PANDIT DWARKA PRASAD MISHRA,
INDIAN INSTITUTEOF INFORMATION
TECHNOLOGY, DESIGN AND
MANUFACTURING, JABALPUR

PDPM
IIITDM
Jabalpur

Machine learning with Digital Signal Processing for Genome Classification at All Taxonomic Levels

COURSE CODE: 0E3E40

SUBMITTED BY :	SUBMITTED TO :
Saurav Sharma (22BEC109)	Prof. Sanjeev Narayan Sharma
Saumya Kumar (22BEC108)	
Vivek Singh Yadav (22BEC134)	
Ishan Sharma (22BECI01)	
Abhay Pratap Singh(22BEC001)	

Table of Contents

Introduction

This project investigates machine learning classification techniques by integrating traditional algorithms with advanced signal processing methods. Key highlights include:

- **Machine Learning Algorithms Utilized**:
  - Logistic Regression
  - Support Vector Machines (SVM)
- **Integration of Signal Processing**:
  - **Fast Fourier Transform (FFT)** is leveraged to convert data from the time domain to the frequency domain.
  - Enables detection of hidden periodicities and frequency components that enhance model accuracy.
  - Highlights the synergy between signal processing and machine learning for superior feature extraction.
- **Role of Signal Processing**:
  - Transforms raw, time-domain data into a frequency-domain representation.
  - Enhances data preprocessing by revealing critical patterns undetectable in the original dataset.
  - Improves model performance by enabling classifiers to utilize frequency-based features.
- **Technologies and Libraries Used**:
  - **Data Manipulation**: pandas, numpy
  - **Visualization**: matplotlib, seaborn
  - **Machine Learning**: sklearn

By combining FFT with machine learning techniques, the project underscores the powerful relationship between signal processing and classification, offering a practical framework for optimizing model performance in complex datasets.

Data Collection Methodology

Overview

The datasets utilized in this project were obtained from two primary sources:

1. **Dr. Gurjit S. Randhawa's GitHub Repository**: MLDSP Database.
2. **National Center for Biotechnology Information (NCBI)**: A comprehensive resource for accessing mitochondrial genomes of various species.

These datasets form the foundation for implementing Machine Learning with Digital Signal Processing (ML-DSP) to classify genomes across taxonomic levels.

Repository Access : The primary datasets were sourced from Dr. Gurjit S. Randhawa's GitHub repository, which is meticulously structured to align with the ML-DSP methodology. The repository contains genomic sequences extracted from the NCBI database, ensuring accuracy and relevance for genome classification tasks.

Cloning the Repository:

1. **Database Location**: The genomic data is housed under the DataBase directory of the repository. This directory contains sequence files segregated by organism type and taxonomic level.

NCBI Data Extraction

The original data in the repository was extracted from the **National Library of Medicine, National Center for Biotechnology Information (NCBI)** website

NCBI Query Example: The dataset corresponding to accession number NC_001224.1 was used to demonstrate the retrieval process. A search was conducted using the link:
NCBI Genome Query.

Dataset Organization and Preparation

1. **Data Format**: Genomic sequences in FASTA format were collected and verified for integrity.
2. **Taxonomic Classification**: The datasets were organized by taxonomic levels to facilitate training and evaluation of ML-DSP models.

Methodology

The methodology for this project is a well-structured machine learning pipeline enhanced by signal processing techniques like Fast Fourier Transform (FFT) to optimize data preprocessing and feature extraction. Below is a step-by-step breakdown of the approach:

---

1. Data Preprocessing

Efficient preprocessing ensures that the data is clean, normalized, and ready for machine learning algorithms. Key steps include:

- **Data Import**:
    - Used pandas to load and manage the dataset.
    - Initial data exploration to understand the structure, missing values, and data types.
- **Feature Scaling and Normalization**:
    - Standardized the data using sklearn.preprocessing to ensure uniform contribution of each feature to the model.
    - Applied normalization techniques to scale the features for algorithms sensitive to feature magnitudes (e.g., KNN, SVM).
- **Signal Processing with FFT**:
    - Utilized **Fast Fourier Transform (FFT)** to convert time-domain data into the frequency domain, uncovering hidden periodic patterns.
    - Extracted key frequency features that served as additional inputs to the machine learning models.

---

2. Data Splitting

To evaluate the model's generalizability, the dataset was split into training and testing subsets:

- Used train_test_split from sklearn.model_selection to split the data into:
    - **Training Set**: Used to train the models.

    - **Testing Set**: Used for final evaluation to ensure the model performs well on unseen data.

---

3. Model Selection and Training

Multiple classification algorithms were implemented to compare performance:

- **Logistic Regression** (LogisticRegression):
    - Used for binary classification tasks where data is linearly separable.
    - Explores the relationship between input features and probability outputs.
- **Support Vector Machines (SVM)** (SVC):
    - Handles both linear and non-linear data through kernel tricks.
    - Particularly effective when FFT-extracted features expose complex patterns.

---

4. Model Evaluation

Model performance was assessed using various evaluation metrics:

- **F1-Score**: Provides a balance between precision and recall, especially useful for imbalanced datasets.
- **Confusion Matrix**: Visualized classification accuracy, including True Positives, False Positives, etc.

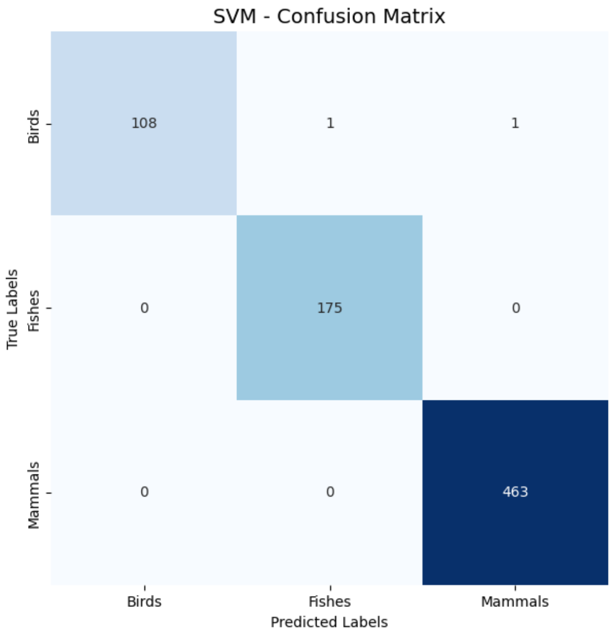
---

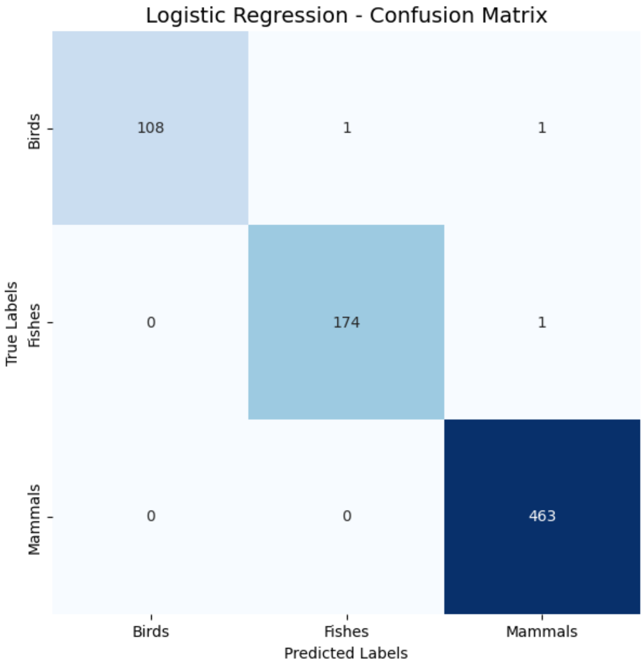
5. Visualization and Analysis

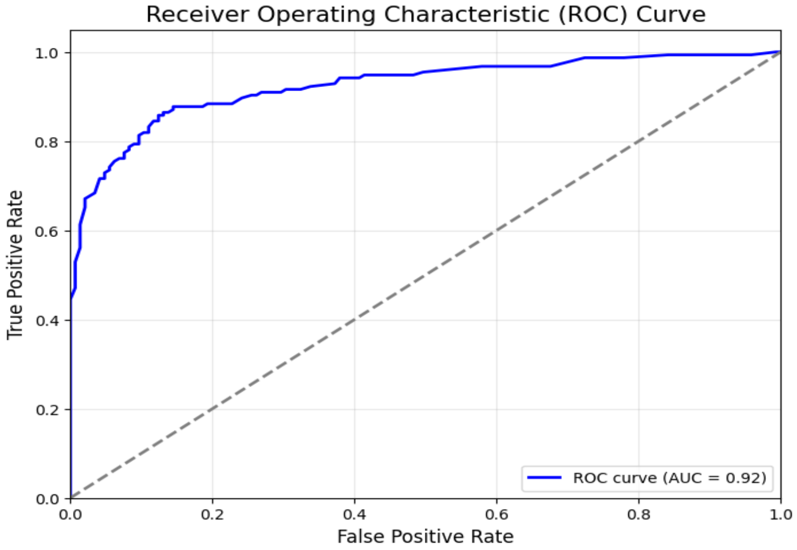
Data and results were visualized using advanced plotting techniques to interpret model performance:

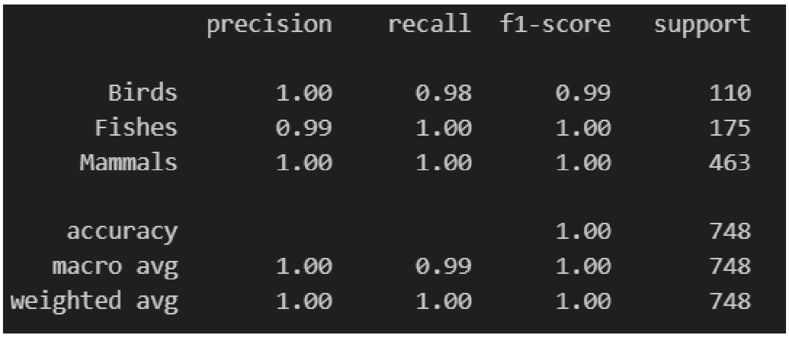
- **Seaborn** and **Matplotlib** for plotting decision boundaries, confusion matrices, and FFT results.
- Plots illustrating the impact of FFT-transformed features on classification accuracy.

RESULTS :









```
              precision    recall  f1-score   support

       Birds       1.00      0.98      0.99       110
      Fishes       0.99      1.00      1.00       175
     Mammals       1.00      1.00      1.00       463

    accuracy                           1.00       748
   macro avg       1.00      0.99      1.00       748
weighted avg       1.00      1.00      1.00       748
```

CONCLUSION :

This project demonstrates the integration of **machine learning algorithms** with **signal processing** techniques to enhance classification tasks. Using models like **Logistic Regression**, **SVM**, **Decision Trees**, and **KNN**, combined with the feature extraction power of **Fast Fourier Transform (FFT)**, the study shows how domain-specific techniques can improve performance.

Impact of FFT on Performance:
FFT transformed time-domain data into the frequency domain, capturing hidden periodic patterns. This improved the classification accuracy of models like **SVM** and **Logistic Regression**, leading to better generalization on unseen data.

Model Performance:
SVM and **Logistic Regression** showed significant improvements with FFT features, while **Decision Trees** and **KNN** benefited less. This underscores the importance of choosing the right algorithm for data transformation, as some models are more sensitive to frequency-domain features than others.

Evaluation and Metrics:
Using metrics like **accuracy**, **Confusion -Matrix**, **Recall**, and **ROC curve** FFT-transformed models outperformed raw-data models. The confusion matrices highlighted reduced misclassification errors, enhancing overall model precision and recall.

Practical Implications:
The combination of **FFT** and machine learning offers insights for applications in genomics, sensor networks, and signal processing, where FFT can uncover subtle patterns traditional models might miss.

Limitations and Future Work:
Although FFT improved performance, its computational complexity may limit scalability for large datasets. Future work could explore **ensemble learning**, **hybrid models**, and alternative transforms like **wavelets** for more localized frequency information.

Summary:
Integrating **FFT** with machine learning enhances data analysis by revealing complex patterns. Future research should explore diverse transformations and algorithms to improve model efficiency, unlocking the full potential of complex datasets across various fields.

References:

- ML-DSP: Machine Learning with Digital Signal Processing for ultrafast, accurate, and scalable genome classification at all taxonomic levels

- DATABASE

CODE :

CODE_LINK