Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	06 November 2023
Team ID	Team-592035
Project Name	Alzheimer Disease Prediction
Maximum Marks	4 Marks

Technical Architecture:

1. Data Collection and Preprocessing:

Data Sources:Obtain datasets containing medical images (such as brain scans) with associated labels indicating the presence or absence of Alzheimer's disease.
Data Preprocessing:Perform image preprocessing to standardize and enhance the quality of the input data.Split the dataset into training and testing sets.

2. Xception Model Architecture:
Deep Learning Model:Utilize the Xception architecture, a convolutional neural network (CNN) designed for image classification tasks.
Transfer learning: Pre-train the Xception model on a large dataset (e.g., ImageNet) to capture general features and then fine-tune it on the Alzheimer's dataset for specific patterns.

3. Model Training and Evaluation:
Training:Train the Xception model on the preprocessed dataset using appropriate deep learning frameworks (TensorFlow, Keras).
Use a suitable loss function, optimizer, and metrics for binary classification (Alzheimer's or non-Alzheimer's).
Evaluation:Assess the model's performance using metrics like accuracy, precision, recall, and F1 score on the testing set.Employ techniques such as cross-validation for robust evaluation.

4. Flask App Development:

Backend:Develop a Flask application to serve as the backend of the system.Create endpoints for receiving image data for prediction.Model Integration:Integrate the trained Xception model into the Flask app. This may involve serializing the model and loading it within the Flask application.

5. Frontend and User Interface:

User Input:Design a user interface (UI) for users to input medical images for prediction.Accept image uploads or links from users.

6. Deployment:

Cloud Deployment: Deploy the Flask application to a cloud platform (e.g., AWS, Google Cloud, or Azure) for scalability and accessibility.

Scalability: Ensure the architecture is scalable by leveraging cloud services and optimizing the application for potential increases in usage.

7. Security:

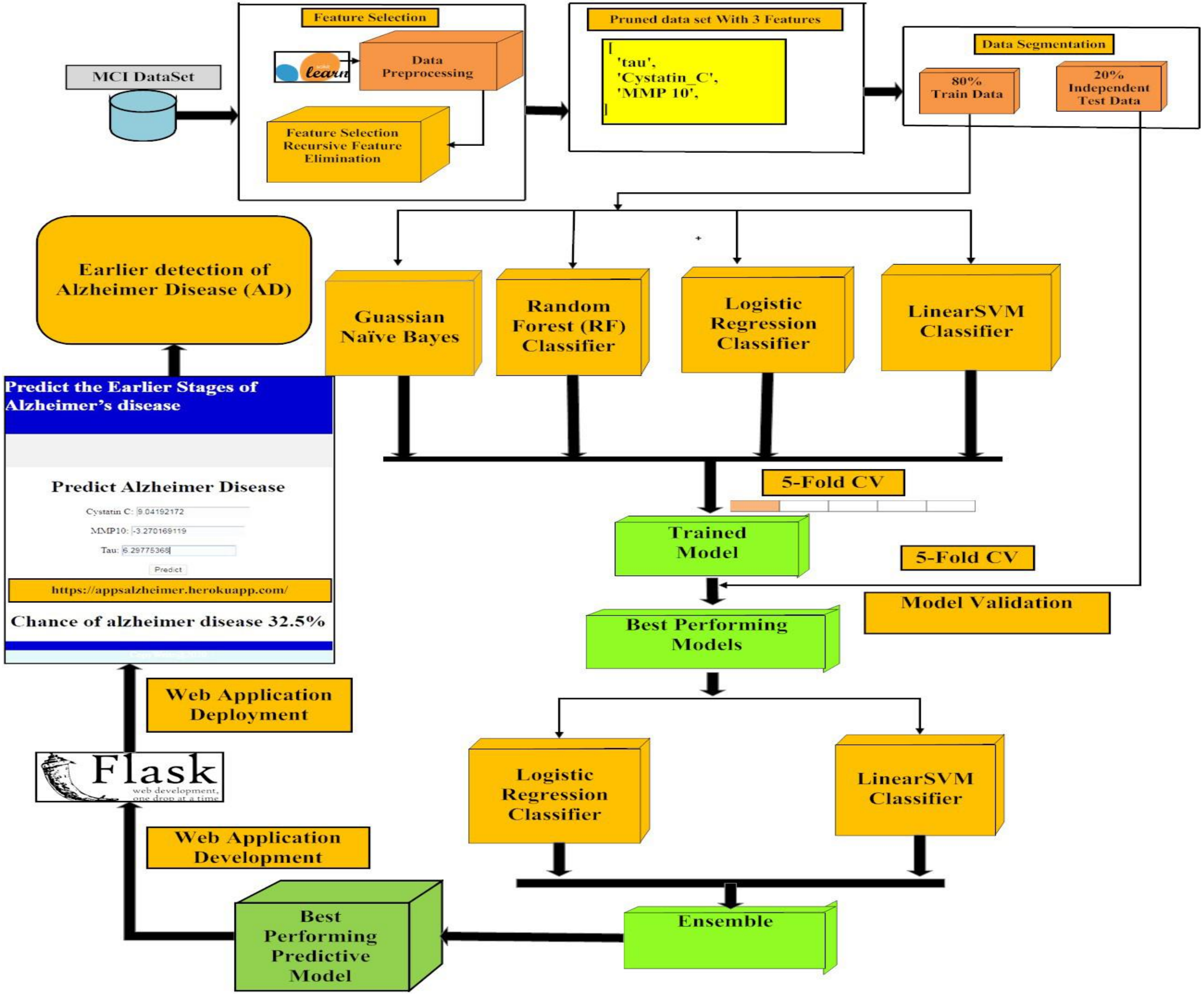
Data Encryption: Implement encryption for data transmission between the user and the server.

Access Controls: Employ access controls and authentication mechanisms to secure the Flask app.

8. Monitoring and Logging:

Logging: Implement logging mechanisms to record user interactions, errors, and predictions.

Monitoring: Set up monitoring tools to track the performance and health of the deployed system.


MCI DataSet
Feature Selection
learn
Data Preprocessing
Feature Selection Recursive Feature Elimination
Pruned data set With 3 Features
[
'tau',
'Cystatin_C',
'MMP 10',
]
Data Segmentation
80% Train Data
20% Independent Test Data
Earlier detection of Alzheimer Disease (AD)
Guassian Naïve Bayes
Random Forest (RF) Classifier
Logistic Regression Classifier
LinearSVM Classifier
5-Fold CV
Trained Model
5-Fold CV
Model Validation
Best Performing Models
Web Application Deployment
Flask
web development, one drop at a time
Web Application Development
Logistic Regression Classifier
LinearSVM Classifier
Ensemble
Best Performing Predictive Model

Predict the Earlier Stages of Alzheimer's disease

Predict Alzheimer Disease

Cystatin C: 9.04192172

MMP10: -3.270169119

Tau: 6.29775368

https://appsalzheimer.herokuapp.com/

Chance of alzheimer disease 32.5%

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	Data Collection	Gathering patient data for training and testing	Electronic Health Records (EHR), CSV, etc.
2.	Data Preprocessing	Cleaning and formatting the data for model training	Python (Pandas, NumPy), Data Augmentation
3.	Feature Extraction	Identifying relevant features from patient data	Deep Learning Models (e.g., CNN for images)
4.	Deep Learning Model-1	Initial neural network for pattern recognition	Convolutional Neural Network (CNN)
5.	Deep Learning Model-2	Refinement of the neural network for improved accuracy	Recurrent Neural Network (RNN) or LSTM
6.	Model Training	Training the deep learning models with labeled data	TensorFlow, PyTorch
7.	Model Evaluation	Assessing the performance of the trained models	Precision, Recall, F1 Score, ROC-AUC
8.	Hyperparameter Tuning	Optimization of model parameters for better results	Grid Search, Random Search
9.	External Data Sources	Integration of external datasets for enriched training	Alzheimer's Disease Neuroimaging Initiative (ADNI)
10.	Patient Interface	User interface for patients to input relevant data	Web UI, Mobile App

11.	Healthcare Provider Interface	Interface for healthcare professionals to access predictions	Web UI, API
12.	Cloud Storage	Storage of large datasets and model checkpoints	AWS S3, Google Cloud Storage, etc.
13.	Deployment	Hosting the prediction system for real-world use	Docker, Kubernetes, Cloud Services (AWS, GCP)
14.	Model Interpretability	Understanding and explaining model decisions	SHAP (SHapley Additive exPlanations), LIME

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used for model development	TensorFlow, PyTorch, Keras, Scikit-learn, etc.
2.	Security Implementations	Security and access control measures implemented	SSL/TLS, SHA-256, Encryptions, IAM Controls, OWASP, etc.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Docker, Kubernetes, AWS Lambda, etc.

S.No	Characteristics	Description	Technology
4.	Availability	Justify the availability of application	Load balancers, Distributed servers, High Availability (HA) configurations, etc.
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Caching mechanisms, Content Delivery Networks (CDN), Efficient algorithms, etc.

References:

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10280208/#:~:text=The%20proposed%20model%20suggests%20a,Alzheimer's%20Disease%20in%20this%20model.

https://www.frontiersin.org/articles/10.3389/fnagi.2019.00220/full

https://www.ibm.com/cloud/architecture

https://aws.amazon.com/architecture

https://www.researchgate.net/publication/342853050_Multi_Disease_Prediction_Model_by_using_Machine_Learning_and_Flask_API