** . 2021. 33 p. Monography (Specialization Degree in Data Science and Its Applications) — Federal University of Technology — Paraná, Curitiba, 2021.

The use of personas to identify audiences help in decision making and service delivery. The company analyzed in this monograph conducted an internal research to determine their personas within employee profiles and developed a survey system to classify them. The purpose of this classification is to improve the distribution of hardware to employees, offering ideal options for each profile. With the progressive growth of employee mapping, the survey rule system may become obsolete and, therefore, the objective of this monograph is to analyze and propose an incremental model for classifying the company's employees, bringing the possibility of replacing the survey system in the future. The tecnhnique used to develop the model was the incremental learning method for data streams, where the use of Adaptive Random Forests (ARF) and Support Vector Machines (SVM) classifiers was presented. In addition, the proposed model is adapted to receive feedback on the result presented by the system administrator, causing the feedback to be incorporated into the learning process.

Keywords: Incremental learning. Data streams. Personas classifier.

LISTA DE ILUSTRAÇÕES

Figura 1 – Arquitetura geral da solução 22
Figura 2 – Arquitetura detalhada da etapa de aprendizado incremental 24
Figura 3 – Etapa de *feedback* do administrador 25
Figura 4 – Resultados de *EvaluatePrequential* para SVM e ARF com tamanho de treino 300 26
Figura 5 – Acurácia ao longo do tempo usando ARF 28
Figura 6 – Resultados de desempenho por personas 28
Gráfico 1 – Histograma de pontuações para *Knowledge workers* 18
Gráfico 2 – Histograma de pontuações para *Field worker* 19
Gráfico 3 – Histograma de cargos 20
Gráfico 4 – Mapa de clusters sobre atributos principais 21

LISTA DE TABELAS

Tabela 1 – Parametros utilizados para o método EvaluatePrequential 26
Tabela 2 – Resultados de desempenho do SVM e ARF 27

SUMÁRIO

1 INTRODUÇÃO 8

2 FUNDAMENTOS E TRABALHOS RELACIONADOS 10
2.1 APRENDIZAGEM DE MÁQUINA 10
2.1.1 *Support Vector Machines* 10
2.1.2 *Random Forests* 11
2.1.3 Adaptive Random Forests 12
2.2 APRENDIZADO INCREMENTAL 12
2.2.1 Biblioteca scikit-multiflow 13
2.3 TRABALHOS RELACIONADOS 14

3 PROCESSAMENTO DE DADOS E ANÁLISE EXPLORATÓRIA . . . 16
3.1 CASO DE USO 16
3.2 TRATAMENTO DOS DADOS 17
3.3 ANÁLISE EXPLORATÓRIA 18

4 ARQUITETURA E IMPLEMENTAÇÃO 22
4.1 ARQUITETURA DO SISTEMA 22
4.2 IMPLEMENTAÇÃO 23
4.3 RESULTADOS 26
4.4 DISCUSSÃO 28

5 CONCLUSÃO 30

REFERÊNCIAS 32

1 INTRODUÇÃO

O conceito do uso de personas para entendimento de público é amplamente utilizado em desenvolvimento de *software* desde sua introdução na década de 1990. As personas são representações de pessoas imaginárias que possuem características que, ao serem agrupadas, ajudam na tomada de decisão de gestores que podem, além de se basear em dados e números, saber quais são os desejos e problemas que os usuários enfrentam (SALMINEN, 2019).

Este trabalho utiliza dados de uma empresa que está mapeando seus funcionários gradativamente em personas, tendo a intenção de mapear todos os funcionários ao final (aproximadamente de 100 mil). A empresa optou por dividir toda a população em 4 grupos, implementando um sistema de questionário com pesos para auxiliar na divisão de grupos. O objetivo desse mapeamento, é conseguir melhorar a distribuição de equipamentos para funcionários, entregando opções que são pensadas para otimizar e facilitar o trabalho exigido de cada um.

Dessa forma, os dados disponíveis são de funcionários e suas respectivas personas vindas como resultado do questionário, e também dados comuns sobre o perfil do usuário vindos do sistema de gerenciamento de TI (região, modelo de máquina, *softwares* instalados, cargo, departamento, organização, etc.).

Visando melhorar a classificação rudimentar do questionário, que é baseada somente na intuição e pesquisa feita previamente, este trabalho tem como objetivo auxiliar o processo de descoberta da persona correta para o usuário final. A arquitetura do projeto foi definida de maneira que, utilizando os dados provenientes da empresa (persona, sistema de TI), será treinado um classificador de aprendizado de máquina com o intuito inicial de servir como prova de consistência da informação vinda do questionário. Esse classificador também possui o potencial de, no futuro, conseguir substituir o sistema de questionário.

Dado o dinamismo e crescente atualização desses dados, métodos tradicionais de processamento de dados em lotes não conseguem ser eficientes, visto que geralmente necessitam olhar para os dados como um todo para aplicar seus algoritmos. Portanto, nesse trabalho foi utilizado o método de aprendizado incremental, que além de conseguir atribuir gradualmente novos conhecimentos conforme são introduzidos em um fluxo de dados, pode se adaptar rapidamente a mudanças de conceito com uso

de detectores de drifts (MONTIEL, 2020).

A monografia é organizada da seguinte forma: no Capítulo 2, temos a base teórica utilizada, juntamente com trabalhos relacionados na área de aprendizado incremental. Em seguida, no Capítulo 3, uma breve explicação sobre o conjunto de dados, tratamento aplicado ao mesmo e constatações da análise exploratória. No Capítulo 4 temos a arquitetura e detalhes da implementação, também há uma discussão sobre resultados apresentados. Por fim, no Capítulo 5 será apresentada a conclusão.

2 FUNDAMENTOS E TRABALHOS RELACIONADOS

2.1 APRENDIZAGEM DE MÁQUINA

A aprendizagem de máquina é uma área da inteligência artificial que aborda questões sobre como tornar as máquinas aptas a aprender. O seu objetivo principal é generalizar além dos exemplos existentes no conjunto de treinamento, pois é muito improvável que os mesmos dados apareçam posteriormente (ROZA et al., 2016). A aprendizagem de máquina pode ser dividida em dois sub-grupos de algoritmos: aprendizagem não-supervisionada e aprendizagem supervisionada.

Na aprendizagem não-supervisionada, os algoritmos não conhecem a classe à qual os exemplos pertencem e procuram encontrar nos valores de atributos similaridades ou diferenças que possam, respectivamente, agrupar os exemplos pertencentes à mesma classe ou dispersar os exemplos de classes distintas (STANGE, 2011).

Na aprendizagem supervisionada, é fornecido ao sistema de aprendizado um conjunto de exemplos com a saída conhecida, ou seja, cada exemplo observado é descrito por um conjunto de atributos e pelo valor da classe à qual o exemplo pertence (STANGE, 2011).

Alguns dos algoritmos de aprendizagem supervisionada que são conhecidos pela capacidade de resolução de problemas de classificação serão detalhados a seguir, sendo eles SVM (*Support Vector Machines*), *Random Forests* e ARF (*Adaptive Random Forests*).

No modelo proposto foi utilizado uma variante da aprendizagem supervisionada, conhecida como aprendizagem incremental, que será detalhada na Seção 2.2. O algoritmo usado no modelo foi o *Adaptive Random Forests*, que é capaz de fazer processamento em lotes de fluxo de dados.

2.1.1 *Support Vector Machines*

As máquinas de vetores de suporte são utilizadas na resolução de problemas de classificação e regressão por possuírem capacidade de generalização através de aprendizado na etapa de treinamento (JUNIOR, 2010). Em problemas de classificação, o principal objetivo é construir um classificador a partir de um conjunto de dados

conhecidos que seja capaz de determinar corretamente a classe de novos exemplos (LORENA, 2006).

Em um problema binário, o objetivo da SMV é separar as instâncias de duas classes através de uma função que será obtida a partir de exemplos conhecidos no treinamento. Um classificador é produzido de acordo com um conjunto de padrões identificados no treinamento, onde a classe é conhecida. Esse classificador deve funcionar para casos não conhecidos, adquirindo a capacidade de predizer saídas de futuras novas entradas (JUNIOR, 2010).

Algumas características de SVMs que fazem com que o seu uso seja atrativo são (LORENA; CARVALHO, 2003):

- Boa capacidade de generalização: os classificadores geralmente alcançam bons resultados em generalização, sem *overfitting*
- Robustez em grandes dimensões: são robustas diante de objetos de grandes dimensões, como imagens
- Convexidade da função objetivo: a aplicação das SVMs implica na otimização de uma função quadrática, que possui apenas um mínimo global
- Teoria bem definida: possuem uma base teórica bem estabelecida dentro da matemática e estatística

Devido a sua capacidade de adaptação ao contexto, utilizaremos o SVM no modelo proposto como um dos classificadores a ser avaliado utilizando o aprendizado incremental.

2.1.2 *Random Forests*

Random forests é um algoritmo de aprendizado de máquina supervisionado baseado em árvores de decisão com aprendizado *ensemble*. Aprendizado *ensemble*, por sua vez, é o tipo de aprendizado em que múltiplos algoritmos do mesmo tipo ou diferentes, são combinados para formar uma predição melhorada (MALIK, 2018).

Uma *Random forest* é composta por um número grande árvores que utilizam a seleção de *features* (características) aleatórias durante o aprendizado e também de re-amostragens do conjunto de dados inicial. A riqueza de cobertura de modelos não correlatos faz com que, quando agrupadas, as árvores previnam erros umas das outras, aumentando no geral a precisão e entendimento sobre os dados (MALIK, 2018).

2.1.3 Adaptive Random Forests

Random forest visa o processamentos de dados em lotes, em que o conjunto de dados como um todo precisa ser dividido entre dados de treino e teste e olhar para uma classe de predição y. *Adaptive Random Forests* ou ARF é uma adaptação do algoritmo de *Random forests* e a grande mudança dessa implementação, é a maneira como ele combina características de algoritmos de processamento em lotes com o atualizações dinâmicas para lidar com fluxos de dados em evolução (GOMES et al., 2017). Os aspectos mais importantes do algoritmo ARF:

- Adiciona diversidade através de *resampling*
- Adiciona diversidade através da seleção aleatória de subconjuntos de *features* na divisão de nós da árvore
- Possui um detector de *drift* por árvore base, o que causa redefinições focadas em reposta a *drifts*.

O algoritmo ARF foi o escolhido para o desenvolvimento do modelo proposto devido a sua capacidade de processamento em lotes de fluxo de dados em evolução. Como o tipo de aprendizagem de máquina escolhido foi a aprendizagem incremental, era necessário que o algoritmo fosse capaz de transformar os dados de entrada em um fluxo de dados para que o classificador pudesse aprender conforme fossem adicionados dados de entrada.

2.2 APRENDIZADO INCREMENTAL

Classificação incremental é uma variante da tarefa tradicional de aprendizado de máquina supervisionado. Ambas têm o objetivo de prever um valor nominal de uma instância não classificada representada por um vetor de características.

A principal diferença entre os métodos é que, no caso de cenários de fluxo de dados, as instâncias não estão prontamente disponíveis para o classificador como parte de um grande conjunto de dados. Alternativamente, as instâncias são disponibilizadas sequencialmente e rapidamente ao longo do tempo, como um fluxo de dados contínuo. Além disso, pedidos de previsão podem ocorrer a qualquer momento e o classificador deve usar seu modelo atual para realizá-las (GOMES et al., 2017).

Tradicionalmente, utilizar técnicas de aprendizado de máquina com processa-

mento por lotes é aplicado sobre um conjunto finito de dados e após a fase de treino não existe a necessidade de alterar o modelo.

Um algoritmo de aprendizado para fluxo de dados deve estar preparado para quando ocorrerem mudanças e o modelo estar defasado conseguir se manter aprendendo novos conceitos enquanto retém conhecimento (GOMES et al., 2017).

(MONTIEL, 2020) lista os principais requisitos de um método de aprendizado em fluxos de dados precisa:

- Processar uma amostra por vez e inspecioná-la apenas uma vez
- Usar uma quantidade limitada de memória
- Trabalhar em um quantidade de tempo limitada
- Estar pronto para executar uma previsão a qualquer momento

Pensando que a distribuição de dados muda conforme o tempo passa, também é necessário entender o conceito de *drift*. Eles podem ser causados por variações que estão fora do escopo dos dados apresentados ao algoritmo de aprendizado e esse tipo de fluxo de dado pode ser identificado como dados evolutivos ou não estacionários. De acordo com (TSYMBAL, 2004), um sistema ideal para identificar *drifts* deve: (1) se adaptar rapidamente ao *drift* de conceito; (2) ser robusto contra ruído e o distinguir de *concept drift*; (3) reconhecer e tratar contextos recorrentes. Um método popular para detecção de *drifts* é o *ADaptive WINdowing* (ADWIN).

O modelo proposto nesta monografia faz uso de aprendizagem incremental com fluxo de dados, pois possui a necessidade de aprender conforme novos registros são inseridos no conjunto de dados de entrada. Com o uso da aprendizagem incremental, o modelo pode se manter aprendendo e produzindo bons resultados, mesmo com novos dados sendo adicionados. Além disso, também é possível incluir rodadas de *feedback* no modelo para que as classificações fiquem mais precisas.

2.2.1 Biblioteca scikit-multiflow

A fim de utilizar a técnica de aprendizado incremental, selecionamos a biblioteca scikit-multiflow[1]. Scikit-multiflow é um *framework* para aprendizagem a partir de fluxos de dados, sendo baseado em populares *frameworks* como scikit-learn, MOA e

[1] https://scikit-multiflow.github.io/.

MEKA e também segue princípios de FOSS (*Free and Open Source Software*). A biblioteca proporciona métodos para geração de fluxos de dados, métodos de aprendizado, detectores de *drift* e métodos de avaliação (MONTIEL et al., 2018).

Outras bibliotecas utilizam técnicas de processamento em lotes. Já no scikit-multiflow os estimadores são incrementais por *design* e o treinamento deles é feito a partir de múltiplas chamadas do método partial_fit(). Os dados são representados pela classe *Stream*, onde o método next_sample() é usado para pedir novos dados (MONTIEL, 2020).

2.3 TRABALHOS RELACIONADOS

O foco dado no aprendizado supervisionado tem mudado para outras tarefas e áreas nos últimos anos, visando incorporar mais aspectos do mundo real. Classificação de dados em fluxo de dados, aprendizado incremental e aprendizado online são termos normalmente associados a algoritmos que atualizam seus modelos dado um fluxo contínuo de dados sem precisar passar várias vezes sobre os dados.

Visando se enquadrar nesses requisitos, em (WOŹNIAK, 2013), foi realizada uma análise de um classificador capaz de detectar e se adaptar a mudanças de conceito durante o processamento de um fluxo de dados em pedaços de tamanho k. Nessa solução foi utilizado um algoritmo popular de *ensemble*, o *Accuracy Weighted Ensemble*, que utiliza uma estratégia de votos com peso dependendo da sua precisão como forma de adaptação.

Já em (MASUD et al., 2008), foi proposto um algoritmo para lidar com dados de treinamento parcialmente rotulados de um fluxo de dados, sendo capaz de produzir resultados tão bons ou melhores quando comparados com outras abordagens que usam dados totalmente rotulados. Os dados do fluxo de dados são divididos em pedaços iguais e um modelo de classificação é treinado para cada pedaço. O modelo foi criado usando um algoritmo de agrupamento semi-supervisionado para criar clusters a partir dos dados parcialmente rotulados de treinamento e, para classificar uma instância de testes, foi utilizado o algoritmo KNN (*K—Nearest Neighbors*), algoritmo de aprendizagem supervisionada de classificação, para encontrar o cluster mais próximo dessa instância.

Em (SHI et al., 2014), foi proposta uma abordagem de aprendizagem incremental que reconhece dinamicamente novas combinações de rótulos e as atualiza em

tempo real, tornando o aprendizado do modelo mais preciso. Para isso, é utilizada é estratégia de aprendizagem em duas fases. Na primeira, uma série de amostras com combinações de rótulos são coletados para inicializar o modelo de aprendizagem. Na segunda fase, para cada amostra subsequente é verificado se o seu rótulo não está no conjunto de combinações; se não estiver, o rótulo será salvo e o número de ocorrência é atualizado em tempo real. Se o número de ocorrência de uma combinação for maior do que qualquer combinação de rótulo já existente, então as amostras correspondentes serão usadas para atualizar o modelo pela estratégia de aprendizagem incremental. Porém, caso o rótulo da amostra recém chegada esteja no conjunto de combinações, então o rótulo será usado para atualizar o modelo pela estratégia de aprendizagem incremental da instância.

Apesar de buscarem resolver problemas de classificação de dados, todos utilizam estratégias diferentes para chegar ao objetivo. O modelo proposto nesta monografia, assim como em (SHI et al., 2014), faz uso de aprendizagem incremental. Além disso, faz o processamento em fluxo de dados, divididos em pedaços menores e, como em (WOŹNIAK, 2013), também utiliza a estratégia de votos com peso, para fazer com que as classificações fiquem mais precisas. Apesar do modelo ser treinado para cada pedaço do fluxo de dados, como feito em (MASUD et al., 2008), o algoritmo utizado para classificação dos registros é o *Adaptive Random Forests*.

3 PROCESSAMENTO DE DADOS E ANÁLISE EXPLORATÓRIA

3.1 CASO DE USO

Uma das dificuldades de departamentos de informática de empresas é otimizar o ciclo de vida dos ativos de TI, fazendo com que os equipamentos sejam utilizados da melhor forma possível para que seus recursos sejam bem aproveitados. Da mesma forma, existe uma preocupação em fornecer aos funcionários os equipamentos que façam com que suas funções sejam executadas de forma eficiente (SILVA LEITE; REIS, 2021).

Dentro da empresa de tecnologia estudada, foi criada uma divisão com foco em estudar e identificar possíveis melhorias para o ciclo de vida dos ativos de TI existente na companhia. Um dos conceitos estudados foi a introdução de perfis de utilização dos equipamentos, visando agrupar usuários em perfis virtuais.

Para a criação dos perfis, foi feita uma entrevista com 50 funcionários da empresa, divididos em 12 países, com o objetivo de descobrir experiências e dificuldades para adquirir e usar computadores. A partir desse estudo foram criadas 4 personas, cada uma com um perfil diferente e que requer um equipamento com características específicas para atender às suas necessidades.

As personas definidas foram *Knowledge workers*, *Power workers*, *Mobile workers* e *Field workers*. Entre as suas características, *Knowledge workers* são aqueles usuários em que seu maior uso dos equipamentos está em editores de texto, apresentações e planilhas, além de algumas ferramentas específicas. Já os definidos como *Knowledge workers* possuem requisitos de alta performance de *hardware* e fazem uso de *softwares* específicos, como desenvolvimento de código, *design* gráfico e edição de áudio e vídeo. Os usuários classificados como *Mobile workers* estão sempre trocando de localidade e tem dificuldades com peso e número de acessórios de *hardware*. *Field workers* não costumam atuar em um escritório tradicional e precisam de equipamentos leves e robustos, além disso, o equipamento não é totalmente necessário para completar o trabalho.

Na tentativa de mapear os usuários e as suas respectivas personas foi desenvolvido um sistema no formato de questionário, com o foco em entender o perfil de trabalho dos funcionários. O questionário foi construído com 6 perguntas de múltipla

escolha e possui um sistema de regras que atribui pesos às respostas de cada pergunta. Ao final do questionário, a classe de persona é determinada de acordo com a pontuação obtida.

Um dos problemas encontrados na implementação do novo modelo foi a falta de acesso ao que foi respondido no questionário pelos funcionários. Todas as respostas são apagadas após o processamento devido à necessidade dos responsáveis pela ferramenta, sendo impossível recuperá-las após a classe de persona ser determinada. Dessa forma, os dados obtidos estão em uma matriz de relação entre usuários e personas, resultado do questionário.

Além disso, também foi fornecido os detalhes do equipamento atual de cada funcionário que respondeu o questionário, contendo quais *softwares* estão instalados, modelo do computador, tamanho de memória, etc. Esses dados, juntamente com a matriz de usuários e personas compõe o conjunto de dados total utilizado. Os dados de máquinas devem ser revistos no futuro, visto que se demonstraram insuficientes ao longo do trabalho.

Sendo assim, temos como objetivo buscar uma forma de complementar o sistema atual e potencialmente substituí-lo por um modelo escalável e adaptativo.

3.2 TRATAMENTO DOS DADOS

A etapa de tratamento de dados começou durante a extração do mapeamento de "Persona vs Usuário" feito na ferramenta de questionário. Durante esse etapa de extração, foi constatado que informações como histórico de respostas não estavam disponíveis para consumo devido a políticas de segurança da informação aplicadas ao banco de dados da aplicação. Dessa forma, somente a combinação de identificador de usuário e persona foi extraído desse sistema. Além disso, existia uma diferença na formatação de alguns campos entre o sistema do questionário e o sistema da TI, que também foi solucionado antes de aplicar a junção entre as informações.

Após os ajustes necessários, foi possível, através do identificador de usuário, localizar detalhes de perfil, como máquinas (modelo, processador, disco, memória ram, etc.), *softwares* instalados e informações relacionadas à identidade do usuário, como país, região, departamento, organização, cargo.

Buscando agrupar as informações, foram excluídas pessoas que, por algum motivo, não tiveram máquinas assinaladas ao seu nome ou que deixariam variáveis nulas. Inicialmente existiam 739 registros de personas, porém esse número foi reduzido por conta da filtragem, restando 633 usuários.

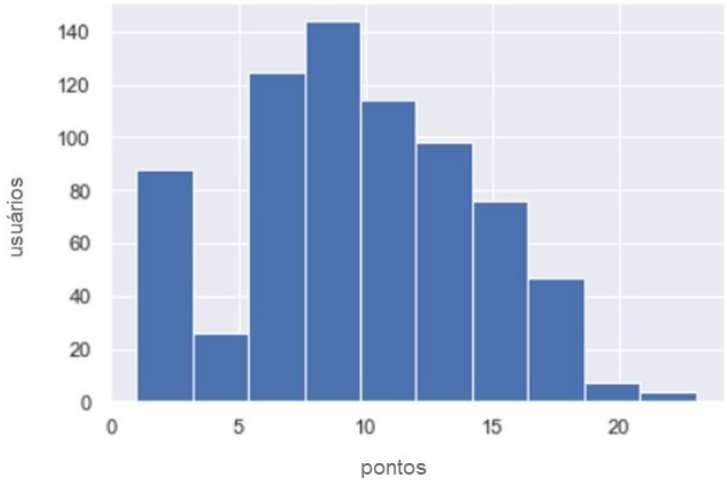
O último tratamento feito antes de iniciar a análise exploratória foi no tipo das colunas existentes. Como a maioria das colunas eram do tipo categórica ou formato texto, foi necessário aplicar o método get_dummies para transformá-las em colunas de 0's e 1's, visto que somente nesse formato é possível aplicar algoritmos e análises em aprendizado de máquinas.

3.3 ANÁLISE EXPLORATÓRIA

Durante a análise exploratória dos dados foi possível observar que algumas personas possuem mais usuários mapeados e, por consequência, seus dados são mais ricos para identificar suas características. Os 633 usuários foram distribuídos entre as 4 personas da seguinte forma: 46% são *Knowledge workers*, 26% são *Knowledge workers*, 16% são *Mobile workers* e 11% são *Field workers*.

Além disso, olhando para a distribuição da pontuação adquirida no questionário, podemos observar que as classes mais completas tem pontuações mais variadas, como no exemplo do Gráfico 1.

Gráfico 1 – Histograma de pontuações para *Knowledge workers*



Fonte: autoria própria (2021).

Como a persona de *Knowledge workers* ocupa quase metade dos usuários mapeados, o intervalo de pontos com várias ocorrências é maior. Outra observação

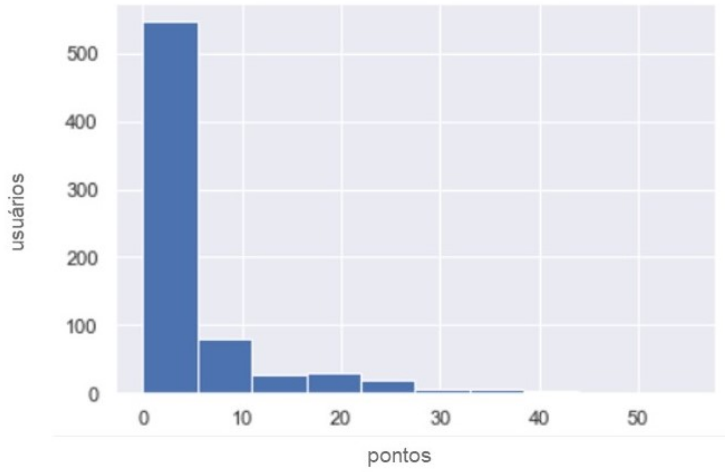
sobre esta classe de personas é que ela representa usuários que têm o perfil mais padronizado em termos de uso de *hardware* e *software*. Logo, é esperado que mesmo usuários que sejam mapeados para outra persona tenham ao menos alguma pontuação atribuída a *Knowledge worker*.

Conceitualmente, a diferença entre personas baseado em dados é definida em pequenos detalhes. Como por exemplo, uma diferença entre um *Knowledge worker* com um perfil padrão de uso de *hardware* e *softwares*, e um *Power worker* que precisa de *hardware* superior, pode ser somente um modelo de processador ou um *software* relacionado a desenvolvimento de sistemas.

Considerar apenas a configuração da máquina atual do usuário também pode ser prejudicial aos dados, visto que o usuário pode ter recebido uma máquina que não condiz com suas necessidades e isso é pré-adoção do estudo de personas.

Em contraste, como mostra o Gráfico 2, a classe da persona de *Field workers* possui a menor quantidade de usuários mapeados e, por consequência, a distribuição da pontuação do questionário fica próxima de zero. O que indica que existem poucas perguntas que atribuem peso a essa persona no questionário e que essas opções são nichadas e direcionadas. Apesar disso, se houvesse uma quantidade maior de dados e, consequentemente, uma maior quantidade de usuários mapeados para essa persona, o comportamento apresentado no Gráfico 2 seria diferente.

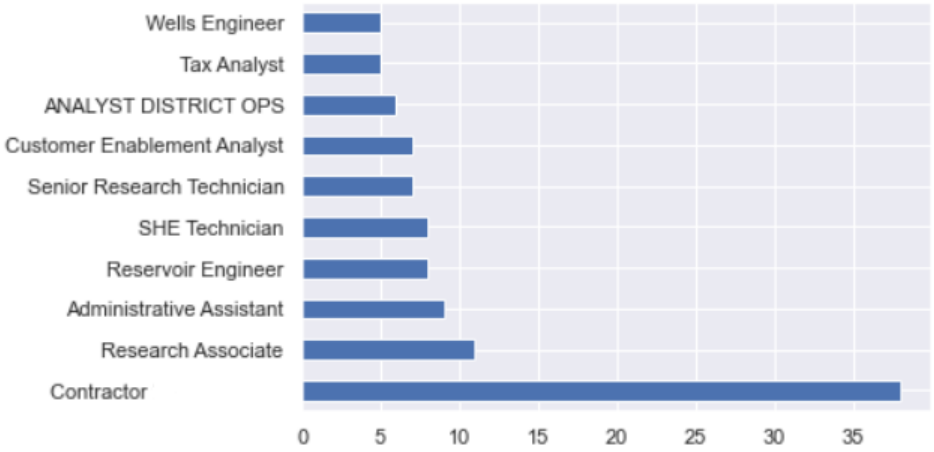
Gráfico 2 – Histograma de pontuações para *Field worker*



Fonte: autoria própria (2021).

Um dos atributos usado como um parâmetro para definir a persona de um usuário é o seu cargo ou título do seu trabalho. No Gráfico 3 estão os 10 cargos com mais ocorrências no conjunto dados.

Gráfico 3 – Histograma de cargos



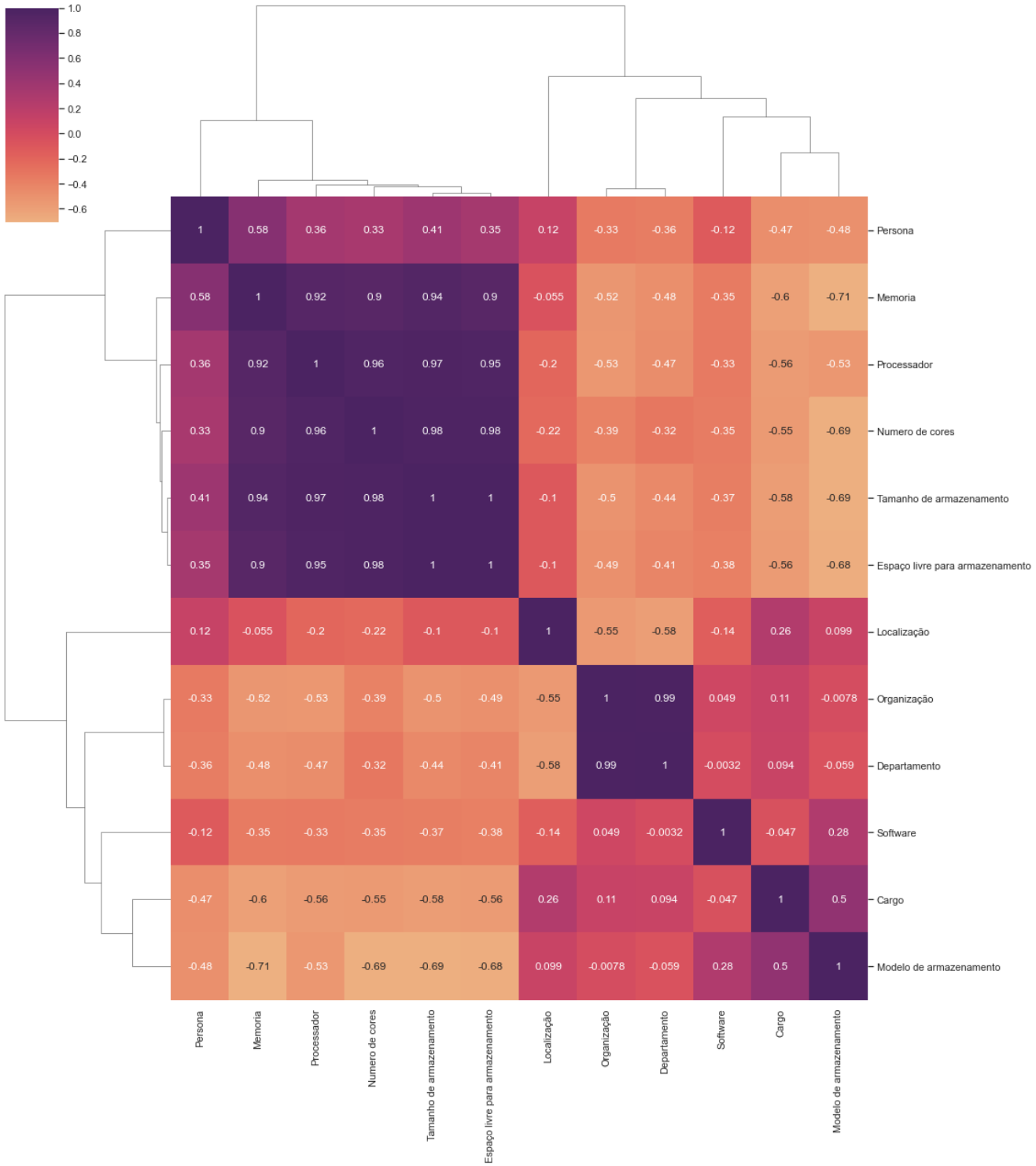
Fonte: autoria própria (2021).

De acordo com a listagem, o classificador pode ter problemas com a relevância do atributo cargo, levando em consideração que a maior ocorrência de cargos é de Contractor, que não é um diferenciador real de perfil. Ou seja, ser do cargo Contractor não ajuda na comparação de características de uma persona.

Ao observar as correlações e agrupamentos do Gráfico 4, é possível notar que os atributos que são relacionados ao *hardware* (Memória, Processador, Número de cores, Tamanho de armazenamento, Espaço livre de armazenamento, Modelo de armazenamento) que a pessoa possui têm correlações fortes entre si, o que indica que os modelos de *hardware* são bem definidos dentro do ambiente com apenas algumas variações. A pesquisa para definição das personas têm um forte relacionamento com a oferta de *hardware*, logo se torna importante e positivo que o atributo da persona tenha uma correlação mais alta com esse conjunto.

A partir do que foi descoberto durante a análise exploratória, foi possível definir que os melhores atributos a serem utilizados pelo classificador seria uma combinação entre atributos de *hardware* e identidade, pois existem indícios que esses atributos podem suportar o processo decisório de um administrador e, por consequência, podem repassar esse conhecimento ao classificador.

Gráfico 4 – Mapa de clusters sobre atributos principais



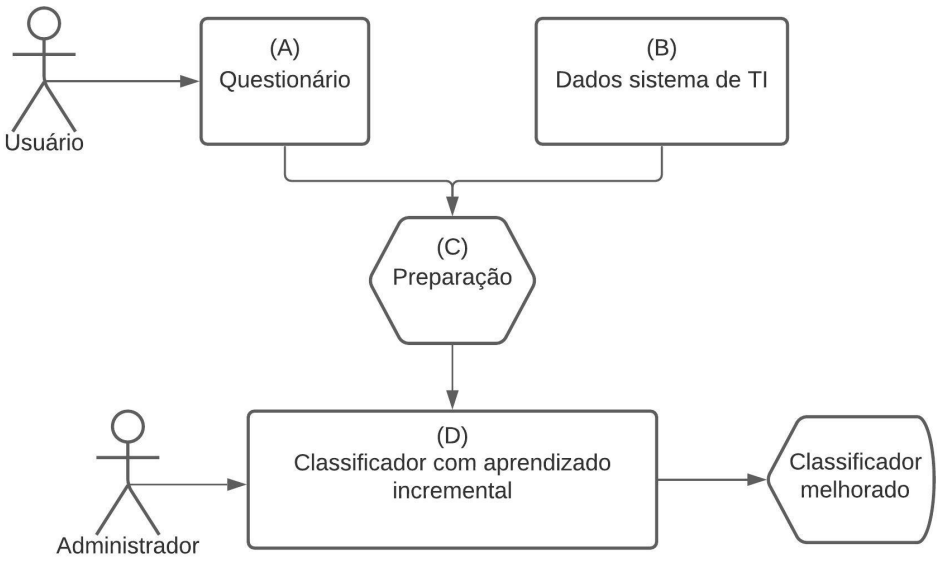
Fonte: autoria própria (2021).

4 ARQUITETURA E IMPLEMENTAÇÃO

4.1 ARQUITETURA DO SISTEMA

A arquitetura foi definida a partir das classificações geradas pela interação do usuário com o questionário (A), que é correspondente ao atual sistema em produção. O questionário é um sistema que aplica pesos a respostas baseado em regras pré-definidas. Como já mencionado, o resultado dessa interação são registros no formato usuário-persona. Para complementar as informações, foi utilizada uma base de dados mantida pelo setor de TI para identificar usuários (B). Essa base é composta por informações de sistemas de RH, TI e controle de acessos.

Figura 1 – Arquitetura geral da solução



Fonte: autoria própria (2021).

Na fase de preparação (C), o usuário teve suas informações detalhadas de perfil combinadas com sua persona. Essa junção é importante para que o classificador consiga criar conhecimento baseado nos atributos que definem o perfil do usuário, sendo que eles estão implicitamente inclusos dentro do questionário, que é o classificador base desse sistema. O classificador com aprendizado incremental (D), ilustrado na Figura 1, recebe as informações formatadas para treinar o modelo de escolha.

Ao final, o resultado deverá ser um classificador treinado continuamente com sistema de *feedback* e pesos. Detalhes sobre a implementação são explicados na

próxima sessão.

4.2 IMPLEMENTAÇÃO

Para conseguir alcançar o resultado esperado na implementação, era necessário encontrar um método de classificação para melhorar a confiança do resultado proposto no questionário, podendo atestar a eficácia da aplicação do questionário. Ao mesmo tempo, o novo classificador poderia ser treinado para que, no futuro, passasse a ser parte da checagem de classificação de usuários dentro das personas. Esse classificador auxiliar também tem o potencial de remover a necessidade de aplicação do questionário no futuro, visto que seria treinado utilizando as informações de registro no sistema de controle de TI (configuração de computador, *software* utilizados, cargo, região).

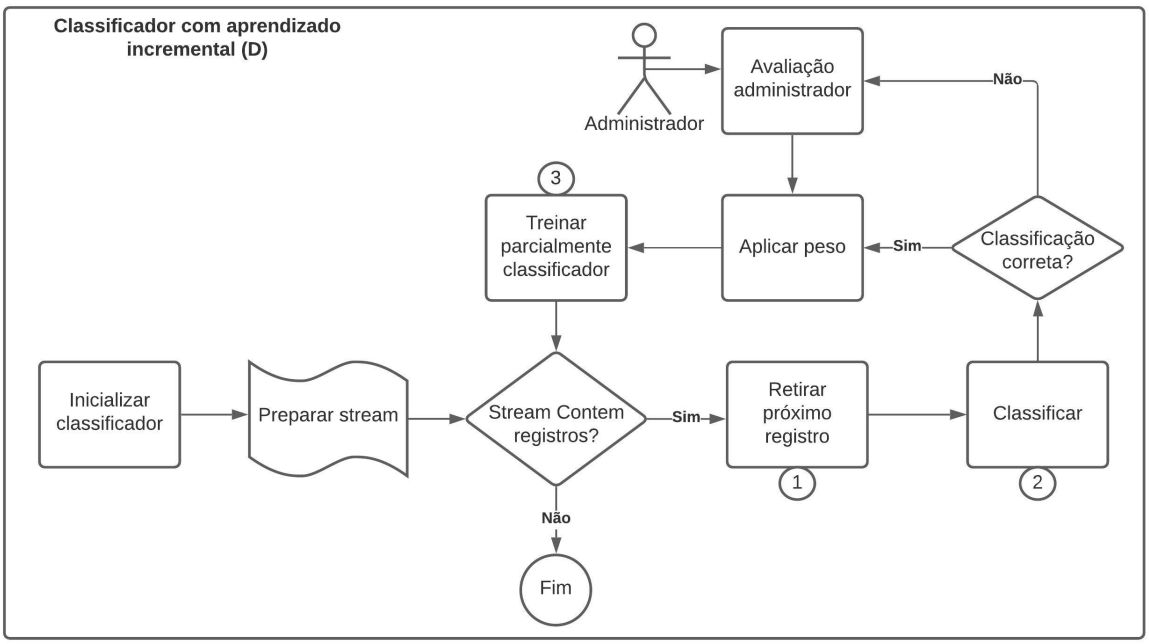
Para a implementação desse classificador foi utilizada a técnica de aprendizado incremental, que tem a capacidade de aprender conforme novos usuários fossem adicionados aos dados de entrada sem prejudicar o conhecimento adquirido previamente. Por consequência, o classificador possui a capacidade de adaptação as mudanças no domínio do problema, o que não é possível utilizando o questionário.

O *framework* escolhido para aplicar o aprendizado incremental é scikit-multiflow, em que a partir de um fluxo de dados, podemos controlar o fluxo de aprendizado de um classificador. Na solução proposta foi utilizado o algoritmo de *Adaptive Random Forests* adaptado ao contexto de aprendizado incremental e à transformação dos dados de entrada em um fluxo de dados. Também realizou-se uma breve comparação entre o comportamento do algoritmo utilizando *Adaptive Random Forests* e SVM.

Scikit-multiflow implementa o método EvaluatePrequential(), que é um avaliador de classificadores que aceita como entrada: um fluxo de dados; quais classificadores serão utilizados; métricas necessárias; taxa de atualização e treinamento de dados conforme o fluxo de dados é percorrido. Esse método é utilizado na implementação para fins de comparação e visualização, mas não é o método principal da implementação proposta por não aceitar uma abertura para tomada decisão pós predição.

Dessa forma, essa implementação proposta segue os mesmos padrões da implementação de EvaluatePrequential, porém introduz uma etapa de *feedback* de um administrador após predições incorretas (Figura 2). Além de treinar o modelo com

Figura 2 – Arquitetura detalhada da etapa de aprendizado incremental

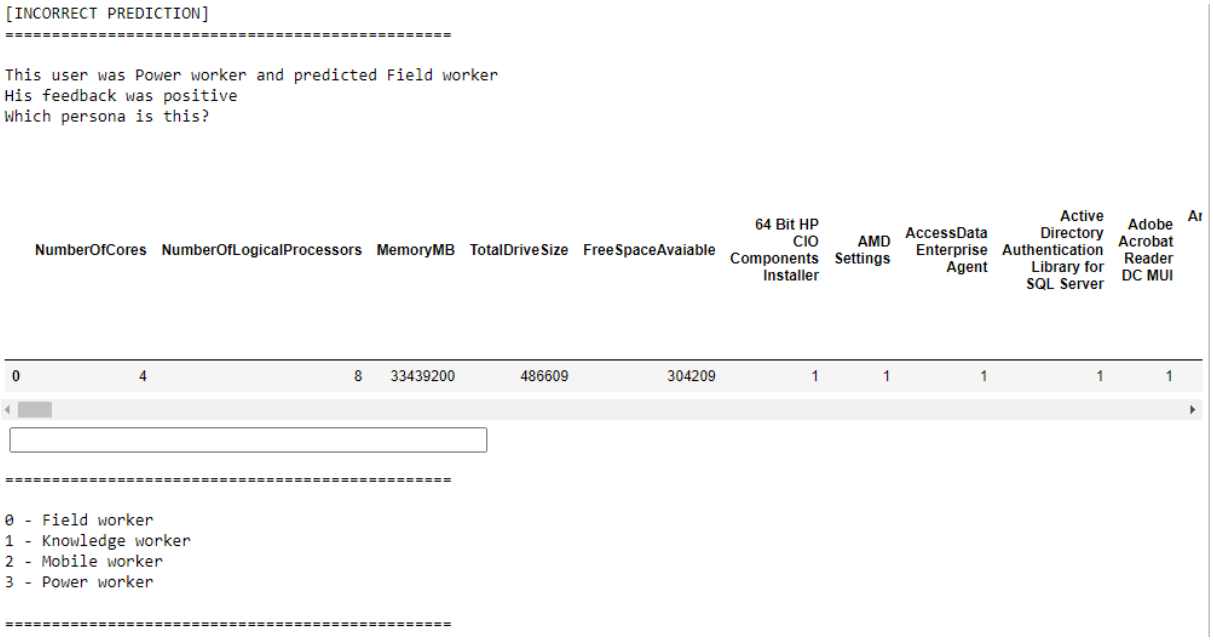


Fonte: autoria própria (2021).

mais confiança após o *feedback*, outra vantagem de customizar a implementação de EvaluatePrequential é a possibilidade de aplicar pesos variados baseados nas combinações de predições, *feedbacks* vindo do administrador e também *feedback* do usuário.

Observando a Figura 2, a implementação no geral pode ser definida da seguinte forma: para cada registro vindo do fluxo de dados (1), é executada uma previsão de classificação (2). Se a previsão for correta, ou seja, tem a mesma classe de persona vinda do questionário, o modelo é treinado com a mesma previsão após aplicação de um peso (3). No caso de previsões incorretas, temos a etapa de pedido de avaliação para um administrador. Nessa etapa o administrador ajudará o modelo a entender qual a classe correta para o registro, tendo a oportunidade de informar a sua classificação baseada nos dados informados. Uma das possibilidades que essa implementação proporciona é a de manter esses registros disponíveis para avaliação conforme a disponibilidade do administrador, sendo possível então que o próprio administrador inicie um novo aprendizado parcial no classificador. O mecanismo de *feedback* também é uma melhoria significativa em comparação com o questionário, em que o *feedback* do usuário precisava ser revisto manualmente e com ajuda da área de negócio decidir mudanças nas perguntas ou pesos atribuídos as personas.

Figura 3 – Etapa de *feedback* do administrador



```
[INCORRECT PREDICTION]
================================================

This user was Power worker and predicted Field worker
His feedback was positive
Which persona is this?
```

	NumberOfCores	NumberOfLogicalProcessors	MemoryMB	TotalDriveSize	FreeSpaceAvaiable	64 Bit HP CIO Components Installer	AMD Settings	AccessData Enterprise Agent	Active Directory Authentication Library for SQL Server	Adobe Acrobat Reader DC MUI	Ar
0	4	8	33439200	486609	304209	1	1	1	1	1	

```
================================================

0 - Field worker
1 - Knowledge worker
2 - Mobile worker
3 - Power worker

================================================
```

Fonte: autoria própria (2021).

Como parte da avaliação, o administrador é informado de como o questionário classificou o registro, como o modelo classificou o registro, qual foi o *feedback* do usuário que respondeu o questionário sobre a classificação e o dados de controle do sistema de TI (Figura 3). Havendo um *feedback* positivo do usuário sobre a classificação do questionário, se o administrador escolher a mesma persona que o modelo, é aplicado um peso significativo a essa previsão. Se a seleção do administrador for diferente do modelo, um peso moderado é aplicado a classificação vinda do administrador. Por fim, se o usuário prover um *feedback* negativo sobre a classificação do questionário, qualquer entrada vinda do administrador será considerada com um peso significativo para o aprendizado do modelo, visto que tem maior chance de estar correta. A aplicação dos pesos é mais uma forma de alcançar adaptabilidade e também dar prioridade ao conhecimento do administrador no julgamento da classificação.

Esse processo se repete até que o fluxo de dados não tenha mais registros. O resultado final é um modelo adaptativo treinado incrementalmente com auxílio de um administrador.

4.3 RESULTADOS

Para ajudar na análise dos resultados foi utilizado o método padrão de avaliação da biblioteca scikit multiflow *EvaluatePrequential* e a implementação customizada com *feedback* de administradores. O método *EvaluatePrequential* aceita como parâmetros os classificadores SVM e ARF, e como métricas f1, recall e precision. A escolha das métricas visa avaliar a distribuição de falsos negativos e positivos no modelo, para então entender suas características.

Tabela 1 – Parametros utilizados para o método EvaluatePrequential

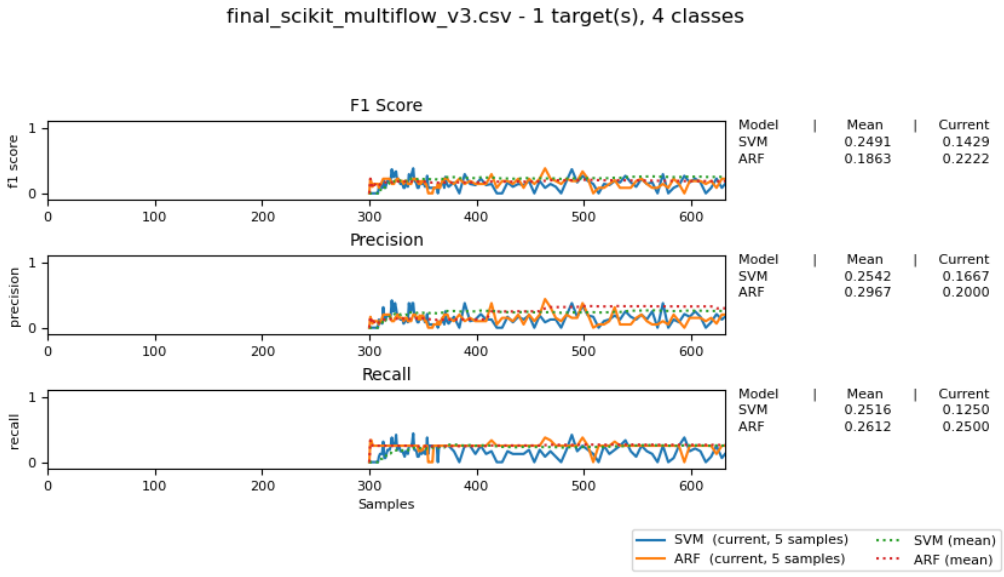
Classificadores	Métricas	Tamanho pré-treino	Tamanho da janela
SVM,ARF	f1,precision,recall	300	5

Fonte: autoria própria (2021).

Para esse experimento foram testadas várias configurações de tamanho de dados de treino, porém como não existe representatividade de todas as classes dentro do conjunto de dados, o treino não teve influência sobre o resultado. Dessa forma, para uma quantidade maior de dados e mais balanceado entre as classes, os resultados tenderiam a ser melhores.

Como podemos observar na Figura 4, todas as métricas sofrem variação constante. O que indica uma introdução de *drifts* de conceito ao decorrer do aprendizado.

Figura 4 – Resultados de *EvaluatePrequential* para SVM e ARF com tamanho de treino 300



Fonte: autoria própria (2021).

De acordo com as métricas do experimento, conforme Tabela 2, podemos observar que o classificador ARF e o SVM desempenharam basicamente da mesma forma. A quantidade de falsos positivos e falsos negativos é muito alta, reduzindo a performance de todas as métricas. Ainda não é possível confiar no classificador atual e isso se deve principalmente à qualidade dos dados usados (*software* e *hardware*). Por quase metade dos dados serem relacionados a *Knowledge workers*, a quantidade das demais classes não foi suficiente para que o classificador pudesse ter um bom aprendizado e tivesse um bom desempenho.

Tabela 2 – Resultados de desempenho do SVM e ARF

	SVM	**ARF**
Precision	0.2542	0.2967
Recall	0.2516	0.2612
F1 Score	0.2491	0.1863

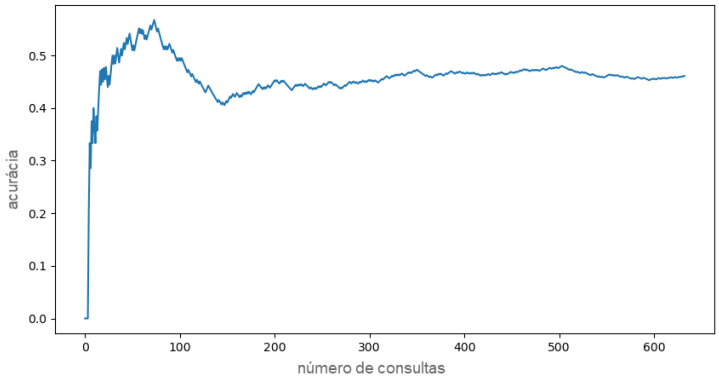
Fonte: autoria própria (2021).

Passando para a implementação customizada do aprendizado incremental com *feedback*, foi utilizado apenas o algoritmo ARF para facilitar a visualização. Nessa implementação, a cada predição incorreta temos a possibilidade de interferência de um administrador para influenciar na classe de predição e no peso atribuído ao registro. Como podemos observar na Figura 5, o classificador tem dificuldades de manter seu desempenho. Um dos motivos pode ser o mesmo citado anteriormente, a quantidade de dados de cada classe não ser significativa para que o aprendizado trouxesse classificações melhores.

Olhando mais atentamente o desempenho das métricas de escolha na Figura 6, podemos ver que as classes *Mobile worker* e *Power worker* não tiveram predições suficientes para o cálculo das métricas, o que já diminui a média geral.

Outra observação é sobre a classe de *Knowledge workers*, que tem o melhor desempenho entre seus pares. Isso se dá parcialmente por um problema já apontado de escassez de dados, mas também porque os atributos parecem favorecer essa classificação, retornando a maioria dos resultados relevantes e resultando em um recall elevado. Porém, mesmo nesse caso, a precision continua sendo baixa com a representatividade de 293 exemplos. Dessa forma, podemos afirmar que essa quantidade e qualidade de distribuição de dados não é suficiente para o conhecimento de um classificador. Porém, em uma implantação do sistema, o acesso aos dados detalhados sobre os usuários

Figura 5 – Acurácia ao longo do tempo usando ARF



Fonte: autoria própria (2021).

Figura 6 – Resultados de desempenho por personas

	precision	recall	f1-score	support
Field worker	0.36	0.06	0.10	71
Knowledge worker	0.46	0.98	0.63	293
Mobile worker	0.00	0.00	0.00	103
Power worker	0.00	0.00	0.00	166
accuracy			0.46	633
macro avg	0.21	0.26	0.18	633
weighted avg	0.26	0.46	0.30	633

Fonte: autoria própria (2021).

resolveria estas questões.

4.4 DISCUSSÃO

Como podemos ver, o classificador tem dificuldades em aprender dado o conjunto atual de dados. Um dos principais motivos que pode causar esse problema é a falta de clareza sobre quais são os fatores mais influentes no processo de decisão para definição da persona de um usuário. Principalmente quando entramos nas classes com menos representatividade no conjunto de dados, a diferença entre os atributos dos usuários é quase inexistente, indicando uma subjetividade na definição de persona.

A dificuldade de retenção do conhecimento também pode estar associada ao volume de atributos selecionados, tendo como exemplo o uso de *softwares* instalados

estar introduzindo entradas recorrentes e repetidas para *softwares* padrão em todas as máquinas.

Com um volume maior de entradas significativas para todas as personas no questionário e a correção no foco dos atributos para o classificador, esta implementação tem o potencial de servir como prova de conceito contra a classificação do questionário. Durante esse processo de aumento do conjunto de dados, a empresa deverá acompanhar a evolução do classificador e fornecer o treinamento por *feedback* corretamente.

Por fim, o questionário poderia disponibilizar mais informações como a resposta exata do texto atribuída no momento da interação com o usuário. Isso poderia enriquecer o conhecimento do classificador e até possibilitar novas análises.

5 CONCLUSÃO

O modelo proposto nesta monografia pode ser utilizado para validação da resposta obtida pelo questionário e, até mesmo, a substituição do sistema atual. O objetivo é informar a classe de persona do usuário dado um conjunto de características.

A maior dificuldade encontrada na construção e validação do modelo proposto na monografia foi a quantidade e qualidade de dados. A quantidade de registros fornecidos pela empresa não foi suficiente para que houvessem resultados satisfatórios entre as classes de personas. Além disso, as *features* disponíveis não eram as ideais para a montagem do modelo, fazendo com que os resultados fossem insuficientes. A adaptação do modelo para que houvessem *feedbacks* do administrador foi desenvolvida com o objetivo de tornar os resultados melhores e fazer com que o modelo proposto aprendesse com os entradas do administrador. Toda a funcionalidade foi implementada e testada localmente, porém não foi possível realizar os testes com o administrador real do sistema para garantir que houve ganho significativo para o modelo proposto. No entanto, os testes executados utilizaram o conhecimento adquirido pelo estudo afim guiar as decisões, simulando o que poderia ser a entrada desses administradores. Mesmo nesse caso não foi possível identificar ganho real sobre a assertividade do modelo.

Apesar das dificuldades e problemas encontrados durante o desenvolvimento da monografia, para fazer com que o modelo proposto seja implementado na empresa em questão existem alguns ajustes que precisam ser feitos. Entre eles, seria necessário a utilização de uma quantidade maior de dados e uma quantidade mais balanceada dos dados de cada classe, para que o modelo pudesse identificar melhor as diferenças entre elas. Além disso, seria interessante ter acesso à *features* que não foram disponibilizadas para a elaboração desta monografia, com isso, seria possível determinar com maior precisão quais são as *features* que fazem com que o desempenho do modelo seja melhor.

Ainda antes da implementação em ambiente produtivo, é necessário que sejam executados os testes com os *feedbacks* do administrador do sistema. Como a adaptabilidade do modelo é um diferencial do classificador, é necessário garantir que os *feedbacks* fornecidos pelo administrador sejam utilizados na aprendizagem do modelo proposto. Com esses ajustes realizados, a implementação do modelo proposto nesta

monografia pode ser feita na empresa.

Os resultados obtidos a partir dos ajustes citados podem ser tratados em trabalhos futuros, detalhando quais novas *features* foram utilizadas e qual a quantidade de dados é necessária para fazer com que o modelo proposto tenha um desempenho melhor do que o apresentado nesta monografia. Além disso, outro estudo que pode ser feito é a comparação do modelo proposto com o uso dos *feedbacks* do administrador após os ajustes e sem esses *feedbacks*.

REFERÊNCIAS

GOMES, H. M. et al. Adaptive random forests for evolving data stream classification. **Machine Learning**, v. 106, p. 1–27, out. 2017. DOI: 10.1007/s10994-017-5642-8.

JUNIOR, G. Máquina de Vetores Suporte: estudo e análise de parâmetros para otimização de resultado. **Ciência da Computação, Universidade Federal de Pernambuco**, 2010.

LORENA, A. C. **Investigação de estratégias para a geração de máquinas de vetores de suporte multiclasses**. 2006. Tese (Doutorado) – Universidade de São Paulo.

LORENA, A. C.; CARVALHO, A. C. de. Introduçaoas máquinas de vetores suporte. **Relatório Técnico do Instituto de Ciências Matemáticas e de Computaçao (USP/Sao Carlos)**, v. 192, p. 11, 2003.

MALIK, U. **Random Forest Algorithm with Python and Scikit-Learn**. [S.l.: s.n.], 2018. https://stackabuse.com/random-forest-algorithm-with-python-and-scikit-learn. Acessado em: 18/04/2021.

MASUD, M. M. et al. A practical approach to classify evolving data streams: Training with limited amount of labeled data. In: IEEE. 2008 Eighth IEEE International Conference on Data Mining. [S.l.: s.n.], 2008. P. 929–934.

MONTIEL, J. Learning from evolving data streams. In: p. 70–77. DOI: 10.25080/Majora-342d178e-00a.

MONTIEL, J. et al. Scikit-Multiflow: A Multi-output Streaming Framework. **Journal of Machine Learning Research**, v. 19, out. 2018.

ROZA, F. S. d. et al. Aprendizagem de máquina para apoio à tomada de decisão em vendas do varejo utilizando registros de vendas. Florianópolis, SC., 2016.

SALMINEN, J. **How to Use Personas? Listing Typical Persona Use Cases**. 2019. Disponível em: https://persona.qcri.org/blog/how-to-use-personas-listing-typical-persona-use-cases/#:~:text=Introduction%5C%20to%5C%20Persona%5C%20Use%5C%20Cases&text=Personas%5C%20capture%5C%20and%5C%20describe%5C%20key,and%5C%20desires%5C%2C%5C%20needs%5C%20and%5C%20wants.. Acesso em: 18 abr. 2021.

SHI, Z. et al. Efficient class incremental learning for multi-label classification of evolving data streams. In: IEEE. 2014 international joint conference on neural networks (IJCNN). [S.l.: s.n.], 2014. P. 2093–2099.

SILVA LEITE, L. da; REIS, A. C. B. Modelo Multicritério para Avaliação de Ciclo de Vida de Ativos de TI. **Brazilian Journal of Development**, v. 7, n. 2, p. 16181–16211, 2021.

STANGE, R. L. **Adaptatividade em aprendizagem de máquina: conceitos e estudo de caso.** 2011. Tese (Doutorado) – Universidade de São Paulo.

TSYMBAL, A. The problem of concept drift: definitions and related work. **Computer Science Department, Trinity College Dublin**, Citeseer, v. 106, n. 2, p. 58, 2004.

WOŹNIAK, M. Application of combined classifiers to data stream classification. In: SPRINGER. IFIP International Conference on Computer Information Systems and Industrial Management. [S.l.: s.n.], 2013. P. 13–23.