

UTILIZANDO ALGORITMOS DE MACHINE LEARNING NA BASE DE DEFEITOS DE UM SISTEMA PARA IDENTIFICAÇÃO DE SUAS ÁREAS CRÍTICAS

Andre Luis Marques Rodrigues, Ismayle de Sousa Santos, Érica Miranda de Sousa, Bruno Sabóia Aragão, Rossana Maria de Castro Andrade

Durante o processo de desenvolvimento de um software é comum a ocorrência de defeitos. Seja em metodologias ágeis ou tradicionais, esses defeitos são documentados e registrados em ferramentas que permitem o gerenciamento e rastreamento dos mesmos, tal base de dados é rica em informações sobre o produto desenvolvido e processos utilizados. A análise desses dados pode fornecer uma ampla visão das características do projeto, suas fragilidades e como afetam a qualidade do produto. Tendo isso em vista, uma equipe da Fábrica de Testes do GREat está aplicando técnicas de aprendizado de máquina na base de dados de defeitos de um software, buscando identificar e classificar as áreas críticas do sistema, a fim de apoiar nas decisões da equipe de testes e no processo de evolução e manutenção do código de produção pelos desenvolvedores. A equipe em questão utiliza o Jira para reporte e gerenciamento de defeitos, os quais seguem um template pré-estabelecido, de onde é possível derivar informações importantes para análise, como descrição do defeito, versão do sistema em que foi identificado e funcionalidade associada. Tais informações foram exportadas em formato CSV a partir do Jira. Seguindo os princípios do KDD (Knowledge-Discovery in Databases), os dados coletados passaram por um processo de limpeza e remoção de dados inconsistentes. Para análise e clusterização de defeitos semelhantes, foi utilizado o algoritmo LDA - Alocação Latente de Dirichlet, um algoritmo não-supervisionado, que se caracteriza por separar os dados, com base na sua relevância, em diferentes tópicos criados. Com o auxílio desta técnica foi possível identificar quais defeitos foram recorrentes em determinada versão do sistema. Ao todo foi coletado um conjunto de 1045 registros de defeitos do software. Como resultados parciais dessa análise pode-se identificar que 22.48% dos defeitos concentram-se em duas funcionalidades básicas do sistema, somando um total de 235 defeitos.

Palavras-chave: TESTE DE SOFTWARE. CLUSTERIZAÇÃO DE DADOS. MACHINE LEARNING. APRENDIZADO NÃO-SUPERVISIONADO.