Optimization Sprint Report

XPredators

Name	University	NIC
Rivindu Ashinsa	IIT	200631103340
Lakindu Minosha	IIT	200633902583
Thuan Naheem	IIT	200601703988
Dulina Samarathunga	IIT	200626402870

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 Dimensionality reduction using correlations
xi. Apply PCA to explore principal components
xii. Split train and test set
xiii. Train Logistic Regression and Random Forest models
xiv. Evaluate model performance
xv. 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
Dimensionality reduction using correlations	Eliminated redundant and minimally informative features.	To mitigate multicollinearity and reduce noise for better generalization.
StandardScaler (for PCA)	Standardized features	PCA requires normalized feature scales
Train-test split (70–30)	Random split with seed 42	Ensures reproducibility

Feature Count after all the pre-processing : 30

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).- Not needed

5. Model Evaluation

5.1. Evaluation Metrics Used

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

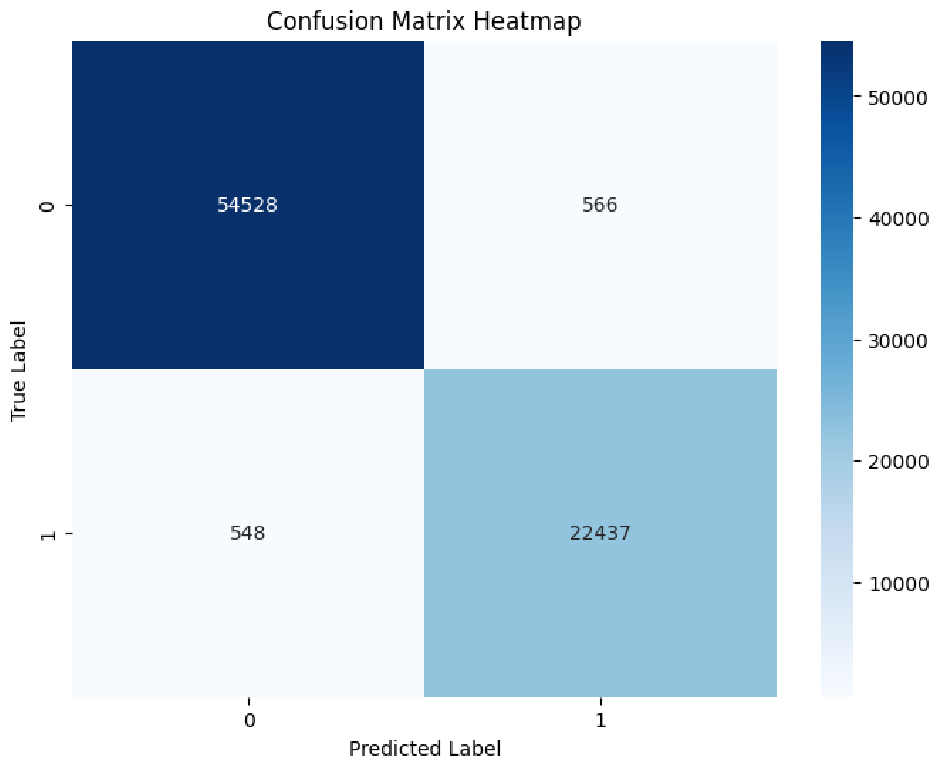
Justifications:

- 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	0.99	0.99	0.99	55094
1	0.98	0.98	0.98	22985
accuracy			0.99	78079
macro avg	0.98	0.98	0.98	78079
weighted avg	0.99	0.99	0.99	78079

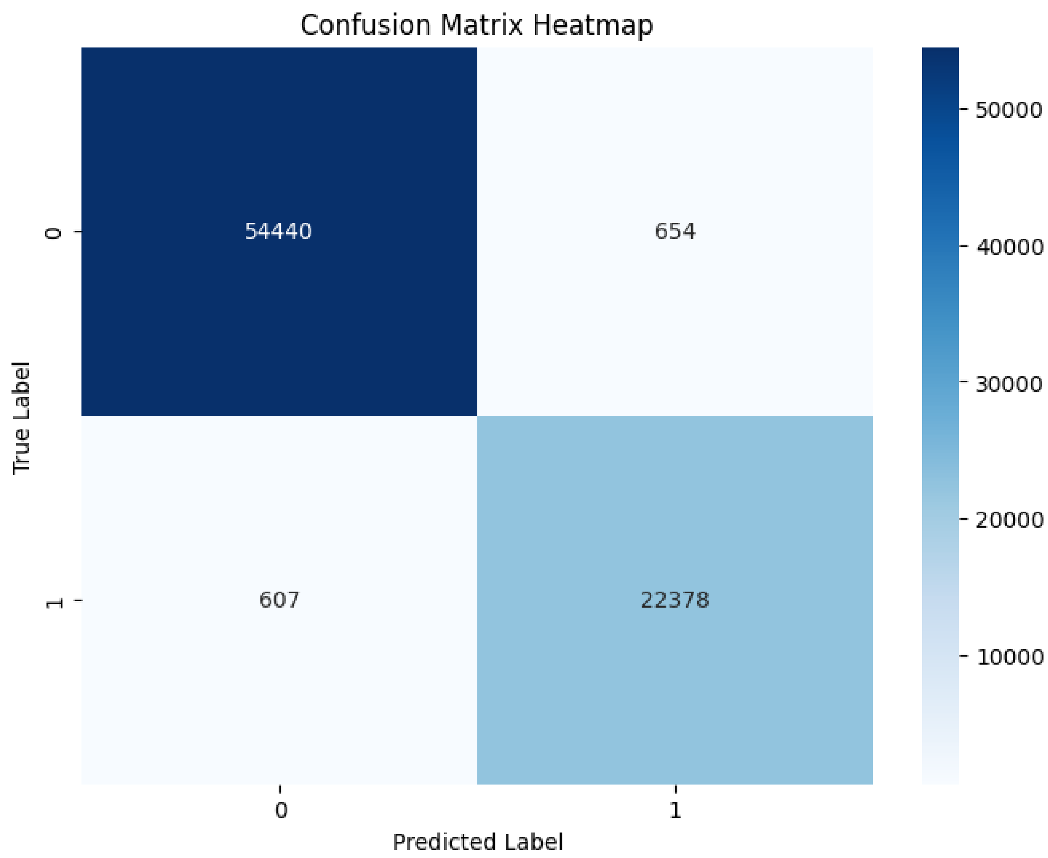


Random Forest Classifier

```
              precision    recall  f1-score   support

           0       0.99      0.99      0.99     55094
           1       0.97      0.97      0.97     22985

    accuracy                           0.98     78079
   macro avg       0.98      0.98      0.98     78079
weighted avg       0.98      0.98      0.98     78079
```



5.3. Model Comparison

Model	Accuracy	Notes
Logistic Regression	99%	Simpler, generalizes better, faster
Random Forest	98%	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:
 MSAIF       0.087526
DELIRIF     0.087525
HIVIF       0.087521
SCHIZOIF    0.087511
IMPSUBIF    0.087505
FTLDMOIF    0.087503
EPILEPIF    0.087493
PTSDDXIF    0.087485
BIPOLDIF    0.087466
ESSTREIF    0.087427
Name: PC1, dtype: float64
Explained Variance Ratio: [0.36061279 0.09582376 0.07389148 0.03709753 0.03381452 0.02889867
 0.0208967  0.01846978 0.01612167 0.01561101 0.01391761 0.0129972
 0.01099671 0.00979027 0.0079736  0.00784659 0.0069379  0.00682237
 0.00623964 0.0061662  0.00556157 0.00537907 0.00527868 0.00488248
 0.00475794 0.00449473 0.00436849 0.00417137 0.00408829 0.0039596 ]
```

7. GitHub Repo Link

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