Universidade Estadual de Campinas
Institudo de Ciência da Computação
Disciplina Aprendizado de Máquina (MO444)
Prof. Dr. Jacques Wainer
Relatório Parcial do Projeto
Aluno: Luiz Alberto Ferreira Gomes - 007275

1. CONTEXTUALIZAÇÃO

Tíquetes (em inglês: *issues*) são empregados em *Bug Tracking Systems* (BTS) para registrar requisições de modificação para um determinado produto de software. Tais pedidos podem contemplar a inclusão ou modificação de requisitos, a correções de bugs encontrados e ou adaptações a uma nova plataforma operacional. Sendo que, cada um, no momento da sua criação, recebe uma prioridade que varia em escala de severidade (*blocker*, *critical*, *major*, *minor* ou *trivial*) que auxilia a equipe de manutenção no planejamento das suas futuras ações. Entretanto, no decorrer do tempo, essa prioridade pode mudar – algumas vezes – tanto para uma severidade maior como para uma menor – fazendo com que esse planejamento tenha que ser refeito inúmeras vezes.

2. OBJETIVOS DO PROJETO

Considerando o contexto acima, o objetivo do projeto da disciplina é construir uma máquina de aprendizado – baseada em um modelo de dados extraído de um BTS – que possa ser capaz de responder adequadamente às seguintes questões de pesquisa:

Q1. A prioridade de um dado tíquete mudará durante o seu ciclo de vida ?

Q2. A prioridade de um dado tíquete mudará para uma severidade mais alta durante o seu ciclo de vida?

Q3. Qual será a prioridade de um dado tíquete ao final do seu ciclo de vida (quando ele for fechado)?

3. ETAPAS DO PROJETO

O projeto descrito neste documento foi subdivido nas etapas apresentadas na Figura 1.

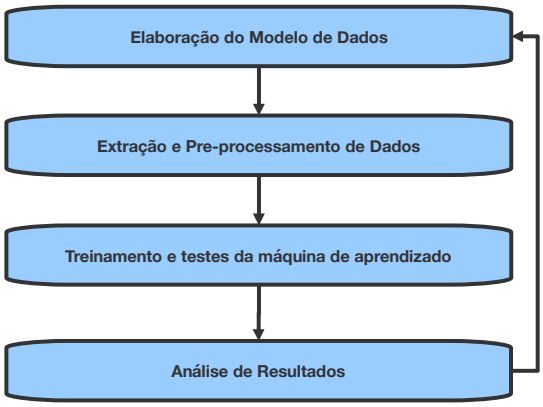

Figura 1: Etapas do projeto.

3.1 Elaboração do modelo de dados

A execução desta etapa resultou na construção do modelo de dados do projeto. Até o momento, ele contêm os campos apresentados abaixo na Tabela 1.

#	Nome	Descrição	Tipo	Escala
1	IssueKey	Identficador do tíquete (em inglês: issue).	Qualitativo	Nominal
2	Type	Tipo do tíquete: Bug, New Feature, Improvement, Task e Sub-task, e Wish.	Qualitativo	Nominal
3	Priority	Prioridade (ou severidade) do tíquete: Blocker, Critical, Major, Minor ou Trivial.	Qualitativo	Ordinal
4	DaysToResolve	Quantidade de dias para resolvê-lo.	Quantitativo discreto	Racional
5	Resolution	Tipo de resolução dada: Fixed, Duplicate, Unresolved, Won't Fix, Cannot Reproduce e Not a Problem.	Qualitativo	Nominal
6	Status	Situação do tíquete: Close, Resolved e Reopened.	Qualitativo	Nominal
7	QuantityOfComments	Quantidade de comentários feito sobre um tíquete.	Quantitativo discreto	Racional
8	HasBlockWord	Indica se a palavra "block" está na descrição do tíquete.	Qualitativo	Nominal
9	HasChangeWord	Indica se a palavra "change" está na descrição do tíquete.	Qualitativo	Nominal
10	HasDebugWord	Indica se a palavra "debug" está na descrição do tíquete.	Qualitativo	Nominal
11	HasErrorWord	Indica se a palavra "error" está na descrição do tíquete.	Qualitativo	Nominal
12	HasExceptionWord	Indica se a palavra "exception" está na descrição do tíquete.	Qualitativo	Nominal
13	HasFailWord	Indica se a palavra "fail" está na descrição do tíquete.	Qualitativo	Nominal
14	HasFixWord	Indica se a palavra "fix" está na descrição do tíquete.	Qualitativo	Nominal
15	HasIssueWord	Indica se a palavra "issue" está na descrição do tíquete.	Qualitativo	Nominal
16	HasPatchWord	Indica se a palavra "patch" está na descrição do tíquete.	Qualitativo	Nominal
17	HasProblemWord	Indica se a palavra "problem" está na descrição do tíquete.	Qualitativo	Nominal
18	HasSupportWord	Indica se a palavra "support" está na descrição do tíquete.	Qualitativo	Nominal
19	HasTestWord	Indica se a palavra "test" está na descrição do tíquete.	Qualitativo	Nominal
20	HasThrowsWord	Indica se a palavra "throws" está na descrição do tíquete.	Qualitativo	Nominal
21	HasWarnWord	Indica se a palavra "warn" está na descrição do tíquete.	Qualitativo	Nominal
23	DepthOfTree	Profundidade da árvore onde se encontra o tíquete.	Quantitativo discreto	Racional
24	QuantityOfChildren	Quantidade de filhos que o tíquete possui.	Quantitativo discreto	Racional
25	WeightOfChildren	Peso total ponderado dos filhos que o tíquete possui.	Quantitativo contínuo	Racional
26	QuantityOfParents	Quantidade de parentes que o tíquete possui.	Quantitativo discreto	Racional
27	WeightOfParents	Peso total ponderado dos parentes que o tíquete possui.	Quantitativo contínuo	Racional
29	ChangedPriority	Indica se houve mudança na prioridade do tíquete	Qualitativo	Nominal

Tabela 1: Modelo de dados.

A elaboração desse modelo baseou-se nas seguintes atividades:

1. Extração dos dados dos tíquetes diretamente da base do BTS. Os campos extraídos a partir dessa atividade são os campos de 1 a 7 da Tabela 1.

2. Mineração de palavras no texto do campo de descrição do texto com intuito de extrair as palavras mais frequentes relacionadas à manutenção de software. Os campos extraídos a partir dessa atividade são aqueles enumerados de 8 a 21 da Tabela 1.

3. Análise dos relacionamentos entre tíquetes extraídos do campo descrição. Os campos extraídos a partir dessa atividade são os campos de 23 a 27 na Tabela 1. Sendo que os campos 26 e 27 ainda apresentam problemas na sua extração.

4. Análise das mudanças registradas nos históricos de cada tíquete com intuído atribuir classes ao tíquetes para o treinamento e o teste do algoritmos de aprendizagem. O campo 28 na Tabela 1 define a classe de cada tíquete para a solução da questão 1 da pesquisa.

Em razão da natureza interativa e incremental do processo adotado para o projeto, o modelo dados apresentado acima deverá ser ajustado até a data da entrega do projeto para que os classificadores para responder às questões 2 e 3 da pesquisa, assim como melhorar os resultados para a questão 1 da pesquisa.

3.2 Extração e pre-processamento dos dados

A execução desta etapa resultou na geração de um arquivo no formato `csv` com os dados formatados e consistente para a máquina de aprendizado. Essa ocorreu a partir de um programa escrito na linguagem Java (`IssueDataExtractor.Java`) e de dois scripts na linguagem R (`IssueHistoryExtractor.R` e `IssueDataPreprocessor.R`) para extração e pré-processamento do dados. A responsabilidade do `IssueDataExtractor` é extrair os dados do tíquetes a partir do respositório remoto do software HADOOP – utilizado como repositório de testes para o projeto – que disponibiliza esses dados no formato XML. As responsabilidades do `IssueHistoryExtractor` são capturar o histórico de mudanças de um tíquete e extrair a partir tabelas as mudanças de prioridades ocorridas. Diferente dos dados dos tíquetes, as informações do histórico são disponibilizadas pelo BTS no HADOOP no formato HTML Por último, as responsabilidades `IssueDataPreProcessor` são: (1) integrar os dados básicos dos tíquetes com o histórico de mudanças de prioridade; (2) remover as colunas irrelevantes (como a coluna `IssueKey`, utilizada para integração com o histórico); (3) converter dados categóricos em númerico utilizando os métodos de conversão direta de categório ordinal para númerico ordinal (campo `Priority`) e o método *one-hot* (campos `Resolution` e `Status`) e (4) normalizar os dados.

3.3 Treinamento e testes das máquinas de aprendizado

Esta etapa resultou na escrita de máquina de aprendizado na linguagem R (`Predict-Q1.R`) para tratar a questão 1 (mais duas serão escritas até o final do projeto serão escritas para tratar as questões 2 e 3 da pesquisa), bem como em um arquivo no formato csv contendo as métricas para a avaliação de cada classificador utilizado. A máquina `Predict-Q 1` executou sobre os dados formatados os algoritmos *Gradiente Boosting Machine*, *Neural Networks*, *Random Forest* e *Support Vector Machine* e cada algoritmo recebeu amostras de 1000 a 6000 (com intervalos de 500 elementos) em três ciclos de execução, para cada método de amostragem utilizado: (1) randômico, a amostra é escolhida aleatoriamente; (2) proporcional, a proporção de cada classe é mantida na amostra; e (3) partes iguais, a amostra é dividida em partes iguais para cada classe. A escolha dos hiperparâmetros de cada algoritmo utilizou o processo de *cross-validation* para cada amostra tratada.

3.4 Análise dos resultados

Esta etapa resultou na escolha do classificador, do tamanho da amostra e do método da amostra mais adequados para responder à questão 1. Essa análise baseou-se nas métricas accurácia, precisão, *recall* e F1 *score* visualizadas a partir de gráficos gerados automaticamente pelo script `AMDataAnalyzer.R`. A avaliação preliminar dos classificadores da máquina de aprendizado da questão 1 mostrou que o melhor classificador, com o método de amostragem proportional e o tamanho da amostra de 6000 foi o SVM. Esse mesmo procedimento será feito para as questões 2 e 3, bem como ajustes nas etapas anteriores.

4. DIFICULDADES ENCONTRADAS

As dificuldades encontradas até momento durante o projeto foram as seguintes:

- O histórico de mudanças de um tíquete não está incluído nos arquivo XML contendo os dados básicos de um tíquete e, sim, em tabela em um arquivo HTML separadamente. Isso forçou a implementação de dois extratores independentes para a copia desses arquivos para o disco local.
- O construção das árvores contendo a relação entre os tíquetes teve que ser construída recursivamente a partir de hiperlinks, contidos na descrição do tíquete, que apontavam para outros tíquete. Percebeu-se que a estrutura de árvore não é a mais adequada para o problema em razão de existência de ciclos e de tíquetes com diversos pais. Pretende-se até o final do projeto migrar essa estrutura de dados para grafos.

- Definição de uma equação que englobe as métricas (acurácia, precisão, recall, F1 score) para aferir o classificador, com o método de amostragem e o tamanho da mostra mais adequados para cada questão de pesquisa que possa possibilitar a escolha automática pelo `AMDataAnalyzer.R`. Percebeu-se ainda que será necessário levar em consideração o tempo de processamento como variável na equação do indicador