APPLYING MACHINE LEARNING

MODULES on a DNA STRAND i.e., on a

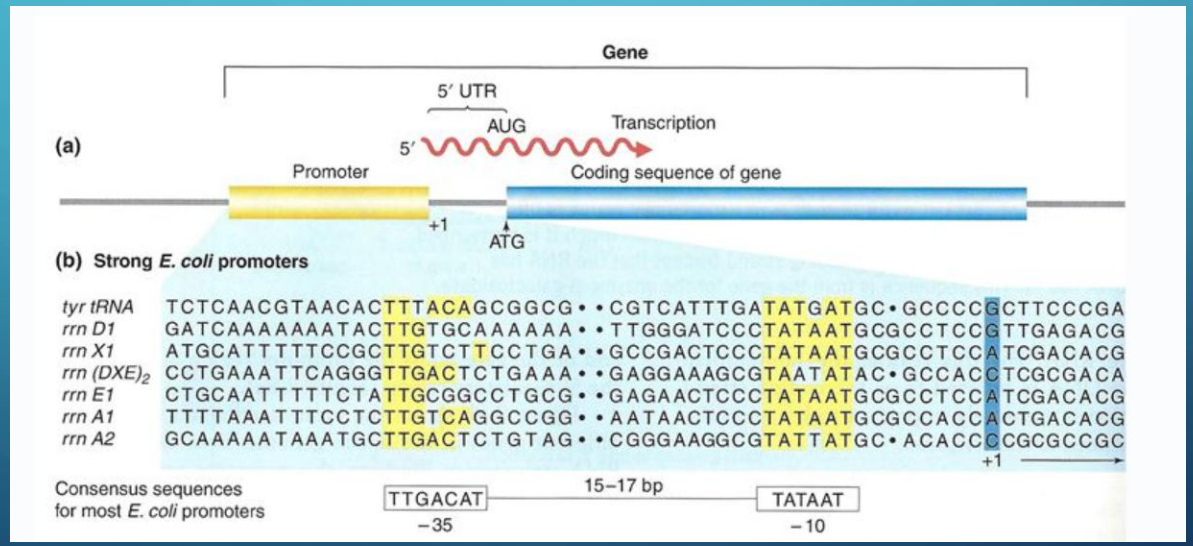
PROMPTER GENES

PROJECT PROGRESS UPDATE

BY: SAROJ SHAH

INTRODUCTION:

- The project focuses on applying various machine learning (ML) techniques to DNA strands of *E. coli* (especially the promoter gene).
- DNA classification is the process of categorizing DNA sequences into specific groups based on their characteristics.



PURPOSE AND GOALS:

- The primary goal of this project is to apply appropriate ML techniques such as preprocessing, training and testing, and evaluation to the imported dataset to develop the most suitable ML algorithm for DNA/RNA strand classification.
- This classification can serve as a foundation for future research, particularly in identifying various disease-causing bacteria and viruses.

APPROACH AND METHODOLOGY:

- MATLAB will be used to write and simulate ML code
- The methodology consists of three phases –
  - The First phase includes Data Collection, encoding, and fixing missing or misaligned nucleosides and other necessary data transformations.
  - The Second phase focuses on training, testing, and evaluating performance (scores) by using various methods like K-Nearest Neighbors, Neural Net, and others.
  - The third phase involves comparing scores to determine the most effective methods for the DNA/RNA classification.

DATA COLLECTION:

- For the project, the dataset can either be used directly from the UC Irvine Machine Learning Repository: https://archive.ics.uci.edu/ml/.../promoters.data

```
%% **Step 1: Importing the Dataset Correctly**
% The data has alternating lines: 1st is label and ID, 2nd is sequence
url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/molecular-biology/promoter-gene-sequences/promoters.data';
filename = 'promoters.data';
if exist(filename, 'file'), delete(filename); end
websave(filename, url);
```

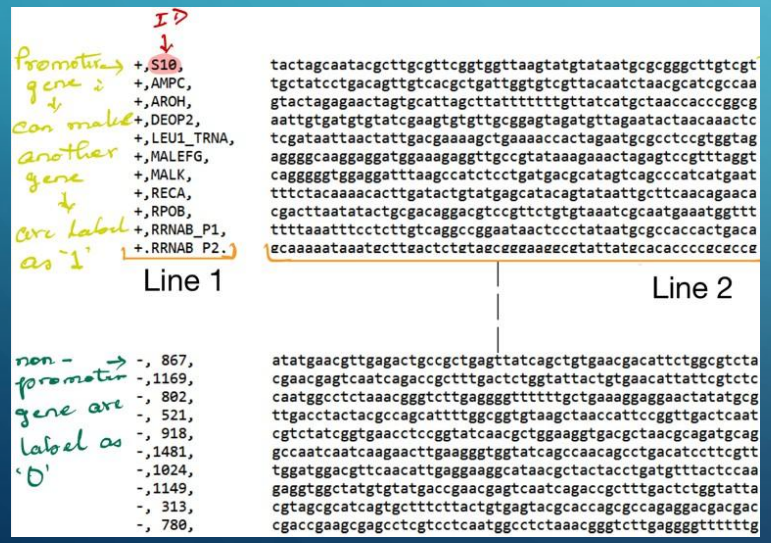
- Or it can be downloaded and stored locally for safer and more reliable access during the project.

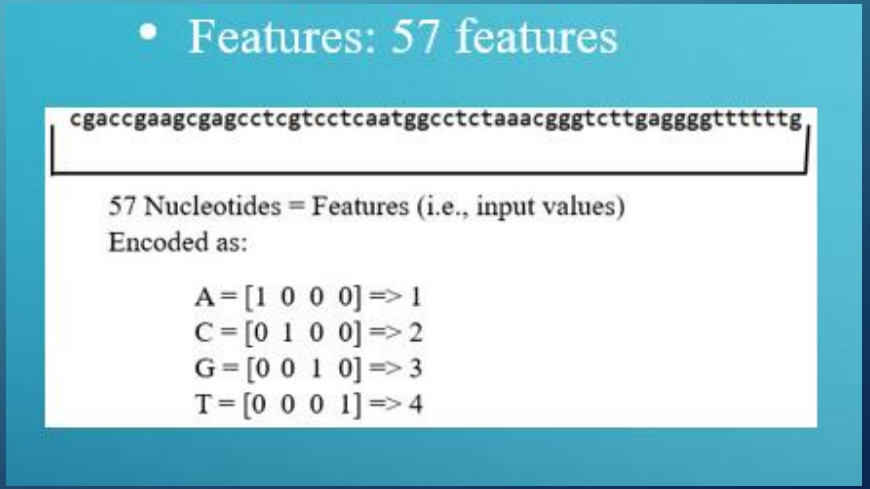
```
% Downloading file from saved location (Specifying the path to the file)
file_path = 'C:\Users\Saroj\Desktop\1 UAlbany\12 Spr 2025\1 Intro to Machine Learning Engr (IECE 565)\3 Project\0a Code\saved_promoters.data';
filename = 'promoters.data';
```

- For this project, the second method is used.

ABOUT DATA:

- The dataset contains 105 rows (DNA sequence) and 2 columns. Each row includes:
  - Column 1 (Line 1): the label (either 1 or 0)
  - Column 2 (Line 2): a sequence of 57 nucleotides (DNA base pairs), i.e., 57 features.
- There are two types of labels:
  - Label 1: Promoter gene — involved in initiating DNA replication
  - Label 0: Non-promoter gene — performs other functions such as protein synthesis.





DATA SET-UP:

- **Data Import & Setup:** After importing the dataset, the columns were renamed as **Class**, **ID**, and **Sequence**.
- The **Class** and **ID** columns were then converted to **string data types** for easier manipulation.

```
% Rename the columns
data.Properties.VariableNames = {'Class', 'id', 'Sequence'};

% Convert 'Class' and 'id' columns from cell arrays to string arrays
data.Class = string(data.Class);
data.id = string(data.id);
```

PRE-PROCESSING:

- **Cleaning:** Removed **whitespace**, **tabs**, and **prefix characters** from raw data
- **Sequence Filtering:** Retained only valid nucleotides: **A, C, G, T**, using **regex**
- **Encoding:** Mapped nucleotides to numbers: **A = 1**, **C = 2**, **G = 3**, **T = 4**

```
% Clean sequences by removing tabs and non-alphabet characters
for i = 1:length(sequences)
    sequence = sequences{i};
    cleaned_sequence = regexprep(sequence, '\t', '');
    cleaned_sequence = regexprep(cleaned_sequence, '[^acgtACGT]', '');
    sequences{i} = lower(cleaned_sequence);
end
```

```
numSequences = length(sequences);
encodedSequences = zeros(numSequences, 57);

for i = 1:numSequences
    seq = sequences{i};
    for j = 1:min(length(seq), 57)
        switch upper(seq(j))
            case 'A', encodedSequences(i, j) = 1;
            case 'C', encodedSequences(i, j) = 2;
            case 'G', encodedSequences(i, j) = 3;
            case 'T', encodedSequences(i, j) = 4;
        end
    end
end
```

TRAINING AND TESTING:

- **Data Splitting:** Split into **70% training** and **30% testing** datasets
- Multiple models used, such as KNN, SVM (Support Vector Machine), Decision Tree, Naive Bayes, Neural Networks, and others.

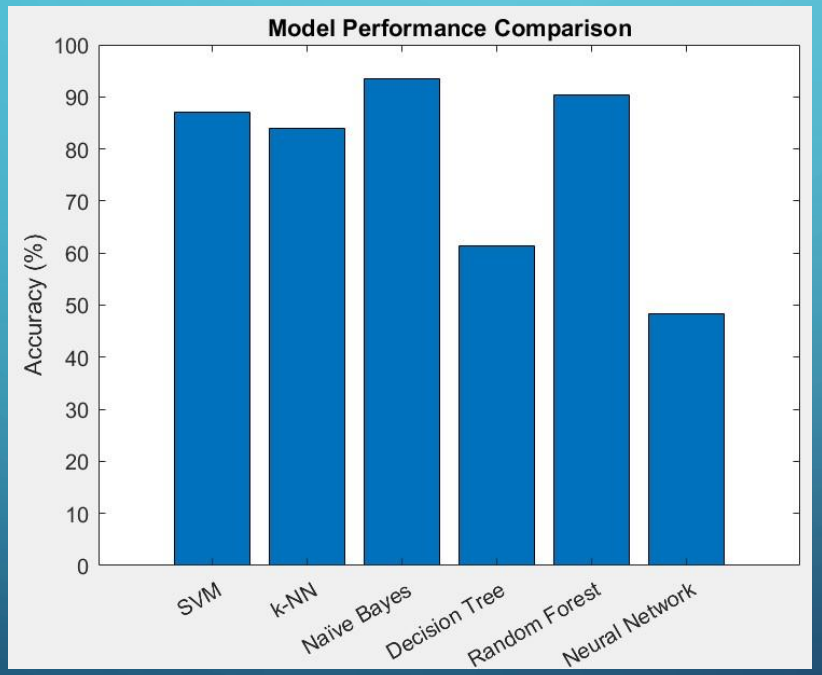
```
% Decision Tree classifier with max depth control using 'MaxNumSplits'
dtModel = fitctree(X_train_cleaned, y_train, 'MaxNumSplits', 5);              % MaxNumSplits limits depth by number of splits

% Other Classifiers
svmModel = fitcsvm(X_train_cleaned, y_train, 'KernelFunction', 'linear');     % Linear SVM (support vector machines)
knnModel = fitcknn(X_train_cleaned, y_train, 'NumNeighbors', 3);              % k-Nearest Neighbors (k-NN)
nbModel = fitcnb(X_train_cleaned, y_train);                                   % Naive Bayes classifier = probalistic classifier based
rfModel = fitcensemble(X_train_cleaned, y_train, 'Method', 'Bag', ...
    'NumLearningCycles', 10, 'Learners', 'tree');                             % Random Forest
mlpModel = fitcnet(X_train_cleaned, y_train, 'Lambda', 1);                    % Neural Network
```

RESULTS:

- Accuracy Result:

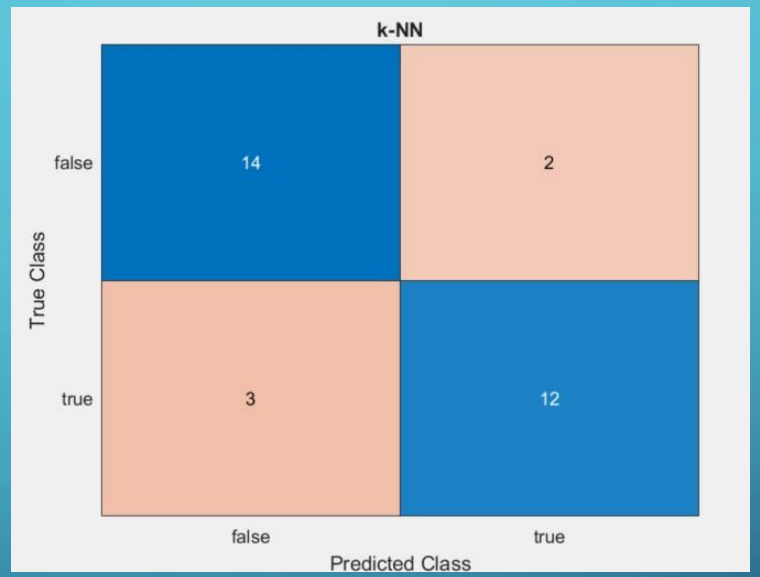
```
SVM Accuracy: 87.10%
k-NN Accuracy: 83.87%
Naïve Bayes Accuracy: 93.55%
Decision Tree Accuracy: 61.29%
Random Forest Accuracy: 90.32%
Neural Network Accuracy: 48.39%
```

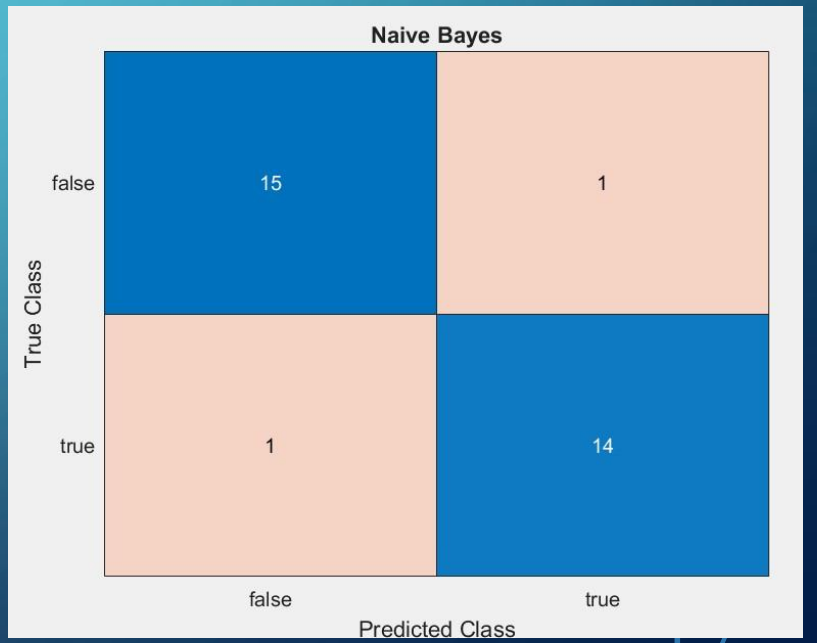
- Visualization:
  - Bar Chart



Confusion Matrices:

	Predicted Positive (1)	Predicted Negative (0)
Actual Positive (1)	True Positive (TP)	False Negative (FN)
Actual Negative (0)	False Positive (FP)	True Negative (TN)





CONCLUSION:

- Comparing the accuracy scores and classification enables the identification of the most effective method for DNA/RNA classification. Here in this project, since my data is small, Naïve Bayes scored outperformed others. But in reality, if you have a bigger data, the SVM is better for DNA Classification.
- As this project involves exploring and applying various machine learning techniques, it provides valuable insights into the underlying processes involved in building ML/AI models.

THANK YOU!

WORKS CITED

- Elgala, Hany. Class Notes, “Introduction to Machine Learning for Engineers- IECE 565.”
- Harley, C., R. Reynolds, and M. Noordewier. "Molecular Biology (Promoter Gene Sequences)." UCI Machine Learning Repository, 1987, https://doi.org/10.24432/C5S01D. Accessed 4 Feb. 2025.
- ajitsingh98. “GitHub - Ajitsingh98/DNA-Classification-Machine-Learning-Project: Classify DNA Sequence into Binary Class Using Different Classification Algorithms.” *GitHub*, 2019, github.com/ajitsingh98/DNA-Classification-Machine-Learning-Project. Accessed 4 Feb. 2025.
- Chat GPT