UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE COMPUTAÇÃO
CURSO DE ENGENHARIA DA COMPUTAÇÃO

DAVI SHINJI MOTA KAWASAKI

CLASSIFICAÇÃO DE QUESTÕES DE EXAMES DA ÁREA DA COMPUTAÇÃO UTILIZANDO PROCESSAMENTO DE LINGUAGEM NATURAL

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO
2018

DAVI SHINJI MOTA KAWASAKI

CLASSIFICAÇÃO DE QUESTÕES DE EXAMES DA ÁREA DA COMPUTAÇÃO UTILIZANDO PROCESSAMENTO DE LINGUAGEM NATURAL

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia da Computação da Universidade Tecnológica Federal do Paraná, como requisito parcial para a obtenção do título de Bacharel.

Orientador: Willian Massami Watanabe
Universidade Tecnológica Federal do Paraná

CORNÉLIO PROCÓPIO
2018

Dedico esse trabalho a Deus e para minha família. Confio tudo Nele e para Ele.

AGRADECIMENTOS

Agradeço, primeiramente, a Deus por ter sempre me direcionado conforme às Suas vontades. Além Dele nada disso poderia ter sido possível sem toda a ajuda e colaboração da minha família. Obrigado Isabel, Celso e Natan por todo incentivo e apoio dado para todas as minhas lutas. Sem vocês não teria alcançado mais esse objetivo. E por fim, agradeço ao meu orientador, Willian Massami Watanabe, pelo auxílio e ajuda nesse e em outros trabalhos.

RESUMO

KAWASAKI, Davi. Classificação de questões de exames da área da computação utilizando Processamento de Linguagem Natural. 2018. 46 f. Trabalho de Conclusão de Curso – Curso de Engenharia da Computação, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2018.

O atraso nos modelos da educação curricular tem sido discutido e questionado, visando mudanças para atender demandas que auxiliem e motivem o estudante na retenção de conteúdos com grande quantidade de informações. Aulas disruptivas e inovadoras, juntamente com abordagens heterodoxas e do ensino adaptativo têm se apresentado como alternativas para aprimoramento do desempenho acadêmico dos alunos. Tais iniciativas têm obtido um alcance no âmbito escolar a partir de tecnologias como mineiração de dados e inteligência artificial, mas ainda não possui uma atuação suficientemente necessária na área computacional do conhecimento. Sendo assim, esse trabalho teve como objetivo o desenvolvimento e avaliação de modelos de classificação de questões de exames da área de computação na língua portuguesa, trabalhando com técnicas de processamento de linguagem natural e aprendizado de máquina. A partir da contextualização educacional e da atuação do ensino adaptativo, o trabalho realiza uma abordagem desde a aquisição de questões até realização de aprendizados supervisionados. Por fim, a análise de modelos de amostras balanceadas apresentou uma média de 79% de precisão na classificação e uma precisão de 83% para Redes *Bayesianas* Multinomiais. Além da utilização desse modelo para futuras classificações de novas questões em um sistema adaptativo, o trabalho também conseguiu disponibilizar uma banco de dados de questões extraídas e classificadas para instituições computacionais de ensino, com um total de 20 temáticas e 28745 questões com texto principal, perguntas e respostas.

Palavras-chave: Processamento de Linguagem Natural. Classificação de Questões Computacionais. Aprendizado de Máquina. Ensino Adaptativo.

ABSTRACT

KAWASAKI, Davi. Computing Exam Questions Classification using Natural-Language Processing. 2018. 46 f. Trabalho de Conclusão de Curso – Curso de Engenharia da Computação, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2018.

The delay on curricular education models have been discussed and questioned. These questionings aim to meet demands that help and motivate students in the comprehension of subjects with high information quantity. Disruptive, innovative, unorthodox approaches and adaptative teaching have been presented as alternatives for students' academic achievement improvements. These approaches have gotten into the scholar scope through technologies such as data mining and artificial intelligence, though the reach wasn't fairly enough in the computer knowledge area. Therefore this work's aimed to develop and evaluate classification models of computer knowledge questions in the portuguese language, using Natural-Language Processing and Machine Learning as processing techniques. Through an educational context and adaptive learning usage, the work develops an approach that goes from questions acquisition to the realization of supervised learning methods. Ultimately, the final balanced model analysis presented a precision average of 79% for all methods and a precision of 83% for Multinomial Naive Bayes. Besides using output model for future questions classification in an adaptive system, the work also managed to make a question database available with extracted and classified questions for computer teaching institutions. The total extraction was of 20 themes and 28745 questions with main text, alternatives and answers.

Keywords: Natural-Language Processing. Computing Question Classification. Machine Learning. Adaptative Teaching.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

BOW	*Bag-of-Words* - Saco de Palavras
ENADE	Exame Nacional de Desempenho dos Estudantes
FN	*False Negative* - Falsos Negativos
FP	*False Positive* - Falsos Positivos
HESA	*Higher Education Statistics Agency*
IA	Inteligência Artificial
LR	*Logistic Regression* - Regressão Logística
NB	*Network Bayesian* - Redes *Bayesianas*
NLTK	*Natural Language Toolkit* - Kit de Ferramentas de Linguagem Natural
NN	*Near Neighbors* - Vizinhos Próximos
OCR	*Optical Character Recognition* - Reconhecimento Ótico de Caracteres
PCA	*Principal Component Analysis* - Análise de Componentes Principais
PLN	Processamento de Linguagem Natural
PP	*Predictive Positive* - Positivos Previstos
QA	*Question Answering* - Perguntas-Respostas
RBF	*Radial Basis Function* - Função de Base Radial
RI	Recuperação da Informação
RP	*Real Positive* - Reais Positivos
SVM	*Support Vector Machine*
TF-IDF	*Frequency-Inverse Document Frequency* - Frequência do Termo-Inverso em Documentos
TN	*True Negative* - Verdadeiros Negativos
TP	*True Positive* - Verdadeiros Positivos

SUMÁRIO

1 INTRODUÇÃO

A educação, presente na vida de cada ser humano antes mesmo da alfabetização, consiste em uma das bases fundamentais da sociedade humana. Junto de outros pilares governamentais, a educação vai além da instituição escolar, estando enraizada em fatores familiares. Essa confluência da diversidade de atuação da educação pode ser analisada pela afirmação de que, de forma onipresente ou discreta, agradável ou ameaçadora, ela faz parte da vida cotidiana de cada família (MONTANDON; PERRENOUD, 1987).

Ao chegar na atual era da informação, a educação passou por várias mudanças e contextos históricos, mas ainda apresenta ampla discussão em função da sua importância na sociedade. Esse trabalho realizou, portanto, um experimento com a tecnologia de PLN, verificando a relevância de seus resultados em ensinos adaptativos. Sendo assim, esse capítulo apresenta, primeiramente, uma contextualização histórica da educação, para então instigar a motivação geral referente à realização do trabalho. Por fim, apresenta-se o objetivo geral com uma abordagem mais técnica e elabora-se a estrutura realização do trabalho.

1.1 CONTEXTUALIZAÇÃO

Embora possua entendimentos e conotações direcionados aos atuais sistemas metodológicos de ensino, a educação possui origem em um aprendizado próprio, humano por natureza, baseado em explorações na cultura da caça (GRAY, 2008). Crianças nesse ambiente possuíam um aprendizado livre e natural, obtido a partir das próprias experiências no campo.

Após uma transição para um aprendizado voltado ao trabalho escravo durante a Idade Média, a ideologia de uma educação obrigatória e universal começou a ser difundida entre os séculos XVI e XIX, períodos os quais foram marcados pelas insurgências e discordâncias religiosas. Esse ímpeto de uma educação universal foi influenciado, em uma boa parcela, pelos reformistas de religiões protestantes emergentes, difundindo que infância e adolescência deveriam ser períodos de estudo em escolas reconhecidas como locais de aprendizado (GRAY, 2008), estabelecendo um modelo tradicional com um responsável - no caso um professor - como ator principal e expositor dos conteúdos.

Apesar da alta preocupação dos reformistas com relação ao futuro das crianças, os mesmos estabeleceram o aprendizado dos infantos de acordo com seus próprios interesses, visando a formação de futuros estudiosos por meio do estímulo da memória por meio da repetição e inúmeros testes (GRAY, 2008).

Embora as intenções dos reformistas fossem as melhores, a preocupação e rigorosidade das instituições fundadas levaram, justamente, à prática do pilar inicial de defesa dessas crianças: repúdio à violência e aos maus tratos praticados durante o trabalho escravo da Idade Média. Essa autoridade das instituições de ensino do século XVIII refletia-se nas regras na escola de

John Wesley, a partir da declaração de que assim como o ser humano não possui dias de gozo, também não deve ser permitido período algum para tal prática; pois aquele que se divertir como criança fará o mesmo como homem (MULHERN, 1959).

Assim como o trabalho escravo consistia de tarefas tediosas para as crianças, as aulas curriculares não podiam ser diferentes. Com o passar dos séculos XIX e XX, o currículo escolar tem se tornado cada vez mais abrangente e extenso para concentrar o fluxo imenso de informações presentes na era da informação (GRAY, 2008). Essa sobrecarga de conteúdos exige uma dedicação, em muito dos casos, integral dos alunos, concentrando matérias em curtos espaços de tempo de aula.

O problema desse fluxo imenso de informações e tarefas consiste na geração de grande sobrecarga na *memória de trabalho* dos alunos, geralmente causada por fatos sem conexões, instruções em vários estágios e pela aplicação quase instantânea de conceitos aprendidos recentemente (WILLINGHAM, 2009). Ao juntar a sobrecarga com a pressão constante, a insegurança e a ansiedade social, o estudante tende se a sentir desmotivado e confuso, levando-o a não conseguir reter o conteúdo repassado.

Esse modelo de ensino, com fortes estruturas ainda remanescentes dos séculos XVI a XIX, possui uma construção curricular e uma metodologia de ensino delineada e construída por especialistas dominantes de um assunto específico (CHRISTENSEN, 2008). Isso significa, por exemplo, que conteúdos respectivos à área de matemática sejam construídos pela perspectiva de professores de exatas, normalmente direcionada com termos teóricos e densos. Essa exclusividade tende a privilegiar os alunos que tenham facilidade para o conteúdo específico em questão, e não-intencionalmente desmotiva outros que possuam outros interesses e capacidades (CHRISTENSEN, 2008).

Em função dessa desmotivação em matérias e conteúdos específicos, pesquisadores de trabalhos como (WILLINGHAM, 2009) e (GARDNER, 1983) desenvolveram, a partir de pesquisas cognitivas, a teoria das sete inteligências. Essa teoria apresenta que estudantes aprendem de maneiras distintas e poderiam ser melhores servidos se (a) as disciplinas pudessem ser apresentadas de inúmeras formas e (b) o aprendizado pudesse ser avaliado por uma variedade de métodos. Esse aprendizado diversificado é apresentado pela teoria, explicando que somos capazes de conceber o mundo pela linguagem, pela análise lógica-matemática, pela representação espacial, pelo pensamento musical, pelo uso do corpo para resolução de problemas ou criação de artefatos, e pela compreensão interpessoal e intrapessoal (GARDNER, 1983).

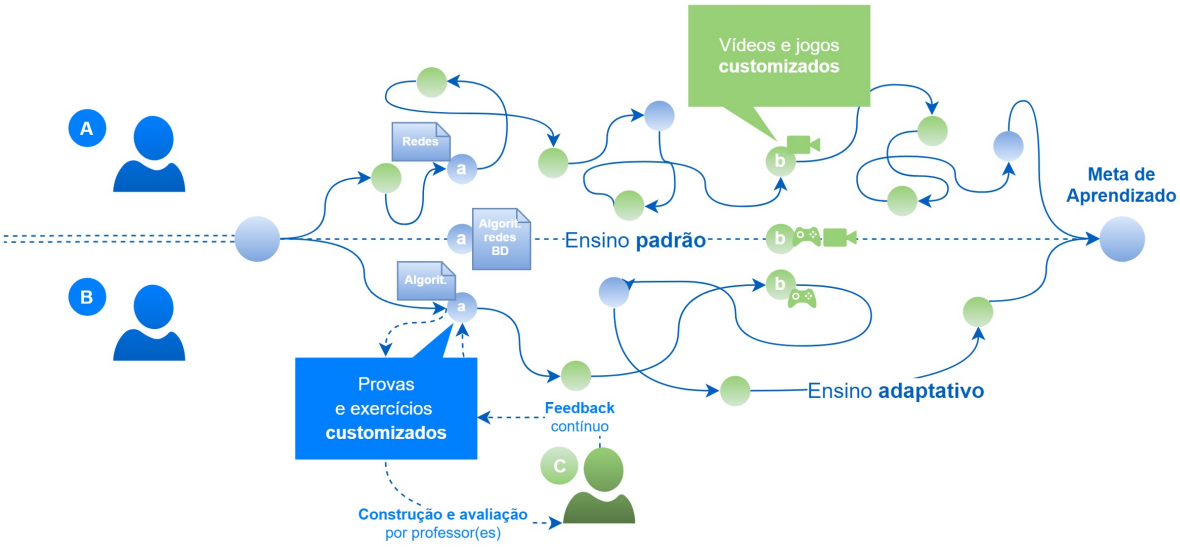
Essa manifestação dos diversos tipos de inteligência para cada estudante tem sido uma motivação relevante para instituições de ensino trabalharem com aulas disruptivas e inovadoras (CHRISTENSEN, 2008). Escolas como o Instituto Lumiar, eleita em 2007 como uma das 12 escolas mais inovadoras do mundo, trabalham um pouco a cada dia com essa mentalidade em abordagens heterodoxas, seguindo a ideia de uma construção de currículo desacoplado da divisão tradicional em disciplinas fixas e com foco na construção e expansão

de competências e habilidades (CARVALHO, 2015). A *Kroton*, maior grupo educacional do mundo, e a *startup Geekie*, por exemplo, são instituições que têm trabalhado na reorientação da prática e implementação de um currículo pedagógico utilizando o ensino adaptativo.

1.2 MOTIVAÇÃO

Explorando a vertente da tecnologia educacional, o ensino adaptativo visa a melhoria do aprendizado individualizado, tentando identificar momentos e fatores que cada aluno tende a aprender com maior facilidade (LORENZONI, 2016). Em contraposição a aulas direcionadas a massas em salas de aulas, essa metodologia tecnológica identifica métodos de apoio individual com embasamento em dados, identificando e mapeando, por meio da interação, acertos e erros dos alunos, além de pontos fortes e de melhoria no conteúdo de cada aluno.

Figura 1 – Abordagem representativa entre o ensino adaptativo e modelo padrão de ensino



Fonte: Autoria Própria

Conforme pode ser visto na Figura 1, o ensino adaptativo trabalha "a linhas tortas", com caminhos totalmente distintos e customizados para cada aluno. Conforme pode ser visualizado nas etapas (a) e (b), o conteúdo apresentado e reforçado é direcionado conforme o mapeamento de pontos fortes e fracos de cada aluno, reforçando pontos de inteligência que o mesmo precisa aprimorar. Por exemplo, ao invés do aluno B se dedicar, na etapa (a), em provas e exercícios de três temáticas computacionais segundo o ensino padrão, o mesmo pode se dedicar apenas na temática que apresenta maior dificuldade, e estimulá-la através de mídias que atuem em uma das sete inteligências que o mesmo possui uma maior capacidade de aprendizagem.

A *startup Studiare*, comprada pela *Kroton* em outubro de 2015, conseguiu alcançar, através do método de Ensino Adaptativo, uma melhoria de 35% na performance acadêmica dos seus 200.000 alunos (BIGARELLI, 2016), utilizando planos individuais de estudos para provas do ENADE 2015. Já a *startup Geekie*, por sua vez, alcançou um aumento de 30% na

performance em provas digitais por parte dos alunos da escola *Mater Dei* entre os anos 2015 e 2016 (RIGBY, 2016).

Figura 2 – Resultados da aplicação do Ensino Adaptativo no mercado

Fonte: (BIGARELLI, 2016) e (RIGBY, 2016)

Além de poder auxiliar no aprimoramento do desempenho acadêmico, o ensino adaptativo tem o potencial de resolver problemas de retenção e aumentar a porcentagem de graduações. Sistemas adaptativos e de gestão escolar como *LeanSmart* e *ALEKS* da *McGraw-Hill Education* conseguiram, por exemplo, melhorar notas acadêmicas em pelo menos um nível, alcançar taxas de retenção escolar em mais de 10% e uma taxa de graduação próximas do 90% (MCGRAW-HILL, 2015).

O quadro educacional em cursos da área tecnológica apresentam resultados desfavoráveis com relação à taxas de retenção de alunos. De acordo com pesquisas realizadas pelo INEP no Brasil em 2011 e pela *HESA* - Agência de Estatísticas do Ensino Superior em 2016, os cursos relacionados a área de Tecnologia da Informação se posicionam entre os que possuem maiores índices de evasão no ensino superior. No Brasil, as graduações de Processamento da Informação e Ciências da Computação são os de maiores taxas em 2011, com, respectivamente, 36% e 32% (SIMAS, 2012). Já na Inglaterra as taxas são menores, mas considera a mais alta a área geral de computação, a qual alcança 10,7% de desistências no primeiro ano, 3% acima do curso de Publicidade, a qual posiciona a segunda posição da lista (LEE, 2017).

A partir da análise realizada, esse trabalho tem como motivação macro futura em contribuir na inversão da atual situação educacional em que se encontram graduações de computação no nível superior, principalmente na área da computação. Para que esse objetivo almejado seja alcançado em um futuro próximo, o trabalho tem como motivação específica a evolução da classificação de questões utilizadas em sistemas adaptativos na língua portuguesa.

1.3 OBJETIVO GERAL

Como apresentado, o ensino adaptativo já tem sido utilizado em algumas escolas do ensino médio e do ensino superior, utilizando algoritmos que analisam desempenhos de estudantes a partir de questões rotuladas de provas, como a do ENADE. Utilizando-se de dados para compreender e captar a melhor forma do desenvolvimento individual, a *Geekie* e a *Studiare*, por exemplo, utilizam-se de algoritmos de inteligência artificial para para a execução do ensino adaptativo, com plataformas em *cloud* que trabalham com *adaptive learning*, *big data*, *data mining*, *analytics*, *blended learning*, estímulos adaptativos.

O objetivo geral desse trabalho é contribuir com um sistema adaptativo que consiste em trabalhar na parte de provas e exercícios customizados, conforme apresentado na etapa (a) da Figura 1. Nessa seção, o trabalho consiste em extrair e categorizar questões de provas utilizadas por diferentes sistemas, focando em possivelmente contribuir na área computacional que possui altas taxas de desistência.

Sendo assim, pretendeu-se desenvolver e avaliar um modelo de classificação taxonômica de questões usadas em avaliações e provas da área da computação, especificamente com questões da língua portuguesa. A intenção consiste em obter um modelo representativo de regras correspondentes a cada temática e subtemáticas, seguindo a seguinte hipótese de pesquisa:

(Q1) A abordagem proposta é capaz de classificar questões de áreas computacionais na língua portuguesa?

De forma complementar, pretendeu-se obter um banco de dados de questões consistentes e rotuladas, o qual pode auxiliar professores e sistemas adaptativos na elaboração de provas direcionadas a cada dificuldade de aluno e possibilitar um *feedback* contínuo apresentado na interação entre professor e aluno na etapa (a) da Figura 1. O resultado geral dessa base de dados foi um total de 20 temáticas e 28745 questões exportadas de dois sites de concurso do Brasil.

1.4 ESTRUTURA GERAL

Para uma melhor compreensão da realização da classificação das questões da área da computação, esse trabalho apresenta no Capítulo 2 a fundamentação teórica necessária. Junto de uma contextualização, em sequência, de trabalhos correlatos ao tema em questão, apresenta-se a proposta principal do trabalho no Capítulo 3, junto da abordagem e da metodologia executada. Por fim, os resultados obtidos são apresentados na Seção 4, e uma discussão final é apresentada no Capítulo 5 a fim de avaliar a contribuição dos resultados e possíveis trabalhos futuros a serem desenvolvidos.

2 FUNDAMENTAÇÃO TEÓRICA

Para conseguir realizar com sucesso a classificação das questões nesse trabalho, é necessário compreender as estratégias e estudos teóricos envolvendo textos de uma linguagem humana. Sendo assim, esse tópico apresenta conceitos básicos do processamento de língua humana e como são realizadas as classificações de documentos e questões por meio do *PLN*. Por fim, alguns trabalhos relacionados a esse trabalho serão descritos para demonstrar possíveis caminhos de desenvolvimento desse projeto.

2.1 PROCESSAMENTO DE LINGUAGEM NATURAL (*PLN*)

Os seres humanos, para conviver em sociedade, desenvolveram na história formas de comunicação, sendo uma delas a forma oral. Por meio dela foram desenvolvidas várias línguas naturais, normalmente adequadas a cada continente ou país que parte da sociedade vivia. A partir das inúmeras linguagens existentes para comunicação humana, surgiu-se o interesse em conseguir, inicialmente, realizar a intercomunicação entre diferentes línguas.

A partir da década de 40, esse interesse se expandiu para a utilização da tecnologia, iniciando com máquinas de tradução até casos mais avançados como reconhecimento de voz (LIDDY, 2001). A área de processamento de linguagem natural, portanto, tem trabalhado com todo tipo de processamento que envolva conceitos e teorias linguísticas, utilizando-se de técnicas computacionais para análise, coleta, compreensão e representação natural de informações de linguagens no meio computacional. A partir dessas tecnologias, o *PLN* tem sido considerado uma subárea da IA, com o objetivo de não apenas conseguir realizar traduções, interpretações de textos e reconhecimentos de voz, mas principalmente almejar uma performance similar à humana (LIDDY, 2001), podendo alcançar estágios cada vez mais próximos de uma interface homem-máquina que passe no teste de *Turing* [1].

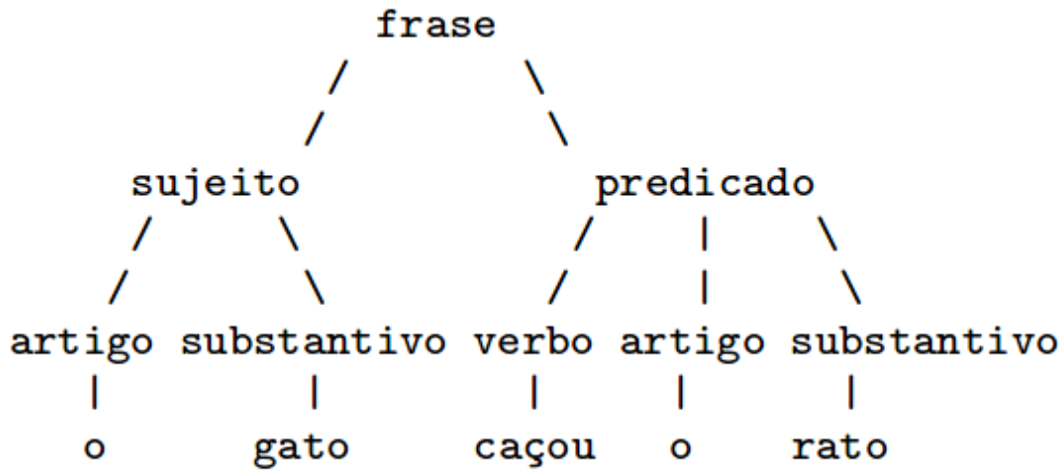
A pesquisa em *PLN* tem um enfoque em três seções da comunicação em linguagem natural, começando na parte da fonologia, a qual trabalha com reconhecimento de sons e vozes; passando pela morfologia e sintaxe para análise estrutural de frases e documentos; até, por fim, trabalhar com a semântica e a pragmática, visando compreender o significado do conteúdo inerente ao texto (COVINGTON; NUTE; VELLINO, 1997). Enquanto a fonologia tem um enfoque maior no reconhecimento dos sons que compõem uma palavra, as outras estruturas entram em uma análise textual específica:

- Morfologia: reconhecimento das palavras em suas formas primitivas, como o radical *fal-* do verbo falar;
- Sintaxe: análise da estrutura relacional de palavras na frase, conforme apresentado na Figura 3;

[1] O teste de *Turing* verifica a capacidade de uma máquina computacional de exibir comportamento inteligente de forma equivalente ou igual a um ser humano.

- Semântica: compreensão do significado de uma estrutura sintática com base na combinação das palavras componentes;
- Pragmática: análise do significado mais apropriado no contexto sintático (PEREIRA, 2018).

Figura 3 – Demonstração de uma árvore sintática de uma frase



Fonte: (PEREIRA, 2018)

Essas classificações, embora sejam bem definidas, são complexas por si só, sendo difícil para humanos em processar e interpretá-las. Sendo assim, o campo de pesquisa do *PLN* começou a utilizar um modelo de processamento adequado para conseguir aprender essas estruturas linguísticas complexas e extensas (MANNING; SCHÜTZE, 1999). Utilizando-se de técnicas estatísticas, o *PLN* conseguiu avanços recentes em pesquisas utilizando-se de ferramentas léxicas como *WordNet* e *corpus* textual padrão, sendo, inclusive, utilizadas em classificações de textos e questões (JAYAKODI; BANDARA; PERERA, 2015).

2.2 CLASSIFICAÇÃO DE TEXTOS

Sendo uma subárea do *PLN*, a classificação de textos trabalha e se baseia na busca, categorização, roteamento de informações em documentos e em sistemas de filtros (LI; JAIN, 1998). Por meio dessas ações, torna-se possível, na categorização de textos, a atribuição de categorias previamente definidas a esses documentos, considerando o conteúdo e suas palavras (GEORGE; JOSEPH, 2014).

O processo padrão de uma classificação de textos consiste em algumas etapas sequenciais, como o pré-processamento dos dados, redução da dimensionalidade da base e, por fim, a classificação em si. No pré-processamento, por exemplo, o texto como um todo é convertido

e processado para treinamento utilizando-se de técnicas como *stopwords*, *stemming* e *BOW*. As técnicas de *stopwords* e *stemming* têm uma correlação na redução da dimensionalidade, sendo a primeira decorrente da remoção de palavras (pronomes, conjunções) não-relevantes para treinamento, enquanto que a segunda tem o objetivo de reduzir palavras flexionadas para a sua forma radical. Já técnica *BOW*, a qual é uma estratégia de *RI*, consiste em uma representação sob outra perspectiva das palavras do documento. A partir de um texto consistente de uma coleção desorganizada de palavras, atribui-se um peso a cada palavra como unigrama dependendo da sua frequência em um documento e entre documentos distintos. Esse conjunto de palavras com pesos, finalmente, formam o(s) vetor(es) de frequência(s) de saco de palavras. Com um processamento consistente e organizado dos dados, o treinamento pode obter acurácicas aceitáveis para categorias desejadas (GEORGE; JOSEPH, 2014).

A partir de algoritmos de busca até outros de aprendizado de máquina, as etapas descritas na classificação de textos apresentam dois grandes desafios. O primeiro desafio, associado ao pré-processamento de dados, considera a dificuldade na obtenção e interpretação, a partir de poucas palavras, de semânticas subjetivas e conceitos abstratos das linguagens naturais. A possibilidade de interpretação de sinônimos e homônimos, por exemplo, podem causar duplas ou confusas interpretações (LI; JAIN, 1998), pois cada uma dessas palavras precisam ser analisadas não apenas no contexto semântico, mas também no pragmático, além de considerar possíveis conotações sarcásticas e/ou irônicas. Passos iniciais já foram trilhados para resolução desses problemas, normalmente envolvendo análise semântica latente, a qual transforma variáveis latentes/escondidas em palavras únicas, tentando reduzir a dimensionalidade do conjunto de dados (LAZYPROGRAMMER, 2016). O segundo, por sua vez associado aos algoritmos de classificação, envolve problemas de alta dimensionalidade e qualidade dos registros, podendo afetar acurácias de resultados de treinamentos supervisionados e não-supervisionados (LI; JAIN, 1998).

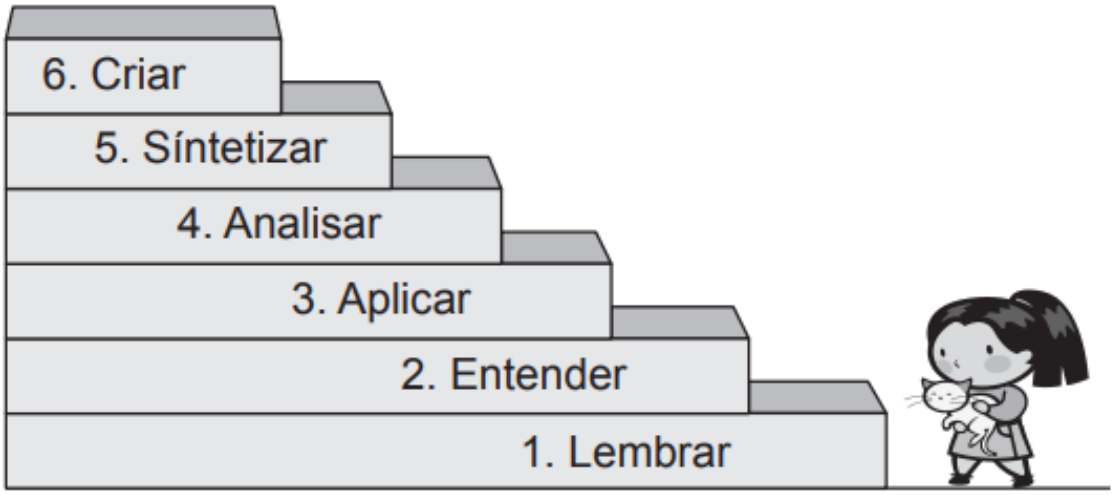
A classificação de textos tem sido intensamente estudada ao considerar estudos em cima de documentos de textos, com uma grande quantidade de palavras. Contudo, outra subárea em vanguarda, muito em função dos sistemas de *QA*, consiste na classificação de questões, com um enfoque reduzido de palavras a serem processadas e classificadas. Além de lidar com o desafio de poucas palavras, esse gênero de classificação textual possui bases de treinamentos reduzidas e precisa de uma atenção especial na etapa de pré-processamento para não comprometer a acurácia dos resultados decorrente de remoção de palavras relevantes (SANGODIAH; AHMAD; AHMAD, 2014).

2.3 TRABALHOS RELACIONADOS

Conforme apresentado na introdução, esse trabalho tem como objetivo trabalhar a extração de questões utilizadas em avaliações computacionais para uma classificação taxonômica baseadas nas principais áreas da computação. Ao analisar, por exemplo, uma classificação taxonômica não restrita a conteúdos e temas computacionais, verificaram-se resultados obtidos

para a classificação de questões em taxonomias educacionais como *Bloom*. O trabalho realizado pelos autores Jayakodi, Bandara e Perera (2015) apresenta a utilização da taxonomia educacional *Bloom* para aplicação de pesos baseados em níveis cognitivos representados pela Figura 4, os quais são representados por verbos identificadores. A partir de um processo de classificação de textos na língua inglesa semelhante ao apresentado na Seção 2.2, o trabalho apresenta o desenvolvimento de uso de regras, as quais são utilizadas para classificação das questões com a colaboração e validação de especialistas de cada área. A partir de 147 questões dos cursos do Departamento de Ciências da Computação e Engenharia da Universidade de Moratuwa, obteve-se uma acurácia final de detecção de 82%.

Figura 4 – Níveis cognitivos da Taxonomia *Bloom*



Fonte: (FERRAZ; BELHOT, 2010)

Outro trabalho correlacionado à taxonomia cognitiva *Bloom* foi apresentado pelos autores Jayakodi, Bandara e Meedeniya (2016), o qual utiliza *WordNet* com o algoritmo de similaridade de cossenos para classificação de 53 questões de três matérias computacionais (programação orientada a objetos, engenharia de *software*, estrutura de dados e de algoritmos). Por meio do auxílio do algoritmo de similaridade, questões que não possuíam verbos também conseguiram ser classificadas de acordo com os níveis cognitivos de *Bloom*, alcançando uma acurácia de 71% questões corretamente classificadas.

Enquanto esses trabalhos envolvendo taxonomias educacionais têm um enfoque na avaliação de desempenho de estudantes, outros artigos foram desenvolvidos para classificações em taxonomias baseadas em *QAs* dos seguintes domínios:

- Temáticas do cotidiano: Fan, Su e Liu (2017);
- Direito: Xiao, Chow e Chen (2017) e Xiao, Mo e Chow (2017);
- Medicina: Athenikos, Han e Brooks (2008), Sarrouti, Lachkar e Ouatik (2015) e Dodiya e Jain (2016);

- Viagem e turismo: Xu, Zhou e Wang (2012), Ying-wei, Zheng-tao e Xiang-yan (2008), Su, Liao e Yu (2009) e Kahaduwa, Pathirana e Arachchi (2017);
- Computação: Rao, S. e Kannan (2016) e Fu, Qu e Wang (2009).

Embora essas pesquisas utilizem questões do tipo *QA* (utilizada para solução de dúvidas gerais), as suas análises permitem elaborar e testar taxonomias através de diferentes tipos de algoritmos de aprendizado de máquina.

O trabalho realizado pelos autores Fan, Su e Liu (2017), por exemplo, envolve a classificação de 2000 questões que podem se enquadrar em múltiplas categorias (*multi-label*) em Mandarim, utilizando-se de uma ferramenta de *deep learning* chamada *word2vec*. Por meio dessa análise complexa, os autores conseguiram obter uma acurácia média de 90,28%, envolvendo categorias como Cultura e Arte, Esportes, Vida Social e Sustento, Ciência e Educação, Economias e Finanças, Leis e Regulações. Já o trabalho dos autores Xiao, Chow e Chen (2017), por sua vez, filtrou as categorias para o ramo do Direito, dividindo e classificando 52834 questões das subáreas bases do sistema legal chinês em Mandarim: Civil, Criminal e Administrativa.

Além do uso da técnica de *word2vec*, o artigo discute e utiliza a utilização da *TF-IDF* como *features* da classificação, considerando também a utilização de algoritmos de aprendizado de máquina como *NN*, *NB*, *LR* e *SVM*. Por meio de um resultado de acurácia de 95,75%, o trabalho consegue propor uma taxonomia para questões de Direito, o mais próximo ao objetivo principal deste trabalho.

Por fim, o trabalho realizado pelos autores Fu, Qu e Wang (2009) envolve a divisão e classificação de 11000 questões do domínio de suporte e serviços computacionais em Mandarim. O trabalho apresenta duas comparações de treinamentos: o primeiro comparando tipos de características (*BOW* e conceitos hierárquicos de ontologia - hierarquia de conceitos específicos de um domínio) e o segundo os tipos de algoritmos de treinamento (*NB*, *SVM* e combinação entre *SVM* e similaridade semântica de questões). Pela análise dos resultados, os autores concluíram que conceitos hierárquicos de ontologia influenciam em melhores resultados de treinamento (91,5% em contraposição à 76,4%) e que a combinação do algoritmo de *SVM* com similaridade semântica também apresenta um maior valor de acurácia (91,5% em contraposição à 84,2% e 77,8%).

3 CLASSIFICAÇÃO DE QUESTÕES DE ÁREAS DA COMPUTAÇÃO

Esse capítulo apresenta detalhes da execução do trabalho, com a abordagem realizada, qual metodologia de pesquisa foi aplicada, como os dados foram adquiridos e, posteriormente, analisados.

3.1 ABORDAGEM

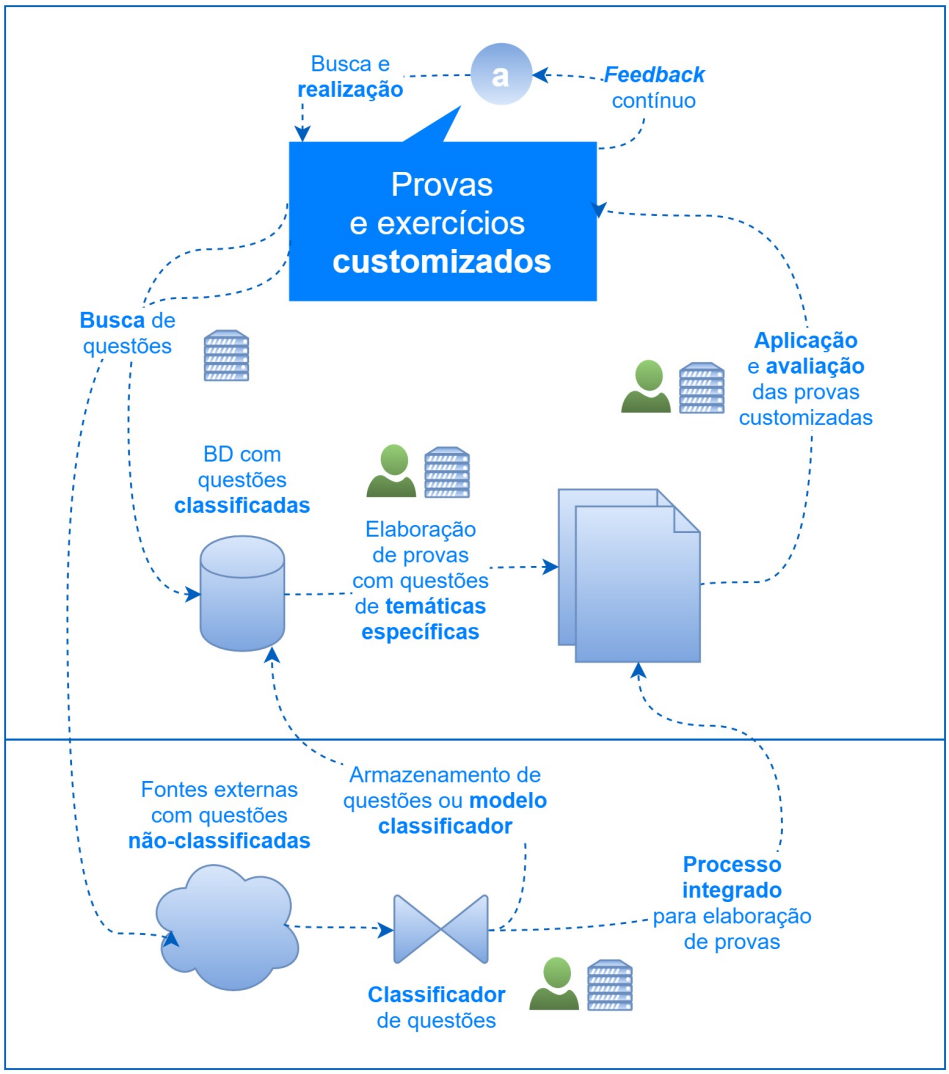
Conforme apresentado na Seção 1.3, o objetivo desse trabalho consiste em atuar na etapa (a) da Figura 1, a qual consiste em contribuir em uma parte da etapa de elaboração de provas e exercícios customizados inerentes ao processo geral do ensino adaptativo. Essa etapa customizada, representada pelo bloco superior da Figura 5, pode ser composta por três subetapas principais: obtenção e/ou busca de questões, composição da prova em si e coleta de informações para avaliação posterior à execução da prova customizada. Normalmente essas subetapas são internas ao sistema adaptativo, com execução automatizada baseadas em regras e metadados disponíveis - como a classificação de cada questão. Esses metadados podem estar disponíveis em algum armazenamento de disco e dados, normalmente rotulados por algum sistema externo ou por algum especialista do assunto. As questões armazenadas em um banco de dados do sistema adaptativo, por exemplo, podem ter sido obtidas por livros ou outras mídias, sendo classificadas manualmente por especialistas ou metadados que eventualmente já estavam disponíveis.

Contudo, essa base de dados pode ser limitada, e o sistema pode eventualmente exigir um incremento desses dados categorizados, afim de prolongar ou aprimorar a experiência customizada de cada aluno. Sendo assim, a abordagem a ser desenvolvida nesse trabalho consiste em auxiliar nesse processo, trabalhando na obtenção de questões e na confirmação da classificação das mesmas, tomando como base o subprocesso apresentado na parte inferior da Figura 5. A classificação de questões pode armazená-las em uma base de dados pré-existente e/ou exportar o modelo classificador de questões, de forma a aprimorá-lo a cada iteração realizada. Essa classificação pode atuar em áreas distintas, todavia esse trabalho tem como enfoque atuar apenas na área de computação, verificando a viabilidade da utilização de algoritmos de PLN para classificação de questões de provas nessa área específica.

3.2 METODOLOGIA

Através da análise bibliográfica apresentada na Seção 2.3, encontrou-se evidências de um possível gargalo de trabalhos envolvendo a classificação de questões em categorias e áreas do domínio da computação, encontrando poucos trabalhos como dos autores Rao, S. e Kannan (2016) e Fu, Qu e Wang (2009), envolvendo QAs, e dos autores Jayakodi, Bandara e Perera (2015), envolvendo taxonomia cognitiva de *Bloom*. Apesar da relevância das taxonomias

Figura 5 – Abordagem da classificação de questões no processo de elaboração de provas e questões customizadas referentes ao processo geral de ensino adaptativo



Fonte: Autoria Própria

educacionais - como a de *Bloom* - na avaliação de desempenhos estudantis e na definição dos objetivos curriculares de cursos, a abordagem desses trabalhos envolvendo taxonomias educacionais é utilizada para discernir o nível de compreensão sob uma perspectiva abstrata (JAYAKODI; BANDARA; PERERA, 2015).

Sendo assim, esse trabalho realizou uma classificação mais próxima dos trabalhos realizados por Xiao, Chow e Chen (2017) e Fu, Qu e Wang (2009), trabalhando com questões de categorias específicas da computação:

1. (Tribunal Regional do Trabalho / 16a Região (TRT 16a) 2014 - Analista Judiciário - Tecnologia da Informação)
   O *SSH* (*Secure Shell*), um dos protocolos do conjunto *TCP/IP*, é vastamente utilizado para as transações na internet que exigem o uso de esquemas de segurança. A técnica de criptografia utilizada no SSH faz uso do esquema de chaves:

a) Simétricas.
b) Distribuídas.
c) Ortogonais.
d) Públicas.
e) Compartilhadas.

2. (Defensoria Pública do Estado do Rio Grande do Sul/RS (DPE/RS) 2017 - Analista - Tecnologia da Informação)
Um sistema operacional multiprogramável possui programas reentrantes que se caracterizam por:
a) permitirem a execução simultânea de dois programas sem que uma execução interfira na outra.
b) realizarem a carga na memória de apenas uma cópia do programa que pode ser compartilhado por vários usuários.
c) poderem ser executados na memória virtual, em vez de utilizarem a memória principal.
d) executar programas com rapidez, pois não podem ser interrompidos pela execução de outros programas simultaneamente em execução.
e) serem utilizados apenas para rotinas de acesso direto à memória.

As duas questões apresentadas podem ser classificadas em mais de uma categoria das áreas computacionais em função das palavras chaves utilizadas. Tais palavras de uma única questão podem representar temáticas específicas - **TCP/IP** pode representar a temática de Redes de Computadores, enquanto **segurança** pode representar Segurança e Auditoria de Sistemas. Além dessa problemática, as questões podem apresentar múltiplas temáticas, muitas vezes sem uma classificação balanceada entre termos e temáticas.

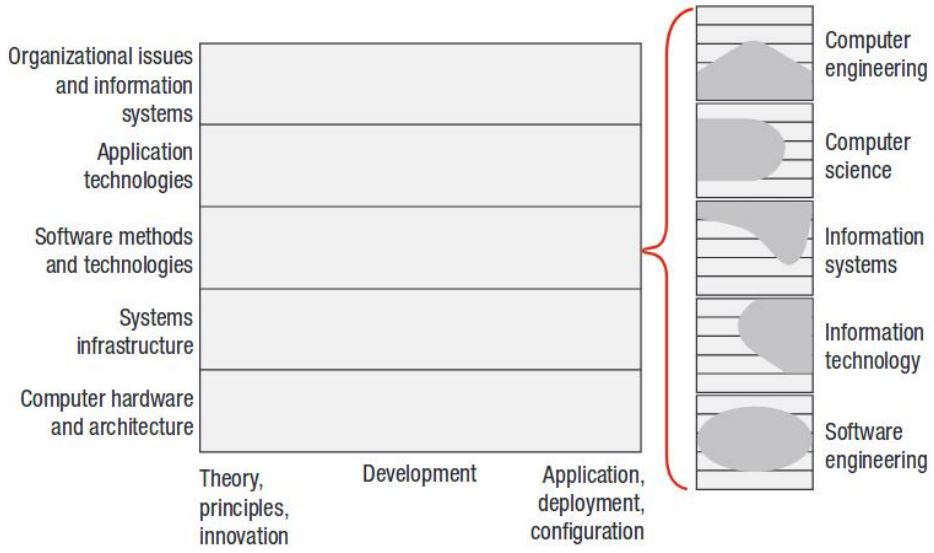
Por meio da análise e validação gradativa de questões entre taxonomias de áreas e subáreas computacionais (conforme à distribuição de cursos computacionais apresentados na Figura 6), um experimento foi conduzido nesse trabalho com a hipótese de avaliar a eficácia de utilização de métodos de classificação em questões de provas da computação em cinco temáticas escolhidas - Banco de Dados, Redes de Computadores, Arquitetura de Computadores, Sistemas de Informação e Sistemas Operacionais. Essa hipótese foi guiada pela seguinte questão:

(Q1) A abordagem proposta é capaz de classificar questões de áreas computacionais na língua portuguesa?

A classificação foi planejada e realizada com base nas palavras em comum entre questões de uma mesma temática e seus respectivos pesos taxonômicos. A escolha das questões foram influenciadas pelas categorias, separadas em áreas do conhecimento e suas unidades, as quais são divididas em função dos cursos principais dos currículos computacionais de Walrad (2016). O curso de engenharia da computação, por exemplo, possui um currículo recomendado em 2016 pela ACM com 12 áreas de conhecimento e 115 unidades internas à essas áreas (ACM; SOCIETY, 2016).

Conforme apresentado na Figura 7, a metodologia experimental se dividiu em cinco

Figura 6 – Distribuição espacial de taxonomias computacionais para cursos segundo Relatório de Currículos Computacionais de 2005

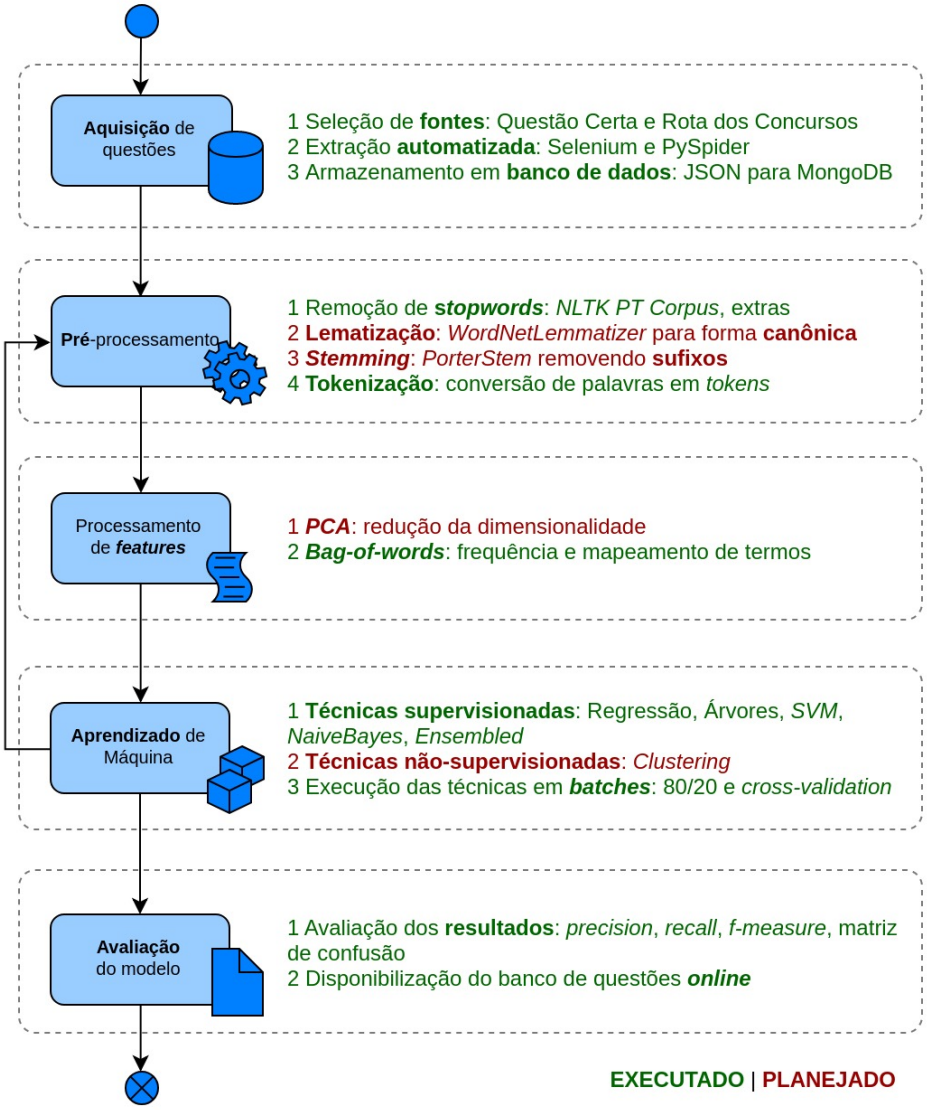


Fonte: (WALRAD, 2016)

etapas, baseando-se nos trabalhos relacionados apresentados anteriormente e no processo base da Seção 2.2, diferenciando-se por partes planejadas (em vermelho) e executadas (em verde):

1. **Aquisição de questões**: seleção de um conjunto de bases de questões (com a parte textual, alternativas e respostas) a serem extraídas e armazenadas a partir de dois *websites* de questões de concursos. Extrações envolvendo temáticas da computação foram realizadas utilizando ferramentas de *scraping* como *Selenium* e *PySpider* - a utilização de duas ferramentas se deve à necessidade de armazenamento de sessões e *login*. Em cada *website* foi realizada uma extração de todas as temáticas, cujos resultados obtidos foram convertidos e armazenados em um banco de dados *NoSQL*.
2. **Pré-processamento**: tratamento do *corpus* textual para obtenção de melhores resultados nos treinamentos. Inicialmente, palavras não-relevantes às temáticas (*stopwords*) são removidas utilizando uma lista de palavras do *corpus* textual português da biblioteca *NLTK* versão 3.3, além de uma lista própria para remoção de alternativas de questões - como, por exemplo, a), A., b-). De forma complementar, pode-se realizar uma simplificação de palavras semelhantes por meio da lematização e do *stemming*. Por fim, as palavras são convertidas em *tokens* e adicionadas em vetores que serão utilizados nas próximas etapas.
3. **Processamento de features**: conversão de vetores de palavras em vetores de frequência, mapeando as palavras de acordo com a técnica de *BOW*. De forma complementar, pode ser realizada a técnica de *PCA*, a qual reduz a dimensionalidade da base de *features* por meio da correlação entre palavras semelhantes ou sinônimas.
4. **Aprendizado de máquina**: realização de treinamentos utilizando algoritmos de apren-

Figura 7 – Etapas propostas e executadas na metodologia experimental



Fonte: Autoria Própria

dizado de máquina disponíveis na biblioteca *Scikit-Learn*. Cada uma das técnicas foram realizadas em *batches*, obtendo resultados utilizando cargas balanceadas e não balanceadas, *cross-validation* de 10 *k-folds* e método de *hold-out* com distribuições 80/20 (divisão de amostras em 80% de treinamento e 20% de testes).

5. **Avaliação do modelo**: após obtenção de relatórios detalhados dos treinamentos, os resultados foram avaliados conforme métricas de eficácia de *precision* (precisão), *recall* (sensibilidade), *f-measure* e matriz de confusão, os quais foram analisados individualmente por temática e pela média geral obtida.

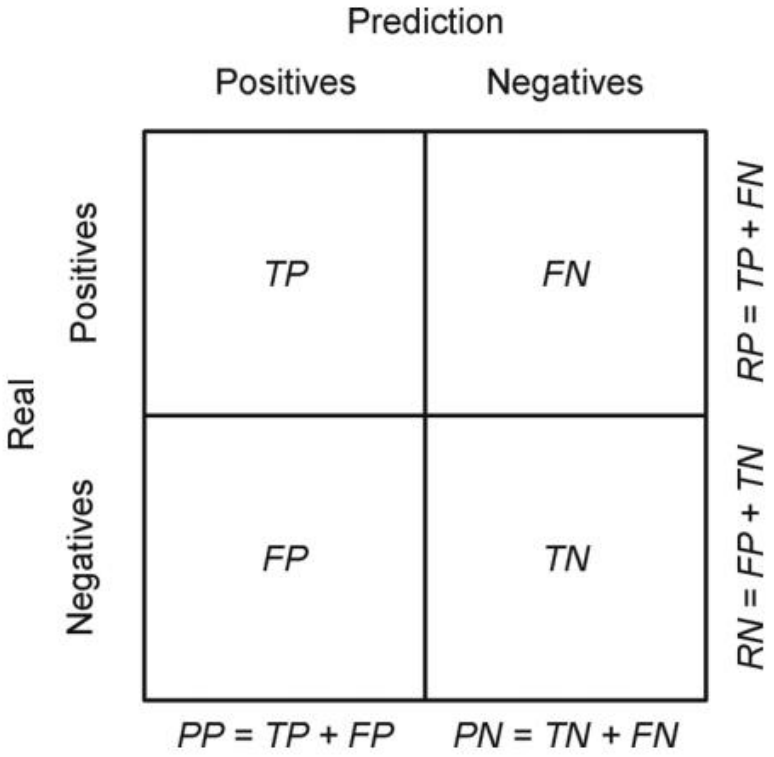
As métricas de performance de *precision*, *recall* e *f-measure* são determinadas para uma melhor eficácia e confiança nos resultados previstos. Um simples valor de acurácia não é capaz de indicar uma alta probabilidade de conseguir predições corretas, já que o resultado pode

ser influenciado por uma quantidade específica de amostras (JIAO; DU, 2016). Esse conjunto de métricas é aplicado, portanto, a modelos de *labels* únicas ou múltiplas, baseando-se na comparação entre a *label* real e a prevista em cada amostra do processo de teste.

No caso de uma classificação entre duas temáticas, por exemplo, cada questão binária pode ser classificada como 1 ou 0, sendo tais números os respectivos *labels* de cada temática. As possíveis combinações entre as duas temáticas e *label* reais e previstas originam em uma tabela de contigência chamada matriz de confusão (vide Figura 8), contendo os seguintes termos:

1. *TP*: quando o *label* previsto possui o mesmo valor verdadeiro do *label* real (ex: questão com temática real 1 é acusada como correta na classificação para a temática 1);
2. *FP*: quando o *label* previsto é verdadeiro, mas o valor do *label* real é falso (ex: questão com temática real 0 é acusada como correta na classificação para a temática 1);
3. *FN*: quando o *label* previsto é falso, mas o valor do *label* real é verdadeiro (ex: questão com temática real 1 é acusada como falsa na classificação para a temática 1);
4. *TN*: quando o *label* previsto é falso, mas o valor do *label* real é verdadeiro (ex: questão com temática real 1 é acusada como falsa na classificação para a temática 0).

Figura 8 – Matriz de confusão com *TP*, *FN*, *FP*, *TN* e respectivas combinações



Fonte: (JIAO; DU, 2016)

Precision, uma das métricas utilizadas para avaliação dos modelos de treinamento,

representa a proporção entre *TP* considerados corretamente como *PP*:

$$Prec = \frac{TP}{PP} = \frac{TP}{TP + FP} \tag{1}$$

Recall, por sua vez definida como sensibilidade, representa a proporção entre *TP* considerados corretamente como *RP*:

$$Recall = \frac{TP}{RP} = \frac{TP}{TP + FN} \tag{2}$$

Por fim, a medida de *f-measure* consiste na média harmônica de *precision* e *recall* (POWERS, 2011):

$$F_{measure} = 2 * \frac{Prec * Recall}{Prec + Recall} = 2 * \frac{TP}{FN + FP + 2} \tag{3}$$

Resumidamente, enquanto a *Precision* consiste em uma medida de relevância do resultado, o *Recall* se refere a quantos resultados relevantes foram retornados no treinamento. O objetivo ideal consiste em obter, portanto, uma alta precisão e uma sensibilidade, retornando vários resultados com grande quantidade de classificações corretas.

Por meio do processo proposto na metodologia experimental, os modelos foram avaliados para responder a questão do experimento, verificando o desempenho por meio das métricas de avaliação na classificação final das questões na língua portuguesa (Q1). As próximas seções apresentam os detalhes de cada uma das etapas apresentadas, desde a mineiração das questões até a análise dos modelos e resultados.

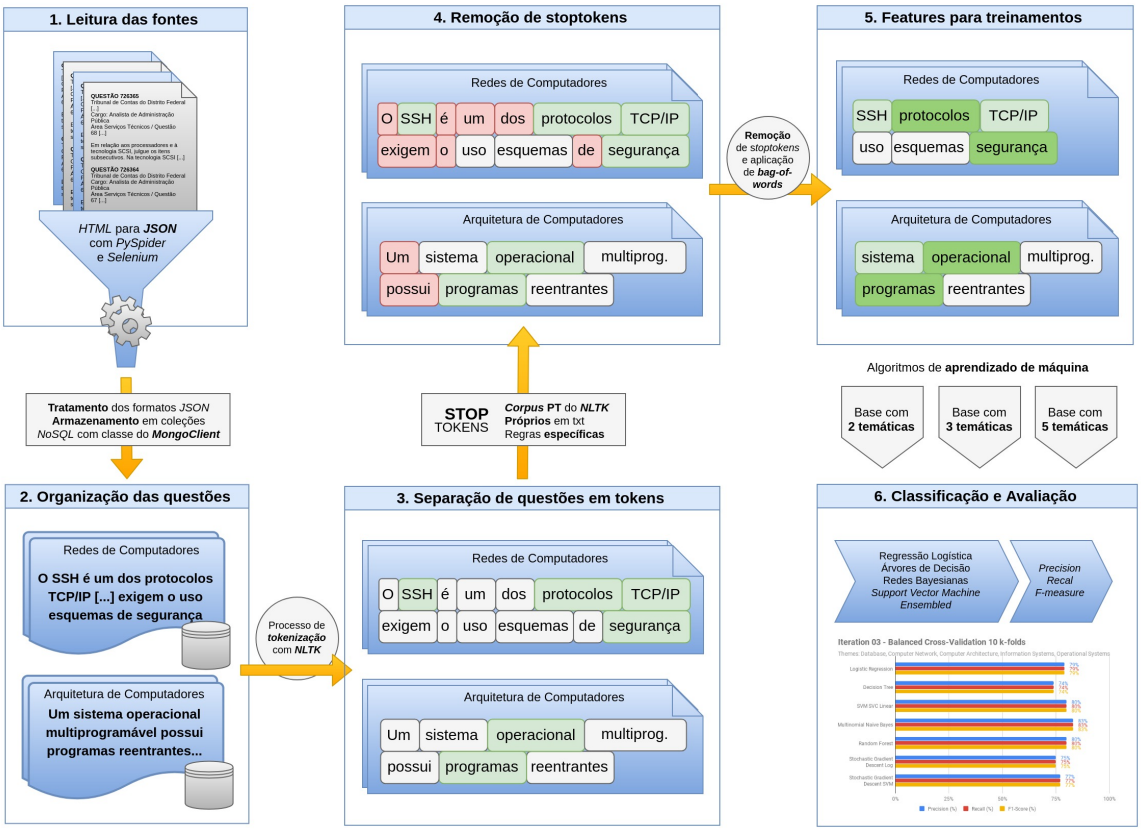
3.3 AQUISIÇÃO DO *CORPUS*

A extração de textos e a realização dos treinamentos com palavras pré-processadas tratam de processos relacionados à mineração de dados (*Data Mining*). Segundo Han, Kamber e Pei (2011), o processo de *Data Mining* atua na descoberta de padrões de conhecimento a partir de grandes quantidades de dados e informações. Para esse trabalho, por sua vez, focou-se na Mineração da *Web* (*Web Mining*), a qual consiste na metodologia de *RI* que utiliza ferramentas de mineração de dados para descobrir e extrair informações de documentos e serviços presentes na *Web* (KOSALA, 2000). Abrangendo o processo padrão de Descoberta do Conhecimento (*Knowledge Discovery*), a mineração da *Web* tem o potencial de obtenção de informações ou conhecimentos - estruturadas ou não-estruturadas - de dados da *Web*, sejam eles on-line ou off-line.

Segundo Kosala (2000), esse processo pode ser subdivido em algumas subtarefas, as quais foram abordadas e representadas nesse trabalho na Figura 9:

1. Procura de recursos (etapa 1): Extração e recuperação de documentos estruturados ou não-estruturados disponíveis na *Web*;

Figura 9 – Abordagem realizada para classificação das questões de áreas da computação



Fonte: Autoria Própria

2. Seleção de informações e pré-processamento (etapas 2, 3 e 4): qualquer processo de transformação das informações originais, podendo ser desde remoção de *stoptokens* até obtenção de formatos finais desejados (ex: extração de frases e *tokens* de palavras);
3. Generalização (etapa 5): descoberta de padrões em *websites* ou no contexto obtido a partir das informações extraídas e processadas nas etapas anteriores - utilizando técnicas de aprendizado de máquina e/ou de mineração de dados;
4. Análise de Resultados (etapa 6): validação e interpretação dos padrões e resultados obtidos na etapa de Generalização.

Cada uma dessas etapas, conforme apresentado na Figura 9, foram executadas nesse trabalho. A procura de recursos foi feita por meio da técnica de Mineração de Conteúdos da *Web* (*Web Content Mining*) utilizando as ferramentas de *webcrawling Selenium* e *PySpider*. As questões não-estruturadas, obtidas em formatos *JSON* de documentos *HTML* de dois sites de concursos - Rota dos Concursos e Questão Certa -, foram organizadas e armazenadas em um banco de dados em *MongoDB*, sendo separadas por temáticas (categorias obtidas por meio do *crawling* realizado). Cada temática obtida e a sua respectiva quantidade de questões pode ser visualizada na Tabela 1, onde cada temática foi armazenada em uma coleção e as questões, por sua vez, como documentos.

Na etapa de seleção de informações e pré-processamento, por sua vez, foi realizada

Tabela 1 – Temáticas inseridas em coleções do banco de dados final em *MongoDB*

Temáticas (Coleções)	**Questões (Docs)**	**Média tamanho/doc.**	**Total tamanho/doc.**
algoritmos_e_estrutura_de_dados	425	713.4 B	296.1 KB
arquitetura_de_computadores	1,062	767.1 B	795.6 KB
arquitetura_de_software	355	870.4 B	301.7 KB
arquiteturas	272	768.7 B	204.2 KB
aspectos_gerais	370	742.8 B	268.4 KB
banco_de_dados	4,449	761.9 B	3.2 MB
engenharia_de_software	1,610	803.3 B	1.2 MB
gestao_de_ti	1,352	821.7 B	1.1 MB
governanca_de_ti	1,345	798.7 B	1.0 MB
programacao	1,555	745.6 B	1.1 MB
redes_de_computadores	5,483	726.8 B	3.8 MB
seguranca_da_informacao	3,081	790.0 B	2.3 MB
servidor	290	681.1 B	192.9 KB
sistema_de_arquivos	62	741.5 B	44.9 KB
sistemas_de_apoio_a_decisao	704	807.3 B	555.0 KB
sistemas_de_informacao	1,040	880.4 B	894.2 KB
sistemas_operacionais	2,483	737.6 B	1.7 MB
software	2,217	739.6 B	1.6 MB
virtualizacao	160	804.6 B	125.7 KB
web_design_computacao_grafica	160	701.4 B	109.6 KB

Fonte: Autoria própria

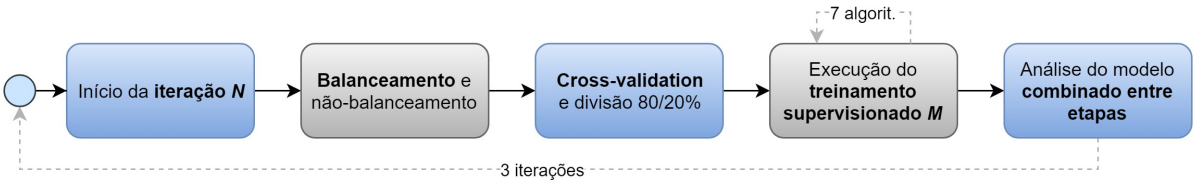
a separação de cada uma das frases das questões obtidas em um vetor de *tokens*, remoção de alguns desses *tokens* a partir de conjunto de *stoptokens* - *Corpus* de língua portuguesa da biblioteca *NLTK* e *tokens* customizados -, e aplicação da técnica de *BOW* - normalização de frequência de palavras em vetores de palavras. A normalização de frequência de palavras (WF_i) utilizando a técnica de *BOW* consiste na razão entre a quantidade de cada palavra *i* (wq_i) e a soma entre todas as quantidades de palavras, representação a qual pode ser visualizada através da Equação (4). A partir da obtenção das *features* processadas foram realizados treinamentos de modelos e os resultados obtidos foram analisados.

$$WF_i = f(wq_i) = \frac{wq_i}{\sum wq_i} \tag{4}$$

3.4 ANÁLISE DOS DADOS

A partir das *features* pré-processadas na etapa anterior de pré-processamento, a etapa de Generalização, representada na Figura 10, teve o papel de dividir e executar três iterações de treinamentos supervisionados incrementais: começando com 2 temáticas, aumentando para 3 e terminando com 5 temáticas.

Figura 10 – Fluxograma correspondente a etapa de Generalização



Fonte: Autoria Própria

3.4.1 Divisão de Iterações e Balanceamento

Para realizar um aprendizado evolutivo da base de temáticas obtida, cada iteração dos treinamentos foi aprimorada por meio da adição de novas temáticas, estimulando e incluindo uma maior complexidade aos modelos obtidos. As temáticas escolhidas apresentam uma maior quantidade de questões para realização de treinamentos mais diversificados e completos, dividindo-se em três iterações:

(IT1) Banco de Dados (4449 questões) e Redes de Computadores (5483 questões);

(IT2) Arquitetura de Computadores (1062 questões), Sistemas de Informação (1040 questões) e Sistemas Operacionais (2483 questões);

(IT3) Banco de Dados (4449 questões), Redes de Computadores (5483 questões), Arquitetura de Computadores (1062 questões), Sistemas de Informação (1040 questões) e Sistemas Operacionais (2483 questões).

Enquanto a primeira iteração possuía uma quantidade próxima de questões entre as duas temáticas escolhidas, a segunda e a terceira iteração possuíam uma proporção de até 1 para 5 entre a menor e a maior classe. Treinamentos supervisionados de classes podem apresentar essa desproporção, também conhecida como desbalanceamento de amostras. Conforme Chawla et al. (2004), esse desbalanceamento pode prejudicar a performance do modelo, no qual classificadores podem atuar tendenciosamente em favor a classes com uma maior quantidade de amostras em detrimento daquelas que possuem menor quantidade. Sendo assim, os treinamentos foram divididos em modelos com amostras desbalanceadas e balanceadas pela menor quantidade de amostras, verificando a influência desse fator nos parâmetros de avaliações finais.

3.4.2 Validação dos Modelos

Antes de avaliar a performance de cada algoritmo de treinamento, é fundamental verificar a estabilidade do modelo a ser obtido por meio de alimentação correta de *features* em cada treinamento. Como cada resultado dos modelos depende das entradas, deve-se averiguar e garantir que cada modelo consiga padrões corretos a partir das amostras, com baixos viéses e variâncias (GUPTA, 2017).

Para esse trabalho, foram escolhidos dois tipos de métodos de *cross-validation* (validação cruzada) não-exaustivos em função da simplicidade e da capacidade computacional

reduzida - métodos exaustivos não seriam possíveis em função do limite computacional e da quantidade de *features* por amostras.

O primeiro método consiste no *hold-out*, procedimento o qual divide as *features* em um grupo de treinamento das classes e outro de teste, garantindo uma alta probabilidade de baixa discrepância na taxa de erros (BLUM; KALAI; LANGFORD, 1999). Esse método trata de extrair uma parte única das *features* de treinamento e utilizá-la como grupo de testes no modelo treinado com o grupo de treinamento. Porém, segundo Dietterich (1998), o *hold-out* não considera a variância em relação ao grupo de treinamento (incerteza de quais *features* serão escolhidas para treinamento e teste), podendo ser ineficiente para treinamentos que possuem pequenas amostras de *features* e causar o problema de *underfitting*.

Sendo assim, para apresentar e verificar uma solução alternativa ao método de *hold-out*, foi utilizado o procedimento de *cross-validation* chamado *k-folds*, o qual reduz viéses ao utilizar a maior parte das *features* na etapa de *fitting* (ajuste). O procedimento em questão repete a técnica de *hold-out* *k* vezes (*k* subgroups com tamanhos iguais de *features*), sendo que em cada *k* iteração um dos *k* subgrupos é usado como grupo de teste e os outros *k-1* subgrupos são unidos para formar um grupo de treinamento (BLUM; KALAI; LANGFORD, 1999). O resultado final da performance do modelo é obtido através da média de todas as *k* iterações, sendo recomendado por experimentos empíricos a utilização de 5 a 10 *k-folds* (GUPTA, 2017).

3.4.3 Algoritmos de Treinamento Supervisionado

A partir da definição obtida nas etapas anteriores, realizaram-se os treinamentos supervisionados utilizando sete algoritmos de classificação: Regressão Logística, Árvores de Decisão, Linear *SVM*, Redes *Bayesianas* Multinomiais, *Random Forest* e Gradiente Descendente Estocástico (Logarítmico e *SVM*). Esses algoritmos estão disponíveis na biblioteca *Scikit-Learn* e foram executados através de *scripts* na linguagem *Python* versão 2.7.

3.4.3.1 Regressão Logística

Modelo linear também conhecido como classificação da máxima entropia, a Regressão Logística apresenta as probabilidades dos possíveis resultados sendo modeladas utilizando uma função logística:

$$f_x = \frac{L}{1 + e^{-k(x-x_0)}} \tag{5}$$

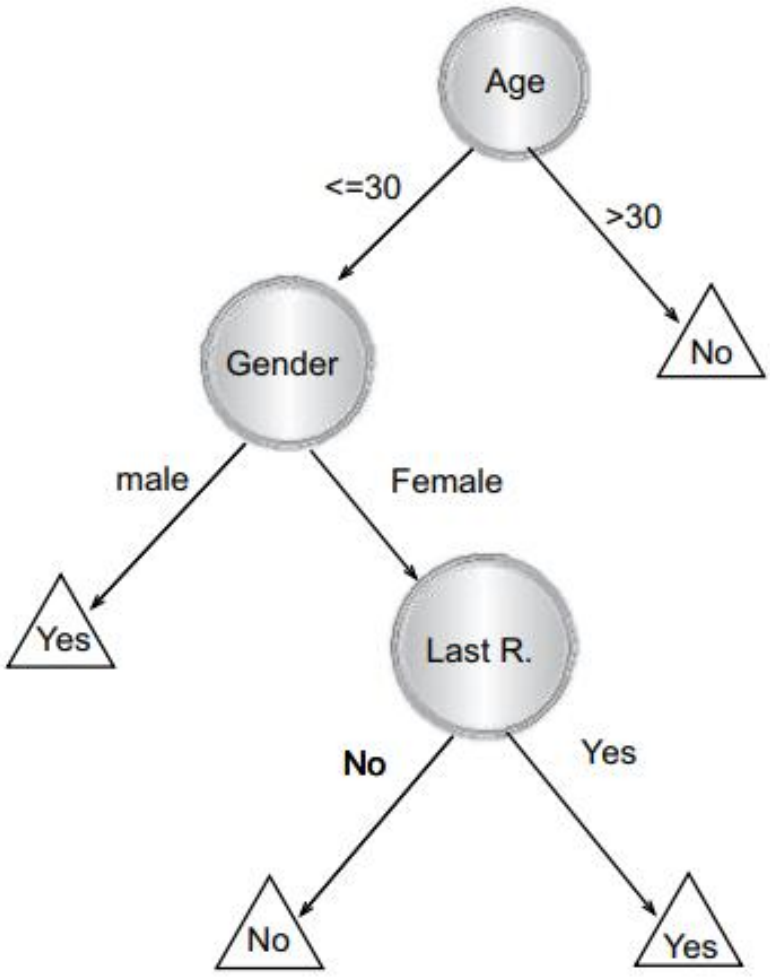
A sua implementação pode seguir um ajuste binário, *one-vs-rest* ou uma regressão logística multinomial com regularizações L1 ou L2, sendo esse último padrão e minimiza a seguinte função de custo:

$$\min_{w,c} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \log(\exp(-y_i(X_i^T w + c)) + 1) \tag{6}$$

3.4.3.2 Árvores de Decisão

Baseado em nós e folhas que formam uma árvore enraizada, algoritmos de árvore de decisão possuem implementações muito próximas à indução de regras. Segundo a indução de árvores, cada caminho da raíz até um dos nós folhas forma uma regra, e o resultado da previsão da classe estabelece a *feature* correspondente à classe em si (ROKACH; MAIMON, 2010). Conforme pode ser visualizado em um contexto de resposta de e-mail *marketing* da Figura 11, cada caminho pode determinar uma regra induzida que o modelo obtido pode seguir. Nessa árore, cada *feature* é representada por um círculo e cada classe por um triângulo nas folhas. Segundo Quinlan e R. (1987) o resultado das regras obtidas pode, por sua vez, ser simplificado para melhorar a compreensão e, possivelmente, a acurácia de cada modelo de árvores de decisão.

Figura 11 – Exemplo de Árvore de Decisão com possíveis induções de respostas à e-mail *marketing*

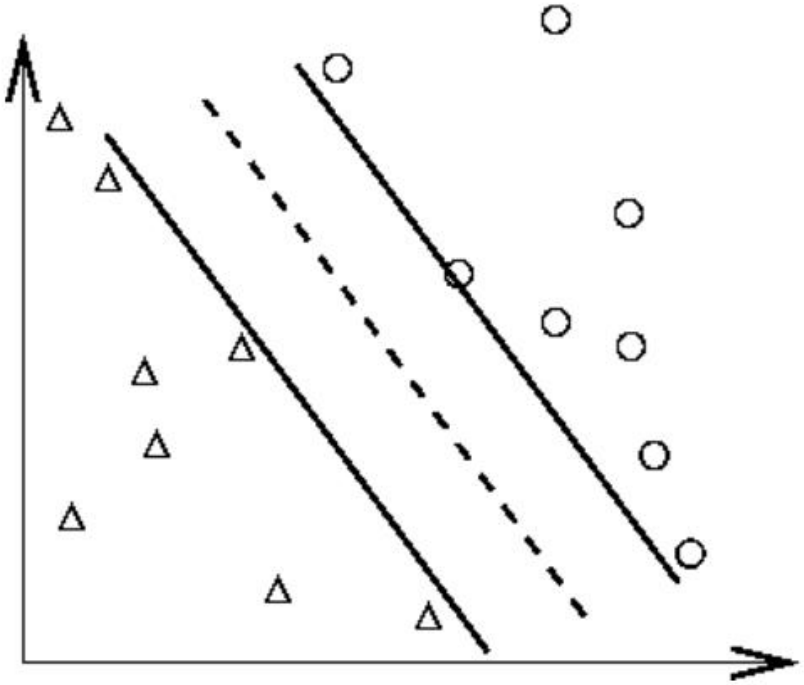


Fonte: (ROKACH; MAIMON, 2010)

3.4.3.3 *SVM*

Abrangindo um conjunto de métodos de aprendizado supervisionado, o algoritmo *SVM* de classificação tem como objetivo dividir dados que estão agrupados em vetores (valor(es) em conjunto da classe correspondente) internos a espaços multidimensionais por meio da descoberta empírica de um hiperplano com máxima distância de margem. O hiperplano em questão visa separar os dados de cada classe e maximiza a distância entre os dados e às margens (SHMILOVICI, 2010), conforme pode ser visto através da Figura 12.

Figura 12 – Exemplo de classificador *SVM* com divisão estabelecida por um hiperplano e margens máximas



Fonte: (SHMILOVICI, 2010)

A definição de qual hiperplano divide o espaço multidimensional depende da função de *Kernel*, a qual basicamente realiza um produto interno entre os vetores para obter o ângulo e a direção do hiperplano (SHMILOVICI, 2010), cuja representação é dada pela Equação (7):

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j) \tag{7}$$

Além dessa representação padrão, os algoritmos de *SVM* são baseados em quatro tipos de Equações de *Kernels* básicos segundo Hsu, Chang e Lin (2003): Linear (8), Polinomial (9), *RBF* (10) e Sigmóide (11).

$$K(x_i, x_j) = x_i^T x_j \tag{8}$$

$$K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0 \tag{9}$$

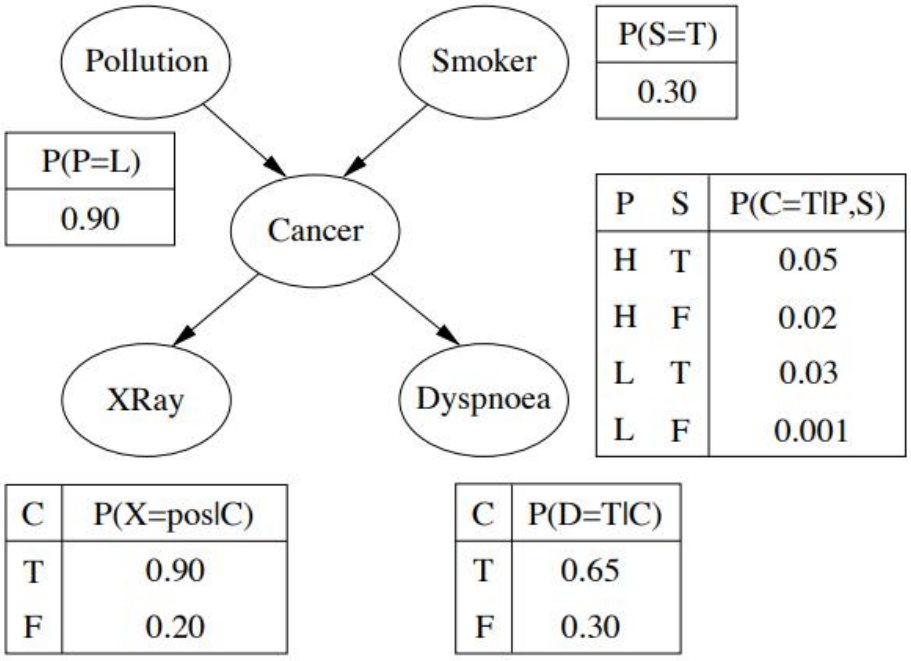
$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0 \tag{10}$$

$$K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r) \tag{11}$$

3.4.3.4 Redes *Bayesianas* Multinomiais

Redes *Bayesianas* consistem em modelos gráficos utilizados para tomadas de decisão sob incerteza, sendo compostos por nós - representam variáveis discretas ou contínuas - e conectados por arcos, formando grafos acíclicos (KORB; NICHOLSON, 2010). Por meio de conexões causais, modelos de Redes *Bayesianas* podem estabelecer e modelar distribuições probabilísticas entre as variáveis conectadas. Baseando-se no Teorema de *Bayes*, as redes formam um modelo gráfico probabilístico, estabelecendo uma estrutura híbrida entre um grafo e tabelas de probabilidade para cada combinação viável entre os nós (SEBASTIANI; ABAD; RAMONI, 2010). Uma representação desse modelo híbrido pode ser visto na Figura 13, a qual apresenta uma Rede *Bayesiana* de avaliação de um câncer de pulmão.

Figura 13 – Exemplo de Rede *Bayesiana* estabelecida para caso de câncer de pulmão



Fonte: (KORB; NICHOLSON, 2010)

O teorema atua na condição independente entre vetores de *features* entre x_1 até x_n além da variável da classe *y*, declarando a relação de probabilidade apresentada na Equação (12):

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)P(x_1, \ldots x_n \mid y)}{P(x_1, \ldots, x_n)} \tag{12}$$

Por meio do desenvolvimento da Equação (12) usando a suposição de independência condicional ingênua e proporcionalidade direta apresentada por Zhang (2004), pode-se obter, por meio de classificadores *Bayesianos*, a frequência relativa da classe *y* em questão, representada pela Equação (13):

$$\hat{y} = \arg\max_y P(y) \prod_{i=1}^{n} P(x_i \mid y) \tag{13}$$

Apesar das suposições simplificadas, classificadores de redes *Bayesianas* possuem resultados relevantes para classificação de documentos, especialmente para o tipo Multinomial, o qual implementa algoritmos de *naive Bayes* para dados distribuídos multinomialmente (ZHANG, 2004). O algoritmo de Redes *Bayesianas* Multinomais estabelece, portanto, a probabilidade da presença de uma *feature i* em uma amostra pertencente à classe *y*. Essa representação é demonstrada pela Equação (14), onde vetores ϕ_y são estimados utilizando uma versão de máxima vizinhança. Essa variação de *Naive Bayes* clássica é utilizada com frequência em classificação de textos, com dados representados por vetores de frequência de palavras.

$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n} \tag{14}$$

3.4.3.5 *Random Forest*

O algoritmo de *Random Forest* consiste no classificador *Ensemble* baseado em métodos de média, o qual desenvolve-se vários modelos independentes e realiza-se a média de suas previsões. Esse conceito de integração de vários modelos para construção de um modelo de previsão único define os métodos classificadores *Ensemble*, obtendo um modelo mais preciso e confiável (ROKACH, 2010).

Através da criação de um conjunto de classificadores baseados em árvores junto da introdução de aleatoriedade na construção do classificador (cada derivação de nó escolhe-se a melhor divisão entre as *features* randômicas), o algoritmo de *Random Forest* consegue um alto viés em função da aleatoriedade, mas consegue uma redução da variância em função da aplicação da média entre os modelos, compensando o alto viés obtido (SCIKITLEARN, 2011a).

3.4.3.6 Gradiente Descendente Estocástico

O Gradiente Descendente trata de um algoritmo de otimização utilizado para encontrar parâmetros ou coeficientes de uma função *f* que minimiza uma função de perda/custo. Por ser

um processo não-analítico, ele depende de um processo empírico, o qual repetem-se iterações até que a função de custo seja o mais próximo de zero - mínimo de uma função de custo (BROWNLEE, 2013).

Como o algoritmo de Gradiente Descendente normalmente é lento para grandes quantidades de dados, a variação do mesmo para o tipo Estocástico apresenta-se como alternativa, no qual realiza-se a atualização dos coeficientes em cada amostra ao invés de passar por todas elas em cada iteração. Isso permite alcançar os mínimos valores desejados de forma mais rápida, sendo útil para problemas no contexto de classificação de textos e PLN - podendo alcançar até 10^5 amostras e mais de 10^5 *features* (SCIKITLEARN, 2011b). A atualização do modelo também pode depender do tipo de função de perda utilizada, podendo ser:

1. *Hinge Loss* (Perda de articulação): utilizada para classificação em *SVM*, classificando de acordo com a margem máxima;
2. *Modified Huber Loss*: função de perda de articulação suavizada;
3. *Logistic Loss*: utilizada para classificação em Regressão Logística.

4 RESULTADOS OBTIDOS

Seguindo a metodologia proposta na Seção 3.2 foi possível realizar a extração de questões de dois sites, resultando em um total de 20 temáticas e 28745 questões exportadas e armazenadas em um banco de dados *NoSQL* - conforme apresentado na Tabela 1. A partir de quatro etapas apresentadas na Figura 10 e nos tópicos 3.4.1, 3.4.2 e 3.4.3, as questões foram adequadas para treinamentos em 7 tipos de algoritmos supervisionados, resultando em 10 modelos treinados em um total filtrado de 14517 amostras de questões entre as três iterações realizadas (vide Tabela 2).

Tabela 2 – Quantidade de amostras de questões utilizadas em cada iteração

Iteração	Banco de Dados	Redes de Comp.	Arqu. de Comp.	Sist. de Inform.	Sist. Operac.	Total
IT1	4449	5483	-	-	-	**9932**
IT2	-	-	1062	1040	2483	**4585**
IT3	4449	5483	1062	1040	2483	**14517**

Fonte: Autoria própria

As iterações em questão foram realizadas através da combinação entre temáticas, balanceamentos, divisões para treinamentos e cada algoritmo supervisionado utilizado. Na última iteração (IT3), todavia, dois modelos não-balanceados não foram possíveis de serem executados. Conforme pode ser visualizado na sequência das Figuras 14 e 15, a execução desses modelos ocasionou um estouro na memória de 8GB do *hardware* utilizado (maiores detalhes no Apêndice C), o que impossibilitou a continuação do processo.

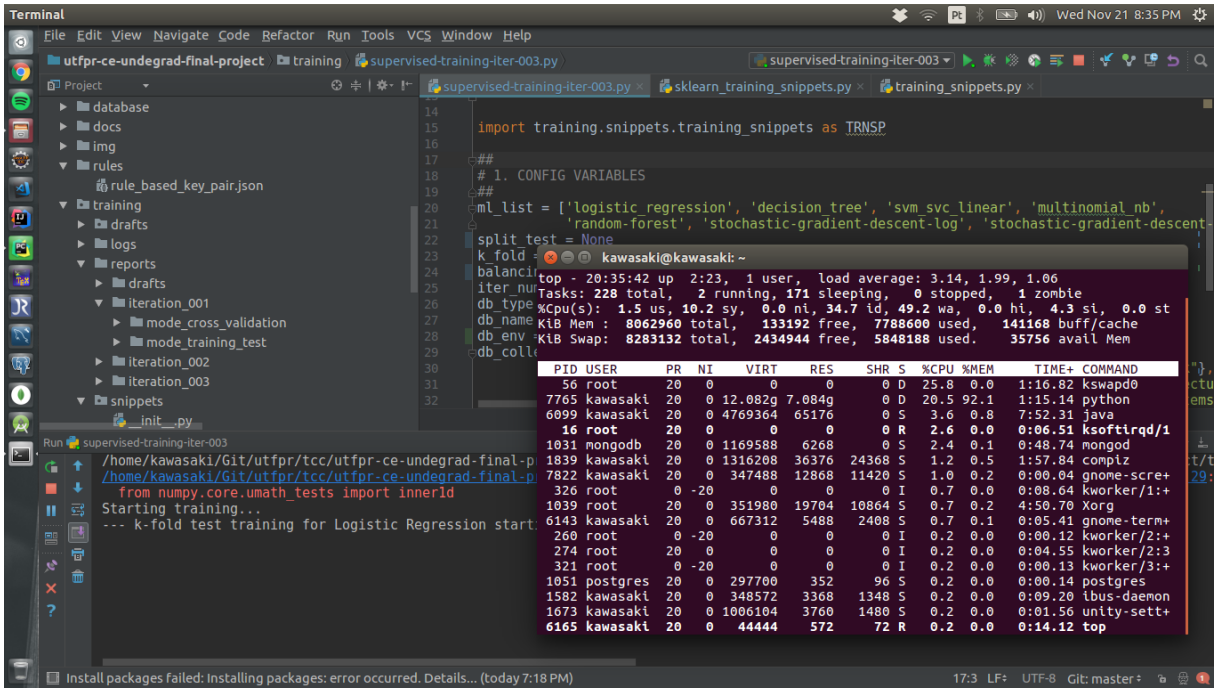
Os resultados médios de *precision*, *recall* e *F1-score* podem ser visualizados através das Tabelas 3 e 4, enquanto os resultados referentes à Iteração 3 e de cada algoritmo supervisionado podem ser visualizados através da Tabela 5 e dos gráficos das Figuras 16 e 17 - resultados referentes às Iterações 1 e 2 podem ser visualizados nos Apêndices A e B.

Tabela 3 – Médias obtidas nos treinamentos para amostras desbalanceadas nas três iterações realizadas

	Hold-out de 80-20%			10 k-folds		
Iter.	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
IT01	95%	95%	95%	95%	95%	95%
IT02	78%	79%	77%	80%	81%	80%
IT03	-	-	-	-	-	-

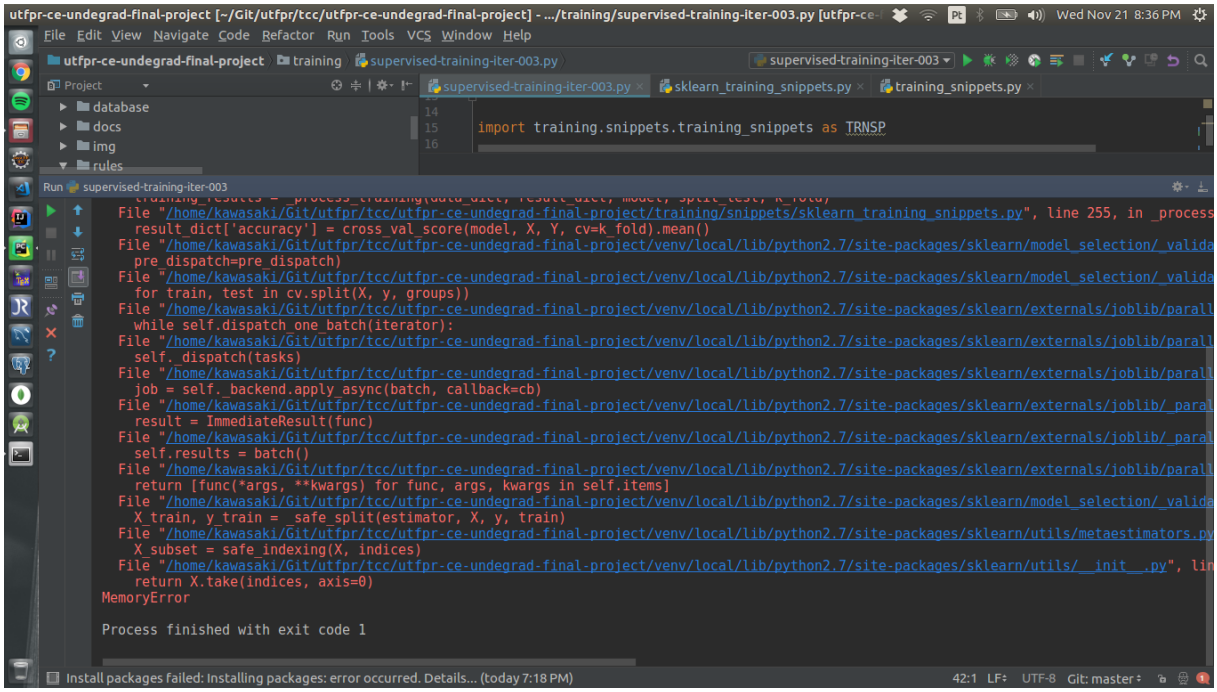
Fonte: Autoria própria

Figura 14 – Execução do modelo desbalanceado utilizando o *cross-validation* de 10 *k-folds*



Fonte: Autoria Própria

Figura 15 – Mensagem de estouro de memória referente à interrupção do processo do modelo desbalanceado



Fonte: Autoria Própria

As Tabelas 3 e 4 mostram que os resultados não apresentaram diferenças relevantes com relação ao tipo de *split* realizado entre métodos de *hold-out* e *k-folds*, porém o balanceamento

Tabela 4 – Médias obtidas nos treinamentos para amostras balanceadas nas três iterações realizadas

	Hold-out de 80-20%			**10 k-folds**		
Iter.	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
IT01	96%	96%	96%	95%	95%	95%
IT02	83%	81%	81%	83%	83%	83%
IT03	79%	77%	78%	78%	78%	78%

Fonte: Autoria própria

Tabela 5 – Resultados de tempo de execução da IT3 utilizando amostras balanceadas

Método	**Hold-out de 80-20%**	**10 k-folds**
Logistic Regression	0.38 s	24.84 s
Decision Tree	11.06 s	293.59 s
SVM SVC Linear	689.39 s	13667.14 s
Multinomial Naive Bayes	0.38 s	25.68 s
Random Forest	3.01 s	86.88 s
Stochastic Gradient Descent Log	5.03 s	135.64 s
Stochastic Gradient Descent SVM	3.34 s	96.32 s

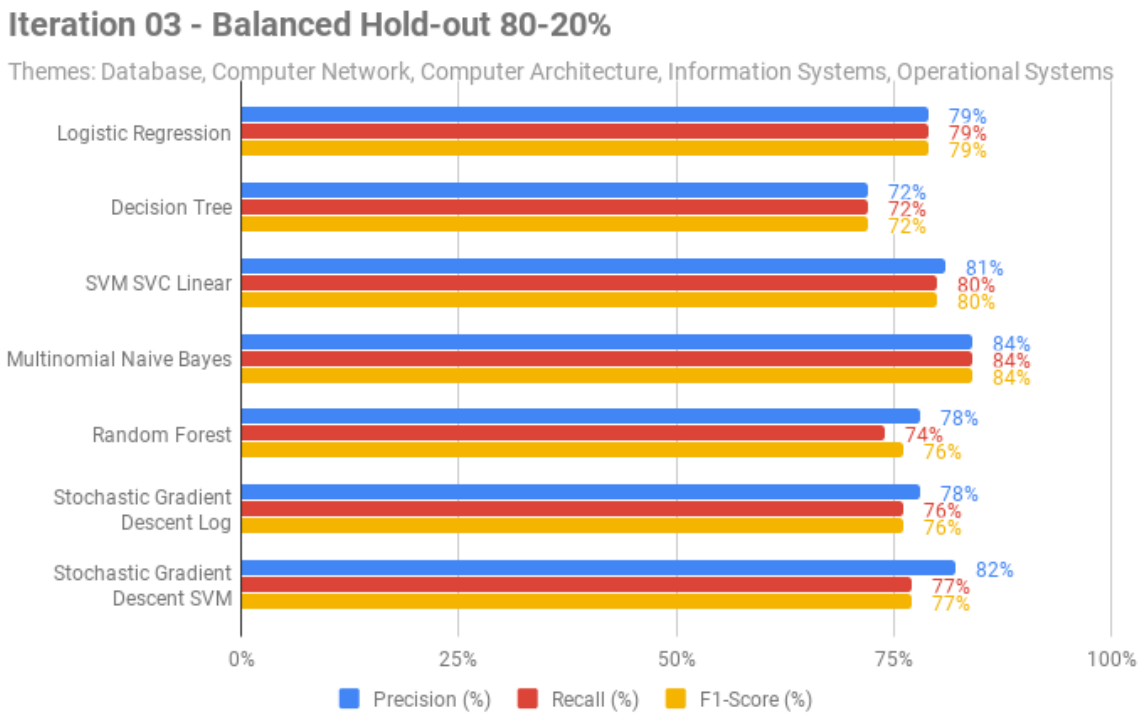
Fonte: Autoria própria

apresentou uma melhoria nos resultados da Iteração 2, chegando a aumentar o valor de *precision* de 3 a 5%. Além disso, os valores de performance foram diminuindo conforme a inserção de novas temáticas em cada iteração (redução de 17% em *precision* entre a primeira e a terceira iteração), revelando a dificuldade do modelo conforme a inserção de novas classes a serem treinadas e testadas.

Segundo resultados de métricas de *precision*, *recall* e *f-score* apresentados nas Figuras 16 e 17, os algoritmos que apresentaram melhores resultados na iteração final foram os mesmos utilizados pelo trabalho relacionado realizado por Fu, Qu e Wang (2009): Redes *Bayesianas* Multinomiais e *SVM* - 84% e 81% para método de *hold-out*, 83% e 80% para método de *k-folds*. Porém, como pode ser visualizado na Tabela 5, o algoritmo de *SVM* obteve um alto tempo de processamento (aproximadamente 11 minutos para *hold-out* e 4 horas para *k-folds*), enquanto o algoritmo de Redes *Bayesianas* Multinomiais conseguiu obter um resultado de *precision* próximo ao algoritmo de *SVM* com aproximadamente 0,06% do tempo para *Hold-out* e 0,2% do tempo para *k-folds*.

Em contraposição, os algoritmos que apresentaram os resultados mais inferiores comparados aos outros foram os de *Decision Tree* (*precision* de 72% para *Hold-out* e 75% para *k-folds*) e *Stochastic Gradient Descent Log* (*precision* de 74% para *Hold-out* e 75% para

Figura 16 – Resultados da IT3 utilizando amostras balanceadas e método de *hold-out*



Fonte: Autoria Própria

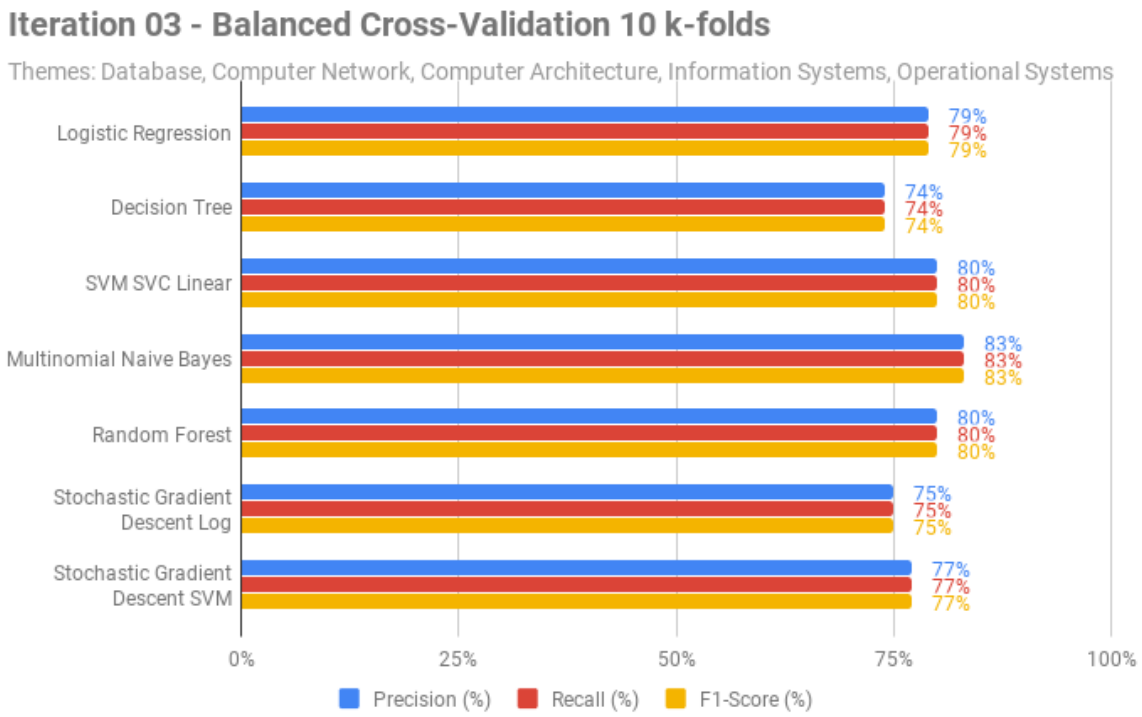
k-folds), além do *Stochastic Gradient Descent SVM* e *Random Forest* para *Hold-out*, os quais apresentaram uma diferença de respectivamente 5% e 4% entre *precision* e *recall*. Em geral, a média dos algoritmos alcançou um valor de 78% de precisão, enquanto os trabalhos de Xiao, Chow e Chen (2017) e Fu, Qu e Wang (2009) apresentaram, respectivamente, médias de acurácia de 95,75% e 91,5%.

Ao comparar esse trabalho de acordo com a quantidade de amostras (total de 14517 de acordo com a Tabela 2), pode-se verificar uma diferença relevante em relação ao trabalho de Xiao, Chow e Chen (2017), o qual teve um total de 52834 amostras no contexto jurídico; em contraste, o trabalho de Fu, Qu e Wang (2009) apresentou um total de 11000 amostras no contexto de suporte computacional, com uma diferença de apenas 3517 amostras a menos com relação a esse trabalho desenvolvido.

O resultado médio desse trabalho, portanto, apresenta um déficit de 17,75% para o trabalho de Xiao, Chow e Chen (2017) e de 13,5% para o trabalho de Fu, Qu e Wang (2009). A diferença, porém, obtida no trabalho de Fu, Qu e Wang (2009) se deve a utilização de um método complementar às etapas de pré-processamento e algoritmos supervisionados. Sem essas etapas, o trabalho obteve, respectivamente, médias de 77% para o algoritmo de Redes *Bayesianas* e de 81% para o algoritmo de *SVM*.

A partir de valores de precisão acima de 80% para os algoritmos supervisionados de Redes *Bayesianas*, *SVM* e *Random Forest* - próprios para treinamentos envolvendo classificações

Figura 17 – Resultados da IT3 utilizando amostras balanceadas e método de 10 *k-folds*



Fonte: Autoria Própria

de textos -, os resultados apresentados nesse trabalho demonstram tendências e evidências da relevância da abordagem proposta na resolução da hipótese (Q1) para classificação de questões da área computacional na língua portuguesa. A partir dessa conclusão, seria possível implementar essa etapa de classificação em um processo de elaboração de provas conforme apresentado na Seção 3.1. Essa etapa do ensino adaptativo poderia se beneficiar não apenas da metodologia de classificação e dos melhores classificadores, mas também incrementar a sua base de questões e estabelecer possíveis regras inferidas a partir do conjunto de palavras inerentes a cada temática.

5 CONSIDERAÇÕES FINAIS

As mudanças tecnológicas percebidas no campo educacional - por meio de aulas disruptivas e inovadoras - se devem a insatisfações quanto à desmotivação e a falta de atualização dos currículos educacionais, resultando em pesquisas e metodologias de ensino, como o ensino adaptativo.

A partir de um direcionamento customizado para cada aluno, o ensino adaptativo consegue um apoio acompanhado e com embasamento fundamentado em dados, apresentando uma alternativa ao caminho único de ensino para todos os alunos. Uma das etapas inerentes ao processo adaptativo consiste na aplicação de provas e questões customizadas para os pontos de melhoria de cada aluno. Por meio dela, o aluno consegue praticar seus pontos fracos por meio de um *feedback* contínuo, enquanto que professores podem construir tais provas e também obter relatórios de desempenho baseados em dados. O trabalho, portanto, baseou-se em aprimorar a parte do fornecimento de questões dessa etapa, obtendo e classificando as mesmas por meio da utilização de técnicas de PLN e aprendizado de máquina.

5.1 TRABALHOS FUTUROS

Os resultados da metodologia implementada demonstraram evidências da relevância do trabalho para classificação de questões na área da computação. Apesar da capacidade de aplicação dessa metodologia em um sistema de ensino adaptativo, algumas melhorias foram observadas para trabalhos futuros:

1. Aumento da quantidade de amostras e temáticas: o trabalho de Xiao, Chow e Chen (2017) apresentou uma quantidade de amostras de aproximadamente 4 vezes maior. Uma melhoria a ser testada seria verificar como o modelo de treinamento lidaria com uma maior quantidade de amostras e de temáticas;
2. Classificação em dois níveis: o trabalho de Fu, Qu e Wang (2009) apresentou, por meio de um processo complementar, resultados mais próximos de uma metodologia de classificação de questões ideal. A partir dos conceitos desse processo complementar, um trabalho futuro relevante seria a implementação de conceitos hierárquicos de ontologia e cálculo de similaridade semântica entre os termos ontológicos para verificar melhorias nos resultados;
3. Classificação de subtemáticas: o trabalho de Fu, Qu e Wang (2009) também realizou a classificação de 220 subtemáticas por meio de modelos de similaridades semânticas. Isso pode enriquecer ainda mais o ensino adaptativo, onde a recomendação de questões pode possuir um nível de assertividade mais profundo, alcançando um melhor nível de customização específica para cada aluno;
4. Textos em imagens: cerca de 13% (1855 amostras) desse trabalho apresentava algum

tipo de imagem na questão, seja no texto ou nas alternativas. Um trabalho futuro seria realizar uma técnica de *OCR*, complementando a quantidade de vetores de palavras e verificando a influência desse aspecto no resultado final.

Além de tais considerações, um trabalho futuro importante consiste na aplicação de treinamentos semi-supervisionados e não-supervisionados, contribuindo para coleta de questões com pouca ou nenhuma classificação. A extração de modelos treinados também pode acelerar o processo de melhoria, onde novas amostras de questões podem ser inseridas nos modelos sem a necessidade de execução repetida de grandes quantidades de amostras.

A discussão desses possíveis trabalhos futuros, os objetivos alcançados e os códigos executados nesse trabalho podem ser acessados no repositório do *Github*, o qual apresenta também como cada processo foi executado em nível de código[1].

Essas evoluções - tanto alcançadas como a serem desenvolvidas - demonstram o nível de complexidade envolvido na classificação de questões, considerando apenas um grau de especificidade na temática de computação. A partir da realização de cada uma dessas evoluções, pode-se chegar mais próximo do objetivo de contribuir e aprimorar o processo customização de provas do ensino adaptativo.

[1] Repositório disponível em https://github.com/davikawasaki/utfpr-ce-undergrad-final-project

Referências

ACM; SOCIETY, I. C. **Computer Engineering Curricula 2016**. [S.l.], 2016. Citado na página 13.

ATHENIKOS, S. J.; HAN, H.; BROOKS, A. D. Semantic analysis and classification of medical questions for a logic-based medical question-answering system. In: IEEE. **IEEE International Conference on Bioinformatics and Biomedicine Workshops**. 2008. p. 111–112. Disponível em: <https://ieeexplore.ieee.org/document/4686218/>. Citado na página 9.

BIGARELLI, B. **A nova fórmula da Kroton**. 2016. Notícia. Disponível em: <http://bit.ly/formula-kroton>. Citado 2 vezes nas páginas 3 e 4.

BLUM, A.; KALAI, A.; LANGFORD, J. Beating the hold-out: Bounds for k-fold and progressive cross-validation. In: **Proceedings of the Twelfth Annual Conference on Computational Learning Theory**. New York, NY, USA: ACM, 1999. (COLT '99), p. 203–208. ISBN 1-58113-167-4. Disponível em: <http://doi.acm.org/10.1145/307400.307439>. Citado na página 21.

BROWNLEE, J. **Gradiente Descent For Machine Learning**. 2013. Disponível em: <https://machinelearningmastery.com/gradient-descent-for-machine-learning/>. Citado na página 26.

CARVALHO, R. **O trabalho de professora em uma das escolas mais inovadoras do mundo**. 2015. Notícia. Disponível em: <https://www.napratica.org.br/trabalho-lumiar/>. Citado na página 3.

CHAWLA et al. Editorial: Special issue on learning from imbalanced data sets. **SIGKDD Explor. Newsl.**, ACM, New York, NY, USA, v. 6, n. 1, p. 1–6, jun. 2004. ISSN 1931-0145. Disponível em: <http://doi.acm.org/10.1145/1007730.1007733>. Citado na página 20.

CHRISTENSEN, C. **Disrupting Class: How Disruptive Innovation Will Change the Way the World Learns**. [S.l.]: McGraw-Hill, 2008. Citado na página 2.

COVINGTON, M. A.; NUTE, D.; VELLINO, A. **Prolog Programming in Depth**. [S.l.: s.n.], 1997. Citado na página 6.

DIETTERICH, T. G. Approximate statistical tests for comparing supervised classification learning algorithms. **Neural Comput.**, MIT Press, Cambridge, MA, USA, v. 10, n. 7, p. 1895–1923, out. 1998. ISSN 0899-7667. Disponível em: <http://dx.doi.org/10.1162/089976698300017197>. Citado na página 21.

DODIYA, T.; JAIN, S. Question classification for medical domain question answering system. In: IEEE. **IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)**. 2016. p. 204–207. Disponível em: <https://ieeexplore.ieee.org/document/8009118/>. Citado na página 9.

FAN, Z.; SU, L.; LIU, X. Multi-label chinese question classification based on word2vec. In: IEEE. **The 2017 4th International Conference on Systems and Informatics (ICSAI 2017)**. 2017. p. 546–550. Disponível em: <https://ieeexplore.ieee.org/document/8248351/>. Citado 2 vezes nas páginas 9 e 10.

FERRAZ, A. P. do C. M.; BELHOT, R. V. Taxonomia de bloom: revisão teórica e apresentação das adequações do instrumento para definição de objetivos instrucionais. **Scielo Gest. Prod**, v. 17, n. 2, p. 421–431, 2010. Disponível em: <http://www.scielo.br/pdf/gp/v17n2/a15v17n2.pdf>. Citado na página 9.

FU, J.; QU, Y.; WANG, Z. Two level question classification based on svm and question semantic similarity. In: IEEE. **International Conference on Electronic Computer Technology**. 2009. p. 366–370. Disponível em: <https://ieeexplore.ieee.org/document/4795985/>. Citado 6 vezes nas páginas 10, 11, 12, 29, 30 e 32.

GARDNER, H. **Frames of Mind: The Theory of Multiple Intelligences**. [S.l.]: Basic Books, 1983. Citado na página 2.

GEORGE, S.; JOSEPH, S. Text classification by augmenting bag of words (bow) representation with co-ocurrence feature. In: IOSR JOURNAL OF COMPUTER ENGINEERING (IOSR-JCE). **IOSR Volume 16, Issue 1**. 2014. v. 16, n. 1, p. 34–38. Disponível em: <https://pdfs.semanticscholar.org/f432/cbc0e35e6560fc657ad6b490aa07ad901575.pdf>. Citado 2 vezes nas páginas 7 e 8.

GRAY, P. A brief history of education. **Psychology Today**, 2008. Disponível em: <http://www.weisun.org/apworld/assign/readings/a_brief_history_of_edu_anno.pdf>. Citado 2 vezes nas páginas 1 e 2.

GUPTA, P. **Cross-Validation in Machine Learning**. 2017. Blog. Disponível em: <https://towardsdatascience.com/cross-validation-in-machine-learning-72924a69872f>. Citado 2 vezes nas páginas 20 e 21.

HAN, J.; KAMBER, M.; PEI, J. **Data Mining: concepts and techniques**. 3rd. ed. [S.l.]: Morgan Kaufmann, 2011. Citado na página 17.

HSU, C.-W.; CHANG, C.-C.; LIN, C.-J. A practical guide to support vector classification. In: . [s.n.], 2003. Disponível em: <https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>. Citado na página 23.

JAYAKODI, K.; BANDARA, M.; MEEDENIYA, D. An automatic classifier for exam questions with wordnet and cosine similarity. In: IEEE. **Moratuwa Engineering Research Conference (MERCon)**. 2016. p. 12–17. Disponível em: <https://ieeexplore.ieee.org/document/7480108/>. Citado na página 9.

JAYAKODI, K.; BANDARA, M.; PERERA, I. An automatic classifier for exame questions in engineering: a process for bloom's taxonomy. In: IEEE. **2015 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)**. 2015. p. 195–202. Disponível em: <http://ieeexplore.ieee.org/document/7386043/>. Citado 4 vezes nas páginas 7, 9, 11 e 12.

JIAO, Y.; DU, P. Performance measures in evaluating machine learning based bioinformatics predictors for classifications. **Quantitative Biology**, v. 4, n. 4, p. 320–330, December 2016. Disponível em: <https://link.springer.com/content/pdf/10.1007%2Fs40484-016-0081-2.pdf>. Citado na página 16.

KAHADUWA, H.; PATHIRANA, D.; ARACHCHI, P. L. Question answering system for the travel domain. In: IEEE. **Moratuwa Engineering Research Conference (MERCon)**. 2017.

p. 449–454. Disponível em: <https://ieeexplore.ieee.org/document/7980526/>. Citado na página 10.

KORB, K. B.; NICHOLSON, A. E. **Bayesian Artificial Intelligence**. 2nd. ed. Boca Raton, FL, USA: CRC Press, Inc., 2010. ISBN 1439815917, 9781439815915. Citado na página 24.

KOSALA, H. B. R. Web mining research: A survey. In: **SIGKDD Explorations**. ACM, 2000. v. 2, n. 1, p. 1–15. Disponível em: <https://arxiv.org/pdf/cs/0011033.pdf>. Citado na página 17.

LAZYPROGRAMMER. **Natural Language Processing in Python: Master Data Science and Machine Learning for spam detection, sentiment analysis, latent semantic analysis, and article spinning (Machine Learning in Python)**. [S.l.]: LazyProgrammer, 2016. Citado na página 8.

LEE, G. **Which universities have the highest first year dropout rates?** 2017. Notícia. Disponível em: <https://www.channel4.com/news/factcheck/which-universities-have-the-highest-first-year-dropout-rates>. Citado na página 4.

LI, Y.; JAIN, A. Classification of text documents. In: IEEE. **Fourteenth International Conference on Pattern Recognition, 1998**. 1998. Disponível em: <http://ieeexplore.ieee.org/document/711938/>. Citado 2 vezes nas páginas 7 e 8.

LIDDY, E. D. Natural language processing. **Library and Information Science Commons**, n. 2, 01 2001. Disponível em: <https://surface.syr.edu/cgi/viewcontent.cgi?article=1043&context=istpub>. Citado na página 6.

LORENZONI, M. **Ensino adaptativo: um guia prático para você começar**. 2016. Notícia. Disponível em: <http://info.geekie.com.br/ensino-adaptativo/>. Citado na página 3.

MANNING, C. D.; SCHÜTZE, H. **Foundations of Statistical Natural Language Processing**. [S.l.]: The MIT Press, 1999. Citado na página 7.

MCGRAW-HILL. **Adaptive Learning's Next Audience: Struggling K-12 Students**. 2015. Disponível em: <https://www.mheducation.com/ideas/adaptive-learnings-next-audience-struggling-k-12-students.html>. Citado na página 4.

MONTANDON, C.; PERRENOUD, P. **Entre parents et enseignants: un dialogue impossible?** [S.l.]: Peter Lang, 1987. Citado na página 1.

MULHERN, J. **A history of education: A social interpretation**. 2. ed. [S.l.]: Ronald Press Co, 1959. Citado na página 2.

PEREIRA, S. do L. **Processamento de Linguagem Natural**. 2018. Article. Disponível em: <https://www.ime.usp.br/~slago/IA-pln.pdf>. Citado na página 7.

POWERS, D. M. W. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. **Machine Learning Technologies**, v. 2, n. 1, p. 37–63, 2011. Disponível em: <https://bioinfopublication.org/files/articles/2_1_1_JMLT.pdf>. Citado na página 17.

QUINLAN; R., J. Simplifying decision trees. **Int. J. Man-Mach. Stud.**, Academic Press Ltd., London, UK, UK, v. 27, n. 3, p. 221–234, set. 1987. ISSN 0020-7373. Disponível em: <http://dx.doi.org/10.1016/S0020-7373(87)80053-6>. Citado na página 22.

RAO, V.; S., V.; KANNAN, V. Automatic identification of subject domain in engineering examination questions. In: IEEE. **IEEE Eighth International Conference on Technology for Education (T4E)**. 2016. p. 106–109. Disponível em: <https://ieeexplore.ieee.org/document/7814803/>. Citado 2 vezes nas páginas 10 e 11.

RIGBY, C. **How software that learns as it teaches is upgrading Brazilian education**. 2016. News. Disponível em: <https://www.theguardian.com/technology/2016/jan/10/geekie-educational-software-brazil-machine-learning>. Citado na página 4.

ROKACH, L. Data mining and knowledge discovery handbook. In: _____. [S.l.]: Springer US, 2010. cap. Ensembled Methods in Supervised Learning, p. 959–980. Citado na página 25.

ROKACH, L.; MAIMON, O. Data mining and knowledge discovery handbook. In: _____. [S.l.]: Springer US, 2010. cap. Classification Trees, p. 149–174. Citado na página 22.

SANGODIAH, A.; AHMAD, R.; AHMAD, W. F. W. A review in feature extraction approach in question classification using support vector machine. In: IEEE. **2014 IEEE International Conference on Control System, Computing and Engineering**. [S.l.], 2014. p. 536–541. Citado na página 8.

SARROUTI, M.; LACHKAR, A.; OUATIK, S. E. A. Biomedical question types classification using syntactic and rule based approach. In: IEEE. **7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K)**. 2015. p. 265–272. Disponível em: <https://ieeexplore.ieee.org/document/7526929/>. Citado na página 9.

SCIKITLEARN. **Ensemble Methods**. 2011. Disponível em: <http://scikit-learn.org/stable/modules/ensemble.html#forest>. Citado na página 25.

SCIKITLEARN. **Stochastic Gradient Descent**. 2011. Disponível em: <http://scikit-learn.org/stable/modules/sgd.html>. Citado na página 26.

SEBASTIANI, P.; ABAD, M. M.; RAMONI, M. F. Data mining and knowledge discovery handbook. In: _____. [S.l.]: Springer US, 2010. cap. Bayesian Networks, p. 175–208. Citado na página 24.

SHMILOVICI, A. Data mining and knowledge discovery handbook. In: _____. [S.l.]: Springer US, 2010. cap. Support Vector Machines, p. 231–248. Citado na página 23.

SIMAS, A. **As graduações campeãs de desistência**. 2012. Notícia. Disponível em: <http://www.gazetadopovo.com.br/educacao/vida-na-universidade/ufpr/as-graduacoes-campeas-de-desistencia-26khijqty1gurtas1veawhyz2>. Citado na página 4.

SU, L.; LIAO, H.; YU, Z. Ensemble learning for question classification. In: IEEE. **IEEE International Conference on Intelligent Computing and Intelligent Systems**. 2009. p. 501–505. Disponível em: <https://ieeexplore.ieee.org/document/5358124/>. Citado na página 10.

WALRAD, C. C. The ieee computer society and acm's collaboration on computing education. **Computer Society**, v. 49, n. 03, p. 88–91, 03 2016. Disponível em: <https://www.computer.org/csdl/mags/co/2016/03/mco2016030088.pdf>. Citado 2 vezes nas páginas 13 e 14.

WILLINGHAM, D. T. **Why Don't Students Like School?: A Cognitive Scientist Answers Questions About How the Mind Works and What It Means for the Classroom**. [S.l.]: Jossey-Bass, 2009. Citado na página 2.

XIAO, G.; CHOW, E.; CHEN, H. Chinese questions classification in the law domain. In: IEEE COMPUTER SOCIETY. **The Fourteenth IEEE International Conference on e-Business Engineering**. 2017. p. 214–219. Disponível em: <https://ieeexplore.ieee.org/document/8119153/>. Citado 5 vezes nas páginas 9, 10, 12, 30 e 32.

XIAO, G.; MO, J.; CHOW, E. Multi-task cnn for classification of chinese legal questions. In: IEEE. **IEEE 14th International Conference on e-Business Engineering (ICEBE)**. 2017. p. 84–90. Disponível em: <https://ieeexplore.ieee.org/document/8119134/>. Citado na página 9.

XU, J.; ZHOU, Y.; WANG, Y. A classification of questions using svm and semantic similarity analysis. In: IEEE. **Sixth International Conference on Internet Computing for Science and Engineering**. 2012. p. 31–34. Disponível em: <https://ieeexplore.ieee.org/document/6239781/>. Citado na página 10.

YING-WEI, L.; ZHENG-TAO, Y.; XIANG-YAN, M. Question classification based on incremental modified bayes. In: IEEE. **Second International Conference on Future Generation Communication and Networking**. 2008. p. 149–152. Disponível em: <https://ieeexplore.ieee.org/document/4734194/>. Citado na página 10.

ZHANG, H. The optimality of naive bayes. In: BARR, V.; MARKOV, Z. (Ed.). **Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2004)**. [S.l.]: AAAI Press, 2004. Citado na página 25.

Apêndices

APÊNDICE A – Resultados obtidos na Iteração 1

A.1 Tempo de Execução

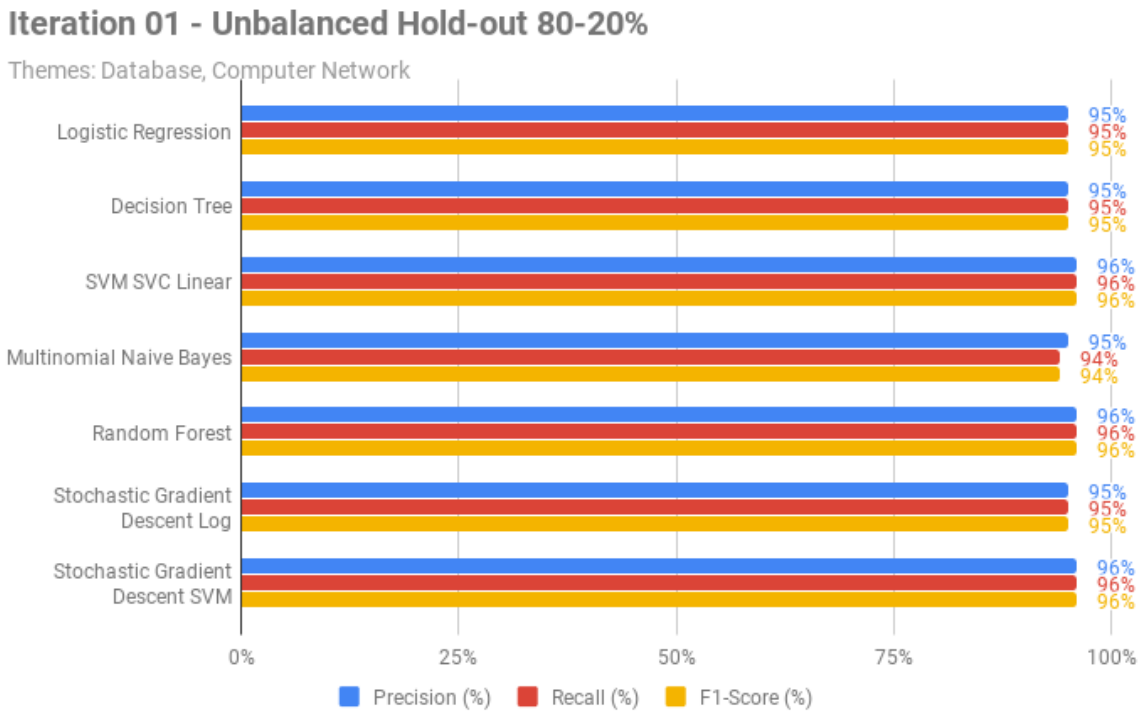
Tabela 6 – Resultados de tempo de execução da IT1 utilizando amostras balanceadas e não-balanceadas

	Amostras Balanceadas		**Amostras Desbalanceadas**	
Método	**Hold-out de 80-20%**	**10 k-folds**	**Hold-out de 80-20%**	**10 k-folds**
Logistic Regression	0.36 s	48.04 s	0.47 s	55.54 s
Decision Tree	28.85 s	826.28 s	38.7 s	1156.35 s
SVM SVC Linear	815.5 s	14714.31 s	1028.65 s	18336.8 s
Multinomial Naive Bayes	0.64 s	46.52 s	0.99 s	51.6 s
Random Forest	5.6 s	162.68 s	6.54 s	193.91 s
Stochastic Gradient Descent Log	2.05 s	75.36 s	4.04 s	87.83 s
Stochastic Gradient Descent SVM	1.43 s	62.4 s	1.64 s	72.96 s

Fonte: Autoria própria

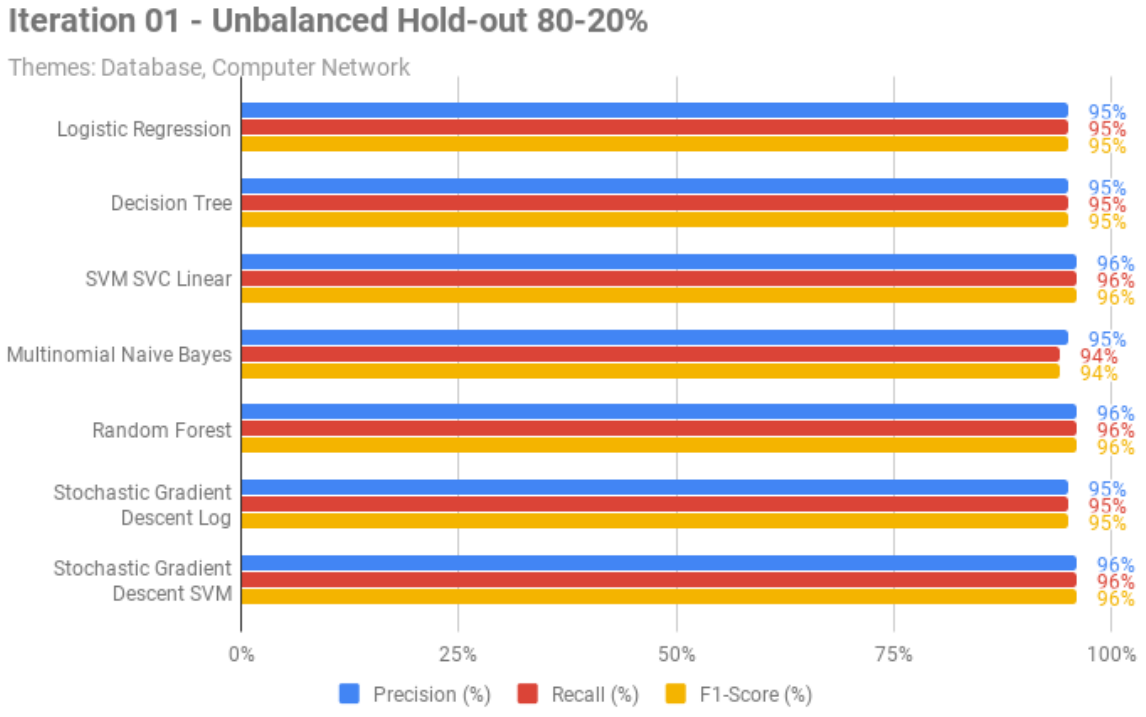
A.2 Métricas de performance

Figura 18 – Resultados da IT1 utilizando amostras desbalanceadas e método de *hold-out*



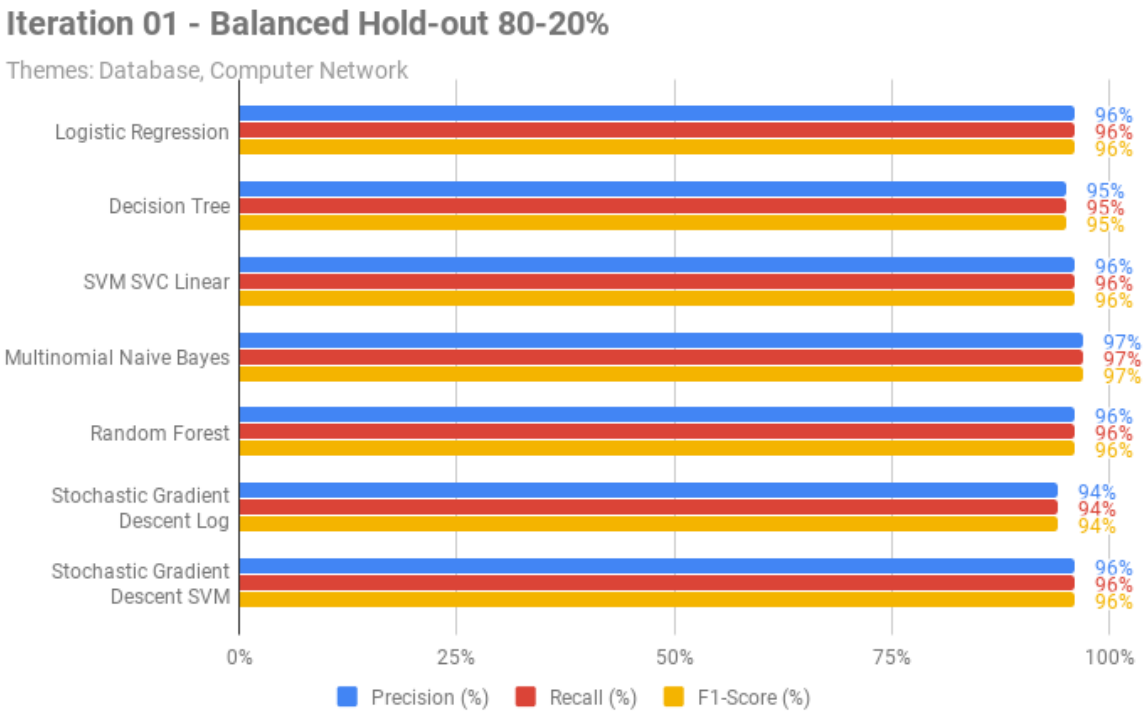
Fonte: Autoria Própria

Figura 19 – Resultados da IT1 utilizando amostras desbalanceadas e método de 10 *k-folds*



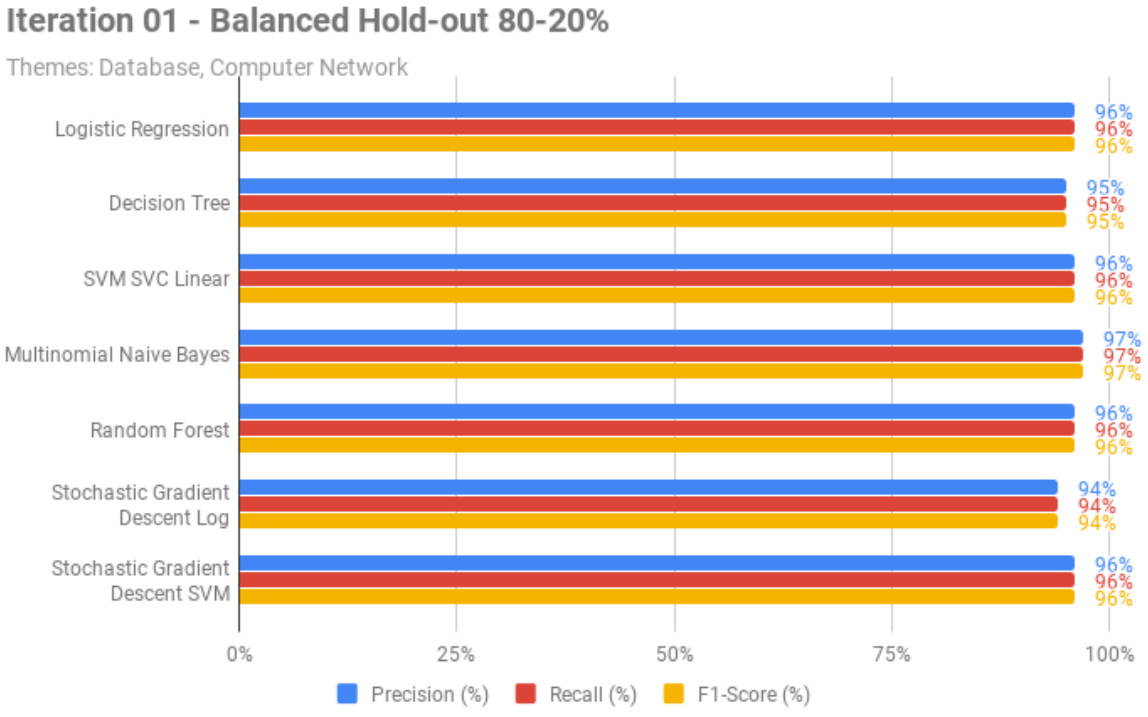
Fonte: Autoria Própria

Figura 20 – Resultados da IT1 utilizando amostras balanceadas e método de *hold-out*



Fonte: Autoria Própria

Figura 21 – Resultados da IT1 utilizando amostras balanceadas e método de 10 *k-folds*



Fonte: Autoria Própria

APÊNDICE B – Resultados obtidos na Iteração 2

B.1 Tempo de Execução

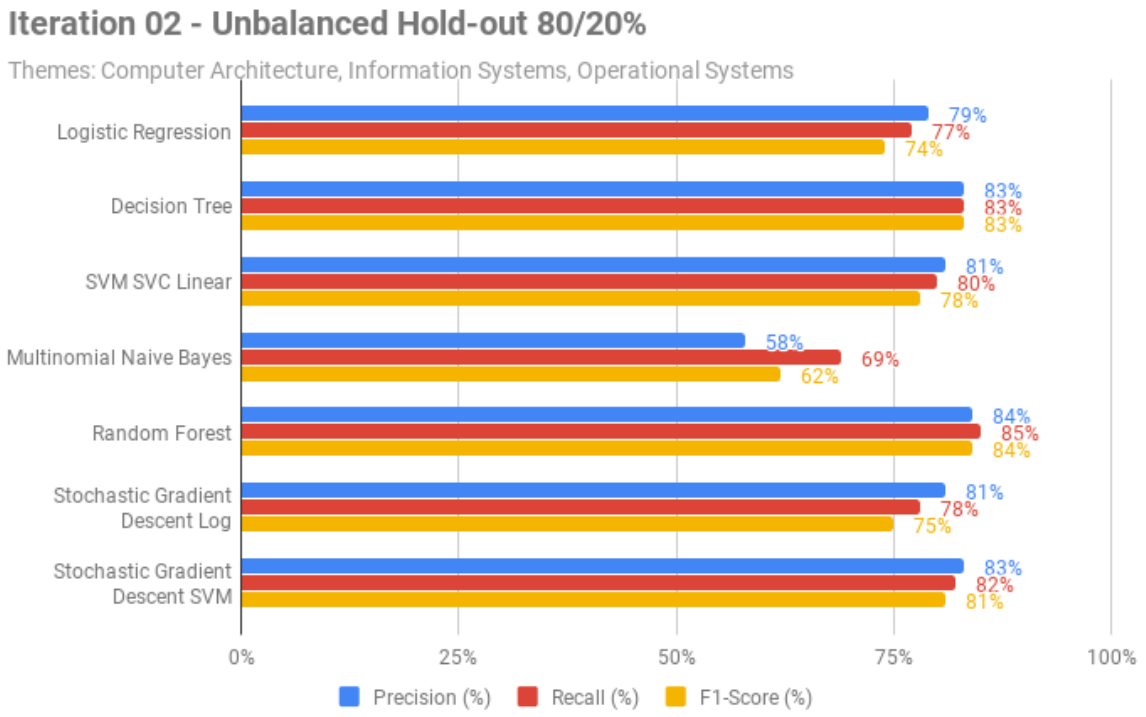
Tabela 7 – Resultados de tempo de execução da IT2 utilizando amostras balanceadas e não-balanceadas

	Amostras Balanceadas		**Amostras Desbalanceadas**	
Método	**Hold-out de 80-20%**	**10 k-folds**	**Hold-out de 80-20%**	**10 k-folds**
Logistic Regression	0.15 s	11.61 s	0.35 s	31.53 s
Decision Tree	4.03 s	112.7 s	22.01 s	568.57 s
SVM SVC Linear	170.75 s	3296.6 s	821.48 s	15731.09 s
Multinomial Naive Bayes	0.15 s	12.14 s	0.5 s	35.01 s
Random Forest	1.16 s	31.77 s	4.34 s	120.54 s
Stochastic Gradient Descent Log	1.46 s	42.49 s	4.37 s	122.15 s
Stochastic Gradient Descent SVM	1.07 s	33.16 s	3.08 s	93.43 s

Fonte: Autoria própria

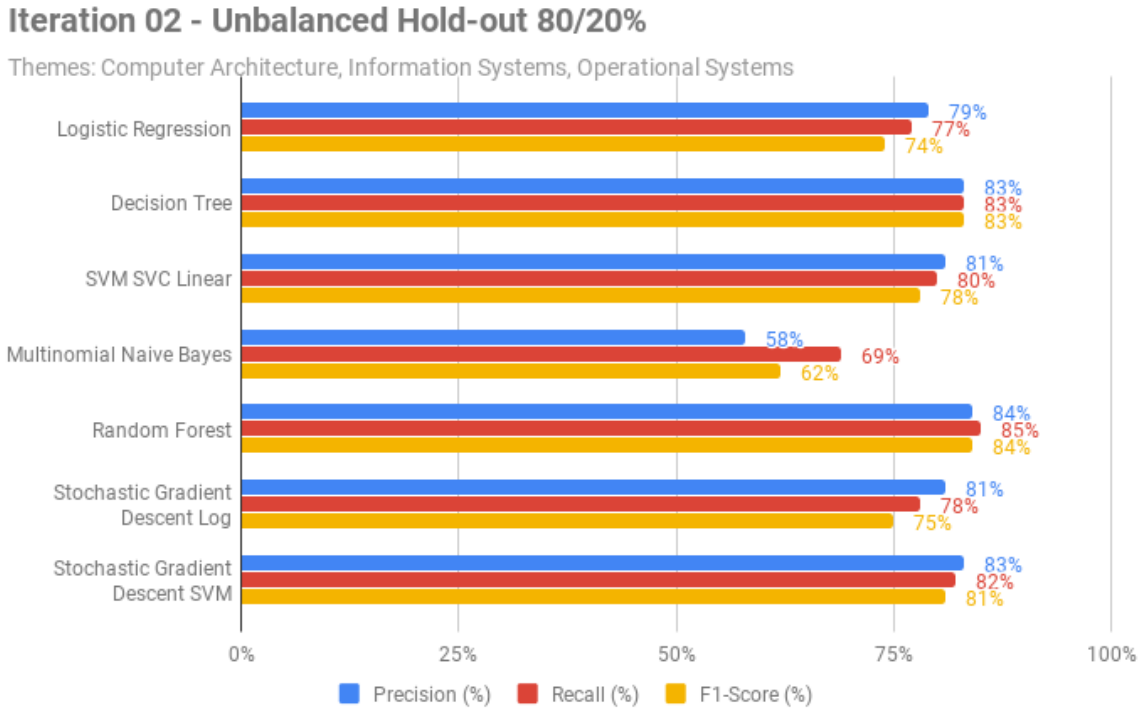
B.2 Métricas de performance

Figura 22 – Resultados da IT2 utilizando amostras desbalanceadas e método de *hold-out*



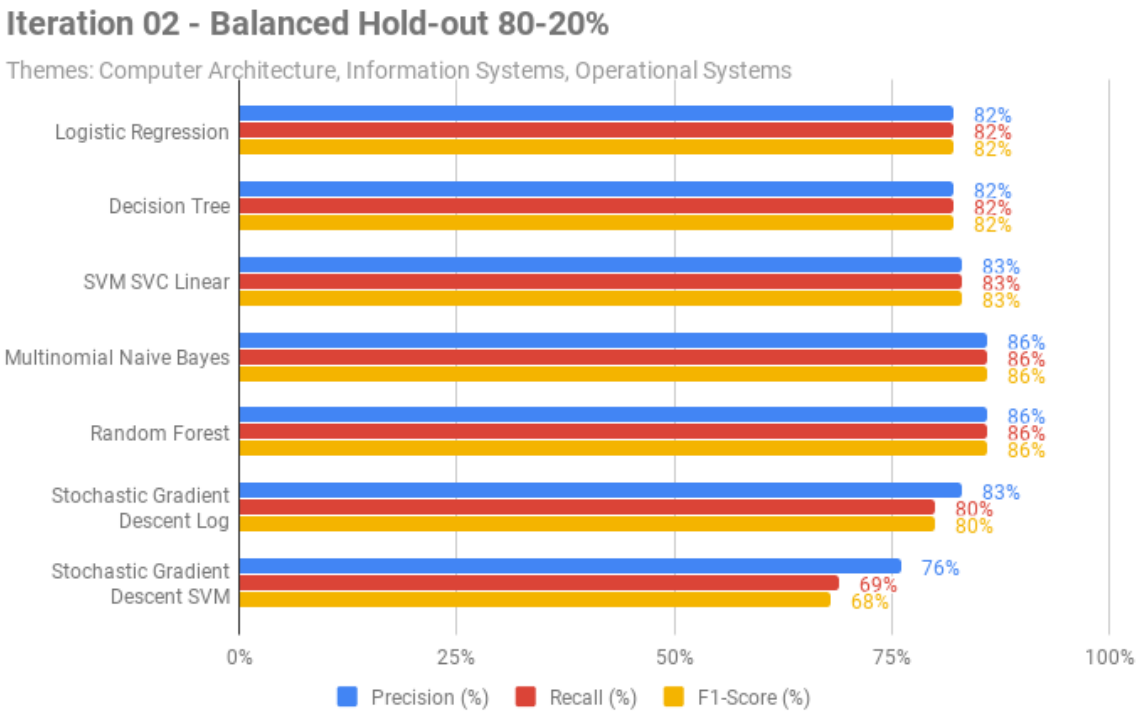
Fonte: Autoria Própria

Figura 23 – Resultados da IT2 utilizando amostras desbalanceadas e método de 10 *k-folds*



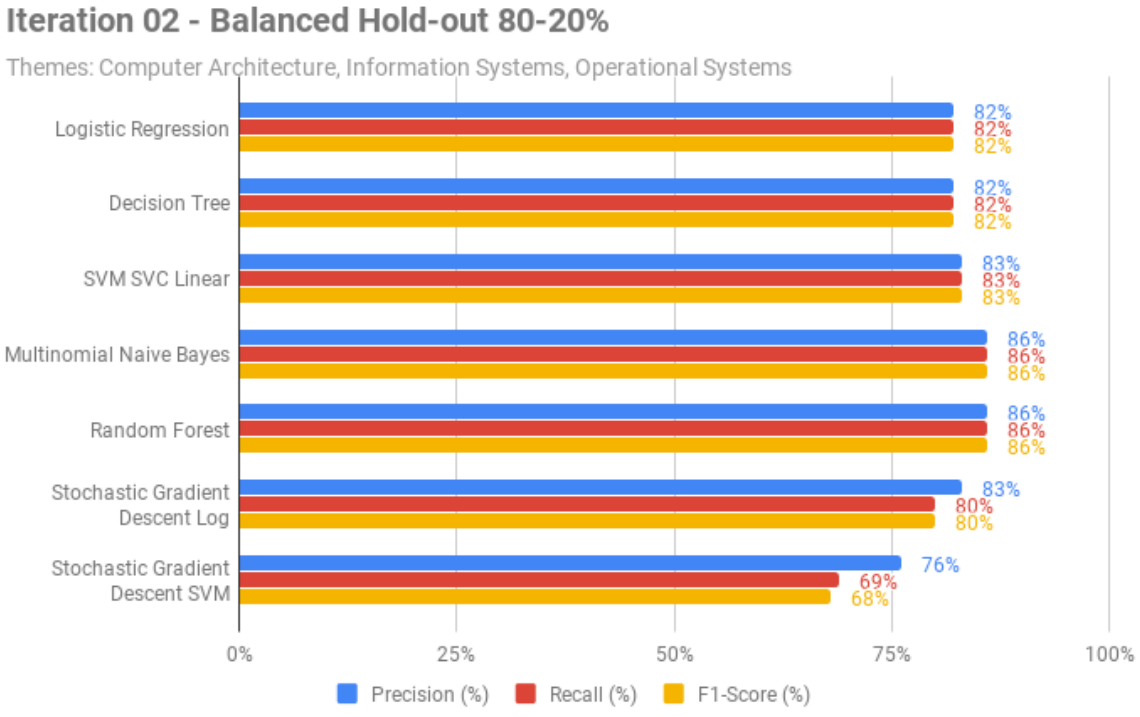
Fonte: Autoria Própria

Figura 24 – Resultados da IT2 utilizando amostras balanceadas e método de *hold-out*



Fonte: Autoria Própria

Figura 25 – Resultados da IT2 utilizando amostras balanceadas e método de 10 *k-folds*



Fonte: Autoria Própria

APÊNDICE C – Especificações de Hardware

As execuções dos algoritmos de treinamento seguiram as seguintes especificações de *hardware*:

1. Modelo: Dell Inspiron 14 7460
2. Memória: 8GB (1x8G) 2133MHz DDR4;
3. Processador: Intel Core i5-7200U CPU 2.50GHz x 4;
4. Placa Gráfica: NVIDIA GeForce; 940MX/PCIe/SSE2 4GB GDDR5;
5. Sistema Operacional: Ubuntu 16.04 LTS 64-bit.