Optimization Sprint Report

XPredators

Name	University	NIC
Rivindu Ashinsa	IIT	200631103340
Lakindu Minosha	IIT	200633902583
Thuan Naheem	IIT	200601703988

1. Data Exploration and Process Flow

1.1. Dataset Description

The dataset used for this project was downloaded from Google Drive using gdown and loaded into a pandas DataFrame. It contains a large number of variables from the NACC (National Alzheimer's Coordinating Center) dataset including:

Medical history variables

Psychological assessments

Cognitive diagnosis labels

Medication usage

Vital measurements

Lifestyle variables

General demographic attributes

1.2. The target variable used was:

DEMENTED

0 = Non-demented

1 = Demented

(If not present, fallback was NACCUDSD.)

1.3. Process Flow Followed in Notebook

i. Load dataset from Google Drive
ii. Explore dtypes and identify categorical columns
iii. Remove all medical-related features
iv. You manually created a list of medical variables from health history, clinical exams, and medications.
v. Drop all non-integer (object) columns
vi. Select numeric-only features
vii. Define features (X) and target (y)
viii. Handle missing values using median imputation
ix. Apply Variance Threshold feature reduction
x. Apply PCA to explore principal components
xi. Split train and test set
xii. Train Logistic Regression and Random Forest models
xiii. Evaluate model performance
xiv. Select Logistic Regression as final model

2. Feature Engineering

2.1. Non-medical Features Selected

You removed ALL medical variables including:

- Health conditions (stroke, diabetes, hypertension, etc.)
- Physical exam attributes (BP, BMI, pulse)
- Medication-related variables (DRUG1 to DRUG40)
- Clinical medical condition diagnoses

This left only non-medical, non-object, numeric attributes, which are:

- Lifestyle
- Basic demographics
- Cognitive screening numerical scores
- Behavioral numeric responses
- Derived numeric codes from survey responses

2.2. Feature Reduction Techniques Used

A. Variance Threshold (threshold = 1)

- Removed features with very low variance (<1)
- Helps eliminate features that do not vary between samples

Remaining columns: printed in notebook

Dropped columns: also printed

B. Principal Component Analysis (PCA, 95% variance)

Scaled data using StandardScaler and applied PCA.

Extracted:

Explained Variance Ratio

Top contributing features to PC1

Loadings matrix

This step was used only for analysis, not for model training.

2.3. Finalized Features

- All non-medical numeric features
- After removing medical vars
- After dropping object columns
- After VarianceThreshold filtering
- Median-imputed missing values

These form your final X_selected dataset.

3. Data Preprocessing

3.1. Steps Performed

Step	Description	Justification
Remove medical columns	Dropped 180+ medical/clinical variables	Required by project to use only non-medical factors
Drop object columns	Removed string/categorical data	Enables fast ML training without encoding
Numeric-only filtering	Ensures models receive only numerical inputs	Avoids encoding complexity
Missing value handling	Filled missing values using **median**	Robust method for skewed numeric data
VarianceThreshold	Removed low-variance features	Improves generalization and reduces noise
StandardScaler (for PCA)	Standardized features	PCA requires normalized feature scales
Train-test split (70–30)	Random split with seed 42	Ensures reproducibility

3.2. Train-Test Split

train_test_split(X_selected, y, test_size=0.3, random_state=42)

4. Model Building

4.1. Models Trained

Logistic Regression

```
lr_model = LogisticRegression()
lr_model.fit(X_train, y_train)
lr_preds = lr_model.predict(X_test)

print(classification_report(y_test, lr_preds))
```

- Used **default parameters**
- Fast and interpretable
- Works well on linearly separable data

Random Forest Classifier

```
rf_model = RandomForestClassifier(
    n_estimators=300,
    max_depth=None,
    random_state=42
)

rf_model.fit(X_train, y_train)
rf_preds = rf_model.predict(X_test)
```

Parameters used:

n_estimators = 300

max_depth = None

random_state = 42

Justification:

- RF can capture non-linear patterns
- Handles feature interactions well

No hyperparameter tuning was performed (e.g., GridSearchCV).- No needed

5. Model Evaluation

5.1. Evaluation Metrics Used

- Classification Report
- Precision
- Recall
- F1-score
- Confusion Matrix

Justifications:

- Dementia prediction is binary classification
- F1-score is important due to medical nature (harmful to misclassify dementia)
- Confusion matrix helps interpret false negatives (critical in screening)

5.2. Model Performance Summary

Logistic regression

```
              precision    recall  f1-score   support

           0       0.94      0.93      0.93     41285
           1       0.84      0.85      0.84     17274

    accuracy                           0.91     58559
   macro avg       0.89      0.89      0.89     58559
weighted avg       0.91      0.91      0.91     58559
```

Random Forest Classifier

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     41285
           1       1.00      1.00      1.00     17274

    accuracy                           1.00     58559
   macro avg       1.00      1.00      1.00     58559
weighted avg       1.00      1.00      1.00     58559
```

5.3. Model Comparison

Model	Accuracy	Notes
Logistic Regression	91%	Simpler, generalizes better, faster
Random Forest	100%	High performance but more complex

Final Model Selected:

Logistic Regression

Justifications:

- Simpler linear model → less risk of overfitting
- Very high performance
- More interpretable

6. Explainability & Model Interpretability

6.1. Explainability Techniques Used

Used:

- **PCA loadings**
- **Top PC1 contributing features**

These showed **which variables contribute most to data variance** and likely to model decisions.

6.2. Insights from Explainability

- Certain non-medical factors strongly influence PC1
- PCA helped validate that **non-medical features still carry strong signals for dementia prediction**

```
Top PC1 Features:
 DELIRIF     0.069433
MSAIF       0.069433
HIVIF       0.069426
FTLDMOIF    0.069420
SCHIZOIF    0.069418
IMPSUBIF    0.069412
EPILEPIF    0.069409
PTSDDXIF    0.069394
BIPOLDIF    0.069381
ESSTREIF    0.069347
Name: PC1, dtype: float64
Explained Variance Ratio: [0.29840745 0.15479302 0.07302789 0.05531144 0.04578605 0.03658191
 0.02853694 0.02459547 0.01600897 0.01272589 0.01153811 0.0103014
 0.00976755 0.00832827 0.00770637 0.0072006  0.00652689 0.00616085
 0.00533562 0.00501557 0.00461612 0.00453236 0.00429953 0.00418977
 0.00403184 0.00390578 0.00384545 0.00365486 0.00340461 0.00331237
 0.00304659 0.00292635 0.0028285  0.00278748 0.00272212 0.00264902
 0.00257168 0.00254132 0.00240037 0.00232889 0.0022083  0.00214355
 0.00212024 0.00206038 0.0020463  0.00193124 0.00186927 0.00183369
 0.00179276 0.00168254 0.00164998 0.00163572 0.0016002  0.00157004
 0.00155105 0.00153027 0.00149196 0.0014637  0.00145489 0.00141947
 0.00137044 0.00134828 0.00132105 0.00129893 0.0012712  0.00126034
 0.00124508 0.00122689 0.00120883 0.00118568 0.00115409 0.00112483
 0.00110333 0.0010827  0.0010356  0.00098704 0.00095603 0.00093203
 0.00092495 0.00091203 0.0009021  0.00089305]
```

7. GitHub Repo Link

https://github.com/Team-XPredators/Dementia-Prediction-xpredators