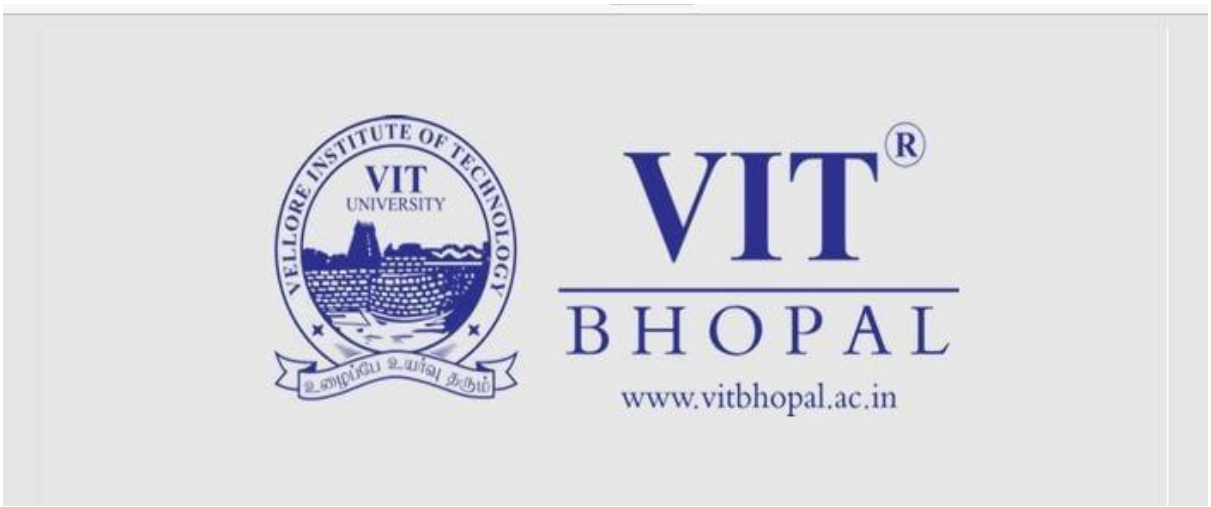
PROJECT REPORT

Alzheimer’s Disease Detection Using Longitudinal Data



Course / Department: — introduction to programming and problem solving

Date: —25.11.2025

TANISHA DESHPANDE

CONTENTS

2. Introduction

- Alzheimer’s Disease (AD) is a progressive neurodegenerative disorder affecting memory, cognitive function, and behavior. Early detection is crucial

because interventions are most effective in the beginning stages, before severe neuronal damage occurs.

- Cross-sectional data provides only a single snapshot of a patient's condition, which limits the understanding of disease progression. Longitudinal data, however, captures multiple observations over time, revealing trends and subtle changes that help machine learning models differentiate between stable cognitive decline and early Alzheimer's onset.

In this project, I have built a machine-learning model to analyze longitudinal clinical and biomarker data and classify whether a patient has Alzheimer's disease.

3. Problem Statement

- Alzheimer's is often diagnosed late due to subtle early symptoms. Traditional diagnosis relies heavily on clinical observation, which can be subjective.

- The problem:How can we use longitudinal data (multiple time-based observations) to detect early signs of Alzheimer's more accurately?

Goal:

Build a model that learns patterns of progression across visits and predicts whether a subject is cognitively normal, mildly impaired, or has Alzheimer's.

4. Functional Requirements

1. Data ingestion

Load longitudinal .csv files containing patient data across multiple visits.

2. Data preprocessing

- Handle missing values
- Normalize numerical features
- Encode categorical variables
- Group patient data by subject ID

3. Model Building

Train ML/DL model (LSTM / Random Forest / Gradient Boosting) on processed longitudinal features.

4. Prediction

Given temporal features for a patient, the model predicts cognitive status.

5. Performance Evaluation

Provide accuracy, confusion matrix, loss curves, etc.

6. Visualization

Graphical results of training and trend patterns.

5. Non-Functional Requirements

1)Accuracy: Model must maintain ≥70% accuracy on validation data.

2)Scalability: Should handle increasing rows of longitudinal data.

3)Reliability: Stable performance across multiple training runs.

4)Efficiency: Reasonable training time (<5 mins on standard hardware).

5)Usability: Clean, modular code; reproducible steps.

6)Explainability: Clear feature importance / trend analysis.

6. System Architecture

Input → Preprocessing → Feature Engineering → Model Training → Evaluation → Predictions

Flow:

1. CSV data loaded
2. Preprocessing pipeline
3. Time-series structuring (group by subject ID + sort by VISIT)
4. Model (LSTM / classical ML)

5. Predictions & metrics

6. Graphs for analysis

7. Design Diagrams

a. Use Case Diagram

- Use cases: Upload data → Preprocess → Train Model → Evaluate → Predict outcome

b. Workflow Diagram

1. Start
2. Load longitudinal CSV
3. Clean & preprocess
4. Split into train/validation
5. Train model
6. Compute metrics
7. Save predictions
8. End

c. Sequence Diagram

- User → System: Upload CSV
- System → Preprocessor: Clean data
- Preprocessor → Model: Provide structured sequences
- Model → System: Predictions
- System → User: Results/metrics

d. Class / Component Diagram

- Components:
  - DataLoader
  - Preprocessor
  - FeatureExtractor
  - ModelTrainer
  - Evaluator
  - Visualizer

8. Design Decisions & Rationale

1)Why longitudinal data?

→ Captures disease progression more accurately than single visit data.

2)Why LSTM or time-series models?

→ They understand temporal patterns across visits — exactly like human doctors observing decline over years.

3)Why normalization?

→ Biomarkers and cognitive scores have different scales.

4)Why grouping by subject ID?

→ ML models must learn how one patient changes over time, not compare different patients' visits as unrelated samples.

9. Implementation Details

Data Source: Kaggle

Longitudinal CSV containing features such as:

- AGE
- GENDER
- MMSE score
- CDR
- Visit date/time
- Clinical diagnosis label (CN / MCI / AD)

Preprocessing Steps

Remove empty rows

Forward-fill small gaps

Normalize MMSE, volumes, biomarkers

Convert diagnosis to integers

Group by patient ID and sort by VISIT_MONTH

Pad sequences to equal length

10. Screenshots / Results

```
Data Preview:
  Subject ID         MRI ID        Group  Visit  MR Delay M/F Hand  Age  EDUC  \
0  OAS2_0001  OAS2_0001_MR1  Nondemented      1         0   M    R   87    14
1  OAS2_0001  OAS2_0001_MR2  Nondemented      2       457   M    R   88    14
2  OAS2_0002  OAS2_0002_MR1     Demented      1         0   M    R   75    12
3  OAS2_0002  OAS2_0002_MR2     Demented      2       560   M    R   76    12
4  OAS2_0002  OAS2_0002_MR3     Demented      3      1895   M    R   80    12

   SES  MMSE  CDR  eTIV   nWBV    ASF
0  2.0  27.0  0.0  1987  0.696  0.883
1  2.0  30.0  0.0  2004  0.681  0.876
2  NaN  23.0  0.5  1678  0.736  1.046
3  NaN  28.0  0.5  1738  0.713  1.010
4  NaN  22.0  0.5  1698  0.701  1.034

Data Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 373 entries, 0 to 372
Data columns (total 15 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Subject ID  373 non-null    object
 1   MRI ID      373 non-null    object
 2   Group       373 non-null    object
 3   Visit       373 non-null    int64
 4   MR Delay    373 non-null    int64
 5   M/F         373 non-null    object
 6   Hand        373 non-null    object
 7   Age         373 non-null    int64
 8   EDUC        373 non-null    int64
 9   SES         354 non-null    float64
 10  MMSE        371 non-null    float64
```

```
 8   EDUC         373 non-null    int64
 9   SES          354 non-null    float64
 10  MMSE         371 non-null    float64
 11  CDR          373 non-null    float64
 12  eTIV         373 non-null    int64
 13  nWBV         373 non-null    float64
 14  ASF          373 non-null    float64
dtypes: float64(5), int64(5), object(5)
memory usage: 43.8+ KB
None

Model trained successfully!

Accuracy: 0.84

Confusion Matrix:
 [[ 0  1 10]
 [ 0 32  0]
 [ 0  1 31]]

Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        11
           1       0.94      1.00      0.97        32
           2       0.76      0.97      0.85        32

    accuracy                           0.84        75
   macro avg       0.57      0.66      0.61        75
weighted avg       0.72      0.84      0.78        75
```

11. Testing Approach

1. Validation testing: During training

2. Hold-out validation set: 20–30% of dataset

3.longitudinal performance comparison

4. Confusion matrix analysis: Error distribution

5. Manual sanity checks:

Predict on random sequences

Predict on artificially modified values (e.g., reduced MMSE)

12. Challenges Faced

- Small dataset size → Hard for deep learning to generalize
- Missing values in longitudinal visits → Required careful imputation
- Different visit lengths → Needed padding or truncating
- Understanding time-series behavior → Harder than cross-sectional ML

13. Learnings & Key Takeaways

Longitudinal data can uncover subtle trends invisible in single-visit data.

Machine learning for healthcare requires thoughtful preprocessing.

Biomarkers + cognitive scores together improve predictions.

Even simple models can outperform deep models on small data.

14. Future Enhancements

Use larger datasets (ADNI, OASIS longitudinal sets)

Add MRI-derived features using CNN + LSTM hybrid

Integrate demographic + genetic data (ApoE4)

Deploy model as a medical decision-support system

Use transformer-based time-series models

Add uncertainty estimation for clinical reliability

15. References

Alzheimer’s Disease Neuroimaging Initiative (ADNI)

OASIS Longitudinal Dataset

Scholarly articles on time-series modeling for neurodegeneration

Deep learning frameworks documentation

oasis_longitudinal.csv

Kaggle dataset

1. Problem Statement

AD is a degenerative neurologic condition leading to loss of memory, decline in cognition, and eventually the loss of independence.

Early detection is important but often problematic because of the subtle and overlapping symptoms.

The aim in this problem is to *use machine learning to predict dementia status* (Demented vs. Non-Demented) based on *longitudinal clinical data*.

This allows for early intervention and assists doctors in decision-making.

---

2. Scope of the Work

- Build an ML pipeline using structured patient data

Detect patterns associated with cognitive decline.

- Provide predictions regarding dementia likelihood

• Early screening might be facilitated-without replacing clinical diagnosis

- Train and evaluate the Random Forest model Not included: Magnetic Resonance Imaging processing