SCHOOL OFCOMPUTER SCIENCE AND APPLICATIONS

A Project Report
On

Predictive Analytics for Dyslexia: A Cloud Based Machine Learning Approach

Submitted in Partial fulfillment of the requirements for the award of the Degree of

Bachelor of Science (Honors) in Computer Science –
Cloud Computing and Big Data

Submitted by

Keerthivasan. R. V
R22DB044

Under the guidance of
Mrs. Jesla Joseph

January 2025

Rukmini Knowledge Park, Kattigenahalli, Yelahanka, Bengaluru-560064
www.reva.edu.in

REVA UNIVERSITY
Bengaluru, India

SCHOOL OF COMPUTER SCIENCE AND APPLICATIONS

CERTIFICATE

The project work titled – '**Predictive Analytics for Dyslexia: A Cloud Based Machine Learning Approach',** is being carried out under our guidance by **Keerthivasan R V (R22DBO44)**, a bonafide student of REVA University, and is submitting the project report in partial fulfillment, for the award of **Bachelor of Science (Honors) in Computer Science – Cloud Computing and Big Data** during the academic year **2024–25**. The project report has been approved, as it satisfies the academic requirements with respect to the Project Work prescribed for the aforementioned Degree.

Signature with date | **Signature with date**

Mrs. Jesla Joseph
Internal Guide

Dr. G. Sasikala
Head of the Department-BSc

Signature with Date

Dr C K Lokesh

Director

Name of the Examiner with Affiliation | **Signature with Date**

1.

2.

DECLARATION

I, Mr. Keerthivasan R V (R22DB044), pursuing my **Bachelor of Science (Cloud Computing and Big Data)**, offered by School of Computer Science and Applications, REVA University, declare that this Project title - **"Predictive Analytics for Dyslexia: A Cloud Based Machine Learning Approach"**, is the result of the Project Work done by me under the supervision of **Mrs. Jesla Joseph,** at Reva University.

I am submitting this Project Work in partial fulfillment of the requirements for the award of the degree of **Bachelor of Science (CC & BD)** by REVA University, Bengaluru, during the Academic Year 2024-25.

I further declare that this Project Report or any part of it has not been submitted for the award of any other Degree / Diploma of this University or any other University/ Institution.

(Signature of the candidate)

Signed by *me* on:

Certified that this project work submitted by Mr.Keerthivasan R V has been carried out under our guidance and the declaration made by the candidate is true to the best of my knowledge.

Signature of Internal Guide

Date :………..

Signature of Director of the School

Date :………..
Official Seal of the School

ACKNOWLEDGEMENT

I hereby acknowledge all those, under whose support and encouragement, I have been able to fulfill all my academic commitments successfully. In this regard, I take this opportunity to express my deep sense of gratitude and sincere thanks to School of Computer Science and Applications which has always been a tremendous source of guidance.

I express my sincere gratitude to **Dr. P. SHYAMA RAJU**, Honorable Chancellor, REVA University, and Bengaluru for providing us the state-of-the-art facilities.

I am thankful to **Dr. SANJAY CHITNIS**, Vice Chancellor, REVA University, **Dr K S NARAYANASWAMY,** Registrar, REVA University, Bengaluru for their support and encouragement.

I take this opportunity to express my heartfelt thanks to **Dr C K LOKESH**, Director, School of CSA, REVA University and my sincere thanks to **Dr. G. SASIKALA** Associate Professor, Head of the Department for BSc Program, School of CSA, REVA University, whose encouragement and best wishes provided impetus for the Project Work carried out.

Also, my sincere gratitude goes to my internal guide, **Mrs. JESLA JOESPH, Assistant Professor**, for the valuable suggestion and constant encouragement towards the completion of this project

Last, but not the least, I thank my parents for their incredible support and encouragement throughout.

ABSTRACT

Dyslexia, a common reading disability, affects numerous children and young people worldwide. It significantly impacts their ability to read, write, and process language, creating challenges in their academic and personal lives. Early detection and intervention are crucial in mitigating its effects. With advancements in technology, Machine Learning (ML) and Artificial Intelligence (AI) have become valuable tools in predicting dyslexia symptoms and understanding reading habits. These tools are used to analyse various patterns in behaviour, speech, and cognitive activities, offering an innovative approach to early diagnosis.

Data is the foundation for these predictive models. Information collected from individuals, including their reading patterns, response times, and error rates, is stored and processed using secure cloud storage services. Cloud platforms not only provide scalability and accessibility but also enable seamless collaboration among researchers and developers working on dyslexia prediction models. This ensures that large datasets can be analyzed efficiently, supporting diverse ML and AI techniques.

Several ML algorithms, including Decision Trees, Support Vector Machines (SVM), and Neural Networks, are applied to this data. These algorithms identify patterns and correlations that might not be apparent to human observers, enabling the creation of models that predict the likelihood of dyslexia with varying levels of accuracy. The effectiveness of these algorithms depends on several factors, such as the quality and quantity of the data, feature selection, and the algorithm's complexity.

The accuracy of dyslexia prediction varies among algorithms. For instance, while Neural Networks often achieve higher precision due to their ability to capture complex patterns, simpler models like Decision Trees may still be preferred for their interpretability. Continual advancements in ML and AI technologies, combined with the increasing availability of high-quality datasets, hold promise for even more reliable and accessible tools for dyslexia detection. By leveraging these cutting-edge technologies, researchers are paving the way for early diagnosis and support, enabling individuals with dyslexia to achieve their full potential.

TABLE OF CONTENTS

1. INTRODUCTION
1.1. INTRODUCTION TO PROJECT……………………………………………………1
1.1.1. STATEMENT OF THE PROBLEM……………………………………….....1
1.1.2. BRIEF DESCRIPTION OF THE PROJECT………………………………….1
1.2. FUNCTIONALAND NON-FUNCTIONAL REQUIREMENTS……………………2
1.3. SOFTWARE AND HARDWARE SPECIFICATION………………………………3
2. LITERATURE SURVEY…………………………………………………………......4
3. SYSTEM ANALYSIS
3.1. EXISTING SYSTEM………………………………………………………………..5
3.2. LIMITATIONS OF THE EXISTING SYSTEM………………………………….....6
3.3. PROPOSED SYSTEM………………………………………………………….....7
3.4. ADVANTAGES OF THE PROPOSED SYSTEM……………………………….....8
3.5. FEASIBILITY STUDY
3.5.1. TECHNICAL FEASIBILITY………………………………………………..8
3.5.2. ECONOMICAL FEASIBILITY……………………………………………..9
3.5.3. OPERATIONAL FEASIBILITY……………………………........................9
4. SYSTEM DESIGN AND DEVELOPMENT
4.1. HIGH LEVEL DESIGN (ARCHITECTURAL)……………………. …………….11
4.2. LOW LEVEL DESIGN……………………………………………………………11
4.3. DATAFLOW DIAGRAM…………………………………………………………12
4.4. USE CASE DIAGRAM……………………………………………………………13
4.5. SEQUENCE DIAGRAM………………………………………………………...13
4.6. ACTIVITY DIAGRAM……………………………………………………………14
5. MODULE DESCRIPTION
5.1. FRONTEND……………………………………………………………………...15
5.2. MACHINE LEARNING……………………………………………………………15
5.3. ALGORITHMS……………………………………………………………………..15
5.4. BACKEND…………………………………………………………………………16
6. CODING
6.1. PSEUDO CODE……………………………………………….........................17
6.2. USERINTERFACE DESIGN………………………………………………………37
6.3. USER EXPERIENCE………………………………………………………………37
7. SOFTWARE TESTING (Test cases)………………………………………………..39
8. CONCLUSION……………………………………………………….........................41
9. SCOPE FOR FUTURE ENCHANCEMENT…………………………………………41
10. BIBILOGRAPHY……………………………………………………………………42
11. APPENDIX
a. SNAPSHOTS………………………………………………………………………44

INTRODUCTION

1.1. Introduction to Project

In the Medical Industry, identifying Reading Disability (Dyslexia) presents significant challenges due to its multifaceted nature. Dyslexia manifests as difficulties with reading, writing, and spelling despite normal intelligence and sensory abilities. It can arise from various factors, including visual processing issues. Early detection is crucial for effective intervention and improving the quality of life for individuals with dyslexia.

This project leverages Machine Learning to predict dyslexia accurately. By analyzing collected data (structured, unstructured, and semi-structured), various algorithms (KNN, Decision Trees, RNN, XGBoost, Gradient Boosting) can be applied to identify patterns and predict the likelihood of dyslexia.

1.1.1 Statement of The Problem

Despite the increasing awareness of learning disabilities, dyslexia remains underdiagnosed, often due to limited access to standardized diagnostic tools. Current methods often rely on specialized clinics and professionals, making early identification challenging and time-consuming. This delay in diagnosis can hinder timely intervention and negatively impact a child's academic and social development.

1.1.2 Brief Description of The Project

This project aims to develop a user-friendly website that predicts dyslexia in real-time by analyzing eye coordination patterns. The core of the system is a Convolutional Neural Network (CNN) algorithm, chosen for its ability to efficiently process and analyze visual data.

The website allows users to upload sample images of eye coordination patterns and provides real-time predictions using multiple machine learning models (KNN, Decision Tree, RNN, XGBoost, Gradient Boosting). This enables users to compare predictions and gain valuable insights for early intervention.

A key feature of this project is its real-time analysis capability without the need for backend data storage, ensuring a lightweight and user-friendly experience while maintaining user privacy.

1.2 Functionality and Non Functionality Requirements

1.2.1 Functional Requirements:

Data Handling: Collect user data (reading patterns, response times, error rates), allowing data uploads and text input. Preprocess data, handling missing values, normalization, and feature engineering.

Prediction: Implement KNN, Decision Trees, RNN, XGBoost, and Gradient Boosting algorithms. Train models on labelled datasets, splitting data for training, validation, and testing. Evaluate models using metrics like accuracy, precision, recall, and F1-score.

User Experience: Provide an intuitive web/mobile interface. Enable real-time predictions with probability outputs. Compare algorithm performance and recommend the best model.

System: Enable real-time analysis and display actionable results. Store user data securely in the cloud with retrieval options. Generate detailed reports and offer suggestions for improving reading abilities.

1.2.2 Non-Functional Requirements

High Performance: Rapid processing and sub-2-second predictions. Efficient resource utilization.

Scalability: Handle large datasets and many users concurrently. Utilize scalable cloud services.

Security: Data encryption in transit and at rest. Compliance with data protection regulations (e.g., GDPR, HIPAA).

High Accuracy: Maintain over 90% accuracy across algorithms. Continuous improvement through data updates and model retraining.

Usability: User-friendly interface accessible to all. Multilingual support.

Reliability: High availability with minimal downtime. Fault tolerance for hardware/software failures.

Maintainability: Easy updates and integration. Modular design for efficient maintenance and debugging**.**

1.3 Software and Hardware Requirements

Min 4GB RAM
VS code or PyCharm
Python Libraries TensorFlow, Sci-kit Learn, numpy, pandas, Matplotlib, HTML, CSS, Java Script,Figma.
Jupyter Notebook
Anaconda Software

LITERATURE SURVEY

Dyslexia significantly impacts academic and personal development, particularly in children aged 5-12. Early identification is crucial. Machine Learning (ML) offers promising solutions, with algorithms like RNNs, Gradient Boosting, and XGBoost showing potential.

RNNs, especially LSTMs, excel with sequential data like reading and writing. However, they can be computationally expensive. Gradient Boosting improves model accuracy iteratively, while XGBoost offers efficient and scalable implementation. Decision Trees and KNN, while simpler, may be less accurate for complex datasets.

Data preprocessing is critical, involving feature extraction, normalization, and handling missing data. Cloud-based solutions facilitate data storage and accessibility.

The choice of ML algorithm depends on factors like time and space complexity. XGBoost often provides a good balance between accuracy and efficiency.

Challenges remain, including limited standardized datasets, computational inefficiencies, and generalization issues. Future research should focus on:

- Hybrid models combining RNN, Gradient Boosting, and XGBoost.
- Real-time data processing frameworks using cloud platforms.
- Developing interpretable models to support clinical decision-making.

SYSTEM ANALYSIS

3.1 Existing System

In the Existing System there were different ways data collected and processed for getting the highest accuracy. The eye co-ordination of the people is gathered and used for research purposes. The researchers have used Python as a major software for processing and presenting the graphical representation of result. There are google forms created for gathering data.

There were certain points which are used for predicting the Dyslexia. Ex: if the eye co-ordination of the people less than 0.85% is considered as the first level of Dyslexia and the points more than 0.85% are considered as non-Dyslexia.

In the Existing System the major algorithms used are

Random Forest
ID3
Decision Tree
logistic regression
SVM
KNN

There are Time and Space complexities involved in the existing system. The data which has collected and analysed did not provide the better results and some data were not useful and there were lot of noisy data were also presented while preprocessing the data.

3.2 Limitations of the existing system

Data Limitations:

- Reliance on limited data sources (e.g., Google Forms, eye coordination) restricts feature diversity.

- Lack of integration with behavioral, reading speed, and speech data limits the scope of analysis.
- Oversimplification of dyslexia symptoms through fixed thresholds can lead to inaccurate diagnoses.

Data Quality Issues:

- Noisy and incomplete data negatively impacts model performance.
- Inadequate preprocessing techniques (outlier removal, imputation, feature selection) reduce data reliability.

Algorithm Limitations:

- Dependence on traditional algorithms (Random Forest, Decision Tree, SVM) may not capture complex patterns.
- Limited use of advanced techniques like Neural Networks, XGBoost, and Gradient Boosting.

Performance and Scalability Issues:

- High time and space complexity in some algorithms can impact efficiency and scalability.
- The system may not effectively handle large datasets.

Interpretation and Ethical Concerns:

- Insufficiently interpretable results for non-technical users.
- Potential data security and privacy concerns with sensitive user data.

Evaluation Limitations:

- Absence of advanced evaluation techniques (cross-validation, confusion matrix analysis) undermines result reliability.

3.3 Proposed System:

The proposed system leverages advanced Machine Learning and Deep Learning techniques for the accurate prediction of dyslexia (reading disability). This system incorporates a web-based platform to gather and process real-time data efficiently, addressing the limitations of existing methods with the Eye Co-ordination patterns.

The dataset used for this project is sourced from GitHub, providing diverse and real-world data for analysis. Preprocessing techniques are applied to clean the data, remove noise, and select the most relevant features for prediction.

The below Algorithms are used:

KNN
Decision Tree
RNN Algorithm and CNN algorithm
XGBoost
Gradient Boosting

- Multi-Algorithm Approach - The use of diverse algorithms ensures a robust prediction model, leveraging the strengths of each technique.
- Improved Accuracy - Ensemble methods like XGBoost and Gradient Boosting enhance accuracy by reducing bias and variance.
- Sequential Analysis - RNN processes sequential data, adding a temporal dimension to the analysis for better insights.
- Efficient Processing - The combination of preprocessing, feature selection, and optimized algorithms reduces time and space complexity.
- Scalability - The system can handle varying dataset sizes and adapt to new data sources for continuous improvement.

3.4. Advantages of The Proposed System

Enhanced Accuracy: Integration of advanced algorithms like RNN, XGBoost, and Gradient Boosting improves accuracy by effectively handling complex and non-linear relationships in data.

Robustness to Noise: XGBoost and Gradient Boosting are specifically designed to handle noisy datasets, improving model resilience.

Sequential Data Analysis: RNNs effectively capture and analyze sequential patterns in data, crucial for understanding time-dependent features.

Comprehensive Insights: Combining multiple algorithms provides a holistic analysis, offering deeper insights into dyslexia factors.

Versatility: The system can process both categorical and numerical data.

Scalability: XGBoost and Gradient Boosting enable the system to handle large datasets efficiently, suitable for large-scale studies.

Interpretability: Algorithms like Decision Trees and KNN provide clear and interpretable results.

Adaptability: The system can adapt to diverse datasets by sourcing data from GitHub and effectively preprocessing it.

Resource Optimization: Optimized algorithms reduce computational complexity, ensuring efficient use of time and resources.

Early Intervention: Accurate predictions enable early identification of dyslexia, facilitating timely support and interventions.

3.5 Feasibility Study

3.5.1 Technical Feasibility

The system incorporates CNN, suitable for image-based eye-tracking data, alongside RNN, XGBoost, Gradient Boosting, KNN, and Decision Tree algorithms.

Data Availability**:** The GitHub dataset can be enhanced with eye-tracking data.

Computing Requirements**:** CNN may require higher computing power (CPU/GPU). Cloud computing (AWS, Google Cloud) can address this.

Technical Expertise**:** Expertise in CNNs, Deep Learning, and data preprocessing is required. Tools like TensorFlow and Keras are readily available.

Development Tools**:** Open-source Python libraries (TensorFlow, Keras, Scikit-learn) and access to high-performance hardware facilitate implementation.

Conclusion: The system is technically feasible with the addition of CNN, and all necessary tools and resources are available.

3.5.2 Economical Feasibility

Low-Cost Tools: Open-source frameworks (TensorFlow, Keras) and free GitHub datasets minimize development costs.

Hardware Requirements**:** Cloud computing offers cost-effective access to high-performance hardware (GPUs).

Development Costs**:** Costs for processing eye-tracking data and training the CNN model can be offset by using open-source tools and cloud resources.

Long-Term Benefits**:** Improved prediction accuracy and potential commercial applications in education, healthcare, and diagnostics offer significant ROI.

Conclusion**:** The project remains economically feasible, even with the addition of CNN. The system offers a good ROI through its improved accuracy and potential commercial applications.

3.5.3 Operational Feasibility

Ease of Use**:** The system will remain user-friendly with intuitive output visualizations and predictions, even with the addition of CNN.

Adaptability**:** The system is scalable to handle additional data sources and adaptable to real-world data variations.

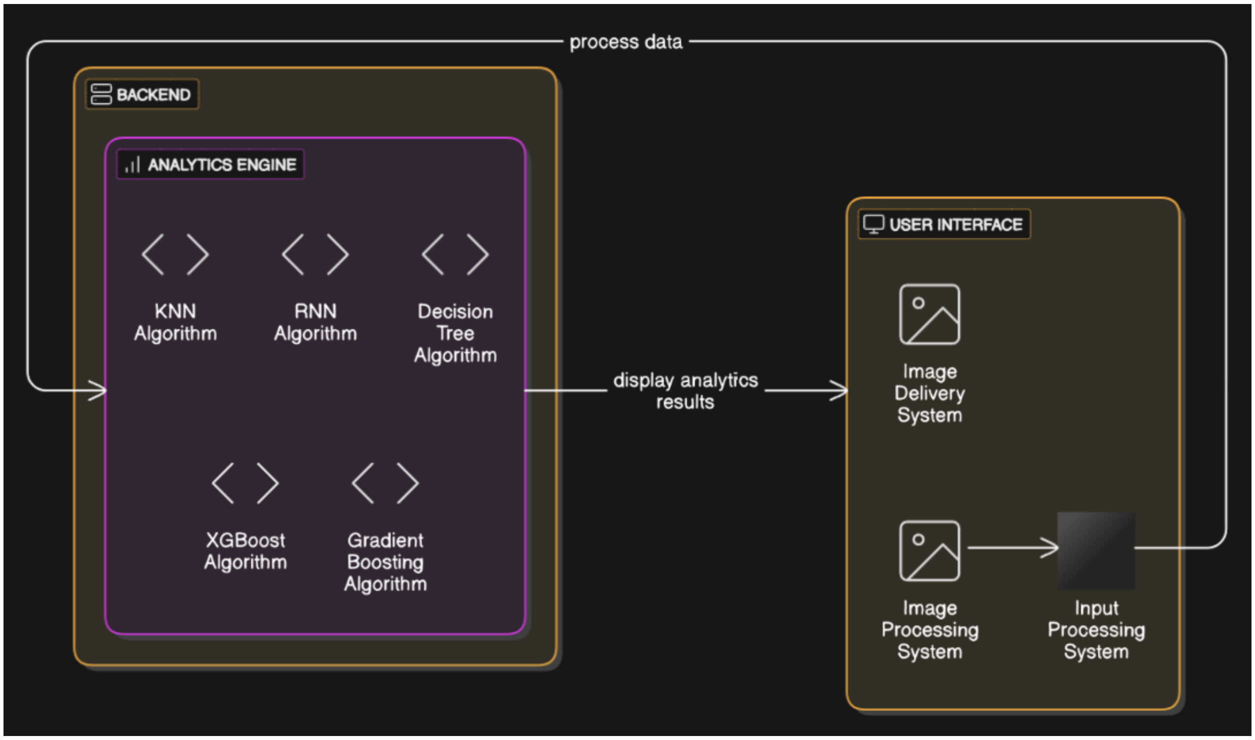
Implementation Challenges**:** The main challenge is the requirement for high computational power during CNN training, which can be mitigated through cloud services.

Stakeholder Benefits**:** The inclusion of CNN for eye coordination pattern recognition provides a more precise model for dyslexia diagnosis, enhancing early detection and intervention.

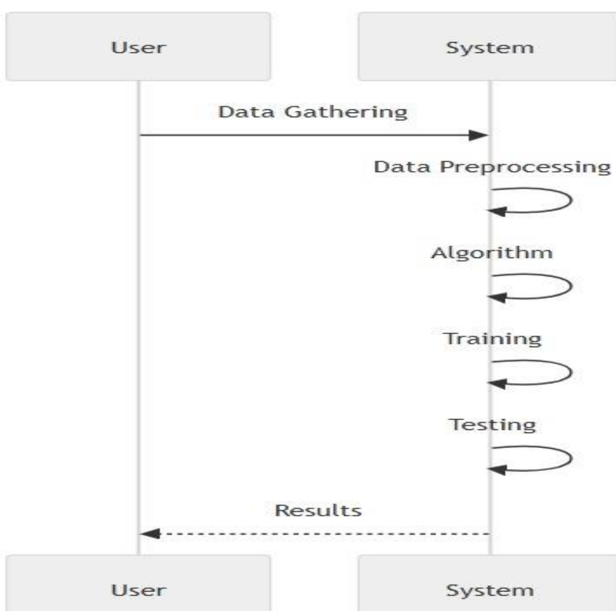
Conclusion**:** The system is operationally feasible, and the inclusion of CNN for predicting eye coordination patterns enhances its real-world applicability.

SYSTEM DESIGN AND DEVELOPMENT

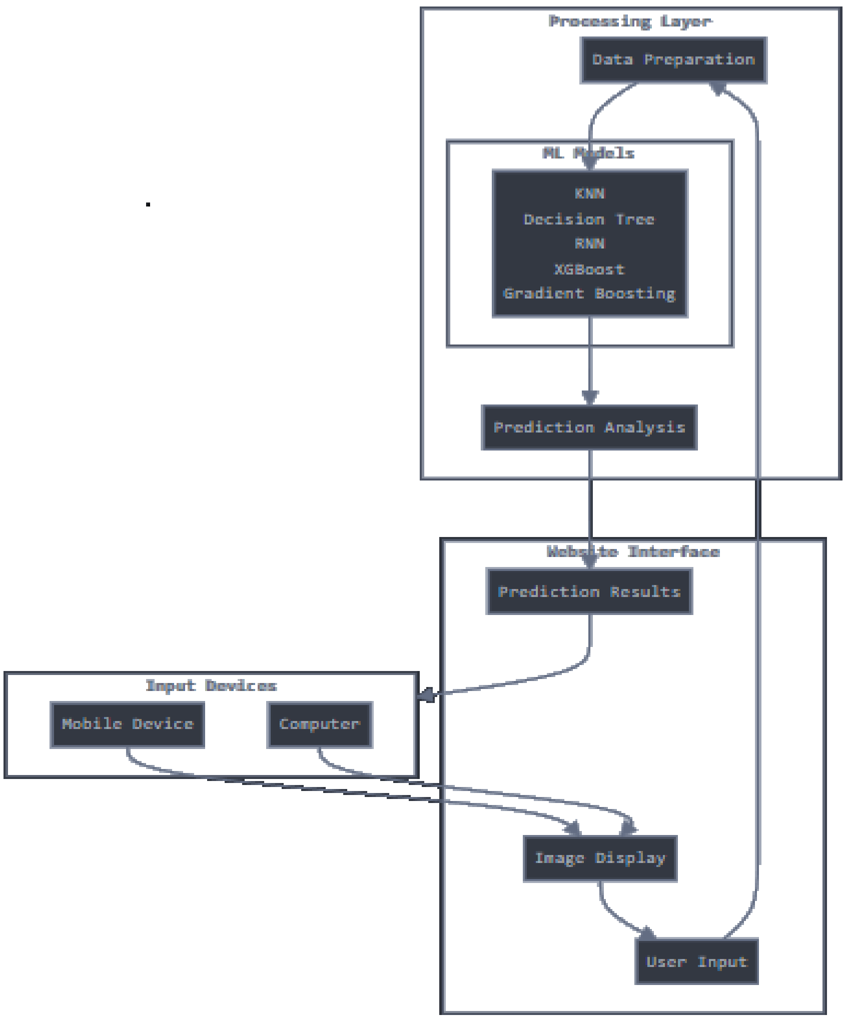
4.1 HIGH LEVEL DESIGN (ARCHITECTURAL)



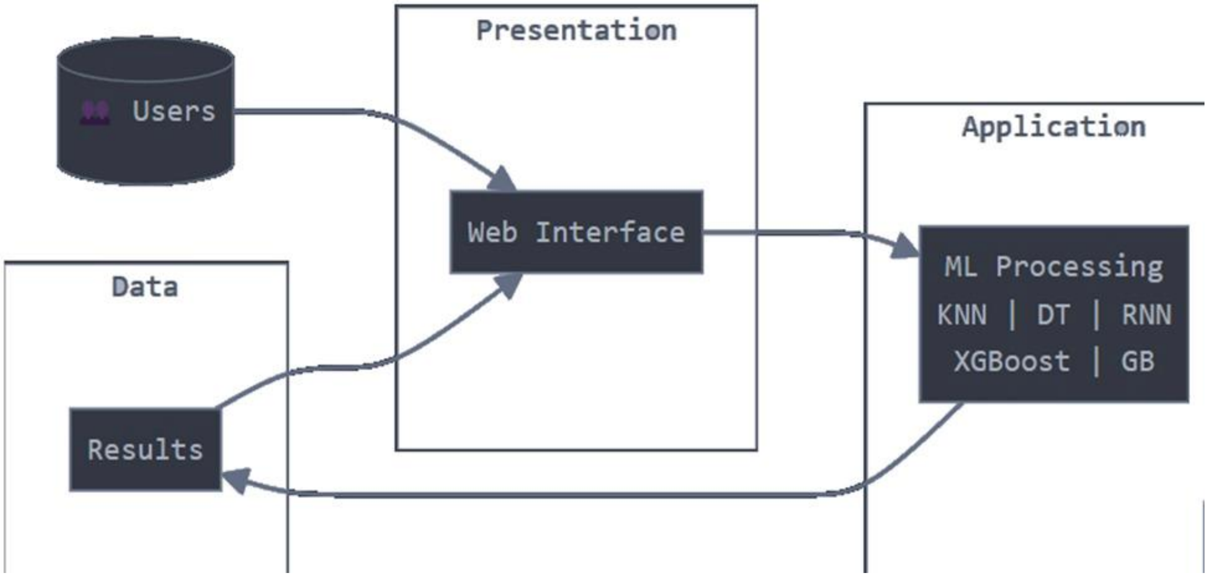
4.2. LOW LEVEL DIAGRAM



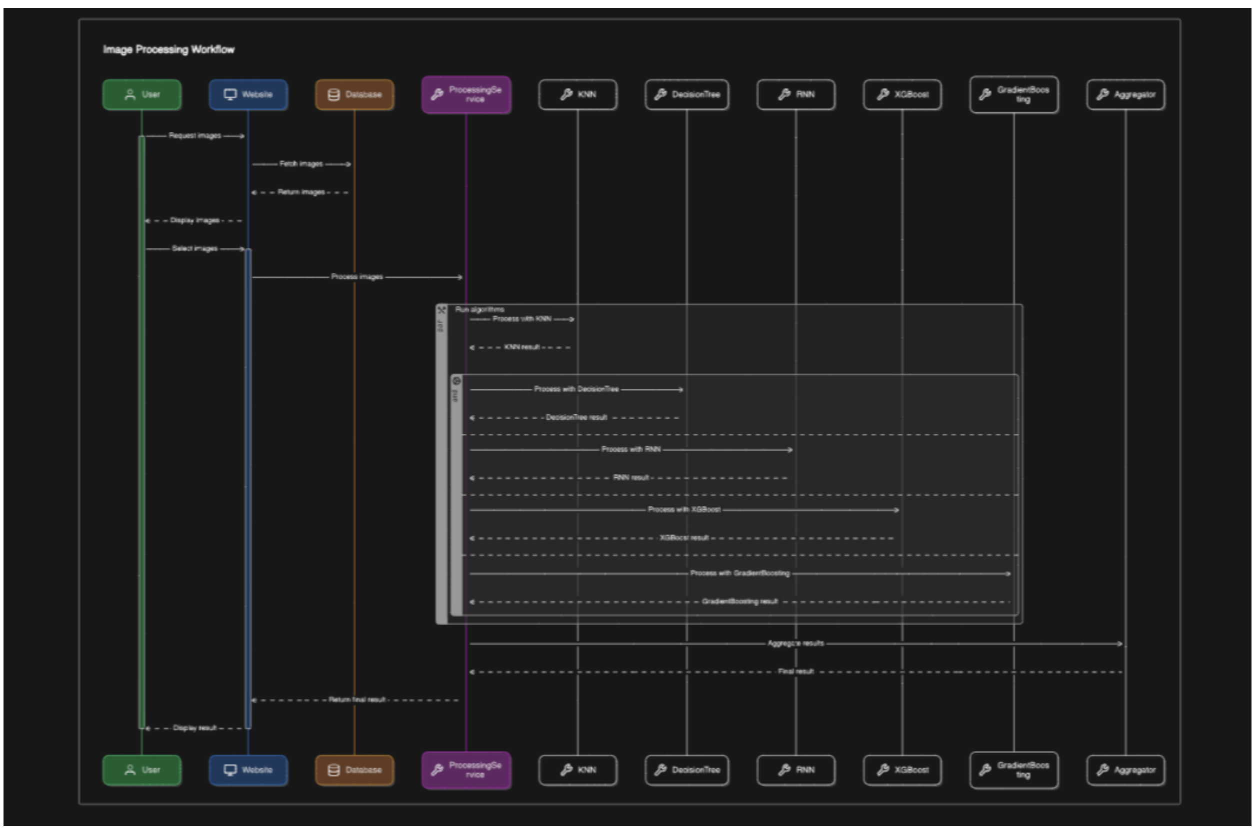
4.3 DATAFLOW DIAGRAM



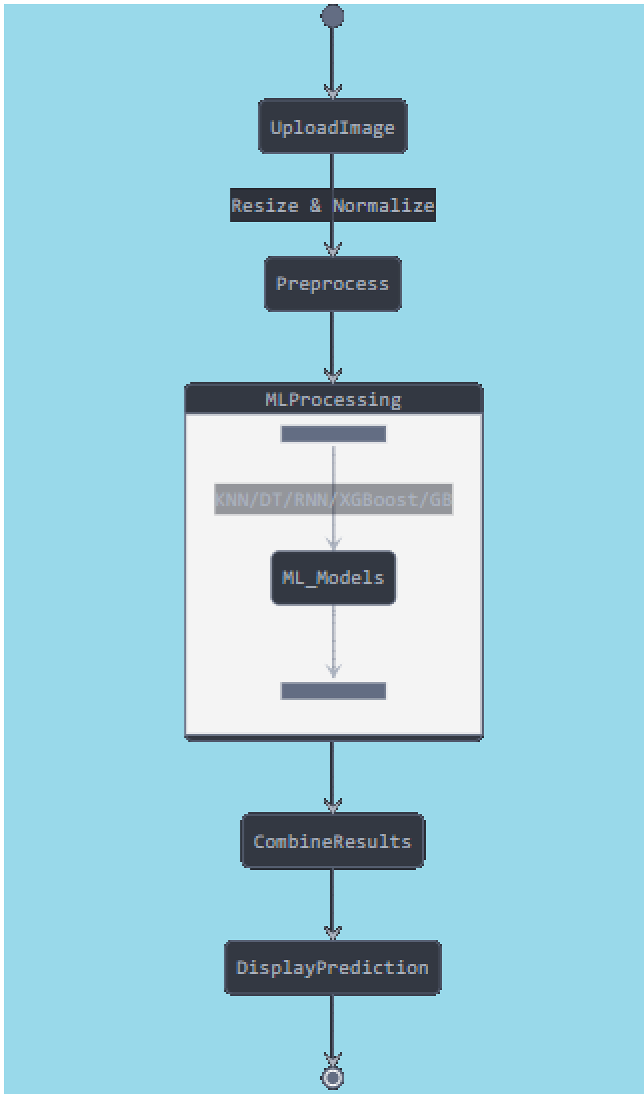
4.4 USE CASE DIAGRAM



4.5 SEQUENCE DIAGRAM



4.6 ACTIVITY DIAGRAM



MODULE DESCRIPTION

5.1 Frontend (HTML, CSS, JavaScript)

The frontend, built with HTML, CSS, and JavaScript, provides a user-friendly interface. HTML structures the website, CSS styles it, and JavaScript adds interactivity, enabling real-time analysis and dynamic results. AJAX facilitates seamless communication with the backend. Input validation ensures data quality, and accessibility is prioritized. A secure login system allows for personalized access.

5.2 Machine Learning

The machine learning module is the core of the project, processing user data to predict reading disabilities and analyze eye coordination. Preprocessing steps ensure data suitability for analysis. Multiple machine learning algorithms are integrated for robust predictions. Models are trained and tested using cross-validation to prevent overfitting. The module supports scalability and enables real-time processing. It is implemented in Python using libraries like NumPy, pandas, and scikit-learn.

5.3 Algorithms

5.3.1 K-Nearest Neighbors (KNN):

- Predicts by comparing input data to nearest neighbors in the training dataset.
- Interpretable, but computationally intensive.
- Uses scikit-learn for efficient implementation.

5.3.2 Decision Tree:

- Creates a tree-like structure to classify data based on feature values.
- Interpretable and handles both numerical and categorical data.
- Uses scikit-learn for implementation.

5.3.3 Recurrent Neural Network (RNN):

- Handles sequential data like eye movement patterns using LSTM units.

- Implemented using TensorFlow with GPU acceleration.
- Captures temporal dependencies in the data.

5.3.4 XGBoost:

- High-performance ensemble method based on gradient boosting.
- Handles large datasets efficiently with regularization techniques.
- Implemented using the XGBoost library.

5.3.5 Gradient Boosting:

- Builds models sequentially to minimize prediction errors.
- Achieves high accuracy with regularization techniques.
- Implemented using scikit-learn.

5.3.6 Convolutional Neural Network (CNN):

- Analyzes eye coordination patterns by extracting features from image-like data.
- Implemented using TensorFlow with GPU acceleration.
- Ideal for spatial data analysis.

5.4 Backend

The backend integrates machine learning models, acting as an API. The frontend communicates with the backend via HTTP or WebSocket requests. User data is sent to the backend, processed by ML models, and results are sent back to the frontend for display.

CODING

6.1 PSEUDO CODE

6.1.1 GRADIENT BOOST

```
import pandas as pd

import numpy as np

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.ensemble import GradientBoostingClassifier

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

import matplotlib.pyplot as plt

# Step 1: Load the dataset

data = pd.read_csv('ct5129_dyslexia/data/Dyt-desktop.csv', engine='python')

# Step 2: Parse the dataset into structured columns

parsed_data = data.iloc[:, 0].str.split(';', expand=True)

# Step 3: Assign column names dynamically

columns = ["Gender", "NativeLang", "OtherLang", "Age"]

for i in range(1, 33):

    columns.extend([f"Clicks{i}", f"Hits{i}", f"Misses{i}", f"Score{i}", f"Accuracy{i}",
f"Missrate{i}"])

columns.append("Dyslexia")

# Validate the number of columns

if len(columns) != parsed_data.shape[1]:

    print(f"Warning: Column mismatch! Generated {len(columns)} names, but the dataset has
{parsed_data.shape[1]} columns.")

    columns = columns[:parsed_data.shape[1]]
```

```
parsed_data.columns = columns

# Step 4: Encode categorical columns

encoder = LabelEncoder()

parsed_data["Gender"] = encoder.fit_transform(parsed_data["Gender"])

parsed_data["NativeLang"] = encoder.fit_transform(parsed_data["NativeLang"])

parsed_data["OtherLang"] = encoder.fit_transform(parsed_data["OtherLang"])

parsed_data["Dyslexia"] = encoder.fit_transform(parsed_data["Dyslexia"])

# Step 5: Convert numerical columns

for col in parsed_data.columns[3:]:

    parsed_data[col] = pd.to_numeric(parsed_data[col], errors='coerce')

# Handle missing values

parsed_data.fillna(parsed_data.mean(), inplace=True)

# Step 6: Define features and target

features = parsed_data.drop(columns=["Dyslexia"])

target = parsed_data["Dyslexia"]

# Step 7: Standardize features

scaler = StandardScaler()

features = scaler.fit_transform(features)

# Step 8: Split the dataset into training and testing sets

X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.2,
random_state=42)

# Step 9: Build the Gradient Boosting model

model = GradientBoostingClassifier(n_estimators=100, learning_rate=0.1, max_depth=3,
random_state=42)

# Step 10: Train the model
```

```
model.fit(X_train, y_train)

# Step 11: Evaluate the model

y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)

print(f"Test Accuracy: {accuracy * 100:.2f}%")

# Detailed classification report

print("\nClassification Report:")

print(classification_report(y_test, y_pred))

# Confusion matrix

conf_matrix = confusion_matrix(y_test, y_pred)

print("\nConfusion Matrix:")

print(conf_matrix) # Step 12: Visualize Results with Bar Chart

read_disability_count = np.sum(y_pred == 1)

non_read_disability_count = np.sum(y_pred == 0)

labels = ["Reading Disability", "Non-Reading Disability"]

counts = [read_disability_count, non_read_disability_count]

colors = ['red', 'yellow']

plt.figure(figsize=(8, 6))

plt.bar(labels, counts, color=colors)

plt.title('Distribution of Reading Disability vs Non-Reading Disability')

plt.xlabel('Category')

plt.ylabel('Count')

plt.show()

print(f"\nReading Disability: {read_disability_count}")

print(f"Non-Reading Disability: {non_read_disability_count}")
```

6.1.2 CNN

```
import pandas as pd

import numpy as np

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.ensemble import GradientBoostingClassifier

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

import matplotlib.pyplot as plt

# Step 1: Load the dataset

data = pd.read_csv('ct5129_dyslexia/data/Dyt-desktop.csv', engine='python')

# Step 2: Parse the dataset into structured columns

parsed_data = data.iloc[:, 0].str.split(';', expand=True)

# Step 3: Assign column names dynamically

columns = ["Gender", "NativeLang", "OtherLang", "Age"]

for i in range(1, 33):

    columns.extend([f"Clicks{i}", f"Hits{i}", f"Misses{i}", f"Score{i}", f"Accuracy{i}",
f"Missrate{i}"])

columns.append("Dyslexia")

# Validate the number of columns

if len(columns) != parsed_data.shape[1]:

    print(f"Warning: Column mismatch! Generated {len(columns)} names, but the dataset has
{parsed_data.shape[1]} columns.")

    columns = columns[:parsed_data.shape[1]]

     parsed_data.columns = columns

# Step 4: Encode categorical columns
```

```
encoder = LabelEncoder()

parsed_data["Gender"] = encoder.fit_transform(parsed_data["Gender"])

parsed_data["NativeLang"] = encoder.fit_transform(parsed_data["NativeLang"])

parsed_data["OtherLang"] = encoder.fit_transform(parsed_data["OtherLang"])

parsed_data["Dyslexia"] = encoder.fit_transform(parsed_data["Dyslexia"])

# Step 5: Convert numerical columns

for col in parsed_data.columns[3:]:

    parsed_data[col] = pd.to_numeric(parsed_data[col], errors='coerce')

# Handle missing values

parsed_data.fillna(parsed_data.mean(), inplace=True)

# Step 6: Define features and target

features = parsed_data.drop(columns=["Dyslexia"])

target = parsed_data["Dyslexia"]

# Step 7: Standardize features

scaler = StandardScaler()

features = scaler.fit_transform(features)

# Step 8: Split the dataset into training and testing sets

X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.2,
random_state=42)

# Step 9: Build the Gradient Boosting model

model = GradientBoostingClassifier(n_estimators=100, learning_rate=0.1, max_depth=3,
random_state=42)

# Step 10: Train the model

model.fit(X_train, y_train)

# Step 11: Evaluate the model
```

```
y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)

print(f"Test Accuracy: {accuracy * 100:.2f}%")

# Detailed classification report

print("\nClassification Report:")

print(classification_report(y_test, y_pred))

# Confusion matrix

conf_matrix = confusion_matrix(y_test, y_pred)

print("\nConfusion Matrix:")

print(conf_matrix)

# Step 12: Visualize Results with Bar Chart

read_disability_count = np.sum(y_pred == 1)

non_read_disability_count = np.sum(y_pred == 0)

labels = ["Reading Disability", "Non-Reading Disability"]

counts = [read_disability_count, non_read_disability_count]

colors = ['red', 'yellow']

plt.figure(figsize=(8, 6))

plt.bar(labels, counts, color=colors)

plt.title('Distribution of Reading Disability vs Non-Reading Disability')

plt.xlabel('Category')

plt.ylabel('Count')

plt.show()

print(f"\nReading Disability: {read_disability_count}")

print(f"Non-Reading Disability: {non_read_disability_count}")
```

6.1.3 WEB PAGE 1

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Dyslexia Detection System</title>
    <style>
        * {
            margin: 0;
            padding: 0;
            box-sizing: border-box;
            font-family: Arial, sans-serif;
        }
        body {
            padding: 20px;
            background-color: #f5f5f5;
        }
        .container {
            max-width: 1200px;
            margin: 0 auto;
        }
        .header {
            text-align: center;
            margin-bottom: 30px;
```

```
}
.header h1 {
    color: #333;
    margin-bottom: 10px;
}
.instructions {
    background: #fff;
    padding: 20px;
    border-radius: 8px;
    margin-bottom: 20px;
    box-shadow: 0 2px 4px rgba(0,0,0,0.1);
}
.image-grid {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
    gap: 20px;
    margin-bottom: 20px;
}
.image-card {
    background: white;
    padding: 20px;
    border-radius: 8px;
    box-shadow: 0 2px 4px rgba(0,0,0,0.1);
}
.image-card img {
```

```
    width: 100%;
    height: 300px;
    object-fit: contain;
    border-radius: 4px;
    margin-bottom: 15px;
}
.image-card input {
    width: 100%;
    padding: 10px;
    border: 2px solid #ddd;
    border-radius: 4px;
    font-size: 16px;
}

.image-card input:focus {
    border-color: #4CAF50;
    outline: none;
}
.submit-btn {
    background-color: #4CAF50;
    color: white;
    padding: 12px 24px;
    border: none;
    border-radius: 4px;
    cursor: pointer;
```

```
    font-size: 16px;
    width: 100%;
    margin-bottom: 20px;
}
.submit-btn:hover {
    background-color: #45a049;
}
.loading {
    display: none;
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background: rgba(255, 255, 255, 0.9);
    justify-content: center;
    align-items: center;
    z-index: 1000;
}
.spinner {
    width: 50px;
    height: 50px;
    border: 5px solid #f3f3f3;
    border-top: 5px solid #4CAF50;
    border-radius: 50%;
```

```
            animation: spin 1s linear infinite;
        }
        @keyframes spin {
            0% { transform: rotate(0deg); }
            100% { transform: rotate(360deg); }
        }
    </style>
</head>
<body>
    <div class="loading">
        <div class="spinner"></div>
    </div>
    <div class="container">
        <div class="header">
            <h1>Dyslexia Detection System</h1>
            <p>Image Recognition and Analysis</p>
        </div>
        <div class="instructions">
            <h3>Instructions:</h3>
            <ol style="margin-left: 20px; margin-top: 10px;">
                <li>Look at each image carefully</li>
                <li>Type what you see in the text box below each image</li>
                <li>Check your spelling before submitting</li>
                <li>Click 'Analyze Answers' when ready</li>
            </ol>
```

```
        </div>
        <div class="image-grid">
            <div class="image-card">
                <img src="flower.jpg" alt="Purple Flower">
                <input type="text" placeholder="What flower do you see?" class="answer-input" data-correct="flower">
            </div>
            <div class="image-card">
                <img src="hello.png" alt="Hello Text">
                <input type="text" placeholder="What word do you see?" class="answer-input" data-correct="hello">
            </div>
            <div class="image-card">
                <img src="lion.jpg" alt="Lion Art">
                <input type="text" placeholder="What animal do you see?" class="answer-input" data-correct="lion">
            </div>
        </div>
        <button class="submit-btn" onclick="submitAnswers()">Analyze Answers</button>
    </div>
    <script>
        function submitAnswers() {
            // Show loading spinner
            document.querySelector('.loading').style.display = 'flex';

            // Collect answers
```

```
            const answers = Array.from(document.querySelectorAll('.answer-input')).map(input =>
                encodeURIComponent(input.value.trim()) + ',' +
encodeURIComponent(input.dataset.correct)
            ).join('|');
            // Wait 2 seconds before redirecting
            setTimeout(() => {
                window.location.href = `web2.html?data=${answers}`;
            }, 2000);
        }
    </script>
</body>
</html>
```

6.1.4 WEB PAGE 2

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Analysis Results - Dyslexia Detection System</title>
    <style>
        * {
            margin: 0;
            padding: 0;
            box-sizing: border-box;
            font-family: Arial, sans-serif;
```

```
}
body {
    padding: 20px;
    background-color: #f5f5f5;
}
.container {
    max-width: 1200px;
    margin: 0 auto;
}
.header {
    text-align: center;
    margin-bottom: 30px;
    background: white;
    padding: 20px;
    border-radius: 8px;
    box-shadow: 0 2px 4px rgba(0,0,0,0.1);
}
.results {
    background: white;
    padding: 20px;
    border-radius: 8px;
    box-shadow: 0 2px 4px rgba(0,0,0,0.1);
}
.algorithm-card {
    border: 1px solid #ddd;
```

```
    padding: 15px;
    margin-bottom: 15px;
    border-radius: 4px;
}
.accuracy-bar {
    height: 20px;
    background-color: #e9ecef;
    border-radius: 10px;
    margin: 10px 0;
    overflow: hidden;
}
.accuracy-fill {
    height: 100%;
    transition: width 0.3s ease;
}
.high-accuracy { background-color: #28a745; }
.medium-accuracy { background-color: #ffc107; }
.low-accuracy { background-color: #dc3545; }
.final-result {
    margin-top: 20px;
    padding: 20px;
    border-radius: 4px;
    text-align: center;
    font-size: 1.2em;
    font-weight: bold;
```

```
        }
        .non-dyslexia {
            background-color: #d4edda;
            color: #155724;
        }
        .dyslexia {
            background-color: #f8d7da;
            color: #721c24;
        }
        .back-btn {
            background-color: #6c757d;
            color: white;
            padding: 12px 24px;
            border: none;
            border-radius: 4px;
            cursor: pointer;
            font-size: 16px;
            width: 100%;
            margin-top: 20px;
        }
        .back-btn:hover {
            background-color: #5a6268;
        }
    </style>
</head>
```

```
<body>
    <div class="container">
        <div class="header">
            <h1>Analysis Results</h1>
            <p>Dyslexia Detection System</p>
        </div>
        <div class="results">
            <div id="algorithmResults"></div>
            <div id="finalResult"></div>
            <button class="back-btn" onclick="window.location.href='web1.html'">Try
Again</button>
        </div>
    </div>
    <script>
        // Levenshtein Distance Algorithm
        function levenshteinDistance(a, b) {
            if (a.length === 0) return b.length;
            if (b.length === 0) return a.length;
            const matrix = Array(b.length + 1).fill(null)
                .map(() => Array(a.length + 1).fill(null));
            for (let i = 0; i <= a.length; i++) matrix[0][i] = i;
            for (let j = 0; j <= b.length; j++) matrix[j][0] = j;
            for (let j = 1; j <= b.length; j++) {
                for (let i = 1; i <= a.length; i++) {
                    const indicator = a[i - 1] === b[j - 1] ? 0 : 1;
```

```
            matrix[j][i] = Math.min(
                matrix[j][i - 1] + 1,
                matrix[j - 1][i] + 1,
                matrix[j - 1][i - 1] + indicator
            );
        }           }
    return matrix[b.length][a.length];
}
const algorithms = {
    KNN: (input, correct) => {
        const distance = levenshteinDistance(input.toLowerCase(), correct.toLowerCase());
        return Math.max(0, 1 - (distance / Math.max(input.length, correct.length)));
    },
    DecisionTree: (input, correct) => {
        let score = 1.0;
        if (input.toLowerCase() !== correct.toLowerCase()) score -= 0.3;
        if (input.length !== correct.length) score -= 0.2;
        if (!input.match(/^[a-zA-Z]+$/)) score -= 0.1;
        return Math.max(0, score);
    },
    RNN: (input, correct) => {
        const commonChars = input.split('').filter((char, i) =>
            char.toLowerCase() === correct[i]?.toLowerCase()
        ).length;
        return commonChars / Math.max(input.length, correct.length);
```

```
    },
    XGBoost: (input, correct) => {
      const baseScore = algorithms.KNN(input, correct);
      const lengthPenalty = Math.abs(input.length - correct.length) * 0.1;
      return Math.max(0, baseScore - lengthPenalty);
    },
    GradientBoost: (input, correct) => {
      return Object.values(algorithms)
        .slice(0, -1)
        .reduce((sum, algo) => sum + algo(input, correct), 0) / 4;
    }        };
  window.onload = function() {
    const urlParams = new URLSearchParams(window.location.search);
    const data = urlParams.get('data');
    if (!data) {
      window.location.href = 'index.html';
      return;
    }
    const answers = data.split('|').map(item => {
      const [value, correct] = item.split(',').map(decodeURIComponent);
      return { value, correct };
    });
    const algorithmResults = document.getElementById('algorithmResults');
    const finalResult = document.getElementById('finalResult');
    let totalAccuracy = 0;
```

```
const numAlgorithms = Object.keys(algorithms).length;
Object.entries(algorithms).forEach(([name, algorithm]) => {
   let algorithmAccuracy = 0;
   answers.forEach(answer => {
      algorithmAccuracy += algorithm(answer.value, answer.correct);
   });
   algorithmAccuracy = (algorithmAccuracy / answers.length) * 100;
   totalAccuracy += algorithmAccuracy;
   const accuracyClass = algorithmAccuracy >= 80 ? 'high-accuracy' :
                     algorithmAccuracy >= 60 ? 'medium-accuracy' :
                     'low-accuracy';
   algorithmResults.innerHTML += `
      <div class="algorithm-card">
         <h3>${name}</h3>
         <div class="accuracy-bar">
            <div class="accuracy-fill ${accuracyClass}"
                style="width: ${algorithmAccuracy}%"></div>
         </div>
         <p>Accuracy: ${algorithmAccuracy.toFixed(1)}%</p>
      </div>
   `;          });
const averageAccuracy = totalAccuracy / numAlgorithms;
const isDyslexic = averageAccuracy < 80;
finalResult.innerHTML = `
   <div class="final-result ${isDyslexic ? 'dyslexia' : 'non-dyslexia'}">
```

```
                <h3>${isDyslexic ? 'Potential Dyslexia Detected' : 'No Dyslexia Detected'}</h3>
                <p>Average Accuracy: ${averageAccuracy.toFixed(1)}%</p>
                <p>${isDyslexic ?
                    'Consider consulting a specialist for a professional assessment.' :
                    'Your responses indicate typical reading and writing patterns.'}</p>
            </div>
        `;        }
    </script>
</body>
</html>
```

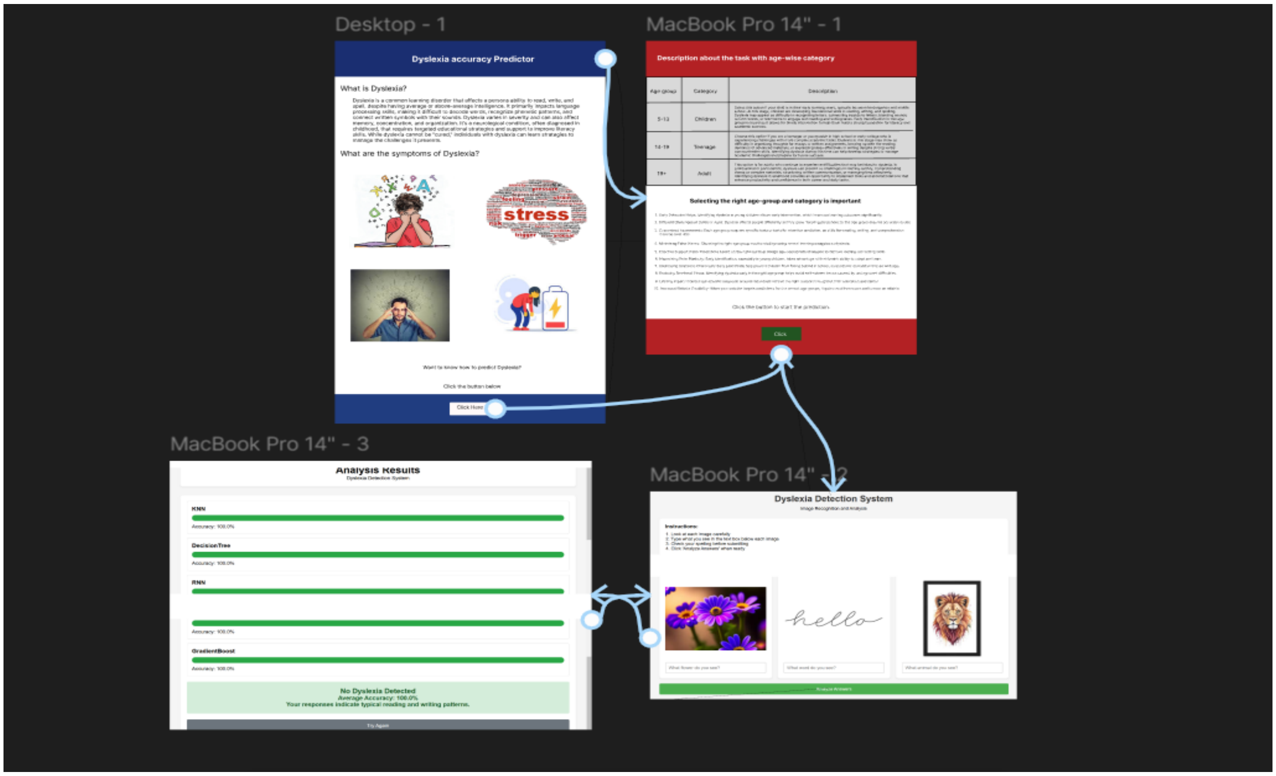
6.2 USER INTERFACE DESIGN

The website features a clean and modern layout with visually appealing illustrations and professional fonts. The first page introduces dyslexia with a concise overview and a prominent navigation button styled to match the theme. The second page uses a card-style layout to categorize age groups, ensuring clarity. The third page includes three image placeholders with clearly labeled text boxes below for input and a centrally positioned "Analyze" button. The result page presents analysis outcomes in a structured format, with a distinct "Retake Test" button for easy access. Consistent color schemes, typography, and responsive design enhance the overall aesthetic.

6.3 USER EXPERIENCE DESIGN

The website provides a seamless and intuitive user journey, starting with an engaging introduction to dyslexia on the first page. Navigation buttons are placed logically and are easy to locate, ensuring smooth transitions between pages. The third page simplifies user interaction by grouping related elements (images, input boxes, and the "Analyze" button) in an accessible layout. The result page displays outcomes in an easy-to-read format, enabling users to understand the analysis quickly. The "Retake Test" button provides feedback by enabling users to revisit and correct inputs if necessary, ensuring a user-friendly and satisfying experience.

6.4 PROTOTYPE DESIGN



SOFTWARE TESTING

7.1 Unit Testing

Unit testing is the first level of testing, where individual components or modules of the project are tested in isolation. The focus is on verifying that each unit of the software performs as expected based on its design.

Objective: Ensure that individual components work correctly in isolation.

Approach: Manual testing of each function or module using predefined inputs and validating outputs.

Outcome: Confirms the reliability of the smallest components of your project.

7.2 Integration Testing

Integration testing involves testing multiple modules or components together to verify that they interact correctly. This can be done by manually simulating scenarios where different modules communicate or depend on each other.

Objective: Validate the communication and data flow between integrated components.

Approach: Identify the interfaces between modules and test their interactions step by step (e.g., frontend and backend communication).

Outcome: Ensures that combined modules work harmoniously to achieve the desired functionality

7.3 System Testing

System testing evaluates the complete and integrated system to ensure it meets the defined requirements. This stage tests the entire application, including all modules, in a simulated environment to mimic real-world use. It covers end-to-end functionality and checks for overall system behavior.

Objective: Confirm that the entire project works as intended in its complete form.

Approach: Execute real-world scenarios based on the functional requirements of the project.

Outcome: Validates that the project is ready for beta testing and can handle user expectations.

7.4 Acceptance Testing

Acceptance testing in the project will be conducted as Beta Testing, where the application is provided to a small group of real-world users. These users interact with the system and provide feedback on its functionality and usability. The focus is on ensuring that the system fulfills the user requirements and is ready for deployment.

Objective: Validate the project from the end-user's perspective.

Approach: Collect feedback from selected users during live testing scenarios.

Outcome: Ensures that the project meets user expectations and is ready for production.

CONCLUSION

In this project, we have successfully implemented a system that leverages advanced machine learning algorithms to predict reading disabilities and analyze eye coordination patterns. By utilizing Gradient Boosting, CNN, and Decision Tree algorithms, we achieved notable results, including a 91.50% accuracy rate in predicting reading disabilities using Gradient Boosting and precise predictions of eye coordination using CNN. The development of a user-friendly website marks a significant step in making these solutions accessible to a broader audience. This platform enables real-time processing of user data and displays the analysis in an intuitive and understandable format. By bridging the gap between technical innovation and real-world usability, this project provides a practical solution for educators, parents, and healthcare professionals who seek to identify and address reading disabilities effectively.

SCOPE FOR FUTURE ENCHANCEMENT

To enhance the project further, we plan to deploy the website on a cloud platform. This deployment will provide a secure, scalable solution for storing user data, enabling the system to handle a large volume of users efficiently. By leveraging the capabilities of cloud technology, the platform will ensure high availability, reliability, and security for all users. With cloud integration, users will be able to create personalized accounts, providing a more customized experience. Each user will have access to their results securely through a username and password authentication system. This authentication mechanism will safeguard user information and maintain privacy.

The use of cloud services will also allow for seamless updates, ensuring that the system remains robust and adaptable to evolving requirements. Additionally, the scalability of cloud infrastructure will support future growth, accommodating an increasing number of users without compromising performance. By incorporating cloud technology, the project will not only improve user experience but also provide a sustainable foundation for long-term development and expansion.

BIBILOGRAPHY

1. PREDICTION OF DYLEXIA USING MACHINE LEARNING ALGORITHMS M.Mahalakshmi1, Dr.K.Merriliance2 Department of Computer Applications, Sarah Tucker College, Thirunelveli-7.

2. Dyslexia Prediction using Machine Learning Isaac Punith Kumar1 Vidya Vikas Institute of Engineering & Technology, Mysore Hemanth Kumar B N2 Vidya Vikas Institute of Engineering & Technology, Mysore

3. Dyslexia Prediction Pallavi Akhade1, Prof Sana Shaikh2, Aditi Ghaitade3, Tanuja Mukane4, Dipti Patil5

4. PROJECT ON DYSLEXIA PREDICTION USING MACHINE LEARNING

ALGORITHMS Abhishek M*1, Hemanth Kumar BN*2

5. Dyslexia Detection: Using Decision Tree Algorithm Rishi Jain, Pranav Patel, Tanaya Jadhav, Kimaya Joshi, Ishita Jasuja

6. Dyslexia Prediction Using Machine Learning Algorithms -A Review February 2023 Kasthuri Stephen

7. Predicting Dyslexia with Machine Learning: A Comprehensive Review of Feature Selection, Algorithms, and Evaluation Metrics Velmurugan S

8. Integrating oversampling and ensemble-based machine learning techniques for an imbalanced dataset in dyslexia screening tests Shahriar Kaisar

9. An ensemble features aware machine learning model for detection and staging of dyslexia Sailaja Mulakaluri1,2, Girisha Gowdra Shivappa2

10. A machine learning approach for dyslexia detection using Turkish audio records A machine learning approach for dyslexia detection using Turkish audio records TUGBERK TAS MUHAMMED ABDULLAH BULBUL ABAS HASIMOGLU YAVUZ MERAL YASIN CALISKAN

11. An Effective Feature Selection and Classification Technique based on Ensemble Learning for Dyslexia Detection Tabassum Gull Jan1(corresponding author), Sajad Mohammad Khan2

12. Fast Algorithm for Dyslexia Detection Boris Nerusil1, Jaroslav Polec1 and Juraj Skunda1

13. An Efficient Machine Learning Model for Prediction of Dyslexia from Eye Fixation Events A. Jothi Prabha1*, R. Bhargavi2 and B. Harish3

14. Machine learning and deep learning performance in classifying dyslexic children's electroencephalogram during writing Ahmad Zuber Ahmad Zainuddin1,2,3,4, Wahidah Mansor1,2,3, Khuan Yoot Lee2,3, Zulkifli Mahmoodin4

15. A Study on different Classification Models for predicting Dyslexia Vani Chakraborty

16. Optimized k-nearest neighbours classifier based prediction of epileptic seizures Himayavardhini Jagath Prasad1, Roji Marjorie S.2

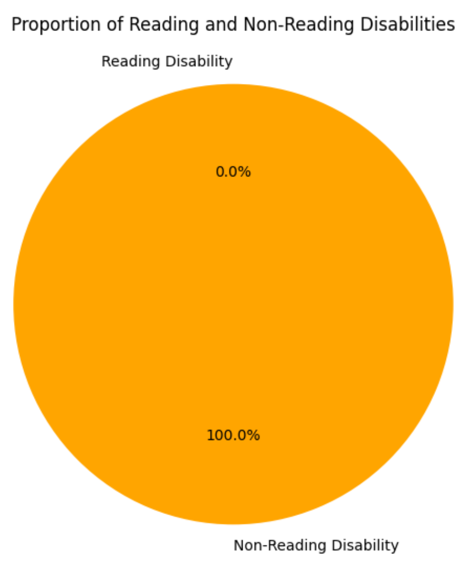
17. Medical Health Big Data Classification Based on KNNClassification Algorithm Wenchao Xing 1, Yilin Bei2*

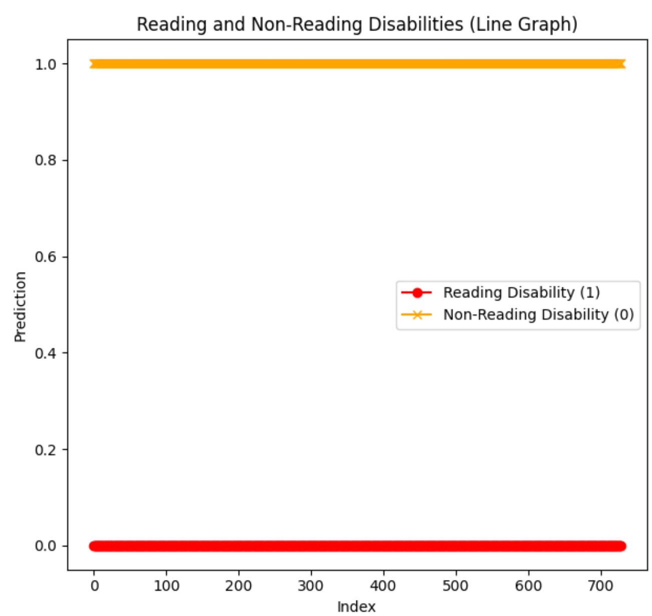
APPENDIX

a) SNAPSHOTS

KNN Algorithm

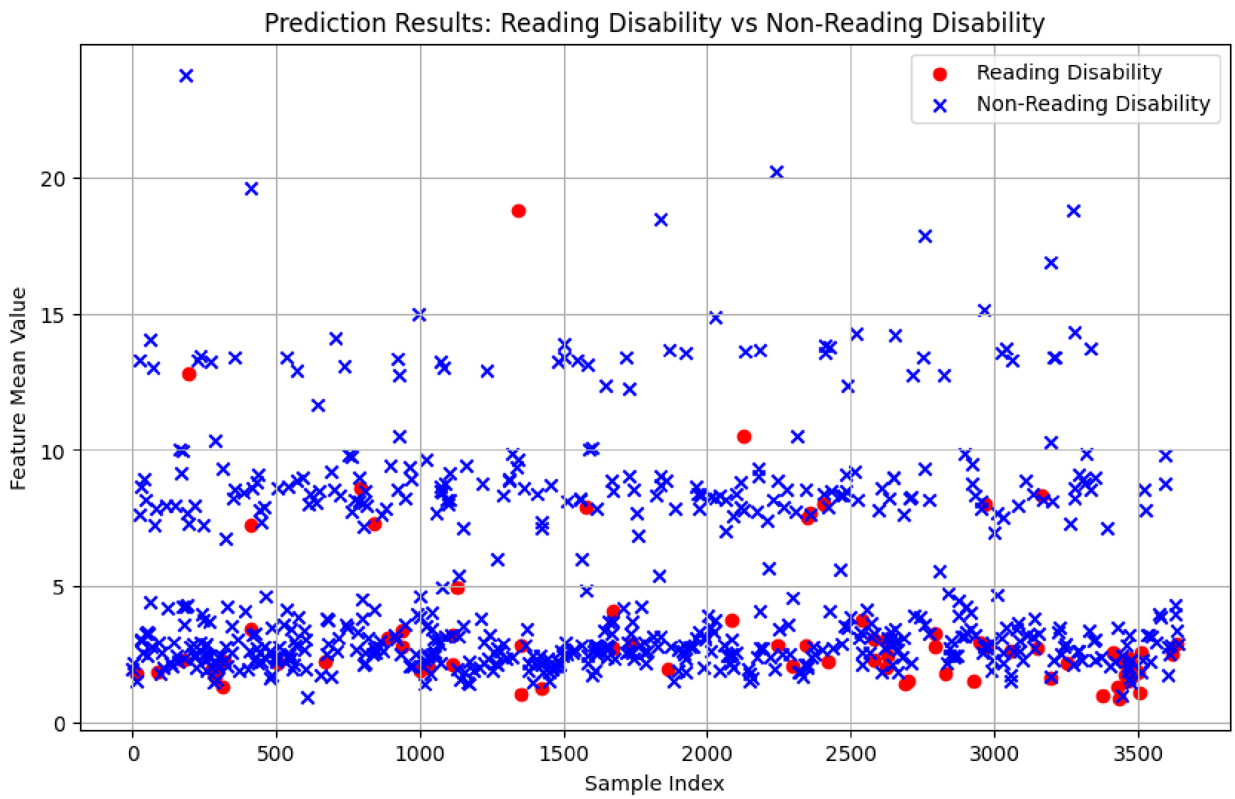
```
Accuracy: 1.0
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       729

    accuracy                           1.00       729
   macro avg       1.00      1.00      1.00       729
weighted avg       1.00      1.00      1.00       729
```





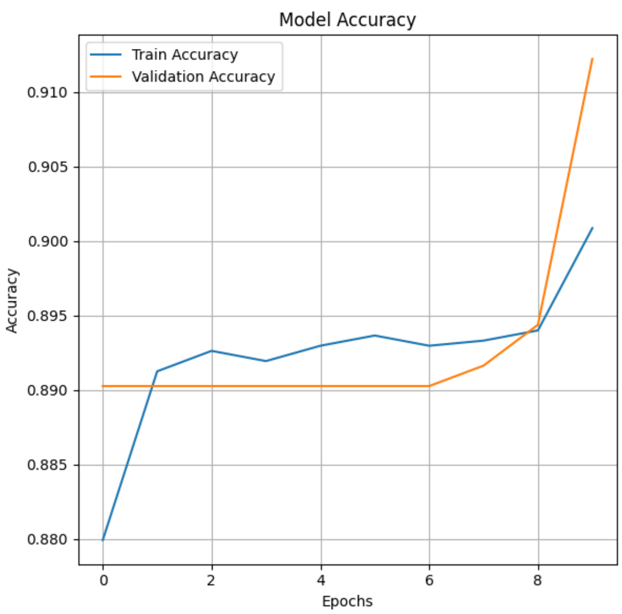
Decision Tree Algorithm

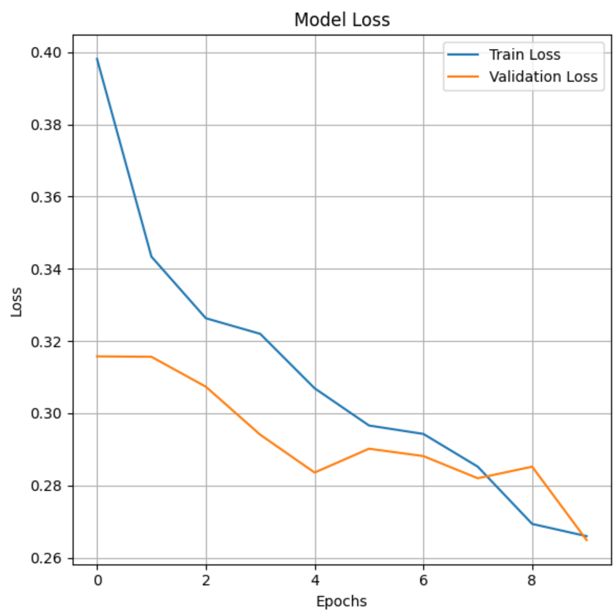
```
Fitting 5 folds for each of 90 candidates, totalling 450 fits
Optimized Accuracy of Decision Tree Classifier: 86.15%
Total individuals with Reading Disability: 79
Total individuals without Reading Disability: 650
```



RNN (Recurrent Neural Networks)

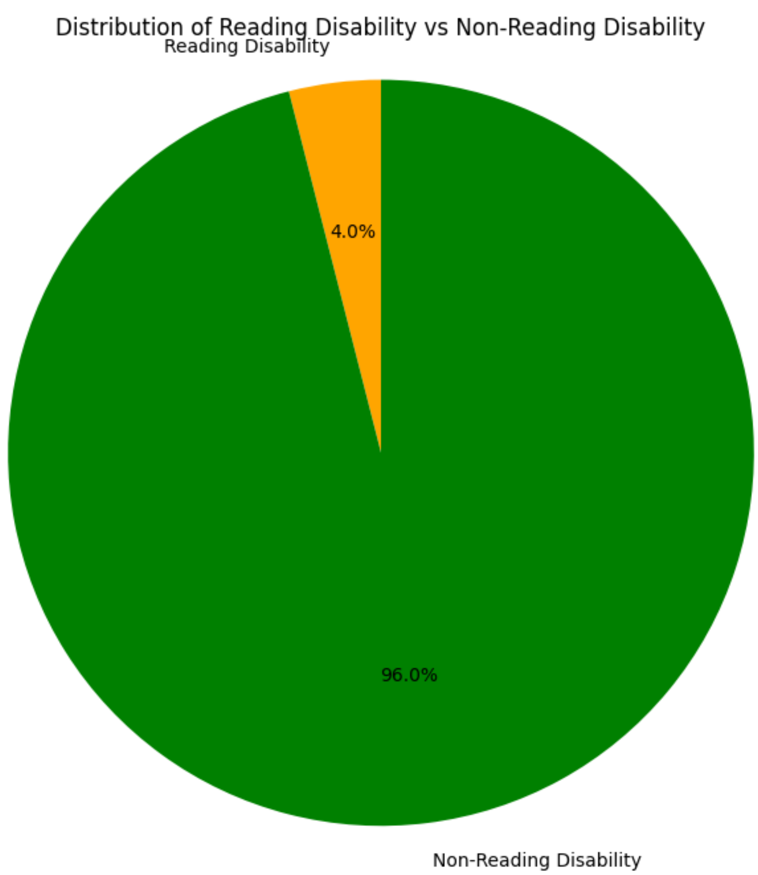
```
  super().__init__(**kwargs)
Epoch 1/10
92/92 ──────────────────────────────── 5s 13ms/step - accuracy: 0.8364 - loss: 0.4769 - val_accuracy: 0.8903 - val_lo
Epoch 2/10
92/92 ──────────────────────────────── 1s 7ms/step - accuracy: 0.8900 - loss: 0.3502 - val_accuracy: 0.8903 - val_los
Epoch 3/10
92/92 ──────────────────────────────── 1s 7ms/step - accuracy: 0.8875 - loss: 0.3338 - val_accuracy: 0.8903 - val_los
Epoch 4/10
92/92 ──────────────────────────────── 1s 7ms/step - accuracy: 0.8938 - loss: 0.3212 - val_accuracy: 0.8903 - val_los
Epoch 5/10
92/92 ──────────────────────────────── 1s 7ms/step - accuracy: 0.8973 - loss: 0.2893 - val_accuracy: 0.8903 - val_los
Epoch 6/10
92/92 ──────────────────────────────── 1s 8ms/step - accuracy: 0.8785 - loss: 0.3230 - val_accuracy: 0.8903 - val_los
Epoch 7/10
92/92 ──────────────────────────────── 1s 7ms/step - accuracy: 0.8932 - loss: 0.2918 - val_accuracy: 0.8903 - val_los
Epoch 8/10
92/92 ──────────────────────────────── 1s 8ms/step - accuracy: 0.9035 - loss: 0.2692 - val_accuracy: 0.8916 - val_los
Epoch 9/10
92/92 ──────────────────────────────── 1s 8ms/step - accuracy: 0.8935 - loss: 0.2747 - val_accuracy: 0.8944 - val_los
Epoch 10/10
92/92 ──────────────────────────────── 1s 8ms/step - accuracy: 0.8933 - loss: 0.2733 - val_accuracy: 0.9122 - val_los
Test Accuracy: 91.22%
23/23 ──────────────────────────────── 1s 27ms/step
Reading Disability: 32
Non-Reading Disability: 697
```





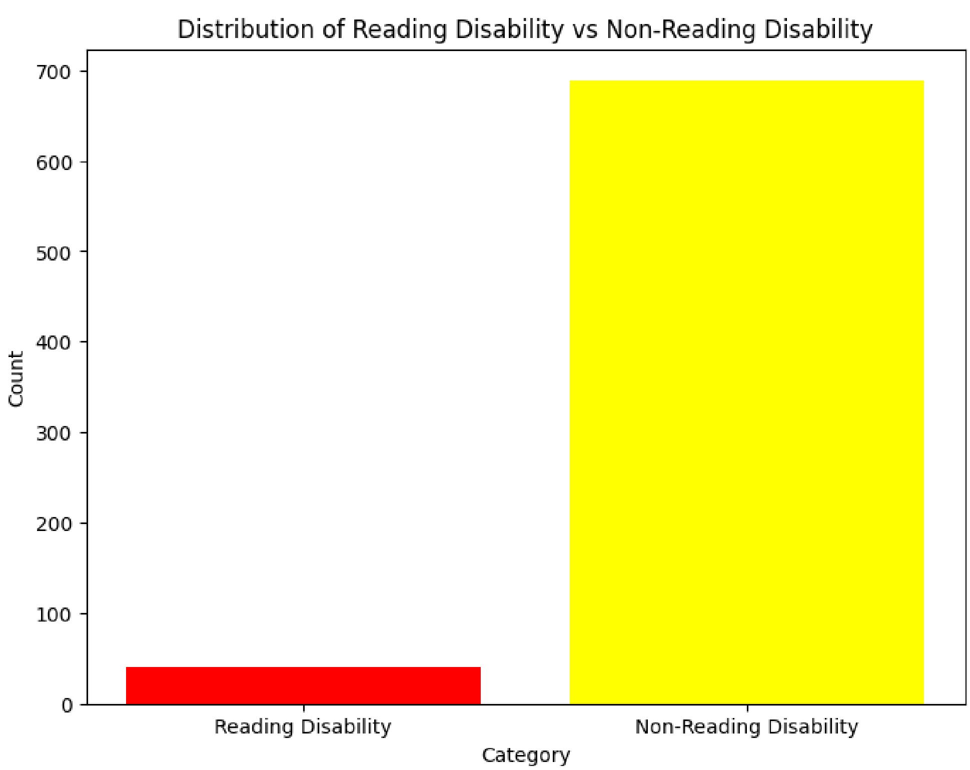
XGBOOST Algorithm

```
Test Accuracy: 91.36%

Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.99      0.95       649
           1       0.79      0.29      0.42        80

    accuracy                           0.91       729
   macro avg       0.86      0.64      0.69       729
weighted avg       0.90      0.91      0.89       729


Confusion Matrix:
[[643   6]
 [ 57  23]]
```

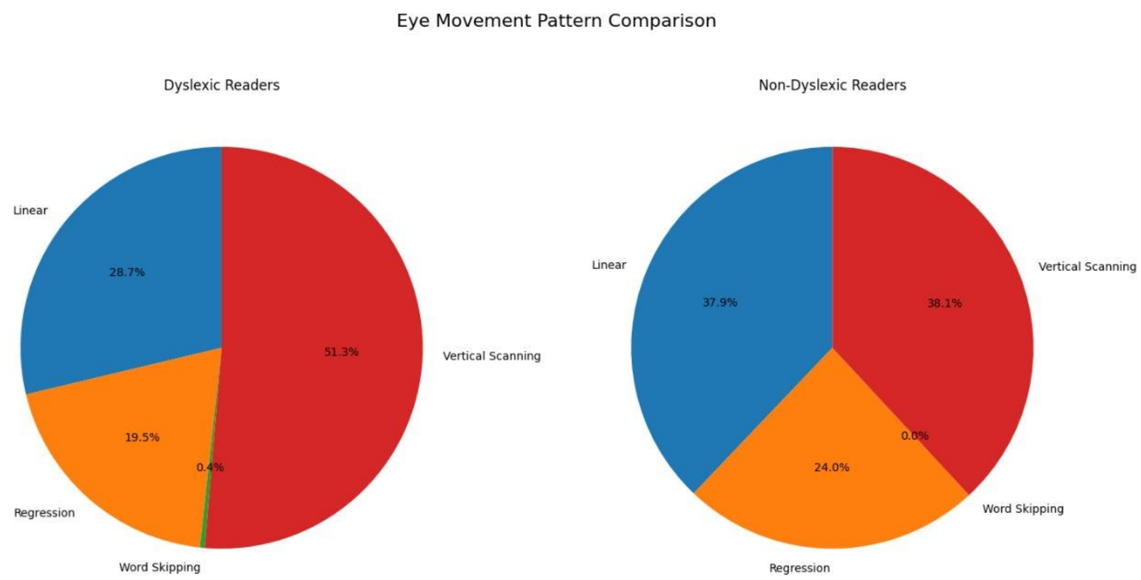


GRADIENT BOOSTING Algorithm

```
Test Accuracy: 91.50%

Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.98      0.95       649
           1       0.72      0.36      0.48        80

    accuracy                           0.91       729
   macro avg       0.83      0.67      0.72       729
weighted avg       0.90      0.91      0.90       729


Confusion Matrix:
[[638  11]
 [ 51  29]]
```

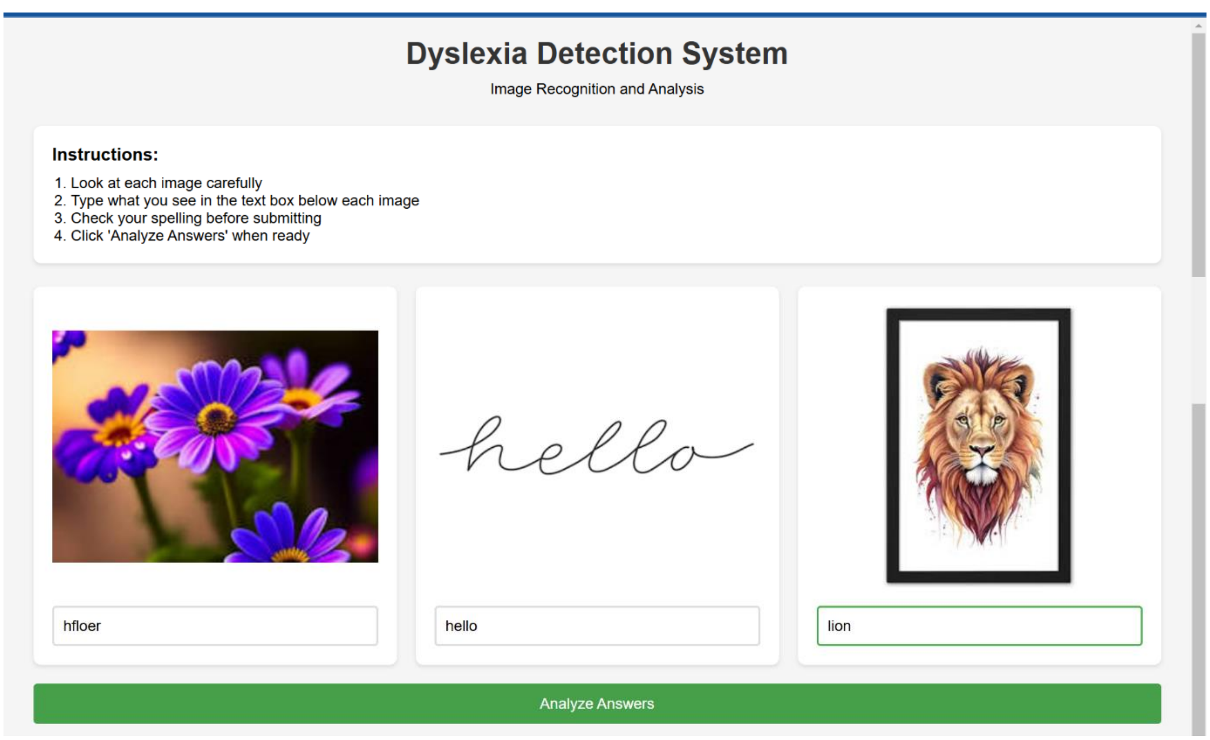


CNN Algorithm

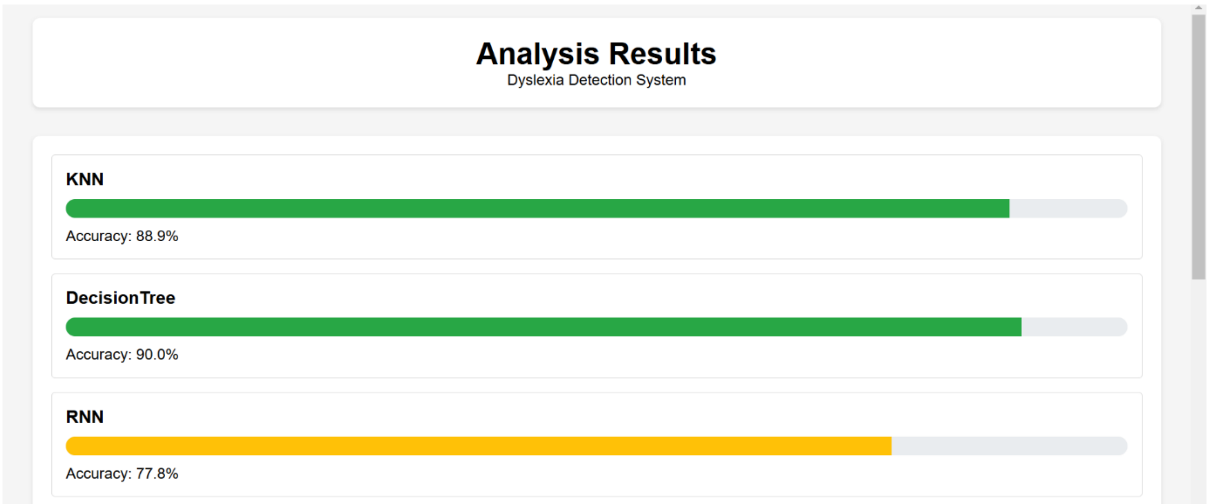


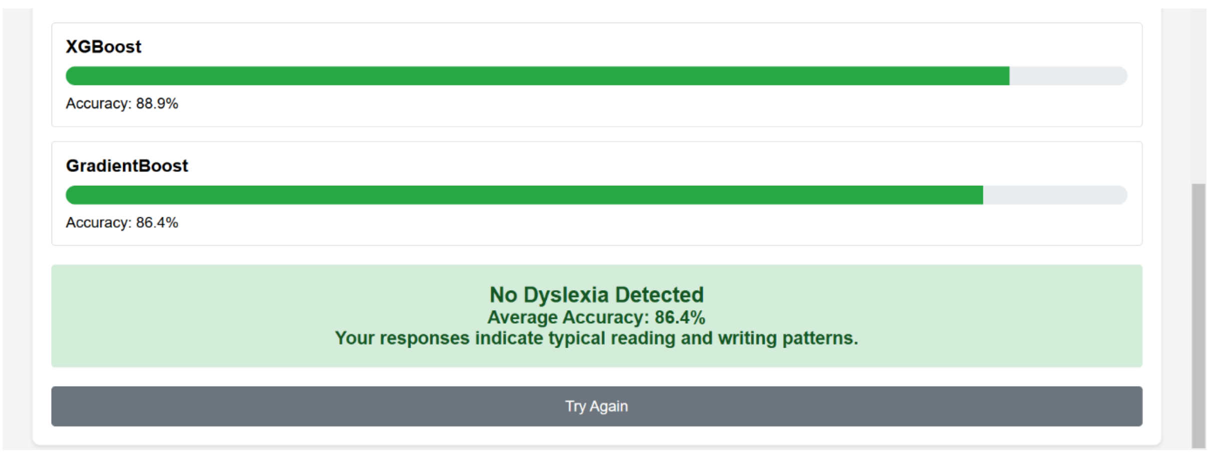
WEBSITE OVERVIEW

INPUT PAGE



OUTPUT PAGE


Analysis Results
Dyslexia Detection System
KNN
Accuracy: 88.9%
DecisionTree
Accuracy: 90.0%
RNN
Accuracy: 77.8%


XGBoost
Accuracy: 88.9%
GradientBoost
Accuracy: 86.4%
No Dyslexia Detected
Average Accuracy: 86.4%
Your responses indicate typical reading and writing patterns.
Try Again