

# Análise de classificadores

Aryane Ast dos Santos  
Departamento de Informática  
Universidade Federal do Paraná  
Email: aras10@inf.ufpr.br

**Resumo**—Abstract goes here  
**Index Terms**—IoT, Ontology, Semantics, SSN, OWL, OBOE, OpenIoT, SWEET, SUMO

## III. CONSIDERAÇÕES FINAIS

Todo o código utilizado no projeto, inclusos ..., pode ser encontrado num repositório Git hospedado no GitHub ...

## I. INTRODUÇÃO

Um problema de classificação consiste em definir um rótulo ou classe para um elemento a partir de um conjunto de elemento com rótulos definidos. Motivação por favor.

Este relatório se propõe a apresentar resultados obtidos com os classificadores KNN (K Nearest Neighbors), Naive Bayes e Árvores de Decisão num problema de classificação de imagens. A base utilizada contém 1901 imagens rotuladas em 9 classes diferentes. Os algoritmos de classificação não utilizam as imagens brutas, de forma que é necessário converter as imagens do formato JPG para vetores de características que os algoritmos de classificação possam utilizar, formando então uma base de treinamento, que denominei "completa", contendo o vetor de características que representa a imagem e seu rótulo respectivo.

Após ter o vetor de características, são realizados as execuções dos classificadores KNN, Naive Bayes e de Árvores de Decisão para todas as bases de treinamento e validação, e então o desempenho de cada classificador é apresentado.

Nas seções a seguir são apresentados maiores detalhes em como obter representações dos dados, os métodos utilizados, as métricas para desempenho, etc, etc.

## II. REPRESENTAÇÃO DOS DADOS

Para cada uma das imagens disponibilizadas para classificação, é realizada uma extração de características, no caso histograma de cores e contorno da imagem, why? Para a extração dos vetores de características, foi utilizada o script `create_bases.py`, que pode ser visualizado no código abaixo (ref das linhas x a y).

### A. Histograma de cores

Explicação

### B. Outro.. LBP?

Explicação

Após extração dos vetores de características, é realizada uma separação aleatória de todos os vetores de características em bases de treinamento e de validação, cada uma com 60% e 40% da completa, respectivamente. Essa separação é realizada com o script em Bash com o script `generate_train_test.sh`, que, essencialmente faz  
show code, baby