**BÁO CÁO PROJECT I**

**PHÂN LOẠI MÀU SẮC TRONG CÁC ĐIỀU KIỆN BÊN NGOÀI TÁC ĐỘNG SỬ DỤNG K-NEAREST-NEIGHBOR**

Người thực hiện : Lê Huy Hoàng - 20182535

Trong báo cáo này gồm có những vấn đề sau:

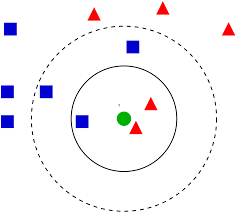
1. Thuật toán K-Nearest-Neighbor.
2. Các điều kiện bên ngoài tác động.
3. Accuracy khi phân loại màu sắc trong các điều kiện khác nhau.
4. Thuật toán KNN (K-nearest-neighbor)
5. Giới thiệu

+ KNN là 1 trong những thuật toán đơn giản và được sử dụng khá nhiều trong Machine Learning để giải quyết vấn đề phân loại (Classification) và hồi quy (Regression).

+ Đây là thuật toán lazy learning. Model không học được gì từ tập huấn luyện. Do đó để có thể dự đoán tốt mô hình dựa vào thuật toán KNN ta cần một lượng dữ liệu huấn luyện phải lớn ( học dựa trên bộ nhớ ).

+ Đối với bài toán Classification: đầu ra của dữ liệu mới dựa trên K điểm dữ liệu huấn luyện gần nhất hoặc có thể đánh trọng số đối với K điểm dữ liệu gần nhất để dự đoán.

+ Đối với bài toán Regression:  đầu ra của một điểm dữ liệu sẽ bằng chính đầu ra của điểm dữ liệu đã biết gần nhất (trong trường hợp K=1), hoặc là trung bình có trọng số của đầu ra của những điểm gần nhất, hoặc bằng một mối quan hệ dựa trên khoảng cách tới các điểm gần nhất đó.



**Hình 1: Dữ đoán điểm xanh lá cây sẽ gần với đỏ hay xanh da trời**

+ Tuy nhiên thuật toán khá là mẫn cảm với nhiễu dẫn đến dự đoán dữ liệu mới có thể không được tốt => Cần bước tiền xử lý dữ liệu (loại bỏ nhiễu) trước khi bắt đầu thực hiện KNN.

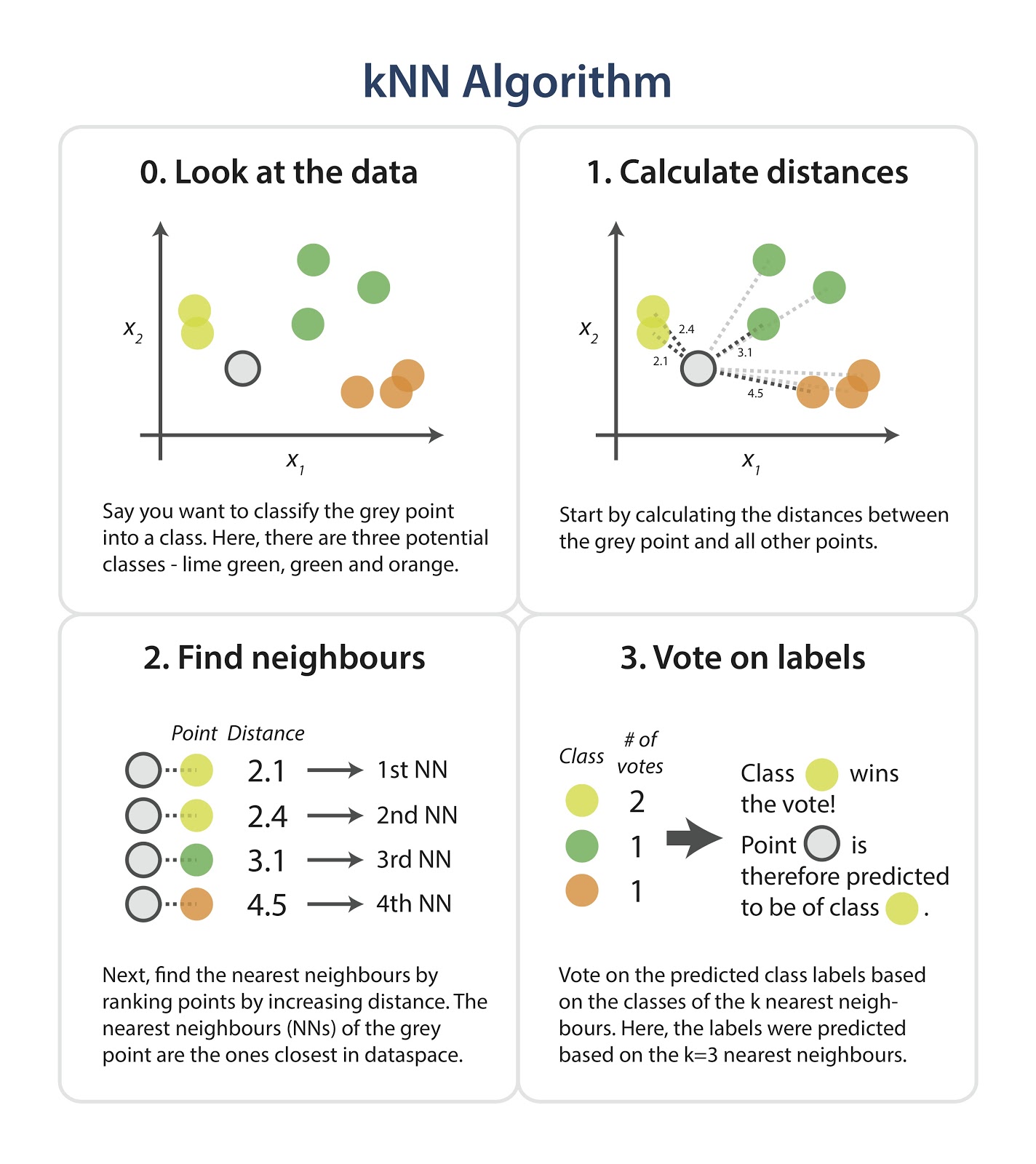
1. Các bước thực hiện thuật toán

+ Bước 1: Phân chia dữ liệu ( tập huấn luyện và tập kiểm tra , có thể cần đến tập kiểm thử).

+ Bước 2: Tính toán khoảng cách giữa các đặc trưng dữ liệu mới với từng dữ liệu điểm huấn luyện ( 1 điểm ở đây bao gồm n features và 1 label ).

+ Bước 3: Lấy K điểm gần với điểm dữ liệu mới.

+ Bước 4 : Dự đoán nhãn dựa trên K điểm huấn luyện ( nhãn nào có nhiều hơn trong K điểm thì khả năng cao nhãn dự đoán sẽ là nhãn đó ).



**Hình 2: Các bước thực hiện thuật toán**

1. Đánh giá accuracy của mô hình

Do bài toán đặt ra ở đây là phân loại màu sắc dựa trên KNN nên sẽ dùng accuracy cho bài toán phân loại.

Accuracy = (true\_predict\_test : Số dự đoán đúng trong tập test , test\_data : số dữ liệu tập test)

1. Ưu và nhược điểm của KNN

+ Ưu điểm:

* Thuật toán đơn giản, dễ hiểu cho người mới bắt đầu.
* Không học gì từ tập dữ liệu so với các thuật toán khác trong ML.
* Phù hợp với những bài toán không cần dữ liệu quá lớn (vừa đủ là được)
* Có thể giải quyết cả 2 bài toán là phân loại và hồi quy.

+ Nhược điểm:

* Lưu trữ rất nhiều khoảng cách do tập huấn luyện lớn => Có thể dẫn đến **tràn RAM.**
* Tính toán nhiều lần => Thời gian để ra được dự đoán đầu ra khá lâu nếu dữ liệu huấn luyện lớn.
* Không phù hợp với bài toán **REAL TIME.**

Ngoài ra để có thể dự đoán mất ít thời gian hơn ta nên sử dụng 2 thuật toán dựa trên cấu trúc dữ liệu và giải thuật là : KD tree và ball tree

1. Các điều kiện bên ngoài tác động đến màu sắc.

+ Trong các môi trường khác nhau như : tối, sang, bình thường thì có thể việc dự đoán màu có thể sẽ có kết quả không được tốt.

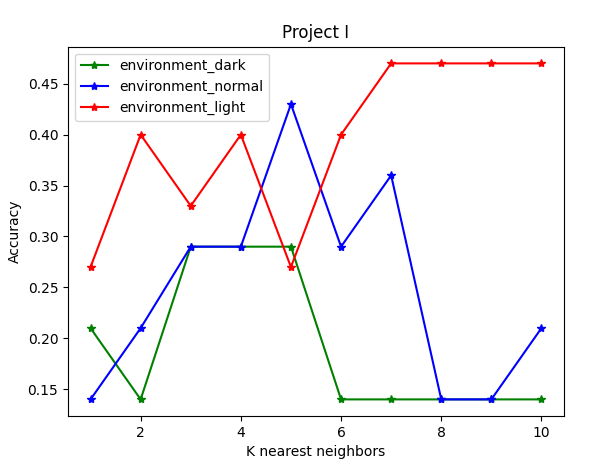
+ Nguyên nhân: có lẽ do việc mức xám trong bức ảnh thay đổi trong các môi trường khác nhau => dẫn đến các features trong mỗi điểm dữ liệu ( 1 bức ảnh ) có thể thay đổi nên khả năng dự đoán sai là có khả năng xảy ra.

1. Accuracy khi phân loại màu sắc trong các điều kiện khác nhau

+ Link\_data: <https://github.com/Nonamehp1234/Machine_Learning/tree/master/training_dataset>

+ Link\_code: <https://github.com/Nonamehp1234/Machine_Learning>

+ Chọn K từ 1 đến 10 để đưa ra accuracy cho từng K:



Nhận xét :

+ Ta thấy độ chính xác khi chụp ảnh ở ngoài trời có tỷ lệ chính xác cao hơn so với chụp ảnh trong điều kiện bình thường và tối.

+ Độ chính xác không cao có thể do tập ảnh train còn quá ít đã dẫn đến tập ảnh train không có đầy đủ các đặc trưng để dự đoán, ngoài ra còn yếu tố tác động như môi trường.

+ Nếu muốn tốt hơn về chính xác thì nên chọn màu áo, màu quần phải đồng với phong cảnh mình đang chụp.

**Tài liệu tham khảo**

<https://machinelearningcoban.com/2017/01/08/knn/>

<https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm>

<https://realpython.com/knn-python/#comparison-of-the-four-models>