

CRISTIAN SIMIONI MILANI

**UM MÉTODO PARA AVALIAÇÃO
AUTOMÁTICA DA QUALIDADE DE
RESPOSTAS EM FÓRUNS TÉCNICOS**

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática.

CURITIBA

2016

CRISTIAN SIMIONI MILANI

**UM MÉTODO PARA AVALIAÇÃO
AUTOMÁTICA DA QUALIDADE DE
RESPOSTAS EM FÓRUNS TÉCNICOS**

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática.

Área de Concentração: *Descoberta de Conhecimento e Aprendizagem de Máquina*

Orientador: Prof. Dr. Emerson Cabrera Paraiso

CURITIBA

2016

Dados da Catalogação na Publicação
Pontifícia Universidade Católica do Paraná
Sistema Integrado de Bibliotecas – SIBI/PUCPR
Biblioteca Central

Milani, Cristian Simioni
M637m Um método para avaliação automática da qualidade de respostas em
2016 fóruns técnicos / Cristian Simioni Milani ; orientador: Emerson Cabrera Paraiso.
– 2016.
79 f. : il. ; 30 cm

Dissertação (mestrado) – Pontifícia Universidade Católica do Paraná,
Curitiba, 2016
Bibliografia: f. 70-75

1. Mineração de dados (Computação). 2. Algoritmos de classificação.
3. Perguntas e respostas. 4. Pesquisa na internet. I. Paraiso, Emerson
Cabrera. II. Pontifícia Universidade Católica do Paraná. Programa de
Pós-Graduação em Informática. III. Título.

CDD 22. ed. – 006.312



Pontifícia Universidade Católica do Paraná
Escola Politécnica
Programa de Pós-Graduação em Informática

ATA DE SESSÃO PÚBLICA

DEFESA DE DISSERTAÇÃO DE MESTRADO Nº 06/2016

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA – PPGIa PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ - PUCPR

Em sessão pública realizada às 14h00 de 01 de Setembro de 2016, na Sala 224 – Escola de Negócios, ocorreu a defesa da dissertação de mestrado intitulada “Um Método para Avaliação Automática da Qualidade de Respostas em Fóruns Técnicos” apresentada pelo aluno **Cristian Simioni Milani**, como requisito parcial para a obtenção do título de **Mestre em Informática**, na área de concentração **Ciência da Computação**, perante a banca examinadora composta pelos seguintes membros:

1. **Prof. Dr. Emerson Cabrera Paraiso (orientador) – PUCPR**
2. **Prof. Dr. Julio Cesar Nievola - PUCPR**
3. **Ricardo Cerri - UFSCAR**
4. **Prof.ª Dr.ª Deborah Ribeiro Carvalho – PUCPR/PPGTS**

Após a apresentação da dissertação pelo aluno e correspondente arguição, a banca examinadora emitiu o seguinte parecer sobre a tese:

Membro	Parecer
Prof. Dr. Emerson Cabrera Paraiso	<input checked="" type="checkbox"/> Aprovada () Reprovada
Prof. Dr. Julio Cesar Nievola	<input checked="" type="checkbox"/> Aprovada () Reprovada
Prof. Dr. Ricardo Cerri	<input checked="" type="checkbox"/> Aprovada () Reprovada
Prof.ª Dr.ª Deborah Ribeiro Carvalho	<input checked="" type="checkbox"/> Aprovada () Reprovada

Portanto, conforme as normas regimentais do PPGIa e da PUCPR, a tese foi considerada:

APROVADA

(aprovação condicionada ao atendimento integral das correções e melhorias recomendadas pela banca examinadora, conforme anexo, dentro do prazo regimental)

() **REPROVADA**

E, para constar, lavrou-se a presente ata que vai assinada por todos os membros da banca examinadora. Curitiba, 01 de Setembro de 2016.



Prof. Dr. Emerson Cabrera Paraiso



Prof. Dr. Julio Cesar Nievola



Prof.ª Dr.ª Deborah Ribeiro Carvalho



Prof. Dr. Ricardo Cerri



ANEXO À ATA DE SESSÃO PÚBLICA
CORREÇÕES E MELHORIAS RECOMENDAS PELA BANCA EXAMINADORA

Dedico este trabalho a minha mãe Maria Luiza
e a todos os futuros pesquisadores que, assim como eu,
sonham transformar o mundo em um lugar melhor.

*“Without ambition one starts nothing.
Without work one finishes nothing.
The prize will not be sent to you.
You have to win it.”*

Ralph Waldo Emerson

Agradecimentos

Em primeiro lugar dedico este trabalho a minha família pela qual tenho um imenso carinho e gratidão, em especial a minha mãe Maria Luiza que mesmo tendo que cumprir o papel de mãe e pai forneceu uma base sólida para todas as minhas conquistas através de seu intenso empenho, dedicação e luta diária. Você é o meu maior orgulho e o melhor exemplo que um filho poderia ter.

A minha namorada, Jéssica Carina Pereira, por sua extrema paciência (mesmo não sendo sua principal característica) e compreensão, pelos momentos importantes em que estive ausente e por sua intensa participação durante toda a execução deste projeto através do seu apoio incondicional, carinho e, principalmente, sua adorável companhia.

Um agradecimento especial ao meu orientador, Prof. Dr. Emerson Cabrera Paraiso, pelo valioso conhecimento compartilhado, incentivo constante, críticas construtivas, conselhos, paciência e, principalmente, por acreditar, mesmo nos momentos mais adversos, que eu conseguiria concluir esta importante pesquisa.

À Prof.^a Dr. Deborah Ribeiro Carvalho que, no início da minha graduação, me aceitou como aluno de Iniciação Científica despertando assim o meu interesse na área da pesquisa. Além de, mesmo sem nenhuma obrigação, expressar por diversas oportunidades ao longo deste trabalho o seu valioso e intenso apoio.

A todos os mestres, desde o ensino fundamental até o presente momento que contribuíram para a minha formação com seus mais variados estilos de lecionar.

Aos amigos, que incentivaram, participaram e entenderam muitas vezes a minha ausência em momentos importantes e datas especiais. Um agradecimento especial a José Eduardo Nunes Lino, o Zé, pela parceria e amizade durante a graduação, mestrado e que, certamente, perdurará pelo resto da vida.

À Siemens, especificamente ao meu gestor, Frederico Papa Neto, por entender a importância da conclusão deste trabalho e apoiá-lo sempre que possível.

À PUCPR pela bolsa de estudos fornecida ao longo desta caminhada.

A todos que por ventura não foram citados, mas que, sem dúvida em algum ponto da minha jornada contribuíram para que eu evoluísse seja no plano pessoal ou profissional. Muito obrigado.

Sumário

Lista de Figuras	x
Lista de Quadros.....	xi
Lista de Tabelas	xii
Lista de Equações	xiii
Lista de Símbolos	xiv
Lista de Abreviaturas.....	xv
Resumo	xvi
Abstract	xvii
Capítulo 1	1
Introdução	1
1.1. Motivação	4
1.2. Objetivos.....	6
1.3. Hipóteses de Trabalho	6
1.4. Contribuição Científica	7
1.5. Escopo	7
1.6. Organização do Documento	7
Capítulo 2	9
Fundamentação Teórica	9
2.1. Fóruns de Perguntas e Respostas	9
2.1.1. Visão Geral.....	9
2.1.2. Estrutura Básica.....	11
2.1.3. Diferença Entre Fóruns Gerais e Fóruns Técnicos.....	12
2.2. O Fórum <i>Stack Overflow</i>	13
2.2.1. Estrutura e Conceitos	13
2.2.2. Sistema de Reputação	15
2.3. Mineração de Dados.....	16
2.3.1. Bases de dados.....	17
2.3.2. Pré-processamento.....	17
2.3.3. Algoritmos de Classificação	18
2.3.4. Pós-processamento	21

2.3.5. Avaliação dos Resultados	22
Capítulo 3	24
Trabalhos Relacionados	24
3.1. Predição da Qualidade de Respostas	24
3.2. A Utilização do Fórum <i>Stack Overflow</i> Durante o Desenvolvimento	28
3.3. Conclusão	31
Capítulo 4	34
Método	34
4.1. Pressupostos	34
4.2. Visão Geral	35
4.3. Seleção de Características	36
4.3.1. Apresentação de Qualidade	36
4.3.2. Avaliação	37
4.3.3. Tempo	37
4.3.4. Reputação	38
4.3.5. Características Seleccionadas	38
4.4. Aprendizagem e Avaliação do Modelo	41
4.5. Conclusão	43
Capítulo 5	44
Procedimentos Metodológicos	44
5.1. Base de Dados	44
5.1.1. Obtenção dos Dados	44
5.1.2. Pré-processamento	45
5.2. Extração de Características	51
5.3. Conclusão	53
Capítulo 6	54
Resultados	54
6.1. Bases de Dados	54
6.2. Experimentos	59
6.2.1. Experimento 1: Modelo Utilizando Todas as Características	60
6.2.2. Experimento 2: Modelo Utilizando Somente as Características Previamente Existentes	61
6.2.3. Experimento 3: Análise do Melhor Modelo Gerado pelo Algoritmo C4.5	62
6.2.4. Comparativo com o Estado da Arte	63

6.3. <i>Stack Overflow Predictor</i>	64
6.3.1. Modelo Preditivo Utilizado.....	65
6.3.2. A Ferramenta.....	65
6.4. Conclusão	66
Capítulo 7	68
Conclusão e Trabalhos Futuros	68
Referências Bibliográficas	70
Apêndice A	76
Aplicação <i>Stack Overflow Predictor</i>	76
A.1. Fluxo da Aplicação.....	76
A.2. Telas da Aplicação e Funcionamento.....	77

Lista de Figuras

Figura 1 – Exemplo de uma Pergunta em um Fórum Dedicado a Programadores.	2
Figura 2 – Estrutura de uma Pergunta em Fórum de Perguntas e Respostas	11
Figura 3 – Exemplo de uma Pergunta Específica em Fóruns Gerais.	12
Figura 4 – Exemplo de uma Resposta que Contém Bloco de Código.....	13
Figura 5 – Exemplo de <i>tags</i> no fórum <i>Stack Overflow</i>	14
Figura 6 – Sistema de Reputação Social no <i>Stack Overflow</i>	15
Figura 7 – Exemplo de Árvore de Decisão.	20
Figura 8 – Método <i>k-Fold</i>	20
Figura 9 – <i>Prompter</i> - Interface do Usuário.....	29
Figura 10 – Método para Predição de Qualidade de Respostas.	36
Figura 11 – Modelo de Saída do Algoritmo C4.5.	42
Figura 12 – Pergunta “Como Inverter uma String em JAVA”.....	46
Figura 13 – Resposta Selecionada pelo Questionador.....	47
Figura 14 – Esquema de Dados para Importação dos Dados do <i>Stack Overflow</i>	49
Figura 15 – Procedimentos Metodológicos e suas Etapas	53
Figura 16 – Presença de Blocos de Código na Resposta	55
Figura 17 – <i>Score</i> das Respostas Depositadas no Stack Overflow.....	56
Figura 18 – Porcentagem de Perguntas Respondidas nas Primeiras 48h.....	57
Figura 19 – Quantidade de Usuários por Faixa Etária.....	58
Figura 20 – Média de Reputação por Faixa Etária	58
Figura 21 – Porcentagem de Respostas Aceitas e Não Aceitas por Faixa de Reputação	59
Figura 22 – A Ferramenta <i>Stack Overflow Predictor</i>	66
Figura 23 – <i>Stack Overflow Predictor</i> : Tela Inicial.....	77
Figura 24 – <i>Stack Overflow Predictor</i> : Características do Usuário.....	78
Figura 25 – <i>Stack Overflow Predictor</i> : Características da Pergunta	78
Figura 26 – <i>Stack Overflow Predictor</i> : Respondendo uma Pergunta.....	79
Figura 27 – <i>Stack Overflow Predictor</i> : Resultado.....	79

Lista de Quadros

Quadro 1 – O que Perguntar e o que Não Perguntar no <i>Stack Overflow</i>	14
Quadro 2 – Ações Necessárias para Obter Medalhas no Fórum <i>Stack Overflow</i>	15
Quadro 3 – Regras de Classificação Extraídas da Árvore de Decisão	21
Quadro 4 – Matriz de Confusão para Duas Classes	22
Quadro 5 – Bases Consultadas e Termos da Busca.....	24
Quadro 6 – Comentário Extraído pela Ferramenta “ <i>AutoComment</i> ”	30
Quadro 7 – Comparação entre os Trabalhos de Predição de Qualidade de Respostas.....	32
Quadro 8 – Características Extraídas para Avaliar a Qualidade das Respostas.....	39
Quadro 9 – Descrição dos Arquivos do <i>Data Dump</i>	45
Quadro 10 – Representação em XML (Posts.xml) da Figura 12.	47
Quadro 11 – Representação em XML (Posts.xml) da Figura 13.	48
Quadro 12 – Modelo do Arquivo Gerado Após a Etapa de Pré-processamento.....	50

Lista de Tabelas

Tabela 1 – Média de Respostas Aceitas por Fórum.....	4
Tabela 2 – Acurácia do Treinamento para cada Conjunto de Características	26
Tabela 3 – Acurácia do Treinamento Utilizando o Algoritmo C4.5.....	27
Tabela 4 – Bases de Dados Utilizadas	55
Tabela 5 – Bases de Dados Pré-processadas.	60
Tabela 6 – Resultados para o Modelo Utilizando Todas as Características.....	61
Tabela 7 – Resultados para o Modelo Utilizando Somente as Características Previamente Conhecidas.....	62
Tabela 8 – Resultados da Análise do Melhor Modelo para o Algoritmo C4.5	63
Tabela 9 – Resultado do Artigo <i>Mining Successful Answers in Stack Overflow</i>	64
Tabela 10 – Comparativo: Estado da Arte vs. Resultados Obtidos	64

Lista de Equações

Equação 1 – Vetor de Características	17
---	----

Lista de Símbolos

TP	Verdadeiro positivo
TN	Verdadeiro negativo
FP	Falso positivo
FN	Falso negativo
$tacc(h)$	Precisão
$cov(h)$	Cobertura
$spec(h)$	Especificidade
$sens(h)$	Sensitividade
x	Vetor de características
x_n	Característica n do vetor de características

Lista de Abreviaturas

AM	Aprendizagem de Máquina
API	<i>Application Programming Interface</i>
CSV	<i>Comma-separated Values</i>
IDE	<i>Integrated Development Environment</i>
KDD	<i>Knowledge Discovery in Databases</i>
HTML	<i>HyperText Markup Language</i>
MLP	<i>Multi Layer Perceptron</i>
PHP	<i>Hypertext Preprocessor</i>
PLN	Processamento de Linguagem Natural
Q&A	<i>Question and Answer Websites</i>
RNA	Rede Neural Artificial
ROC	<i>Receiver Operator Characteristic Curve</i>
XML	<i>eXtensible Markup Language</i>

Resumo

Fóruns de perguntas e respostas se tornaram grandes repositórios para a obtenção e compartilhamento de conhecimento. Os fóruns direcionados a questões de programação são diariamente acessados por milhares de desenvolvedores que buscam soluções para seus problemas. Entretanto, o principal problema desses *websites* é a qualidade do conteúdo ali depositado. Cerca de 50% das perguntas não possui uma resposta julgada adequada pelo questionador. Além disso, dados recentes do *Stack Overflow*, em sua versão em Português, apontam que 23% das respostas fornecidas possuem um *score* negativo, revelando, assim, a insatisfação dos usuários. Na literatura alguns autores, buscando soluções para este problema, construíram processos automáticos para identificar o porquê as perguntas não são respondidas, determinar a qualidade dos blocos de código utilizados como exemplo e também analisar a qualidade das respostas fornecidas. Porém, resultados recentes, mostram que os modelos construídos para a predição da qualidade de respostas não estão adequados. Diante do exposto, o objetivo deste trabalho é desenvolver um modelo com o auxílio de algoritmos de Aprendizagem de Máquina (AM) capaz de prever a qualidade de respostas (sobre a perspectiva do questionador) em fóruns de desenvolvedores. Para isso, foram utilizadas 39 características quantitativas, específicas e não específicas do domínio da programação, extraídas das perguntas, respostas e do perfil dos usuários. As bases históricas utilizadas são provenientes do maior fórum de desenvolvedores, o *Stack Overflow*. Como métrica de avaliação de desempenho foi utilizada a acurácia dos modelos preditivos. Dentre os resultados, o melhor modelo (acurácia de 80,5%) foi construído pelo algoritmo *RandomForest*. Para os algoritmos que fornecem um modelo legível (capazes de serem interpretados por humanos) como saída, o algoritmo C4.5 obteve como melhor resultado uma acurácia de 77,8%. Finalmente, comparou-se os resultados obtidos com o estado da arte e conclui-se que o conjunto de características utilizado neste trabalho é mais eficiente na predição da qualidade de respostas em relação aos anteriormente sugeridos.

Palavras-Chave: Fóruns de Perguntas e Respostas, Qualidade de Respostas, Classificação, *Stack Overflow*.

Abstract

Question and Answer websites have become major repositories for obtaining and sharing knowledge. Forums that address programming issues are accessed daily by thousands of developers looking for solutions to their problems. However, the main problem with these websites is the quality of the content deposited there. About 50% of the questions do not receive a response judged adequate by the asker. Moreover, recent data from the Portuguese version of *Stack Overflow* indicate that 23% of the responses provided have a negative score, thus revealing the dissatisfaction of users. In the literature, some authors seeking solutions to this problem have built automated processes to identify why the questions are not answered, determine the quality of the code snippets used as examples, and also examine the quality of answers provided. However, recent results show that the models constructed to predict response quality are inadequate. With this in mind, the objective of this work is to develop a model using machine learning (ML) algorithms that are capable of predicting response quality (from the asker's perspective) in developer forums. To do so, 39 features extracted from the questions, answers, and user profiles were used; these characteristics were quantitative, specific and not specific to the programming domain. The historical bases used come from the largest developer forum, *Stack Overflow*. The accuracy of the predictive models was used as a performance evaluation metric. Among the results, the best model (80.5% accuracy) was built by the RandomForest algorithm. For algorithms that provide a readable model (which can be interpreted by humans) as output, the C4.5 algorithm yielded the best result with an accuracy of 77.8%. Finally, the results obtained were compared with the state of the art and it was concluded that the set of characteristics used in this work are more effective predictors of response quality than those mentioned previously.

Keywords: Question and Answers Websites, Answer Quality, Classification, Stack Overflow.

Capítulo 1

Introdução

O conteúdo disponível na *World Wide Web* sofreu grandes transformações desde o final do século passado. Em meados de 1990, quando a Web 1.0 surgiu, o conteúdo era exclusivamente estático, ou seja, a grande maioria dos *websites* disponíveis servia apenas para consultas simples. Em outras palavras, os usuários finais eram apenas consumidores das informações e opiniões ali disponíveis, pois a possibilidade de gerar conteúdo, modificar ou até mesmo interagir com os demais usuários da rede era limitada (CHOUDHURY, 2014). A criação do conteúdo era restrita a um pequeno grupo de publicadores o qual disponibilizava páginas estáticas construídas em HTML (*HyperText Markup Language*). Este cenário foi modificado com o surgimento da Web 2.0. O principal fator para essa mudança está diretamente ligado à sua característica bidirecional (AGHAEI, 2012), pois o conteúdo disponível se tornou dinâmico e, conseqüentemente, os usuários deixaram de ser apenas consumidores e passaram criar e interagir com conteúdo. Desta forma, a Web 2.0 potencializou a geração, organização e compartilhamento de conteúdo. As redes sociais são um ótimo exemplo: textos, fotografias, áudios, vídeos e outros diversos tipos de mídias são compartilhados em larga escala diariamente. Contextualizando em números, em um minuto: os usuários *Facebook*, por exemplo, geram 4 milhões de “curtidas”, 300 horas de novos vídeos são disponibilizadas no *Youtube*, 51 mil aplicativos são baixados da *Apple Store* e mais de 110 mil chamadas são realizadas utilizando o *Skype*¹.

Valendo-se deste novo contexto, novas formas de troca de conhecimento foram desenvolvidas. Fóruns de perguntas e respostas, conhecidos como Q&A (*Question and*

¹ Maiores informações em: <http://wersm.com/how-much-data-is-generated-every-minute-on-social-media/>

Answers) *websites*, surgiram devido ao sucesso das redes sociais e são construídos a partir da ideia de uma “arquitetura de participação” (TREUDE; BARZILAY; STOREY, 2011). Estes *websites* são um caso bastante particular de geração de conteúdo, pois são baseados no compartilhamento de conhecimento. Existem diversos *websites* deste gênero, como: *Amazon’s Askville*², *Yahoo! Answers*³, *Stack Overflow*⁴, entre outros. O mecanismo básico de funcionamento dos fóruns de perguntas e respostas baseia-se em: alguém com uma determinada dúvida elabora uma pergunta pública sobre o tema do fórum; o restante dos usuários responde – ou não – de acordo com seus conhecimentos; as respostas recebem votos positivos ou negativos dos usuários; as “melhores respostas” geralmente ficam no topo da lista das respostas fornecidas.

Alguns fóruns de perguntas e respostas são bastante gerais, como o *Yahoo! Answers*, onde há desde perguntas referentes ao cotidiano, até perguntas técnicas sobre um determinado assunto específico. Por outro lado, existem fóruns dedicados a determinados assuntos. A *Stack Exchange* é uma rede com mais de 150 *websites* de perguntas e respostas, onde cada um deles é relacionado a um tema diferente. Dentre eles, o *Stack Overflow*, dedicado a programadores e entusiastas da área com dúvidas relacionadas a questões de programação. A Figura 1 exibe um exemplo de uma pergunta, sobre a linguagem de programação JAVA, realizada no *Stack Overflow*.

Converter String com data para DateTime em Java

The screenshot shows a question on Stack Overflow. At the top, there is a title bar with a triangle icon, the text 'Tenho um arquivo CSV e nele tem uma coluna com a String:', a vote count of '5', and a timestamp 'Sun Oct 05 20:59:57 BRT 2014'. Below the title bar, the question text reads: 'Preciso passar os dados do CSV para uma tabela mysql. Sendo assim, preciso converter essa string para o formato DateTime para que eu consiga inserir essa string como uma DateTime no mysql.' There is a star icon and a '1' next to the question text. Below the text, there are three tags: 'java', 'string', and 'datetime'. At the bottom left, there are links for 'compartilhar' and 'editar'. At the bottom right, there is a box indicating the question was asked on '29/08/15 às 15:32' by 'Roger Rubens', who has 137 reputation points and 8 gold medals.

Figura 1 – Exemplo de uma Pergunta em um Fórum Dedicado a Programadores.

Fonte: <http://pt.stackoverflow.com/questions/83281>

² Maiores informações em: <http://askville.amazon.com>

³ Maiores informações em: <http://answers.yahoo.com>

Atualmente⁵ o *Stack Overflow*, em sua versão em Inglês, possui cerca de 12 milhões de perguntas e mais de 20 milhões de respostas sobre as mais diversas tecnologias. Nesta quantidade de conteúdo, as respostas disponíveis são frequentemente consultadas e utilizadas como substitutas à documentação oficial do produto ou tecnologia, especialmente quando este é extenso ou incompleto (TREUDE; BARZILAY; STOREY, 2011). Além disso, é comum na área da computação o surgimento de novas tecnologias, novas aplicações e até mesmo novas linguagens de programação, elevando assim a importância dos fóruns de perguntas e respostas como ferramenta para a troca de conhecimento e interação de programadores e entusiastas.

Neste cenário público e com APIs (*Application Programming Interface*) disponíveis para a obtenção dos dados existentes nestes fóruns, algumas pesquisas foram e estão sendo realizadas com o intuito de explorar o conhecimento ali depositado. A grande maioria dos trabalhos se concentra na identificação de especialistas em determinado assunto em fóruns de perguntas e respostas (PAL; CHANG; KONSTAN, 2012). Outra parte está focada em desenvolver aplicações que facilitem a vida do desenvolvedor, como: geradores automáticos de comentários para códigos (WONG; YANG; TAN, 2013), ranqueamento de conhecimento (DE SOUZA; CAMPOS; MAIA, 2014) e recomendadores de informação (PONZANELLI; BAVOTA; PENTA, 2014). Recentemente, os pesquisadores começaram a se preocupar em como classificar a qualidade do conteúdo disponível em fóruns de perguntas e respostas utilizando características quantitativas (numericamente mensuráveis) como, por exemplo, quantidade de palavras, quantidade de caracteres, entre outros (ARAI, 2014; CALEFATO; LANUBILE; MARASCIULO, 2015; JEON; CROFT; LEE, 2006; SHAH; ST; BRUNSWICK, 2010). Entretanto, os resultados obtidos não são adequados pois possuem baixa capacidade de generalização. Os modelos sugeridos podem ser aprimorados unindo os conjuntos de variáveis sugeridas pelos autores ou ainda buscando variáveis não exploradas. Além disso, a grande maioria dos trabalhos que busca explorar esse campo se concentra em fóruns gerais.

Desta forma, construir um método utilizando características quantitativas, específicas do escopo de programação (aquelas que são encontradas apenas em fóruns deste gênero) e não específicas, para a classificação da qualidade de respostas (percepção de valor do

⁴ Maiores informações em: <http://stackoverflow.com>

⁵ Consulta realizada em março de 2016.

conteúdo por parte do questionador) em fóruns de perguntas e respostas para desenvolvedores é a motivação para este trabalho de pesquisa.

1.1. Motivação

Fórum de perguntas e respostas criaram um novo paradigma no que se diz respeito a compartilhamento e, principalmente, busca de conhecimento. Milhões de usuários consultam esses fóruns diariamente em busca de soluções para seus problemas (LIU; AGICHTEIN; DROR, 2011). Entretanto, um dos principais desafios destes fóruns é lidar com a variação da qualidade da informação ali depositada. O conteúdo varia desde perguntas bem formuladas, respostas precisas e bem escritas até a baixa qualidade, como por exemplo, conteúdo abusivo (AGICHTEIN; CASTILLO; DONATO, 2008). Todos esses aspectos somados a uma quantidade cada vez maior de material inserido nestes fóruns, nos leva a uma necessidade de processos que automatizem a avaliação deste conteúdo.

Neste ponto da discussão é necessário apresentar, em números, a disparidade desta qualidade. Dada à modelagem dos fóruns de perguntas e repostas, cada pergunta pode receber inúmeras respostas. O usuário que elaborou a pergunta tem o direito de escolher uma delas como a que o ajudou na solução do problema. O fato de quem elaborou a pergunta escolher uma única resposta como a melhor é um indicador de satisfação (LIU; BIAN; AGICHTEIN, 2008). A Tabela 1 exibe a média de perguntas com respostas aceitas nos fóruns especializados para desenvolvedores disponíveis na rede *Stack Exchange*⁶.

Tabela 1 – Média de Respostas Aceitas por Fórum

Fonte: <http://stackexchange.com/sites>

Fórum	Quantidade de Perguntas	Quantidade de Perguntas com Respostas Aceitas	Porcentagem de Perguntas com Respostas Aceitas
stackoverflow (Inglês)	11.479.915	6.350.665	55,32%
pt.stackoverflow (Português)	41.643	21.946	52,70%
es.stackoverflow (Espanhol)	2.478	1.320	53,27%
ru.stackoverflow (Russo)	87.408	46.451	53,14%
Média	2.902.861	1.605.095	53,61%

⁶ Mais informações em: <http://stackexchange.com/>

Em média, cerca de apenas 50% das perguntas possui uma resposta selecionada pelo questionador, comprovando assim que os fóruns de perguntas e respostas necessitam de uma atenção especial relacionada à qualidade. Nesse contexto, a primeira motivação deste trabalho é criar um modelo que não apenas classifique a qualidade das respostas em fóruns de desenvolvedores, mas que também auxilie e estimule os usuários a fornecerem respostas com a qualidade esperada pelo questionador. Destarte, um modelo capaz de prever se a resposta provida é útil ou não para os usuários poderia incrementar a percentagem de perguntas com respostas aceitas e, conseqüentemente, a qualidade do conteúdo disponível em fóruns de perguntas e respostas.

A segunda grande motivação é resolver uma lacuna observada pelo autor na literatura. Trabalhos recentes (CALEFATO; LANUBILE; MARASCIULO, 2015; LIU; FENG; LIU, 2015) que se propuseram construir um modelo para avaliar a qualidade das respostas em fóruns de perguntas e respostas para desenvolvedores não obtiveram um bom resultado (taxa de acerto de apenas 64,58%). Por outro lado, autores (ARAI, 2014; JEON; CROFT; LEE, 2006; SHAH; ST; BRUNSWICK, 2010) que obtiveram uma acurácia superior a 90% considerando fóruns gerais não garantiram a generalização de seus modelos. Isto é, provavelmente se aplicados em respostas futuras, a taxa de 90% de acerto não seria respeitada. Além disso, a grande parte dos autores construiu os modelos sob a perspectiva dos fóruns gerais, quando estamos tratando de fóruns direcionados a desenvolvedores a percepção de uma resposta de qualidade pode ser diferente.

Por fim, vale destacar ainda uma curiosidade sobre os fóruns de desenvolvedores, especialmente o *Stack Overflow*. Ao fornecer respostas consideradas úteis pelo autor da pergunta e demais usuários da comunidade, o usuário que forneceu a resposta aumenta sua reputação. A reputação é um valor numérico que é incrementado de acordo com a realização de algumas ações, entre elas, fornecer respostas aceitas como a ideal pelo questionador. Esse valor vem sendo utilizado como fator, não decisivo, mas respeitável em entrevistas de emprego ao redor do mundo, pois comprova publicamente sua experiência em resolver problemas de programação⁷. Desta forma, ferramentas que forneçam um subsídio para os usuários aumentarem a sua reputação são bem vistas pela comunidade, principalmente pelo fato acima descrito.

⁷ Mais informações em: <http://www.itworld.com/article/2707360/it-management/your-stack-overflow-reputation-can-help-get-you-hired---by-stack-overflow.html>

Pelos motivos supracitados, este projeto se propõe a construir um modelo, contendo características quantitativas, específicas do domínio da programação (como por exemplo, a presença ou não de código na resposta, quantidade de blocos de código, entre outras) e não específicas, capaz de prever a qualidade de respostas em fóruns de desenvolvedores.

1.2. Objetivos

O objetivo principal desta pesquisa é desenvolver um método capaz de automaticamente prever a qualidade de respostas em fóruns de perguntas e respostas para desenvolvedores.

Objetivos específicos:

- Identificar as características quantitativas, específicas e não específicas, do escopo de programação que podem ser utilizadas para determinar a qualidade de uma resposta;
- Avaliar o modelo gerado por meio de métricas específicas;
- Desenvolver um protótipo computacional capaz de classificar novas respostas de acordo com a sua qualidade.

1.3. Hipóteses de Trabalho

As hipóteses deste projeto de pesquisa são:

- É possível construir um modelo utilizando características quantitativas, específicas e não específicas do escopo de programação, para determinar a qualidade de respostas em fóruns técnicos;
- O modelo construído auxilia os usuários a elaborarem respostas mais precisas e completas e, conseqüentemente, a percentagem de perguntas com respostas aceitas tende a aumentar.

1.4. Contribuição Científica

As principais contribuições científicas desta pesquisa são, (i) a criação de um método capaz de classificar automaticamente, sob a perspectiva do questionador, a qualidade de respostas em fóruns de desenvolvedores com o auxílio de modelos preditivos, (ii) o conjunto de características quantitativas extraídas para a construção do modelo e, por fim, (iii) um protótipo computacional que, em tempo real, classifica a resposta que está sendo redigida. O método resultante pode ser útil para melhorar a qualidade das respostas fornecidas pelos usuários em fóruns de perguntas e respostas. Além de contribuir diretamente com o aumento da reputação do usuário, que se demonstra um número valioso não somente no âmbito do fórum, mas também no âmbito profissional. Por fim, devido ao possível aumento da qualidade das respostas, espera-se também que a porcentagem de perguntas com respostas aceitas aumente.

Estas contribuições podem beneficiar toda a comunidade de programadores e entusiastas da área, proporcionando respostas mais precisas, completas e corretas a perguntas futuramente questionadas.

1.5. Escopo

O escopo deste projeto de pesquisa limita-se a classificação de respostas oriundas do fórum *Stack Overflow* em Português. É importante destacar que o método sugerido não possui nenhuma restrição em relação à linguagem.

1.6. Organização do Documento

Este trabalho está organizado em sete capítulos. O Capítulo 1 contém as considerações iniciais, a motivação, objetivos, hipóteses de trabalho, contribuições científicas e o escopo.

O Capítulo 2 está dividido em três seções que abordam os tópicos principais para o entendimento completo desta pesquisa: Fóruns de Perguntas e Respostas (2.1), O Fórum *Stack Overflow* (2.2) e Mineração de Dados (2.3).

O Capítulo 3 apresenta os trabalhos relacionados. Este capítulo foi dividido em três seções: Predição da Qualidade de Respostas (3.1), A Utilização do Fórum *Stack Overflow*

Durante o Desenvolvimento (3.2) e, por fim, uma breve conclusão sobre os trabalhos relacionados a este projeto (3.3).

O Capítulo 4 apresenta o método proposto e foi dividido em quatro seções: Pressupostos (4.1), Visão Geral (4.2), Seleção de Características (4.3), Aprendizagem e Avaliação do Modelo (4.4) e, finalmente, uma breve conclusão (4.5).

O Capítulo 5 traz os procedimentos metodológicos, descrevendo a parte técnica do projeto e está organizado em três seções: Base de Dados (5.1), Extração de Características (5.2) e, novamente, uma breve conclusão (5.3).

O Capítulo 6 apresenta os resultados obtidos neste trabalho, destacando as principais conquistas. Este capítulo está organizado em quatro seções: Base de Dados (6.1), Experimentos (6.2), *Stack Overflow Predictor* (6.3) e a conclusão do capítulo (6.4).

O Capítulo 7 apresenta a conclusão deste trabalho e os trabalhos futuros.

Por fim, no final do documento são listadas as referências bibliográficas utilizadas ao longo deste documento, bem como, os apêndices.

Capítulo 2

Fundamentação Teórica

Neste capítulo são apresentados os principais conceitos necessários para o entendimento completo deste trabalho. Além disso, é apresentado como os dados disponíveis em fóruns de perguntas e respostas podem ser utilizados por meio do processo de Aprendizagem de Máquina.

Na Seção 2.1 são apresentadas as características e as vantagens de um fórum de perguntas e repostas, bem como a diferença entre as versões gerais e a especializadas para desenvolvedores. Na sequência, a Seção 2.2 apresenta o fórum *Stack Overflow*, fonte de dados para este trabalho. Por fim, a Seção 2.3, é apresentada a Mineração de Dados, bem como suas principais tarefas de extração de conhecimento.

2.1. Fóruns de Perguntas e Respostas

2.1.1. Visão Geral

Fóruns de perguntas e respostas são ferramentas públicas de discussão disponíveis online e armazenam informações que muitas vezes não são encontradas em mecanismos de busca, como o *Google* (PERA; NG, 2011). A principal ideia destes *websites* é armazenar e organizar as informações fornecidas por seus usuários. Tipicamente, estes fóruns adotam um modelo simples onde a discussão é centralizada em volta de uma pergunta (GKOTSIS; STEPANYAN; PEDRINACI, 2014). Recentemente os fóruns de perguntas e respostas se

tornaram populares. Em 2002, um portal coreano⁸ lançou um *website* de perguntas e respostas que em 2006 já possuía mais de 37 milhões de perguntas (IHLWAN; WOYKE; ELGIN, 2006). Estes *websites* se diferenciam dos fóruns de discussão, pois, como destacado anteriormente, a ideia central é compartilhar o conhecimento com os usuários da comunidade por meio da resolução das perguntas ali depositadas.

Porém, estes *websites* levantaram uma questão social: “*Quais são as motivações para os usuários responderem perguntas em fóruns?*”. Neste ponto é necessário destacar que estes fóruns são formados graças a dois instintos humanos complementares: a necessidade de questionar e a inclinação das pessoas em contribuir com as outras através do compartilhamento do seu conhecimento (RABAN; HARPER, 2008). Existem dois grupos de motivação: as motivações intrínsecas e as motivações extrínsecas. Dentre as intrínsecas, destacam-se: percepção de valor, interação, conhecimento social, reciprocidade e gratidão. Por outro lado, os motivos extrínsecos são: acesso a tecnologia, reputação, status e pagamento (o *Google Answer* - extinto em 2006 - dava à oportunidade das pessoas que perguntavam oferecerem uma quantia em dinheiro, entre \$2 e \$200, pela resposta).

A grande maioria dos fóruns de perguntas e respostas possui um sistema de reputação social (motivação extrínseca) para incentivar os usuários a contribuírem com perguntas e respostas de qualidade (HART; SARMA, 2014). Além disso, este sistema tende a manter o usuário ativo no fórum e não apenas utilizá-lo como fonte de informação. Geralmente, com o aumento da reputação o usuário se torna capaz de interagir com outras funcionalidades do fórum, como por exemplo: inserir comentários em perguntas e respostas, sugerir edições para respostas, entre outras. Porém, mesmo com um sistema bastante robusto de incentivo a geração de conteúdo de qualidade é comum encontrar perguntas e respostas com déficit de qualidade. Bons exemplos de baixa qualidade são perguntas que contém spam ou que não possuem informações suficientes para prover uma resposta. Como consequência, a experiência do usuário é reduzida (LI; JIN; LYU, 2012). Independentemente do problema de qualidade, esses *websites* possuem algumas vantagens. Dentre elas destacam-se: acesso público, inúmeras perguntas previamente respondidas, diferentes contextos analisados para uma mesma pergunta e a organização da informação.

⁸ Mais informações em: www.naver.com

2.1.2. Estrutura Básica

A maioria dos fóruns possui como estrutura básica a possibilidade de realizar novas perguntas, responder as perguntas existentes e visualizar os perfis dos usuários que responderam as perguntas. A Figura 2 utiliza o fórum *Yahoo! Answers* para exemplificar as estruturas básicas de uma pergunta em um fórum de perguntas e repostas.

The screenshot shows a forum post on Yahoo! Answers. The breadcrumb navigation at the top is "Computadores e Internet > Internet" (1). The question title is "Quero saber o que é o tal URL?" (2). Below the title are options to "Seguir" (follow) and "7 repostas" (7 answers). A "Respostas" (Answers) section (3) is visible, featuring a "Melhor resposta" (Best answer) by Marco Antônio, posted 9 years ago. The best answer explains that a URL is a Universal Resource Locator and provides examples like "http://yahoo.com.br". Below the best answer are thumbs-up and thumbs-down icons (0 each) and a "Comentário" (Comment) button. A section for "Outras Respostas (6)" (4) shows another answer by pitagoras0583, also 9 years ago, which explains the components of a URL. This answer has 1 thumbs-up and 1 thumbs-down icon (5), and a "Comentário" button (6). At the bottom, there is a text input field for a new answer (7) with the placeholder text "Inclua sua resposta".

Figura 2 – Estrutura de uma Pergunta em Fórum de Perguntas e Respostas

Fonte: <https://br.answers.yahoo.com/question/index?qid=20060923211904AAkUq2x>

Quase sempre os fóruns possuem uma maneira de organizar suas postagens. O *Yahoo! Answers* separa as perguntas e repostas por categorias (1), neste caso “Computadores e Internet > Internet”. Por outro lado, fóruns para desenvolvedores utilizam, para organizar seu

conteúdo, outros marcadores que serão apresentados posteriormente. Toda postagem possui a pergunta (2) que o usuário compartilhou com a comunidade. Geralmente essa pergunta é sucedida por um texto explicativo, todavia, neste caso, como a pergunta é bastante direta, não houve tal necessidade. Posteriormente a lista de respostas é apresentada. A melhor resposta (3), considerada a escolhida pelo usuário como a que lhe ajudou a solucionar sua dúvida, usualmente aparece na primeira posição. As outras respostas (4) são listadas logo abaixo e também podem conter informações suficientemente úteis para ajudar outros usuários. Se este for o caso, os usuários podem avaliar (5) as repostas como úteis, caso contrário é possível realizar um voto negativo. Além disso, é possível realizar comentários (6) os quais geralmente contêm agradecimentos e informações adicionais à resposta anteriormente respondida. Por fim, pode-se submeter uma nova resposta (7) para a pergunta. A maioria dos fóruns suporta uma série de itens extras para a resposta como, por exemplo, o uso de código HTML para melhor organizar o conhecimento e a possibilidade de anexar arquivos.

2.1.3. Diferença Entre Fóruns Gerais e Fóruns Técnicos

Os fóruns de perguntas e respostas gerais são aqueles que não reduzem seu escopo a um determinado contexto. Dentre eles destacam-se: *Yahoo! Answers*, *Amazon Askvile*, entre outros. Nestes fóruns é comum encontrar perguntas sobre o cotidiano, no estilo “como fazer tal coisa”, “como consertar tal coisa” e, até mesmo, perguntas técnicas sobre uma determinada linguagem de programação, como exibido na Figura 3.

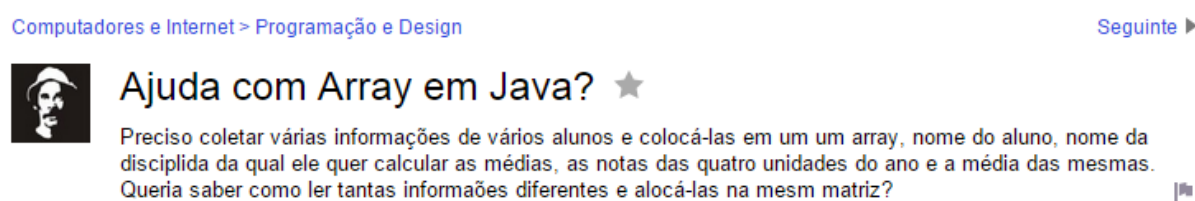


Figura 3 – Exemplo de uma Pergunta Específica em Fóruns Gerais.

Fonte: <https://br.answers.yahoo.com/question/index?qid=20110715213527AAKgbIc>

Por outro lado, os fóruns técnicos (fóruns para desenvolvedores) reduzem seu escopo para perguntas direcionadas exclusivamente a programação. São exemplos destes fóruns:

*Apple Developer Forum*⁹ e *Stack Overflow*. A diferença mais notável em fóruns para programadores são as características das perguntas e respostas, pois estas, quase em sua totalidade, possuem blocos de código para melhor exemplificar o problema, como demonstrado na Figura 4.

Exemplo de utilização:

```
$arquivos = glob("*."); // Captura todos os arquivos do diretório atual
$extensoes = array('gif', 'png', 'jpg'); // Extensões de imagens que você quer zipar

$resultado = create_zip($arquivos, $extensoes);
if ($resultado)
    echo "O arquivo ZIP foi criado com sucesso! \n";
else
    echo "Não foi possível criar o arquivo ZIP. \n";
```

Figura 4 – Exemplo de uma Resposta que Contém Bloco de Código

Fonte: <http://stackoverflow.com/questions/5216533/>

2.2. O Fórum *Stack Overflow*

O fórum *Stack Overflow*, é um dos maiores e mais populares fóruns de perguntas e respostas para desenvolvedores (PONZANELLI; MOCCI; BACCHELLI, 2014) e vem sendo amplamente utilizado por usuários que necessitam solucionar uma dúvida sobre determinada linguagem de programação ou sobre algum aspecto computacional. Este é um facilitador, pois normalmente desenvolvedores estão sujeitos a manuais e/ou tutoriais sobre determinado produto o que acaba sendo improdutivo dado o tempo gasto para encontrar uma resposta neste tipo de material. O fórum *Stack Overflow* possui sua versão principal, em Inglês, e outras três derivações, Português, Espanhol e Russo, os quais foram criados para usuários que não possuem domínio da língua inglesa. Estes fóruns fazem parte de uma rede maior, a *Stack Exchange*, que possui cerca de 150 *websites* de perguntas e respostas.

2.2.1. Estrutura e Conceitos

Assim como a grande maioria dos fóruns, o *Stack Overflow* possui uma organização simples. Um tópico (postagem) possui apenas uma única pergunta e sua respectiva lista de

⁹ Mais informações em: https://discussions.apple.com/community/developer_forums

respostas. As melhores respostas são votadas como positivo e, conseqüentemente, aparecem no topo, sendo assim, fáceis de serem encontradas. O usuário que realizou a pergunta tem o direito de escolher uma das respostas como “aceita”, isso não quer dizer que esta é exatamente a melhor resposta para a pergunta, mas é um indicativo para tal, uma vez que é a resposta que ajudou o questionador a solucionar sua dúvida¹⁰. É importante destacar que uma vez selecionada, a resposta aceita não pode mais ser alterada.

Todas as perguntas no *Stack Overflow* são etiquetadas para facilitar eventuais buscas por questões interessantes (similar as categorias existentes nos fóruns gerais). Essas etiquetas (*tags*) são inseridas pelo usuário no momento da criação da pergunta e são relacionadas ao seu conteúdo. A Figura 5 exibe um exemplo de *tags* (retângulo vermelho) para uma pergunta sobre PHP (*Hypertext Preprocessor*). Ao clicar em uma dessas *tags* o usuário visualiza uma lista de perguntas que possuem a mesma *tag*.

Qual a diferença entre else e elseif?

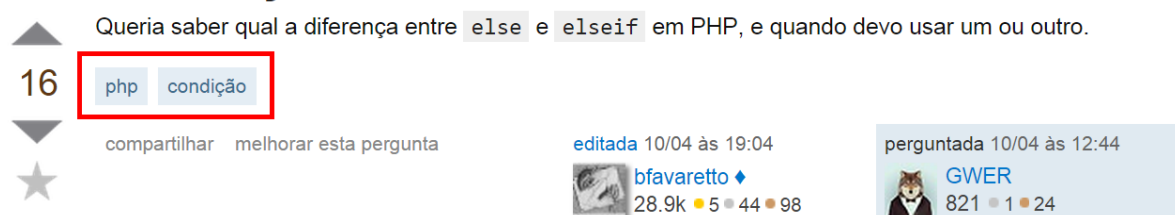


Figura 5 – Exemplo de *tags* no fórum *Stack Overflow*

Fonte: <http://stackoverflow.com/questions/24001778>

Visando evitar perguntas não relacionadas com o fórum¹¹, bem como perguntas que já foram respondidas, o *Stack Overflow* possui um código de conduta sobre o que perguntar e o que não perguntar, demonstrado no Quadro 1.

Quadro 1 – O que Perguntar e o que Não Perguntar no *Stack Overflow*

Perguntar sobre...	Não Perguntar Sobre...
Problemas específicos de Programação.	Dúvidas já solucionadas em outras perguntas.
Algoritmos de Software.	Recomendação de serviços e comparações.
Técnicas de Codificação.	Solicitação de listas de coisas, opiniões e discussões.
Ferramentas de Desenvolvimento.	Nada não relacionado diretamente com programação.

¹⁰ Mais informações em: <http://pt.stackoverflow.com/help/accepted-answer>

¹¹ Mais informações em: <http://pt.stackoverflow.com/help/on-topic>

O fórum também lista algumas recomendações para fornecer novas respostas. Dentre elas destacam-se a que o usuário deve ser respeitoso e ir direto ao ponto, pois o objetivo do fórum é obter respostas e não gerar discussões.

2.2.2. Sistema de Reputação

Sistemas de reputação fazem parte da maioria dos fóruns de perguntas e respostas e tem como objetivo engajar os usuários. A Figura 6 exibe o modelo de reputação social utilizado no *Stack Overflow*, onde o usuário “John Doe” possui 245 pontos de reputação. Ao longo do tempo o usuário pode aumentar a sua reputação de acordo com seu comportamento. Por exemplo, a cada resposta inserida no fórum que outro usuário considerar positiva (votada a favor), lhe garantirá um acréscimo de 10 pontos em sua reputação. Além disso, a reputação no *Stack Overflow* é composta por medalhas divididas três categorias: ouro, prata e bronze. No caso do usuário “John Doe”, três de ouro, 22 de prata e 74 de bronze.

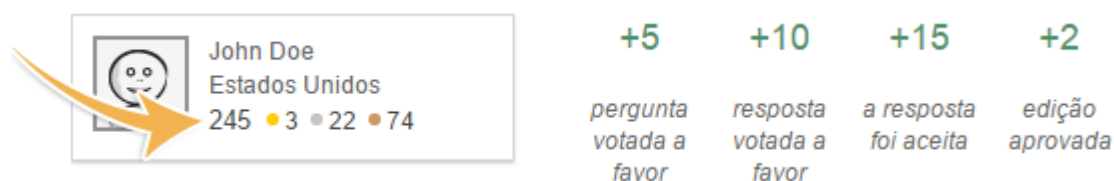


Figura 6 – Sistema de Reputação Social no *Stack Overflow*

Fonte: <http://pt.stackoverflow.com/tour>

O Quadro 2 exibe algumas ações que podem ser realizadas para a obtenção das respectivas medalhas. O fórum conta com mais de 50 diferentes medalhas que podem ser obtidas.

Quadro 2 – Ações Necessárias para Obter Medalhas no Fórum *Stack Overflow*

Fonte: <http://pt.stackoverflow.com/help/badges>

Categoria	Medalha	Ação Necessária
Ouro	Questão Famosa	Realizar uma pergunta com mais de 10 mil visualizações.
Ouro	Legendário	Ganhou 200 de reputação em um dia 150 vezes.
Prata	Boa Resposta	Responder uma pergunta e obter mais de 25 votos positivos.
Prata	Questão Notável	Realizar uma pergunta com mais de 2.5 mil visualizações.
Prata	Esclarecido	Primeiro a responder e aceito com uma pontuação de 10 ou superior.
Bronze	Estudante	Responder a primeira pergunta e obter uma pontuação maior que 1.

Dependendo do valor da reputação que o usuário possui, alguns privilégios são a ele disponibilizados, como por exemplo: avaliar perguntas e respostas como úteis (positivas) e não úteis (negativas), escrever comentários, entre outros. Nos níveis mais altos o usuário tem acesso a ferramentas de moderação e se torna um colaborador para o site se manter consistente e útil. A reputação, somada ao status do usuário são umas das motivações extrínsecas para o usuário responder questões em fóruns de perguntas e respostas (RABAN; HARPER, 2008).

O acúmulo de dados disponíveis nestes fóruns pode não ser uma fonte interessante de informação e conhecimento apenas para desenvolvedores, mas também para mecanismos automáticos de Recuperação da Informação e Mineração de Dados. Nesta pesquisa propomos o uso deste tipo de tecnologia para explorar os dados disponíveis em fóruns de perguntas e respostas.

2.3. Mineração de Dados

A Mineração de Dados consiste em extrair conhecimento novo, útil, surpreendente e compreensível ao ser humano de bases de dados (YANG; ZHENG; YAN, 2012). Esta tarefa é considerada, na maioria das vezes, impossível (considerando o tempo necessário) de ser realizada manualmente, pois geralmente as bases de dados utilizadas são significativamente grandes. A Mineração de Dados é parte de um processo maior conhecido como *Knowledge Discovery in Databases* (KDD).

Existem algumas técnicas disponíveis para a descoberta de conhecimento/padrões. Dentre elas, destacam-se: regras de associação, agrupamento e classificação. As regras de associação são regras do tipo SE (X) \rightarrow (Y), onde se lê se X então Y, em que X e Y são conjuntos de informação não vazios. Por exemplo, “SE leite E pão ENTÃO manteiga”. A técnica de agrupamento consiste em agrupar um conjunto de instâncias de dados pela sua similaridade. Por fim, a técnica de classificação busca identificar à qual classe uma determinada instância não rotulada pertence. Para tal um modelo é construído (treinado) utilizando o conjunto de entrada, onde cada instância já possui a identificação da classe à qual pertence. Em seguida, o modelo treinado é capaz de rotular (predizer) o valor da classe de uma nova instância. Um bom exemplo prático é o de um banco. O modelo pode ser construído valendo-se da base de dados clientes atuais e suas características perante ser ou

não um devedor. Quando um novo cliente ingressar ao banco, há a possibilidade de classificá-lo como possível devedor e então não ceder crédito a ele.

Existem duas etapas iniciais que podem definir o sucesso ou fracasso do processo, são elas: obtenção da base de dados (2.3.1) e pré-processamento (2.3.2). Em seguida os modelos são construídos e testados utilizando os algoritmos de classificação (2.3.4). A etapa de pós-processamento (2.3.5) é muitas vezes necessária para tornar os padrões extraídos legíveis. Por fim, os resultados são submetidos a um processo de avaliação (2.3.6).

2.3.1. Bases de dados

A matéria prima indispensável para qualquer técnica de Mineração de Dados é a uma base de dados sobre o tema a ser explorado. Ao contrário do que vemos normalmente na modelagem de bases de dados de softwares, isto é, bases de dados normalizadas (consistente e sem redundância), para o processo de Mineração de Dados é necessário que a base de dados esteja na forma não normalizada, em um único arquivo, onde uma linha (instância) representa todas as características de determinado elemento e/ou acontecimento. Considerando a tarefa de classificação, além das características, um atributo chamado “classe” é necessário. Este conjunto de atributos é conhecido como vetor de características. A expressão abaixo apresenta o formato de um vetor de características.

$$x = [x_1 \ x_2 \ x_3 \dots x_i]$$

Equação 1 – Vetor de Características

Onde, x representa o vetor de características, $[x_1 \ x_2 \ x_3 \dots]$ os atributos e x_i o atributo classe.

2.3.2. Pré-processamento

O Pré-processamento é uma etapa em geral trabalhosa em função dos dados disponíveis não estarem de tal forma organizados a permitir a aplicação direta aos algoritmos de Mineração de Dados. Esta etapa é responsável por diversos procedimentos que podem aumentar significativamente a qualidade obtida no final do processo. Alguns métodos disponíveis no pré-processamento são:

Seleção de Atributos: às vezes nem todos os atributos disponíveis na base de dados são úteis para o processo de Mineração de Dados, nestes casos, existem algoritmos que eliminam os atributos que tendem a diminuir a eficiência do processo;

Limpeza: em bases de dados é comum encontrar valores ausentes (nulos) devido a um erro de leitura e/ou preenchimento, por exemplo. Esses valores podem comprometer o processo dependendo da forma como são utilizados. Normalmente instâncias com valores ausentes são eliminadas;

Transformação: é comum encontrar algoritmos de classificação que só aceitam atributos numéricos. Neste caso é necessário por meio de algum método, alterar os valores nominais para números. Às vezes o inverso também é necessário, ou seja, discretizar dados numéricos.

2.3.3. Algoritmos de Classificação

Os algoritmos de classificação podem ser organizados em: algoritmos baseados em instância, algoritmos baseados em otimização, algoritmos bayesianos e algoritmos de árvores de decisão (MITCHELL, 1997).

Os algoritmos baseados em instância são os algoritmos mais simples da tarefa de classificação. Dentre eles, destaca-se o kNN (*k-Nearest Neighbor*), o algoritmo do vizinho mais próximo. Sua classificação se baseia em analisar os exemplos de treinamento e identificar os vizinhos mais próximos no espaço de características. O único problema deste algoritmo é definir a função responsável por calcular a distância entre os pontos, particularmente se os atributos não forem todos numéricos (WITTEN; FRANK; HALL, 2011). Com esta função definida, para calcular um exemplo desconhecido é necessário: calcular a distância do exemplo desconhecido para os demais pontos da base de treinamento; identificar os k (valor definido pelo usuário) vizinhos mais próximos; realizar uma votação majoritária em relação ao resultado dos rótulos dos vizinhos mais próximos; rotular o exemplo desconhecido com o maior resultado obtido. Outro algoritmo bastante utilizado nesta área é o SVM (*Support Vector Machine*) que busca resolver problemas de classificação e regressão. Em problemas onde as classes são linearmente separáveis o SVM “aprende” hiperplanos de separação com margem máxima. A margem máxima é necessária para minimizar o erro de generalização. A partir disto os vetores de suporte são construídos

utilizando os exemplos de treinamento mais próximos da função de separação. Em seguida, através dos vetores de suporte, as novas instâncias podem ser classificadas. Já para casos não linearmente separáveis o SVM expande a entrada para um espaço de alta dimensionalidade. Para tal, são necessárias funções *kernels* (polinomial, gaussiano etc.) capazes de separar o plano de forma não linear.

Algoritmos baseados em otimização podem ser implementados como Redes Neurais Artificiais (RNAs). Essas redes são inspiradas no cérebro humano e em sua arquitetura paralela. Obviamente não há como comparar a eficácia de um cérebro com uma rede neural, uma vez que o cérebro humano é bom em reconhecer padrões e associar, ao contrário do computador que é eficiente em cálculos e lógica. As formas mais básicas de aprendizado em Rede Neurais são: *Perceptron* e o *MLP (Multi Layer Perceptron)*. O *Perceptron* é um algoritmo para aprendizagem de redes neurais simples, pois utiliza uma única camada. Já o *MLP* (desenvolvido nos anos 80) é mais complexo, pois considera o uso de múltiplas camadas. O objetivo principal do aprendizado em Redes Neurais Artificiais é a obtenção de modelos com boa capacidade de generalização tendo como base o conjunto de dados (REZENDE, 2003).

O modelo bayesiano pode ser representado pelo algoritmo *Naive Bayes*. Este algoritmo é um método probabilístico que pressupõe que todos os atributos são independentes, isto é, o valor de um atributo não influencia no valor de outro atributo. Para classificar um exemplo desconhecido, basta calcular a probabilidade dos seus elementos divididos em partes em relação à classe. Posteriormente, a classe com maior probabilidade preencherá o rótulo deste exemplo desconhecido.

Os algoritmos de árvores de decisão são os mais conhecidos e utilizados na tarefa de classificação. Para classificar um exemplo desconhecido a *Árvore de Decisão* utiliza como entrada um conjunto de atributos e a sua saída é a classe para rotular o novo exemplo. A Figura 7 exibe a *Árvore de Decisão* da base de dados “*wheater.nominal*” extraída do repositório público de bases de dados *UCI Machine Learning Repository*. Esta base tem como objetivo prever, de acordo com as condições climáticas, se um jogador irá ou não praticar golf.

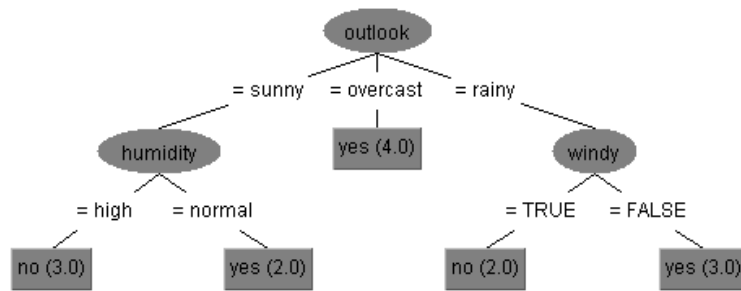


Figura 7 – Exemplo de Árvore de Decisão.

Fonte: (HALL; FRANK; HOLMES, 2009)

Cada nó da árvore de decisão corresponde a um atributo. Os ramos são rotulados com os resultados possíveis do teste. Por fim, os nós folhas contém o valor da classe. Para determinar o valor da classe de um elemento desconhecido basta percorrer a árvore fazendo os devidos testes. Ao chegar ao nó folha, o valor dele corresponderá ao novo rótulo para o atributo desconhecido. Este é um modelo bastante interessante, pois, ao contrário das Redes Neurais Artificiais, possui uma representação natural para o ser humano. Dentre os algoritmos de árvores de decisão, destaca-se o C4.5 desenvolvido por (QUINLAN, 1993). Este algoritmo possui uma versão disponível no ambiente de Mineração de Dados WEKA (HALL; FRANK; HOLMES, 2009) desenvolvida em Java chamada J48.

Existem alguns métodos para a avaliação de algoritmos de classificação, dentre eles destacam-se: validação cruzada (*k-Fold*) e *Holdout*. Estes métodos são necessários para avaliar o classificador perante a sua capacidade de generalização. No método *k-Fold* (onde o valor de *k* é definido pelo usuário), a base de dados é dividida *k* vezes em subconjuntos mutuamente exclusivos, na sequência utiliza-se um dos subconjuntos para a validação do modelo e os demais para o treinamento. A Figura 8 exhibe o funcionamento deste método.

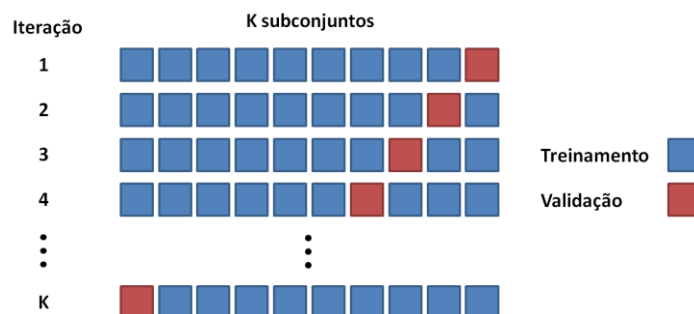


Figura 8 – Método *k-Fold*.

Fonte: (MITCHELL, 1997)

Por outro lado, o método *Holdout* consiste, normalmente, em separar 2/3 do conjunto de instâncias da base dados para o treinamento e 1/3 para a avaliação. Esta separação ocorre de maneira aleatória.

2.3.4. Pós-processamento

O pós-processamento visa facilitar a avaliação dos padrões encontrados durante o processo de Mineração de Dados. Algumas medidas vêm sendo estudadas buscando auxiliar o usuário a analisar o conhecimento extraído. Elas são divididas em dois grupos: medidas de desempenho e medidas de qualidade.

As medidas de desempenho serão discutidas na seção seguinte (2.3.6) por se tratarem de medidas que avaliam o desempenho dos resultados.

Medidas de qualidade podem ser divididas em dois grandes campos: compreensibilidade e interessabilidade. A compreensibilidade está relacionada a capacidade do especialista em avaliar o conhecimento. Considerando a tarefa de classificação a compreensibilidade se torna inviável para modelos “caixa-preta”, pois não existe um modelo de saída natural ao ser humano. Diferentemente, o modelo de Árvore de Decisão pode ser transformado em regras de classificação. A transcrição da árvore de decisão visa facilitar a compreensibilidade dos padrões extraídos. O Quadro 3 lista as regras de classificação extraídas da Árvore de Decisão da Figura 7.

Quadro 3 – Regras de Classificação Extraídas da Árvore de Decisão

Número	Regra de Classificação
1	SE outlook=sunny E humidity=high ENTÃO play=no
2	SE outlook=sunny E humidity=normal ENTÃO play=yes
3	SE outlook=overcast ENTÃO play=yes
4	SE outlook=rainy E windy=yes ENTÃO play=no
5	SE outlook=rainy E windy=no ENTÃO play=yes

A interessabilidade está relacionada ao quão interessante é o conhecimento apresentado ao especialista, valendo-se das premissas da Mineração de Dados: descobrir conhecimento novo, útil e surpreendente. Para tal, considerando um conjunto de regras, existem medidas como suporte (proporção de transações na base de dados que possuem o antecedente e o consequente) e confiança (suporte dividido pela proporção de transações que

possuem o consequente). Porém, como a classificação é uma tarefa que visa prever o valor da classe de novas instâncias, outras medidas como a de generalizações sucessivas existem para melhor mensurar o real grau de interesse para cada regra de classificação.

Se mesmo após a análise do conhecimento os padrões extraídos ainda não sejam interessantes e/ou satisfatórios, o processo pode ser repetido ajustando-se os parâmetros ou melhorando o processo de escolha dos dados visando melhores resultados na próxima iteração (REZENDE, 2003).

2.3.5. Avaliação dos Resultados

A medida mais comum para a avaliação de classificadores é a taxa de acerto (acurácia), isto é, quantidade de acertos dividida pela quantidade de instâncias classificadas. Além disso, a avaliação também está relacionada à interpretabilidade do modelo, como citado anteriormente. Apesar da taxa de acerto cobrir a maioria das necessidades, existem algumas medidas alternativas para calcular o desempenho do classificador para cada classe. O Quadro 4 representa uma matriz de confusão para um problema de classificação binário, onde TP, TN, FP e FN representam os resultados verdadeiro/positivos, verdadeiro/negativos, falso/positivos e falso/negativos, respectivamente. A classificação binária envolve apenas duas classes (perguntas com boa qualidade ou não, por exemplo), e a classificação multiclasse envolve três ou mais classes.

Quadro 4 – Matriz de Confusão para Duas Classes

		Classes Reais	
		Positiva	Negativa
Classes Preditas	Positiva	TP	FN
	Negativa	FP	TN

As medidas alternativas podem ser extraídas da matriz de confusão, são elas:

Cobertura: $cov(h) = \frac{TN}{TN + FN}$

Especificidade: $spec(h) = \frac{TN}{TN + FP}$

Precisão: $prec(h) = \frac{TP}{TP + FP}$

Sensibilidade: $sens(h) = \frac{TP}{TP + FN}$

Capítulo 3

Trabalhos Relacionados

Neste capítulo são apresentados trabalhos que se relacionam com o escopo desta pesquisa.

Para uma melhor apresentação, os trabalhos estão agrupados por temas. Os temas são: Predição da Qualidade de Respostas (3.1) e A Utilização do Fórum *Stack Overflow* Durante o Desenvolvimento (3.2). Por fim, com o intuito de auxiliar na visualização, na seção 3.3 é apresentada uma conclusão contemplando uma visão geral dos trabalhos relacionados.

Os trabalhos foram encontrados nas seguintes bases de dados: ACM, *CiteSeerX*, IEEE, *ScienceDirect* e *SpringerLink*. O Quadro 5 lista as bases de dados, bem como os termos utilizados.

Quadro 5 – Bases Consultadas e Termos da Busca

Bases	Termos da busca
ACM - http://dl.acm.org/ CiteSeerX - http://citeseerx.ist.psu.edu/index IEEE - http://ieeexplore.ieee.org/Xplore/home.jsp ScienceDirect - http://www.sciencedirect.com/ SpringerLink - http://link.springer.com/	<i>"Answer Quality"</i> <i>"Question and Answer"</i> <i>"Predicting Quality of Answer"</i> <i>"Collaborative Question and Answer"</i>

3.1. Predição da Qualidade de Respostas

Na literatura, alguns trabalhos se preocuparam em analisar a qualidade de respostas em fóruns de perguntas e respostas. Ainda que alguns esforços tenham sido realizados

recentemente, este ainda é um tópico com baixa atenção dos pesquisadores (HART; SARMA, 2014).

Os três principais meios utilizados para a predição de qualidade de respostas são: classificação, regressão e modelos probabilísticos. Esta seção se propõe a analisar essas três abordagens.

Considerando a tarefa de classificação, os pesquisadores vêm avaliando a qualidade de respostas por meio da resposta escolhida, dentro do conjunto de respostas, como a que ajudou o usuário que realizou a pergunta. Esta resposta é etiquetada como uma boa resposta e as demais são consideradas respostas que não agradaram e/ou não ajudaram o usuário. Outra maneira adotada para prever a qualidade da resposta é por meio de anotadores que etiquetam previamente as respostas como boas ou ruins.

Desta forma, os pesquisadores extraem diversas características do texto e dos usuários e submetem a classificadores. A taxa de acerto (acurácia) é utilizada para determinar a qualidade do modelo construído.

O trabalho de (HANDAYANI, 2013) utilizou o fórum de perguntas e repostas *Yahoo! Answer* para pessoas da Indonésia¹². Os pesquisadores extraíram dados das categorias que possuíam maior índice de atividade (maior número de questões resolvidas), chegando as seguintes categorias: música e entretenimento, cultura e sociedade, Internet e computador, família e relacionamentos e, por fim, consumidores e eletrônicos. Entretanto, os autores optaram em utilizar somente perguntas da categoria “Internet e computação” por se tratarem de respostas mais técnicas e que normalmente independem da opinião religiosa e espiritual das pessoas. Os autores extraíram 18 características quantitativas, 9 relacionadas à resposta (AF) e 9 relacionadas ao histórico do usuário que respondeu (AUH). Estas características foram extraídas apenas da resposta escolhida pelo usuário. As demais respostas foram ignoradas, o que tende a prejudicar a acurácia do classificador uma vez que estas podem conter características pela qual o usuário não a escolheu como a melhor e que poderiam ter sido consideradas. Ao realizar um teste de correlação entre as características, os autores descobriram que os atributos: quantidade de palavras e o quantidade de caracteres possuíam a maior correlação com a qualidade da resposta, ou seja, quanto mais detalhada a resposta, melhor a sua qualidade. Por fim, os autores submeteram a base de dados a três classificadores,

¹² Mais informações em: <http://id.answers.yahoo.com/>

considerando três conjuntos diferentes: todas as características (*All*), somente características da resposta (*AF*) e, por fim, somente características do usuário (*AUH*). Os resultados estão demonstrados na Tabela 2.

Tabela 2 – Acurácia do Treinamento para cada Conjunto de Características

Fonte: (HANDAYANI, 2013)

Classificador	All	AF	AUH
<i>Naive Bayes</i>	73,13%	79,14%	50,80%
<i>Adaboost</i>	81,10%	81,10%	53,00%
C4.5	91,90%	88,83%	66,50%

Considerando as acurácias obtidas é possível perceber que as características do usuário não contribuem muito para determinar a qualidade da resposta e que o classificador C4.5 obteve a melhor taxa de acerto.

Em outro trabalho (ARAI, 2014) utilizou a mesma fonte de dados (categoria Internet e computação do *Yahoo! Answers*) para testar novos conjuntos de características. Primeiramente, os autores construíram anotadores para mapear as perguntas de acordo com a sua qualidade: boa, médias e ruins. Esses anotadores se baseiam em algumas premissas: as respostas têm que ser informativas (contém informação suficiente para responder a questão), educadas (a resposta não ofende quem perguntou), completas (responder a questão inteira, não somente uma parte), legível (a resposta possui um bom vocabulário), relevante (a informação da resposta condiz com a pergunta), detalhada (contém detalhes suficiente pra cobrir toda a pergunta) e útil (ajudar o usuário a solucionar a sua dúvida). Essas mesmas características foram sugeridas por (ZHU; BERNHARD; GUREVYCH, 2009) para a construção de um modelo dinâmico para avaliar a qualidade de respostas em fóruns de perguntas e respostas. O uso de anotadores é interessante em termos práticos, pois as respostas são avaliadas como boas, médias ou ruins automaticamente. Porém, esta prática pode manipular os dados para que o processo de classificação se comporte de maneira mais eficaz. O modelo “ideal” para anotadores seriam pessoas reais avaliando as perguntas de acordo com a sua qualidade. Porém, este método é bastante custoso e demorado dado o tempo gasto por um humano para analisar uma única pergunta e a proporção de dados existente. Após o uso de anotadores automáticos, o autor criou duas categorias de características: textuais (quantitativas) e sobre a legibilidade. As características textuais são: quantidade de caracteres, quantidade de palavras e quantidade de sentenças na resposta. Já as características

sobre a legibilidade são oriundas de um método para estimar o nível escolar, nos Estados Unidos, em relação a capacidade de compreender um texto. Usando o coeficiente de relação entre as características, descobriu-se que a maioria das características sobre legibilidade tende a ajudar a construir uma boa resposta. A que obteve o maior índice foi a existência de pronomes abstratos (ideias, ciência, sonhos etc.). Por fim, o autor submeteu esses conjuntos de características ao classificador C4.5 e obteve os resultados da Tabela 3.

Tabela 3 – Acurácia do Treinamento Utilizando o Algoritmo C4.5

Fonte: (ARAI, 2014)

Características	Base de Treino	Base de Teste
Todas	93,6%	72,8%
Textuais	90,4%	70,4%
Legibilidade	93,2%	73,6%

O trabalho de (SHAH; ST; BRUNSWICK, 2010) não faz o uso de anotadores: os autores partem do princípio que a melhor resposta é a resposta aceita pelo usuário que realizou a pergunta, se e somente se, a resposta possuir no mínimo a avaliação três de cinco (60%) de aceite dos demais usuários. Se o usuário não escolher uma das respostas, os autores assumem que quem realizou a pergunta está insatisfeito com as respostas. Os autores selecionaram 120 perguntas aleatórias do fórum *Yahoo! Answers* e suas respectivas respostas (limitadas em 5). Foram extraídas duas listas de características tanto da pergunta como das repostas, para posteriormente submeter a um algoritmo de regressão. A primeira lista de características (resposta detalhada, educada etc.) foi avaliada por humanos. A segunda lista de características foi construída automaticamente utilizando características quantitativas (número de respostas na questão, quantidade de caracteres na resposta etc.). Dentre os testes realizados, a maior taxa de acerto foi de 80,33% para as características avaliadas por humanos e 84,17% para as características quantitativas extraídas automaticamente. Segundo os autores, as características que mais influenciaram na qualidade da resposta foram as características do usuário que respondeu ao questionamento e a ordem em que essa resposta aparecia na lista de repostas disponível. Essas características que determinam se a resposta possui boa qualidade ou não diverge dos trabalhos anteriormente apresentados, os quais determinavam que essas características acrescentavam pouco ou nenhum ganho ao modelo.

O trabalho de (JEON; CROFT; LEE, 2006) construiu um framework para prever a qualidade de repostas utilizando características não textuais, pois, segundo os autores, estas

possuem um bom potencial para aumentar a qualidade da resposta. O método proposto utilizou os métodos probabilísticos de entropia máxima e um *kernel* de estimação de densidade como forma de avaliação dos resultados. Foram extraídas 1.700 pares de perguntas e respostas do fórum *Naver* para esta avaliação. As classes utilizadas neste processo foram: respostas com alta qualidade, média e baixa, analisadas por um anotador. Neste cenário os autores chegaram a um resultado de 92,27% de acerto para a classe “respostas com alta qualidade”. As principais características utilizadas foram: tamanho da resposta e quantidade de estrelas (1 a 5) que o usuário forneceu para a resposta escolhida. Utilizando a mesma base de dados, porém considerando a classe “respostas de média qualidade” como respostas de qualidade, o trabalho de (LEE; SONG; RIM, 2007) divergiu em relação às características consideradas importantes por (JEON; CROFT; LEE, 2006), como por exemplo, a quantidade de caracteres na resposta.

Todos os trabalhos supracitados foram construídos sobre a perspectiva de fóruns de perguntas e respostas gerais. Na literatura, apenas um trabalho (CALEFATO; LANUBILE; MARASCIULO, 2015) se propôs a prever a qualidade de respostas em fóruns de perguntas e respostas para desenvolvedores. Os autores utilizaram como fonte de dados o fórum *Stack Overflow* em sua versão em Inglês. Assim como os demais autores o conceito de resposta de qualidade foi atribuído a resposta selecionada pelo questionador. Foram extraídas características referentes a apresentação de qualidade como número de URLs na resposta e presença de blocos de código. Além disso, características referentes a percepção de qualidade como quantidade de votos positivo e negativos atribuídos a resposta também foram utilizados. Em seguida os autores construíram uma base com mais de 300 mil respostas e a submeteram a um classificador. O melhor resultado obtido foi de 64,98% para a métrica da área em baixo da curva ROC.

3.2. A Utilização do Fórum *Stack Overflow* Durante o Desenvolvimento

Esta seção aborda outras frentes de pesquisa que estão sendo aplicadas sobre o fórum *Stack Overflow*. Durante o desenvolvimento de software é comum os desenvolvedores buscarem na Internet recursos para a resolução de suas dúvidas. Cedo ou tarde, por exemplo, para corrigir um erro no sistema, os desenvolvedores tem que lidar com o desconhecido (PONZANELLI; BAVOTA; PENTA, 2014). O grande problema nesse cenário é que os desenvolvedores têm que interromper o seu fluxo de trabalho para buscar a informação. Para

solucionar esta questão, Ponzanelli propôs uma ferramenta que, automaticamente, recupera e recomenda por meio de pequenas notificações, discussões relevantes do fórum *Stack Overflow*. Esta ferramenta, chamada *Prompter*, é um *plugin* disponível na IDE (*Integrated Development Environment*) Eclipse, como mostra a Figura 9.

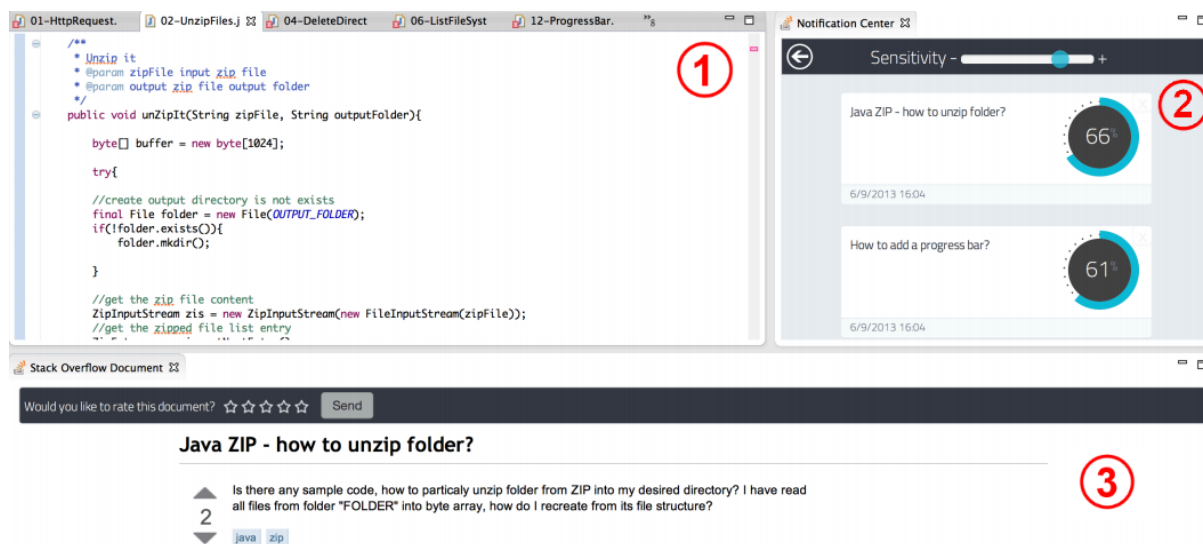


Figura 9 – *Prompter* - Interface do Usuário

Fonte: (PONZANELLI; BAVOTA; PENTA, 2014)

Área de Desenvolvimento (1): área comum em todas as IDEs, onde o desenvolvedor escreve o código de seu software;

Área de Notificações (2): de acordo com o código que está sendo desenvolvido, o *plugin*, em tempo real, recupera discussões relevantes para o contexto atual. Na Figura 9, o desenvolvedor estava interessado em extrair o conteúdo de um arquivo comprimido. Na central de notificações uma discussão sobre o tema é exibida. Os autores utilizaram aspectos do código (códigos clonados e correspondência por tipo), aspectos conceituais (similaridade do texto) e aspectos do fórum *Stack Overflow* (reputação do usuário) para fornecer aos desenvolvedores discussões que condizem com o com o seu código atual;

Área do Conteúdo do Fórum (3): ao clicar na discussão na central de notificações, o desenvolvedor tem acesso ao conteúdo em uma caixa separada e sem precisar sair do IDE, isto facilita enormemente, pois em cenários normais o desenvolvedor precisa sair do IDE e navegar por conta própria nos tópicos do *Stack Overflow*;

Em outro contexto, mas também buscando auxiliar o desenvolvimento de software, alguns pesquisadores trabalharam sobre a documentação de código fonte. É muito comum nesta área nos depararmos com códigos fonte sem comentários e isto geralmente causa um retrabalho para um desenvolvedor que está fazendo manutenção. Um código fonte bem documentado é fácil de ler e manter, pensando em longo prazo, mas esta é uma tarefa que muitas vezes é ignorada por programadores (MOVSHOVITZ-ATTIAS; COHEN, 2013). Os autores (WONG; YANG; TAN, 2013) criaram um método para inserir comentários automaticamente no código através da mineração do fórum *Stack Overflow*. Primeiramente, foram selecionados 23 projetos escritos em linguagem JAVA para o experimento. O código é comparado com uma base de dados do *Stack Overflow* que contém todas as postagens. Em seguida uma lista de segmentos de código e seus comentários correspondentes são gerados. A partir deste ponto, os autores utilizaram o Processamento de Linguagem Natural (PLN) para extrair comentários úteis, concisos e adequados, como o exibido no Quadro 6.

Quadro 6 – Comentário Extraído pela Ferramenta “*AutoComment*”

Código	<pre>Public String getToolTipText (MouseEvent e) { java.awt.Point p = e.getPoint(e); int rowIndex = rowAtPoint(p); int colIndex = columnAtPoint(p); }</pre>
<i>Stack Overflow</i>	<p>“The problem is that you set tooltips on subcomponents of the component returned by your CellRenderer. To perform what you want, you should consider override <code>getToolTipText(MouseEvent e)</code> on the <code>JTable</code>. From the event, you can find on which row and column the mouse is, using:</p> <pre>java.awt.Point p = e.getPoint(e); int rowIndex = rowAtPoint(p); int colIndex = columnAtPoint(p);”</pre>
Comentário Gerado	Find on which row and column the mouse is.

Alguns pesquisadores começaram a se preocupar com a quantidade de perguntas criadas no fórum *Stack Overflow*. De acordo com o próprio site, mais de 6 mil perguntas são criadas por dia, de segunda a sexta-feira. Com isso, surge a demanda para a criação de um processo automático para identificar tópicos (perguntas) que não são válidos. Basicamente, perguntas podem ser encerradas pelos seguintes motivos: por estar fora do tópico, por não ser exatamente uma pergunta, por não ser construtiva ou por ser duplicada. Atualmente, este processo é realizado pelos próprios usuários “colaboradores”. Neste cenário, o trabalho de (LEZINA; KUZNETSOV, 2013) elaborou um método para prever quais tópicos devem ser

fechados de forma automática. Assim como no processo de predição da qualidade de respostas, os autores extraíram características para esta tarefa. Essas características foram separadas em 3 grupos: características do usuário (reputação, perfil preenchido, votos positivos etc.), características do tópico (número de blocos de código, número de links, número de sentenças etc.) e características das etiquetas (quantidade de tópicos fechados por etiqueta). Através da análise do resultado do método *Multiclass Logarithmic Loss* os autores concluíram que as características do tópico (textuais) são mais eficazes para predizer quais são os tópicos que devem ser fechados.

3.3. Conclusão

Fóruns de perguntas e respostas oferecem aos pesquisadores diversas linhas de pesquisa promissoras. Considerando a predição da qualidade de respostas em fóruns de perguntas e respostas alguns métodos foram desenvolvidos. Entretanto, a grande maioria dos pesquisadores se concentrou em analisar fóruns de perguntas e repostas gerais. Alguns autores se preocuparam em filtrar um conjunto de perguntas específicas sobre a área de computação por serem respostas mais técnicas. Apesar disso, muitas dessas perguntas não estão relacionadas diretamente a dúvidas sobre linguagens de programação, foco deste trabalho. Se tratando da predição da qualidade de respostas em fóruns de perguntas e respostas para desenvolvedores apenas um trabalho foi encontrado na literatura. Desta forma, é necessário analisar se o método apresentado pode ser melhorado.

Portanto, a construção de modelos que predizem a qualidade de resposta se tornam válidos para melhorar a qualidade dos dados disponíveis em fóruns de perguntas e respostas para desenvolvedores. A automatização de algumas tarefas (atualmente realizadas manualmente) facilitaria a manutenção da consistência do fórum e, conseqüentemente, a qualidade do fórum como um todo tende a aumentar. Além disso, este processo pode aprimorar ferramentas que já utilizam o fórum *Stack Overflow* como fonte de dados, uma vez que essas ferramentas exploram as perguntas e respostas do fórum sem considerar a qualidade das mesmas. Para facilitar a visualização o Quadro 7 exibe uma comparação entre os trabalhos de predição de qualidade de respostas exibidos nesta seção.

Quadro 7 – Comparação entre os Trabalhos de Predição de Qualidade de Respostas

Trabalho	Fórum	Lista de Características	Modelo Preditivo
(CALEFATO; LANUBILE; MARASCIULO, 2015)	<i>Stack Overflow</i>	<p>Número de URLs: quantidade de URLs na resposta;</p> <p>Presença de bloco de código: valor binário para a presença ou não de blocos de código na resposta.</p> <p>Tamanho da resposta: quantidade de caracteres na resposta;</p> <p>Taxa de caracteres em caixa alta: quantidade de caracteres em caixa alta dividido pelo montante final;</p> <p>Resposta com sentimento positivo: quantidade de respostas que apresentaram sentimento positivo;</p> <p>Resposta com sentimento negativo: quantidade de respostas que apresentaram sentimento negativo;</p> <p>Comentários positivos: quantidade de comentários que apresentaram sentimento positivo;</p> <p>Comentários negativos: quantidade de comentários que apresentaram sentimento negativo;</p> <p>Reputação do questionador: quantidade de reputação do questionador e de quem respondeu;</p> <p>Número de medalhas: quantidade de medalhas obtidas por quem respondeu a pergunta.</p>	Sim
(ARAI, 2014)	<i>Yahoo! Answer</i>	<p>Tamanho da resposta: quantidade de caracteres na resposta;</p> <p>Palavra: quantidade de palavras na resposta;</p> <p>Sentenças: quantidade de sentenças na resposta;</p>	Sim
(HANDAYANI, 2013)	<i>Yahoo! Answer</i>	<p>Estrelas: número de estrelas de 1 a 5;</p> <p>Referência: se a resposta tem alguma referência da pergunta;</p> <p>Contribuidor: quantidade categorias do usuário que respondeu;</p> <p>Votos positivos: quantidade de votos positivos;</p> <p>Votos negativos: quantidade de votos negativos;</p> <p>Caracteres: quantidade de caracteres na resposta;</p> <p>Palavra: quantidade de palavras na resposta;</p> <p>Sentenças: quantidade de sentenças na resposta;</p> <p>Membro: data que o usuário que respondeu ingressou no fórum;</p> <p>Level: nível do usuário que respondeu;</p> <p>Pontos: quantidade de pontos do usuário que respondeu;</p> <p>Lista de respostas: quantidade de respostas para a pergunta;</p> <p>Lista de melhores respostas: quantidade de melhores respostas;</p> <p>Melhores respostas: quantidade de melhores respostas respondidas pelo usuário;</p> <p>Demais respostas: quantidade de respostas respondidas pelo usuário não consideradas como a melhor;</p> <p>Melhores e demais respostas: quantidade de melhores respostas respondidas pelo usuário somado as demais respostas.</p>	Sim

(SHAH; ST; BRUNSWICK, 2010)	<i>Yahoo! Answer</i>	<p>Título da pergunta: tamanho do título em caracteres; Conteúdo da pergunta: tamanho do conteúdo em caracteres; Lista de Respostas: quantidade de repostas para a pergunta; Comentários: número de comentários para a pergunta; Resposta: tamanho da resposta em caracteres; Referência: se a resposta tem alguma referência da pergunta; Ranking da reposta: posição da resposta em relação às demais; Usuário que realizou a pergunta: nível, quantidade de pontos, quantidade de respostas, quantidade de “melhores respostas”, quantidade de perguntas, quantidade de questões resolvidas e quantidade de estrelas; Usuário que realizou a resposta: As mesmas características do usuário que realizou a pergunta;</p>	Não
(JEON; CROFT; LEE, 2006)	<i>Naver</i>	<p>Melhores respostas: quantidade de melhores respostas respondidas pelo usuário que perguntou; Caracteres: quantidade de caracteres na resposta; Especialização: nível do usuário para determinada categoria; Impressão: número de vezes que o usuário imprimiu as respostas; Cópias: número de vezes que os usuários exportam a reposta para blogs; Cliques: número de vezes que a pergunta foi consultada; Respostas: número de respostas; Recomendação do usuário: número de vez que o usuário recebeu alguma recomendação; Recomendação do editor: número de vez que o usuário recebeu alguma recomendação por edição; Votos negativos: quantidade de votos negativos;</p>	Não

Capítulo 4

Método

Neste capítulo é apresentado o método proposto nesta pesquisa para prever a qualidade de respostas em fóruns de perguntas e respostas para desenvolvedores. Na seção 4.1 são apresentados os pressupostos. Em seguida, na seção 4.2, é apresentada uma visão geral sobre o método proposto. Na seção 4.3 são apresentadas as características propostas para a construção do modelo preditivo. Em sequência, na seção 4.4, é apresentada a tarefa de Aprendizagem de Máquina (AM) selecionada para cumprir o objetivo proposto, a justificativa para o seu uso e a forma como será avaliado o modelo preditivo construído. Por fim, a seção 4.5 apresenta uma breve conclusão do capítulo.

4.1. Pressupostos

Primeiramente é importante destacar os pressupostos para a construção ou replicação do método posteriormente apresentado. O método é baseado na classificação de respostas oriundas de fóruns de perguntas e respostas para desenvolvedores, isto é, não é possível garantir sua efetividade em fóruns gerais. Além disso, algumas características mínimas para a construção do modelo preditivo devem existir no fórum selecionado. Na estrutura das perguntas e respostas, por exemplo, é fundamental que exista a possibilidade da extração de características como: quantidade de caracteres, presença de blocos de código, entre outros. Também é necessário acesso as características referentes aos usuários, com por exemplo: idade, reputação, entre outros. Por fim, mas não menos importante, o acesso aos dados históricos do fórum deve ser possível para que o modelo preditivo possa ser construído.

4.2. Visão Geral

A necessidade da construção de novas ferramentas que automatizem funcionalidades em fóruns de perguntas e respostas é um tema útil tendo em vista a quantidade de dados disponíveis nestes *websites*.

Neste trabalho de pesquisa é proposto um método capaz de classificar automaticamente a qualidade de respostas em fóruns para desenvolvedores através de um modelo construído utilizando conceitos de Aprendizagem de Máquina. A construção deste modelo (com possibilidade de ser integrado ao fórum) tende a melhorar a qualidade das respostas fornecidas pelos usuários e, conseqüentemente, a qualidade do conteúdo depositado nesses fóruns.

Além disso, considerando as perguntas existentes em fóruns técnicos é sabido que, em média, quase metade das perguntas não possui uma resposta aceita pelo usuário. Isso é um indicativo importante, pois demonstra que os usuários que realizaram perguntas nestes *websites* não se mostram satisfeitos com as respostas obtidas. Desta forma, nosso método também busca incrementar a quantidade de perguntas com respostas aceitas.

Outro ponto a ser destacado é o escopo das pesquisas já realizadas neste contexto. A grande maioria dos pesquisadores utilizou dados de fóruns gerais, isto é, fóruns sem um escopo definido. Quando estamos tratando de fóruns de perguntas e respostas para desenvolvedores, muitas especificidades aparecem, como por exemplo, o fato da grande parte das perguntas/respostas conter blocos de código (MILANI; PARAISO, 2015). Neste cenário é necessário comparar os grupos de características propostos por outros autores e analisar se novas características (específicas de fóruns de desenvolvedores) podem vir a aumentar a acurácia da predição da qualidade de respostas em fóruns de perguntas e respostas.

Para tal, o método apresentado na Figura 10 foi desenvolvido. As perguntas e respostas são extraídas de fóruns para desenvolvedores, isto é, fóruns cujo foco principal é resolver problemas de programação. Em seguida, as características para construção do vetor de características são extraídas dos pares de perguntas e respostas. A lista de características é apresentada na seção 4.3. O valor para a classe deste vetor de características é baseado no fato que a escolha do autor da pergunta em selecionar uma das respostas como a melhor, indica que esta é uma resposta de qualidade (LIU; BIAN; AGICHTEIN, 2008). Em seguida, a base de dados construída é submetida a classificadores responsáveis por prever a qualidade das respostas (seção 4.4).

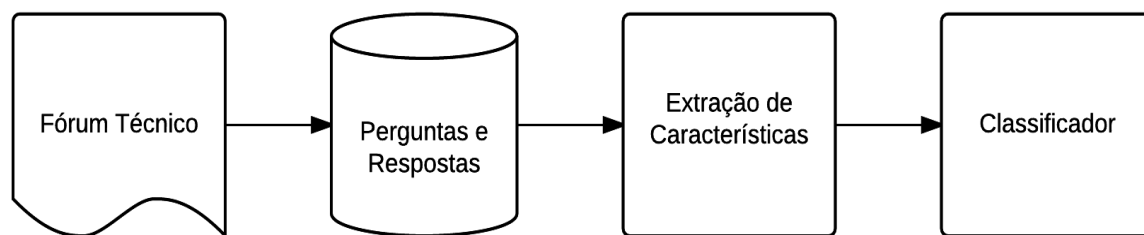


Figura 10 – Método para Predição de Qualidade de Respostas.

Fonte: o Autor.

Posteriormente, o modelo gerado é avaliado por meio das métricas definidas na seção 4.5. Finalmente, o modelo gerado pelo classificador é integrado a uma ferramenta capaz de, em tempo real, fornecer ao usuário a probabilidade da resposta que está sendo escrita ser aceita como a melhor resposta ou não. Esta ferramenta será detalhada posteriormente no Capítulo 5 (Procedimentos Metodológicos).

4.3. Seleção de Características

Analisar a qualidade das respostas é um desafio subjetivo (JEON; CROFT; LEE, 2006), por isso é comum nos fóruns de perguntas e respostas os usuários discordarem da escolha da resposta selecionada como a melhor. Esse método busca entender quais características quantitativas tendem a formar uma boa resposta, ignorando assim o campo da subjetividade. As características utilizadas nesta pesquisa se baseiam, principalmente, em explorar e unir as características previamente definidas como importantes por outros autores na literatura. Além disso, devido ao escopo desta pesquisa contemplar somente fóruns de desenvolvedores, algumas características diferentes de outras tradicionalmente utilizadas foram extraídas. As características podem ser divididas em quatro conjuntos: apresentação de qualidade, avaliação, tempo e reputação.

4.3.1. Apresentação de Qualidade

Escrever uma resposta de qualidade num fórum de perguntas e respostas compreende, minimamente, contemplar todos os requisitos de padrão de qualidade previamente descritos pelo próprio fórum. Por exemplo, o *Stack Overflow* vale-se de um conjunto de métricas

textuais para publicar ou não uma resposta, como tamanho (quantidade de caracteres), número de palavras, entre outros (CALEFATO; LANUBILE; MARASCIULO, 2015). Além destas, outras características comuns encontradas na literatura são: porcentagem de caracteres em caixa alta, número de *URLs* e tamanho do título da pergunta (LIU; FENG; LIU, 2015). Além disso, quando estamos trabalhando com fóruns de desenvolvedores outra característica se torna primordial na percepção da qualidade, seja da pergunta (DUIJN; KU; BACCHELLI, 2015) ou da resposta (TREUDE; BARZILAY; STOREY, 2011), a presença de blocos de código.

4.3.2. Avaliação

Expressar estados afetivos durante uma comunicação presencial por meio de sentimentos é comum na interação humana. Entretanto esse tipo de comunicação (não verbal) não é perceptível em fóruns de perguntas e respostas. O próprio *Stack Overflow* define orientações¹³ sobre como “ser respeitoso”, destacando que os usuários devem ser pacientes, receptivos e não menosprezar os demais usuários. Orientações necessárias pois discussões fortemente negativas desencorajam a participação (ASADUZZAMAN; MASHIYAT; ROY, 2013). Devido a não presença de interação nos fóruns de perguntas e respostas esse campo pode ser avaliado de duas diferentes maneiras: quantitativamente (usando a avaliação dos usuários através dos votos fornecidos) ou qualitativamente (através da análise de sentimentos). Quantitativamente, os votos binários (positivos ou negativos) fornecidos pelos usuários torna-se uma métrica de qualidade (PAL; FARZAN; KONSTAN, 2011). Por outro lado, acredita-se que respostas que possuem comentários com polaridade positiva indicam melhor qualidade e satisfação dos usuários (TOBA; MING; ADRIANI, 2014).

4.3.3. Tempo

Em um ambiente altamente participativo e com diversos usuários fornecendo informações paralelamente a velocidade é um dos fatores chaves para o sucesso. A mediana, no fórum *Stack Overflow* (versão em Inglês), para a resposta ser selecionada pelo questionador como a melhor é de 21 minutos e 20 segundos, já para a primeira resposta ser

¹³ Mais informações em <http://pt.stackoverflow.com/help/be-nice>

fornecida esse tempo cai para apenas 11 minutos (MAMYKINA; MANOIM; MITTAL, 2011). Usuários denominados *experts* (aqueles que possuem maiores reputações no fórum) são ainda mais rápidos (ANDERSON; HUTTENLOCHER; KLEINBERG, 2012; BOSU; CORLEY; HEATON, 2013). Sabendo-se que o questionador na maioria das vezes precisa da resposta rapidamente pois já está enfrentando o problema em questão, o tempo entre a pergunta e a resposta é primordial para que a resposta seja selecionada como a melhor (MORRIS; TEEVAN; PANOVICH, 2010).

4.3.4. Reputação

A reputação nos fóruns de perguntas e respostas é uma maneira de manter os usuários engajados e incentivar a contribuição (HART; SARMA, 2014). Além disso, um usuário com uma boa reputação tende a passar confiança aos demais usuários sobre o conteúdo depositado, pois são, teoricamente, mais efetivos em prover respostas de qualidade (ANDERSON; HUTTENLOCHER; KLEINBERG, 2012). Apesar disso, este é um tópico amplamente discutido na literatura, pois não necessariamente um novato no fórum é um iniciante no que se refere ao seu conhecimento sobre programação, sendo assim características textuais (apresentação de qualidade) contribuem mais que as características provenientes dos usuários para a classificação da qualidade de conteúdo (LEZINA; KUZNETSOV, 2013). Outro ponto relevante refere-se à completude das informações do perfil do usuário, quanto mais completo, maior é a chance do usuário prover uma resposta de qualidade (GINSKA; POPESCU, 2013).

4.3.5. Características Seleccionadas

Antes de listar as características utilizadas neste trabalho é importante destacar que além de estarmos interessados em prever a qualidade das respostas em fóruns técnicos também queremos entender quais são as principais características e fatores para uma resposta de qualidade. Esses dois pontos se diferem no sentido que, quando estamos prevendo a qualidade de uma resposta no momento de sua escrita algumas características ainda não estão disponíveis, como por exemplo: quantidade de comentários e *score* da resposta. Estas características apenas existem quando estamos considerando a base histórica. O Quadro 8 apresenta as características utilizadas nesta pesquisa, bem como sua disponibilidade para a predição e se foi ou não utilizada por outros autores.

Quadro 8 – Características Extraídas para Avaliar a Qualidade das Respostas

Fonte	Característica	Descrição	Conjunto	Disponível para Predição	Utilizada Anteriormente por outros Autores
Usuário que Perguntou	Idade	Idade do usuário.	-	Sim	Não
	Votos Positivos	Quantidade de votos positivos do usuário considerando todas suas contribuições.	Avaliação	Sim	Sim
	Votos Negativos	Quantidade de votos negativos do usuário considerando todas suas contribuições.	Avaliação	Sim	Sim
	Reputação	Reputação acumulada do usuário ao longo de suas contribuições.	Reputação	Sim	Sim
	Visualizações	Quantidade de visualizações em seu perfil.	Reputação	Sim	Não
	Medalhas de Ouro	Medalhas de ouro obtidas pelo usuário.	Reputação	Sim	Sim
	Medalhas de Prata	Medalhas de prata obtidas pelo usuário.	Reputação	Sim	Sim
	Medalha de Bronze	Medalhas de bronze obtidas pelo usuário.	Reputação	Sim	Sim
	Medalhas de Tag	Medalhas de tag obtidas pelo usuário.	Reputação	Sim	Não
Usuário que Respondeu	Idade	Idade do usuário.	-	Sim	Não
	Votos Positivos	Quantidade de votos positivos do usuário considerando todas suas contribuições.	Avaliação	Sim	Sim
	Votos Negativos	Quantidade de votos negativos do usuário considerando todas suas contribuições.	Avaliação	Sim	Sim
	Reputação	Reputação acumulada do usuário ao longo de suas contribuições.	Reputação	Sim	Sim
	Visualizações	Quantidade de visualizações em seu perfil.	Reputação	Sim	Não
	Medalhas de Ouro	Medalhas de ouro obtidas pelo usuário.	Reputação	Sim	Sim
	Medalhas de Prata	Medalhas de prata obtidas pelo usuário.	Reputação	Sim	Sim
	Medalha de Bronze	Medalhas de bronze obtidas pelo usuário.	Reputação	Sim	Sim
	Medalhas de Tag	Medalhas de tag obtidas pelo usuário.	Reputação	Sim	Não

Pergunta	Score	Quantidade de votos recebidos pelos usuários.	Sentimento	Sim	Sim
	Favorito	Quantidade de usuários que escolheram a pergunta como a favorita.	Apresentação de Qualidade	Sim	Não
	Bloco de Código	Contém ou não código na pergunta.	Apresentação de Qualidade	Sim	Sim
	Tamanho da Pergunta	Quantidade de caracteres no corpo da pergunta.	Apresentação de Qualidade	Sim	Sim
	Tamanho do Título	Quantidade de caracteres no título da pergunta.	Apresentação de Qualidade	Sim	Sim
	Tags	Quantidade de tags utilizadas na pergunta.	Apresentação de Qualidade	Sim	Não
	Respostas	Quantidade de respostas enviadas.	Apresentação de Qualidade	Sim	Sim
	Comentários	Quantidade de comentários.	Sentimento	Sim	Não
	Sentenças	Quantidade de sentenças no corpo da pergunta.	Apresentação de Qualidade	Sim	Sim
	Palavras	Quantidade de palavras no corpo da pergunta.	Apresentação de Qualidade	Sim	Sim
	URL	Quantidade de links para outros sites.	Apresentação de Qualidade	Sim	Sim
Resposta	Score	Quantidade de votos recebidos pelos usuários.	Avaliação	Não	Sim
	Bloco de Código	Contém ou não código na resposta.	Apresentação de Qualidade	Sim	Sim
	Tamanho da Resposta	Quantidade de caracteres no corpo da resposta.	Apresentação de Qualidade	Sim	Sim
	Sentenças	Quantidade de sentenças no corpo da resposta.	Apresentação de Qualidade	Sim	Sim
	Palavras	Quantidade de palavras no corpo da resposta.	Apresentação de Qualidade	Sim	Sim
	URL	Quantidade de links para outros sites.	Apresentação de Qualidade	Sim	Sim
	Comentários	Quantidade de comentários.	Avaliação	Não	Não
Perguntas e Respostas	Visualizações	Quantidade de pessoas que visualizaram a pergunta e suas respectivas repostas.	Reputação	Sim	Não
	Intervalo de Tempo	Intervalo de tempo entre a pergunta e a resposta.	Tempo	Sim	Sim
	Classe	Positivo para respostas selecionadas pelos questionadores. Negativo para as demais.	-	Sim	-

As características foram divididas em cinco grupos distintos de acordo com a fonte do dado: usuário que perguntou, usuário que respondeu, pergunta, respostas e, por fim, perguntas e respostas. No primeiro grupo estão as características do usuário que perguntou. Uma boa resposta começa com uma pergunta bem estruturada. Partindo deste princípio estas características foram extraídas pois acredita-se que dependendo do nível do usuário (reputação) o mesmo não tenha o conhecimento necessário para formular uma pergunta de qualidade e/ou ainda que não tenha conhecimento suficiente para escolher a melhor resposta. Partindo desta premissa essas características foram extraídas para serem testadas no modelo. Ao contrário, no grupo das características do usuário que respondeu, espera-se que a qualidade de respostas construídas por usuários com elevada reputação seja um fator importante para a resposta ser selecionada como a melhor. O grupo de características extraídas das perguntas e respostas possivelmente acrescentará maior ganho ao modelo, uma vez que essas características independem de o usuário ser novato no fórum ou não. Por fim, o grupo que considera ambas as fontes (perguntas e respostas) busca encontrar uma relação de qualidade entre elas.

A extração destas características será realizada através de um processo automático. As perguntas e respostas serão carregadas em um banco de dados e pré-processadas através de rotinas e consultas (*queries*) para a geração dos pares de perguntas e respostas. Posteriormente, os dados serão novamente processados por uma ferramenta para que as características supracitadas sejam extraídas. O resultado deste processamento será um arquivo texto preparado para os algoritmos de classificação. Os detalhes técnicos para que essa extração de características seja possível são apresentados no capítulo 5.

4.4. Aprendizagem e Avaliação do Modelo

Como visto na fundamentação teórica o campo da Aprendizagem de Máquina possui diversas tarefas para a descoberta de conhecimento em bases de dados. Para o intuito deste trabalho, a tarefa escolhida é a classificação. A escolha se justifica, pois, a ideia principal é construir um modelo capaz de prever a qualidade de respostas em fóruns de perguntas e respostas para desenvolvedores. As regras de associação poderiam ser utilizadas para encontrar padrões úteis na base de dados, mas esse tipo de tarefa não poderia, por exemplo, prever se uma nova resposta seria ou não selecionada pelo questionador.

Dentre os algoritmos de classificação disponíveis, o algoritmo C4.5 do ambiente WEKA (HALL; FRANK; HOLMES, 2009) foi escolhido para gerar o modelo. A Figura 11 apresenta, parte do modelo gerado pelo algoritmo.

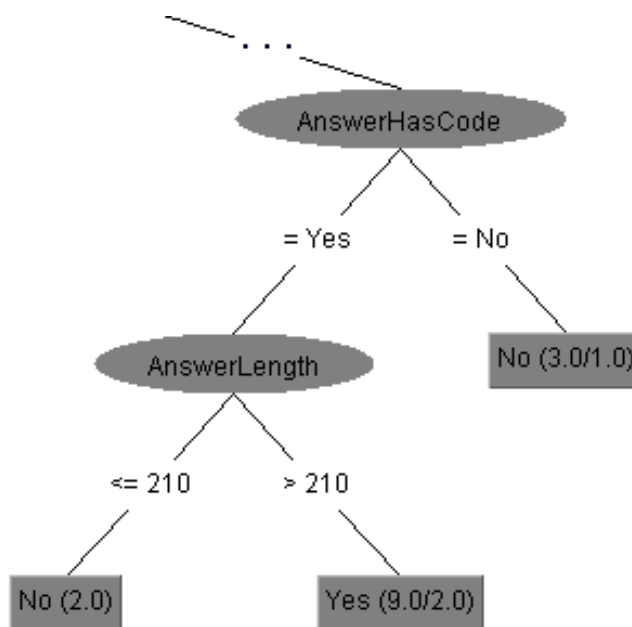


Figura 11 – Modelo de Saída do Algoritmo C4.5.

Fonte: o Autor.

A justificativa para a escolha do algoritmo C4.5 é atribuída à ferramenta (capaz de classificar em tempo real novas respostas) construída ao longo deste trabalho. Esta ferramenta tem como objetivo fornecer, além da probabilidade da nova resposta ser escolhida como a melhor pelo questionador, como o modelo chegou a essa conclusão, deixando, assim, o usuário final com uma clara explicação sobre o porquê sua resposta está sendo classificada como a melhor, ou não. Esta explicação consiste em demonstrar o caminho (nós) percorrido na árvore de decisão até o nó folha através de regras do tipo se → então.

Buscando garantir que o modelo gerado possui uma boa capacidade de generalização, as bases históricas mais antigas são utilizadas para treinamento e as mais recentes para teste. Finalmente, como métrica de avaliação de desempenho dos modelos preditivos, será utilizada a acurácia, isto é, a probabilidade de o modelo prever corretamente uma nova resposta (se será ou não selecionada pelo questionador).

4.5. Conclusão

Como visto neste capítulo, o método proposto contempla uma série de atividades necessárias para sua construção. Este método automatiza a análise da qualidade de respostas em fóruns de perguntas e respostas para desenvolvedores, buscando alavancar, assim, a qualidade das respostas fornecidas. Ademais, o método se diferencia dos demais por focar em fóruns de perguntas e respostas para desenvolvedores e explorar características ainda não utilizadas. No próximo capítulo serão fornecidos todos os detalhes técnicos referentes à implementação do método exposto.

Capítulo 5

Procedimentos Metodológicos

Neste capítulo é descrito tecnicamente o processo de predição de qualidade de respostas em fóruns de perguntas e respostas para desenvolvedores e suas respectivas etapas que compõem o método proposto. Este capítulo está organizado da seguinte forma: inicialmente, na Seção 5.1, é apresentada a base de dados histórica utilizada para a predição de qualidade de respostas. Na Seção 5.2 é explicado em detalhes o método para a extração das características anteriormente selecionadas, bem como as particularidades de processamento necessárias de algumas características. Finalmente, a Seção 5.3 apresenta uma conclusão para o capítulo.

5.1. Base de Dados

A base de dados utilizada nesta pesquisa para prever a qualidade de respostas é extraída do fórum de perguntas e respostas *Stack Overflow* em Português. É importante destacar que o método é passível de ser aplicado nos demais fóruns da rede em outras línguas, como por exemplo, o *Stack Overflow* em Inglês. A escolha do fórum se dá, primeiramente, pelo fato da sua base de dados ser pública, além de o *Stack Overflow* ser o maior e mais popular fórum de desenvolvimento de software (DUIJN; KU; BACCHELLI, 2015).

5.1.1. Obtenção dos Dados

Os dados do *Stack Overflow*, como dito anteriormente, são públicos e podem ser compartilhados e modificados livremente. É possível acessar as perguntas e respostas através

de uma série de APIs disponíveis em diferentes linguagens de programação, desenvolvidas pelos próprios usuários. O fórum também conta com uma API oficial para acesso aos seus dados. Além das APIs, o fórum possui um suporte a consultas no banco de dados em tempo real¹⁴. Outra possibilidade para a obtenção dos dados do *Stack Overflow*, é um *data dump*¹⁵ gerado a cada três meses pelos administradores da rede e disponibilizado para download via *torrent*. Este *data dump* possui todo o histórico do fórum em arquivos XML (*eXtensible Markup Language*) desde a sua criação. Neste trabalho foram utilizados a API oficial do fórum para a criação da ferramenta de predição de qualidade e os *data dumps* disponibilizados (ao longo destes dois anos de pesquisa) para a construção dos modelos preditivos. Esses arquivos foram carregados em um banco de dados e pré-processados visando à criação de um arquivo texto para posterior extração de características. As seções a seguir explicam passo a passo como foi realizado o carregamento dos dados, o pré-processamento e a estrutura do arquivo final.

5.1.2. Pré-processamento

Primeiramente é necessário entender a estrutura dos dados disponíveis no *data dump* disponibilizado pelo *Stack Overflow*. Cada arquivo XML representa uma tabela do banco de dados interno do fórum. Vale ressaltar que apenas as tabelas com os dados históricos são exportadas para os arquivos XML. Tabelas que representam características internas da base, como por exemplo, os tipos de votos, razões pela qual as respostas foram fechadas, entre outras, ficam disponíveis para consulta apenas via API oficial do fórum. O Quadro 9 exhibe todos os arquivos fornecidos e seu respectivo conteúdo.

Quadro 9 – Descrição dos Arquivos do *Data Dump*.

Arquivo	Conteúdo
Votes.xml	Contém todos os votos realizados pelos usuários no fórum.
Users.xml	Informações relacionadas aos usuários, como: idade, quantidade de votos positivos recebidos, quantidade de votos negativos recebidos, entre outros.
Tags.xml	Todas as <i>tags</i> utilizadas no fórum, bem como quando e em qual pergunta foram criadas.

¹⁴ Mais informações em: <http://data.stackexchange.com/stackoverflow/query/new>

¹⁵ Mais informações em: <https://archive.org/details/stackexchange>

Posts.xml	Contém todo o conteúdo das perguntas, como: data de criação, usuário que perguntou, quantidade de respostas, entre outros.
PostLinks.xml	Utilizado para armazenar as perguntas duplicadas do fórum e relacioná-las com a pergunta original.
PostHistory.xml	Armazena toda e qualquer alteração de uma pergunta, como por exemplo: edição, novo comentário, entre outros.
Comments.xml	Contém todos os comentários realizados no fórum, tanto para pergunta como para as respostas.
Badges.xml	Informação sobre quando e qual usuário recebeu uma nova medalha, bem como sua classe (ouro, prata ou bronze).

Buscando clarificar um pouco mais a estrutura destes arquivos a Figura 12 exibe uma pergunta do fórum sobre como inverter uma *string* em JAVA e o Quadro 10 a respectiva representação em XML. É importante destacar que o momento em que o *data dump* foi gerado e a captura da imagem realizada não são os mesmos, desta forma existem algumas divergências entres os valores apresentados entre a imagem e o XML. Além disso, as *tags* em HTML foram removidas do atributo “*Body*” para facilitar a visualização.

Como inverter uma String?

▲ Um usuário digitará uma frase e devo mostrar esta frase invertida e em maiúsculo. Como posso fazer isto?

7

▼

★

1

```
import javax.swing.JOptionPane;

public class ExerLar01 {
    public static void main(String args[])
    {
        char vetor[];
        String frase = JOptionPane.showInputDialog(null, "Digite uma frase: ");
        vetor = frase.toCharArray();
    }
}
```

java string

Figura 12 – Pergunta “Como Inverter uma String em JAVA”

Fonte: pt.stackoverflow.com/questions/116062

Quadro 10 – Representação em XML (Posts.xml) da Figura 12.

```
<row Id="116062"
  PostTypeId="1"
  AcceptedAnswerId="116064" CreationDate="2016-03-01T22:32:08.560"
  Score="3" ViewCount="34"
  Body="<p>Um usuário digitará uma frase e devo mostrar esta frase invertida
e em maiúsculo. Como posso fazer isto?</p> <pre><code>import
javax.swing.JOptionPane; public class ExerLar01 { public static void main(String
args[]) {char vetor[]; String frase = JOptionPane.showInputDialog(null, "Digite
uma frase: "); vetor = frase.toCharArray(); }}
</code></pre>"
  OwnerUserId="38658" LastEditorUserId="4808"
  LastEditDate="2016-03-02T01:18:52.960"
  LastActivityDate="2016-03-02T01:18:52.960"
  Title="Como inverter uma String?" Tags="&lt;java&gt;&lt;string&gt;"
  AnswerCount="1" CommentCount="0" FavoriteCount="1"/>
```

Analisando a figura e o XML é fácil perceber as relações, por exemplo, o atributo “*Title*” refere-se ao título da pergunta, já o atributo “*AnswerCount*” a quantidade de respostas para esta pergunta. O atributo “*AcceptedAnswerId*” fornece o identificador de qual foi a resposta selecionada pelo questionador, tornando assim possível extrair os pares de perguntas e respostas e etiquetar as que formam selecionadas como as respostas de qualidade. Da mesma forma as respostas são representadas nos arquivos XML. A Figura 13 exibe a resposta selecionada pelo questionador da pergunta anterior (“Como inverter uma *string* em JAVA”) e o Quadro 11 o respectivo trecho do XML.

▲ Tente usar o método `reverse` do `StringBuilder` :

11

```
String hi = "Hello world";
System.out.println(new StringBuilder(hi).reverse().toString());
```

▼

✓ Veja este exemplo funcionando aqui no [IDEONE](#).

Adaptando para seu caso:

```
String frase = JOptionPane.showInputDialog(null, "Digite uma frase: ");
String fraseInvertida = new StringBuilder(frase).reverse().toString();
System.out.println(fraseInvertida);
//ou
JOptionPane.showMessageDialog(null, fraseInvertida);
```

Isso só funciona se a versão do JDK for 1.5 ou superior. Para versões mais antigas, basta alterar `StringBuilder` por `StringBuffer` .

Referencia: [Reverse a string in Java](#).

Figura 13 – Resposta Selecionada pelo Questionador

Fonte: pt.stackoverflow.com/questions/116062

Quadro 11 – Representação em XML (Posts.xml) da Figura 13.

```

<row Id="116064"
  PostTypeId="2"
  AcceptedAnswerId="116064"
  CreationDate="2016-03-01T22:32:08.560"
  Score="3"
  ViewCount="34"
  Body=" <p>Tente usar o método <a
href="https://docs.oracle.com/javase/7/docs/api/java/lang/StringBuilder.html#reverse()"><code>reverse</code></a> do <a
href="https://docs.oracle.com/javase/7/docs/api/java/lang/StringBuilder.html"><code>StringBuilder</code></a>:</p>
<pre><code>String hi = "Hello world";
System.out.println(new StringBuilder(hi).reverse().toString());
</code></pre>
<p>Veja este exemplo funcionando aqui no <strong><a
href="http://ideone.com/OttF79">IDEONE</a></strong>.</p>
<p>Adaptando para seu caso:</p>
<pre><code> String frase = JOptionPane.showInputDialog(null, "Digite uma frase:
");
String fraseInvertida = new StringBuilder(frase).reverse().toString();
System.out.println(fraseInvertida);
//ou
JOptionPane.showMessageDialog(null, fraseInvertida);
</code></pre>
<blockquote>
  <p>Isso só funciona se a versão do JDK for 1.5 ou superior. Para versões
  mais antigas, basta alterar <code>StringBuilder</code> por
  <code>StringBuffer</code>.</p>
</blockquote>
<p>Referencia: <a
href="http://stackoverflow.com/a/7569370/5524514"><strong>Reverse a string in
Java</strong></a>.</p>
  OwnerUserId="38658"
  LastEditorUserId="4808"
  LastEditDate="2016-03-02T01:18:52.960"
  LastActivityDate="2016-03-02T01:18:52.960"
  Title="Como inverter uma String?"
  Tags="&lt;java&gt;&lt;string&gt;"
  AnswerCount="1"
  CommentCount="0"
  FavoriteCount="1"
/>

```

Estes arquivos foram importados na ferramenta *MySQL Workbench* por meio da adaptação de um script¹⁶ SQL (*Structured Query Language*) preexistente. A Figura 14 exibe a estrutura do banco de dados utilizada para que a importação fosse possível.

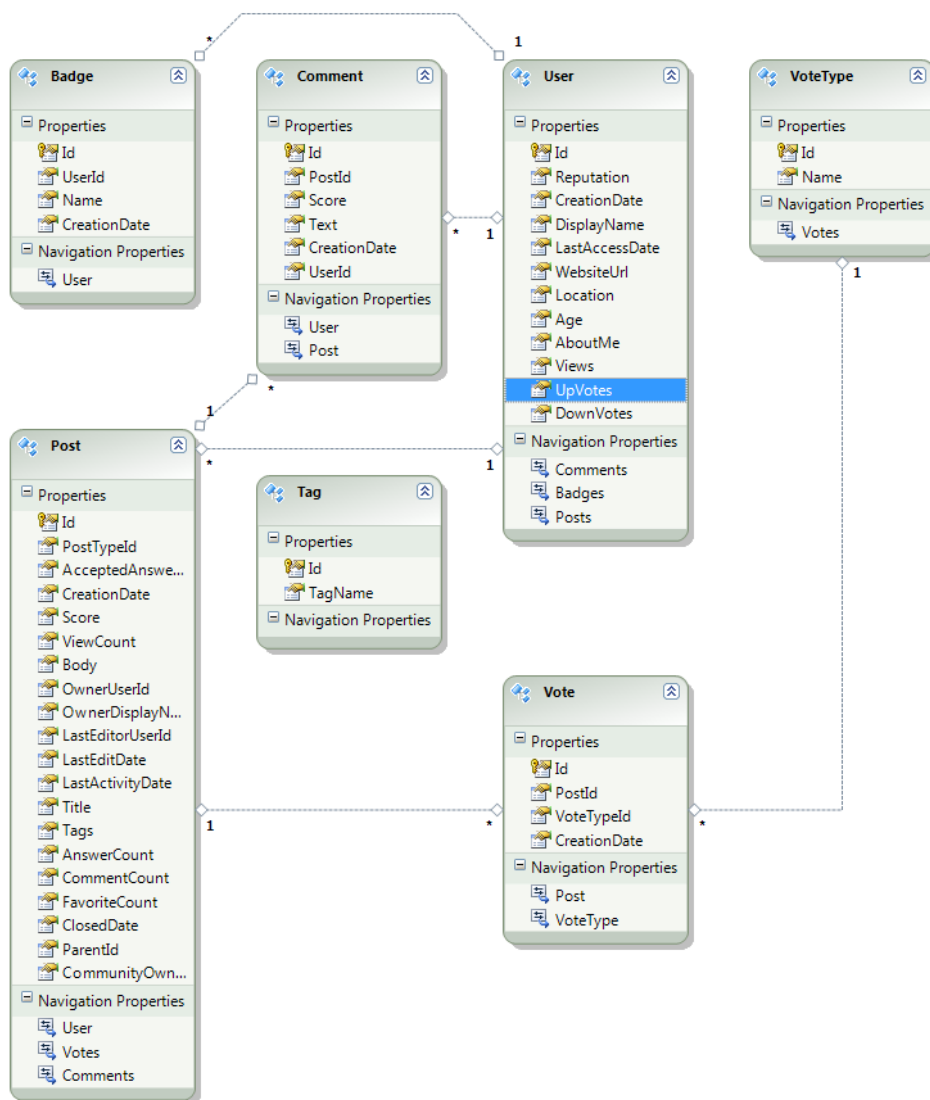


Figura 14 – Esquema de Dados para Importação dos Dados do *Stack Overflow*

Fonte: o Autor.

Como o trabalho se propõe analisar as respostas considerando a escolha de quem perguntou, apenas foi considerado o conjunto de perguntas que contemplem a resposta selecionada pelo questionador. Desta forma, todas as perguntas sem o atributo “*AcceptedAnswerId*” são automaticamente eliminadas do processo. É de suma importância

¹⁶ Mais informações em: <https://gist.github.com/gousiosg/7600626>

ressaltar que muitas perguntas possuem mais de uma resposta, sendo assim a selecionada pelo questionador é rotulada como “resposta de qualidade” e, as demais, rotuladas como “qualidade baixa”. Outros três filtros são utilizados nesta fase. A eliminação de perguntas e respostas com *score* negativo, pois estas indicam um claro descontentamento por parte dos usuários do fórum. A remoção de respostas que foram fornecidas após o questionador já ter escolhido uma como a que lhe ajudou. Por fim, o *Stack Overflow* adota uma política que, quando um usuário é excluído, suas perguntas e respostas deixam de possuir seu identificador. Como os atributos dos usuários são parte do nosso método, perguntas e respostas sem identificação do usuário são também automaticamente eliminadas.

A saída desta etapa é um arquivo texto contemplando todas as perguntas, as suas respectivas respostas e as informações do usuário (quem respondeu e quem perguntou), desde que atendam os requisitos anteriormente citados. O Quadro 12 exhibe detalhadamente o modelo do arquivo gerado.

Quadro 12 – Modelo do Arquivo Gerado Após a Etapa de Pré-processamento.

Id da Pergunta	Pergunta	Usuário que Perguntou	Id da Resposta	Resposta	Usuário que Respondeu	Classe (Resposta Selecionada pelo Questionador)
1	Dados referentes a Pergunta 1	Dados referentes ao usuário que realizou a Pergunta 1	1	Dados referentes a Resposta 1	Dados referentes ao usuário que realizou a Resposta 1	Sim
1			2	Dados referentes a Resposta 2	Dados referentes ao usuário que realizou a Resposta 2	Não
2	Dados referentes a Pergunta 2	Dados referentes ao usuário que realizou a Pergunta 2	3	Dados referentes a Resposta 3	Dados referentes ao usuário que realizou a Resposta 3	Sim
3	Dados referentes a Pergunta 3	Dados referentes ao usuário que realizou a Pergunta 3	4	Dados referentes a Resposta 4	Dados referentes ao usuário que realizou a Resposta 4	Não
3			5	Dados referentes a Resposta 5	Dados referentes ao usuário que realizou a Resposta 5	Sim

3			6	Dados referentes a Resposta 6	Dados referentes ao usuário que realizou a Resposta 6	Não
X_n	Dados Referentes a Pergunta X_n	Dados referentes ao usuário que realizou a Pergunta X_n	X_n	Dados referentes a Resposta X_n	Dados referentes ao usuário que realizou a Resposta X_n	Classe para a Resposta X_n

Vale destacar que esta etapa apenas organiza (desnormaliza) os dados. Nenhum processamento adicional é realizado sobre os dados nesta etapa.

5.2. Extração de Características

O processo de extração de características é realizado através de um programa em JAVA (apelidado de *Stack Overflow Explorer*) desenvolvido ao longo deste trabalho. Este programa utiliza o arquivo texto gerado pela etapa anterior para processar as características que necessitam de tratamento um específico além de extrair outras adicionais. Da lista de características propostas (Quadro 8) as que dependem de processamento para serem obtidas são: tamanho do título, tamanho da pergunta, tamanho da resposta, *tags*, intervalo de tempo, bloco de código, sentenças, palavras e URL. Sendo as quatro últimas válidas tanto para a pergunta como para a resposta.

Tamanho do título: a extração desta característica compreende em simplesmente contar a quantidade de caracteres no título da pergunta. Por exemplo, para o título “Como inverter uma string?” o resultado é 22 caracteres.

Tamanho da pergunta/resposta: ao contrário do processamento para a extração do tamanho do título a obtenção do tamanho da pergunta/resposta possui uma etapa inicial. Os dados brutos fornecidos pelo Stack Overflow para o conteúdo das perguntas e respostas contêm marcadores em HTML (como o marcador “<p>” utilizado em parágrafos) necessários para o site melhor organizar e exibir a informação. Entretanto, esses marcadores são removidos para a contagem dos caracteres. Além disso, a quantidade de caracteres referentes a blocos de códigos é subtraída do montante final.

Tags: as tags (etiquetas) são fornecidas nos dados brutos separadas por ponto e vírgula (por exemplo, “Java; string”), nesse contexto um *parser* é utilizado para determinar a quantidade de *tags* utilizadas.

Intervalo de tempo: o intervalo de tempo é calculado, em segundos, através da diferença entre o tempo de criação da resposta e pergunta.

Bloco de código: esta característica é binária e se refere ao corpo, seja da pergunta ou resposta, conter um bloco de código. Ao contrário da característica para extrair o tamanho da resposta/pergunta, os marcadores HTML neste caso facilitam essa verificação. O marcador “<code>” é utilizado para iniciar blocos de código no *Stack Overflow*. Sendo assim, caso o corpo da pergunta/resposta possua este marcador, a característica é verdadeira, caso contrário, falsa.

Sentenças: a quantidade de sentenças é extraída do corpo da pergunta/resposta (sem os marcadores HTML) através de expressões regulares.

Palavras: assim como a quantidade de sentenças, a quantidade de palavras também é extraída através de expressões regulares do corpo da pergunta/resposta sem os marcadores HTML.

URL: o marcador HTML para URLs é o “<a>”. A quantidade de URLs nas perguntas/respostas é determinada pela contagem destes marcadores.

O resultado deste processamento é arquivo CSV (*Comma-separated Values*), seguindo o mesmo modelo gerado pela etapa anterior, ou seja, preparado para os algoritmos de classificação. A Figura 15 ilustra, passo a passo, as etapas necessárias para a sua construção. Posteriormente, este arquivo é carregado no ambiente WEKA para a geração dos modelos preditivos.

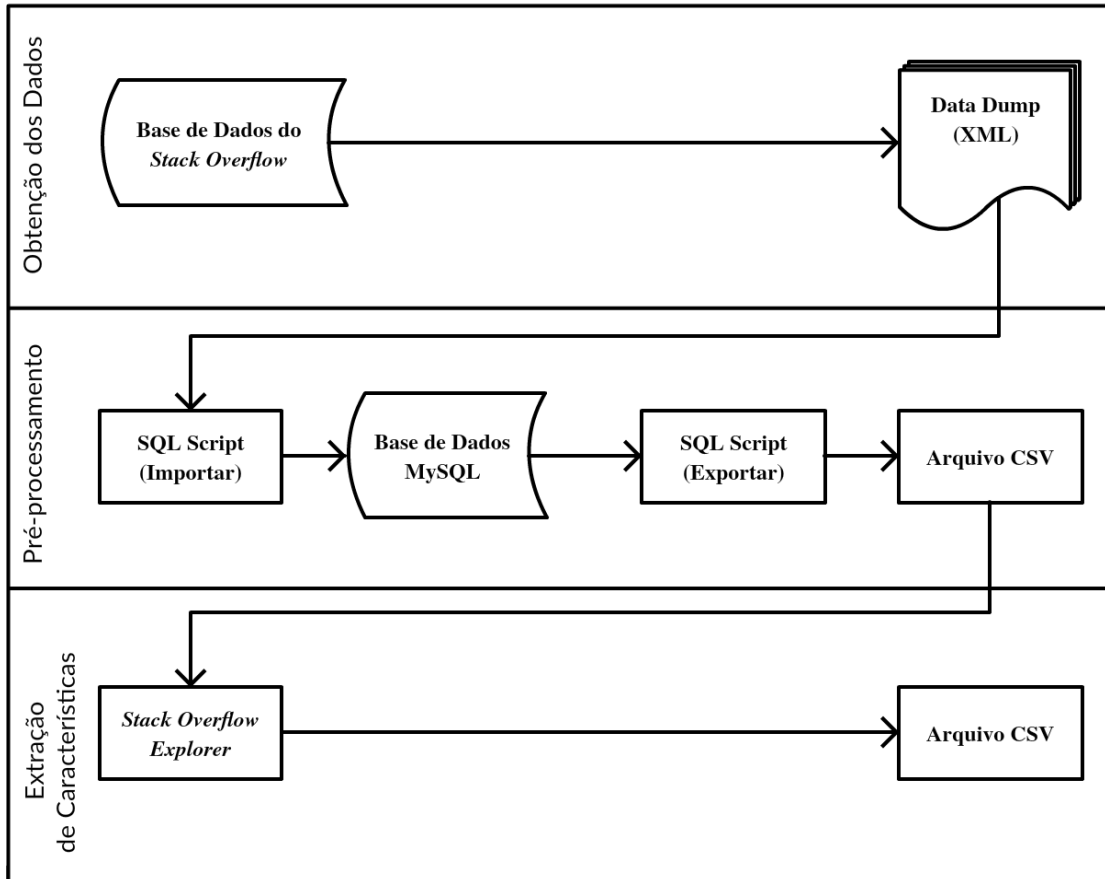


Figura 15 – Procedimentos Metodológicos e suas Etapas

Fonte: o Autor.

5.3. Conclusão

Neste capítulo foi demonstrado, passo a passo, as etapas que compõem o método proposto. Fica evidente, na prática, que a etapa de pré-processamento é bastante custosa e demorada, pois exige uma série de cuidados, como a extração das características e a geração de arquivos bem definidos. Além disso, esse capítulo apresentou todos os detalhes técnicos necessários para a aplicação da predição da qualidade de respostas no fórum *Stack Overflow*.

No próximo capítulo serão apresentados os resultados obtidos com a utilização do método proposto, bem como a ferramenta construída para a predição da qualidade de respostas.

Capítulo 6

Resultados

Neste capítulo são descritos os resultados desta pesquisa. Os resultados foram obtidos através da realização de três diferentes experimentos utilizando amostras da base de dados histórica do fórum *Stack Overflow*. Primeiramente, na Seção 6.1 são apresentadas as bases de dados utilizadas, bem como uma análise quantitativa. Em seguida, na Seção 6.2 são apresentados os três experimentos realizados: classificação contemplando todas as características extraídas, classificação contemplando somente os atributos previamente disponíveis, uma análise sobre o melhor modelo obtido nesta pesquisa e, por fim um comparativo com o estado da arte. Na Seção 6.3 é apresentada a ferramenta construída para a predição da qualidade de respostas no fórum *Stack Overflow*. Finalmente, na Seção 6.4, é apresentada uma conclusão sobre os resultados obtidos.

6.1. Bases de Dados

As bases de dados utilizadas para a construção dos resultados foram extraídas do fórum *Stack Overflow* ao longo desta pesquisa. Vale destacar novamente que os dados são disponibilizados a cada três meses de maneira incremental, isto é, cada base de dados contempla todos os dados históricos desde o momento da criação do fórum até a geração do seu respectivo *data dump*. A Tabela 4 exhibe as bases utilizadas nesta pesquisa, bem como a quantidade de perguntas, a quantidade de respostas e a quantidade de perguntas com respostas aceitas pelo questionador.

Tabela 4 – Bases de Dados Utilizadas

<i>Data Dump</i>	Quantidade de Perguntas	Quantidade de Respostas	Quantidade de Perguntas com Respostas Aceitas
Setembro de 2014	9.204	14.776	5.275
Março de 2015	16.190	24.089	9.164
Janeiro de 2016	35.302	47.140	18.752
Março de 2016	38.962	51.706	20.717

Visando melhor destacar as informações disponíveis nestas bases, é imprescindível apresentar uma análise quantitativa sobre a perspectiva dos quatro conjuntos de características destacados durante a apresentação do Método, na Seção 4. São eles: apresentação de qualidade, avaliação, tempo e reputação. As análises apresentadas a seguir foram extraídas do último *data dump* disponível (março de 2016).

Sobre o conjunto referente à **apresentação de qualidade**, um dos atributos principais é a presença de blocos de código na resposta. Na base de dados, 83% das respostas contém pelo menos um bloco de código em seu conteúdo, ou seja, apenas 17% responderam à pergunta sem inserir nenhum trecho de código. A Figura 16 exibe o gráfico referente a essa percentagem.

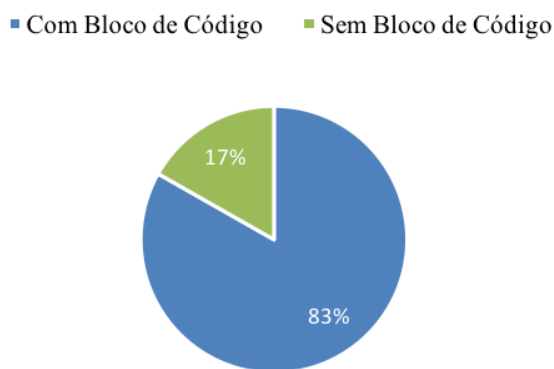


Figura 16 – Presença de Blocos de Código na Resposta

Fonte: o Autor.

Analisando somente as respostas que foram aceitas pelos questionadores, esse valor sobe de 83%, para 88%. Ou seja, ao responder uma pergunta no *Stack Overflow* espera-se que seu conteúdo contemple um trecho de código para o problema em questão.

Em seguida, o conjunto de características de **avaliação** (lembrando que estes estão sendo trabalhados sobre o ponto de vista quantitativo e não subjetivo) apresenta alguns dados

interessantes para o entendimento do fórum. Por exemplo, através da quantidade de votos positivos e negativos é possível determinar a percepção dos usuários sobre o conteúdo depositado no *Stack Overflow*. Neste contexto, a Figura 17 exibe que 23% das respostas fornecidas no fórum tem um *score* menor ou igual à zero (tratado como *score* negativo neste trabalho), ou seja, os usuários desaprovaram mais de 11 mil respostas das quase 52 mil respostas fornecidas pela comunidade. Este número reforça a afirmação que fóruns de perguntas e respostas necessitam de uma atenção especial quanto à qualidade do conteúdo ali depositado.

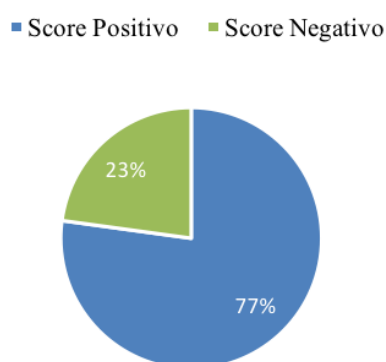


Figura 17 – *Score* das Respostas Depositadas no Stack Overflow

Fonte: o Autor.

Devido a grande quantidade de usuários e a velocidade que as perguntas e respostas são inseridas no fórum, a característica **tempo** se demonstra fundamental para determinar se a resposta será ou não escolhida pelo questionador. De acordo com a base, a mediana da diferença entre o momento da postagem da pergunta e a primeira resposta é de apenas 52 minutos, já para a resposta selecionada pelo questionador esse tempo sobe para 55 minutos. A Figura 18 exibe um gráfico acumulativo com a percentagem de perguntas respondidas (primeira resposta e resposta aceita) nas primeiras 48 horas.

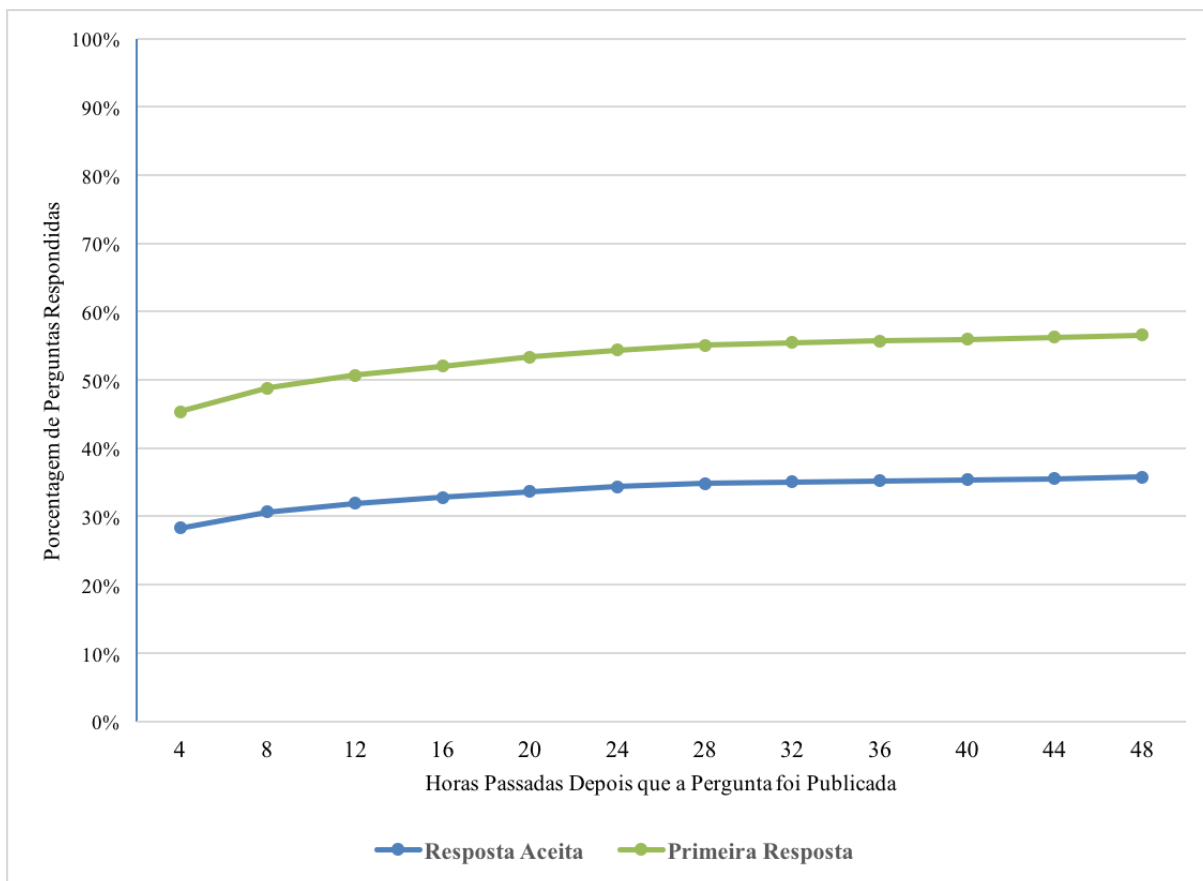


Figura 18 – Porcentagem de Perguntas Respondidas nas Primeiras 48h

Fonte: o Autor.

Quase 60% das perguntas recebe uma resposta nas primeiras 48 horas, destas, 35,73% são escolhidas pelo questionador como a que lhe ajudou a resolver seu problema. Apesar de um pouco mais lento que o fórum em sua versão em Inglês (11 minutos para a primeira resposta e 21 minutos para a resposta aceita), a velocidade se demonstra uma característica essencial ao analisar se uma resposta será ou não aceita pelo questionador. Como curiosidade, 5645 (14,46%) perguntas não possuem nenhuma resposta para a base de dados em questão.

Por fim, o conjunto de características referentes à **reputação**. Esse conjunto está diretamente ligado à história do usuário no fórum (reputação, votos negativos, positivos, entre outros). Porém, primeiramente, é necessário entender a composição dos usuários no *Stack Overflow*. A Figura 19 exibe a quantidade de usuários por faixa etária. O gráfico leva em consideração apenas os usuários que forneceram essa informação, pois esta é opcional ao criar o perfil. Além disso, foram removidos dados considerados *outliers* (dados mentirosos), como por exemplo idades superiores a 100 anos e inferiores a 10.

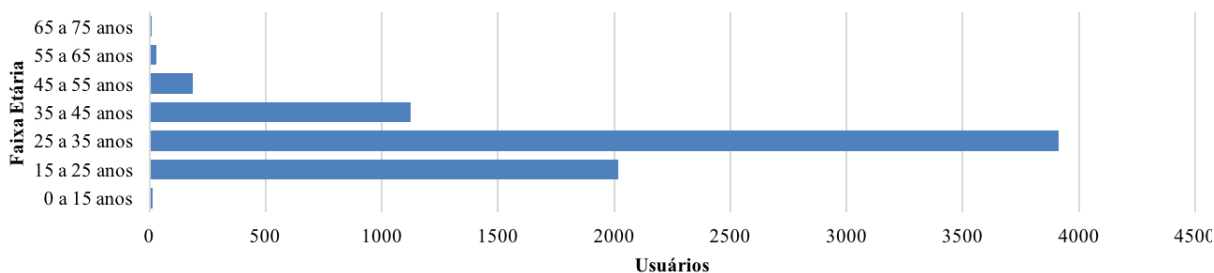


Figura 19 – Quantidade de Usuários por Faixa Etária

Fonte: o Autor.

É possível observar que a maioria, quase quatro mil usuários, está localizada na faixa etária entre 25 e 35 anos, a qual é possivelmente formada por profissionais da área de computação. Outra faixa que se destaca é a de 15 a 25 anos, provavelmente formada por estudantes de áreas relacionadas à computação e entusiastas. Infelizmente o *Stack Overflow* não possui o gênero do usuário, desta forma não é possível analisar se a quantidade de usuários femininos e masculinos é proporcional.

Dito isso, é possível analisarmos duas situações levantadas ao longo deste documento: (i) usuários mais experientes (na vida real) fornecem melhores respostas e, conseqüentemente possuem maior reputação; (ii) usuários com maior reputação fornecem respostas com qualidade superior. A Figura 20 exibe a média de reputação por faixa etária, validando a ideia que usuários mais experientes tendem a fornecer respostas melhores.

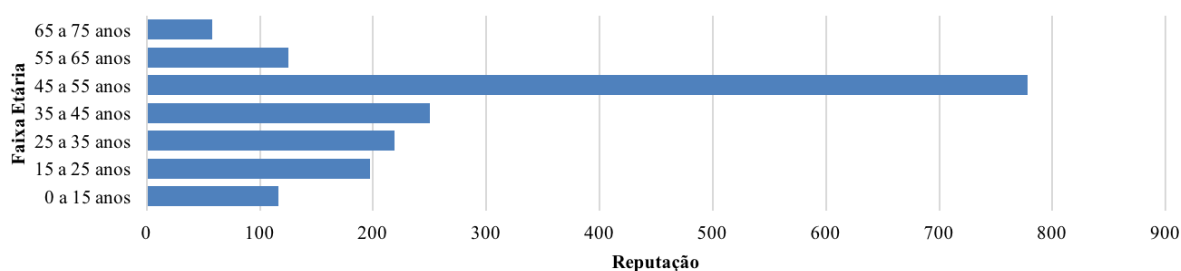


Figura 20 – Média de Reputação por Faixa Etária

Fonte: o Autor.

A percepção que um usuário com maior reputação oferece melhores respostas também é verdadeira. A Figura 21 exibe a percentagem de perguntas aceitas e não aceitas por faixa de reputação. Usuários novatos (0 a 1000 de reputação) têm cerca de 30% das respostas

fornecidas aceitas pelos questionadores, já usuários experts (com mais de 7000 de reputação) chegam à marca de quase 60%.

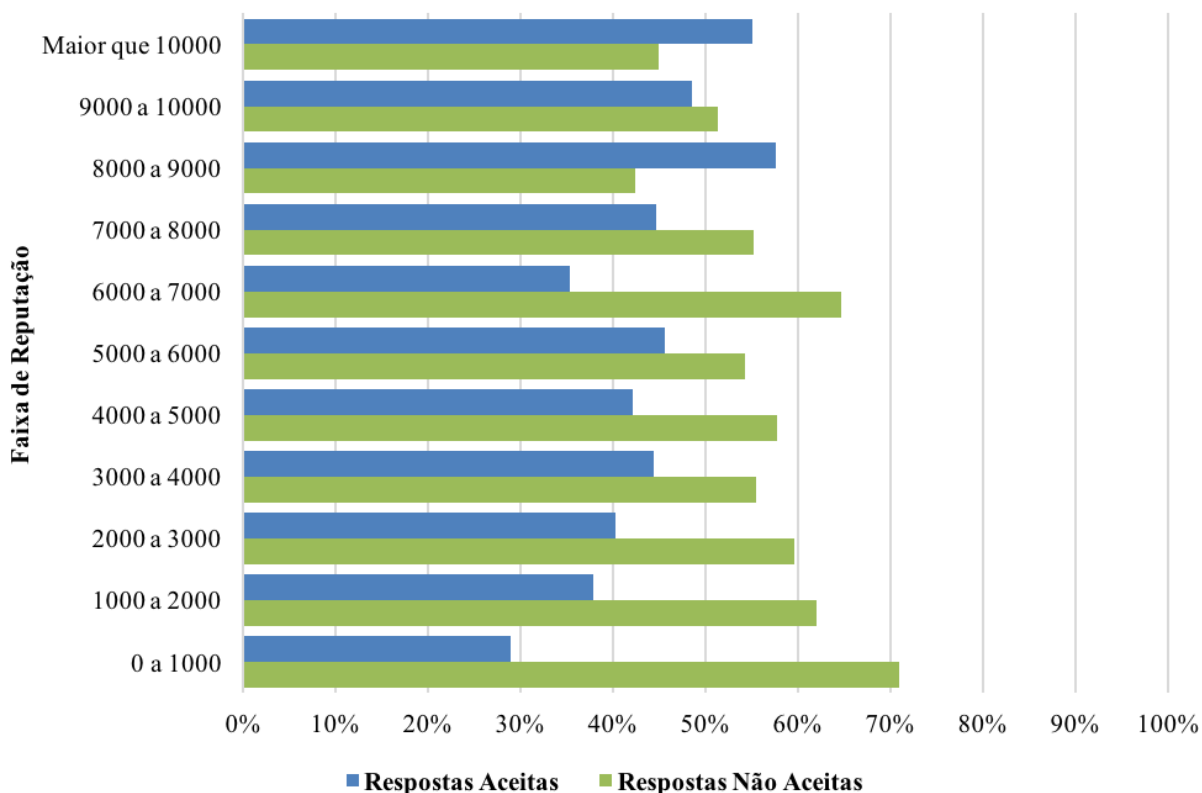


Figura 21 – Porcentagem de Respostas Aceitas e Não Aceitas por Faixa de Reputação

Fonte: o Autor.

Porém, é preciso cautela sobre essa afirmação, pois analisando os números, das 20.717 respostas aceitas, 5.777 (27,88%) foram escritas por usuários com menos de 1000 pontos de reputação. Isso, possivelmente, deve-se ao fato que um usuário experiente (45 a 55 anos), pode ser um novato no fórum. Isto é, criou sua conta há pouco tempo e ainda não possui uma base de perguntas respondidas para lhe garantir uma boa reputação.

6.2. Experimentos

As bases utilizadas para a realização dos experimentos a seguir tiveram uma queda no número de perguntas e respostas, pois os filtros mencionados nos Procedimentos Metodológicos (Seção 5) foram aplicados, são eles: remoção das perguntas que não possuem uma resposta selecionada como a melhor, eliminação de perguntas e respostas com *score*

negativo ou igual à zero, eliminação das respostas enviadas depois da escolha do questionador e, por fim, a remoção de perguntas e respostas criadas por usuários que foram excluídos da base de dados. A Tabela 5 exibe as bases de dados, bem como suas respectivas quantidades de perguntas e respostas após o pré-processamento.

Tabela 5 – Bases de Dados Pré-processadas.

Data Dump	Quantidade de Perguntas	Quantidade de Respostas
Setembro de 2014	3.921	6.636
Março de 2015	6.533	10.382
Janeiro de 2016	13.471	20.518
Março de 2016	14.773	22.404

Todas as características (Quadro 8) foram extraídas para cada uma das bases de dados de acordo com os processos anteriormente definidos, e, posteriormente, submetidos a seis algoritmos de classificação: C4.5 (fator de confiança 0,25 e número mínimo de instâncias por folha igual a 3), PART (fator de confiança 0,25 e *seed* igual a 2), *RandomTree* (*seed* igual a 2), *RandomForest* (número de árvores a serem geradas igual a 100), *DecisionTable* (algoritmo de busca *BestFirst*) e *DecisionStump* (parâmetros padrão), disponíveis no ambiente de Aprendizado de Máquina, WEKA. É importante destacar que o algoritmo utilizado no modelo final será o C4.5, os outros algoritmos foram selecionados apenas para a realização de um comparativo no que se refere a suas respectivas acurácias. A base de setembro de 2014 foi utilizada para o treinamento e a base de março de 2016 (eliminando as perguntas e respostas contidas na base de setembro) para teste.

6.2.1. Experimento 1: Modelo Utilizando Todas as Características

A ideia deste experimento é analisar o conjunto total de características extraídas da base de dados contra os algoritmos de classificação anteriormente citados. A Tabela 6 exibe os resultados, ordenados decrescentemente pela acurácia, para esse experimento através das medidas: Acurácia (*Accuracy*), *F-measure*, Sensitividade (*Recall*), Precisão (*Precision*) e área embaixo da curva ROC ou AUC (*area under curve*).

Tabela 6 – Resultados para o Modelo Utilizando Todas as Características.

Base de Treinamento	Base de Teste	Algoritmo	Accuracy	F-measure	Recall	Precision	AUC
Setembro de 2014	Março de 2016	<i>RandomForest</i>	0,805	0,862	0,883	0,841	0,893
Setembro de 2014	Março de 2016	<i>DecisionTable</i>	0,796	0,855	0,868	0,842	0,88
Setembro de 2014	Março de 2016	PART	0,781	0,838	0,82	0,857	0,836
Setembro de 2014	Março de 2016	C4.5	0,778	0,841	0,851	0,832	0,797
Setembro de 2014	Março de 2016	<i>DecisionStump</i>	0,726	0,752	0,603	1	0,801
Setembro de 2014	Março de 2016	<i>RandomTree</i>	0,722	0,796	0,788	0,805	0,686

Analisando os resultados, o modelo com melhor desempenho foi construído pelo algoritmo *RandomForest*. O modelo obteve 80,5% de acurácia (taxa de acerto), isto é, dada uma nova resposta, esta seria classificada erroneamente em apenas 19,5% dos casos. Por outro lado, o pior desempenho ficou com o algoritmo *RandomTree*, com 72,2% de taxa de acerto. Já o C4.5, principal algoritmo deste trabalho, teve como melhor desempenho a taxa de acerto de 77,78%.

O fator principal a ser analisado nestes resultados é a capacidade de generalização do modelo, pois, com o passar do tempo, a taxa de acerto se manteve estável mesmo considerando diferentes algoritmos. Esta é uma conquista importante pois trabalhos anteriores, como explicitado em seus resultados, não garantiam a generalização de seus modelos.

6.2.2. Experimento 2: Modelo Utilizando Somente as Características Previamente Existentes

O experimento 2 tem como finalidade gerar o modelo a ser utilizado na ferramenta *Stack Overflow Predictor* (Seção 6.3). A diferença em relação ao experimento 1 se dá pela utilização somente das características previamente conhecidas, isto é, aquelas que estão disponíveis no momento em que uma nova resposta está sendo escrita. Para tal, foram removidas as características: *score* da pergunta e quantidade de comentários existentes na pergunta. A Tabela 7 apresenta os resultados, novamente ordenados decrescentemente pela acurácia do classificador.

Tabela 7 – Resultados para o Modelo Utilizando Somente as Características Previamente Conhecidas

Base de Treinamento	Método/Base de Teste	Algoritmo	Accuracy	F-measure	Recall	Precision	AUC
Setembro de 2014	Março de 2016	<i>RandomForest</i>	0,783	0,848	0,875	0,822	0,875
Setembro de 2014	Março de 2016	DecisionTable	0,774	0,846	0,896	0,801	0,868
Setembro de 2014	Março de 2016	PART	0,76	0,842	0,921	0,774	0,82
Setembro de 2014	Março de 2016	C4.5	0,748	0,745	0,748	0,743	0,784
Setembro de 2014	Março de 2016	<i>RandomTree</i>	0,731	0,808	0,821	0,795	0,68
Setembro de 2014	Março de 2016	<i>DecisionStump</i>	0,726	0,752	0,603	1	0,801

Neste experimento, o melhor resultado foi de 78,3% considerando a acurácia, obtido pelo algoritmo *RandomForest*. O pior resultado (72,6%) ficou com o algoritmo *DecisionStump*. Considerando o modelo construído pelo algoritmo C4.5, o resultado atingiu 74,8% de acurácia.

Visando fazer um comparativo entre os dois primeiros experimentos um teste estatístico *t-student* foi executado para os resultados provenientes do algoritmo C4.5, para determinar se os conjuntos de características (todas e somente as preexistentes) apresentam média de acurácia igual (hipótese nula) ou diferente (hipótese alternativa). Primeiramente foi executado um teste *F* para verificar que a variância para todas as características (0,000166) e somente as preexistentes (0,00332) não diferem significativamente ($p < 0,05$), desta forma foi utilizado o teste *t-student* para variâncias iguais. A significância utilizada foi de 5% (0,05) para $p = 0,00259$, desta forma rejeita-se a hipótese nula e aceita-se a alternativa. Em outras palavras, a média de acurácia para o algoritmo C4.5 considerando todos os atributos é significativamente maior que a média considerando apenas os atributos preexistentes.

6.2.3. Experimento 3: Análise do Melhor Modelo Gerado pelo Algoritmo C4.5

O experimento 3, busca avaliar mais a fundo o modelo construído pelo algoritmo C4.5 no experimento 2 (74,8% de acurácia). Os resultados aqui apresentados visam verificar se a remoção de alguma característica do conjunto modifica a sua acurácia. Para tal, foram construídos quatro grupos. O primeiro contempla todas as características previamente existentes. Os demais grupos são: todas as características exceto a presença de blocos de

código, todas as características menos as que se referem ao usuário (questionador e quem respondeu) e, por fim, todas as características menos o tempo (diferença em segundos entre a pergunta e a resposta). A Tabela 8 exibe os resultados obtidos.

Tabela 8 – Resultados da Análise do Melhor Modelo para o Algoritmo C4.5

Configuração	Accuracy	% Decréscimo na Acurácia	F-measure	Recall	Precision	AUC
Todas as Características	0,748		0,745	0,748	0,743	0,784
- Bloco de Código	0,7361	-1,59%	0,733	0,736	0,731	0,774
- Características do Usuário	0,748	0,00%	0,746	0,748	0,745	0,816
- Tempo	0,7315	-2,20%	0,732	0,732	0,732	0,789

Ao excluir a presença de blocos de código o modelo teve um decréscimo de 1,59% em relação ao conjunto contemplando todas as características. Já a remoção das características do usuário (reputação, votos positivos, votos negativos, entre outras) não alterou a acurácia do modelo, colocando em evidência que, não necessariamente o usuário precisa possuir uma reputação alta no fórum para prover uma resposta de qualidade. Por fim, a remoção da característica tempo resultou em um decréscimo de 2,2% na acurácia do classificador, revelando-se uma característica importante para o modelo.

6.2.4. Comparativo com o Estado da Arte

Neste ponto da discussão, é necessário apresentar um comparativo com o estado da arte. Como previamente apresentado no capítulo 3, Trabalhos Relacionados, apenas um artigo (publicado em 2015) com o objetivo de prever a qualidade de respostas em fóruns de desenvolvedores pode ser utilizado para realizar uma comparação direta com os resultados apresentados neste trabalho. A Tabela 9 apresenta os resultados obtidos pelos autores. É importante destacar que a métrica de avaliação utilizada foi a área abaixo da curva (AUC) ao contrário da proposta neste trabalho, onde foi utilizado a acurácia.

Tabela 9 – Resultado do Artigo *Mining Successful Answers in Stack Overflow*

Fonte: (CALEFATO; LANUBILE; MARASCIULO, 2015)

Configuração	AUC	%	F-measure	Recall	Precision
Todas as Características	0,6498	Decréscimo na AUC	0,6043	0,7064	0,6479
- Características do Usuário	0,6371	-1,96%	0,5874	0,7061	0,6216
- Tempo	0,5912	-9,03%	0,5984	0,7056	0,6394

Os resultados apresentados foram obtidos para modelos avaliados com o método *Holdout* (sendo 2/3 para treinamento e 1/3 para teste). Neste experimento em específico o modelo a ser comparado foi construído utilizando a mesma base de dados utilizada pelos pesquisadores, avaliadas pelo mesmo método, porém considerando os atributos extraídos nesta pesquisa. Desta forma, foi realizada uma comparação direta entre os resultados. A Tabela 10 apresenta esses resultados.

Tabela 10 – Comparativo: Estado da Arte vs. Resultados Obtidos

Configuração	AUC Estado da Arte	AUC Resultados Obtidos	Diferença (%)
Todas as Características	0,650	0,731	+11,08%
- Características do Usuário	0,637	0,784	+18,75%
- Tempo	0,591	0,758	+22,03%
Média	0,626	0,757	+17,28%

Como exposto, em média, o modelo apresentado neste trabalho é 17,28% mais eficiente que o modelo disponível no estado da arte.

6.3. *Stack Overflow Predictor*

A ferramenta, apelidada *Stack Overflow Predictor*, foi criada para apresentar na prática os conceitos e resultados obtidos neste trabalho. A ideia é prover aos usuários do *Stack Overflow* um auxílio no momento de redigir uma nova resposta. A aplicação fornece como resultado se a resposta que está sendo escrita será ou não aceita pelo questionador, bem como sua respectiva probabilidade. Além disso, a aplicação exibe como o resultado foi obtido. A seção 6.3.1 apresenta o modelo preditivo utilizado e a seção 6.3.2, uma breve explicação

sobre a ferramenta. Detalhes técnicos sobre a sua construção e um exemplo de funcionamento podem ser vistos no Apêndice A.

6.3.1. Modelo Preditivo Utilizado

O modelo selecionado para ser integrado à ferramenta foi gerado pelo algoritmo C4.5, pois, como justificado ao longo deste trabalho, o modelo gerado por esse algoritmo pode ser transformado em regras do tipo *se* → *então*. Os modelos com melhor acurácia poderiam ser utilizados, mas a ferramenta não seria capaz de exibir como o resultado foi obtido. Assim sendo, o modelo do algoritmo C4.5 com 74,80% de taxa de acerto, exibido no experimento 2, foi o escolhido para ser integrado à ferramenta. É importante destacar que a acurácia a modelo se manteve estável com o passar do tempo, o que implica em uma não necessidade de atualização constante, embora, sempre que o *Stack Overflow* disponibilizar um novo *data dump*, seja recomendável a sua atualização.

6.3.2. A Ferramenta

A ferramenta é um *website* que utiliza os dados públicos do fórum *Stack Overflow* para o seu funcionamento. O usuário necessita de apenas duas informações para utilizar a aplicação, seu número identificador facilmente obtido em seu perfil e o número identificador da pergunta que está interessado em responder. Essas informações são necessárias para a obtenção das características do usuário, bem como as características da pergunta. Posteriormente basta o usuário escrever sua resposta, como se estivesse escrevendo no fórum original e, em seguida, submetê-la ao teste. Na sequência o resultado é exibido com o seguinte conteúdo: atributo binário se a resposta será ou não selecionada pelo questionador, a probabilidade de isto ocorrer segundo o modelo e a regra utilizada para chegar a esse resultado. A Figura 22 exibe a aplicação em funcionamento para uma resposta sem qualidade, bem como o resultado e os seus respectivos atributos anteriormente citados. Ainda, no canto direito, uma lista de perguntas não respondidas é disponibilizada para estimular o usuário a, possivelmente, responde-las.

Stack Overflow Predictor
Papers About Contact

Welcome to Stack Overflow Predictor

Only two informations are required in order to make the correct use of this application, the **id of your user** on Stack Overflow and the **question's id** that you are interested in answering. The data used in this application is public and obtained by Stack Exchange API, so it is not required any kind of authentication.

User

ID:

Question

ID:

Validação com bootstrapValidator tendo conflito com maskedinput

Estou tendo um problema com um formulario onde no campo que recebe dados do telefone do cliente, há um conflito entre as bibliotecas bootstrapValidator e jquery.maskedinput. Simplesmente, quando preencho o campo telefone, ele reconhece aqueles dados já com a máscara do telefone como inválido. Segue abaixo o codigo que utilizo para validar com o bootstrapValidator.

```
$(document).ready(function() {
  $('#pesquisadorForm').bootstrapValidator({
    message: 'This value is not valid',
    // live: 'disabled',
    feedbackicons: {
```

Fail!

According to the model your answer **won't be accepted** by the questioner. The precision of the model for this case is **100%**.

Rule: IF questionScore <= 4 AND answerBodySize <= 613 AND answererReputation <= 23 AND questionerUpVotes <= 1 AND answererBronzeMedals <= 2 THEN answerNotAccepted

Answer:

Unanswered Questions

- [85803] Minha imagem não aparece no ImageView
- [147131] Validação com bootstrapValidator tendo conflito com maskedinput
- [147128] Formatar data escrita para o padrão americano PHP
- [147111] Fechar todas as Activities
- [105499] Manipulando com Update Panel
- [147120] Aplicação não cria arquivo na memória externa
- [147126] Mover em massa arquivos .XYZ em X diretórios acima
- [147123] Como usar o comando "print" para mostrar variáveis em string junto com inteiros em MATLAB?
- [147124] Xamarin.forms configuração
- [133905] Ler arquivo txt sempre que houver alterações
- [146991] Criando um ARPING
- [147121] Problema com criação dinâmica de elementos
- [138955] Remover imagem pdf usando itext
- [147122] Meteor com problemas
- [147091] Anexar arquivo usando WebBrowser Delphi

Figura 22 – A Ferramenta *Stack Overflow Predictor*

6.4. Conclusão

Neste capítulo foram apresentados os resultados obtidos neste trabalho. A análise quantitativa reforçou a afirmativa que fóruns de perguntas e respostas possuem um déficit na qualidade de seu conteúdo, onde 23% das respostas fornecidas possuem um score negativo. Além disso, a presença de blocos de código nas respostas é uma característica predominante do fórum. Ainda sobre a análise quantitativa, viu-se que cerca de 60% das perguntas são respondidas nas primeiras 48 horas.

Através dos resultados dos experimentos é possível levantar alguns pontos principais. Primeiramente, é possível perceber que as características dos usuários (responsáveis pela pergunta e resposta) diminuíram a acurácia do classificador. Essa situação pode ser visualizada no seguinte contexto: um usuário não precisa necessariamente ter uma grande reputação para fornecer uma resposta de qualidade. É possível que ele seja um usuário novato no fórum, mas que tenha uma boa experiência na vida real. Alguns autores (JEON; CROFT; LEE, 2006) apontavam as características do usuário como as mais importantes para a construção do modelo, ao contrário do que foi visualizado nos resultados obtidos. Além disso, a remoção de atributos como a presença de código na resposta e tempo apresentou um pequeno decréscimo na acurácia do classificador. O destaque, para os resultados obtidos fica por conta da capacidade de generalização dos modelos construídos. Com o passar o tempo os modelos se mantiveram estáveis, se analisado as suas respectivas acurácias. Isto nos leva a afirmar que, um modelo construído com uma base histórica mais antiga vai ser capaz de prever a qualidade de respostas corretamente em um ponto no futuro.

Com isso, conclui-se que é possível construir um modelo capaz de prever a qualidade das respostas em fóruns de perguntas e respostas para desenvolvedores com uma acurácia superior ao estado da arte.

Capítulo 7

Conclusão e Trabalhos Futuros

Fórum de perguntas e respostas são fontes públicas de conhecimento compartilhado. Como amplamente discutido ao longo deste trabalho, devido a grande quantidade de dados que são inseridas diariamente, estes fóruns carecem de processos automáticos capazes de desonerar os moderadores de tarefas como: eliminação de spam, remoção de perguntas duplicadas, avaliação da qualidade de respostas, entre outros. Todos esses tópicos geram o problema principal desses *websites*, a variação de qualidade. Neste cenário, alguns autores propuseram a criação de alguns métodos para a resolução destes problemas. Entretanto, na área que se refere qualidade de respostas foi identificada uma lacuna em relação aos resultados obtidos. Primeiramente é importante lembrar que apenas um trabalho (CALEFATO; LANUBILE; MARASCIULO, 2015) se propôs a avaliar a qualidade de respostas em fóruns de perguntas e respostas para desenvolvedores, os demais autores focaram em fóruns gerais que possuem componentes diferentes.

Desta forma, este trabalho teve por objetivo a construção de um método capaz de prever a qualidade de respostas em fóruns de perguntas e respostas através de características quantitativas. Para tal, foram utilizados algoritmos da área de Aprendizagem de Máquina, mais especificadamente, árvores de decisão. Os modelos gerados foram separados em dois conjuntos. Primeiramente analisou-se a eficiência dos modelos utilizando todas as 39 características extraídas, e obteve-se, como melhor resultado, uma acurácia de 80,5%. Porém, como o objetivo é prever a qualidade das respostas, foram eliminadas as características que não estão disponíveis no momento da escrita da resposta pois não é possível determinar seus valores antes da sua respectiva publicação. Vale ressaltar que os modelos construídos considerando as 39 características podem ser utilizados para avaliar as respostas já existentes

no fórum. Em seguida, valendo-se somente conjunto de características conhecido antes da resposta ser publicada, também foram construídos diversos modelos. O melhor resultado neste caso foi de 78,3% acurácia. Todavia, o algoritmo gerador deste modelo não possui uma saída capaz de ser interpretada por um humano. Então, o C4.5 foi adotado neste trabalho como o algoritmo principal. Para avaliar a qualidade do modelo gerado, foram comparados os resultados obtidos neste trabalho com o único trabalho similar encontrado no estado da arte e, provou-se que o modelo proposto neste trabalho obteve uma eficiência, em média, 17,28% maior. Com isso, prova-se a hipótese que é possível a construção de um método capaz de prever a qualidade de respostas em fóruns de perguntas e respostas de desenvolvedores. Além de todas essas conquistas, uma ferramenta para a predição da qualidade de respostas foi construída e encontra-se disponível para ser utilizada por toda a comunidade.

Como trabalhos futuros, pode-se (i) coletar as respostas que se valeram do uso desta nova ferramenta e validar, após um espaço de tempo, se as mesmas foram selecionadas pelos questionadores. Desta forma seria possível provar a hipótese que a ferramenta ajuda o usuário a fornecer respostas mais completas e, conseqüentemente, o número de perguntas com respostas aceitas aumenta. Além disso, pode-se (ii) adotar outro conceito de respostas de qualidade e confrontar os resultados com os expostos neste trabalho, pois no *Stack Overflow* o conceito de resposta aceita e melhor resposta são diferentes. A resposta aceita é a aprovada pelo questionador, enquanto a melhor resposta, teoricamente, é a mais votada por todos os usuários. Este trabalho utilizou a resposta aceita como o conceito de resposta de qualidade, pois partiu-se do conceito que o questionador tem a melhor capacidade de julgar qual resposta solucionou seu problema. Entretanto, a resposta mais votada pode ser melhor uma vez que o questionador pode selecionar uma resposta erroneamente. Por fim, ainda há espaço para (iii) avaliar a importância de novas características, como por exemplo, incrementar a característica referente à presença ou não de código na resposta com a qualidade do bloco de código (compilável, erro de sintaxe, etc.).

Referências Bibliográficas

- AGHAEI, S. Evolution of the World Wide Web: From Web 1.0 to Web 4.0. *International journal of Web & Semantic Technology*, v. 3, n. 1, p. 1–10, 31 jan. 2012.
- AGICHTEIN, E.; CASTILLO, C.; DONATO, D. Finding High-quality Content in Social Media. *International Conference on Web Search and Data Mining*, p. 183–193, 2008.
- ANDERSON, A.; HUTTENLOCHER, D.; KLEINBERG, J. Discovering Value from Community Activity on Focused Question Answering Sites : A Case Study of Stack Overflow. *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, p. 850–858, 2012.
- ARAI, K. Predicting Quality of Answer in Collaborative Question Answer Learning. *International Journal of Advanced Research in Artificial Intelligence*, v. 3, n. 2, p. 23–26, 2014.
- ASADUZZAMAN, M.; MASHIYAT, A. S.; ROY, C. K. Answering questions about unanswered questions of Stack Overflow. *2013 10th Working Conference on Mining Software Repositories (MSR)*, p. 97–100, 2013.
- BOSU, A.; CORLEY, C.; HEATON, D. Building reputation in StackOverflow: An empirical investigation. *IEEE International Working Conference on Mining Software Repositories*, p. 89–92, 2013.
- CALEFATO, F.; LANUBILE, F.; MARASCIULO, M. C. Mining Successful Answers in Stack Overflow. *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*, p. 430–433, 2015.

- CHOU DHURY, N. World Wide Web and Its Journey from Web 1 . 0 to Web 4.0. International Journal of Computer Science and Information Technologies (IJCSIT), v. 5, n. 6, p. 8096–8100, 2014.
- DE SOUZA, L.; CAMPOS, E.; MAIA, M. Ranking Crowd Knowledge to Assist Software Development. Proceedings of the 22Nd International Conference on Program Comprehension, p. 72–82, 2014.
- DUIJN, M.; KU, A.; BACCHELLI, A. Quality Questions Need Quality Code : Classifying Code Fragments on Stack Overflow. The 12th Working Conference on Mining Software Repositories, 2015.
- GINSCA, A. L.; POPESCU, A. User profiling for answer quality assessment in Q&A communities. Proceedings of the 2103 workshop on Data-driven user behavioral modelling and mining from social media - DUBMOD '13, p. 25–28, 2013.
- GKOTSIS, G.; STEPANYAN, K.; PEDRINACI, C. It's all in the content: state of the art best answer prediction based on discretisation of shallow linguistic features. Proceedings of the 2014 ACM Conference on Web Science (WebSci), p. 202–210, 2014.
- HALL, M.; FRANK, E.; HOLMES, G. The WEKA data mining software: An update. SIGKDD Explorations, v. 11, n. 1, p. 10–18, 2009.
- HANDAYANI, A. N. Predicting Quality of Answer in Collaborative Q&A Community. v. 2, n. 3, p. 9–13, 2013.
- HART, K.; SARMA, A. Perceptions of answer quality in an online technical question and answer forum. Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering - CHASE 2014, p. 103–106, 2 jun. 2014.
- IHLWAN, M.; WOYKE, E.; ELGIN, B. NHN: The little search engine that could.

- JEON, J.; CROFT, W. B.; LEE, J. H. A framework to predict the quality of answers with non-textual features. Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval SIGIR 06, p. 228, 2006.
- LEE, J.; SONG, Y.; RIM, H. Predicting the Quality of Answers Using Surface Linguistic Features. Proceedings of the Sixth International Conference on Advanced Language Processing and Web Information Technology (ALPIT 2007), p. 111–116, 2007.
- LEZINA, G.; KUZNETSOV, A. Predict Closed Questions on StackOverflow. In Proceedings of the Ninth Spring Researcher's Colloquium on Database and Information Systems CEUR-WS, v. 1031, p. 10–14, 2013.
- LI, B.; JIN, T.; LYU, M. Analyzing and predicting question quality in community question answering services. Proceedings of the 21st international conference companion on World Wide Web - WWW '12 Companion, p. 775–782, 2012.
- LIU, B.; FENG, J.; LIU, M. Predicting the quality of user-generated answers using co-training in community-based question answering portals. Pattern Recognition Letters, v. 58, p. 29–34, 2015.
- LIU, Q.; AGICHTEIN, E.; DROR, G. Predicting web searcher satisfaction with existing community-based answers. Proceedings of the 34th international ACM SIGIR conference on Research and development in Information - SIGIR '11, p. 415, 2011.
- LIU, Y.; BIAN, J.; AGICHTEIN, E. Predicting information seeker satisfaction in community question answering. Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '08, n. 2, p. 483, 2008.
- MAMYKINA, L.; MANOIM, B.; MITTAL, M. Design Lessons from the Fastest Q&A Site in the West. Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11, p. 2857, 2011.

- MILANI, C. S.; PARAISO, E. C. Predição da Qualidade de Respostas em Fóruns de Desenvolvedores Utilizando Mineração de Dados. ENIAC, p. 8, 2015.
- MITCHELL, T. Machine Learning. 1. ed. [s.l.] McGraw-Hill Science, 1997.
- MORRIS, M. R.; TEEVAN, J.; PANOVICH, K. What Do People Ask Their Social Networks, and Why? *Chi*, v. 69, n. 1, p. 1739, 2010.
- MOVSHOVITZ-ATTIAS, D.; COHEN, W. Natural Language Models for Predicting Programming Comments. *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, p. 35–40, 2013.
- PAL, A.; CHANG, S.; KONSTAN, J. A. Evolution of Experts in Question Answering Communities. *Proceedings of the Sixth International AAAI Conference on Weblogs and Social Media*, p. 1–8, 2012.
- PAL, A.; FARZAN, R.; KONSTAN, J. Early detection of potential experts in question answering communities. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, p. 231–242, 2011.
- PERA, M. S.; NG, Y. K. A community question-answering refinement system. *Proceedings of the 22nd ACM conference on Hypertext and hypermedia - HT '11*, n. Abril, p. 251–260, 2011.
- PONZANELLI, L.; BAVOTA, G.; PENTA, M. DI. Prompter: A Self-Confident Recommender System. *International Conference on Software Maintenance and Evolution*, p. 577–580, 2014.

- PONZANELLI, L.; MOCCI, A.; BACCHELLI, A. Improving Low Quality Stack Overflow Post Detection. 2014 IEEE International Conference on Software Maintenance and Evolution, p. 541–544, 1 set. 2014.
- QUINLAN, J. R. C4.5: programs for machine learning. mar. 1993.
- RABAN, D.; HARPER, F. Motivations for answering questions online. *New Media and Innovative Technologies*, 2008.
- REZENDE, S. O. *Sistemas Inteligentes: Fundamentos e Aplicações*. [s.l.] Manole, 2003.
- SHAH, C.; ST, H.; BRUNSWICK, N. Evaluating and Predicting Answer Quality in Community QA. *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, n. March 2008, p. 411–418, 2010.
- TOBA, H.; MING, Z. Y.; ADRIANI, M. Discovering high quality answers in community question answering archives using a hierarchy of classifiers. *Information Sciences*, v. 261, p. 101–115, 2014.
- TREUDE, C.; BARZILAY, O.; STOREY, M. How do programmers ask and answer questions on the web? 2011 33rd International Conference on Software Engineering (ICSE), p. 804–807, 2011.
- WITTEN, I.; FRANK, E.; HALL, M. *Data Mining: Practical Machine Learning Tools and Techniques*. 3. ed. [s.l.: s.n.].
- WONG, E.; YANG, J.; TAN, L. AutoComment: Mining question and answer sites for automatic comment generation. 2013 28th IEEE/ACM International Conference on Automated Software Engineering, ASE 2013 - Proceedings, p. 562–567, 2013.
- YANG, D.; ZHENG, H.; YAN, J. Semantic Social Network Analysis with Text Corpora. *Advances in Knowledge Discovery and Data Mining*, v. 7301, p. 493–504, 2012.

ZHU, Z.; BERNHARD, D.; GUREVYCH, I. A Multi-Dimensional Model for Assessing the Quality of Answers in Social Q&A Sites. Proceedings of 14th International Conference on Information Quality, v. 1, p. 264–265, 2009.

Apêndice A

Aplicação *Stack Overflow Predictor*

Neste apêndice é apresentado a aplicação, nomeada “*Stack Overflow Predictor*”, desenvolvida para demonstrar na prática o funcionamento do método proposto neste trabalho. A aplicação é um *website*, desenvolvido com o mais famoso framework para HTML, CSS e JS, o *Bootstrap*. Tornando-se assim o *website* responsivo, isto é, disponível para dispositivos móveis e *desktop*. Além disso, a aplicação utiliza a API pública do *Stack Overflow*, a *Stack Exchange API v2.2*, realizando assim as consultas necessárias na base de dados para o correto funcionamento da aplicação. A seção A.1 exhibe o fluxo da aplicação e a seção A.2 as telas e seu funcionamento.

A.1. Fluxo da Aplicação

O usuário da aplicação precisa de apenas duas informações para utilizá-la, o número identificador do seu usuário no fórum e o número identificador da pergunta que deseja solucionar. Esses valores são facilmente obtidos na página do usuário e na página da pergunta, respectivamente. Ao fornecer esses valores, a aplicação executa uma consulta via a API do *Stack Overflow* para obter o conjunto de características. Por exemplo, ao selecionar o usuário cujo número identificador é um, a aplicação solicita para a API as informações referentes a este usuário. Essas informações são transformadas nas características que foram utilizadas neste trabalho para prever a qualidade das respostas. Posteriormente, o usuário redige a sua resposta e submete-a para teste. Neste momento a aplicação extrai as características da resposta e, juntamente com as demais características já extraídas (usuário e pergunta), o vetor de características é construído. Finalmente, o vetor é testado contra o

modelo do algoritmo C4.5 apresentado neste trabalho e o resultado contém as seguintes informações: valor binário para identificar se a resposta vai ou não ser selecionada pelo questionador; a probabilidade para essa afirmação; e como o a aplicação chegou neste resultado (caminho percorrido na árvore de decisão). Caso a resposta não esteja adequada o usuário pode continuar testando até que obtenha sucesso. A seção a seguir apresenta o funcionamento da aplicação.

A.2. Telas da Aplicação e Funcionamento

Esta seção apresenta passo a passo o funcionamento da aplicação. A Figura 23 exibe a ferramenta no momento em que é aberta.

Welcome to Stack Overflow Predictor

Only two informations are required in order to make the correct use of this application, the **id of your user** on Stack Overflow and the **question's id** that you are interested in answering. The data used in this application is public and obtained by Stack Exchange API, so it is not required any kind of authentication.

User

ID:

Question

ID:

Question:

Answer:

Unanswered Questions

- [147148] PHP Mailer erro 344
- [147140] Update dinâmico com verificação de variáveis \$_POST
- [106291] duvida com ajax e angular
- [147145] PySide instalação
- [42293] Formulário de comentários não aparece no tema
- [147063] Notificações Toast c#
- [104854] Verificar se dado já existe no banco com codeigniter
- [147142] Enviando a requisição anterior, Backbone Events
- [111581] Listar Pastas e Subpastas de um FTP com C#
- [147139] Como carregar um jar externo em tempo de execução no Android?
- [147138] Como alterar a chave de registro responsável pelas tarefas agendadas do Windows?
- [104758] Capturar nome do usuário logado na máquina via Aplicação Web
- [147133] SQL não retorna todos os resultados
- [147131] Validação com bootstrapValidator tendo conflito com maskedinput
- [147132] Problema com inserção de TIMER na tabela MySQL usando JDBC

Figura 23 – *Stack Overflow Predictor*: Tela Inicial

Os campos “User” e “Question” são os valores que devem ser completados para que a ferramenta seja capaz de identificar as características. Ao selecionar um usuário e clicar em “Apply” as características são extraídas e exibidas no botão “Details”. A Figura 24 exibe esses detalhes.

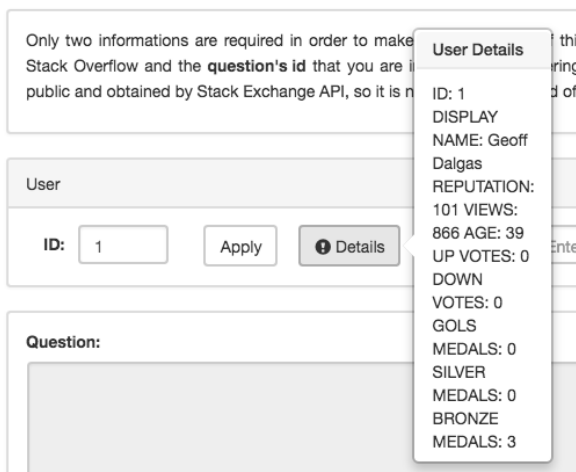


Figura 24 – Stack Overflow Predictor: Características do Usuário

Da mesma forma, as características da pergunta são extraídas. Neste caso o usuário pode optar em selecionar uma pergunta não respondida na listagem disponível ao lado direito da aplicação ou selecionar uma pergunta qualquer do fórum. A Figura 25 exibe o resultado desta ação.

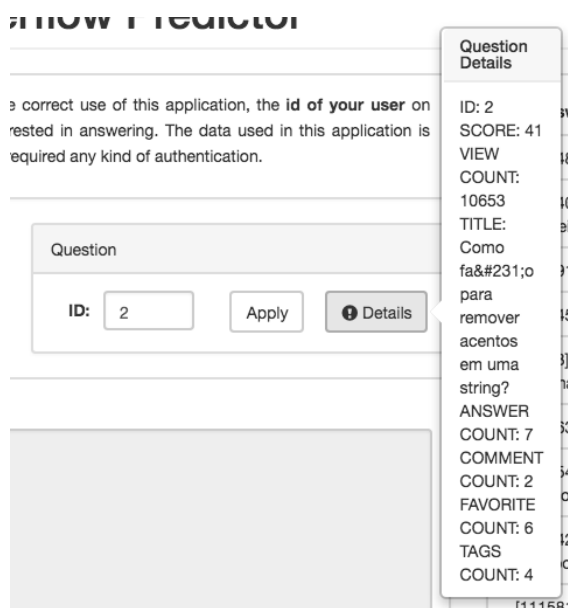
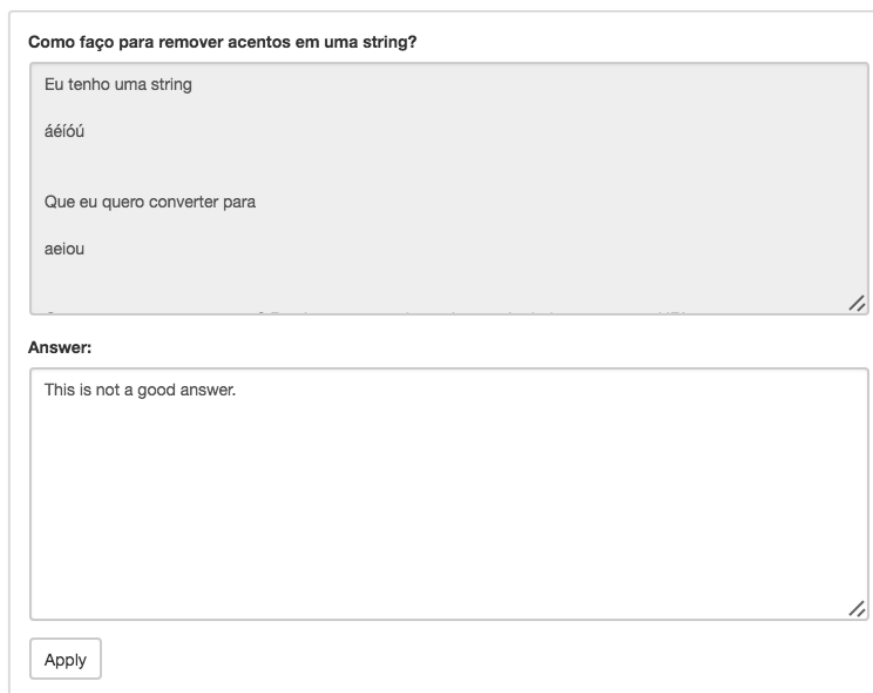


Figura 25 – Stack Overflow Predictor: Características da Pergunta

Então, o usuário deve redigir sua resposta, como se estivesse respondendo uma pergunta no fórum original e clicar em “Apply” na parte de baixo da aplicação. A Figura 26 ilustra essa ação.



The screenshot shows a web interface for the Stack Overflow Predictor. At the top, there is a question: "Como faço para remover acentos em uma string?". Below the question, there is a text area containing the text: "Eu tenho uma string", "áéíóú", "Que eu quero converter para", and "aeiou". Below the text area, there is a section labeled "Answer:" with a text area containing the text: "This is not a good answer.". At the bottom left of the interface, there is a button labeled "Apply".

Figura 26 – *Stack Overflow Predictor*: Respondendo uma Pergunta

Por fim, como a resposta fornecida como exemplo é de baixa qualidade, o modelo avisa o usuário que a resposta não será aceita pelo questionador, além de fornecer a probabilidade de esta ação ocorrer e o motivo. A Figura 27 exibe a mensagem em questão.

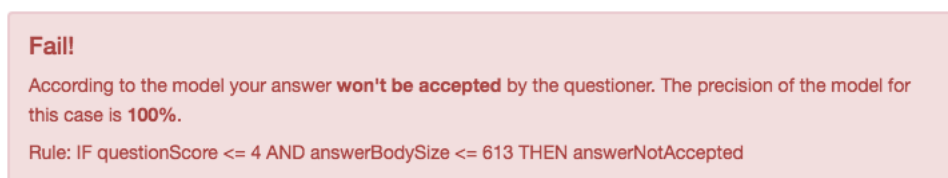


Figura 27 – *Stack Overflow Predictor*: Resultado