Machine Learning Progress Report

Paula Lozano Gonzalo

April 14, 2025

1 Model Overview

This report documents my progress in developing a classifier for diabetes prediction. After identifying significant class imbalance issues in my previous work, I have now implemented a balanced dataset approach and evaluated three model types:

- SGDClassifier (baseline)
- Random Forest classifier (current best performer)
- RidgeClassifierCV (new linear model for comparison)

The task remains binary classification - predicting diabetes outcomes based on patient characteristics.

2 Dataset & Preprocessing

Key updates to the dataset and preprocessing:

- **Class Balancing**: I made a new file to make sure I cleaned the data for equality in between negative and positive cases. And this is how the data is looking now.

  ```
  ./clean_data.py
  Reading data from data/diabetes_dataset.csv...
  Original dataset:
  Positive cases: 8500
  Negative cases: 91500

  Balanced dataset:
  Total cases: 17000
  Positive cases: 8500
  Negative cases: 8500
  ```

  As we can see, I now have $17,000$ cases versus the original $100,000$. However, I don't think that this is a problem since we still have plenty of data for the model to learn trends.

- **Data Splitting**: I now have to split the data again and here is how it's looking.

```
Original data shape: (17000, 17)
Test data shape: (3400, 17)
Training data shape: (10880, 17)
Validation data shape: (2720, 17)
```

- Maintained previous preprocessing:
  - Missing value imputation (most frequent for categorical, median for numerical)
  - Binary encoding for categorical features
  - Polynomial features (degree=2) where applicable

3 Training Progress

After balancing the dataset, I wanted to re-train the two past models to see if there would be a difference and all models showed improved ability to identify positive cases, though with slightly reduced overall accuracy:

Model	Training Acc.	Validation Acc.	Validation F1
SGDClassifier	0.881	0.890	0.827
Random Forest	0.899	0.898	0.907
RidgeClassifierCV	0.881	0.887	0.879

Table 1: Model performance on balanced dataset

Confusion Matrices (Validation Data):

```
==== SGDClassifier ====    ==== Random Forest ====   ==== RidgeClassifierCV ====
t/p  F     T               t/p   F     T               t/p   F     T
F 1140.0 242.0              F 1258.0 124.0                F 1231.0 151.0
T 223.0 1115.0              T 126.0 1212.0                T 171.0 1167.0

Precision: 0.822           Precision: 0.907           Precision: 0.885
Recall:    0.833           Recall:    0.906           Recall:    0.872
F1:        0.827           F1:        0.907           F1:        0.879
```

The best model so far is the Random Forest so I will be talking about this one mostly to make the report as short as I can but also showing all the work I'm doing.

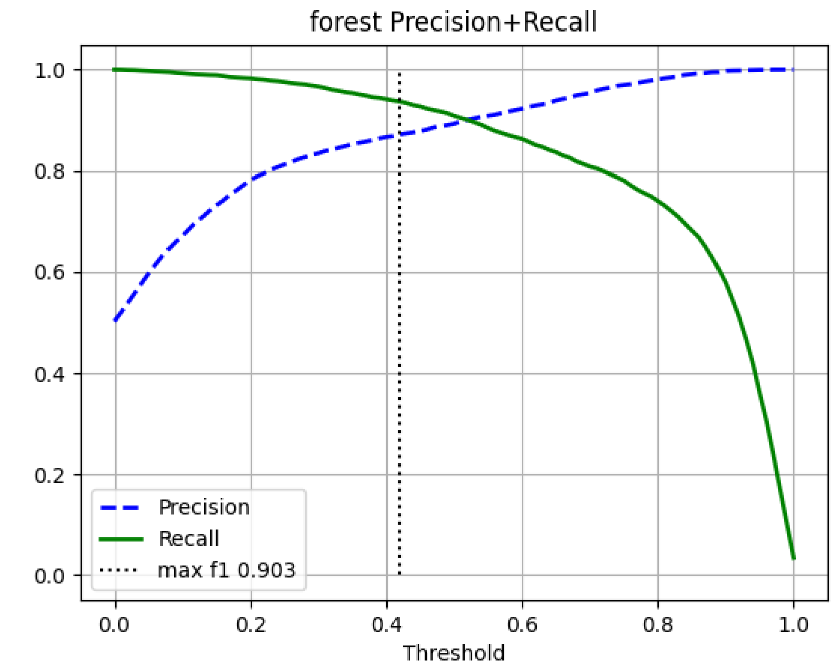

Figure 1: Precision-Recall Plot for Random Forest (Balanced Data)

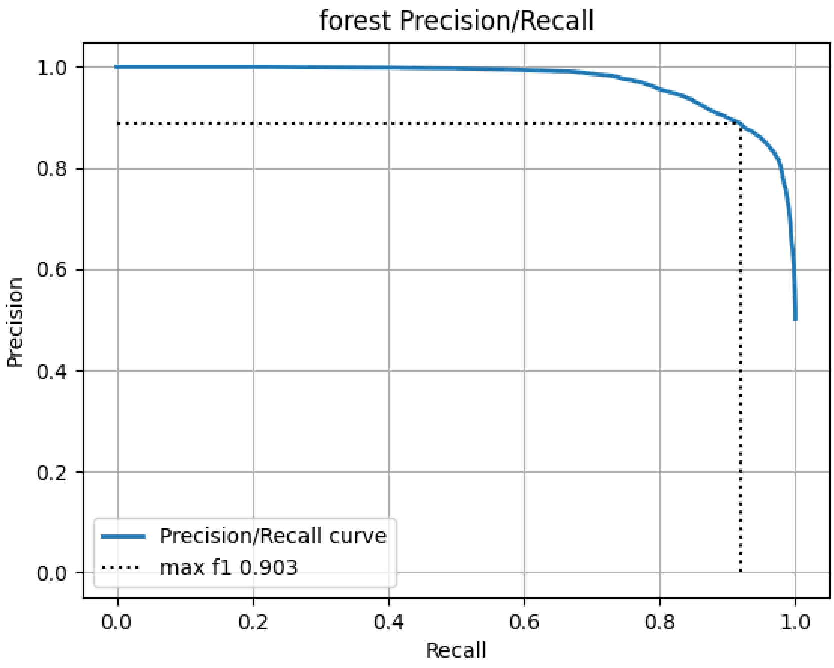

Figure 2: Precision-Recall Curve for Random Forest (Balanced Data)

Key observations:

- Random Forest emerges as the best performer (F1=0.907) on balanced data
- All models show much better balance between precision and recall compared to imbalanced data (As seen in the confusion matrices)

- The number of false negatives decreased significantly:
  - Random Forest: 126 (vs 240 with imbalanced data)
  - SGDClassifier: 223 (vs 1990 with imbalanced data)
- Training and validation scores are now much closer, suggesting reduced overfitting

By all these improvements I can tell that the decision of cleaning the data was the correct one and I will stick to it.

4 Challenges & Solutions

Challenge 1: Reduced Dataset Size

- Solution: Accepted smaller dataset (17,000 vs original 100,000) to achieve class balance
- Impact: Models perform well despite smaller size, suggesting quality over quantity

Challenge 2: Model Selection

- Solution: Compared three model types
- Outcome: Random Forest performs best, but linear models remain competitive

Challenge 3: Metric Interpretation

- Solution: Focused on F1 score rather than accuracy as primary metric
- Benefit: Better captures model performance on both classes

5 Next Steps

- Keep messing around with the hyper-parameters to see if I can improve random forest.
- Train neural networks as well and compare with the past models.