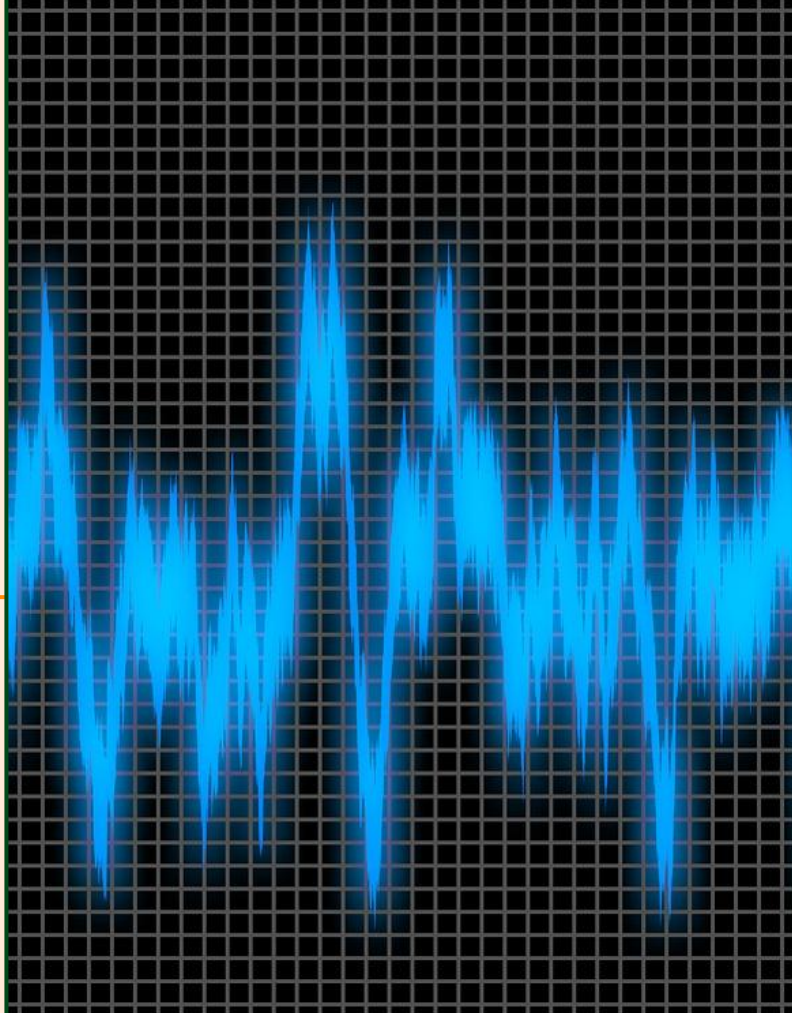




Inteligência Artificial

Projeto 2



Especificação Problema- *classificação*

N) Speaker accent recognition



Objetivo

Reconhecer e detectar o sotaque (*accent*) de nativos de 6 países diferentes (Espanha – ES, França – FR, Alemanha – GE, Itália – IT, Reino Unido – UK, Estados Unidos – US) ao dizerem uma única palavra em Inglês.



Dataset

Arquivos de áudio de 329 pessoas de um dos 6 países, com 1 segundo cada, em que uma única palavra em Inglês é enunciada.

Cada um dos arquivos de som foi decomposto em 12 **MFCC** (*Mel-frequency cepstral coefficients*) $\{X_1, X_2 \dots X_{12}\}$, que descrevem a onda de som.

Bibliografia

Wikipedia

Mel-frequency cepstrum, cross-validation, Training and test set, Decision Trees, Neural Networks, K-NN, SVM

https://en.wikipedia.org/wiki/Mel-frequency_cepstrum
[https://en.wikipedia.org/wiki/Cross-validation_\(statistics\)](https://en.wikipedia.org/wiki/Cross-validation_(statistics))
https://en.wikipedia.org/wiki/Training_validation_and_test_data_sets
https://en.wikipedia.org/wiki/Decision_tree
https://en.wikipedia.org/wiki/Artificial_neural_network
https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
https://en.wikipedia.org/wiki/Support_vector_machine

Scikit-learn

Exemplos, guias, documentação e comunidade da biblioteca

<https://scikit-learn.org/stable/>

Introduction to Data Mining, 2nd edition (2019)

Livro recomendado pelo Professor Regente.
Contém capítulos sobre todos os assuntos abordados no trabalho

Implementação

Linguagem

Python

Bibliotecas

Pandas: importação de dados de csv

Matplotlib: criação de gráficos

Scikit-Learn: machine learning

Algoritmos

- Decision Trees
- Neural Networks
- K-NN
- SVM

Métricas para comparação

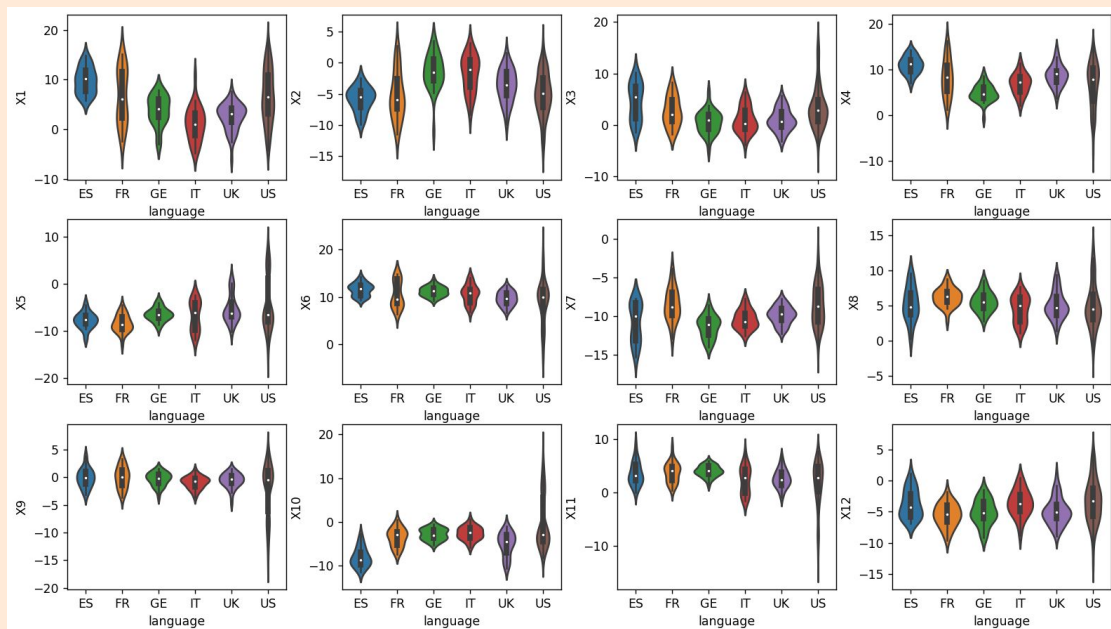
- Tempo dispendido a testar e treinar
- F-score
- Precision
- Recall
- Accuracy
- Confusion Matrix

Passos na implementação

1. Analisar dataset (detectar erros, NA, pré-processamento)
2. Definição dos training and test sets e Cross Validation
3. Implementação e parametrização dos diferentes algoritmos
4. Avaliação

Pré-processamento de Dados

No pré-processamento de dados, é de realçar que não havia valores incompletos, nulos, ou bastante discrepantes em relação ao resto dos dados. No entanto, havia uma maior presença do número de dados com a *label* US e UK, sendo que usamos uma função para normalizar os dados, eliminando valores aleatórios destas mesmas, de forma a termos um *dataset* bastante mais uniforme. Quanto à distribuição dos dados pelos 12 **MFCC**:



Modelos desenvolvidos

Exploramos 4 modelos diferentes:

- Support Vector Machines (SVM)
- k-Nearest Neighbors (KNN)
- Decision Trees
- Neural Networks

De forma a comparar as suas diferenças, avaliamos 6 parâmetros diferentes:

1. Accuracy
2. Precisão
3. Recall
4. F-Score
5. Training Time
6. Testing Time

Modelos Desenvolvidos

Support Vector Machines e k-Nearest Neighbors

SVM consiste em encontrar um hiperplano ótimo que separa os pontos dados de diferentes classes com a maior margem possível, sendo eficaz no tratamento de dados de elevada dimensão.

KNN é um algoritmo não paramétrico e *lazy*. O valor de uma nova instância é determinado pela média dos seus *k-neighbors* mais próximos no espaço de características.

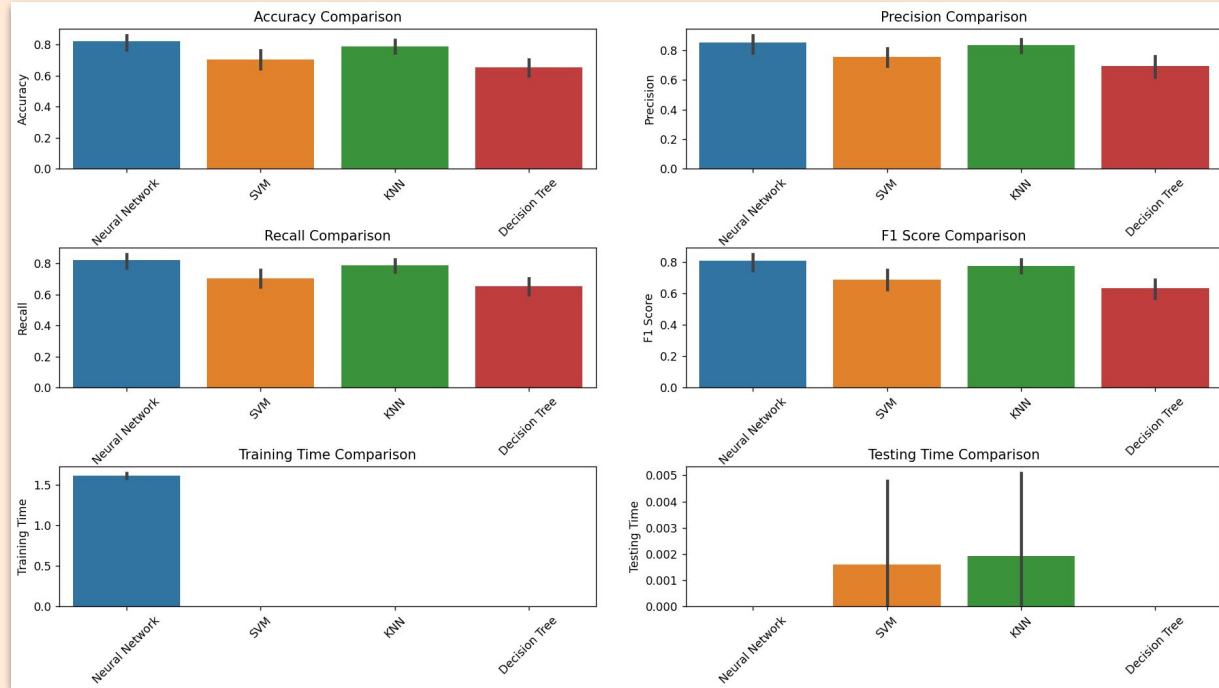
Modelos Desenvolvidos

Decision Tree e Neural Network

Decision Trees são estruturas hierárquicas compostas por nós internos, que representam testes de atributos, e nós de folhas, que são as *labels* (valores previstos). Dividem recursivamente o espaço com base nos valores dos atributos para formar regiões bastante próximas da variável-alvo.

Neural Networks são inspirados na estrutura e funcionamento dos neurónios biológicos. São constituídas por nós interligados, organizados em camadas, sendo que cada “neurónio” recebe entradas, aplica uma função de ativação, e produz uma saída. Estas aprendem padrões nos dados, ajustando os pesos das ligações entre “neurónios”.

Resultados



Average Accuracies:		
	Algorithm	Average Accuracy
0	Decision Tree	0.653922
1	KNN	0.787255
2	Neural Network	0.821242
3	SVM	0.703922

Conclusões

SVM tem uma boa performance em dados com um enorme número de amostras.

KNN é simples e fácil de implementar, mas computacionalmente dispendioso para grandes números de dados.

Decision Tree tem uma tendência a “treinar demais” os dados, especialmente quando se torna muito complexa e/ou profunda.

Neural Networks são eficientes, pois conseguem adaptar-se a diferentes problemas. No entanto gastam muito tempo e são também computacionalmente dispendiosos.

Em conclusão, a escolha do algoritmo depende do problema específico, da dimensão do conjunto de dados, dos requisitos de interpretabilidade, dos recursos computacionais e do compromisso entre exatidão e simplicidade. É sempre vantajoso experimentar vários algoritmos e comparar o seu desempenho antes de finalizar a escolha para uma determinada tarefa.