# Mini Weka

**Larissa Andrade[1], Rafael Castilho[1], Sergio Augusto[1]**

[1]Instituto de Computação – Universidade Federal do Amazonas (UFAM)
Av. General Rodrigo Octávio Jordão Ramos, 1200 - Coroado I, 69067-005
Manaus – Amazonas – Brasil

{las,rcc,sacbj}@icomp.ufam.edu.br

***Resumo.*** *Este relatório descreve as decisões e os formatos dos modelos escolhidos para o primeiro trabalho de Aprendizagem de Máquina e Mineração de Dados - Mini Weka. Neste também, serão descritas quaisquer modificações feitas para estender a especificação do trabalho, justificativas para restrições e Instruções claras de compilação e execução para este trabalho.*

## 1. As decisões de implementação dos indutores e Formato de representação dos modelos

Ao implementarmos, mapeamos todos os atributos para distribuição gaussiana, pois estamos classificando e é a forma mais simples para construir os modelos.

Foi usado o pickle para guardar uma instância treinada. Uma função que guarda a instância com o nome do modelo.sav no diretório modelos. Foi criada uma função que recebe o caminho até a uma instância salva e a retorna no notebook. O objetivo foi facilitar o armazenamento das instâncias.

Os modelos foram exibidos como:

- A representação do Naive bayes foi através de uma tabela com as probabilidades a priori e posteriori.
- A representação da Árvore de decisão foi uma árvore B.
- A representação do modelo de Regras utilizando o Prism foi uma lista de regras que foram aplicadas ao conjunto teste.
- A representação do k-nn foi uma matriz com todos os pontos de referência que serão os vizinhos.

### Formato de representação dos modelos

K-nearest neighbors (KNN) utiliza uma representação baseada em instâncias,

## 2. Modificações realizadas para estender o trabalho

- Fizemos uma investigação exploratória em uma das bases de dados para conhecimento prévio utilizando gráficos e diagramas para notarmos as possíveis distribuições dos dados no espaço e identificar novos padrões;
- Na validação dos dados utilizamos o 10-fold estratificado;

- Foi utilizado a matriz de confusão para expor os dados;
- Utilizamos métricas para validar os resultados dos modelos.

## 3. Justificativa para quaisquer restrições

- Como o modelo de Regras só poderia ser utilizados com atributos categóricos, utilizamos o Contact-lenses.arff para treinamento e validação do modelo.

## 4. Instruções para Uso deste trabalho

- Utilizar o Jupyter notebook para abrir o arquivo Mini Weka.ipynb;
- Rodar todas as pendências para a utilização desse notebook;
- Treinar a instância através do método fit.
- Após o treino, será possível usar o método predict recebe uma lista de exemplos não catalogados e as cataloga.

## Referências


CAMPOS, Raphael. Árvores de Decisão. [*S. l.*], 2017. Disponível em: https://medium.com/machine-learning-beyond-deep-learning/%C3%A1rvores-de-decis%C3%A3o-3f52f6420b69. Acesso em: 11 jun. 2019.

VANDERPLAS, Jake. Python Data Science Handbook. O'Reilly Media, 2016. Disponível em: https://jakevdp.github.io/PythonDataScienceHandbook/. Acesso em: 11 jun. 2019.

GIUSTI, Rafael. Slides de Aula Tópico 4: Classificação (Parte 2, atualizado 18/05). Disponivel em https://colabweb.ufam.edu.br/mod/resource/view.php?id=18696 . Acesso em: 11 jun. 2019.