**BÁO CÁO DỰ ÁN: PHÂN LOẠI LÁ CÂY Sử DỤNG MACHINE LEARNING**

**1. Giới Thiệu**

**1.1 Mục Tiêu**

- Dự án nhằm phát triển một hệ thống phân loại lá cây dựa trên ba nhóm đặc điểm:

* **Hình dáng** (shape)
* **Rìa lá** (margin)
* **Kết cấu** (texture)

- Sử dụng nhiều thuật toán machine learning khác nhau nhằm đánh giá và tìm ra mô hình hiệu quả nhất.

**1.2 Thuật Toán**

-Trong dự án này, sử dụng các thuật toán machine learning:

* **Naive Bayes**
* **Support Vector Machine (SVM)**
* **Logistic Regression**
* **k-Nearest Neighbors (k-NN)**
* **Linear Discriminant Analysis (LDA)**

- Bên cạnh đó, sử dụng một số thuật toán khác:

* **Chuẩn hóa dữ liệu** bằng StandardScaler.
* **Cross-validation** để cải thiện độ tổng quát.
* **Grid-search** để tìm tham số tối ưu.

**2. Dữ Liệu**

- Dữ liệu gồm **99 loài thực vật**, mỗi mẫu là một hình ảnh lá cây với các đặc điểm đã trích xuất.

* **Tập huấn luyện (train.csv)**: Chứa danh tính loài cây và các đặc trưng.
* **Tập kiểm tra (test.csv)**: Chứa đặc trưng của lá cây, không có nhãn.

- Dưới đây là một số dòng dữ liệu huấn luyện:

import pandas as pd

train\_raw = pd.read\_csv('train.csv')

train\_raw.head()

**3. Tiền Xử Lý Dữ Liệu**

**3.1 Mã hóa Nhãn (Label Encoding)**

Sử dụng LabelEncoder để chứng minh danh tính của loài bằng số:

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder().fit(train\_raw.species)

labels = le.transform(train\_raw.species)

classes = list(le.classes\_)

**3.2 Chuẩn Hóa Dữ Liệu**

Sử dụng StandardScaler:

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler().fit(train)

X\_train\_scaled = scaler.transform(train)

**4. Huấn Luyện Mô Hình**

**4.1 Naive Bayes**

from sklearn.naive\_bayes import GaussianNB

clf = GaussianNB().fit(X\_train\_scaled, labels)

preds = clf.predict(X\_test)

**4.2 Support Vector Machine (SVM)**

from sklearn.svm import SVC

clf = SVC(probability=True)

clf.fit(X\_train\_scaled, labels)

preds = clf.predict(X\_test)

**4.3 Logistic Regression**

from sklearn.linear\_model import LogisticRegression

clf = LogisticRegression(solver='newton-cg', multi\_class='multinomial')

clf.fit(X\_train\_scaled, labels)

preds = clf.predict(X\_test)

**5. Đánh Giá Kết Quả**

Sử dụng Accuracy và Log-loss:

from sklearn.metrics import accuracy\_score, log\_loss

acc = accuracy\_score(y\_test, preds)

ll = log\_loss(y\_test, clf.predict\_proba(X\_test))

print(f'Accuracy: {acc:.4%}')

print(f'Log Loss: {ll:.6}')

Kết quả:

* **Logistic Regression** đạt độ chính xác cao nhất (~99%)
* **Gaussian Naive Bayes** hoạt động kém hiệu quả

**6. Kết Quả & Hướng Phát Triển**

**6.1 Kết Quả Cuối Cùng**

* Mô hình Logistic Regression được chọn là tốt nhất.
* Đạt Top 23% trên Kaggle với Log-loss = **0.03157**.

**6.2 Hướng Phát Triển**

1. Tối ưu thêm Logistic Regression.
2. Thử nghiệm các mô hình khác.
3. Phát triển pipeline xử lý ảnh màu.

**7. Tài Liệu Tham Khảo**

* Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
* Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825-2830.
* Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.