THYROID DISEASE DETECTION AND PREDICTION

IS597 Final Project Presentation

Group 7: Danni Wu, Pranav Rajesh Charakondala, Arundhati Raj

Project Overview

- Objective: Predict thyroid disease using machine learning.
- Type: Binary Classification (Thyroid Disease vs No Thyroid Disease)
- Tools: AWS SageMaker, Amazon S3, Jupyter Notebooks
- Key Focus: Early diagnosis, reproducibility, model scalability

Dataset Summary

Total Records: 9,000+ patients

Features: Hormone levels (TSH, T3, TT4, T4U, FTI), demographic info

Target Variable: Thyroid condition (yes/no)

Challenges: Missing values, slight class imbalance

Methodology

Models Implemented:

- Decision Tree Classifier
- Random Forest Classifier
- Logistic Regression (Baseline Model)

Data Handling:

- Missing Value Imputation (Median/Mode)
- Stratified Train/Test Split (80/20)

Feature Engineering:

- One-Hot Encoding for categorical variables
- Principal Component Analysis (PCA) for dimensionality reduction

SageMaker Workflow

Data upload and storage on Amazon S3

Notebook-based Data Preprocessing

Model Training using SageMaker built-in algorithms

Evaluation Metrics Calculation

Model Export: Models saved as Joblib files to S3

MODEL TRAINING

80/20 stratified split

```
Training set shape: (7336, 26)
Testing set shape: (1835, 26)
Class distribution in y_train:
0    0.738
1    0.262
```

Data Loading and Preprocessing

```
One-hot encoding complete.
X_train shape: (7336, 26)
X_test shape: (1835, 26)
```

```python
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, accuracy_score

# Define baseline models
models = {
    "Decision Tree": DecisionTreeClassifier(random_state=42),
    "Random Forest": RandomForestClassifier(random_state=42),
    "Logistic Regression": LogisticRegression(max_iter=2000, solver='lbfgs', random_state=42)
```

Model Results

```
===== Decision Tree =====
 Accuracy: 0.937
              precision    recall  f1-score   support

           0       0.96      0.96      0.96      1355
           1       0.88      0.88      0.88       480

    accuracy                           0.94      1835
   macro avg       0.92      0.92      0.92      1835
weighted avg       0.94      0.94      0.94      1835

----------------------------------------
===== Random Forest =====
 Accuracy: 0.942
              precision    recall  f1-score   support

           0       0.97      0.95      0.96      1355
           1       0.86      0.93      0.89       480

    accuracy                           0.94      1835
   macro avg       0.92      0.94      0.93      1835
weighted avg       0.94      0.94      0.94      1835
```

```
===== Logistic Regression =====
 Accuracy: 0.837
              precision    recall  f1-score   support

           0       0.83      0.98      0.90      1355
           1       0.87      0.44      0.59       480

    accuracy                           0.84      1835
   macro avg       0.85      0.71      0.74      1835
weighted avg       0.84      0.84      0.82      1835
```

Model Performance Evaluation

Model	Accuracy	Precision (Class 1)	Recall (Class 1)	F1-Score (Class 1)	AUC-ROC
Decision Tree	93.7%	0.88	0.88	0.88	0.933
Random Forest	94.2%	0.86	0.93	0.89	0.947
Logistic Regression	83.7%	0.87	0.44	0.59	0.850

Random Forest Classifier

- High recall is critical to minimize false negatives.
- Strongly recommended for healthcare predictions needing reliability.

Decision Tree Classifier

- Slightly weaker generalization compared to Random Forest.
- Prone to overfitting on noisy or redundant data.

Logistic Regression

- Underperformed
- Low recall — missed many true thyroid disease cases.

ROC CURVE

Decision Tree

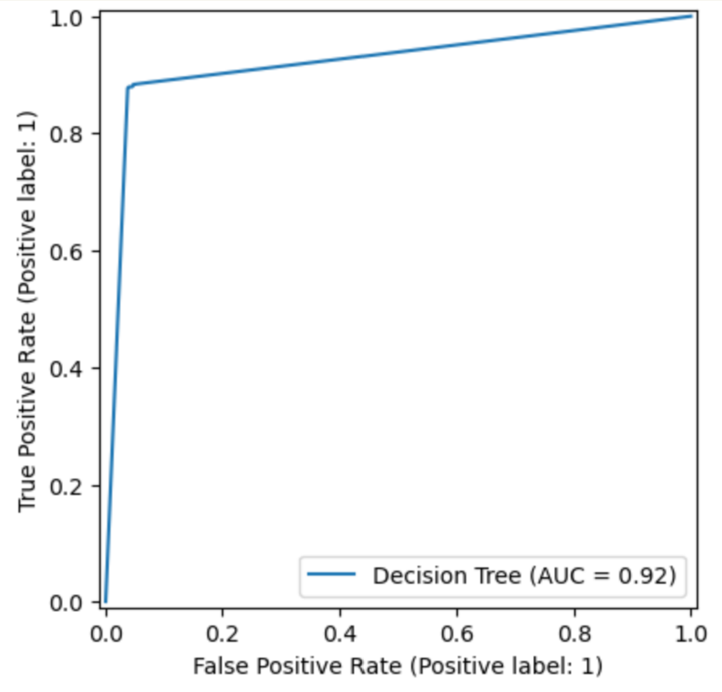
- AUC of 0.92
- balancing accuracy and model interpretability.

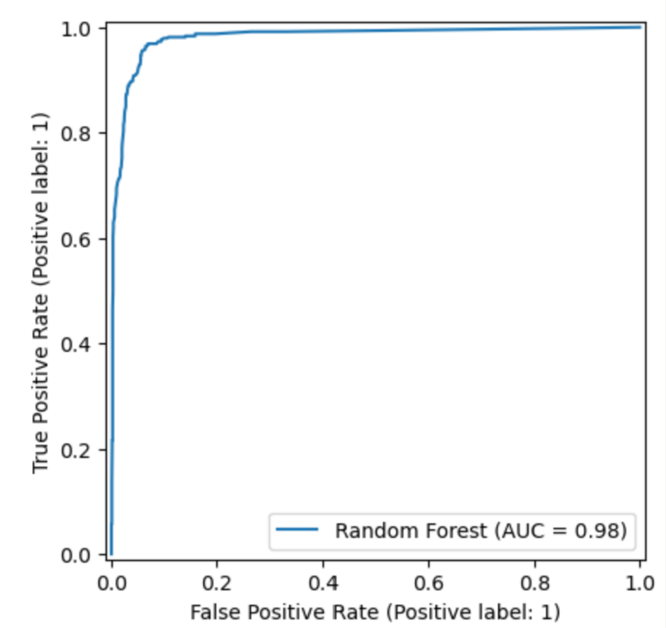
Random Forest :

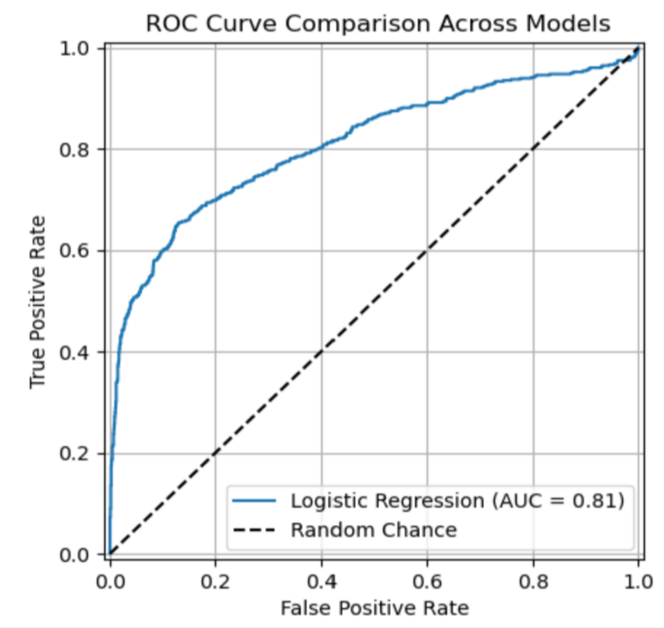
- best performance
- AUC of 0.98
- high sensitivity
- low false positive rates

Logistic Regression

- AUC of 0.81
- struggling to detect positive thyroid cases







Confusion Matrix

Decision Tree Classifier

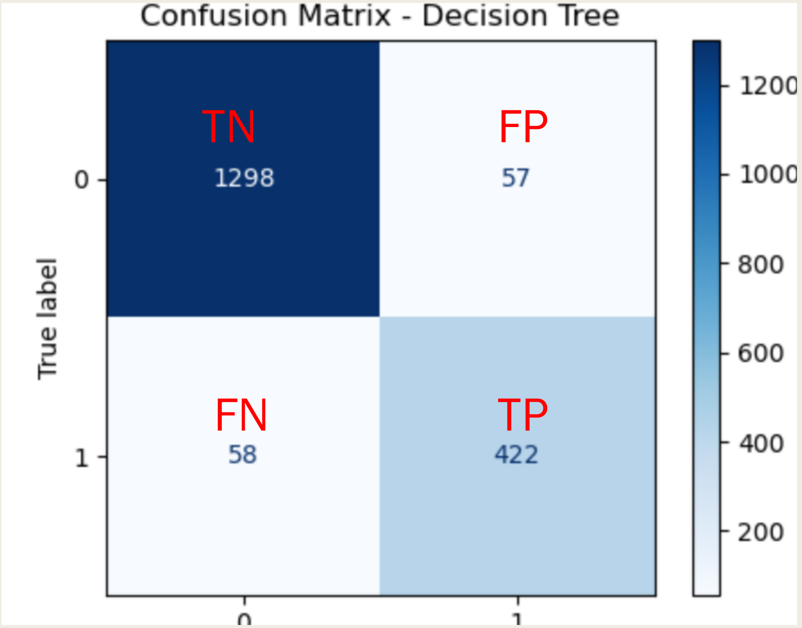
- Balanced performance
- moderate FN :58.

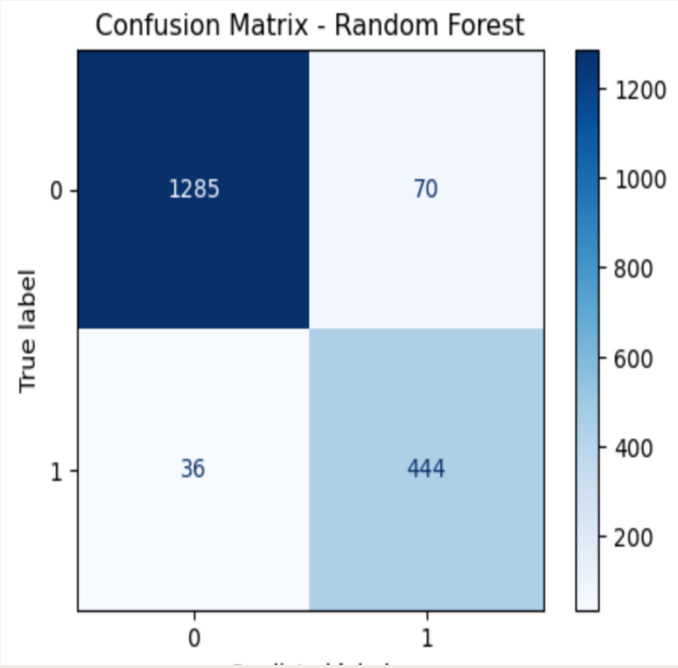
Random Forest Classifier

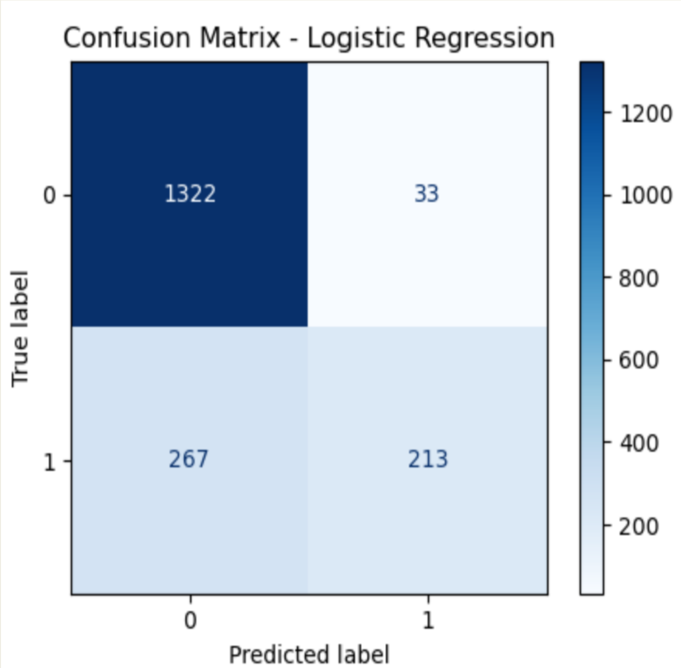
- Best recall , Low FN: 36 .
- Stronger at detecting thyroid disease
- Trade-off in FP.

Logistic Regression

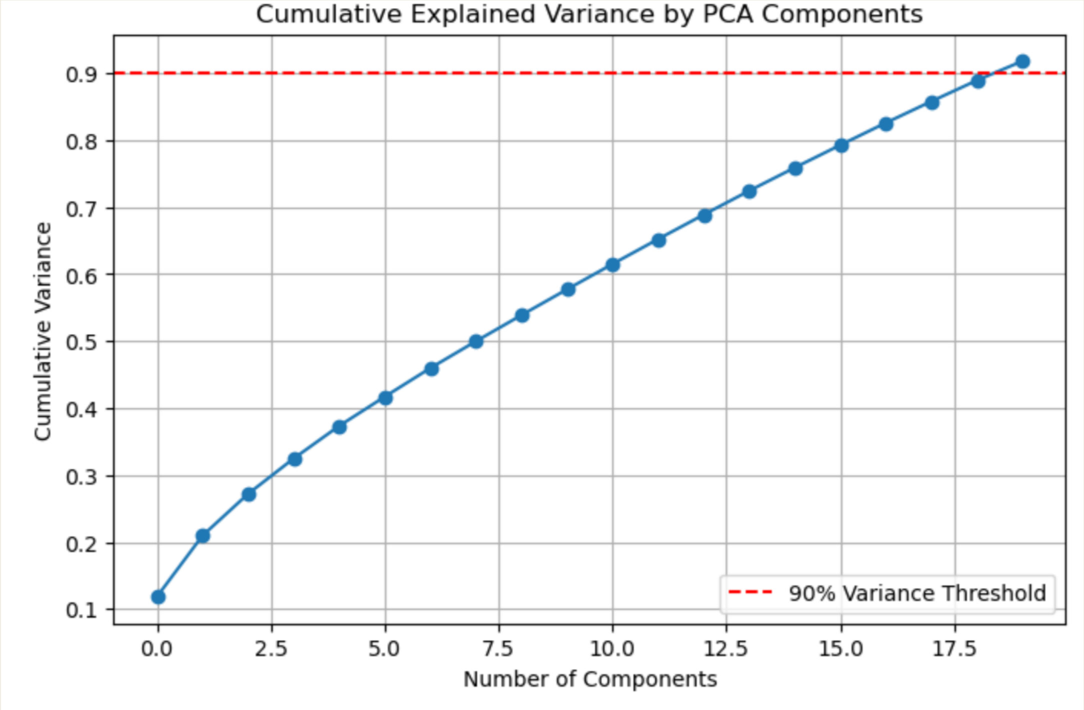
- High TN, high FN: 267
- Poor recall: misses many thyroid cases.





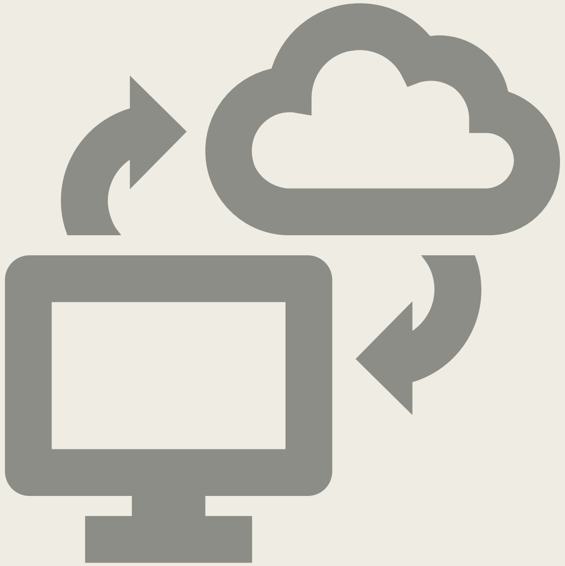


```
==== Random Forest on PCA-Reduced Features (20 Components) ====
              precision    recall  f1-score   support

           0       0.88      0.93      0.90      1355
           1       0.75      0.63      0.68       480

    accuracy                           0.85      1835
   macro avg       0.81      0.78      0.79      1835
weighted avg       0.84      0.85      0.84      1835
```



Impact of PCA

- **Accuracy**: 85% on the test set (compared to ~94% without PCA).
- **Recall (for class 1 - thyroid cases)**: Dropped to 0.63, indicating reduced sensitivity.
- **F1-score (for class 1)**: Reduced to 0.68.



Strengths & Challenges

- **Strengths:**
- Cloud-native and scalable pipeline
- Effective preprocessing for noisy healthcare data
- Strong model generalization
- **Challenges:**
- PCA improved efficiency but slightly reduced thyroid detection sensitivity.
- Dataset demographic bias (Mostly adult patients)
- Minor class imbalance affected logistic regression performance

Future Enhancements

- Automate full SageMaker pipeline using AWS Step Functions
- Integrate XGBoost and LightGBM for potentially higher accuracy
- Deploy REST API for real-time thyroid prediction
- Utilize SHAP or LIME for model explainability in healthcare compliance

Conclusion

ACHIEVED 94.2% ACCURACY USING RANDOM FOREST

DEMONSTRATED FEASIBILITY OF ML-ASSISTED THYROID DIAGNOSIS

READY FOR SCALING WITH MORE DIVERSE DATASETS AND REAL-WORLD INTEGRATION

Team Contributions

Pranav Rajesh Charakondala:

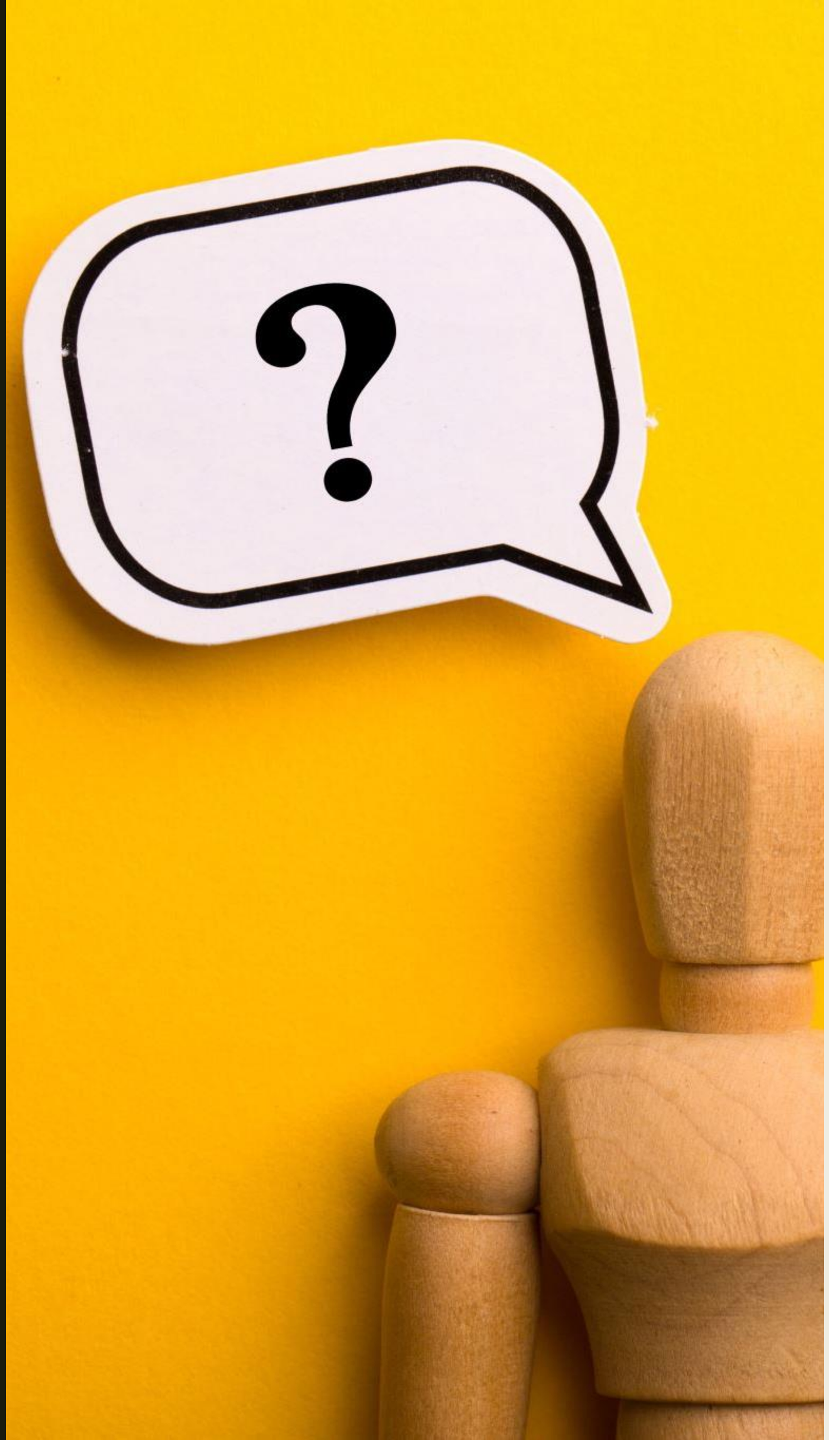
Model Training using SageMaker built-in algorithms

Evaluation Metrics Calculation

Danni Wu and Arundhati Raj:

Choosing final dataset and uploading it on S3

Notebook-based Data Preprocessing



Thank You!