



Identificação da Marca do Carro

Alunos:

José Vitor R. Galbardi
Lucas Hideki Yamanaka
Diego Takahashi

RA:

109885
109883
109889



Introdução

- Este trabalho tem como objetivo utilizar aprendizagem de máquina para fazer a classificação de marca de carro a partir de uma imagem do carro.
- Técnica original: Rede neural convolucional (CNN).
- Técnicas utilizadas neste trabalho: KNN, SVM, Rede Neural - MLP.



Técnica Original

- O notebook python “Predict the car brand by CNN” pode ser encontrado no site kaggle.
- O autor busca classificar as imagens dos carros com suas respectivas marcas utilizando *Convolutional Neural Network* (CNN) .
- O autor utilizou um dataset com 60000+ imagens de carros com diversas marcas diferentes.
- Porém ele utilizou apenas 3 marcas sendo elas Chevrolet, Toyota e Ford.
- Número total de imagens de cada marca:

Chevrolet	5079
Toyota	4598
Ford	4416



Técnicas Utilizadas

- Diferente do autor foram feitos testes utilizando *Support Vector Machines* (SVM), *Nearest Neighbors* (KNN) e *Multi-layer Perceptron* (MLP) .
- Foi utilizado 4 marcas sendo elas Chevrolet, Toyota, Ford e BMW.
- Número total de imagens de cada marca:

Chevrolet	5079
Toyota	4598
Ford	4416
BMW	4121

- Apenas 4000 de cada classe foi utilizada, para manter o equilíbrio entre elas.

Exemplos de imagens





Feature engineering

- As imagens foram convertidas para grayscale.
- Tamanhos padronizados de 200x300.
- Foi utilizado os pixels da imagem como características.
- Alteração do tipo dos pixels de int para float.
- Normalização dos valores dos pixels de 0 ~ 255 para 0 ~ 1.
- Transformando as imagens em um vetor de características.

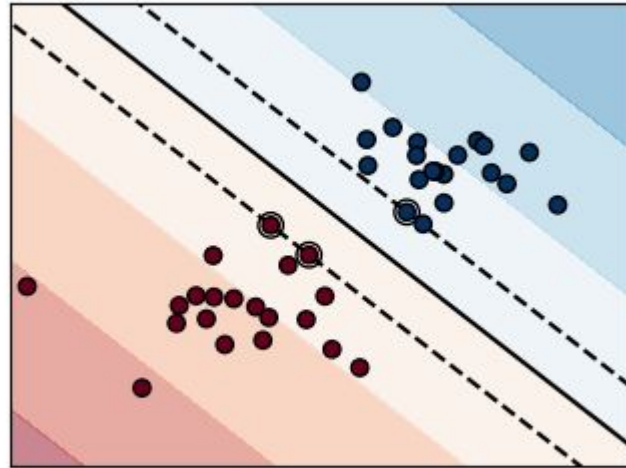


Holdout e cross validation

- Divisão Holdout
 - 70% treino e 30% teste
 - `random_state = 100`
 - `stratify`
- Cross validation
 - Stratified K-Folds
 - 5 folds

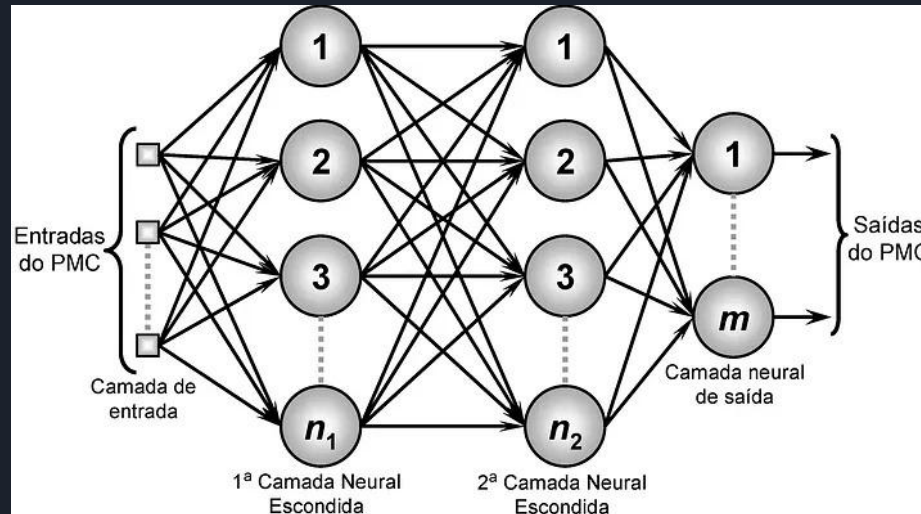
SVM - Support Vector Classification

- O objetivo de um SVM é encontrar o hiperplano de separação ideal o qual maximiza a margem da base de treinamento.
- Kernels testados:
 - rbf
 - poly
 - sigmoid
 - linear



Rede Neural - MLP

- Perceptron Multicamadas (Multi Layer Perceptron), é uma rede neural com uma ou mais camadas ocultas com um número indeterminado de neurônios.






KNN

(k-nearest neighbors)

- Aprendizagem baseada em exemplo/memória.
 - Não generaliza bem.
- A distância entre a instância e os K-vizinhos é calculada a partir de uma distância euclidiana.
- Os K-vizinhos mais próximos podem ser ponderados pela distância
 - Maior peso para os vizinhos mais próximos.

Comparação

Técnica	Acurácia (%)
CNN	76%
KNN(K = 3, ponderado)	71,2%
SVM (kernel = 'poly')	70%
SVM (kernel = 'linear')	64,6%
SVM (kernel = 'rbf')	64%
KNN(K = 3)	63,4%
MLP	59,6%
SVM (kernel = 'sigmoid')	22,9%



Comparação Stratified cross validation (folds = 5)

Técnica	Acurácia (%)
KNN(K = 3, ponderado)	73,27%
SVM (kernel = 'poly')	72,24%
MLP	60,7%



Referências

- **trabalho original:** <https://www.kaggle.com/code/jeongbinpark/predict-the-car-brand-by-cnn>
- <https://towardsdatascience.com/multilayer-perceptron-explained-with-a-real-life-example-and-python-code-sentiment-analysis-cb408ee93141>
- <https://www.simplilearn.com/tutorials/deep-learning-tutorial/multilayer-perceptron>
- https://keras.io/examples/vision/mlp_image_classification/
- <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>
- <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>