

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

MILLER BIAZUS

**Recomendação de Recursos de
Infraestrutura Necessários para a
Implementação de Processos a Partir de
Rótulos de Modelos de Processo**

Dissertação apresentada como requisito parcial para
a obtenção do grau de Mestre em Ciência da
Computação

Orientadora: Profa. Dra. Lucinéia Heloisa Thom

Porto Alegre
2018

CIP — CATALOGAÇÃO NA PUBLICAÇÃO

Biazus, Miller

Recomendação de Recursos de Infraestrutura Necessários para a Implementação de Processos a Partir de Rótulos de Modelos de Processo / Miller Biazus. – Porto Alegre: PPGC da UFRGS, 2018.

80 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2018. Orientador: Lucinéia Heloisa Thom.

1. Gerenciamento de Processos de Negócio. 2. Sistemas de recomendação. 3. Aprendizagem de Máquina. 4. Recomendação de recursos. 5. Revisão Sistemática da Literatura. I. Thom, Lucinéia Heloisa. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Rui Vicente Oppermann

Vice-Reitora: Prof^a. Jane Fraga Tutikian

Pró-Reitor de Pós-Graduação: Prof. Vladimir Pinheiro do Nascimento

Diretora do Instituto de Informática: Prof^a. Carla Maria Dal Sasso Freitas

Coordenador do PPGC: Prof. João Luiz Dihl Comba

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

*“Deus concede o progresso a passos lentos,
porque a luz repentina ofusca a vista.”*

— ARAÚJO PORTO ALEGRE

AGRADECIMENTOS

Meu primeiro agradecimento é direcionado à Profa. Dra. Lucinéia Heloisa Thom, pelo incentivo, por acreditar em mim, me guiar e inspirar. Não tenho palavras para agradecer pela paciência de orientar um aluno com horários divididos entre o trabalho e a universidade, por me instigar em momentos que eu duvidei da minha capacidade. Agradeço ainda por, desde o trabalho de graduação, me conduzir e ter me tornado melhor a cada capítulo escrito. Tudo isso tornou possível a conclusão deste trabalho. Agradeço ainda a oportunidade de ser orientado por uma profissional tão bem qualificada e com ética profissional exemplar.

Dedico meu agradecimento também à banca examinadora:

- Ao Prof. Dr. Márcio Dorn, que tive a honra de ter como meu coorientador durante a graduação, me trazendo conhecimento, agregando a cada palavra escrita, sendo paciente e colaborativo em todos os momentos, e que desde então possui minha admiração;
- À Profa. Dra. Isabela Gasparini, que se prontificou, mesmo à distância, para avaliar e discutir sobre meu trabalho, trazendo sugestões e críticas;
- Ao Prof. Dr. Marcelo Soares Pimenta, por aceitar participar da banca e, consequentemente, trazer toda sua experiência e expertise para agregar no meu trabalho.

Dedico meu trabalho aos meus amigos, colegas de curso, colegas de trabalho e professores. Dedico a todos os que me incentivaram, me cobraram, me auxiliaram e me instigaram de uma forma ou de outra. Dedico também a todos os colegas do grupo de pesquisa de BPM que dedicaram um tempo para colaborar na minha pesquisa e apresentação.

Dedico especialmente este trabalho à minha família, que conhece meus problemas, fraquezas, anseios, medos e defeitos, e mesmo assim nunca deixou de transmitir amor incondicional, e que mesmo à distância me compreendem e me incentivam a cada passo que dou.

RESUMO

A falta da informação de quais são os recursos de infraestrutura necessários para implementar um processo de negócio de uma organização pode causar uma ruptura na ordem das fases do ciclo de vida de BPM, mais especificamente um retorno da fase de Implementação do Processo à fase de Redesenho do Processo. Essa ruptura se dá quando a organização não possui os recursos necessários para implementar um processo de negócio que atingiu a fase de Implementação do Processo. A proposta do presente trabalho é uma abordagem para recomendação dos recursos de infraestrutura necessários para implementar processos de negócio baseando-se no modelo do processo produzido pelo analista de processo, especificamente nos rótulos das tarefas do processo. Tal recomendação é proposta em duas fases: a classificação, onde são identificados os tipos de recurso necessários para implementar as tarefas do processo, e a segunda, onde propõe-se a recomendação dos recursos, a partir dos tipos de recurso identificados, mais similares ao perfil organizacional. A recomendação visa, portanto, possibilitar que o analista e o dono do processo fiquem cientes, ainda na fase de Redesenho do Processo, de quais são os recursos necessários para implementar o processo, assim possibilitando uma tomada de decisão (como remodelagem, cancelamento do projeto, aquisição de recursos etc) antes da fase de Implementação do Processo. No trabalho ainda é apresentada a implementação da abordagem proposta, possibilitando validar a abordagem. A recomendação foi validada para um conjunto de processos de uma universidade pública federal, e apresentou uma acurácia de 83%. O trabalho ainda apresenta, como trabalhos relacionados, um Revisão Sistemática da Literatura relacionada à recomendações de recursos para o Gerenciamento de Processos de Negócio.

Palavras-chave: Gerenciamento de Processos de Negócio. Sistemas de recomendação. Aprendizagem de Máquina. Recomendação de recursos. Revisão Sistemática da Literatura.

Infrastructure Resource Recommendation for Process Implementation Using Process Model Labels

ABSTRACT

The lack of information regarding the needed infrastructure resources in order to implement an organization's business process can lead to a break in the order of the BPM life cycle phases, specifically a return from Process Implementation phase to Process Redesign Phase. This break occurs when the organization does not have the resources needed to implement a business process that has reached the Process Implementation phase. This work proposes an approach for recommending the infrastructure resources needed to implement business processes based on the process model produced by the process analyst using task labels. This recommendation is proposed in two phases: classification, where the types of resources needed to implement the tasks of the process are identified, and the second, the recommendation of the resources which are most similar to the organizational profile. The recommendation aims to give the information to the process analyst and to the process owner to be aware, during the Process Redesign phase, of the necessary resources to implement the process, thus enabling decision making (such as remodeling, cancellation of the project, acquisition of resources etc) before the Process Implementation phase. The implementation of the proposed approach is also presented in order to validate it. The recommendation was validated for a set of processes of a federal public university using cross-fold validation, and reached 83% of accuracy. The recommendation was The paper also presents a Systematic Literature Review related to recommendation of resources for Business Processes Management.

Keywords: Business Process Management, Recommender System, Machine Learning, Systematic Literature Review, Software Engineering.

LISTA DE FIGURAS

Figura 2.1	Ciclo de vida de processos (DUMAS et al., 2013).....	17
Figura 2.2	Representação do elemento Tarefa do tipo serviço	21
Figura 2.3	Representação do elemento Tarefa do tipo usuário	21
Figura 2.4	Representação do elemento Tarefa do tipo envio	21
Figura 2.5	Representação do elemento Tarefa do tipo recebimento	21
Figura 2.6	Representação do elemento Tarefa do tipo manual	22
Figura 2.7	Representação do elemento Tarefa do tipo regra de negócio	22
Figura 2.8	Representação do elemento Tarefa do tipo script	22
Figura 2.9	Representação do elemento Tarefa abstrata.....	22
Figura 3.1	Exemplo de formação de hiperplano para duas classes.....	27
Figura 6.1	Exemplo de XML Schema para de Tarefa do tipo Script.....	51
Figura 6.2	Exemplo de XML de uma tarefa do tipo Script.....	51
Figura 6.3	Fluxograma do algoritmo de radicalização RSLP, adaptado de (HUYCK; ORENGO, 2001)	54
Figura 6.4	Aplicação WEKA	57
Figura 6.5	Exemplo de entrada de dados para a aplicação WEKA	58
Figura 7.1	Exemplo de cadastro de recursos na aplicação.....	63
Figura 7.2	Exemplo de cadastro de organização na aplicação.....	63
Figura 7.3	Exemplo de importação de processo na aplicação	63
Figura 7.4	Diagrama UML das classes da aplicação	65
Figura 7.5	Exemplo da exibição da classificação e recomendação na aplicação.....	65
Figura 7.6	Exemplo da exibição da classificação na aplicação.....	66

LISTA DE TABELAS

Tabela 4.1	Palavras-chave da busca	32
Tabela 4.2	Seleção de publicações	34
Tabela 4.3	Publicações selecionadas	35
Tabela 4.4	Algoritmos de recomendação utilizados pelas publicações	37
Tabela 4.5	O termo recurso e sua utilização	38
Tabela 4.6	Características relacionadas ao recurso	39
Tabela 4.7	Características relacionadas à tarefa.....	39
Tabela 4.8	Características relacionadas à tarefa e ao recurso	40
Tabela 4.9	Características relacionadas ao recurso de informação	40
Tabela 4.10	Características relacionadas ao processo.....	40
Tabela 6.1	Conjunto de processos para teste dos classificadores.....	56
Tabela 6.2	Comparação da acurácia dos algoritmos	58
Tabela 7.1	Exemplo de valores de atributos para os recursos	66
Tabela 7.2	Tipos de recursos utilizados com exemplos de recursos	67

SUMÁRIO

1 INTRODUÇÃO	11
1.1 Objetivos do trabalho	12
1.2 Contribuições do trabalho.....	13
1.3 Organização do trabalho	14
2 GERENCIAMENTO DE PROCESSOS DE NEGÓCIO E NOTAÇÃO E MO- DELO DE PROCESSO DE NEGÓCIO	16
2.1 Gerenciamento de Processos de Negócio	16
2.2 Notação e Modelo de Processo de Negócio	20
2.3 Definição de elementos BPMN	20
2.4 Resumo.....	23
3 APRENDIZAGEM DE MÁQUINA E RECOMENDAÇÃO	24
3.1 Fundamentos de Aprendizagem de Máquina.....	24
3.2 Classificação de texto	25
3.2.1 Árvores de Decisão e o Algoritmo C4.5	25
3.2.2 Máquinas de Vetores de Suporte.....	26
3.2.3 Naïve Bayes	27
3.3 Validação cruzada k-fold e acurácia	28
3.4 Sistemas de Recomendação.....	29
3.5 Resumo.....	30
4 TRABALHOS RELACIONADOS	31
4.1 Protocolo da Revisão Sistemática	31
4.1.1 Questões de busca	31
4.1.2 Seleção de estudos	33
4.1.2.1 Critérios de inclusão	33
4.1.2.2 Critérios de exclusão.....	33
4.1.2.3 Seleção de publicações	33
4.2 Extração de dados	35
4.2.1 Abordagens para recomendação de recursos	36
4.2.2 Comparações da utilização do termo recurso	37
4.2.3 Características utilizadas para recomendação.....	39
4.2.4 Resumo da revisão sistemática da literatura	43
5 FUNDAMENTAÇÃO DO PROBLEMA	45
5.1 Fatores de custo em BPM.....	45
5.2 Estimativa de Recursos.....	46
5.3 Ciclo de Vida de Processo.....	47
5.4 Resumo.....	48
6 RECOMENDAÇÃO DE RECURSOS PARA PROCESSOS DE NEGÓCIO	49
6.1 Classificação do tipo de recurso.....	50
6.1.1 Leitura do modelo	50
6.1.2 Normalização de rótulos de tarefas.....	51
6.1.3 Aplicação de algoritmo classificador de texto	54
6.1.4 Cálculo de acurácia	59
6.2 Recomendação de recursos	59
6.3 Resumo.....	61
7 IMPLEMENTAÇÃO DA ABORDAGEM PARA RECOMENDAÇÃO DE RECUR- SOS	62
7.1 Experimento	67
7.2 Resumo.....	69

8 CONCLUSÃO	70
8.1 Contribuições do trabalho.....	70
8.2 Limitações.....	71
8.3 Trabalhos Futuros.....	71
REFERÊNCIAS.....	73
ANEXO I - CÓDIGO PARA CLASSIFICAÇÃO DE TIPO DE RECURSOS	78

1 INTRODUÇÃO

Um processo de negócio, ou simplesmente processo, é uma coleção de eventos inter-relacionados, atividades e pontos de decisão que envolvem vários atores e objetos que, coletivamente, levam a um resultado que agrega valor para pelo menos um cliente (DUMAS et al., 2013). Neste sentido, Gerenciamento de Processos de Negócio (*Business Process Management* - BPM) é uma disciplina que visa supervisionar como o trabalho é realizado em uma organização para garantir resultados consistentes e identificar oportunidades de melhoria, como redução de custos, do tempo de execução e de taxas de erro de processo (DUMAS et al., 2013). O ciclo de vida de BPM é dividido nas seguintes fases: Identificação do Processo (cujo resultado é uma arquitetura do processo), Descoberta do Processo (cujo resultado é o modelo do processo *as-is*), Análise do Processo (onde identificam-se as fraquezas do processo e seus impactos), Redesenho do Processo (cujo resultado é o modelo de processo *to-be*), Implementação do Processo (que tem como resultado o modelo do processo executável), e Monitoramento e Controle do Processo (onde se elabora a avaliação de desempenho do processo).

Apesar de BPM ser amplamente utilizado em organizações, existem algumas considerações quanto a sua adoção. Como citado por (CONFORT, 2016), a falta de recursos e orçamento é um dos grandes problemas na área de BPM no Brasil. Além disso, um dos riscos envolvidos na adoção do BPM está na fase de implementação de processos, quando os modelos de processo resultantes não se encaixam na infraestrutura atual da organização (MUEHLEN; HO, 2006). A infraestrutura de uma organização é composta, entre outras entidades, por recursos como *softwares*, *hardwares* e pessoas. Neste contexto, os recursos de infraestrutura (RI) necessários para executar um processo deveriam ser previstos antes da fase de implementação a fim de evitar tentativas de implementar um processo sem os recursos necessários para sua implementação.

A recomendação dos RI necessários após a fase de Redesenho do Processo suprimiria a falta de informação do que é necessário para que o processo seja implementado e posteriormente executado. No contexto de recomendação, *sistemas de recomendação* são ferramentas e técnicas de software que fornecem sugestões para itens com maior probabilidade de interesse para um usuário específico (RICCI; ROKACH; SHAPIRA, 2015). Sistemas de recomendação utilizam em geral aprendizagem de máquina e mineração de dados a fim de prever a preferência de um usuário por determinado item (SATTAR; GHAZANFAR; IQBAL, 2017). O interesse na área de sistemas de recomendação é alto porque constitui-se de uma área de pesquisa com diversas aplicações práticas que ajudam os usuários a lidar com a sobrecarga de informações e fornecer recomendações personalizadas, conteúdo e serviços para os mesmos (ADOMAVI-

CIUS; TUZHILIN, 2005). A utilização de sistemas de recomendação vai desde aplicações para E-Commerce (SARWAR et al., 2000) até áreas como a saúde (SANCHEZ-BOCANEGRA et al., 2015).

Neste trabalho é apresentada uma abordagem de recomendação de recursos necessários para a execução de um processo, para que o analista de processo possua a informação dos recursos necessários no momento do redesenho do processo, evitando assim que engenheiros de sistema tentem implementar um processo sem os recursos necessários. O analista, nesse caso, deverá tomar a decisão, juntamente com o proprietário do processo, se o processo será levado adiante caso possua os recursos, se será cancelado por não possuir orçamento, ou se deverá ser redesenhado para que se adapte aos recursos da organização.

Sistemas de recomendação vêm sendo pouco utilizados em conjunto com a área de BPM, como visto mais adiante na revisão sistemática realizada neste trabalho (capítulo 4). Porém, se mostram úteis para a área uma vez que podem auxiliar em tomadas de decisão rápida, como por exemplo a continuidade de um processo ou não de acordo com as recomendações feitas para executar tal processo. Tais tomadas de decisão, porém, devem ser de responsabilidade do proprietário do processo, levando em conta outros fatores como a acurácia da recomendação e a disponibilidade de recursos na organização.

1.1 Objetivos do trabalho

O principal objetivo do trabalho é apresentar uma abordagem para minimizar o problema mencionado anteriormente - de que modelos de processo resultantes da fase de Redesenho do Processo não se encaixam na infraestrutura atual da organização - utilizando-se a recomendação de recursos de infraestrutura, especificamente de aplicações invocadas em um Sistema de Gerenciamento de Processos de Negócio (Business Process Management System - BPMS) por tarefas do modelo de processo. A recomendação é feita utilizando o processo modelado, especialmente os rótulos das tarefas do processo organizacional.

Como objetivos específicos, espera-se neste trabalho:

- A apresentação de uma revisão sistemática para identificar o estado da arte na recomendação de recursos em BPM;
- A comparação da acurácia entre métodos de classificação de texto (Árvores de Decisão, Máquinas de Vetores de Suporte e Naïve Bayes) no contexto de rótulos de processo, utilizando como treinamento somente os processos da organização.

- Definição de abordagem para a classificação das tarefas de um processo por tipo de recurso utilizado para executá-la em um BPMS, utilizando para isso o rótulo da tarefa na modelagem do processo;
- A recomendação, baseada nos tipos de recursos identificados para as tarefas de um processo, dos recursos necessários para executar uma tarefa, disponibilizando assim essa informação para que o analista e o proprietário do processo possam tomar as devidas providências;
- A realização de experimento, utilizando o desenvolvimento de uma aplicação, para comprovar a eficácia da abordagem de recomendação.

1.2 Contribuições do trabalho

A principal contribuição do trabalho é a definição de uma abordagem para recomendação de quais aplicações devem ser invocadas pelo BPMS, a partir de um modelo do processo, utilizando-se a) classificação das tarefas de acordo com o tipo de recurso necessário para realizá-la e b) recomendação de aplicações para a tarefa de acordo com o tipo de recurso identificado para a mesma.

Assim, como contribuições do trabalho, temos:

- A identificação de um possível problema entre as fases de Redesenho do Processo e Implementação do Processo, que é a tentativa de implementar um processo na organização sem possuir os recursos necessários;
- Uma abordagem que possibilita que o analista de processo possua a informação, ainda na fase de redesenho, de quais são os recursos necessários para implementar um determinado processo, prevenindo assim que o processo retorne à fase de Redesenho do Processo quando se está na fase de Implementação do Processo.
- Comparação entre métodos de classificação - Árvores de Decisão, Máquinas de Vetores de Suporte e Naïve Bayes - para classificar os tipos de recurso necessários por tarefas de processos de negócio;
- Uma revisão sistemática da literatura que possibilita identificar o estado da arte da recomendação de recursos na área de BPM.

1.3 Organização do trabalho

O restante do trabalho está organizado da seguinte forma:

- O capítulo 2 apresenta fundamentos de BPM, usos, aplicações, e motivadores para a implantação de BPM em organizações, além de discutir sobre as fases do ciclo de vida de BPM e o que acontece em cada uma delas. No capítulo ainda são apresentados os principais *stakeholders* em um processo, assim como as responsabilidades de cada um deles. São definidos e explicados, em seguida, os elementos da Notação e Modelo de Processo de Negócio (*Business Process Model and Notation* - BPMN) que serão utilizados no texto. Por fim, definem-se também os conceitos de *infraestrutura* e *recursos*, que serão utilizados no restante do texto;
- O capítulo 3 apresenta fundamentos de aprendizagem de máquina, assim como principais tipos de problemas abordados pela área (classificação e regressão). Ainda, são apresentados os tipos de *feedback* que determinam esses tipos de aprendizado (supervisionado, por reforço e não supervisionado). São apresentados, no capítulo, três algoritmos para classificação de texto que serão utilizados no restante do trabalho, além da validação cruzada que é utilizada para medir suas acurácias. Por fim, são apresentados conceitos de sistemas de recomendação, como as abordagens de filtragem colaborativa, filtragem baseada em conteúdo e filtragem híbrida;
- O capítulo 4 relata uma revisão sistemática buscando responder às três seguintes perguntas de pesquisa:
 - Quais são as principais alternativas existentes para estimativa ou recomendação de recursos na área de processos de negócio?
 - Como o termo *recurso* é utilizado em cada uma dessas abordagens?
 - Quais características as abordagens levam em conta para a recomendação?

A revisão busca expor o estado da arte da conexão entre recursos, BPM e recomendação, abordando ainda a utilização do termo recurso e o que é considerado em recomendações em BPM, além de verificar o que já foi realizado na área; É discutido, no capítulo, o protocolo utilizado para a revisão sistemática, e a partir dos artigos selecionados pela revisão são apresentadas:

- a comparação da utilização do termo recurso;
- as abordagens utilizadas para a recomendação de recursos utilizadas pelos autores;

- as características dos processos ou dos recursos utilizadas para a recomendação.
- O capítulo 5 visa aprofundar o problema que motiva o presente trabalho. São definidos os fatores de custo em BPM, sendo eles estáticos e dinâmicos. Após, é discutida a estimativa de recursos em projetos, e quais os principais impactos da falta de recursos para implementar um processo em BPM e no ciclo de vida de BPM. O capítulo visa ser introdutório para fundamentar o desenvolvimento do trabalho realizado nos capítulos subsequentes;
- No capítulo 6 é descrita a recomendação de recursos de processos propriamente dita. Logo após são apresentadas as definições iniciais necessárias para se classificar uma tarefa por tipo de recurso necessário para executá-la. São apresentados, então, os passos utilizados para a realização da classificação do tipo de recurso que é necessário para que uma tarefa possa ser executada. Subsequentemente, é apresentada uma comparação entre diferentes métodos de classificação quando aplicados em rótulos de tarefas. Por último, são mostradas as condições necessárias para a realização da recomendação dos recursos necessários para executar um processo;
- No capítulo 7 é apresentado um experimento da abordagem utilizando processos de um departamento de uma universidade pública e federal brasileira. São apresentadas as ferramentas e bibliotecas utilizadas para o desenvolvimento da aplicação desenvolvida para o experimento, como a linguagem de programação Python e o framework Django, além de uma descrição da aplicação em si, com pseudo-códigos utilizados e imagens de telas da aplicação desenvolvida. São discutidos, ainda, a metodologia para a realização do experimento e, em seguida, os resultados do experimento;
- Finalmente, o capítulo 8 apresenta as conclusões relacionadas ao trabalho realizado, resume as principais atividades desenvolvidas na pesquisa, comenta os resultados obtidos no experimento, e ainda analisa possíveis trabalhos futuros oriundos da pesquisa realizada neste trabalho.

2 GERENCIAMENTO DE PROCESSOS DE NEGÓCIO E NOTAÇÃO E MODELO DE PROCESSO DE NEGÓCIO

Neste capítulo são apresentados, na seção 2.1, conceitos relacionados a BPM e ao ciclo de vida de BPM. Além disso, é apresentada uma descrição, na seção 2.2, da BPMN, sua utilização, e apresentação dos elementos da BPMN que são utilizados ao longo do trabalho.

2.1 Gerenciamento de Processos de Negócio

Organizações podem, a partir da modelagem de processos e automação, documentar, otimizar e controlar melhor seus processos de negócio Utilizando BPM. Ainda, como disciplina gerencial, o BPM exige uma visão organizacional completa e abrangente e um bom senso comum - sendo ambos muitas vezes escassos (MOHAPATRA, 2013). Mohapatra (2013) ainda menciona alguns motivadores para a implantação de BPM em uma organização, como:

- Falta de padrões de processo;
- Problemas de treinamento para empregados novos;
- Redução de custos;
- Eliminação gradual de antigos sistemas de aplicação;
- Implantação de uma nova arquitetura.

Em relação ao ciclo de vida de BPM, neste trabalho são consideradas as fases descritas em (DUMAS et al., 2013), que são mostradas na Figura 2.1:

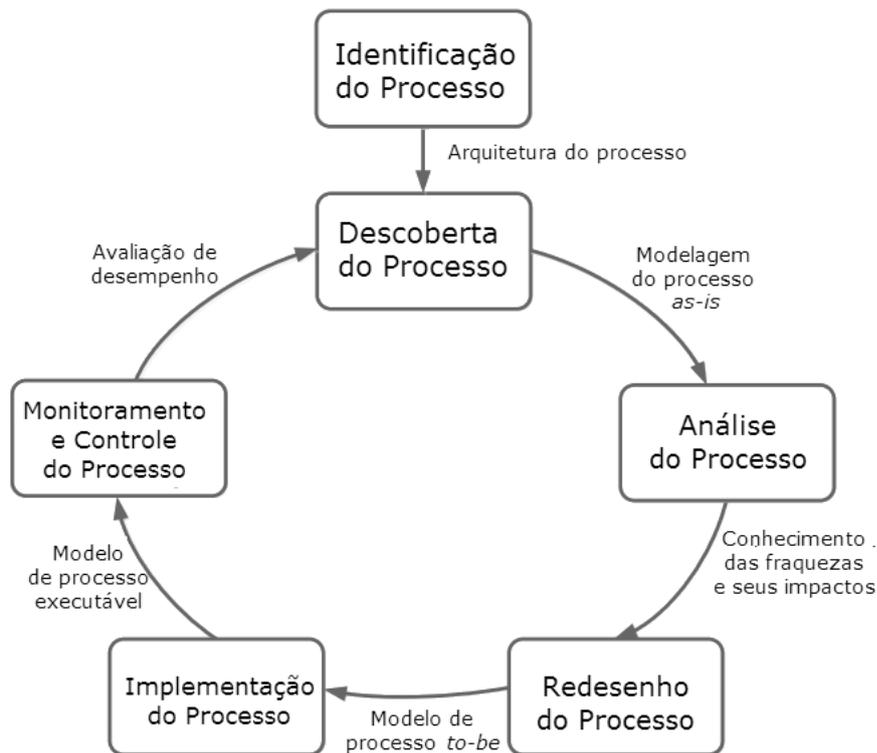


Figura 2.1 – Ciclo de vida de processos (DUMAS et al., 2013)

- *Identificação do processo*: Um problema (ou um objetivo) de negócio é definido, e os processos relativos ao problema de negócio definido são identificados, delimitados e relacionados. O resultado típico dessa fase do ciclo de vida é uma arquitetura de processos existentes na organização que, em geral, assume a forma de uma coleção de processos relacionados entre si;
- *Descoberta do processo* (também chamada de modelagem do processo *as-is*): A partir do momento em que os processos foram identificados é necessário entender o processo de negócio em detalhes. Para isso, o estado atual dos processos é documentado em sua forma *as-is*. Esta fase deve refletir o que as pessoas da organização entendem sobre como o negócio funciona. Entre as técnicas mais utilizadas para elicitar requisitos a fim de modelar um processo pode-se mencionar: *brainstorming*, análise de documentos e entrevistas. Como notação, comumente utiliza-se a BPMN, aprofundada na seção 2.2. Tal notação é utilizada em grande parte dos softwares de modelagem de processos, como Bizagi e Bonita (DUMAS et al., 2013).
- *Análise do processo*: Após a descoberta do processo, é necessário identificar, documentar e analisar problemas relacionados ao processo *as-is*. Algumas informações devem ser coletadas nessa fase, como quantidade de tempo no qual os participantes estão realmente trabalhando, tempo ocioso dos participantes do processo, tempo de retrabalho (tarefas

repetidas por algum problema no processo). Um analista de processo pode, então, aplicar diversas técnicas para identificar alternativas de modificar o processo a fim de reduzir o tempo de seu ciclo. Os problemas resultantes dessa fase são priorizados em termos de impacto e esforço para resolvê-los;

- *Redesenho do processo* (também chamada de melhoria do processo): O objetivo dessa fase é identificar mudanças no processo que auxilia a resolver os problemas identificados na fase anterior. É importante salientar que analista de processos devem considerar múltiplas opções possíveis para corrigir um problema, afinal uma mudança no processo para corrigir um problema pode trazer outros problemas. O resultado dessa fase do ciclo de vida é o processo *to-be*.
- *Implementação do processo*: Assim que redesenhado, é necessário adaptar a infraestrutura com os recursos necessários a fim de que o processo *to-be* possa ser colocado em execução. Além de preparo do ambiente ainda é necessário o treinamento de participantes do processo a fim de que eles operem de acordo com o processo redesenhado e façam o melhor uso dos recursos disponíveis;
- *Controle e monitoramento do processo*: Uma vez que o processo esteja sendo executado, é provável que ajustes sejam necessários para corrigir eventuais problemas. Para isso, existe a fase de monitoramento e controle, onde dados relevantes são coletados e analisados a fim de determinar ajustes para melhor controle de execução do processo. Esta fase ainda é útil para evitar a degradação de um processo ao longo do tempo.

Ao longo do ciclo de vida de um processo existem diversos *stakeholders* envolvidos. Os principais *stakeholders* em um processo, segundo Dumas e colaboradores (2013), são identificados da seguinte forma:

- *Equipe de gerenciamento* é responsável por supervisionar todos os processos, tomar iniciativas de redesenho de processos e fornecer recursos e orientação estratégica aos envolvidos em todas as fases do ciclo de vida do processo de negócio.
- O *proprietário* do processo é responsável pela eficiente e eficaz operação de um determinado processo. Assim, o proprietário do processo é envolvido na modelagem, análise, redesenho, implementação e monitoramento de processos. O proprietário de um processo é responsável, por um lado, pelo planejamento e organização e, por outro lado, por monitorar e controlar o processo. Em sua função de planejamento e organização, o proprietário do processo é responsável por definir métricas e metas de desempenho, além de iniciar e liderar projetos de melhoria relacionados ao processo.

- *Participantes do processo* são atores humanos que executam as tarefas de um processo de negócio em seu dia-a-dia, de acordo com as normas e diretrizes da empresa. São coordenados pelo proprietário do processo, que é responsável por lidar com aspectos incomuns do processo. Participantes do processo também estão envolvidos como especialistas de domínio durante a descoberta e análise de processos, e apoiam atividades de redesenho e a implementação.
- *Analistas de processo* conduzem identificação de processo, descoberta (em particular modelagem), análise e redesenho de tarefas (DUMAS et al., 2013). Eles também coordenam a implementação de processos, bem como monitoramento e controle de processos. Eles se reportam aos proprietários do processo e interagem de perto com os participantes do processo. Um analista de processo normalmente tem um de dois tipos de conhecimento: analistas de processo preocupados com requisitos organizacionais, desempenho, e o gerenciamento de mudanças tem conhecimento de negócios e, enquanto isso, analistas de processo preocupados com a automação de processos têm conhecimento em TI. Em geral, analistas de processo são familiares com notação para modelagem de processos, porém possuem um limitado conhecimento do processo concreto que deve ser modelado - o qual é fornecido por especialistas do domínio.
- *Engenheiros de sistema* estão envolvidos no redesenho de processos e implementação. Eles interagem com os analistas de processo para capturar os requisitos do sistema. Eles traduzem requisitos em um design de sistema e são responsável pela implementação, teste e implantação deste sistema. Os engenheiros de sistema também fazem a ligação com o proprietário e participantes do processo para garantir que o sistema desenvolvido suporte seu trabalho de maneira efetiva. Muitas vezes, a implementação do sistema, teste e implantação são terceirizados, caso em que a equipe de engenheiros de sistema será constituída, pelo menos parcialmente, por contratados.

A fim de implementar um processo em uma organização, torna-se necessário a existência de entidades que realizem as tarefas de um processo ou que auxiliem nas tarefas de um processo. O termo *recurso* é amplamente utilizado na literatura de BPM, porém sua utilização varia de acordo com o autor. Em geral, autores utilizam do termo recurso para definir recursos humanos (ARIAS et al., 2016), tecnológicos (WOITSCH; UTZ, 2015) ou de informação (ŠTAJNER; MLADENIĆ; GROBELNIK, 2012), sendo que alguns autores ainda utilizam o termo recurso como abrangendo tanto entidades humanas quanto tecnológicas, como em KaraGiannis, 2013.

Neste trabalho recurso é utilizado no contexto tecnológico, referenciando particularmente aplicações de *software* invocadas por um BPMS com base nas tarefas do processo.

2.2 Notação e Modelo de Processo de Negócio

Considerando notações para modelagem de processos, a notação gráfica de BPMN é reconhecida como padrão pela *Object Management Group* (OMG) (RECKER, 2010), sendo BPMN um padrão ISO¹. Segundo White (2014), o objetivo principal da criação de BPMN era fornecer uma notação que fosse compreensível por todos os *stakeholders*, desde os analistas de negócios que criam os rascunhos iniciais dos processos, até os desenvolvedores técnicos responsáveis pela implementação da tecnologia que executa esses processos, e finalmente, para os responsáveis de negócio que gerenciam e monitoram esses processos.

Áreas como análise de negócios e requisitos, documentação, simulação e execução de processos utilizam regularmente a notação BPMN (MATEJA et al., 2015), afinal a notação gráfica de BPMN possibilita a representação dos mais diferentes aspectos de um processo. Existem diversas ferramentas comerciais que podem ser utilizadas para modelagem de processos (por exemplo *Bizagi*, *Signavio* e *Bonita*).

2.3 Definição de elementos BPMN

A BPMN inclui um amplo conjunto de elementos notacionais. Neste trabalho, o elemento notacional denominado tarefa é particularmente relevante, pois seu rótulo será utilizado para a recomendação de RI. O elemento *tarefa* é utilizado para representar uma ação atômica dentro de um processo. Possui forma retangular com bordas arredondadas, e possui um rótulo identificador, comumente representando uma ação a ser realizada por um participante do processo específico. Um exemplo de rótulo é “Cadastrar dados do aluno” na Figura 2.3.

Existem diversos tipos de tarefas com o propósito de separar os tipos de comportamento que uma tarefa pode representar. Uma tarefa pode ou não ser estendida com o símbolo identificador de tipo no topo superior esquerdo, sendo que tarefas que não o possuem são *tarefas abstratas*. Os tipos utilizados na BPMN são os seguintes:

- Tarefa de serviço (Figura 2.2): uma tarefa de serviço utiliza de algum tipo de serviço (como uma aplicação automática ou um *web service*). É identificada por possuir o símbolo similar à uma engrenagem no topo superior esquerdo.

¹<https://www.iso.org/standard/62652.html>

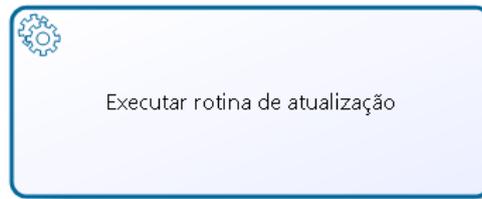


Figura 2.2 – Representação do elemento Tarefa do tipo serviço

- Tarefa de usuário (Figura 2.3): tarefa do tipo usuário identifica que o executante é um humano, com o auxílio de uma aplicação de software. Este tipo de tarefa é identificado com o símbolo de uma pessoa.

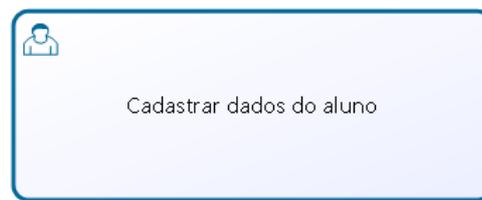


Figura 2.3 – Representação do elemento Tarefa do tipo usuário

- Tarefa de envio (Figura 2.4): a tarefa de envio é especificada para um envio de mensagem a algum participante externo (relativo ao processo). É identificada com o símbolo de um envelope preenchido.

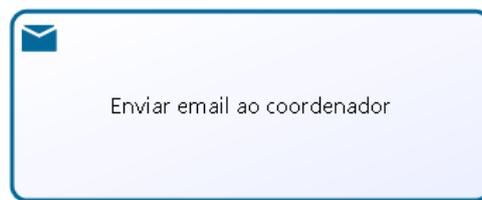


Figura 2.4 – Representação do elemento Tarefa do tipo envio

- Tarefa de recebimento (Figura 2.5): a tarefa de recebimento é especificada para a espera de uma mensagem de algum participante externo (relativo ao processo). Análogamente à tarefa de envio, é identificada com o símbolo de um envelope, porém não preenchido.

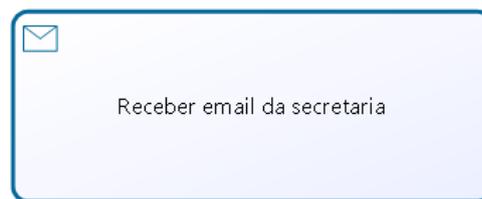


Figura 2.5 – Representação do elemento Tarefa do tipo recebimento

- Tarefa manual (Figura 2.6): uma tarefa manual representa uma tarefa que deve ser executada sem o auxílio de aplicações ou de BPMS. É identificada com o símbolo de uma mão aberta.



Figura 2.6 – Representação do elemento Tarefa do tipo manual

- Tarefa de regra de negócio (Figura 2.7): uma tarefa de regra de negócio representa um mecanismo para o processo fornecer entrada a motores de regras de negócio e esperar o resultado de cálculos. Este tipo de tarefa é identificado com um símbolo similar à uma tabela.

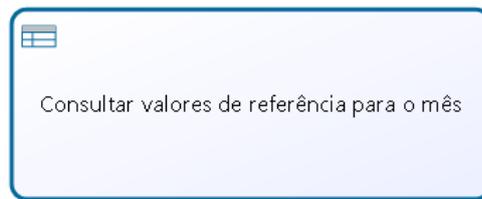


Figura 2.7 – Representação do elemento Tarefa do tipo regra de negócio

- Tarefa de *script* (Figura 2.8): Uma tarefa de script é executada por um BPMS. O símbolo identificador de tarefa de *script* é similar a uma folha pautada sinuosa.

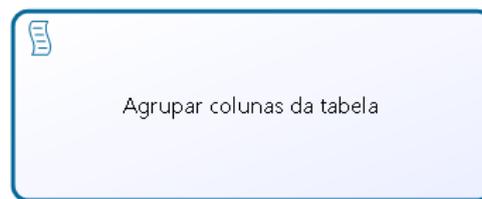


Figura 2.8 – Representação do elemento Tarefa do tipo script

- Tarefa abstrata (Figura 2.9): Tarefas abstratas não possuem símbolo identificador e representam tarefas que não possuem um tipo definido.



Figura 2.9 – Representação do elemento Tarefa abstrata

2.4 Resumo

Neste capítulo, portanto, foram apresentados os conceitos de BPM e ciclo de vida de BPM, que possui as fases de identificação do processo, descoberta do processo, análise do processo, redesenho do processo, implementação do processo e controle e monitoramento do processo. Também foi elaborada uma descrição dos *stakeholders* de um processo, que são a equipe de gerenciamento, o proprietário do processo, os participantes do processo, os analistas de processo e os engenheiros de sistema. Além disso foi apresentada a BPMN e os elementos notacionais que são utilizados ao longo do trabalho, que são as variantes do elemento *tarefa*.

3 APRENDIZAGEM DE MÁQUINA E RECOMENDAÇÃO

A proposta desse capítulo é explicar, na seção 3.1, alguns dos principais conceitos de aprendizagem de máquina (como tipos de problemas e tipos de *feedback*), e introduzir, na seção 3.2.3 o algoritmo *Naïve Bayes*, utilizado neste trabalho.

3.1 Fundamentos de Aprendizagem de Máquina

Nas últimas duas décadas, Aprendizado de Máquina (AM) tornou-se um dos pilares da tecnologia da informação e, com isso, uma parte central, embora usualmente oculta, de nossa vida (SMOLA; S.V.N.VISHWANATHAN, 2008). De certa forma, a análise de dados se tornou estrategicamente importante dentro de organizações e, com isso, o surgimento de algoritmos para processar tais dados tem sido essencial na vida de cientistas de dados, analistas, etc.

Os dados a serem tratados com o auxílio de AM podem assumir diferentes formas, entre elas listas, vetores, grafos e strings. Em alguns casos, ainda, os dados são compostos por várias dessas formas (o que chamamos de estruturas compostas). Levando em consideração que a quantidade de problemas de aprendizado é muito grande, alguns tipos de problema são mais comumente vistos: classificação binária, classificação multiclases e regressão.

- **Classificação binária:** em sua mais simples forma pode ser entendido como a classificação, entre duas classes, de elementos pertencentes a um dado conjunto de dados. Exemplo: classificar um documento como sendo útil ou não para a empresa;
- **Classificação multiclasse:** a classificação multiclasse é basicamente uma extensão lógica da classificação binária. Ao invés de classificar elementos entre duas classes, o número de classes não é mais binário. Exemplo: classificar um documento como sendo de um ds seguintes idiomas: português, espanhol, inglês e japonês. Repare que classificação envolve classes discretas;
- **Regressão:** o propósito da regressão é estimar um uma variável y de valor real dado um padrão x . Por exemplo, pode-se utilizar regressão para estimar o valor de uma ação no dia seguinte.

Um agente é tudo o que pode ser considerado capaz de perceber seu ambiente por meio de sensores e agir sobre esse ambiente por intermédio de atuadores (RUSSELL; NORVIG, 2009).

Existem três tipos de *feedback* que determinam os três tipos de aprendizado: supervisi-

onado, por reforço e não supervisionado, sendo que:

- Supervisionado: o agente observa exemplos de pares de entrada e saída e aprende uma função que faz o mapeamento de entrada para saída.
- Não supervisionado: o agente aprende padrões na entrada mesmo quando o nenhum feedback explícito é fornecido ao agente.
- Por reforço: o agente aprende por uma série de reforços (recompensas ou punições). Por exemplo, os dois estados de vitória no final de uma partida de xadrez informam ao agente que ele fez algo certo. A partir disso fica a critério do agente decidir quais ações de reforço tiveram mais impacto para alcançar objetivos.

3.2 Classificação de texto

O problema de classificação de texto tem sido amplamente estudado em comunidades de mineração de dados, aprendizagem de máquina, bancos de dados e recuperação de informação em um grande número de domínios como diagnóstico médico, organização de documentos e mercados-alvo (AGGARWAL; ZHAI, 2012). Para realização de classificação de texto de forma supervisionada, utiliza-se um conjunto de dados para treinar um modelo de classificação que é, então, usado para prever a classe de uma determinada instância cuja classe é desconhecida. A frequência de uma palavra específica, por exemplo, assume papel importante na classificação. Alguns métodos são comumente utilizados para classificação de texto, como por exemplo os seguintes (AGGARWAL; ZHAI, 2012):

- Árvores de Decisão;
- Máquinas de Vetores de Suporte;
- Naïve Bayes.

3.2.1 Árvores de Decisão e o Algoritmo C4.5

Como o nome já sugere, uma árvore de decisão é uma partição no estilo de árvore no espaço dos dados, na qual o particionamento se obtém com uma série de condições de subdivisões (decisões) nos atributos. A ideia é particionar o espaço de dados em fatias de valores de atributos que são fortemente inclinadas para uma determinada classe durante a fase de treinamento. Portanto, as partições são associadas a seus rótulos de classe *favoritos*. Para isso,

a ideia básica é escolher as condições de divisão de tal forma que as subdivisões são dominadas por uma ou mais classes (AGGARWAL, 2018). C4.5, nesse contexto, (QUINLAN, 1993) é um algoritmo que possibilita a criação de árvores de decisão a partir de dados de treinamento. O C4.5 e seu antecessor, ID3, usam fórmulas baseadas na teoria da informação para avaliar qual o melhor teste a ser utilizado em um nodo específico de uma árvore de decisão; em particular, eles escolhem o teste que tem o maior ganho de informação de um conjunto de casos, dada a restrição de que apenas um atributo será testado.

C4.5 usa uma técnica de poda baseada em estimativas estatísticas confiáveis, permitindo assim que todos os dados rotulados disponíveis sejam usados para treinamento. A base para a poda estatística é o cálculo de um intervalo de confiança para a taxa de erro. Para decidir se substitui um nó que é próximo à folha e seus filhos por um único nó folha, C4.5 compara os limites superiores dos intervalos de confiança de erro para as duas árvores. Para a árvore onde não há a poda, a estimativa de erro superior é calculada como uma média ponderada sobre suas folhas-filho. Qualquer árvore que tenha o limite superior estimado mais baixo para taxa de erro é selecionada (SALZBERG, 1994).

3.2.2 Máquinas de Vetores de Suporte

Uma máquina de vetor de suporte (SVM) é, em sua essência, um hiperplano que separa diferentes classes, onde o objetivo é maximizar a margem de separação, sendo este um problema de otimização (CORTES; VAPNIK, 1995). Na Figura 3.1 é exibido um exemplo de SVM. Os pontos com as menores margens são exatamente os mais próximos do limite de decisão (representado pela linha sólida); na Figura, esses são os três pontos que estão nas linhas tracejadas paralelas ao limite de decisão. Esses três pontos são chamados de vetores de suporte nesse problema.

A formulação mais simples do SVM é a linear, onde o hiperplano se encontra no espaço dos dados de entrada (EVGENIOU; PONTIL, 2001). Neste caso, o espaço de hipóteses é um subconjunto de todos os hiperplanos da forma

$$f(x) = wx + b$$

Existe o caso onde os dados não são linearmente separáveis, o que sugere que seja encontrada uma função que mapeie os pontos para um espaço dimensional maior onde eles sejam linearmente separáveis, fazendo a classificação nesse novo espaço dimensional.

O fato de que as SVMs não perdem sua eficiência ou sua capacidade de generalizar pelo crescimento da entrada, torna-se possível aplicá-las à tarefa de classificação de documentos, usando todas as palavras de um documento diretamente como atributos (PAASS; KINDERMANN; LEOPOLD, 2004).

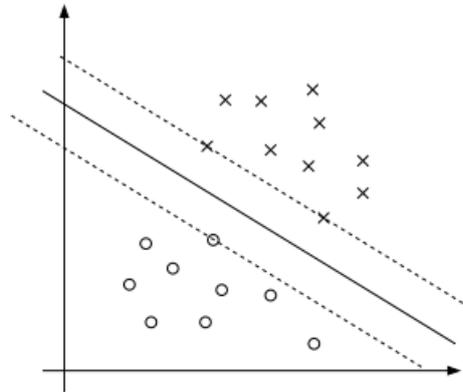


Figura 3.1 – Exemplo de formação de hiperplano para duas classes

3.2.3 Naïve Bayes

Naïve Bayes (NB) é um algoritmo simples, porém bastante preciso que é comumente utilizado para classificação de texto.

Para classificar um determinado documento (texto composto por um conjunto de palavras) pertencente a uma coleção de documentos N , o método NB estima a probabilidade a posteriori das classes utilizando a regra de Bayes:

$$P(c_j | d_i) = \frac{P(d_i | c_j) P(c_j)}{P(d_i)}$$

, onde

- $P(c_j)$ é a probabilidade a priori de que um documento aleatório em N pertença à classe c_j ;
- $P(d_i | c_j)$ é a probabilidade de que um documento escolhido aleatoriamente entre os documentos pertencentes à classe c_j seja o documento d_i ;
- $P(d_i)$ é a probabilidade de que um documento escolhido aleatoriamente em N seja o documento d_i .

A partir disso, o documento é associado à classe com a maior probabilidade. Um detalhe importante é que no algoritmo Naïve Bayes assume-se que as palavras de um determinado

documento são mutuamente independentes umas das outras e a probabilidade de ocorrência de uma palavra no documento é independente de sua posição no documento. Como NB é um algoritmo supervisionado (ou seja, devem ser passadas instâncias já classificadas para treinar o algoritmo), necessita de documentos de treinamento para realizar a classificação. A qualidade dos documentos usados para treinar o algoritmo é de extrema importância para a eficácia do mesmo.

3.3 Validação cruzada k-fold e acurácia

A ideia principal da validação cruzada utilizando a acurácia é estimar a acurácia preditiva de um modelo e descobrir se o modelo generaliza bem. Na validação cruzada k-fold, cada documento cumpre os dois papéis - como dado de treinamento e dado de teste. Primeiro, divide-se os dados em k subconjuntos iguais. Em seguida, são executadas k rodadas de aprendizado; em cada rodada, 1 / k dos documentos são mantidos como dado de testes e os documentos restantes são usados como dado de treinamento. Os valores populares para k são 5 e 10 - o suficiente para fornecer uma estimativa estatisticamente provável de ser precisa, com um custo de 5 a 10 vezes mais tempo de computação (RUSSELL; NORVIG, 2009). O cálculo de acurácia, que é feito em cada rodada de execução da validação cruzada, é definido da seguinte forma:

$$\text{Acurácia} = \frac{\text{Número de predições corretas}}{\text{Número total de predições}}$$

No final das k rodadas de aprendizado calcula-se a média das acurácias derivadas de cada uma das rodadas para, assim, se obter a acurácia do modelo.

A acurácia, porém, não é recomendada quando existe um desbalanceamento entre as classes. Por exemplo, em um problema onde uma classe A contém 98% das amostras, um classificador pode facilmente conseguir 98% de precisão classificando todas as amostras como pertencentes à classe A e, apesar da alta acurácia, o classificador não é útil caso o objetivo seja a detecção de eventos raros (CASTRO; BRAGA, 2011). Portanto, existem outras medidas que também podem ser úteis a fim de avaliar o classificador, como o caso da revocação (utilizada para verificar a quantidade de instâncias pertencentes a uma classe que foram descobertas pelo classificador) e precisão (utilizada para verificar a quantidade de instâncias que foram classificadas para uma classe e realmente pertencem a tal classe).

3.4 Sistemas de Recomendação

Sistemas de recomendação, ao fornecer recomendações de conteúdo, itens e serviços personalizados, lidam com o problema de sobrecarga de informações que usuários normalmente encontram.

Existem várias abordagens para a construção de sistemas de recomendação que podem utilizar filtragem colaborativa, filtragem baseada em conteúdo ou filtragem híbrida.

- **Filtragem colaborativa:** A filtragem colaborativa é uma técnica de previsão independente de domínio para conteúdo que não pode ser fácil e adequadamente descrito por metadados. A técnica de filtragem colaborativa funciona criando um banco de dados (matriz de itens do usuário) de preferências para itens por usuários. Em seguida, ele associa os usuários a interesses e preferências relevantes, calculando as semelhanças entre seus perfis para fazer recomendações. Esses usuários constroem um grupo chamado vizinhança. Um usuário recebe recomendações para os itens que ele não avaliou antes, mas que já foram avaliados positivamente pelos usuários em sua vizinhança. As recomendações que são produzidas pela filtragem colaborativa podem ser de previsão ou recomendação. A previsão é um valor numérico, R_{ij} , expressando a pontuação prevista do item j para o usuário i , enquanto recomendação é uma lista dos principais N itens que o usuário mais gosta (ISINKAYE; FOLAJIMI; OJOKOH, 2015).
- **Filtragem baseada em conteúdo:** enfatiza a análise dos atributos do conjunto de itens para fazer as recomendações. A recomendação é feita com base no perfil do usuário usando atributos extraídos do conteúdo dos itens que o usuário avaliou no passado. A principal desvantagem desta técnica é a necessidade de ter o registro das características dos itens a serem recomendados.
- **Filtragem híbrida:** A técnica de filtragem híbrida combina diferentes técnicas de recomendação para obter uma melhor otimização a fim de evitar algumas limitações e problemas de sistemas puros de recomendação (ADOMAVICIUS; ZHANG, 2012).

No contexto de sistemas de recomendação com filtragem baseada em conteúdo, são recomendados, portanto, itens semelhantes àqueles que o usuário selecionou no passado. Um perfil de usuário contém informações sobre os gostos, preferências e necessidades do usuário que podem ser extraídos em questionários para usuários ou aprendidos com o comportamento do mesmo ao longo do tempo, enquanto um perfil de item contém um conjunto de atributos de itens. Em seguida, o sistema calcula a similaridade entre o perfil do usuário e o perfil de todos

os itens e recomenda os itens que satisfazem as necessidades ou gostos do usuário (LI; CAI; LIAO, 2012).

Um sistema de recomendação tipicamente utiliza dois tipos de entrega de informação ao fornecer recomendações ao usuário:

- O sistema escolhe objetos semelhantes àqueles para os quais o usuário já expressou uma preferência;
- O sistema escolhe objetos em que o perfil do usuário não fornece evidências para prever a reação do usuário.

3.5 Resumo

Neste capítulo foram introduzidos os problemas de aprendizagem de máquina mais comuns, que são classificação binária, classificação multiclasse e regressão. Além disso foram definidos os termos agente e os tipos de *feedback*, que são supervisionado, não supervisionado e por reforço. Foram ainda abordados três tipos de algoritmos para classificação de texto, sendo eles Árvores de Decisão, Máquinas de Vetores de Suporte e Naïve Bayes, sendo ainda discutida a validação cruzada para estimativa de acurácia de modelos. Por último, foram apresentadas as abordagens, para recomendação, de filtragem colaborativa, filtragem baseada em conteúdo e filtragem híbrida.

4 TRABALHOS RELACIONADOS

RSLs são conduzidas com o objetivo de identificar, analisar e interpretar as evidências disponíveis relacionadas a uma questão de pesquisa específica (KITCHENHAM; CHARTERS, 2007).

RSL é uma técnica científica eficiente. Embora às vezes árduo e demorado, uma revisão é geralmente mais rápida e menos onerosa do que embarcar em um novo estudo. Tão importante quanto isso, uma revisão pode evitar seguir um caminho já explorado (MULROW, 1994).

Para este trabalho, foi realizada uma revisão sistemática da literatura (RSL) que investiga a existência de recomendação de recursos em BPM. A revisão sistemática apresentada neste trabalho tem como objetivo analisar o estado da arte da recomendação de recursos na área de BPM.

Neste capítulo, é descrita a RSL, bem como outros trabalhos relacionados. A RSL relatada visa, principalmente, responder às seguintes perguntas:

- Quais são as principais abordagens para estimativa ou recomendação de recursos em BPM?
- Como o termo *recurso* é utilizado em cada uma dessas abordagens?
- Quais características as abordagens levam em conta para a recomendação?

4.1 Protocolo da Revisão Sistemática

Como a RSL segue os padrões propostos por (KITCHENHAM; CHARTERS, 2007), esta seção visa descrever a definição do protocolo da revisão, ou seja, o planejamento de como a revisão é conduzida.

4.1.1 Questões de busca

Questões de busca são fundamentais em uma RSL dado que tais revisões convencionalmente agregam estudos referentes a uma ou várias questões. As questões utilizadas neste estudo são as seguintes:

- Quais são as principais abordagens para estimativa ou recomendação de recursos em BPM?

- Como o termo *recurso* é utilizado em cada uma dessas abordagens?
- Quais características as abordagens levam em conta para a recomendação?

A primeira questão de busca é intencionalmente abrangente o suficiente a fim de buscar por qualquer método relacionado a recomendação de recursos baseando-se em quaisquer critérios dado o contexto de BPM.

O processo de busca foi uma busca digital usando-se três bibliotecas digitais seguido de uma busca manual baseando-se nas referências dos estudos obtidos pelas bibliotecas digitais a fim de evitar que referências não indexadas nas bibliotecas digitais ficassem de fora. A busca abrangeu os seguintes tipos de publicações: anais de conferências, *journals*, artigos e capítulos de livros completos. Nenhum limite inferior de ano foi estabelecido a fim de evitar a exclusão de publicações antigas que porventura pudessem ser relevantes. A RSL foi finalizada em 2017; portanto, publicações posteriores a 31 de dezembro de 2017 não foram incluídas na revisão (nenhum outro trabalho que pudesse ser encaixado na RSL foi encontrado até setembro de 2018). O idioma das publicações foi limitado ao inglês, o que significa que publicações escritas em qualquer outro idioma não foram incluídas na RSL.

Foram consideradas as seguintes bibliotecas digitais acadêmicas para a realização da RSL:

- IEEE Xplore Digital Library;
- Springer Link
- ACM Digital Library

A Tabela 4.1 exibe as palavras-chave relacionadas à questão de busca, as quais são interrelacionadas utilizando-se operadores lógicos booleanos e utilizadas nas bibliotecas digitais a fim de gerar *strings de busca*. As palavras-chave foram escolhidas baseando-se na motivação da RSL de recomendação de recursos na área de BPM. Os plurais das palavras-chave também foram utilizados nos motores de busca das bibliotecas digitais.

Tabela 4.1 – Palavras-chave da busca

Primeira palavra-chave		Segunda palavra-chave
business process		resource estimation
OR		OR
BPM	AND	resource recommendation
		OR
		resource recommender

4.1.2 Seleção de estudos

A fim de avaliar quais publicações relacionavam-se com o propósito da busca e deveriam ser incluídas na extração de dados, foram estabelecidos critérios de exclusão e inclusão.

4.1.2.1 Critérios de inclusão

Os critérios de inclusão descrevem quais características essenciais uma dada publicação deve possuir para que seja incluída na análise de resultados da revisão sistemática. Os critérios foram definidos como segue:

- a publicação utiliza diretamente os termos recomendação ou estimativa de recursos, ao invés de apenas mencioná-los;
- a publicação é relacionada à BPM;
- a publicação é um estudo primário, ou seja, foi divulgada diretamente pelo pesquisador que realizou o estudo.

4.1.2.2 Critérios de exclusão

Os critérios de exclusão foram definidos a fim de excluir publicações consideradas ineligiáveis baseando-se na proposta da RSL atual:

- a publicação não é escrita no idioma inglês;
- a publicação não é acessível pelo revisor;
- a publicação é uma duplicata ou reporta o mesmo estudo de outra publicação (no segundo caso, a publicação menos completa é excluída).

4.1.2.3 Seleção de publicações

O primeiro critério de exclusão foi utilizado diretamente no filtro de busca das bibliotecas digitais, enquanto os critérios restantes exigiram uma análise manual dos artigos.

A fim de avaliar se as publicações resultantes das buscas nas bibliotecas digitais estavam de acordo com os critérios de inclusão e não cumpriam os critérios de exclusão, foi seguido o passo-a-passo abaixo:

1. Aplicar as *strings* de busca em cada uma das bibliotecas digitais;
2. Usando as publicações resultantes do passo anterior, aplicar os critérios de exclusão: ex-

cluir as publicações em qualquer idioma diferente de inglês, inacessíveis pelo autor e duplicatas;

3. Usando as publicações resultantes do passo anterior, excluir publicações irrelevantes à RSL baseando-se em seus títulos, resumos e conclusões;
4. Usando as publicações resultantes do passo anterior, excluir publicações irrelevantes baseando-se em seus textos completos.

A Tabela 4.2 exibe o conjunto de publicações resultante na aplicação de cada um dos passos anteriormente descritos:

Tabela 4.2 – Seleção de publicações

Passo	Número de publicações resultantes
1	138
2	128
3	26
4	7

Na Tabela 4.3 são exibidas as publicações selecionadas na revisão sistemática da literatura:

Tabela 4.3 – Publicações selecionadas

Publicação	Autor	Título	Ano
[1]	Arias M., Rojas E., Munoz-Gama J., Sepúlveda M	A Framework for Recommending Resource Allocation Based on Process Mining	2015
[2]	Zhao, W., Liu, H., Dai, W. et al.	An entropy-based clustering ensemble method to support resource allocation in business process management	2015
[3]	Sindhgatta R., Ghose A., Dam H.K.	Context-Aware Analysis of Past Process Executions to Aid Resource Allocation Decisions	2016
[4]	Štajner, T., Mladenčić, D, Grobelnik, M.	Information Resource Recommendation in Knowledge Processes	2015
[5]	Brander S. et al.	Refining Process Models through the Analysis of Informal Work Practice	2011
[6]	Cabanillas, C., García, J.M., Resinas, M., Ruiz, D., Mendling, J., Ruiz-Cortés, A	Priority-based human resource allocation in business processes	2013
[7]	Liu, T., Cheng, Y., Ni, Z.	Mining event logs to support workflow resource allocation	2012

4.2 Extração de dados

Como definido na subseção 4.1.1, as questões de busca usadas nesta RSL são as seguintes:

- Quais são as principais abordagens para estimativa ou recomendação de recursos em BPM?
- Como o termo *recurso* é utilizado em cada uma dessas abordagens?
- Quais características as abordagens levam em conta para a recomendação?

Nesta seção as questões de busca são respondidas a partir dos dados extraídos das publicações selecionadas. Na subseção 4.2.1 exploram-se as abordagens para recomendação de recursos. Na subseção 4.2.2 responde-se a pergunta sobre a utilização da palavra recursos. Na

subseção 4.2.3 são analisadas as características consideradas para a recomendação de recursos feita por cada um dos estudos. Por último, na subseção 4.2.4 é feito um resumo da revisão sistemática.

4.2.1 Abordagens para recomendação de recursos

O estudo de Arias e colaboradores (2015) utiliza BPA (algoritmo de melhor posição) (AKBARINIA; PACITTI; VALDURIEZ, 2011) para o ranqueamento de uma recomendação final. BPA é um algoritmo utilizado para resolver consultas top-k, similar ao *Threshold Algorithm*, porém provado ser menos custoso e mais rápido (AKBARINIA; PACITTI; VALDURIEZ, 2011). Consultas top-k são basicamente um conjunto ranqueado de k tuplas em um banco de dados que melhor satisfazem uma determinada condição (BRUNO; CHAUDHURI; GRAVANO, 2002).

Utiliza-se K-means, em Zhao e colaboradores (2015), em subespaços gerados a partir do espaço de atributos dos recursos, para fazer clusterização de recursos, a fim de avaliar o grau de preferência de uma determinada tarefa em cada subespaço. É utilizado, ainda, o algoritmo kNN (COVER; HART, 2006) para predição de preferências, e a recomendação é feita utilizando-se Top-k de acordo com tais preferências.

Sindhgatta e colaboradores (2016) utilizam aprendizado por árvore de decisão para prever a performance de uma potencial execução do processo, e do algoritmo kNN para auxiliar na decisão para alocar recursos às tarefas em tais instâncias, identificando instâncias passadas do processo similares à atual. A regressão kNN é utilizada para identificar condições contextuais que levam à performance ou ao resultado desejado.

São apresentadas, em Štajner e colaboradores (2015), avaliações sobre qual a probabilidade de um recurso de informação aparecer dado o histórico observado. Tal informação é utilizada para um ranqueamento de escores. Não é mencionado um método específico para fazer a recomendação.

Brander e colaboradores (2011) analisam a coocorrência entre recursos em históricos do processo. A clusterização de recursos de informação associados a uma tarefa é feita utilizando-se o algoritmo EM. É analisada a frequência com que os recursos de informações ocorrem e são recomendados os recursos mais frequentes sempre que um usuário trabalha na tarefa novamente.

Propõe-se, em Cabanillas e colaboradores (2013), um algoritmo que recebe como entrada uma preferência e um conjunto de recursos a serem ranqueados de acordo com a preferência. O algoritmo simplesmente itera sobre a lista de recursos fornecidos e adiciona uma aresta

$r \rightarrow r'$ ao grafo se r' for preferido em relação a r de acordo com a preferência fornecida e as informações na base de conhecimento.

Já em Liu e colaboradores (2012) são geradas regras de associação a partir de conjuntos de itens frequentes e, assim, são selecionadas de regras de associação com maior confiança e com suporte superior a um mínimo estipulado. A medida de confiança então é utilizada para ordenar as regras de associação para gerar uma lista de alocação de recursos para auxílio na tomada de decisão.

A Tabela 4.4 exibe a utilização de algoritmos para recomendação de recursos utilizados nas publicações selecionadas da RSL.

Tabela 4.4 – Algoritmos de recomendação utilizados pelas publicações

	[1]	[2]	[3]	[4]	[5]	[6]	[7]
k-Means		x					
kNN		x	x				
Árvores de Decisão			x				
Maximização de Expectativa (EM)					x		
Regras de associação							x

4.2.2 Comparações da utilização do termo recurso

Recurso, em Arias e colaboradores (2015), é descrito como sendo uma entidade que possui propriedades como capacidade e experiência para realizar uma determinada tarefa do processo. O estudo não menciona um tipo de recurso específico, e tampouco abrange a definição formal de recursos, procurando ser o mais abrangente possível.

Já Zhao e colaboradores (2015) seguem a linha de que recursos são executores de tarefas, utilizando a mesma definição de Xiao e Ming (2011): “*Recurso: um ator ou agente que realiza tarefas de fluxo de trabalho. Dependendo dos domínios de aplicação, os recursos podem ser recursos humanos ou recursos não humanos, como máquinas, dinheiro, software etc. Aqui, consideramos apenas os recursos duráveis, isto é, recursos alocados e liberados durante a execução, porém não os recursos criados ou destruídos. E nós assumimos que um recurso só pode trabalhar em uma tarefa ao mesmo tempo.*” Porém, os autores destacam que, neste estudo, consideram apenas recursos humanos por possuírem criatividade e habilidade de aprendizado.

Sindhgatta e colaboradores (2016) também trabalham com a ideia de recursos como sendo executores de uma tarefa de negócio (por exemplo, um empregado, um mecânico), e

não limitam a definição de recursos a um tipo de recurso específico (por exemplo, recursos humanos).

A palavra recurso, para Štajner e colaboradores (2015), é relacionada a recursos de informação como documentos, websites e mensagens de email, os quais podem ser acessados com base no conhecimento de usuários. Neste caso, portanto, um recurso é uma entidade informacional que pode ser acessada por um executor de uma tarefa a fim de ajudá-lo a realizar tal tarefa.

Brander e colaboradores (2011) utilizam o termo recursos como sendo entidades utilizadas em tarefas (como documentos e pessoas colaboradoras de tarefas): por exemplo, quando uma pessoa executa uma tarefa, precisa frequentemente consultar recursos, como documentos de texto (baseando-se em diversos fatores, como preferência). Uma tarefa pode, portanto, possuir diversos recursos associados à ela.

Apesar de priorizar recursos do tipo humano na pesquisa, o estudo de Cabanillas e colaboradores (2013) contextualiza recursos como sendo potenciais executores de tarefa (tanto serviços automáticos quanto humanos). Em tempo de execução, uma alocação de recursos considera tais potenciais executores de uma tarefa a fim de selecionar os executores reais da tarefa. Em geral, uma determinada tarefa é designada à lista de trabalho de apenas um recurso.

Em Liu e colaboradores (2012) o termo recurso é brevemente definido como um ator ou agente (tanto sistema quanto pessoa) do processo, sendo um executor ou participante de tarefas do processo.

Baseando-se na análise de cada uma das publicações selecionadas, podemos resumir o uso do termo recurso na Tabela 4.5 como segue:

Tabela 4.5 – O termo recurso e sua utilização

Publicação	Definição de recurso
[1]	Executor de tarefa
[2]	Executor de tarefa
[3]	Executor de tarefa
[4]	Informação
[5]	Informação / Colaborador de tarefas
[6]	Executor de tarefa
[7]	Executor de tarefa

Portanto, em relação à utilização do termo recurso nas publicações selecionadas na RSL, o termo é utilizado mais frequentemente no sentido de um agente que potencialmente executa

uma tarefa. Duas publicações usam o termo para representar recursos de informação, enquanto apenas uma publicação utiliza o termo para referenciar um colaborador na execução de uma tarefa.

4.2.3 Características utilizadas para recomendação

A fim de prover recomendação de recursos, os artigos selecionados utilizam características específicas tanto do processo em si quanto ao contexto do mesmo (por exemplo, a carga de trabalho de um determinado recurso). Na Tabela 4.6 são mostradas, por publicação selecionada, as características relacionadas a recursos; na Tabela 4.9, características relacionadas a tarefas; na Tabela 4.8, características relacionadas a recursos e tarefas; por último, na Tabela 4.10, características relacionadas ao processo. As características obtidas não são mutualmente exclusivas (por exemplo, a performance de um recurso na execução de uma tarefa pode fazer parte do perfil do recurso). São elas:

Tabela 4.6 – Características relacionadas ao recurso

	[1]	[2]	[3]	[4]	[5]	[6]	[7]
Perfil	x	x				x	
Meta-modelo	x					x	
Carga de trabalho	x	x					
Disponibilidade		x				x	
Cooperação		x					
Adaptabilidade		x					

Tabela 4.7 – Características relacionadas à tarefa

	[1]	[2]	[3]	[4]	[5]	[6]	[7]
Importância		x					
Urgência		x					

Tabela 4.8 – Características relacionadas à tarefa e ao recurso

	[1]	[2]	[3]	[4]	[5]	[6]	[7]
Qualidade na execução de tarefa	x	x					
Custo na execução de tarefa	x	x					
Expertise para executar tarefa	x						
Performance na execução de tarefa	x	x					
Frequência na execução de tarefa		x					

Tabela 4.9 – Características relacionadas ao recurso de informação

	[1]	[2]	[3]	[4]	[5]	[6]	[7]
Similaridade				x			
Conteúdo					x		
Remetentes e receptores					x		

Tabela 4.10 – Características relacionadas ao processo

	[1]	[2]	[3]	[4]	[5]	[6]	[7]
Histórico de execução	x	x	x	x		x	x

Perfil do recurso: O estudo de Arias e colaboradores (2015) menciona que um perfil de recurso é definido para especificar capacidades e aptidões de um recurso, sem entrar em detalhes sobre o escopo do perfil, mas utilizando históricos de execução do processo para obter os dados. O estudo de Zhao e colaboradores (2015) define perfil de recurso segundo alguns atributos do recurso: qualidade, tempo, custo, cooperação e adaptabilidade. Tais atributos são gerados a partir de análises qualitativas e quantitativas, utilizando mineração de logs de evento de execuções do processo. Algumas propriedades para definir o perfil de um recurso no estudo de Cabanillas e colaboradores (2013): dados pessoais e organizacionais (como idade, nome, posição na organização), habilidades (conhecimento em tecnologias específicas, aplicações, metodologias etc), informação profissional (salário, anos de experiência) lista de trabalho e histórico (informações sobre execuções passadas do processo, como tarefas realizadas, tempo de execução, etc).

Meta-modelo do recurso: Alguns estudos utilizam metamodelos a fim de caracterizar um recurso. Por exemplo, o estudo de Cabanillas e colaboradores (2013) utiliza SOUP - Ontologia Semântica de Preferências de Usuário (GARCÍA et al., 2013) - para definir preferências de recursos e, a partir dele, gerar um ranqueamento de recursos. Arias e colaboradores (2015)

também mencionam utilização de meta-modelos de recurso para recomendar recursos, porém não mencionam como tais meta-modelos são gerados ou utilizados.

Carga de trabalho do recurso: Arias e colaboradores (2015) consideram o número de tarefas simultâneas com as quais um recurso está trabalhando em um dado momento para a definição de carga de trabalho. Já Zhao e colaboradores (2015) definem carga de trabalho baseando-se na porcentagem de tempo de trabalho em um certo período no passado.

Disponibilidade do recurso: Para Zhao e colaboradores (2015) a disponibilidade do recurso é relacionada à ocupação do recurso em determinado momento: por exemplo, um recurso não executando uma tarefa em um tempo t é considerado disponível. Para Cabanillas e colaboradores (2013), porém, disponibilidade possui o sentido de tamanho da lista de trabalho, ou seja, quanto menor a lista de trabalho, maior a disponibilidade do recurso.

Cooperação do recurso: A cooperação de um recurso, para Zhao e colaboradores (2015), reflete sua habilidade de comunicação: recursos com alta cooperação podem transferir itens de trabalho a participantes sucessivos em curtos períodos de tempo. Em geral esse dado pode ser extraído a partir da eficiência histórica de transferência de trabalho de um dado recurso para outros recursos a partir de logs de evento.

Adaptabilidade do recurso: Adaptabilidade, para Zhao e colaboradores (2015) é utilizada para medir o quão rápido um recurso pode se ajustar a uma tarefa diferente, ou seja, recursos com alta adaptabilidade podem se adaptar a tarefas e ambientes diferentes em um curto espaço de tempo. São utilizados dois indicadores: o número de tarefas que um recurso pode executar e o tempo de adaptação.

Perfil da tarefa: O perfil da tarefa, em Arias e colaboradores (2015) é relacionado ao ponto de vista da tarefa: por exemplo, quais são os requisitos necessários para um dado recurso executar a tarefa.

Importância da tarefa: Algumas tarefas têm mais influência na performance de um processo e, portanto, faz sentido possuírem mais prioridade do que outras. Zhao e colaboradores (2015) especificam que a importância pode ser levantada a critério de especialistas do domínio.

Urgência da tarefa: Para Zhao e colaboradores (2015) a urgência de uma tarefa pode ser dada a partir do quão importante tal tarefa é no sentido de reduzir gargalos no processo e, portanto, dar urgência a algumas tarefas pode acelerar o fluxo do processo.

Qualidade da execução da tarefa: Qualidade pode ser indicada de algumas formas, como: a partir da avaliação, por parte do cliente, da execução de uma determinada tarefa por determinado recurso, como exibido em Arias e colaboradores (2015), e experiência baseando-se na frequência com que tal recurso executou a tarefa, como em Zhao e colaboradores (2015).

Custo na execução da tarefa: Em Arias e colaboradores (2015) o custo indica essencialmente o quão custosa é a execução de uma tarefa por um recurso, porém não definem-se métricas para estipular tal custo. Para Zhao e colaboradores (2015) o custo é relacionado a despesas de mão-de-obra (na maioria das vezes, estável e pouco flutuante ao longo do tempo) para a execução de certa tarefa por um recurso.

Expertise na execução da tarefa: Em Arias e colaboradores (2015) a expertise pode ser definida como a habilidade ou capacidade de um recurso executar uma determinada tarefa, baseando-se em seus conhecimentos ou competências específicos.

Performance na execução da tarefa: Arias e colaboradores (2015) definem performance como sendo a duração média que um recurso precisa para executar uma tarefa. Para Zhao e colaboradores (2015) o tempo para executar uma tarefa diminui de acordo com o aprendizado obtido pela execução da mesma múltiplas vezes. Utilizam, também a intuição de que execuções mais recentes podem definir melhor a eficiência de trabalho de um recurso do que execuções mais antigas e, portanto, devem ter um peso maior.

Frequência de execução da tarefa: Para Zhao e colaboradores (2015) a frequência é o número de vezes que uma tarefa é executada por um dado participante do processo. Por exemplo, se o mesmo processo foi executado dez vezes por um mesmo participante, a frequência é dez.

Similaridade entre recursos de informação: O estudo de Štajner e colaboradores (2015) utiliza similaridade entre recursos de informação, porém não informa como ela é obtida ou calculada.

Conteúdo do recurso de informação: Em Brander e colaboradores (2011) o conteúdo do recurso de informação é um dado extraído, por exemplo, do corpo não estruturado de um e-mail.

Remetentes e receptores do recurso de informação: Para Brander e colaboradores (2011), são pessoas relacionadas com o envio e o recebimento dos recursos de informação (no caso do e-mail, remetentes e receptores).

Histórico de execução: Históricos de execução de processos são instâncias já executadas do processo, as quais podem ser acessadas de tal forma que seja possível a extração de informações sobre o processo, como análises estatísticas do mesmo, atores que executaram cada tarefa etc.

4.2.4 Resumo da revisão sistemática da literatura

O termo recurso tem diferentes utilizações, dependendo da publicação analisada. A maior parte das publicações selecionadas na RSL usa o termo referenciando *recursos humanos*, ou mesmo *o executante de uma tarefa*. Duas das publicações usam o termo recurso como um recurso de informação. No entanto, em nenhuma publicação selecionada o termo recurso foi utilizado no contexto da infraestrutura necessária à implementação de um processo.

O que pode ser inferido é que não há uma padronização específica para a utilização do termo recurso no contexto das publicações analisadas. Há um certo descuido no uso do termo recurso que dificulta a interpretação do seu real significado. Para contornar tal problema, é interessante que, quando o termo recurso for utilizado, haja a definição explícita do mesmo e sejam caracterizadas entidades que são abrangidas pelo termo, por exemplo aplicações de software, pessoas etc. Isso é importante para facilitar a replicabilidade do estudo e definir o que pode e não pode ser utilizado para representar um recurso.

Em um ramo de pensamento mais ontológico, podemos ainda trazer à tona questões como a seguinte: um agente treinado por inteligência artificial que é capaz de executar tarefas de um processo é considerado um recurso humano? Todos os artigos analisados que usam o termo recursos humanos relacionam recursos humanos a executores de tarefas. Em um processo de negócio, executores de tarefas são chamados de atores, sendo que atores podem ser humanos, organizações ou sistemas de software que agem em nome de humanos ou organizações.

Além disso, é importante observar que o perfil dos recursos é usado em três dos artigos selecionados, e também podemos ver que a caracterização de um recurso é mais utilizada do que a caracterização de uma tarefa. A importância ou urgência de uma tarefa não tem impacto na maioria das publicações. Isso significa que, de certa forma, a prioridade das publicações analisadas é em estabelecer relações entre a tarefa e seu executor e recomendar o mais apto ou que traga mais ganho para a tarefa de modo local. Em outras palavras, questões globais como a urgência de uma tarefa em um processo, em geral, são ignoradas.

Duas publicações enfatizam a relação entre recurso e tarefa a fim de recomendar um recurso, levando em consideração a qualidade, custo e desempenho da execução. Não houve evidência de características, além do histórico de execução, que não estão relacionadas ao recurso ou à tarefa. Isso significa que nenhuma das publicações usou informações de contexto que não estavam relacionadas a tarefas ou recursos (por exemplo, o dia da semana) e nem utilizaram aprendizado a partir dos rótulos das tarefas.

Para a recomendação, nenhuma das publicações relatou o uso de serendipidade, propri-

idade sob a qual sistemas de recomendação oferecem sugestões inesperadas: itens que não são apenas relevantes e novos para o alvo, mas também significativamente diferentes dos itens já avaliados ou utilizados (KOTKOV; WANG; VEIJALAINEN, 2016). Isso se deve ao fato de que a recomendação age, nas publicações analisadas, de maneira similar à alocação de recursos, onde prioriza-se a otimização e não a satisfação (a não ser quando devidamente explícita e ponderada) do usuário.

Embora uma ampla variedade de algoritmos seja usada no processo de recomendação de recursos em BPM, apenas dois trabalhos relataram o uso de mais de um algoritmo clássico de aprendizagem. Os artigos selecionados relatam o uso de k-means, kNN, árvores de decisão, EM e regras de associação para aprendizagem e top-k para classificação de recomendação.

Por fim, com base na SLR, pode-se observar que existem poucas publicações que abordam recomendação no contexto de BPM e, em geral, a solução de recomendação de recursos fornecida pelos artigos selecionados está relacionada principalmente à alocação de recursos humanos em um processo.

5 FUNDAMENTAÇÃO DO PROBLEMA

Neste capítulo é feita uma análise do problema abordado neste trabalho. Na seção 5.1 são expostas informações relacionadas aos fatores de custo em BPM. Na seção 5.2 é explorado o tema de estimativa de recursos e suas aplicações em projetos. Na seção 5.3 é definido o problema em si e suas implicações.

5.1 Fatores de custo em BPM

Em relação aos custos de um processo, apesar de estimativa de custo ter recebido uma atenção grande nas últimas décadas e se tornado uma tarefa essencial na engenharia de sistemas de informação, é razoavelmente difícil aplicar abordagens existentes à BPM (MUTSCHLER; REICHERT, 2013), principalmente pelo fato da imensa variedade de fatores de custo tecnológico, organizacional e de projeto a serem considerados em projetos BPM.

Para fins de classificação de fatores de custo, neste trabalho utiliza-se a definição encontrada em (MUTSCHLER; REICHERT, 2013), onde são apontados dois diferentes fatores: fatores de custo estáticos (FCE) e fatores de custo dinâmicos (FCD), definidos a seguir:

- Fatores de custo estático (FCEs) representam custos cujos valores não mudam durante um projeto de BPM (exceto seu valor de tempo, que não é mais considerado no seguinte). Exemplos típicos: custos de licença de software, custos de software e custos de hardware para consultores externos.
- Fatores de custo dinâmicos (FCDs), por sua vez, representam os custos que são determinados pelas atividades relacionadas intrinsecamente a um projeto BPM. Essas atividades causam esforços mensuráveis que, por sua vez, variam devido à influência de fatores de impacto intangíveis. Exemplo típico: o redesenho relacionado a uma modificação na empresa.

Parte dos custos estáticos para implementação de processos está relacionada à aplicações invocadas por um *BPMS*. *BPMS* pode ser visto como o motor de execução de um processo de negócio definido (BEHESHTI et al., 2016), sendo que uma instância de um processo é tipicamente a execução do *workflow*. De modo geral, um *BPMS* age como um escalonador, distribuindo as tarefas a serem executadas entre os possíveis participantes do processo. O participante do processo é informado sobre a tarefa a ser executada e, dependendo do caso¹, é

¹Alguns tipos de tarefas, por sua natureza, não invocam aplicações, como é o caso das tarefas manuais, por

invocada uma aplicação que é necessária para que o participante execute tal tarefa.

Vale a pena mencionar que custos das operações do sistema, incluindo os custos de manutenção do sistema e administração do sistema, como apontado em (DRAHEIM, 2010), são difíceis de prever e às vezes complexos de determinar quando o sistema está em execução. Assim, com antecedência, os custos de um sistema de informação, às vezes, só podem ser estimados e não calculados.

Os custos de BPM podem ser definidos com base nas fases do ciclo de vida de BPM (MUTSCHLER; REICHERT, 2013). Como neste trabalho são discutidos os custos entre as fases de Redesenho do Processo e de Implementação do processo, pode-se dizer que os principais custos envolvidos são:

- **Custo de Redesenho de Processo:** O custo de redesenho do processo é determinado principalmente por dois custos estáticos: custos planejados para análise de processo e custos planejados para modelagem de processo. Alguns fatores impactam no custo do redesenho pelo analista, como complexidade do processo, engajamento da gerência, medos do usuário final, conhecimento do processo, conhecimento do domínio e a capacidade de redesenhar processos de negócios.
- **Custo de Implementação do Processo:** Os principais fatores envolvidos na implementação são custos de modelagem de processo ajustados², custos ajustados de gerenciamento de usuário e funções, custos de desenho ajustados, custos de modelagem de dados, custos de teste e custos diversos.

5.2 Estimativa de Recursos

Com relação à estimativa de recursos, sabe-se que mesmo gerentes que entendem como estimar as necessidades gerais de um projeto geralmente têm dificuldade em estimar os recursos necessários para cada tarefa dentro do projeto (ROYCE; GUPTA, 2000). Estimativas são muito importantes para a noção de, por exemplo, expectativas e gastos que podem existir em um projeto. Por ser um processo contínuo, estimativa é uma atividade crítica e pode fazer com que projetos sejam bem-sucedidos ou, por outro lado, rejeitados. Estimar os requisitos de recursos de um dado projeto ainda é algo arbitrário e *ad hoc* em muitas organizações.

Existem inúmeros métodos na literatura para a estimativa de recursos dentro de projetos, como em (ROYCE; GUPTA, 2000), (DELICATO; PIRES; BATISTA, 2017) e (), porém em

exemplo

²Custos ajustados se referem a custos influenciados por fatores adicionais relacionados a um processo específico

geral necessitam de mão-de-obra humana para tal e não levam em consideração informações do modelo do processo de negócio e, portanto, não consideram a modelagem relacionada ao redesenho do processo que é resultante da fase de Redesenho do Processo do ciclo de vida de processo. Isso implica que, quando existe a intenção de implementar um BPMS na organização, os recursos não são estimados baseando-se nas tarefas a serem executadas pelos participantes do processo.

5.3 Ciclo de Vida de Processo

De acordo com a descrição do ciclo de vida relatado no capítulo 2, a implementação de processos é uma etapa que engloba mudanças gerenciais e automação de processo. O analista de processo (tipicamente o responsável por coordenar o redesenho do processo), na maioria dos casos, utiliza BPMN para o redesenho do processo e, a partir do modelo resultante, inicia-se a implementação do processo por parte dos engenheiros de sistema, que envolve o desenvolvimento e as implantações na infraestrutura da organização que darão suporte ao processo *to-be*.

No momento da implementação, portanto, os engenheiros de sistema possuem a informação dos recursos tecnológicos que devem dar suporte a um dado processo. Repare que a disponibilidade, o custo, a quantidade e demais estimativas de recursos que são necessários para a implementação do processo são pensados na fase de implementação. Isso implica que quaisquer problemas decorrentes da falta de recursos necessários (ou da falta de intenção da organização em adquiri-los) para implementar o processo serão considerados apenas nesta fase, o que faz com que o analista de processo não consiga saber de antemão se os recursos que a organização necessita são um gargalo ao processo até que os engenheiros de sistema tentem implementar o processo *to-be*.

Existem algumas implicações decorrentes disto na seguinte situação: se uma organização não possui (ou não tem a intenção de adquirir) os recursos necessários para implementar um processo e isto é visto apenas na fase de Implementação do Processo, existem dois caminhos independentes que podem ser tomados pelo proprietário do processo, que são os seguintes:

1. *Retorno do processo à fase de Redesenho do Processo*, para que o mesmo seja redesenhado pelo analista de processo ou para que sejam tomadas as medidas necessárias para o andamento do projeto;
2. *Cancelamento temporário ou total do projeto de Implementação do Processo* por inviabilidade financeira.

No caminho 1, com o retorno do processo à fase de Redesenho do Processo a partir da fase de Implementação do Processo, há uma ruptura na ordem das fases do ciclo de vida de processo, já que a fase subsequente à Implementação do Processo é a fase de Monitoramento e Controle do Processo; porém, neste caso, há o retorno do processo para o Analista do Processo a fim de que o mesmo faça os ajustes necessários no modelo ou entre em contato com o proprietário do processo para tomar as medidas necessárias. Outros impactos que podem ser descritos são: a) o reacionamento do analista de processo após um provável término do redesenho, e b) acionamento da equipe de engenharia de processo sem que a mesma efetivamente consiga realizar a implantação dos recursos necessários na infraestrutura da organização. Repare que tais impactos podem aumentar consideravelmente o custo do processo, já que demandam tempo extra das equipes responsáveis pelas fases de Redesenho do Processo e de Implementação do Processo, e portanto aumentando os custos totais de Implementação e Redesenho mencionados na seção 5.1.

Já o caminho 2 representa, organizacionalmente, um gasto desnecessário na tentativa de implementar um processo inviável, já que neste momento do ciclo de vida tem-se o custo acumulado de todas as fases do ciclo de vida já executadas.

Sendo assim, os dois caminhos neste caso, implicam em gastos desnecessários e, portanto são indesejáveis. O ideal é que o analista de processos consiga a informação dos recursos necessários de antemão e, desta forma, tome conhecimento da possibilidade de implementação do processo a partir dos recursos existentes e não escale trabalho desnecessário para os engenheiros do sistema. Caso o modelo de processo redesenhado não seja suportado pelos recursos existentes, o analista pode ou redesenhar o processo ou, caso seja impossível implementar o processo com os recursos disponíveis, escalar tal informação para o proprietário do processo para que tome as devidas providências.

5.4 Resumo

Neste capítulo foi discutida a fundamentação do problema. Foram discutidos fatores de custo em BPM, sendo eles custos estáticos, como aplicações invocadas por BPMS, e dinâmicos. Foram discutidos fatores de custo nas fases de Redesenho e de Implementação do Processo. Foi, ainda, discutido o problema de se identificar os recursos necessários para um processo apenas na fase de Implementação de Processo e subsequentemente suas possíveis consequências.

6 RECOMENDAÇÃO DE RECURSOS PARA PROCESSOS DE NEGÓCIO

Este capítulo tem como objetivo a proposta de uma abordagem para identificação de recursos de infraestrutura necessários para implementar um processo, utilizando como base:

- o modelo de processo criado (ou redesenhado) pelo analista resultante da fase de redesenho;
- outros modelos de processos que a organização possui;
- recursos que a organização possui.

A proposta é recomendar aplicações (por exemplo, a aplicação *Microsoft Word* do tipo de recurso *editor de texto*) que devem ser invocadas pelo BPMS a partir do modelo de processo *to-be* desenhado pelo analista de processo para que o mesmo consiga identificar se a organização possui os recursos necessários para implementar o processo na organização ou tem a intenção de adquiri-los, ou ainda verificar se o orçamento previsto condiz com a realidade. Aplicações fazem parte dos custos estáticos de organizações. A escolha da recomendação de aplicações surgiu em virtude da observação, por parte do autor, da ausência de aplicações no momento de executar processos de organizações conhecidas.

A fim de estimar quais são os recursos necessários para implementar um determinado processo, é necessária a definição inicial de:

- Tipos de recurso com o qual uma dada tarefa do processo pode ser executada. Por exemplo, no caso de aplicações, os tipos de recurso podem ser *navegador, impressora, planilha de excel* etc.
- Recursos propriamente ditos, seu custo de aquisição, seu tipo, além de seus atributos. Por exemplo, Microsoft Excel® - proprietário, sem versão online, não colaborativo, etc. Os atributos podem ser definidos a critério da organização;
- Um conjunto de processos de negócio da organização. Os processos já devem ter a definição de quais os recursos que atualmente a organização utiliza para executar suas tarefas.

Além disso, é necessário que utilize-se, no conjunto de processos da organização, o padrão orientado pelas 7PMG (MENDLING; REIJERS; AALST, 2010) que se refere à utilização de nomear as tarefas utilizando-se verbo-objeto (por exemplo: "Criar o Roteiro de Viagem" ao invés de "Criação de Roteiro de Viagem"). Todas as outras diretrizes são incentivadas, mas não essenciais para o andamento do trabalho e portanto não são mencionadas.

A partir dessa configuração inicial, propõe-se fazer a recomendação dos recursos necessários para implementar o processo em duas etapas, sendo elas a *classificação de tipo de*

recurso e recomendação de recursos:

1. Na classificação do tipo de recurso, as tarefas do modelo de processo de negócio são classificadas pelos tipos de recursos necessários para executá-las;
2. Na recomendação de recursos, baseando-se nos tipos de recursos que as tarefas necessitam, são analisados os recursos existentes e recomendam-se os recursos que provavelmente a organização tenha mais interesse em utilizar ou adquirir.

6.1 Classificação do tipo de recurso

Para a classificação do tipo de recurso, são executados os passos a seguir:

1. Leitura do modelo de processo e extração do tipo de tarefa (por exemplo, manual) e do rótulo de cada tarefa;
2. Normalização dos rótulos das tarefas;
3. Aplicação de algoritmo classificador de texto;
4. Cálculo da acurácia da classificação.

6.1.1 Leitura do modelo

O resultado da fase de Redesenho do Processo é um modelo do processo *to-be*, desenhado pelo analista de processo. O modelo pode ser desenhado utilizando-se BPMN e alguma ferramenta comercial que pode ser utilizada para modelagem de processos (por exemplo *Bizagi*, *Signavio* e *Bonita*).

A representação de um modelo de processo para que possa ser lido fora de uma ferramenta de modelagem pode ser feita de diversas maneiras, como por exemplo a exportação, a partir da ferramenta de modelagem, para um arquivo em formato *.bpmn*. O arquivo no formato *.bpmn* é representado em XML (*Extensible Markup Language*), o qual é estruturado seguindo especificações definidas em OMG (2011) a partir de um *XML Schema*¹. É possível, a partir disso, extrair os dados do XML gerado utilizando-se, por exemplo, uma biblioteca que contenha funções para leitura de arquivos XML. Um exemplo de XML de uma tarefa do tipo *script* é exibido na Figura 6.2, enquanto um trecho de um XML Schema para tarefas do tipo *script* é exibido na Figura 6.1.

¹Um XML Schema especifica os elementos em um documento XML, validando sua estrutura

```

<xsd:element name="scriptTask" type="tScriptTask"
substitutionGroup="flowElement"/>
<xsd:complexType name="tScriptTask">
  <xsd:complexContent>
    <xsd:extension base="tTask">
      <xsd:sequence>
        <xsd:element ref="script" minOccurs="0" maxOccurs="1"/>
      </xsd:sequence>
      <xsd:attribute name="scriptFormat" type="xsd:anyURI"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:element name="script" type="tScript"/>
<xsd:complexType name="tScript" mixed="true">
  <xsd:sequence>
    <xsd:any namespace="##any" processContents="lax" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

```

Figura 6.1 – Exemplo de XML Schema para de Tarefa do tipo Script

```

<scriptTask id="Id_9574ea67-b34a-4fa5-9ab7-34388a2430e3" name="Atualizar dados do sistema">
  <documentation />
  <extensionElements>
    <bizagi: BizagiExtensions xmlns:bizagi="http://www.bizagi.com/bpmn20">
      <bizagi: BizagiProperties>
        <bizagi: BizagiProperty name="bgColor" value="#ECEFFF" />
        <bizagi: BizagiProperty name="borderColor" value="#03689A" />
      </bizagi: BizagiProperties>
    </bizagi: BizagiExtensions>
  </extensionElements>
  <incoming>Id_5d1511c0-bf45-41cd-bdb1-26a8b30415d9</incoming>
  <outgoing>Id_a837e728-d972-4eb1-a2fa-dd2b49821d5f</outgoing>
  <script />
</scriptTask>

```

Figura 6.2 – Exemplo de XML de uma tarefa do tipo Script

Assim que o modelo é lido, os dados extraídos do modelo (nome do processo, rótulos e tipos de tarefas) devem ser persistidos em algum banco de dados para análises e processamentos posteriores, afinal as informações do modelo devem servir de entrada para os demais passos da classificação. Por exemplo, a persistência pode ser feita em um banco de dados relacional com uma estrutura de dados definida, assim garantindo propriedades ACID (atomicidade, consistência, isolamento e durabilidade) (VOSSSEN, 1995).

6.1.2 Normalização de rótulos de tarefas

A normalização dos rótulos das tarefas de um processo inicia com a remoção de palavras vazias (*stopping words*). Palavras vazias, também chamadas de palavras de ruído, são palavras que dificilmente serão úteis para indexação de documentos. São classificadas em dois tipos (LI; CHEN; WANG, 2014):

- Palavras de função frequentemente usadas: são palavras que possuem pouca relevância em representar significado, como por exemplo as palavras "que", "é" e "em";
- Palavras de conteúdo frequentemente usadas: são palavras que possuem informação semântica mais sólida do que palavras de função, porém são tratadas como ruídos em virtude de sua alta frequência, como por exemplo as palavras "querer" e "através";

É importante salientar que quaisquer grupos de palavras podem ser removidas de um documento, dependendo do objetivo do processamento. Neste trabalho foram removidas apenas as palavras de função frequentemente usadas para evitar palavras semanticamente úteis ao rótulo, pois a média de palavras de um rótulo de tarefa de processo é relativamente baixa (em geral, de 3 a 5 palavras) e, portanto, uma redução maior implica em uma perda significativa no documento.

Portanto, o processo é o seguinte: para cada palavra dos rótulos das tarefas, é verificado se não pertencem às palavras vazias (em geral, ferramentas de linguagem natural já definem as palavras vazias mais comuns de um idioma específico - a biblioteca NLTK (LOPER; BIRD, 2002), por exemplo, especifica palavras como 'a', 'ao', 'aos', 'aquela', 'aquelas', 'aquele' e 'aqueles' como sendo palavras vazias). Caso a palavra pertença, ela não é incluída, para fins de classificação, na lista de palavras pertencentes ao rótulo. Este processo é executado em todos os rótulos das tarefas do processo sob o qual ocorrerá a classificação dos tipos de recurso.

Em um segundo momento é aplicada a radicalização (HUYCK; ORENGO, 2001). A radicalização é o processo de remover afixos da palavra (por exemplo, gerúndios) mantendo o radical da palavra. O processo de radicalização é baseado nas regras morfológicas de linguagens naturais, variando conforme o idioma, ou seja, implica na utilização de diferentes algoritmos de acordo com o idioma utilizado. A ideia é que uma palavra típica possui um radical que referencia uma ideia central ou um significado, e certos afixos são adicionados para modificar o significado ou ajustar a palavra ao seu papel sintático, e portanto a proposta da radicalização é retirar os afixos e reduzir a palavra à sua essência (PAICE, 1994).

Radicalização, quando utilizada no pré-processamento de documentos antes de sua indexação, torna-se particularmente útil para indexação e busca de documentos, objetivando aumentar o nível de recuperação de documentos (VIERA; VIRGIL, 2007). É um processo que faz uso de regras para determinar o radical sem considerar o vocabulário, contexto da palavra e classe gramatical. A diferença entre radicalização e lematização é que a lematização é um processo mais complexo de deflexionar uma palavra para determinar o seu lema, utilizando para isso a determinação da classe gramatical e o contexto da palavra a fim de retornar seu lema (SINGH; GUPTA, 2017).

Neste trabalho é sugerida a radicalização para padronizar palavras com mesmos radicais e indexá-las como sendo iguais, utilizando o seu radical. Por exemplo, as palavras agenda, agendar e agendamento são indexadas como *agend*, evitando assim que essas palavras sejam interpretadas como palavras diferentes na aplicação de algoritmos de classificação de texto e melhorando, portanto, a precisão da classificação.

Para executar a normalização dos rótulos das tarefas do processo, deve-se radicalizar cada uma das palavras que compõem o rótulo. Portanto, as palavras de um rótulo são normalizadas individualmente. Para cada uma das palavras do rótulo de uma tarefa específica aplica-se o processo de radicalização com auxílio de algoritmo de língua portuguesa para tal, como por exemplo utilizando pacotes da biblioteca NLTK. O processo de radicalização, enfim, retorna os grupos de palavras radicalizadas para cada um dos rótulos das tarefas do processo e cada rótulo, então, é representado por um grupo de radicais. Um exemplo de radicalização é feito pelo algoritmo RSLP (FLORES; MOREIRA; HEUSER, 2010), conforme o diagrama ilustrado na Figura 6.3.

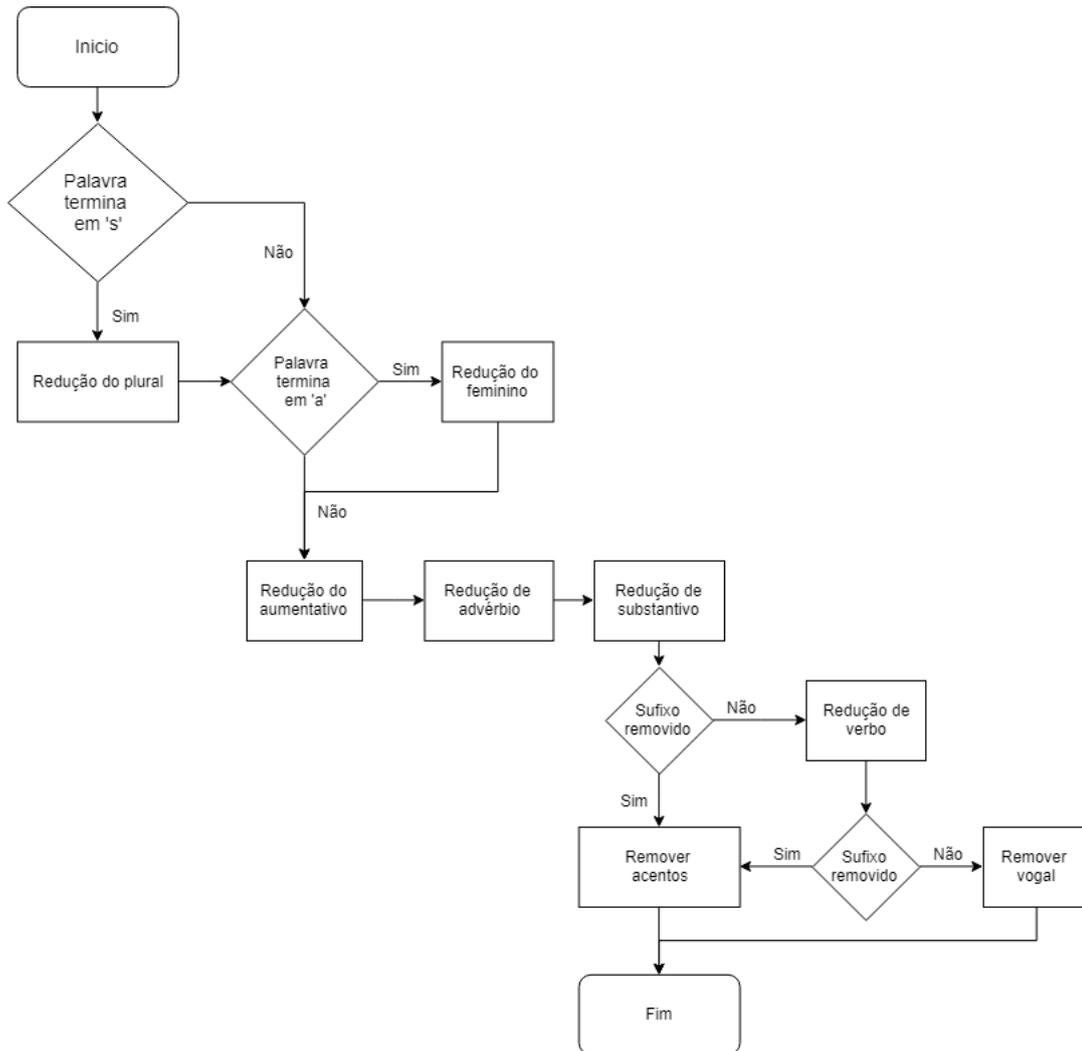


Figura 6.3 – Fluxograma do algoritmo de radicalização RSLP, adaptado de (HUYCK; ORENGO, 2001)

6.1.3 Aplicação de algoritmo classificador de texto

Após a normalização dos rótulos, é feita a classificação das tarefas em tipos de recursos. Neste momento, os tipos de tarefa extraídos anteriormente na seção 6.1.1 se tornam essenciais:

- Tarefas do tipo manual não precisam ser classificadas, afinal tarefas manuais não são executadas com auxílio de aplicações;
- Tarefas do tipo serviço também não necessitam de classificação, afinal também não são executadas com o auxílio de aplicações de usuário. Tarefas de serviço utilizam por padrão um web service ou uma aplicação automatizada que não entra no escopo deste trabalho;
- Todos os outros tipos de tarefas (usuário, envio, recebimento, regra de negócio e abstrata) são classificados.

Para classificação de textos, existem diversos algoritmos como Naïve Bayes, Árvores de Decisão, Máquina de Vetores de Suporte, dentre outros. Comparações entre algoritmos de classificação sobre conjuntos de dados diferentes são exploradas em diversos estudos na literatura, como em (JAISWAL; LOKHANDE, 2014) e (VEENA; SHENOY; SHENOY, 2018). Dependendo do conjunto de dados, a acurácia dos algoritmos difere e, portanto, a escolha do algoritmo é fortemente relacionada ao conjunto de dados a ser classificado. Para decidir sobre o algoritmo a ser utilizado, foi feita uma comparação de acurácia entre os algoritmos de Árvore de Decisão, Naïve Bayes e Máquinas de Vetores de Suporte para classificação de rótulos de processos utilizando-se o conjunto de tarefas de processos de uma universidade pública e federal relatado na Tabela 6.1. A tabela exhibe os nomes dos processos e, conseqüentemente, o número de tarefas que cada um dos processos possui. Num total foram utilizadas 521 rótulos de tarefas para o teste comparativo de acurácia.

Tabela 6.1 – Conjunto de processos para teste dos classificadores

Nome do processo	Número de Tarefas
Atualização sazonal do sistema de alocação	11
Inserção de novos ocupantes no espaço	14
Seleção de professor para gabinete individual vago	13
Cadastro de aluno especial	16
Aquisição de equipamentos	10
Aquisição - fomento	12
Criação de ação de extensão	24
Preparação de solenidade recepção aos alunos	18
Processo de compras	51
Processo de concurso	33
Processo de eleições internas	29
Processo de Oferecimento Semestral de Disciplinas	8
Processo de entrega de trabalho de graduação	53
Processo de progressão funcional	8
Processo de patrimoniamiento	18
Aprovação de projetos	24
Tramitação de contratos	24
Solicitação de Auxílio à Comissão de Pesquisa	21
Cadastro PPGC	5
Defesa de Tese	22
Definição de elaboradores POSCOMP	4
Definição de Sedes POSCOMP	9
Inscrição Pós	13
Matrícula Pós	9
Controle de Ponto Diário	4
Controle de Ponto Semanal	16
Seleção doutorado	11
Seleção empresa aplicadora POSCOMP	9
Seleção mestrado	12
Solicitação bolsa PPGC	20

Para a comparação entre as acurácias dos classificadores foi utilizado o software WEKA², reconhecido como um sistema de referência em mineração de dados e aprendizado de máquina, sendo que alcançou aceitação generalizada no meio acadêmico e nos círculos de negócios, e se tornou uma ferramenta amplamente usada para pesquisas de mineração de dados (HALL et al., 2009). Na Figura 6.4 é mostrada a interface da aplicação WEKA.

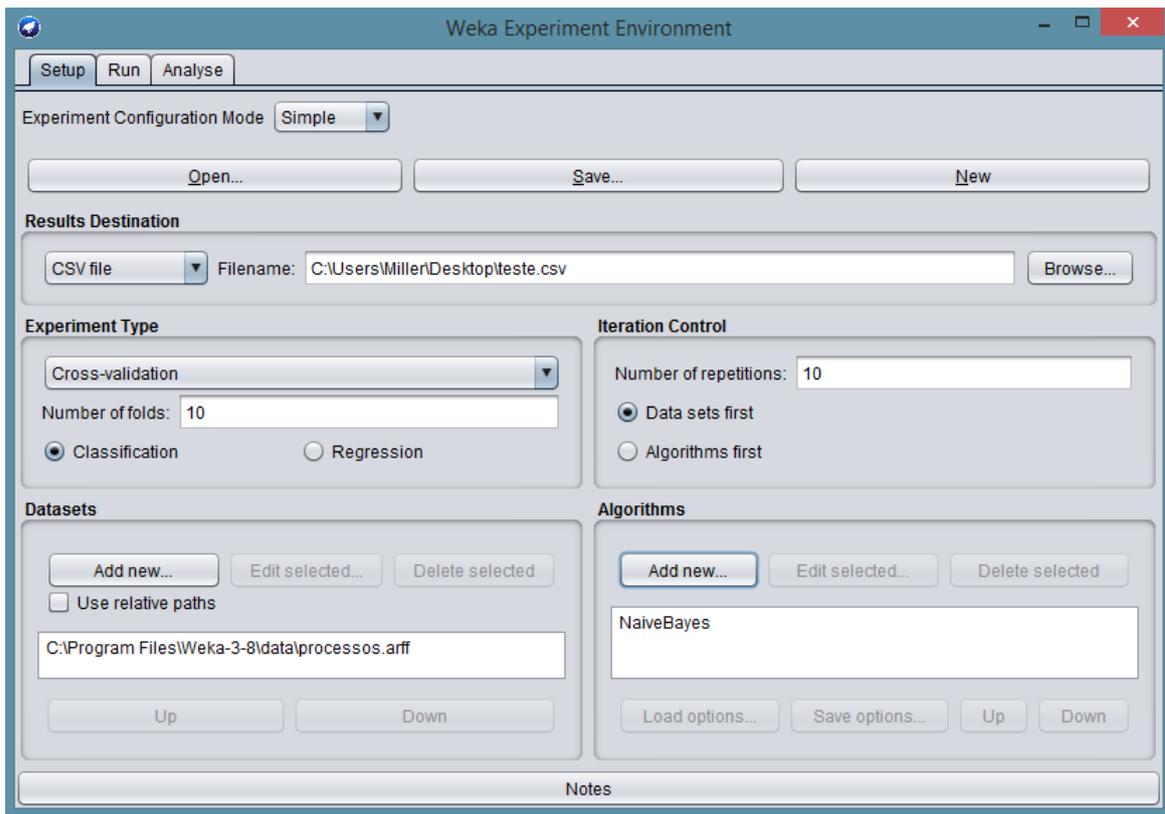


Figura 6.4 – Aplicação WEKA

A acurácia dos algoritmos foi calculada, portanto, utilizando a seguinte configuração:

- Aplicação WEKA num computador com sistema operacional Windows 8;
- Validação cruzada com $k=10$;
- Arquivo no formato .ARFF contendo os dados para treinamento dos 521 rótulos de tarefas dos processos exibidos na Tabela 6.1). Na Figura 6.5 é apresentado um exemplo de como foi feita a disposição dos dados que serviram de entrada para os classificadores.

²<https://www.cs.waikato.ac.nz/ml/weka/>

```

@relation recurso

@attribute label String
@attribute tipo {agenda, navegador, planilha, cliente_email}

@data
"Agendar entrevista", agenda
"Alterar planilha de presencas", planilha
"Notificar resultado aos entrevistados", cliente_email

```

Figura 6.5 – Exemplo de entrada de dados para a aplicação WEKA

Os algoritmos foram testados utilizando suas configurações padrão no Weka. A acurácia foi medida utilizando-se todas as possíveis classes para as instâncias, ou seja, foram checados todos os exemplos classificados corretamente e dividiu-se pelo número total de instâncias que deveriam ser classificadas. As instâncias foram randomicamente separadas em 10 diferentes grupos para ser feita a validação cruzada, sendo que tal processo foi executado 10 vezes.

Na Tabela 6.2 são exibidas as acurácias de cada um dos algoritmos analisados para o conjunto de dados da Tabela 6.1. Adicionalmente, são mostrados a revocação e a precisão.

Tabela 6.2 – Comparação da acurácia dos algoritmos

Algoritmo	Acurácia	Revocação	Precisão
Árvore de Decisão / C4.5	77.823	83.3	79.87 %
Naïve Bayes	80.211	83.93	79.901 %
Máquinas de Vetores de Suporte	78.430	82.44	77.094 %

Como apresentado na Tabela 6.2 é possível observar que todos os algoritmos tiveram acurácia próxima a 80%, sendo que a melhor acurácia pertence ao algoritmo Naïve Bayes, assim como a revocação e a precisão. Em virtude disso, para a classificação do tipo de tarefa, neste trabalho, é proposta a utilização do algoritmo Naïve Bayes. Antes da execução do algoritmo, como discutido no início do capítulo, necessita-se que existam tipos de recursos pré-definidos (por exemplo, planilha eletrônica, editor de texto, agenda eletrônica, etc) e, como Naïve Bayes é um algoritmo supervisionado, necessita-se também que existam processos de negócio com tarefas que já foram anteriormente classificadas por tipos de recurso (por exemplo, "Agendar Entrevista" classificada com o tipo de recurso "agenda"). Quanto mais precisa a classificação que antecede o algoritmo, melhor será o treinamento e a classificação será mais precisa.

O algoritmo Naïve Bayes é proposto, portanto, para ser aplicado no conjunto de tarefas de um processo de negócio específico, utilizando para treinamento o conjunto de tarefas dos processos dessa organização. Por ser um classificador probabilístico, quanto maior o número de

tarefas já classificadas, melhor o desempenho do algoritmo. O algoritmo calcula a probabilidade de uma palavra da tarefa pertencer a uma determinada classe (no caso, tipo de recurso).

É importante salientar que existe a possibilidade de que algumas palavras não apareçam no treinamento e, portanto, normalmente seria atribuída probabilidade 0 à ocorrência dessas palavras, o que não é verdade, pois tal probabilidade é baixa, porém não nula. Caso a probabilidade de uma palavra fosse nula, haveria um 0 na multiplicação das probabilidades, e assim toda a equação resultaria em 0. Para evitar esse tipo de situação, propõe-se a técnica de *sua-
vização Laplaciana* (ZHAI; LAFFERTY, 2004), onde diminui-se a probabilidade das palavras encontradas para, assim, sobrar uma fatia de probabilidade para palavras não encontradas no treinamento.

6.1.4 Cálculo de acurácia

Para calcular-se a acurácia da classificação do tipo de recurso, utilizou-se aqui a validação cruzada k-fold, onde os rótulos foram divididos em 10 grupos diferentes. A cada rodada de validação um desses grupos foi utilizado para teste, enquanto os outros 9, para treinamento. A média de acurácia preditiva resultante foi de 81,7%, valor esperado para o conjunto de dados existente (SORDO; ZENG, 2005). Existem na literatura diversas técnicas para aumento de acurácia do classificador Naïve Bayes, como em Sotiris (1013) ou Sordo e Zeng (2005), porém não serão abordadas nesse trabalho pois são recomendadas como trabalho futuro.

6.2 Recomendação de recursos

Após a classificação do tipo de recurso necessário por uma tarefa de um processo específico, é feita a recomendação dos recursos mais apropriados para a organização. A recomendação é feita utilizando filtragem baseada em conteúdo, ou seja, analisando-se os atributos dos recursos e o perfil da organização. A escolha se deve ao fato de que processos e aplicações utilizadas nas tarefas de processos podem divergir muito de organização para organização e, portanto, o escopo das escolhas dos recursos foi feita apenas internamente, a partir de decisões passadas da organização.

Para a recomendação ser feita, as seguintes condições devem ser cumpridas:

1. *Identificação dos recursos da organização*: devem ser definidos todos os recursos que a organização possui;

2. *Identificação dos tipos de recurso*: para cada um dos recursos da organização deve ser definido o tipo (por exemplo, Microsoft Excel - editor de planilha). Essa categorização é importante para recomendar apenas recurso do tipo identificado pela classificação de tipo de recurso;
3. *Perfil da organização*: Para criar um perfil de uma organização específica, devem ser identificadas as características relevantes nos recursos que a organização possui (por exemplo: fabricante, disponibilidade *web*, etc). No momento em que tais informações existem é possível identificar quais recursos são mais adequados ao perfil da organização e que devem ser recomendados.

A partir dessas condições propõe-se a recomendação de recursos utilizando-se similaridade. Na subseção 6.1 foi discutido um método para definir o tipo de recurso necessário para cada uma das tarefas de um processo de uma organização; para cada um dos tipos de recursos identificados para um processo são recomendados os recursos mais similares ao perfil da organização.

Para identificar a similaridade entre a organização e os possíveis recursos, devem ser definidos:

- um ponto, no espaço dimensional, com base no perfil da organização. Tal ponto é a média aritmética dos atributos dos recursos da organização. Os atributos dos recursos da organização podem ser valorados de qualquer forma (por exemplo, 0 ou 1 indicando existência ou não do atributo - como por exemplo se é ou não *open source*). Esses valores são somados e, após, divididos pelo número de recursos utilizados nos processos a fim de obter a média aritmética;
- n outros pontos, referentes aos n recursos que são do tipo de recurso identificado para uma tarefa específica.

Como essa recomendação visa ser um alicerce para identificação de recursos e custos necessários para implementar um processo, será executada iterativamente entre todas as tarefas do processo para identificar todos os recursos necessários para o processo. A partir disso, propõe-se calcular a distância entre o perfil da organização e os recursos existentes utilizando-se alguma medida de distância, por exemplo, distância euclidiana.

6.3 Resumo

Neste capítulo foi discutida a abordagem para recomendação de recursos para a execução de um processo. Foi proposta uma abordagem em duas etapas, sendo elas a classificação por tipo de recurso e a recomendação dos recursos baseando-se nos tipos de recurso identificados.

Para a classificação por tipo de recurso sé necessária a leitura do modelo de processo (em formato bpmn), extração do tipo de tarefa e rótulo, normalização dos rótulos, aplicação de algoritmo classificador de texto e cálculo de acurácia. Foram avaliadas, utilizando processos de uma universidade, as acurácias de três algoritmos classificadores de texto, com destaque para o Naïve Bayes que obteve o melhor resultado.

Para a recomendação dos recursos baseando-se nos tipos de recurso identificados é necessário identificar os recursos da organização, os tipos de recurso existentes na organização e identificar o perfil da organização para assim identificar os recursos mais similares ao perfil, utilizando para isso cálculo de distância.

7 IMPLEMENTAÇÃO DA ABORDAGEM PARA RECOMENDAÇÃO DE RECURSOS

Foi desenvolvida uma implementação da abordagem para recomendar recursos necessários na execução de processos com o propósito de testar a abordagem. A implementação foi feita utilizando-se a linguagem Python e o framework Django.

Python é uma linguagem de programação interpretada e orientada a objetos, e combina poder notável com sintaxe muito clara (ROSSUM, 1995). Além disso, é portátil, podendo ser executado em muitas variantes do Unix, no Mac e no Windows 2000 e posterior.

Em relação a módulos para linguagem natural, Python possui uma ampla variedade de bibliotecas, com destaque para a NLTK. NLTK é um conjunto de módulos de código aberto que fornecem suporte computacional para processamento de linguagem. NLTK foi originalmente criada em 2001 como parte de um curso de lingüística computacional no Departamento de Ciência da Computação e da Informação da Universidade da Pensilvânia. Desde então, foi desenvolvido e expandido com a ajuda de dezenas de colaboradores.

Python ainda possui uma vasta variedade de *frameworks*, como é o caso de Django, *framework* de código aberto que incentiva o desenvolvimento rápido e desenho de código limpo e pragmático, sendo seguro e escalável, além de possuir uma comunidade imensa. Django possui o padrão arquitetural de três camadas MTV (*model*, *template* e *view*). A camada *model* é a fonte única e definitiva de informações sobre os dados do sistema. Ela contém os campos e comportamentos essenciais dos dados que a serem armazenados. Os *models* são estruturas definidas dentro da camada, e geralmente cada um representa uma única tabela do banco de dados. O banco de dados utilizado para registrar as tabelas definidas na camada de modelo da aplicação foi o Postgres¹. Postgres é um banco relacional de código aberto, robusto e conhecido por sua arquitetura comprovada, confiabilidade, integridade de dados, conjunto robusto de recursos, extensibilidade e dedicação da comunidade de código aberto por trás do software.

Neste contexto, Django foi utilizado para a implementação do classificador de tipos de recurso e para a recomendação dos recursos. Foi desenvolvida uma página onde se é possível:

- Cadastrar recursos, como exibido na Figura 7.1;
- Cadastrar organizações e seus recursos, como exibido na Figura 7.2;
- Importar processos (arquivos .bpmn) em organizações, como apresentado na Figura 7.3;

O desenvolvimento foi feito no ambiente Linux (distribuição Ubuntu), e os seguintes pacotes adicionais foram utilizados durante o desenvolvimento:

¹<https://www.postgresql.org/>

- `virtualenv`²: é uma ferramenta para criar ambientes Python isolados. Com ela é possível facilmente reproduzir o ambiente e os pacotes utilizados em outras máquinas;
- `psycopg`³: é um adaptador para integração do banco de dados PostgreSQL com Python;
- `pyflakes`⁴: ferramenta para verificação de erros nos arquivos de código-fonte escritos em Python.

Resources

Create New Resource

Name	Price	Collaborative	Open Source	Available at web	Multiplatform	Resource Type	Delete
Firefox	Free	No	No	Yes	Yes	Browser	X
Google Chrome	Free	No	No	Yes	Yes	Browser	X
Google SpreadSheet	Free	Yes	No	Yes	Yes	Planilha Eletrônica	X
Google Drive	Free	Yes	No	Yes	Yes	Gerenciador de Arquivos	X
Shared File System	Free	Yes	No	No	Yes	Gerenciador de Arquivos	X

Figura 7.1 – Exemplo de cadastro de recursos na aplicação

Organizations

Create New Organization

Name	Suite	Delete
UFRGS	Microsoft Excel, Microsoft Word	X
Organization Test	Microsoft Excel	X
PPGC	Shared File System, Webmail	X

Figura 7.2 – Exemplo de cadastro de organização na aplicação

New Process

Name:

Raw file:

Organization:

Figura 7.3 – Exemplo de importação de processo na aplicação

²<https://virtualenv.pypa.io/en/latest/>

³<http://initd.org/psycopg/>

⁴<https://pypi.org/project/pyflakes/>

Na etapa de classificação do tipo de recurso, foram seguidos os seguintes passos:

- Para extração dos elementos de um modelo de processo foi utilizado o módulo interno do Python ElementTree⁵ que facilita a manipulação de XML. Os dados extraídos do XML foram salvos em tabelas utilizando o banco de dados Postgres. A estrutura do modelo é exibida na forma de diagrama UML na Figura 7.4. No diagrama UML são apresentadas as classes "tarefa", "processo", "organização", "perfilrecurso" e "tipo de recurso", bem como as associações, composições, cardinalidades e atributos de cada uma delas.
- A radicalização dos rótulos das tarefas dos processos foi feita utilizando-se o módulo RSLP Stemmer (FLORES; MOREIRA; HEUSER, 2010), algoritmo que é incluído na biblioteca NLTK. O algoritmo remove os sufixos das palavras conforme o diagrama ilustrado na Figura 6.3. Para a definição das palavras vazias também foi utilizada a biblioteca NLTK, que possui o registro de um conjunto de palavras vazias do idioma português. O resultado da radicalização são rótulos de tarefas dos processos sem sufixos e sem palavras de função comumente usadas no português.
- Foi desenvolvido o algoritmo Naïve Bayes para fazer a classificação dos tipos de recurso necessários para implementar tarefas de um processo específico. O algoritmo pode ser encontrado em <https://github.com/Biazus/master>.
- Por último, foi implementada a validação cruzada para testar a acurácia do classificador. A validação cruzada foi feita em 10 rodadas de validação, e as tarefas foram aleatoriamente selecionadas para compor os grupos de teste e treinamento em cada uma das rodadas, onde os grupos foram divididos em 10, sendo 1 grupo de teste e o restante utilizado para treinamento. No final foi calculada a média aritmética entre os resultados da validação cruzada para cada uma das rodadas.

⁵<https://docs.python.org/2/library/xml.etree.elementtree.html>

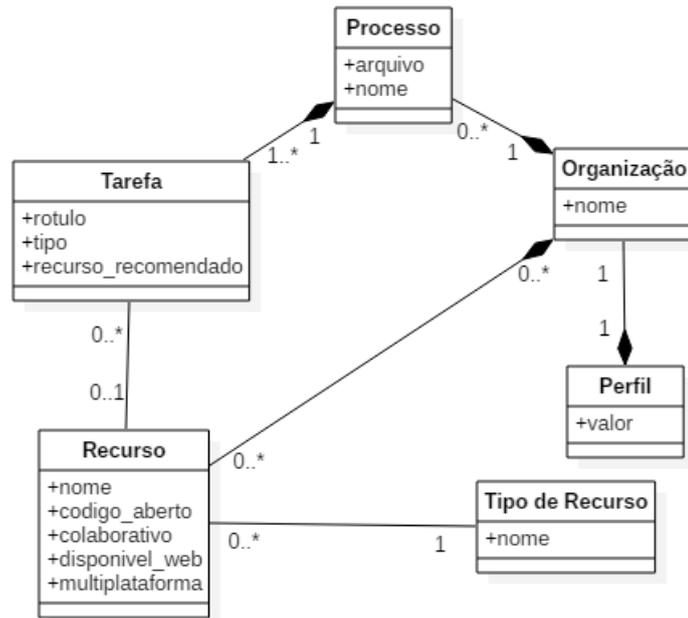


Figura 7.4 – Diagrama UML das classes da aplicação

Um exemplo de classificação de tipo de recurso é exibida na Figura 7.5. Na Figura, o botão "*Classify Tasks*" é utilizado para gerar os dados da coluna "*Classified Resource Type*".

Process "definicao de elaboradores poscomp"

Name:

Organization:

Tasks

Label	Type	Resource	Resource Type	Classified Type	Recommended Resource
Disparar email de chamada aos PGs	user	Webmail	Cliente de email	Cliente de email	Webmail
Solicitar convocação de elabores à empresa aplicadora	user	Webmail	Cliente de email	Cliente de email	Webmail
Convocar elaboradores	user				
Avaliar candidatos a elaboradores	user				

Figura 7.5 – Exemplo da exibição da classificação e recomendação na aplicação

Na Figura 7.6 é exibido o botão "Cross Fold Validation", onde é possível clicar e exibir a acurácia do modelo classificador para as tarefas existentes nos processos da organização.

Figura 7.6 – Exemplo da exibição da classificação na aplicação

Para construir o perfil da organização, utiliza-se a média aritmética entre os atributos de todos os recursos. O cálculo de similaridade para recomendação foi desenvolvido utilizando-se distância euclidiana entre o perfil da organização e o processo para o qual os recursos são recomendados.

Supondo, nesse caso, que fossem utilizados 3 recursos para treinamento com os valores dos atributos mostrados na Tabela 7.1, por exemplo, o perfil da organização seria um conjunto de cinco valores contendo a média aritmética para cada um dos atributos:

- Atributo 1: $1 + 0 + 0 / 3 = 0.333$
- Atributo 2: $0 + 0 + 1 = 0.333$
- Atributo 3: $0 + 1 + 1 = 0.666$
- Atributo 4: $0 + 1 + 1 = 0.666$
- Atributo 5 = $0 + 0 + 0 = 0$

Tabela 7.1 – Exemplo de valores de atributos para os recursos

Recurso	Colaborativo	Aberto	Versão Web	Multiplataforma	Interno
Microsoft Word	0	0	0	0	
Google Agenda	1	0	1	1	
Firefox	0	1	1	1	

O perfil da organização seria, nesse caso, um ponto com as coordenadas [0.333, 0.333, 0.666, 0.666, 0].

A similaridade entre um recurso e o perfil organizacional é feita com distância Euclidiana, que é a distância entre dois pontos no espaço dimensional, calculada por:

$$d(p, q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + (q_3 - p_3)^2 + \dots + (q_n - p_n)^2}$$

A distância é medida, portanto, entre o ponto no espaço dimensional baseado no perfil da organização e os pontos referentes aos recursos candidatos a serem recomendados. Supondo

que um recurso seja identificado pelo ponto [0, 0, 0, 1, 1] no espaço dimensional, sua distância para o perfil da organização seria de

$$\sqrt{(0 - 0.333)^2 + (0 - 0.333)^2 + (0 - 0.666)^2 + (1 - 0.666)^2 + (1 - 0)^2} = 1.3330.$$

Esse valor resultante seria comparado entre os valores dos outros recursos candidatos à recomendação, e seria recomendado o recurso com menor valor (ou seja, o que representa a menor distância, portanto mais similar).

Na Figura 7.5 pode-se observar a coluna "Recommended Resource" onde são descritos os recursos, que é gerada a partir do botão "Recommend Resources".

O código-fonte da classificação se encontra no anexo 1. O restante do código-fonte pode ser obtido pelo link <https://github.com/Biazus/master>.

7.1 Experimento

Para validar a abordagem de recomendação foi feito um experimento utilizando o conjunto de processos do departamento de pós-graduação de computação da universidade.

Os processos utilizados para o experimento são os mesmos utilizados para testar os classificadores, apresentados na Tabela 6.1. O conjunto de processos utilizados é resultante da modelagem de processos feita por diferentes alunos da disciplina de Modelagem e Gerenciamento de Processos de Negócio ministrada na própria universidade. Os processos foram revisados no decorrer da disciplina.

Tabela 7.2 – Tipos de recursos utilizados com exemplos de recursos

Tipo	Exemplo
Cliente de Email	Microsoft Outlook
Navegador	Firefox
Gerenciador de arquivos	Google Drive
Aplicativo Bancário	Aplicativo Banco do Brasil
Planilha Eletrônica	Microsoft Excel
Sistema de Alocação	Sistema alocação de salas da Universidade
Agenda	Google Agenda
Editor de texto	Microsoft Word
Impressão	PDF Printer

Foram cadastrados na aplicação desenvolvida, ainda, recursos disponíveis no departa-

mento da universidade, além de um conjunto de recursos possíveis para aquisição. Logo após, foram definidos seus tipos. Os tipos de aplicações identificados são exibidos na Tabela 7.2.

Para o experimento foram utilizados os seguintes atributos de aplicações, definidos com base nas funcionalidades das aplicações:

- Open-source ou proprietário;
- Possui versão web ou não possui;
- Colaborativo ou não colaborativo;
- Multiplataforma ou plataforma específica;
- Desenvolvido na universidade (interno) / Desenvolvido fora da universidade.

Os recursos foram, então, categorizados a partir dos atributos acima. Logo após, foi feita a validação cruzada com $k=10$ utilizando, portanto, os processos do departamento. Em cada rodada da validação, a partir as tarefas do processo usadas para treinamento foi criado o perfil do departamento da universidade, baseando-se nas médias dos valores dos atributos. Como os atributos identificados são binários, ou seja, assumem valor 1 para caso positivo e 0 para negativo, então o perfil para cada rodada é composto por 5 valores entre 0 e 1.

Já que as tarefas de teste, neste caso, eram o alvo da recomendação, a aplicação recomendou recursos para as tarefas em cada uma das rodadas da validação cruzada, sendo os recursos relacionados aos tipos de recurso que o classificador Naïve Bayes sugeriu para as tarefas. A recomendação foi realizada por similaridade entre o perfil do departamento gerado em cada rodada e os recursos cadastrados na aplicação.

A acurácia da recomendação entre todas as 521 tarefas dos processos de treinamento (Tabela 6.1) após a validação cruzada foi de 83%. Apesar de que os experimentos revelaram que é possível recomendar recursos a partir de modelos de processos utilizando os rótulos das tarefas algumas considerações devem ser levadas em conta:

- Um número maior de processos e, conseqüentemente, tarefas para treinamento no momento da classificação de tipos de recurso tenderia a melhorar a acurácia da classificação pelo Naïve Bayes;
- O Naïve Bayes, apesar de se mostrar eficiente, não leva em consideração, por exemplo, a posição relativa das palavras no rótulo das tarefas. Não foi medido, portanto, o impacto de se classificar utilizando um classificador que considera a posição relativa das palavras;
- Como os processos foram modelados por diferentes alunos existem vieses linguísticos bem maiores do que se fossem por apenas uma pessoa;
- O grau de relevância dos atributos não foi definido e, portanto, o peso é igual para todos

eles. Existem alguns trabalhos na literatura, porém, que abordam a questão de peso em atributos como Panday (2018) e Yongguang (2002).

7.2 Resumo

Neste capítulo foram discutidas as tecnologias utilizadas para o experimento do trabalho, como Python, Django e NLTK. Foi também apresentada a aplicação desenvolvida para demonstrar a abordagem de recomendação de recursos para processos de negócios em uma organização. Por último, verificou-se que a acurácia da recomendação foi de 83% e foram mencionadas algumas considerações sobre o resultado obtido.

8 CONCLUSÃO

Neste trabalho foi proposta uma abordagem para recomendação de recursos de infraestrutura com o objetivo de minimizar o problema no contexto onde modelos de processo oriundos da fase de Redesenho do Processo não são compatíveis com a infraestrutura da organização, impactando assim em custos desnecessários (como por exemplo a tentativa de implementar processos quando não se possuem todos os recursos necessários) e numa ruptura na ordem do ciclo de vida (processo que se encontra na fase de Implementação do Processo precisando voltar para a fase de Redesenho do Processo).

Com a recomendação, o analista do processo fica ciente de quais são os recursos necessários e, assim, em conjunto com o proprietário do processo, pode tomar as medidas cabíveis, como por exemplo continuar o Redesenho do Processo, modificando o processo de forma que o mesmo se encaixe na infraestrutura da organização.

O trabalho apresentou, ainda, a implementação da abordagem proposta, onde foi possível cadastrar processos e testar a abordagem. Com a aplicação também foi possível calcular a acurácia utilizando validação cruzada. A acurácia resultante da recomendação foi de 83%.

8.1 Contribuições do trabalho

Como principais contribuições do trabalho destacam-se:

- A análise comparativa entre três algoritmos classificadores de texto no contexto de rótulos de tarefas de processos. A análise utilizou processos de uma universidade pública federal, e a acurácia do Naïve Bayes se destacou em relação aos outros dois algoritmos analisados (Máquinas de Vetores de Suporte e Árvores de Decisão);
- A definição de uma abordagem para a recomendação dos recursos de infraestrutura necessários para implementar um processo de negócio específico baseando-se em modelos de processo, onde sugere-se o treinamento com os rótulos das tarefas dos processos da organização. É proposta ainda a criação de um perfil organizacional, com base nos atributos dos recursos que a organização utiliza, para recomendar os recursos mais similares à organização;
- A implementação da abordagem proposta utilizando Python e outras tecnologias relacionadas. A implementação é útil para demonstrar como a abordagem pode ser utilizada para obter informações sobre os recursos necessários para implementar um processo de

negócio;

- O desenvolvimento de uma RSL que visa obter o estado da arte em relação a abordagens para estimativa ou recomendação de recursos em BPM. A RSL identifica também as características dos recursos e das tarefas que são consideradas para a recomendação nos trabalhos. Ainda é discutido na RSL a utilização do termo "recurso" nos trabalhos selecionados. Como limitações da RSL, esta foi desenvolvida considerando apenas três bibliotecas digitais e apenas trabalhos no idioma inglês, e portanto sugere-se como trabalho futuro extendê-la com a utilização de mais bibliotecas e considerando trabalhos em outros idiomas.

8.2 Limitações

Algumas limitações foram identificadas ao longo do trabalho e são dispostas nessa seção. Uma das principais limitações é relacionada aos processos utilizados para testar os classificadores e para testar a recomendação. Os processos são modelados por diferentes alunos da universidade, o que pode influenciar negativamente na classificação, pois não existe a normalização de sinônimos, por exemplo (gerar e criar são tratados com significados distintos pelo classificador).

Ainda, é importante ressaltar que a abordagem foi testada em um único departamento da universidade, e não foi utilizada em outros departamentos ou organizações. Isso significa que podem existir outras variáveis que sejam importantes no contexto de outras organizações e processos e não foram levadas em conta no trabalho.

Outra limitação é em relação ao classificador de tipo de recurso, Naïve Bayes, que não considera, por exemplo, posição relativa das palavras nos rótulos das tarefas. Também não se consideram as influências que as posições das tarefas na modelagem dos processos teriam na classificação de tipos de recurso.

8.3 Trabalhos Futuros

Como trabalhos futuros são sugeridas customizações na abordagem, como a utilização de filtragem colaborativa na recomendação, ou seja, possibilitar a recomendação baseando-se no que outras organizações com "gostos" similares utilizam, ou até mesmo entre diferentes departamentos dentro da mesma organização.

Ainda, sugere-se investigar a acurácia da recomendação para processos criados em outros idiomas além do português, para processos criados por um único analista e também para processos com maior número de tarefas em relação aos utilizados.

Na implementação foi utilizada a distância euclidiana entre um perfil organizacional e os recursos candidatos a recomendação. É sugerida também, como trabalho futuro, a investigação de diferentes cálculos de similaridade e seu impacto na acurácia, na revocação e na precisão da recomendação.

Por fim, como trabalho futuro sugere-se explorar recomendar o recurso mais eficiente para a organização baseando-se em outros atributos além do perfil organizacional.

REFERÊNCIAS

- ADOMAVICIUS, G.; TUZHILIN, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. **IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING**, v. 17, n. 6, p. 734–749, 2005.
- ADOMAVICIUS, G.; ZHANG, J. Impact of data characteristics on recommender systems performance. **ACM Trans. Manage. Inf. Syst.**, ACM, New York, NY, USA, v. 3, n. 1, p. 3:1–3:17, abr. 2012. ISSN 2158-656X. Available from Internet: <<http://doi.acm.org/10.1145/2151163.2151166>>.
- AGGARWAL, C. C. Text classification: Basic models. In: _____. **Machine Learning for Text**. Cham: Springer International Publishing, 2018. p. 113–157. ISBN 978-3-319-73531-3. Available from Internet: <https://doi.org/10.1007/978-3-319-73531-3_5>.
- AGGARWAL, C. C.; ZHAI, C. **A SURVEY OF TEXT CLASSIFICATION ALGORITHMS**. 2012.
- AKBARINIA, R.; PACITTI, E.; VALDURIEZ, P. Best position algorithms for efficient top-k query processing. **Information Systems**, v. 36, n. 6, p. 973 – 989, 2011. ISSN 0306-4379. Available from Internet: <<http://www.sciencedirect.com/science/article/pii/S0306437911000445>>.
- ARIAS, M. et al. A framework for recommending resource allocation based on process mining. In: REICHERT, M.; REIJERS, H. A. (Ed.). **Business Process Management Workshops**. Cham: Springer International Publishing, 2016. p. 458–470. ISBN 978-3-319-42887-1.
- BEHESHTI, S.-M.-R. et al. Business process paradigms. In: _____. **Process Analytics: Concepts and Techniques for Querying and Analyzing Process Data**. Cham: Springer International Publishing, 2016. p. 19–60. ISBN 978-3-319-25037-3. Available from Internet: <https://doi.org/10.1007/978-3-319-25037-3_2>.
- BRUNO, N.; CHAUDHURI, S.; GRAVANO, L. Top-k selection queries over relational databases: Mapping strategies and performance evaluation. **ACM Trans. Database Syst.**, ACM, New York, NY, USA, v. 27, n. 2, p. 153–187, jun. 2002. ISSN 0362-5915. Available from Internet: <<http://doi.acm.org/10.1145/568518.568519>>.
- CASTRO, C. L. d.; BRAGA, A. P. Aprendizado supervisionado com conjuntos de dados desbalanceados. Sba: Controle Automação Sociedade Brasileira de Automatica, p. 441–466, 2011.
- CONFORT, V. T. F. **The BPM issues in Brazilian perspective**. Dissertation (Master) — Universidade Federal do Estado do Rio de Janeiro, Brazil, 2016.
- CORTES, C.; VAPNIK, V. Support-vector networks. **Machine Learning**, v. 20, n. 3, p. 273–297, Sep 1995. ISSN 1573-0565. Available from Internet: <<https://doi.org/10.1023/A:1022627411411>>.
- COVER, T.; HART, P. Nearest neighbor pattern classification. **IEEE Trans. Inf. Theor.**, IEEE Press, Piscataway, NJ, USA, v. 13, n. 1, p. 21–27, sep. 2006. ISSN 0018-9448. Available from Internet: <<https://doi.org/10.1109/TIT.1967.1053964>>.

DELICATO, F. C.; PIRES, P. F.; BATISTA, T. The activities of resource discovery and resource estimation. In: _____. **Resource Management for Internet of Things**. Cham: Springer International Publishing, 2017. p. 33–44. ISBN 978-3-319-54247-8. Available from Internet: <https://doi.org/10.1007/978-3-319-54247-8_4>.

DRAHEIM, D. Business process excellence. In: _____. **Business Process Technology: A Unified View on Business Processes, Workflows and Enterprise Applications**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 11–43. ISBN 978-3-642-01588-5. Available from Internet: <https://doi.org/10.1007/978-3-642-01588-5_2>.

DUMAS, M. et al. **Fundamentals of Business Process Management**. [S.l.]: Springer, 2013. I-XXVII, 1-399 p. ISBN 978-3-642-33142-8.

EVGENIOU, T.; PONTIL, M. Support vector machines: Theory and applications. v. 2049, p. 249–257, 01 2001.

FLORES, F. N.; MOREIRA, V. P.; HEUSER, C. A. Assessing the impact of stemming accuracy on information retrieval. In: PARDO, T. A. S. et al. (Ed.). **Computational Processing of the Portuguese Language**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 11–20. ISBN 978-3-642-12320-7.

GARCÍA, J. M. et al. Integrating semantic web services ranking mechanisms using a common preference model. **Know.-Based Syst.**, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 49, p. 22–36, sep. 2013. ISSN 0950-7051. Available from Internet: <<http://dx.doi.org/10.1016/j.knosys.2013.04.007>>.

HALL, M. et al. The weka data mining software: An update. **SIGKDD Explor. Newsl.**, ACM, New York, NY, USA, v. 11, n. 1, p. 10–18, nov. 2009. ISSN 1931-0145. Available from Internet: <<http://doi.acm.org/10.1145/1656274.1656278>>.

HUYCK, C.; ORENGO, V. A stemming algorithm for the portuguese language. In: **String Processing and Information Retrieval, International Symposium on(SPIRE)**. [s.n.], 2001. v. 00, p. 0186. Available from Internet: <doi.ieeecomputersociety.org/10.1109/SPIRE.2001.10024>.

ISINKAYE, F.; FOLAJIMI, Y.; OJOKOH, B. Recommendation systems: Principles, methods and evaluation. **Egyptian Informatics Journal**, v. 16, n. 3, p. 261 – 273, 2015. ISSN 1110-8665. Available from Internet: <<http://www.sciencedirect.com/science/article/pii/S1110866515000341>>.

JAISWAL, R.; LOKHANDE, S. Erratum: Analysis of early traffic processing and comparison of machine learning algorithms for real time internet traffic identification using statistical approach. In: KUNDU, M. K. et al. (Ed.). **Advanced Computing, Networking and Informatics- Volume 2**. Cham: Springer International Publishing, 2014. p. E1–E1. ISBN 978-3-319-07350-7.

KITCHENHAM, B.; CHARTERS, S. **Guidelines for performing Systematic Literature Reviews in Software Engineering**. 2007.

KOTKOV, D.; WANG, S.; VEIJALAINEN, J. A survey of serendipity in recommender systems. **Knowledge-Based Systems**, v. 111, p. 180 – 192, 2016. ISSN 0950-7051. Available from Internet: <<http://www.sciencedirect.com/science/article/pii/S0950705116302763>>.

- LI, H.; CAI, F.; LIAO, Z. Content-based filtering recommendation algorithm using hmm. In: **2012 Fourth International Conference on Computational and Information Sciences**. [S.l.: s.n.], 2012. p. 275–277.
- LI, H.; CHEN, Q.; WANG, X. An improved method for semantic similarity calculation based on stop-words. In: WANG, X. et al. (Ed.). **Machine Learning and Cybernetics**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. p. 339–347. ISBN 978-3-662-45652-1.
- LOPER, E.; BIRD, S. Nltk: The natural language toolkit. In: **Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1**. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002. (ETMTNLP '02), p. 63–70. Available from Internet: <<https://doi.org/10.3115/1118108.1118117>>.
- MATEJA, K. et al. Business process model and notation: The current state of affairs. **Computer Science and Information Systems**, v. 12, p. 509, 539, 2015.
- MENDLING, J.; REIJERS, H. A.; AALST, W. M. P. van der. Seven process modeling guidelines (7pmg). **Inf. Softw. Technol.**, Butterworth-Heinemann, Newton, MA, USA, v. 52, n. 2, p. 127–136, feb. 2010. ISSN 0950-5849. Available from Internet: <<http://dx.doi.org/10.1016/j.infsof.2009.08.004>>.
- MOHAPATRA, S. Business process management (process life cycle, process maturity). In: _____. **Business Process Reengineering: Automation Decision Points in Process Reengineering**. Boston, MA: Springer US, 2013. p. 69–94. ISBN 978-1-4614-6067-1. Available from Internet: <https://doi.org/10.1007/978-1-4614-6067-1_4>.
- MUEHLEN, M. z.; HO, D. T.-Y. Risk management in the bpm lifecycle. In: BUSSLER, C. J.; HALLER, A. (Ed.). **Business Process Management Workshops**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 454–466. ISBN 978-3-540-32596-3.
- MULROW, C. D. Systematic reviews: Rationale for systematic reviews. **BMJ**, BMJ Publishing Group Ltd, v. 309, n. 6954, p. 597–599, 1994. ISSN 0959-8138. Available from Internet: <<https://www.bmj.com/content/309/6954/597>>.
- MUTSCHLER, B.; REICHERT, M. Understanding the costs of business process management technology. In: _____. **Business Process Management: Theory and Applications**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 157–194. ISBN 978-3-642-28409-0. Available from Internet: <https://doi.org/10.1007/978-3-642-28409-0_7>.
- PAASS, G.; KINDERMANN, J.; LEOPOLD, E. Text classification of news articles with support vector machines. In: SIRMAKESSIS, S. (Ed.). **Text Mining and its Applications**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. p. 53–64. ISBN 978-3-540-45219-5.
- PAICE, C. D. An evaluation method for stemming algorithms. In: CROFT, B. W.; RIJSBERGEN, C. J. van (Ed.). **SIGIR '94**. London: Springer London, 1994. p. 42–50. ISBN 978-1-4471-2099-5.
- QUINLAN, R. **C4.5: Programs for Machine Learning**. San Mateo, CA: Morgan Kaufmann Publishers, 1993.

RECKER, J. C. Opportunities and constraints : the current struggle with bpmn. **Business Process Management Journal**, Emerald, v. 16, n. 1, p. 181–201, 2010. Available from Internet: <<http://eprints.qut.edu.au/20316/>>.

RICCI, F.; ROKACH, L.; SHAPIRA, B. Recommender systems: Introduction and challenges. In: _____. **Recommender Systems Handbook**. Boston, MA: Springer US, 2015. p. 1–34. ISBN 978-1-4899-7637-6. Available from Internet: <https://doi.org/10.1007/978-1-4899-7637-6_1>.

ROSSUM, G. **Python Reference Manual**. Amsterdam, The Netherlands, The Netherlands, 1995.

ROYCE, G.; GUPTA, U. G. Resource estimation of information systems projects: A case study. *Journal of International Information Management*, volume='9', number=, 2000.

RUSSELL, S.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. 3rd. ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2009. ISBN 0136042597, 9780136042594.

SALZBERG, S. L. C4.5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993. **Machine Learning**, v. 16, n. 3, p. 235–240, Sep 1994. ISSN 1573-0565. Available from Internet: <<https://doi.org/10.1007/BF00993309>>.

SANCHEZ-BOCANEGRA, C. L. et al. Introduction on health recommender systems. In: _____. **Data Mining in Clinical Medicine**. New York, NY: Springer New York, 2015. p. 131–146. ISBN 978-1-4939-1985-7. Available from Internet: <https://doi.org/10.1007/978-1-4939-1985-7_9>.

SARWAR, B. et al. **Analysis of Recommendation Algorithms for E-Commerce**. 2000.

SATTAR, A.; GHAZANFAR, M. A.; IQBAL, M. Building accurate and practical recommender system algorithms using machine learning classifier and collaborative filtering. **Arabian Journal for Science and Engineering**, v. 42, n. 8, p. 3229–3247, Aug 2017. ISSN 2191-4281. Available from Internet: <<https://doi.org/10.1007/s13369-016-2410-1>>.

SINGH, J.; GUPTA, V. A systematic review of text stemming techniques. **Artificial Intelligence Review**, v. 48, n. 2, p. 157–217, Aug 2017. Available from Internet: <<https://doi.org/10.1007/s10462-016-9498-2>>.

SMOLA, A.; S.V.N.VISHWANATHAN. **An Introduction to Machine Learning**. [S.l.]: Cambridge University Press, 2008.

SORDO, M.; ZENG, Q. On sample size and classification accuracy: A performance comparison. In: OLIVEIRA, J. L. et al. (Ed.). **Biological and Medical Data Analysis**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. p. 193–201. ISBN 978-3-540-31658-9.

ŠTAJNER, T.; MLADENIĆ, D.; GROBELNIK, M. Information resource recommendation in knowledge processes. In: GARCÍA-CASTRO, R.; FENSEL, D.; ANTONIOU, G. (Ed.). **The Semantic Web: ESWC 2011 Workshops**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 186–193. ISBN 978-3-642-25953-1.

VEENA, K. M.; SHENOY, K. M.; SHENOY, K. B. A. Performance comparison of machine learning classification algorithms. In: SINGH, M. et al. (Ed.). **Advances in Computing and Data Sciences**. Singapore: Springer Singapore, 2018. p. 489–497. ISBN 978-981-13-1813-9.

VIERA, A. F. G.; VIRGIL, J. Uma revisão dos algoritmos de radicalização em língua portuguesa. **Information Research**, 2007. Available from Internet: <<http://InformationR.net/ir/12-3/paper315.html>>.

VOSSEN, G. Database transaction models. In: _____. **Computer Science Today: Recent Trends and Developments**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995. p. 560–574. ISBN 978-3-540-49435-5. Available from Internet: <<https://doi.org/10.1007/BFb0015267>>.

WOITSCH, R.; UTZ, W. Business process as a service (bpaas). In: JANSSEN, M. et al. (Ed.). **Open and Big Data Management and Innovation**. Cham: Springer International Publishing, 2015. p. 435–440. ISBN 978-3-319-25013-7.

XIAO, Z.; MING, Z. A method of workflow scheduling based on colored petri nets. **Data Knowl. Eng.**, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 70, n. 2, p. 230–247, feb. 2011. ISSN 0169-023X. Available from Internet: <<http://dx.doi.org/10.1016/j.datak.2010.11.005>>.

ZHAI, C.; LAFFERTY, J. A study of smoothing methods for language models applied to information retrieval. **ACM Trans. Inf. Syst.**, ACM, New York, NY, USA, v. 22, n. 2, p. 179–214, abr. 2004. ISSN 1046-8188. Available from Internet: <<http://doi.acm.org/10.1145/984321.984322>>.

ANEXO I - CÓDIGO PARA CLASSIFICAÇÃO DE TIPO DE RECURSOS

```

import math
import unicodecode
import copy
import xml.etree.ElementTree as ET
from nltk.stem import RSLPStemmer
from nltk.corpus import stopwords

from .models import Task

from django.db.models import Q
from django.core.paginator import Paginator
from resources.models import ResourceType

class ServicesProfiles(object):

    stopwords = set(stopwords.words('portuguese'))
    porter = RSLPStemmer()

    def parse_file(self, instance):
        task_strings = ['task', 'Task']
        tree = ET.parse(instance.raw_file)
        root = tree.getroot()
        for child in root:
            for subchild in child:
                if any(string in subchild.tag for string in task_strings):
                    print(subchild.attrib)
                    self._save_task(subchild.tag, subchild.attrib, instance)
        return instance.raw_file

    def _save_task(self, tag, attrib, instance):
        """
        Saves the tasks as part of the instance
        """
        task_types_map = {
            'businessRuleTask': Task.BUSINESS_RULE_TASK,
            'userTask': Task.USER_TASK,
            'scriptTask': Task.SCRIPT_TASK,
            'serviceTask': Task.SERVICE_TASK,
            'sendTask': Task.SEND_TASK,
            'receiveTask': Task.RECEIVE_TASK,
            'task': Task.TASK,
            'manualTask': Task.MANUAL_TASK
        }

        task_type = task_types_map[tag.split('.')[1]]
        label = attrib['name']
        Task.objects.create(label=label, task_type=task_type, process=instance)

    def _get_probability_by_resource(self, training):
        """
        Returns:
        dict: probability that a word belongs to a resource type group of words
        """
        priori_prob_by_resource_types = {}
        for resource in ResourceType.objects.all():
            resource_task_count = resource.task_set.filter(id__in=[i.id for i in training]).count()
            priori_prob_by_resource_types[resource.name] = resource_task_count

        task_count = len(training)
        tasks_without_resource = len([task for task in training if task.application_type is None])
        for row in priori_prob_by_resource_types:
            priori_prob_by_resource_types[row] = priori_prob_by_resource_types[row] / task_count
        priori_prob_by_resource_types['undefined'] = tasks_without_resource / task_count
        return priori_prob_by_resource_types

    def _get_all_words_from_each_resource(self, training):

```

```

"""
Returns:
    dict: for each resource type, the words that are part of the resource type tasks
"""
all_words_from_a_resource = {'undefined': []}
for task in training:
    cleaned_label = self._clean_label(task.label)
    application_name = task.application_type.name if task.application_type else 'undefined'
    all_words_from_a_resource.setdefault(application_name, [])
    all_words_from_a_resource[application_name] = all_words_from_a_resource[application_name] + cleaned_label
return all_words_from_a_resource

def _get_word_counter_by_resource_type(self, all_words_from_a_resource):
    """
Returns:
    dict: number of times a given word appears in a given resource type
"""
word_counter = {}
for resource in all_words_from_a_resource:
    word_counter[resource] = {}
    for word in all_words_from_a_resource[resource]:
        word_counter[resource][word] = all_words_from_a_resource[resource].count(word)
return word_counter

def _calculate_likelihood(self, prob_condit, set_of_unique_words_all_docs, number_unique_words_all_docs,
    all_words_from_a_resource):
    """
Calculates likelihood of word - resource type
"""
for resource in prob_condit:
    for word in set_of_unique_words_all_docs:
        p_word = prob_condit[resource][word]+1 if word in prob_condit[resource] else 1 # laplace smoothing
        number_of_all_words_in_category = len(all_words_from_a_resource[resource])
        prob_condit[resource][word] = p_word/(number_of_all_words_in_category+number_unique_words_all_docs)
return prob_condit

def classify(self, test, training):

    priori_prob_by_resource_types = self._get_probability_by_resource(training)
    all_words_from_a_resource = self._get_all_words_from_each_resource(training)

    set_of_unique_words_all_docs = set()
    for i in all_words_from_a_resource:
        set_of_unique_words_all_docs.update(all_words_from_a_resource[i])

    word_counter = self._get_word_counter_by_resource_type(all_words_from_a_resource)

    prob_condit = copy.deepcopy(word_counter)
    number_unique_words_all_docs = len(set_of_unique_words_all_docs)

    prob_condit = self._calculate_likelihood(
        prob_condit,
        set_of_unique_words_all_docs,
        number_unique_words_all_docs,
        all_words_from_a_resource
    )
    self._classify_test_set(test, prob_condit, set_of_unique_words_all_docs, priori_prob_by_resource_types)

def _classify_test_set(self, test, prob_condit, unique_words, priori):
    probability = {}
    for task in test:
        probability[task.label] = {}
        for app_class in prob_condit:
            label = self._clean_label(task.label)
            probability[task.label][app_class] = priori[app_class]
            existent_words = [i for i in label if i in unique_words]
            if len(existent_words) == 0:
                probability[task.label][app_class] = 0

        else:

```

```

        for word in label:
            if word in prob_condit[app_class]:
                probability[task.label][app_class] = probability[task.label][app_class] * prob_condit[app_class][word]

for label in probability:
    max_val = 0
    class_value = ''
    for app in probability[label]:
        if probability[label][app] > max_val:
            max_val = probability[label][app]
            class_value = app
    if class_value != '' and class_value != 'undefined':
        tasks_to_update = [task for task in test if task.label == label]
        # cleanup para checar se task um servico ou manual: se for, nao classifica
        tasks_to_update = [task for task in tasks_to_update if task.task_type != 'service' and
                            task.task_type != 'manual']
        for task in tasks_to_update:
            task.classified_type = ResourceType.objects.get(name=class_value)
            task.save()

def _clean_label(self, label):
    """
    Returns a list of the words that compose the provided label
    """
    list_of_words = []
    for word in label.split():
        stemmed_word = self.porter.stem(word.lower())
        new_word = unicode.decode(stemmed_word).replace('-', '')
        if new_word not in self.stopwords:
            list_of_words.append(new_word)
    return list_of_words

def cross_validation(self, organization_pk):
    all_tasks = Task.objects.filter(process__organization__id=organization_pk).order_by('?')
    all_tasks.update(classified_type=None)
    p = Paginator(all_tasks, math.ceil(len(all_tasks)/10))
    matches = 0
    mismatches = 0
    for i in range(10):
        page = p.page(i+1)
        test = page.object_list
        training = all_tasks.filter(~Q(id__in=[task.id for task in test]))
        self.classify(test, training)
        for t in test:
            print('%-60s%-25s%-25s' % (t.label, t.application_type, t.classified_type))
            if t.classified_type == t.application_type:
                matches = matches + 1
            else:
                mismatches = mismatches + 1
    accuracy = matches/(matches+mismatches)
    return accuracy

```