



UNIVERSIDADE ESTADUAL DE CAMPINAS  
INSTITUTO DE COMPUTAÇÃO



# k-Nearest Neighbors (kNN) e Análise de Componentes Principais (PCA) com JAX

Álgebra Linear e Otimização para Machine Learning  
Prof. Marcos M. Raimundo

## Descrição

Este trabalho prático tem como objetivo aprofundar seu conhecimento em algoritmos de aprendizado de máquina e técnicas de redução de dimensionalidade, explorando a implementação do kNN e das projeções (similar ao PCA) utilizando a biblioteca JAX. Você terá a oportunidade de implementar um kNN flexível, capaz de generalizar o produto interno, e investigar a relação entre as projeções e o kNN, explorando como a projeção pode ser incorporada ao algoritmo kNN. Dentre os objetivos teremos:

- Implementar o algoritmo kNN em JAX, permitindo a utilização de diferentes métricas de distância através da generalização do produto interno.
- Calcular a matriz de projeção usando SVD utilizando JAX.
- Analisar e discutir como a matriz de projeção pode ser incorporada ao algoritmo kNN, avaliando o impacto na performance e interpretabilidade do modelo.

## Tarefas

- Implementação do kNN:
  - Implemente uma função `knn(X_train, y_train, X_test, k, metric)` que realize a classificação kNN.
  - `X_train` e `y_train` representam os dados de treinamento, `X_test` os dados de teste, `k` o número de vizinhos e `metric` uma função que calcula a distância entre dois pontos (generalizando o produto interno).
  - Utilize JAX para otimizar a implementação e permitir a execução em GPU/TPU, se disponível.
- Cálculo da Matriz de Projeção:
  - Implemente uma função `pca(data, num_components)` que realize a Análise de Componentes Principais (PCA) em um conjunto de dados e retorne a matriz de projeção para as `num_components` principais componentes.
  - Utilize a decomposição em valores singulares (SVD) em JAX (`jax.scipy.linalg.svd`) para encontrar uma base ortonormal.
  - Escolha diferentes conjuntos de vetores para fazer uma base, verifique se eles formam uma projeção.

- Incorporação do de projeções no kNN:
  - Aplique projeções aos dados de treinamento e teste antes de executar o kNN.
  - Avalie o impacto da redução de dimensionalidade na performance do kNN (acurácia, tempo de execução).
  - Ao invés de escolher vetores arbitrariamente, escolha aqueles com os maiores autovetores. Discuta como a escolha do número de componentes de maiores autovetores afeta os resultados.

## Entregáveis

Um notebook Jupyter (ou similar) contendo:

- Implementações em JAX do kNN e do PCA.
- Experimentos comparando o kNN com e sem PCA no conjunto de dados indicado.
- Análise e discussão dos resultados, incluindo gráficos e tabelas relevantes.
- Conclusões sobre o impacto do PCA no kNN e considerações sobre a escolha do número de componentes.

O notebook base com a base de dados wine se encontra aqui: [colab](#)

## Crítérios de Avaliação

- Clareza do código e organização do notebook.
- Correção e eficiência das implementações.
- Qualidade da análise e discussão dos resultados.
- Profundidade da compreensão dos conceitos e técnicas exploradas.