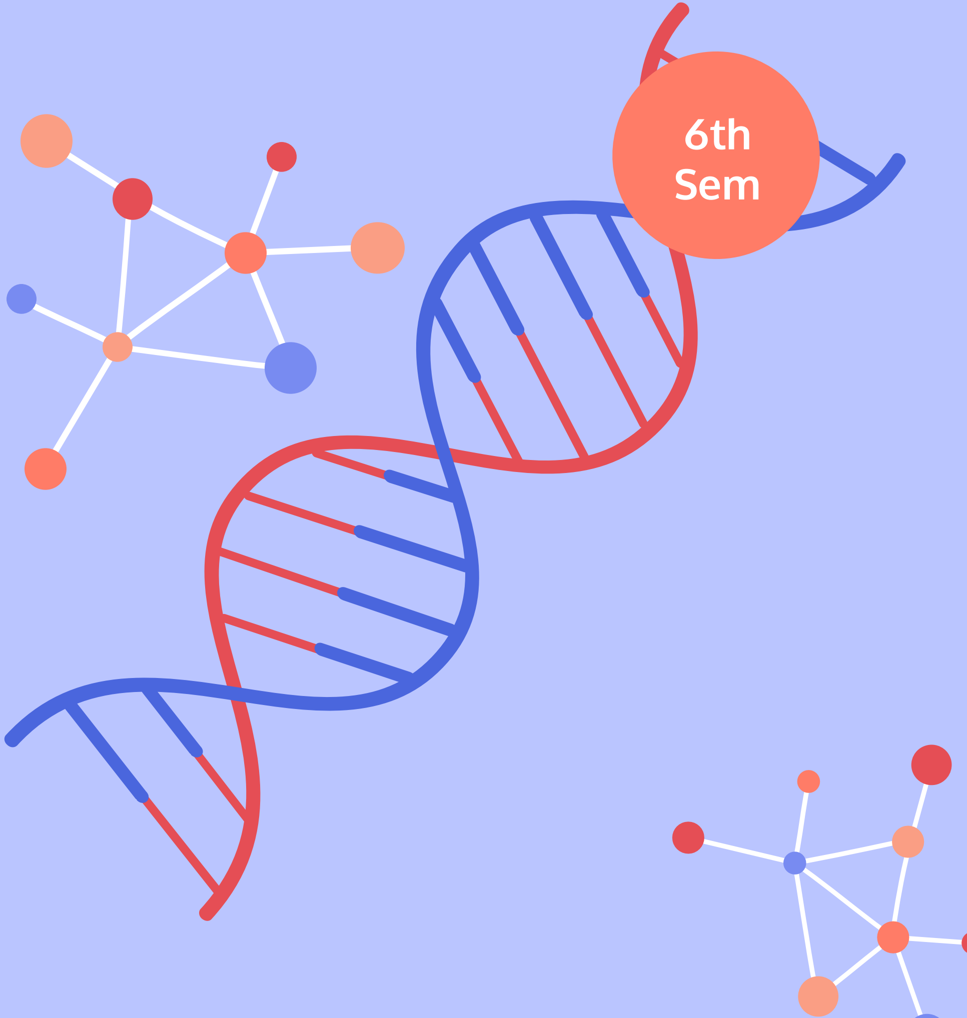
DNA SEQUENCE Classification Using Machine Learning and NLP



6th Sem

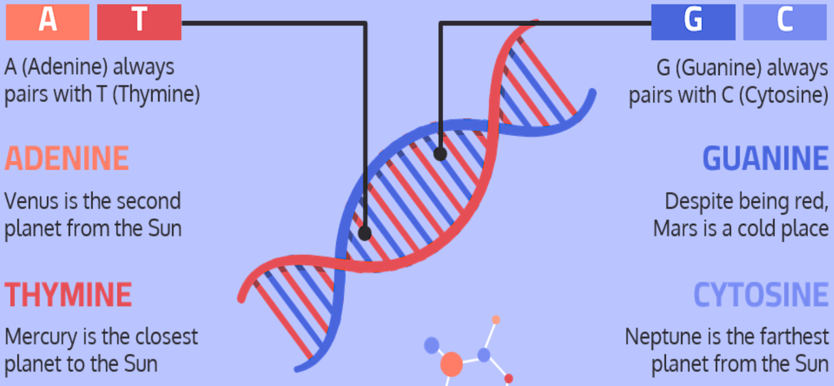
Presented By
RAJAT SINGH – 2205230
YASH MISHRA – 2205433
JAY ANMOL RATH- 2205212
AYUSH RAJ - 2205197

TABLE OF CONTENTS

Contents:

1. Introduction
2. SDLC Phases Overview
3. Phase 1: Planning
4. Phase 2: Requirements Analysis
5. Phase 3: System Design
6. Phase 4: Implementation
7. Phase 5: Testing
8. Phase 6: Deployment
9. Phase 7: Maintenance
10. Conclusion

DNA STRAND



A T

A (Adenine) always pairs with T (Thymine)

ADENINE

Venus is the second planet from the Sun

THYMINE

Mercury is the closest planet to the Sun

G C

G (Guanine) always pairs with C (Cytosine)

GUANINE

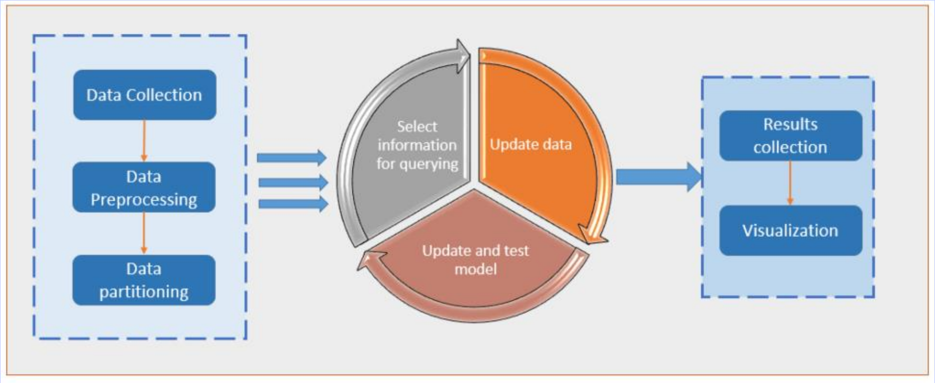
Despite being red, Mars is a cold place

CYTOSINE

Neptune is the farthest planet from the Sun

Introduction

- DNA classification enables species identification and functional region analysis.
- Combines Machine Learning and NLP for accuracy and efficiency.
- User-friendly desktop GUI provides real-time results.



SDLC Overview

01 Planning

02 Requirements Analysis

03 System Design

04 Implementation

05 Testing

06 Deployment

07 Maintenance

PLANNING

Objectives:

Develop an accurate and scalable classification system.
Create an intuitive desktop GUI with minimal latency.
Applications: research, forensic analysis, veterinary diagnostics, education.

02 Requirements Analysis

Functional Requirements:

Accept and validate DNA sequences.

Preprocess with k-mer Tokenization and Sequence Padding.

Classify sequences into predefined categories.

Non-Functional Requirements:

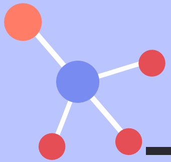
Performance, scalability, security, and usability.

03

System Design

Architecture:

Input: Web form for DNA sequence submission.
Processing: Tkinter API with ML model (CNN).
Output: Real-time dynamic display.
Key Components:
Frontend: Tkinter
Backend: `TensorFlow`, `Keras`, `NumPy`, `pickle`, `re`



Implementation 04

Technologies Used: Python, Tkinter,
Steps:
Train ML model with DNA datasets.
Develop Tkinter API for predictions.
Build desktop GUI with interactive design.

05 Testing

Types of Testing:

Unit Testing: Input validation, preprocessing.

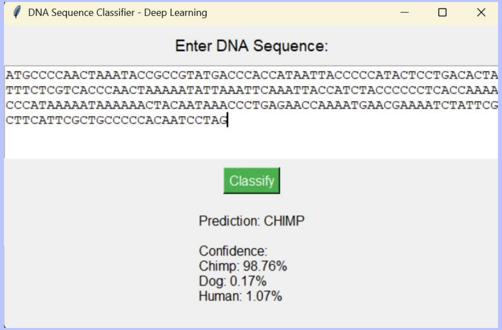
Integration Testing: Frontend-Backend communication.

Performance Testing: Response time and edge cases.

Valid Input: Classification success.

Invalid Input: Error message displayed.

Example Test Cases:



Enter DNA Sequence:

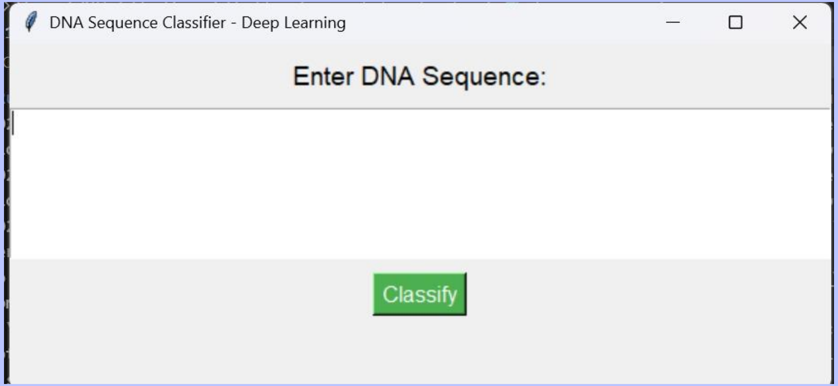
```
ATGCCCCAACTAAATACCGCCGTATGACCCACCATAATTACCCCCATACTCCTGACACTA
TTTCTCGTCACCCAACTAAAAATATTAAATTCAAATTACCATCTACCCCCCTCACCAAAA
CCCATAAAAATAAAAAACTACAATAAACCCTGAGAACCAAAATGAACGAAAATCTATTCG
CTTCATTCGCTGCCCCCACAATCCTAG
```

```
Prediction: CHIMP

Confidence:
Chimp: 98.76%
Dog: 0.17%
Human: 1.07%
```

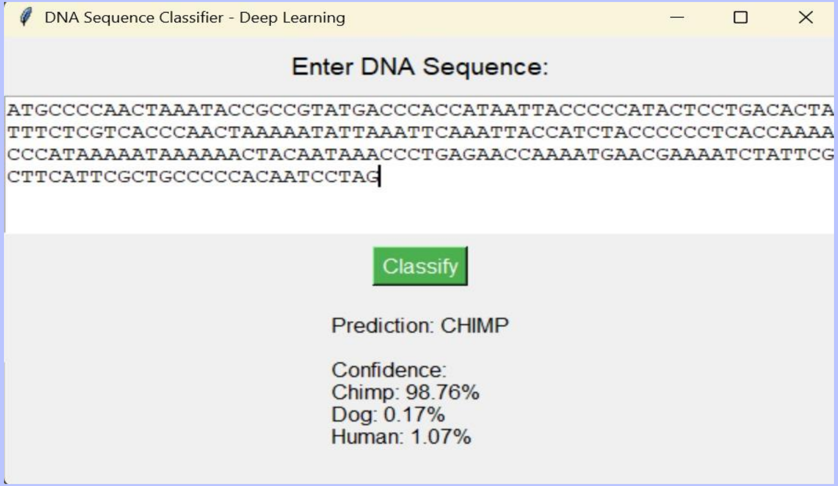
06 Deployment

Deployment Steps:
Run the Tkinter app locally.
Access the desktop GUI at local executable GUI.
Submit DNA sequences for real-time classification.
Features:
Interactive frontend with gradient backgrounds.
Responsive, user-friendly interface.



DNA Sequence Classifier - Deep Learning

Enter DNA Sequence:



DNA Sequence Classifier - Deep Learning

Enter DNA Sequence:

```
ATGCCCCAACTAAATACCGCCGTATGACCCACCATAATTACCCCCATACTCCTGACACTA
TTTCTCGTCACCCAACTAAAAATATTAAATTCAAATTACCATCTACCCCCCTCACCAAAA
CCCATAAAAATAAAAAACTACAATAAACCCTGAGAACCAAAATGAACGAAAATCTATTCG
CTTCATTCGCTGCCCCCACAATCCTAG
```

```
Prediction: CHIMP

Confidence:
Chimp: 98.76%
Dog: 0.17%
Human: 1.07%
```

07 Maintenance

Future Enhancements:

- Cloud migration for wider accessibility.
- Extend the dataset for improved classification.
- Use advanced preprocessing like embeddings.

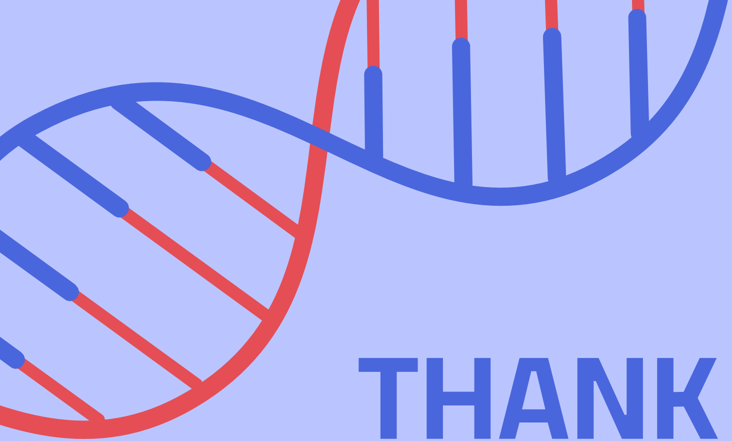
Monitoring:

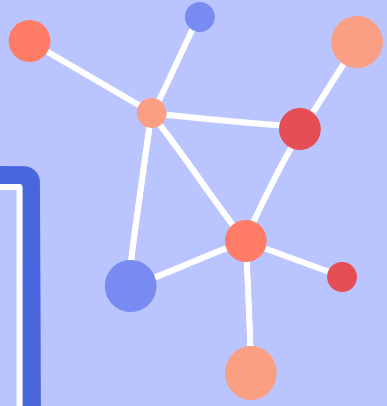
- Regular testing with updated datasets.
- Server performance tracking.

CONCLUSIONS

Successfully developed a DNA classification system.

Combines ML and NLP with a user-friendly desktop GUI.





THANK YOU!

DNA SEQUENCE Classification Using Machine Learning and NLP

Presented By:
Rajat Singh - 2205230
Yash Mishra - 2205433
Ayush Raj - 2205197
Jay Anmol Rath - 2205212