5/12/2025

Predicting Diabetes with Machine Learning

This project uses clinical data from a Frankfurt hospital to compare machine learning models for diabetes prediction. By cleaning and preparing data thoroughly, then testing classifiers from LDA to XGBoost, we evaluate performance and identify key health indicators associated with diabetes.

Benny Frisella
UNIVERSITY OF TEXAS AT DALLAS

Introduction

According to the World Health Organization (WHO), "***diabetes is a major contributor to severe health complications, including heart disease, stroke, blindness, kidney failure, and lower limb amputations," (World)***. In the **United States alone**, the American Diabetes Association reports that, "*as of 2021, **38.4 million Americans—approximately 11.6% of the population—were living with diabetes***, with 8.7 million cases undiagnosed," (*Statistics*).

To explore the characteristics and predictors of diabetes, this analysis uses a **dataset of 2,000 clinical records collected from a hospital in Frankfurt, Germany, last updated seven years ago**. Despite a couple of limitations such as missing values and potential outliers, the dataset offers meaningful insight into the relationships between patient health indicators and diabetes outcomes.

This study explores clinical predictors of diabetes using machine learning models. The process emphasizes the **importance of careful data preparation** and **comparative evaluation of multiple classifiers**, ranging from linear discriminant analysis to boosted ensemble methods.

Preliminary analysis indicated that Glucose, Insulin, and BMI were the strongest predictors of diabetes. Models that captured non-linear interactions, particularly Random Forest and XGBoost, performed best.

Dataset Overview

Missing values

- **Insulin**: 956
- **Skin Thickness**: 573
- **Blood Pressure**: 90
- **BMI**: 28
- **Glucose**: 13

Outliers

- **Pregnancies**: 3 records had 17 pregnancies. This is the maximum value of this attribute.

Figure 1

Luckily, there are **no missing values in Diabetes Pedigree Function nor** in **Age**, while all missing values were filled with zeroes. As seen above, around half of the Insulin fields are missing and over one-quarter of the skin thickness category as well.

PROBLEM: *Analyze the patterns in the dataset to accurately predict a diabetic **Outcome**.*

Initial Models – Linear/Quadratic Discriminant Analysis

Initial classification using **Linear Discriminant Analysis (LDA)** yielded a model accuracy of 78%, with a misclassification rate of 22%. The model demonstrated stronger performance in identifying non-diabetic patients (specificity = 79.73%) than diabetic ones (sensitivity = 73%). The **ROC curve** for LDA showed an Area Under the Curve (AUC) of **0.84**, indicating solid discriminative ability. Comparatively, **Quadratic Discriminant Analysis (QDA)** produced slightly lower performance: 76% accuracy, 24% misclassification, specificity of 79.47%, and sensitivity of 68.52%. Its ROC curve also had an AUC of **0.84**, suggesting similar predictive capacity, but with slightly lower true positive performance.

Although both models performed similarly, LDA slightly outperformed QDA in both overall accuracy and sensitivity. As a result, LDA was favored when using the default posterior probability cutoff of 50%. However, when experimenting with lower thresholds to improve recall, LDA at a **33% cutoff** offered the highest accuracy among its variants, while QDA at a **25% cutoff** provided the best specificity within its group. These adjustments highlight how changing classification thresholds can fine-tune the balance between precision and recall, even if the models themselves don't change structurally.

Initial results with LDA/QDA showed moderate accuracy but underscored a deeper issue — the dataset was not yet ready for robust modeling.

After I prematurely ran LDA and QDA, I returned to square one to spend more time in the Exploratory Data Analysis (EDA) phase. Upon further investigation, the proportions of the Outcome classes were not severely imbalanced, the **non-diabetics represented ~66%** and **diabetics ~34%** of the population. Fortunately, **LDA and QDA were warranted since that is balanced enough for them.**

Regarding missing values, the **Insulin variable was *excluded* from further analysis** due to **excessive missing values (~50%),** which could compromise model reliability. Furthermore, **Glucose** and **BMI** had relatively few missing entries (13 and 28, respectively) and were imputed using mean substitution, given the minimal risk of distortion.

The remaining numeric features (excluding the response variable, Outcome) were then **standardized** using z-scores, which is appropriate given the dataset's size (n > 2000) and justified under the Central Limit Theorem. **Blood Pressure**, with **90 missing values**, was imputed next using k-Nearest Neighbors (kNN) from the VIM library, followed by **Skin Thickness**, which had **573 missing entries**. This sequential imputation—starting with variables with the least number of missing values—ensures that most predictors are complete when imputing variables with more missing data. This approach improves the reliability of kNN, which relies on complete, standardized predictors for distance-based estimation.

In a check for multicollinearity between predictor variables, ***no correlation exceeded 0.8***, and with only 7 variables there was no reason to run Principal Component Analysis or any dimensional reduction. *Figure 2* below shows a quick summary of the most correlated predictors.

	Var1 <fctr>	Var2 <fctr>	value <dbl>
7	Age..	Pregnancies..	0.53945719
26	BMI..	SkinThickness..	0.38309800
14	Age..	Glucose..	0.25980465
21	Age..	BloodPressure..	0.23837508
12	BMI..	Glucose..	0.23265420
18	SkinThickness..	BloodPressure..	0.19880047
27	DiabetesPedigreeFunction..	SkinThickness..	0.17829888
19	BMI..	BloodPressure..	0.16841352
3	BloodPressure..	Pregnancies..	0.14967246
10	BloodPressure..	Glucose..	0.14426855

Figure 2

Following data preparation and imputation, LDA and QDA were run again on the freshly imputed dataset. The results can be found below in *Figure 3*.

Model <chr>	Accuracy <dbl>	Sensitivity <dbl>	Specificity <dbl>
LDA	0.788	0.555	0.905
QDA	0.767	0.570	0.865

Figure 3

Running it back again, LDA was chosen for its balance and higher overall accuracy, while QDA could have been considered if the priority was identifying diabetic patients more aggressively, albeit with slightly reduced precision.

Unfortunately, despite the extensive data imputation process, the **overall performance of the models remained largely unchanged besides a 10% increase in Specificity for LDA over the original dataset**. This outcome is not uncommon and can be attributed to several factors. First, the imputation methods used were appropriate and aligned with best practices: mean imputation for variables with few missing values and kNN for those with substantial non-applicable values. Second, the most predictive features in the dataset, such as Glucose, Age, and BMI, had relatively few missing values and were likely already providing most of the prediction. Third, the models used—**LDA, and QDA—are relatively robust to modest levels of missing values, especially when those missing values occur in less influential predictors.** Interestingly, removing the Insulin variable, which had a high proportion of missing data, may have improved model stability by eliminating a noisy and unreliable feature. While the imputations did not lead to a significant boost in predictive accuracy, they were crucial for ensuring the validity and generalizability of the models.

To exit the diminishing returns loop of the LDA/QDA analyses, the next algorithm used was a **Logistic Regression Model (LRM)** which was trained and evaluated using a 70/30 train-test split. The resulting Logistic Regression Model identified **Glucose, BMI, and Diabetes Pedigree Function as the strongest positive predictors of diabetes**, with **all variables** showing statistical significance **except Skin Thickness,** according to *Figure 4* below.

```
Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)               -0.82551    0.07011 -11.775  < 2e-16 ***
Pregnancies..              0.34144    0.07811   4.371 1.24e-05 ***
Glucose..                  1.01952    0.07697  13.246  < 2e-16 ***
BloodPressure..           -0.16180    0.07319  -2.211   0.0271 *
SkinThickness..           -0.12879    0.07286  -1.768   0.0771 .
BMI..                      0.67125    0.08052   8.337  < 2e-16 ***
DiabetesPedigreeFunction.. 0.30570    0.07135   4.285 1.83e-05 ***
Age..                      0.17026    0.08011   2.125   0.0335 *
```

Figure 4: Summary of Logistic Model:

Logit(Outcome) = -0.8255 + 0.3414(Pregnancies) + 1.0195(Glucose) – 0.1618(BloodPressure) – 0.1288(SkinThickness) + 0.6713(BMI) + 0.3057(DiabetesPedigreeFunction) + 0.1703(Age)

```
Error rate: (FP + FN) / (TP + TN + FP + FN) = 0.2216667
Sensitivity: TP / (TP + FN) = 0.6093023
Specificity: TN / (TN + FP) = 0.8727273
```

Figure 5

As shown above in Figure 5, the LRM model achieved an **overall error rate of ~22% (accuracy of ~78%)**, with a **sensitivity of 60.9%** and a **specificity of 87.28% as seen below in** *Figure 5*.

While the specificity indicated decently strong performance in correctly identifying non-diabetic patients, the **sensitivity revealed a notable weakness** in correctly identifying diabetic cases — with nearly **half of the positive cases misclassified**. This imbalance would undoubtedly be a concern in medical applications, where **false negatives can have serious implications** for patient outcomes.

Regularization was added to **shrink noisy, non-semantic predictors** like skin thickness, to **automatically perform feature selection** and improve generalization. Using cv.glmnet(), we found the optimal coefficients as seen in cross-validation plot below.

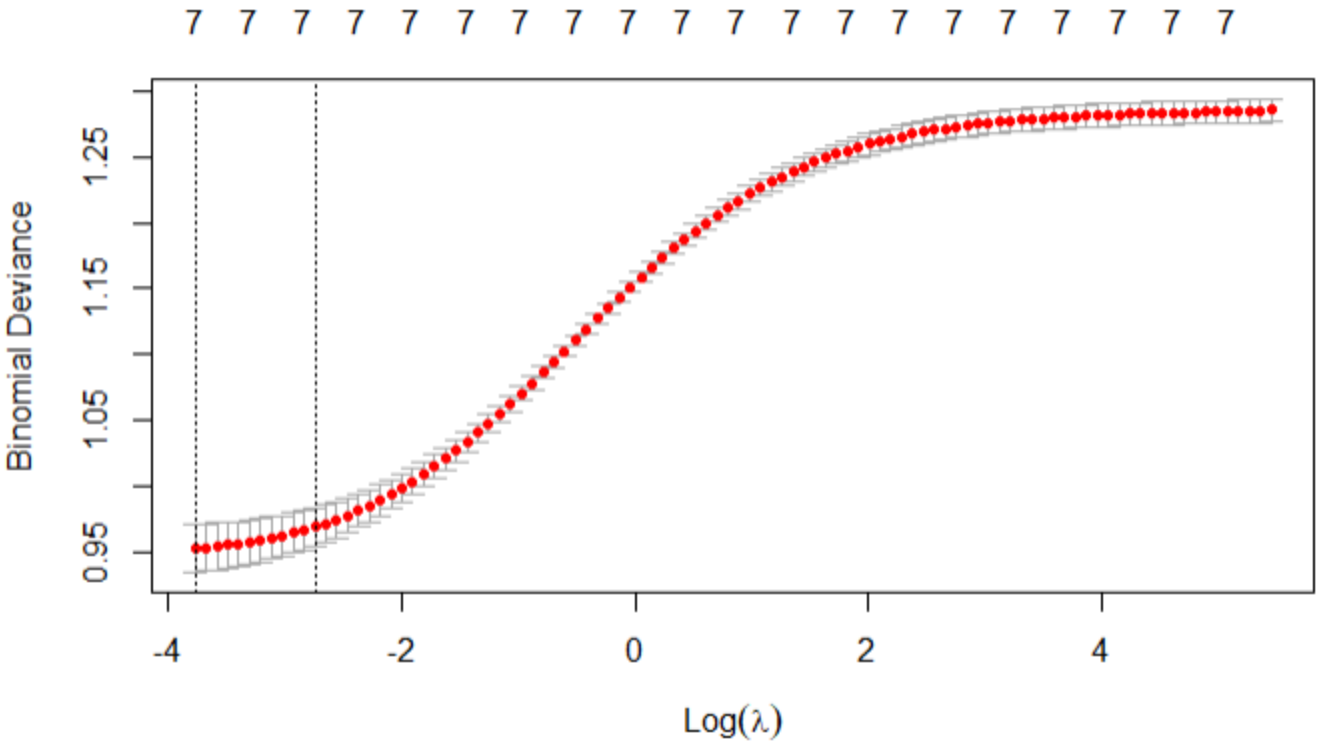

Figure 6: Ridge-Regularization. Here the x-axis is the penalty term, and the y-axis is a proxy for model error.

Experiencing marginal gains in Specificity, Accuracy and Sensitivity of around 1%, it was time to try Lasso Regularization. *Figure 7* below shows the cross-validation plot of Lasso Regularization.

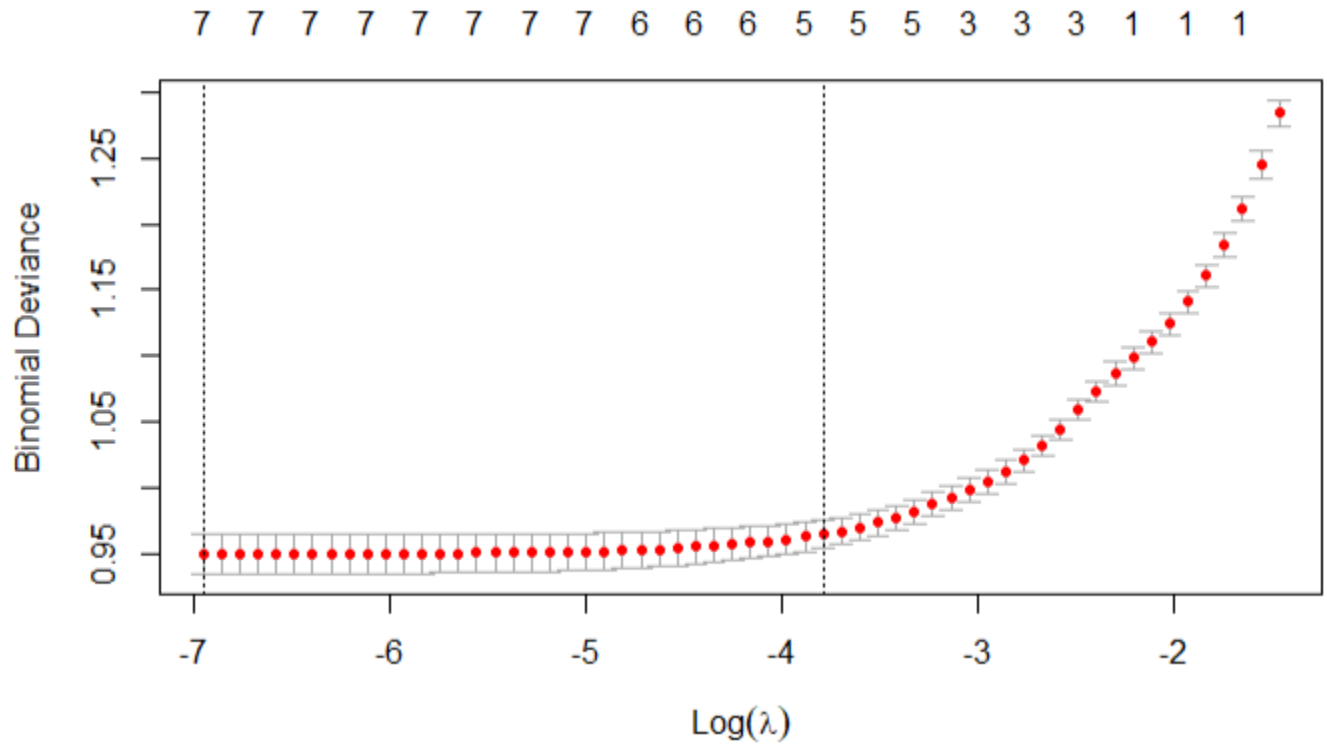

Figure 7: Lasso Regularization Analysis

Unlike Ridge regression, which shrinks all coefficients gradually, Lasso shrinks some coefficients to exactly zero, **effectively performing variable selection**. Simply put, **Lasso penalizes complexity more aggressively** which is why the deviance curve drops off more steeply and flattens earlier.

Overall, Lasso had achieved the highest accuracy among all models thus far (only slightly edging out Ridge). More importantly, it did so while performing automatic feature selection, likely improving model simplicity and interpretability. The results from Lasso's implicit feature selection can be seen below in *Figure 8*.

	Predictor <chr>	Coefficient <dbl>
1	(Intercept)	-0.87522793
2	Pregnancies..	0.41760961
3	Glucose..	1.08004720
4	BloodPressure..	-0.14063404
5	SkinThickness..	-0.05156698
6	BMI..	0.56244126
7	DiabetesPedigreeFunction..	0.26086377
8	Age..	0.14519151

Figure 8

Importantly, glmnet ignores the Intercept, so no penalty was applied as it was estimated freely. Lasso **did not eliminate any predictors**, suggesting that each variable contributes semantically to the model. However, the penalty term still shrank the coefficients from the LRM, reducing model complexity and helping to prevent an overfit (not here though). Interestingly, skin thickness shrunk the most, down less than 50% from the LRM.

To compare all models, we compared their respective ROC curves in *Figure 14* below.

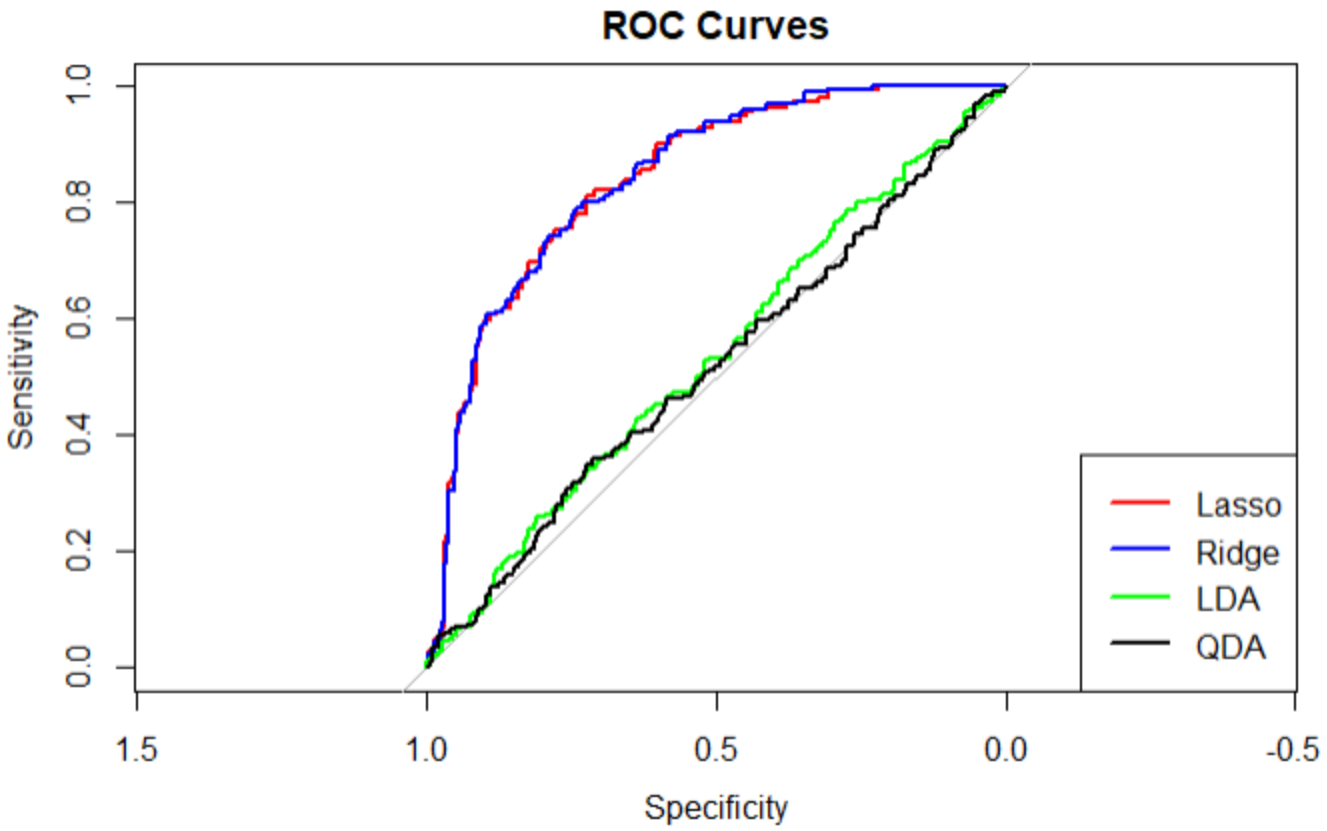

Figure 9: Receiving Operator Characteristics (ROC). A top-most left-most performance indicates better discriminatory power as it represents 1 – (False Positive Rate).

- **Ridge** was almost identical to **Lasso**, suggesting multicollinearity was not a major problem.
- **Lasso** showed slightly stronger sensitivity in the mid-range and high-specificity regions — likely due to its ability to regularize and shrink irrelevant coefficients.
- **LDA and QDA** visibly underperformed across all thresholds — **reinforcing that discriminative models are better suited here.**

A bit down after minimal improvements, we returned for one last walk through the EDA phase. We end up with a matrix of feature pairs, shown below in *Figure 10*.

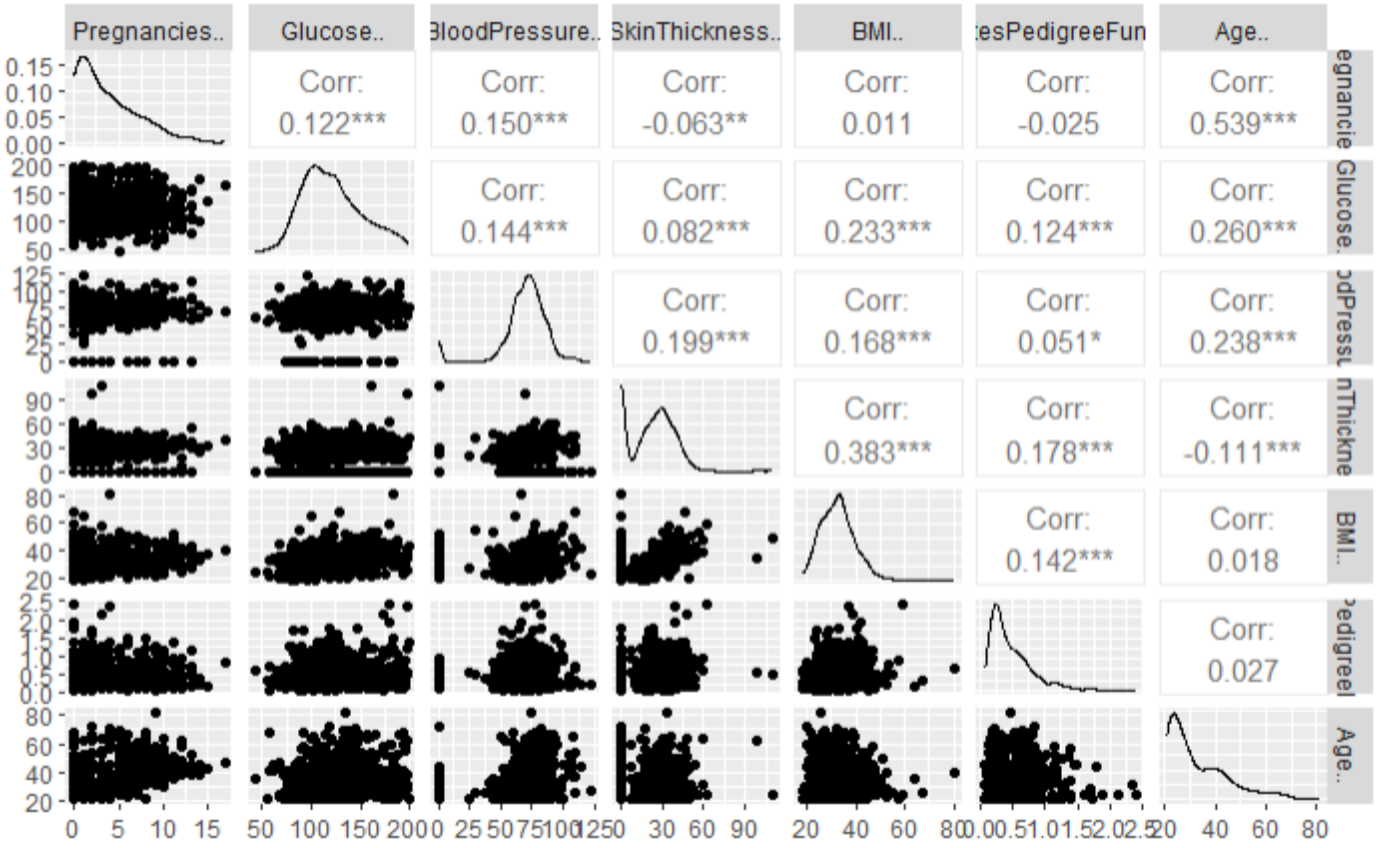

Figure 10: Features along the diagonal from top-left to bottom-right are densities showing how the feature behaves across all observations. Below those are scatter plots to analyze heteroskedasticity, when one feature depends on another.

From the marginal density plots along the diagonal, we observe that **Pregnancies**, **BMI**, **Diabetes Pedigree Function**, and **Age** exhibit negative skewness, while **Glucose**, **Blood Pressure**, and **Skin Thickness** appear approximately normally distributed. The scatterplots and correlation coefficients reveal **no evidence of multicollinearity** among predictors. Since Random Forests are inherently robust to multicollinearity, applying them in this context may not yield substantial advantages for that reason alone. Instead, we pivot our focus to the **Insulin** variable, which—despite having excessively abundant amount missing values—is widely recognized as a clinically relevant predictor of diabetes. To preserve the integrity of this variable, we will reintroduce it to the dataset and proceed by **removing all 956 rows where Insulin equals zero**, thereby retaining only the complete cases for analysis. The following, *Figure 11,* shows the new data subset of feature correlations.

	Var1 <fctr>	Var2 <fctr>	Freq <dbl>
8	Age..	Pregnancies..	0.662
30	BMI..	SkinThickness..	0.577
13	Insulin..	Glucose..	0.560
16	Age..	Glucose..	0.307
24	Age..	BloodPressure..	0.303
22	BMI..	BloodPressure..	0.255
38	BMI..	Insulin..	0.252
14	BMI..	Glucose..	0.237
12	SkinThickness..	Glucose..	0.215
29	Insulin..	SkinThickness..	0.200

Figure 11: Feature Correlations among the data subset with Insulin, now including 1,045 over the previous 2,001 observations.

The **correlations** in this data subset are **notably higher** than before, making it a **suitable candidate** for tree-based models like **Random Forest and XGBoost**. These models are more robust to multicollinearity than Logistic Regression—where coefficient estimates can become unstable and hard to interpret—and LDA/QDA, which rely on estimated covariance matrices that may be distorted by highly correlated predictors. *Figure 12* below demonstrates the results of Random Forest on the subset-ed data.

```
         Actual
Predicted   0   1
        0 214   5
        1  11  84
Random Forest Accuracy: 0.9490446
                                0        1 MeanDecreaseAccuracy MeanDecreaseGini
Pregnancies..               28.52215 35.79850             37.38383         25.25456
Glucose..                   56.89275 61.00928             68.86853         85.06965
BloodPressure..             27.22220 35.26050             37.87453         23.16380
SkinThickness..             29.49233 35.87794             39.53703         27.43589
Insulin..                   38.62958 41.03994             47.52260         50.98446
BMI..                       35.82315 40.03039             46.01047         38.16485
DiabetesPedigreeFunction.. 40.60155 38.34604             46.90057         33.01311
Age..                       39.66349 46.20575             50.19095         43.83755
```

Figure 12

```
Random Forest Statistics:

Sensitivity: 0.944
Specificity: 0.951
Error Rate: 0.051
Accuracy: 0.949
F1 Score: 0.913
```

Figure 13

Clearly, following Figure 13 above, this is the best model yet as the Random Forest model's accuracy was 94.9%.

Furthermore, the top predictors based on the mean decrease Gini Index tell us the top predictors shown below in *Figure 14*, which align with what most clinical research will say; particularly, shown by glucose and insulin predictors.

TOP PREDICTORS:

1. **Glucose** – 85.07
2. **Insulin** – 50.98
3. **Age** – 43.84
4. **Diabetes Pedigree Function** – 33.01

Figure 14

To elaborate, each time a feature is used to split a node in a decision tree, it contributes to a reduction in Gini Impurity. **A higher Mean Decrease Gini value** indicates that the feature consistently produces more **informative splits across the forest**, making it more important for accurate classification.

While Random Forest delivered strong performance with high accuracy and clear feature importance, it builds trees independently and may not always optimize for the most difficult-to-classify cases. To further refine predictive performance, we now turn to XGBoost, a gradient boosting method that builds trees sequentially, with **each tree correcting the errors of the previous one.** This boosting approach is often more effective in capturing subtle patterns in the data and can outperform Bagging (i.e. Random Forest) methods in many structured classification tasks.

The resulting **metrics** of the XGBoost can be found below in Figure 15.

```
XGBoost Statistics:

Sensitivity: 0.798
Specificity: 0.889
Error Rate: 0.137
Accuracy: 0.863
F1 Score: 0.768
```

Figure 15

XGBoost outperformed the earlier linear models by better capturing non-linear relationships and interactions, making it one of the **most effective classifiers tested** in this analysis with an **Accuracy of 86.3%**. However, we know ***this model can benefit immensely from hyperparameter tuning***, thus we will implement it to try and improve over the Bagging (Random Forest) model.

To tune XGBoost, I used the Caret library with a **grid that spans 9 combinations of 3 depths by 3 learning rates**. It used a group of 3 training rates (**hyperparameter/lambda at 0.01, 0.1, 0.3**) along with **five-fold cross-validation** to evaluate each combination where it trained on 4 partitions and tested on 1 for every 5 partitions. Upon completion, we were provided with the following in Figure 16,

```
Aggregating results
Selecting tuning parameters
Fitting nrounds = 100, max_depth = 9, eta = 0.3, gamma = 0, colsample_bytree = 1, min_child_weight = 1, subsample = 1
on full training set
```

Figure 16

Where Caret automatically chose the **best combination of parameters by minimizing the loss (or maximizing the resampled accuracy).** For example, the number of rounds was 100 which is moderate, the maximum depth was 9 so the grid was used to its **full extent which allows for complex interactions**. Finally, the main component here is highlighted in red above, the **eta = 0.3**. That **hyperparameter is a bit aggressive as it is a quick learning rate that does still pose a risk of overfit**. In the end, the result is shown below in *Figure 17*,

```
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No  206   3
       Yes   6  98

               Accuracy : 0.9712
                 95% CI : (0.9461, 0.9868)
    No Information Rate : 0.6773
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9347

 Mcnemar's Test P-Value : 0.505

            Sensitivity : 0.9717
            Specificity : 0.9703
         Pos Pred Value : 0.9856
         Neg Pred Value : 0.9423
             Prevalence : 0.6773
         Detection Rate : 0.6581
   Detection Prevalence : 0.6677
      Balanced Accuracy : 0.9710

       'Positive' Class : No
```

Figure 17

As seen above, the **freshly tuned XGBoost model demonstrated remarkedly strong performance**, achieving **an overall accuracy of 97.12%**, with balanced sensitivity (97.17%) and specificity (97.03%). This indicates that the model is **highly effective at correctly identifying both diabetic and non-diabetic** cases. The precision (98.56%) and negative predictive value (94.23%) further reinforce the model's reliability in its predictions.

A high Kappa value (0.9347) suggests strong agreement between the predicted and actual classes, and the **McNemar's test p-value of 0.505 shows no evidence of classification bias**. Compared to previous models, **this tuned XGBoost outperforms all others** and appears to **generalize well without overfitting, making it the strongest candidate for deployment in this classification task of identifying diabetics.**

Model Performance Summary

Model	Accuracy	Sensitivity	Specificity	F1 Score	Notes
LDA	78.0%	73.0%	79.7%	—	Balanced linear classifier
QDA	76.0%	68.5%	79.5%	—	Lower sensitivity than LDA
Logistic Reg.	78.0%	60.9%	87.3%	—	High specificity but low recall
Ridge	~79.0%	~62%	~88%	—	Small boost via regularization
Lasso	~80.0%	~64%	~88%	—	Best linear model; simpler, no loss in accuracy
Random Forest	94.9%	94.4%	95.1%	91.3%	Excellent performance; interpretable via Gini
XGBoost	86.3%	79.8%	88.9%	76.8%	Untuned; strong but outperformed by RF
XGBoost (Tuned)	97.1%	97.1%	97.0%	**95.7%**	Best overall; balanced, high generalization

Figure 18

After cleaning and wrangling the dataset through multiple iterations, seven supervised learning models were tested and compared: LDA, QDA, Logistic Regression, Ridge, Lasso, Random Forest, and XGBoost. **Early discriminant analysis models (LDA and QDA) showed moderate success**, with LDA slightly outperforming QDA in terms of sensitivity and accuracy. Logistic regression improved upon these results, and regularization (Ridge and Lasso) marginally enhanced the performance while simplifying the model. **However, the real leap came when tree-based models were introduced**. **Random Forest delivered high accuracy (94.9%) and a robust F1 Score (0.913),** outperforming all prior models with clear feature importance **led by Glucose and Insulin**. **XGBoost**, **after hyperparameter tuning**, achieved the highest accuracy (97.1%) and balanced sensitivity and specificity, making it the **best-performing model overall.**

Conclusion

This project demonstrates how **critical proper data cleaning and preparation** are to the **success of machine learning models**. Initially, models like LDA and QDA yielded average results, but rather than discard them, **revisiting the data revealed deeper insights**: missing values, outlier treatment, and appropriate imputation methods drastically influence model performance. Exploring the impact of feature standardization, correlations, and variable importance reinforced the importance of a methodical approach. Ultimately, ensemble methods like **Random Forest and XGBoost proved the most powerful** — capturing non-linear relationships and improving accuracy. The journey underscored a key lesson: effective modeling begins with data integrity. Taking time to understand, validate, and refine the dataset often yields greater dividends than endlessly tweaking models. Through this experience, I learned that retracing steps is not regression but refinement, and that the **best models come from respecting the data first. Thorough exploratory analysis isn't optional; it's where meaningful models truly begin.**

Future Work

Limitations of this analysis include the moderate sample size, the sheer magnitude of missing values, and a lack of deeper domain-specific features. In future work, additional clinical variables, temporal data, or ensemble stacking methods could be explored to further improve predictive performance.

Works Cited

World Health Organization. (2024). *Diabetes*. World Health Organization. https://www.who.int/news-room/fact-sheets/detail/diabetes#:~:text=Diabetes%20causes%20blindness%2C%20kidney%20failure,caused%20by%20high%20blood%20glucose.

***Statistics about diabetes*. Diabetes in America: Prevalence, Statistics, and Economic Impact. (2023). https://diabetes.org/about-diabetes/statistics/about-diabetes#:~:text=Overall%20numbers,5%2C300%20with%20type%202%20diabetes.**

Dataset. https://www.kaggle.com/datasets/johndasilva/diabetes

Checklist

🔍 Project Type 1: ML Problem & Solution

Goal: Pose a real-world problem as an ML task (classification, regression, clustering), train models, and evaluate.

✅ Checklist

- [x] Choose a **high-quality dataset** (from sources like Kaggle, UCI, OpenML, etc.)
- [x] Perform **Exploratory Data Analysis (EDA)**:
  - Correlations between features and target
  - Feature distributions and label histograms
  - Missing values and imputation strategy
  - Feature variance and standardization
  - Class imbalance analysis
  - Add TSNE or PCA visualization to visualize features and data to understand aspects like separability and so on.
- [x] If applicable, please perform **comprehensive feature engineering and feature selection** to identify the best features for your task. Look into different feature transformations (e.g., polynomial, categorical, count based, and so on).
- [x] Apply and compare **multiple ML models**:
  - Linear models (SVMs/Logistic/Linear Regression)
  - Tree-based models (Decision Trees, Random Forests, XGBoost)
  - Neural Networks (MLPs or CNNs if relevant)
- [x] Conduct **extensive hyperparameter tuning**:
  - Grid search / Random search / Bayesian optimization
  - Use of validation set or cross-validation
- [x] Provide **reasoning for hyperparameter choices**
- [x] Use **appropriate evaluation metrics**:
  - Accuracy, Precision, Recall, F1, AUC, RMSE, etc.
- [x] Include **visualizations** of results (confusion matrices, ROC curves, etc.)
- [x] Analyze and explain **what works and what doesn't**
- [x] Discuss **limitations** and future directions
- [x] Summarize **lessons learned and key takeaways**