Predicting Diabetes: A Demographic & Lifestyle Analysis of CDC Health Indicators

Team 13: Keito Taketomi, Nan Xiao, Shenghong Wu, Cassidy Gorsky, Obiora Ifeanyichukwu Okeke

1 Research Question & Goals

2 Data Acquisition & Preparation

3 Exploratory Data Analysis (EDA)

4 Feature Engineering & Preprocessing

5 Model Building & Comparison & Selection part1

6 Model Building & Comparison & Selection part2

Research Question & Goals

Which Demographic & Lifestyle Factors Best Predict Diabetes?

- ***Context & Motivation***
  - Rising diabetes prevalence → important to detect at-risk individuals early.
  - Large CDC dataset (~253k samples) with both demographic (Age, Sex, Income, Education) & lifestyle (BMI, Fruits/Veggies, Smoking, etc.) features.

- ***Primary Research Question***
  - ***"Which combination of demographic and lifestyle features most effectively predicts diabetes status (healthy vs. diabetic)?"***

- ***Project Objectives***
  - Acquire & clean data, ensuring no missing/duplicate entries.
  - Explore data relationships, detect anomalies.
  - Engineer features (composite scores, interactions) to capture synergy.
  - Build multiple models (Logistic, RF, XGBoost), tune for **recall** to catch as many diabetics as possible.
  - Compare & select final model, interpret key insights.

✔ ***Takeaway:*** We aim to minimize missed diabetic cases by focusing on recall, investigating which risk factors (demographic vs. lifestyle) drive diabetes the most.

Data Acquisition & Preparation

CDC Diabetes Health Indicators – Data Acquisition

- ***Source & Credibility***
  - UCI Repository (ID=891), originally from CDC BRFSS (Behavioral Risk Factor Surveillance System).
  - ~253,680 records, 21 features.
  - Target: *Diabetes_binary* (0 = healthy, 1 = diabetic/prediabetic).
- ***Loading & Validation***
  - Fetched using **ucimlrepo** library.
  - Checked metadata for consistency (shape: 253,680 × 23).
  - Confirmed presence of `Diabetes_binary` and created **DataFrame** (`df`) with features + target.
- ***Key Checks***
  - ID column handling: fallback by resetting index if missing.
  - Verified **no missing** entries (0 missing in X, 0 in y).
  - Found 0 duplicated IDs.

✔ ***Takeaway:*** Data is large, *no missing* & *no duplicates*, ideal for advanced modeling.

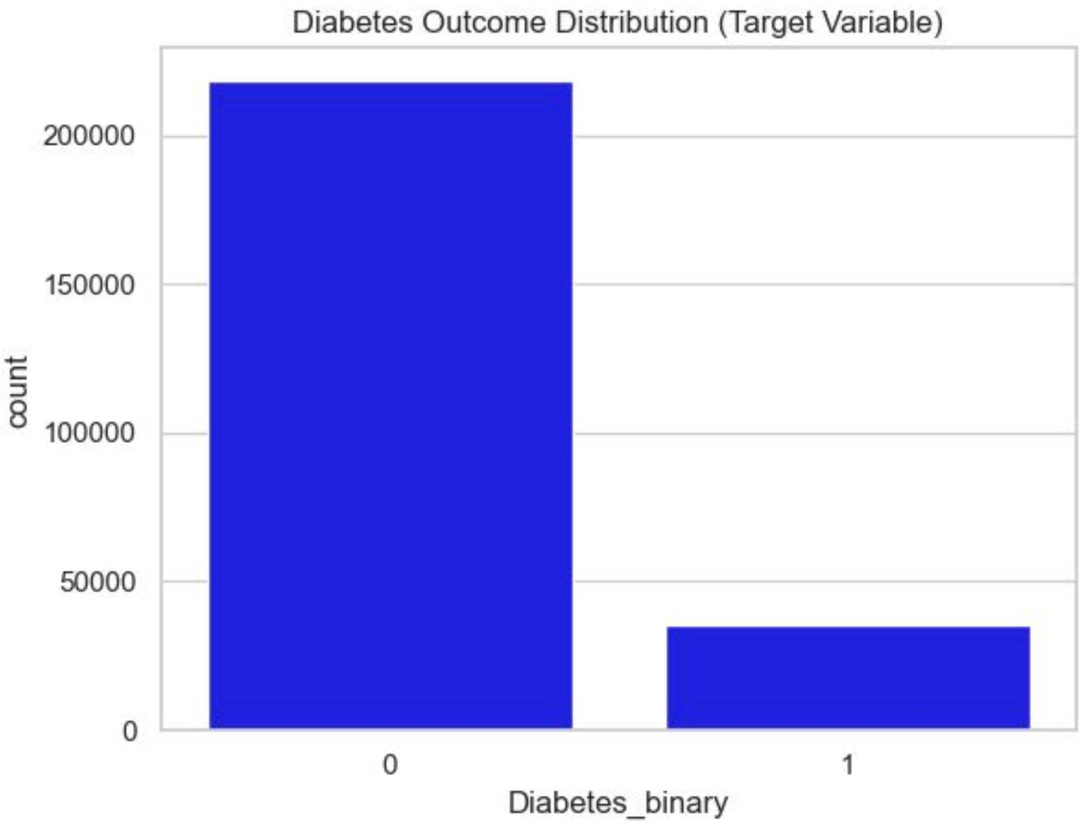
Data Cleaning & Preparation

1. ***Basic Stats & Data Validation***
   - Calculated descriptive stats: **BMI** (min=12, max=98).
   - Used **pandera** to enforce domain constraints (e.g., BMI ≤ 100, Age ≤ 13).
   - **Outliers**: none exceeding [10, 100] for BMI → no capping needed.
2. ***Correlation & Domain Checks***
   - Generated correlation heatmap.
   - Example: *Correlation(Age, BMI) ≈ –0.037* (low).
   - Confirmed *Age* in valid range (1..13).
3. ***Categorical Encoding***
   - One-hot for `Education`, `Income`, `Sex` as a **demo**.
   - Alternative: ordinal encoding if desired.
4. ***Clean Final Dataset***
   - **X** shape: (253,680 × 21) → raw features ready for feature engineering.
   - **y** shape: (253,680,) → binary label.

✔ ***Takeaway:*** Thoroughly checked data integrity, validated ranges, & prepared final features for next EDA/modeling steps.

Exploratory Data Analysis (EDA)

Target Variable



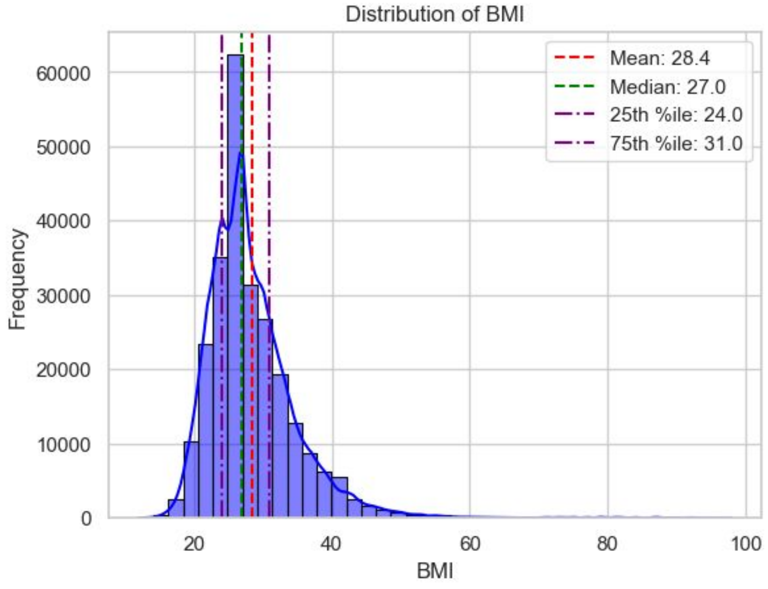
Number of records: 253,680

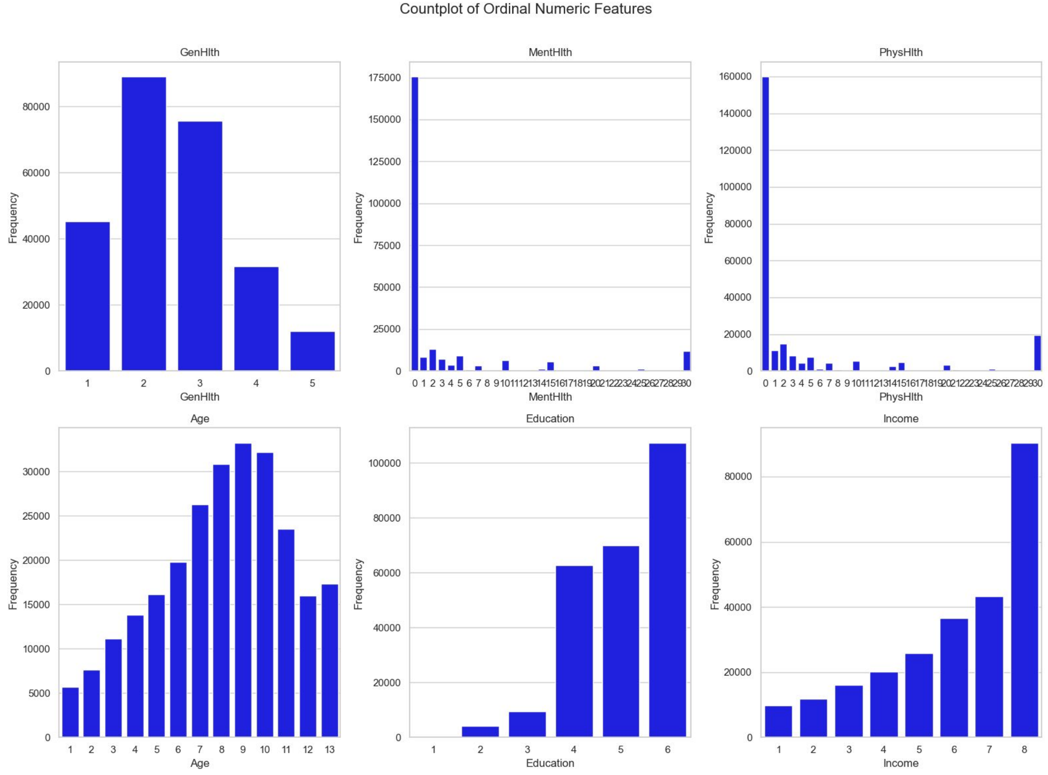
No diabetes (0): 218,334 / 86.07%

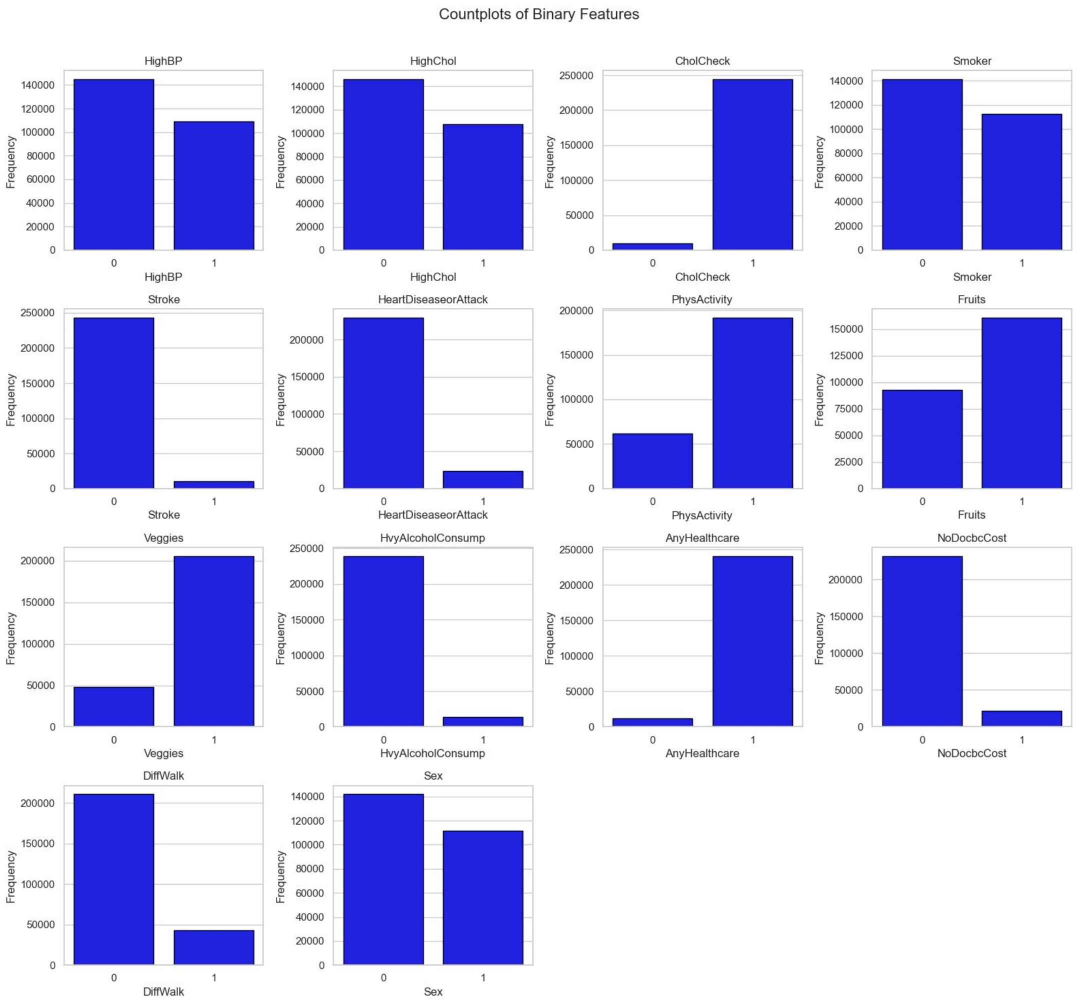
Prediabetes/ diabetes (1): 35,346 / 13.93%.

Distribution of features

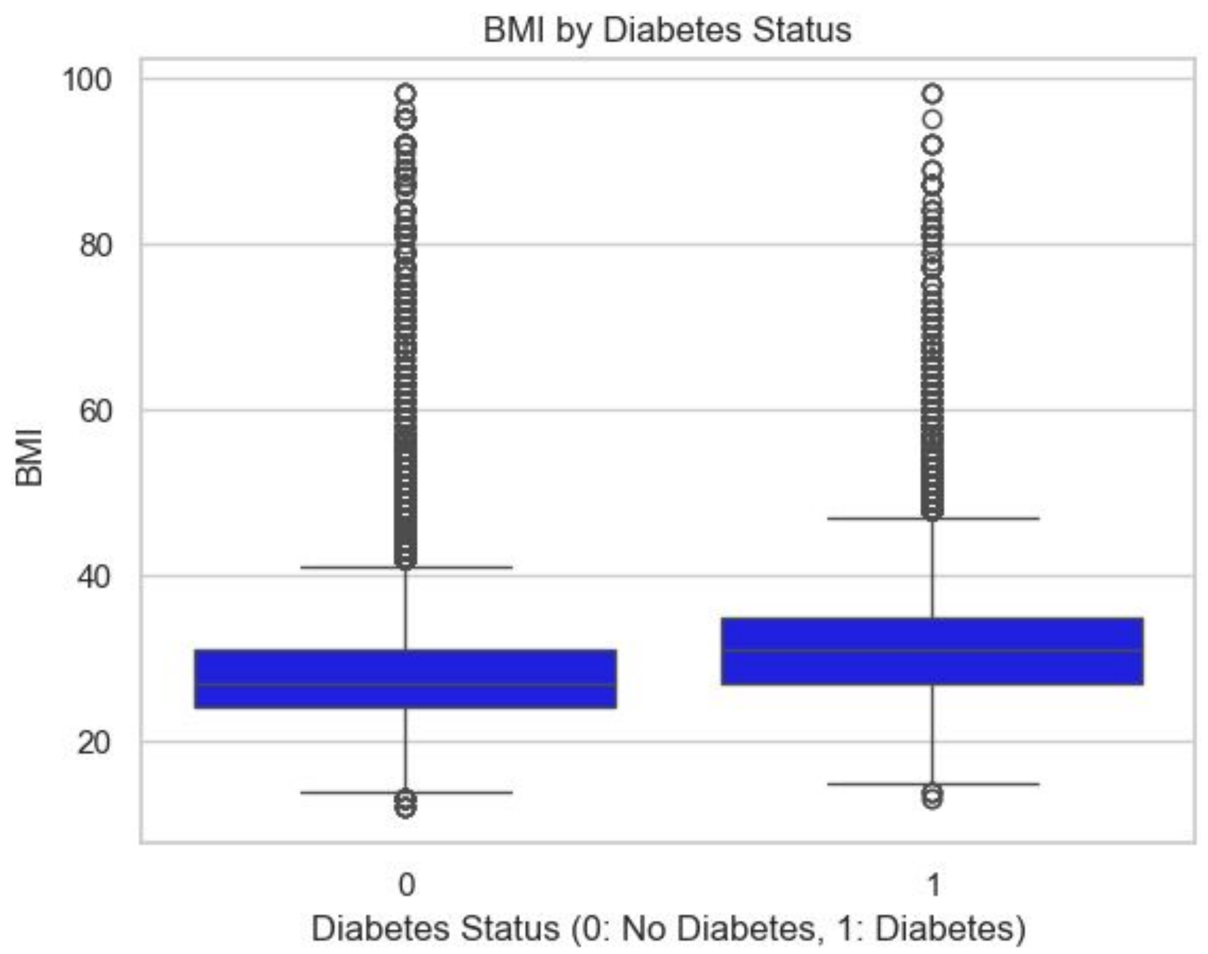
21 features: 1 continuous variable, 6 ordinal numeric variables, & 14 binary columns



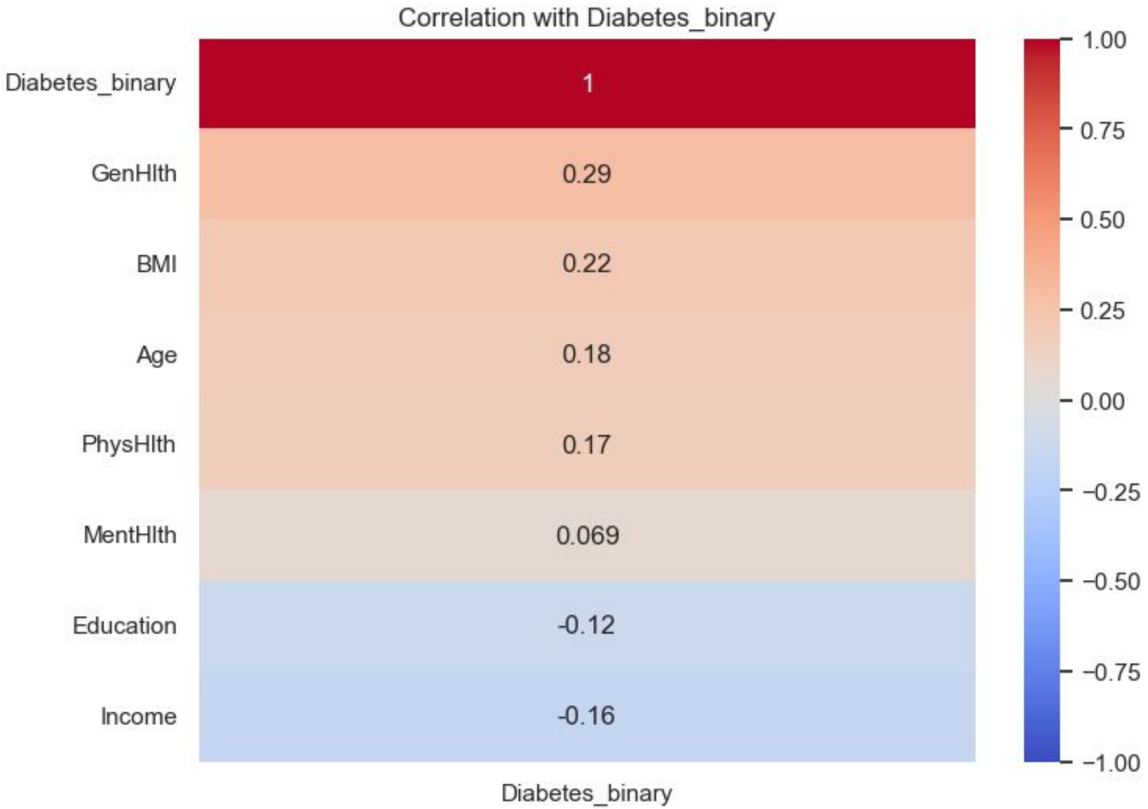




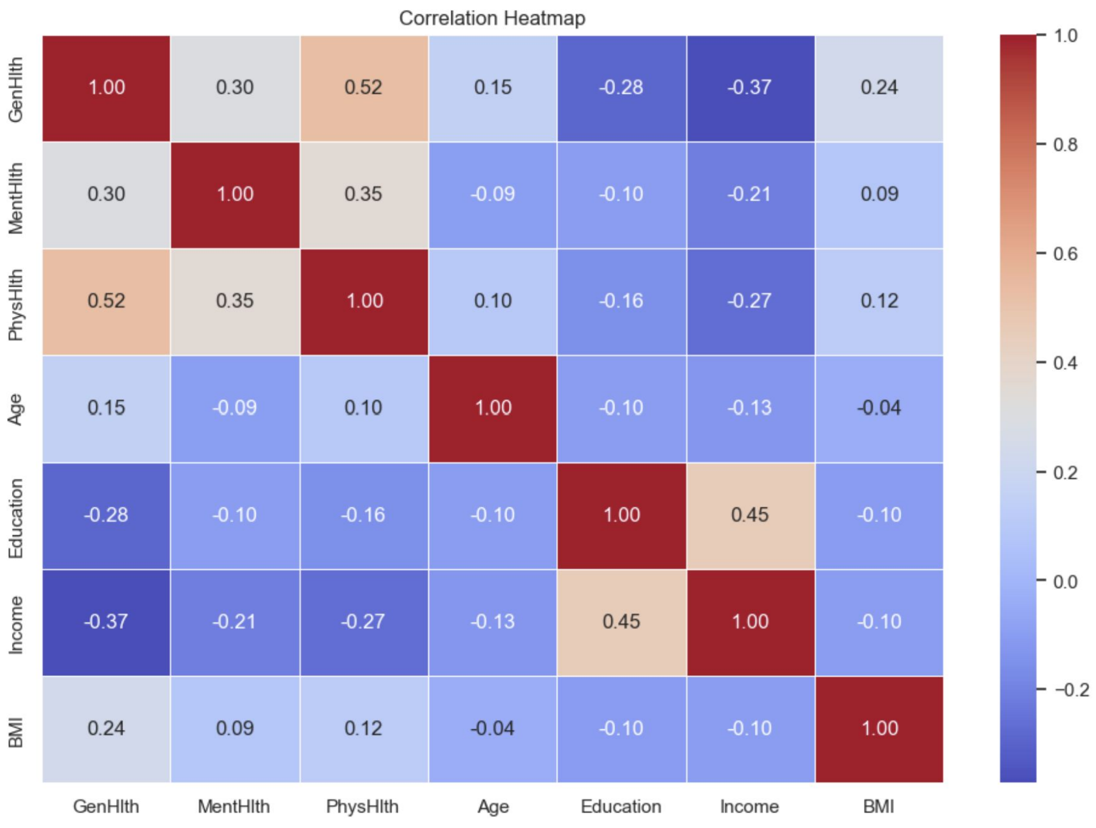
Countplots of Binary Features
HighBP
HighChol
CholCheck
Smoker
Stroke
HeartDiseaseorAttack
PhysActivity
Fruits
Veggies
HvyAlcoholConsump
AnyHealthcare
NoDocbcCost
DiffWalk
Sex
Frequency


BMI by Diabetes Status
100
80
60
40
20
BMI
0
1
Diabetes Status (0: No Diabetes, 1: Diabetes)

Numeric features vs Target Variable



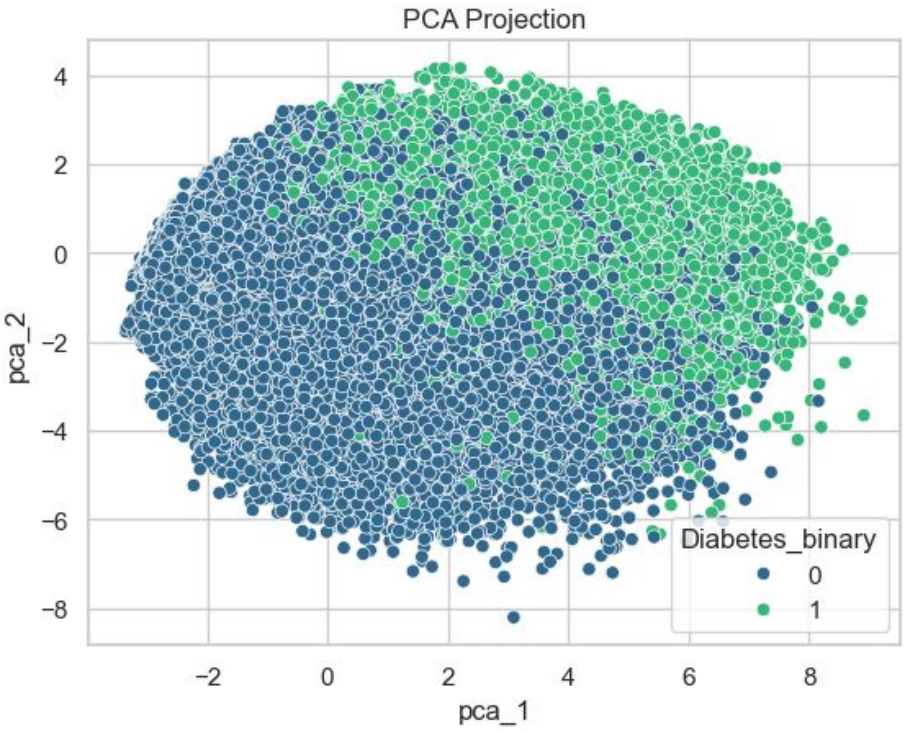
Correlation between numeric features



Unsupervised Learning Methods

- PCA for Dimensionality Reduction & Visualization
- K-means Clustering
- UMAP (Uniform Manifold Approximation and Projection)

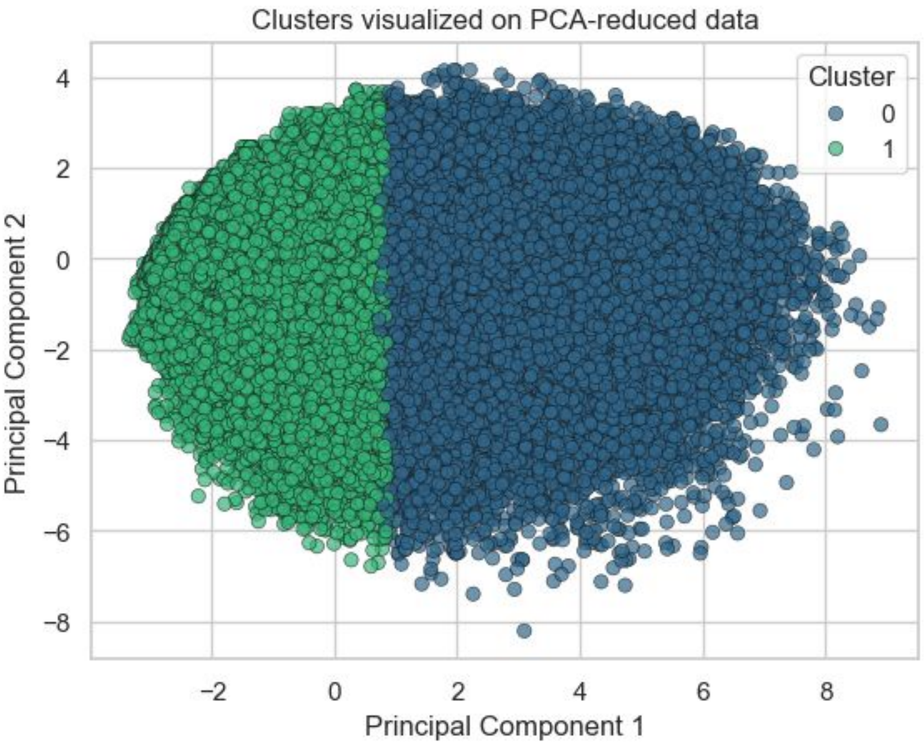
PCA



Goal: Do diabetes vs non-diabetes separate into clusters in PCA space?

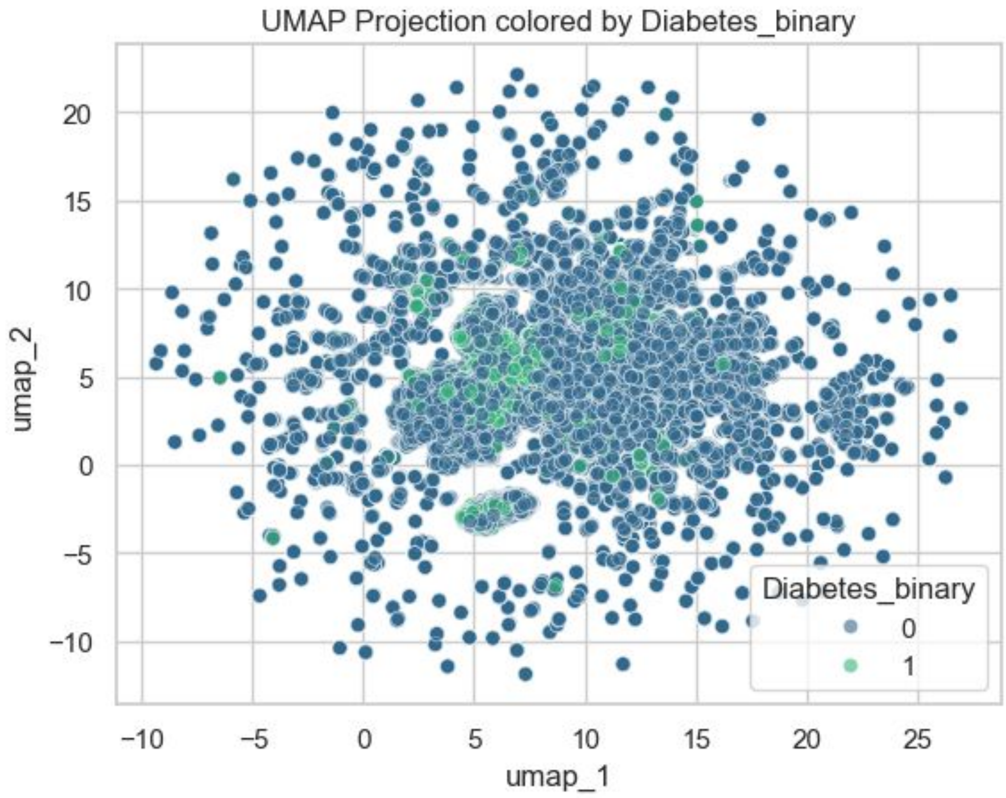
Together, the first 2 principal components explain ~24.8% of the total variance.

K- Means Clustering



Goal: Do the clusters roughly align with diabetic/ non-diabetic?

UMAP



This method works well for mixed data

Since both classes are mixed together, then the features may not separate well without supervision

Weak unsupervised signal for diabetes

Feature Engineering & Preprocessing

Creating Interaction and Combination Features

BMI Categories:

- Converts continuous BMI values into clinical categories: Underweight (<18.5), Normal (18.5-25), Overweight (25-30), and Obese (>30)
- Converts these categories into dummy variables (one-hot encoding)

Age Groups:

- Transforms the original age variable (which appears to be coded as ranges) into more intuitive groups: Young Adult, Middle Age, Senior, and Elderly. Also converts these to dummy variables

Composite Scores:

- `Health_Risk_Score`: Combines binary health risk factors (high blood pressure, high cholesterol, smoking, stroke history, heart disease) — so higher means "higher health risk"
- `Lifestyle_Score`: Combines positive lifestyle choices (physical activity, fruit and vegetable consumption) minus heavy alcohol consumption — so higher means "healthier habits"
- `Healthcare_Access`: Subtract "no doctor because of cost" from "any healthcare"—so positive means better access.

Interactions:

- `BMI_Age_Interaction`: BMI*Age, to capture their combined effect (e.g., maybe high BMI in older folks is extra risky)
- `Health_Activity_Interaction`: Health Risk × Activity, to see if exercise offsets risk.

Creating New Features

3. Add Polynomial Terms

- Squaring BMI and the health-risk score helps models capture nonlinear relationships (e.g., doubling BMI might not just double risk).

4. Ratio Features

- `GenHlth_PhysHlth_Ratio`: Ratio of general health score to physical health issues, reflects whether someone's self-rated health holds up against the number of days they actually felt unwell
- `BMI_PhysHlth_Ratio`: Ratio of BMI to physical health issues, high ratio -> a higher BMI but few health complaints

5. Log-Transforms for Skewed Data

- `PhysHlth` and `MentHlth` (days of bad physical/mental health) are heavily skewed, so we take logarithmic transformations `log1p` (log(x + 1)) to smooth out the extremes.

6. Feature Scaling

- Excludes binary and dummy variables from scaling
- Then we apply standardization to numerical features: shifting them to have mean = 0 and standard deviation = 1.

Feature Selection and Visualization

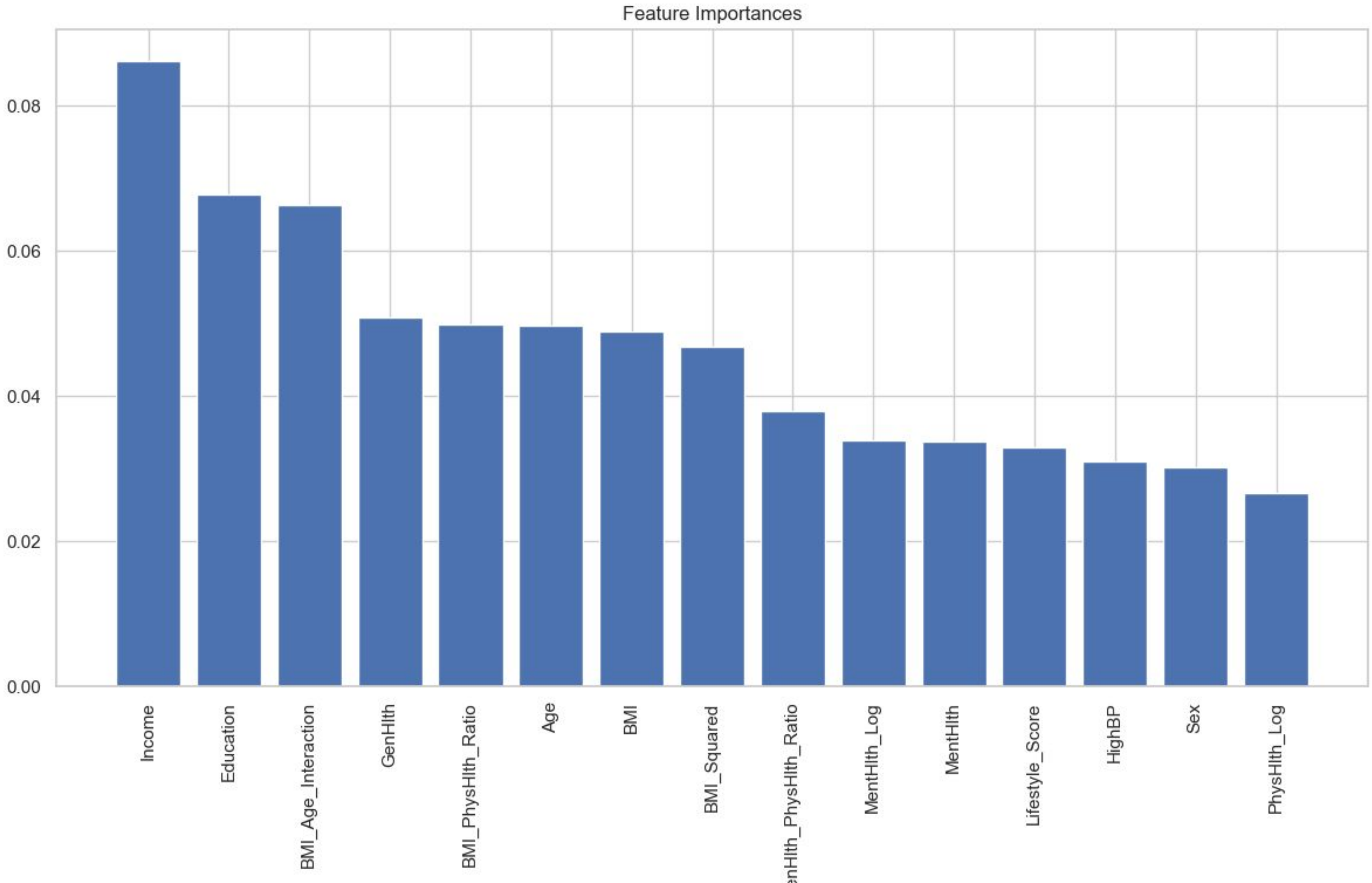
7. Feature Selection with a Tree-Based Model

- We train an `ExtraTreesClassifier` on all these features to see which ones carry the most predictive power.
- Then we plot the top 15 most important features—kind of like a tournament ranking.
- We pick the top 15 features and create a pared-down dataset (`X_fe_selected`) that keeps just those winners.

8. Correlation Heatmap

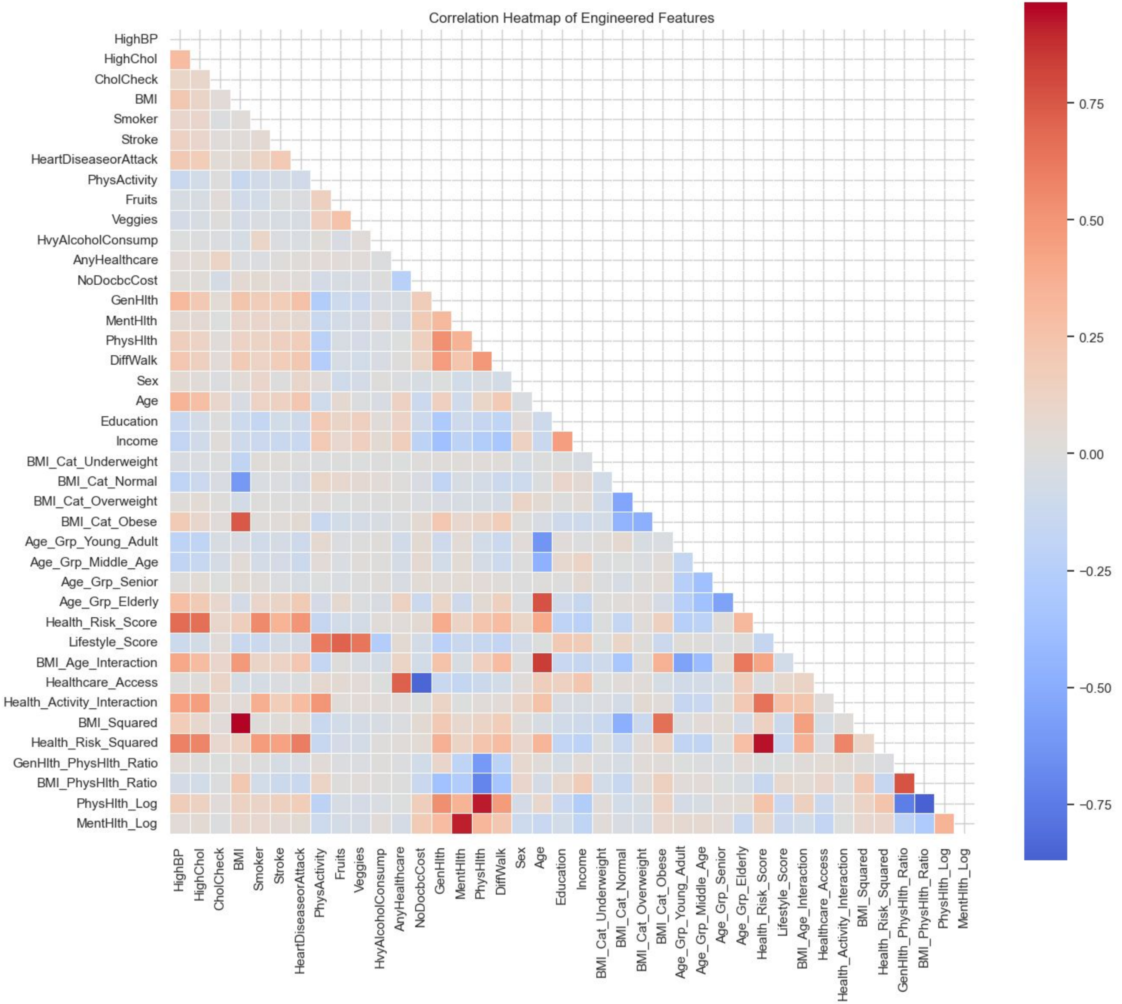
- To check for redundant features (ones that are nearly copies of each other), we draw a heatmap of all feature-to-feature correlations. If two features are super highly correlated, you might consider dropping one.

Feature Importance



Correlation Heatmap

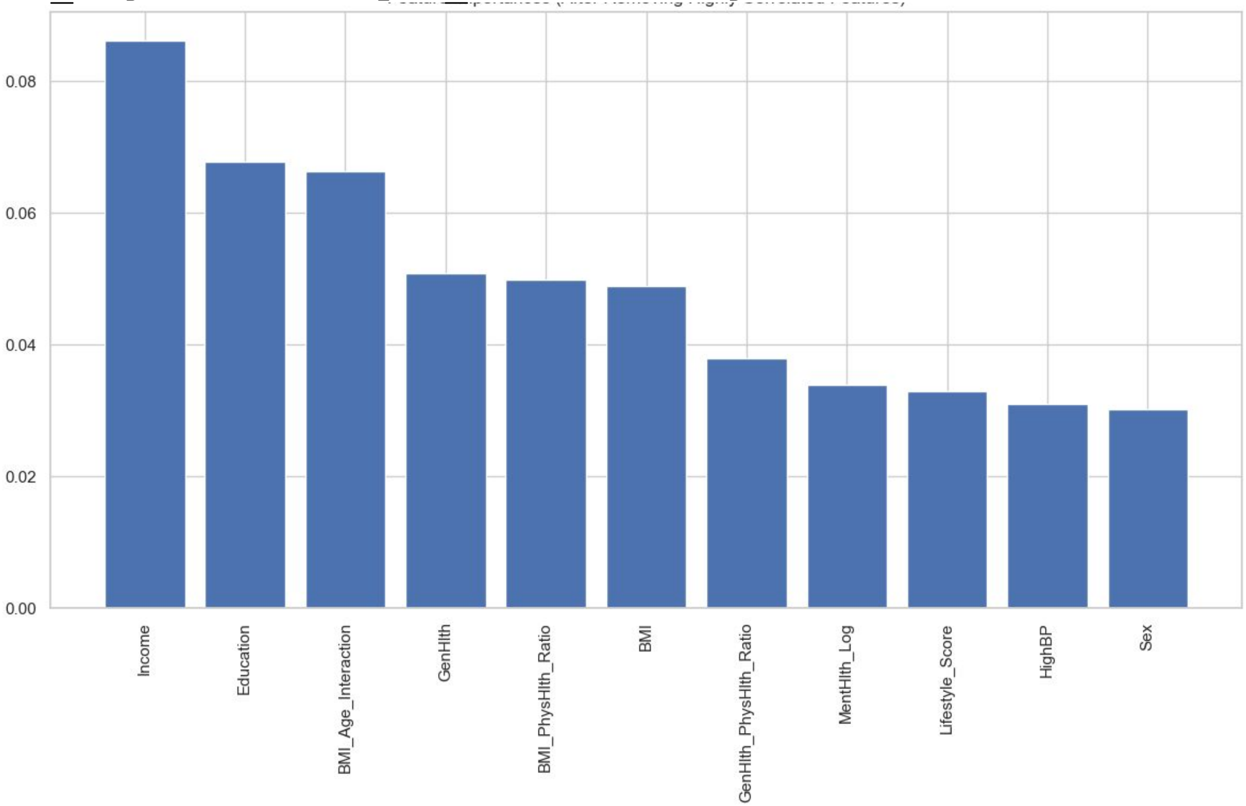
- To check for redundant features (ones that are nearly copies of each other), we draw a heatmap of all feature-to-feature correlations.
- **<u>If two features are super highly correlated (>0.8), we consider dropping the one that has less importance, and keep the feature with higher importance.</u>**



```
Correlation details:
Age (importance: 0.049740) is correlated with:
  - BMI_Age_Interaction (importance: 0.066254), correlation: 0.8396
PhysHlth_Log (importance: 0.026675) is correlated with:
  - PhysHlth (importance: 0.026150), correlation: 0.9209
  - BMI_PhysHlth_Ratio (importance: 0.049919), correlation: 0.8692
NoDocbcCost (importance: 0.009956) is correlated with:
  - Healthcare_Access (importance: 0.011943), correlation: 0.8422
MentHlth (importance: 0.033704) is correlated with:
  - MentHlth_Log (importance: 0.033963), correlation: 0.9129
BMI_Squared (importance: 0.046823) is correlated with:
  - BMI (importance: 0.048982), correlation: 0.9649
PhysHlth (importance: 0.026150) is correlated with:
  - PhysHlth_Log (importance: 0.026675), correlation: 0.9209
Health_Risk_Squared (importance: 0.018384) is correlated with:
  - Health_Risk_Score (importance: 0.021301), correlation: 0.9317
```

Final Feature Selection

```
Final selected features: ['Income', 'Education', 'BMI_Age_Interaction',
'GenHlth', 'BMI_PhysHlth_Ratio', 'BMI', 'GenHlth_PhysHlth_Ratio',
'MentHlth_Log', 'Lifestyle_Score', 'HighBP', 'Sex']
```



Model Building & Comparison & Selection part1

Model Overview

Goal

- Our goal is to build a model that effectively predicts whether an individual has diabetes.

  Since only about 17% of the data represent positive cases (people with diabetes),

  we focus primarily on recall which reflects the model's ability to identify those individuals.

Model Selection

We selected three representative supervised learning models to compare and optimize:

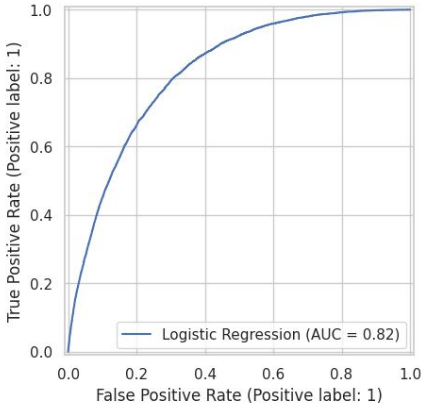
- **Logistic Regression**: A simple linear baseline model;
- **Random Forest**: A nonlinear ensemble model capturing more complex patterns;
- **XGBoost**: A powerful boosting algorithm widely used for structured data.

Modeling Procedure

- **Round 1**: Baseline modeling, to assess initial performance;
- **Round 2**: Grid search tuning, aiming to improve recall;
- **Round 3**: Fine-tuning, making smaller adjustments to optimize the models further.

Round 1 – Baseline Model Training and Comparison

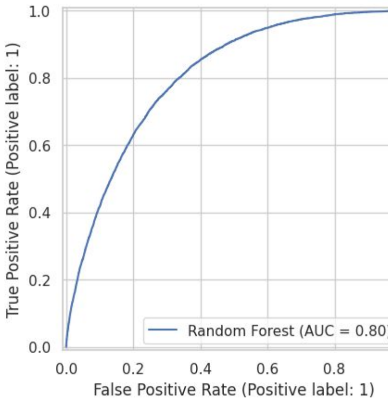
Try three baseline models (Logistic Regression, Random Forest, XGBoost) and compare their performance. Focus on finding which one gives a good balance between recall and overall performance.

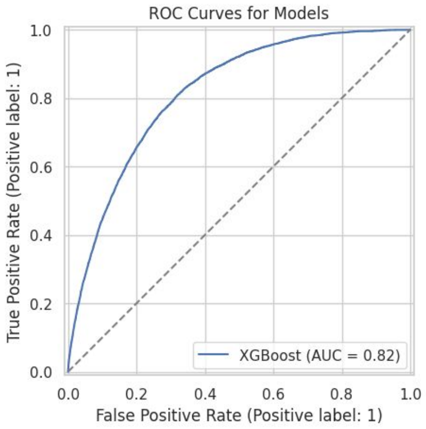


Method

- Selected three baseline models: **Logistic Regression**, **Random Forest**, and **XGBoost**
- Applied **GridSearchCV** with basic hyperparameter grids
- Used **5-fold Cross-Validation** to validate model performance
- Chose **Recall** as the selection metric for identifying the best model

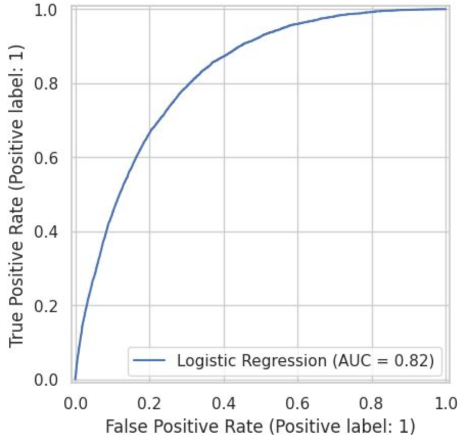
Model	Accuracy	Precision	Recall	F1	ROC AUC
LR	0.863568	0.533034	0.167775	0.255218	0.818848
RF	0.860769	0.501080	0.164097	0.247229	0.804426
XGB	0.863292	0.530131	0.165511	0.252264	0.817038





Round 2 - Parameter Optimization with Recall Priority

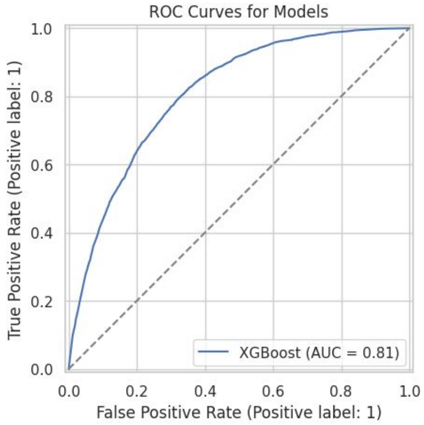
To improve the model's ability to identify diabetic cases by increasing recall, even if it comes at the cost of slightly lower precision or overall accuracy.

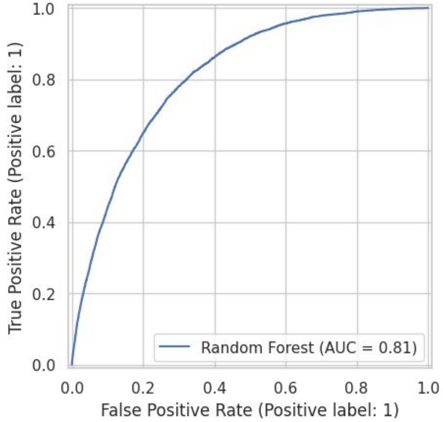


Method

- Kept the same three models: **Logistic Regression**, **Random Forest**, and **XGBoost**
- Expanded hyperparameter search space for each model
- Focused on **recall** as the optimization target to better capture minority (positive) class
- Applied **GridSearchCV** with 5-fold cross-validation for fine-tuning
- Selected the model with the highest **recall score** on the validation set

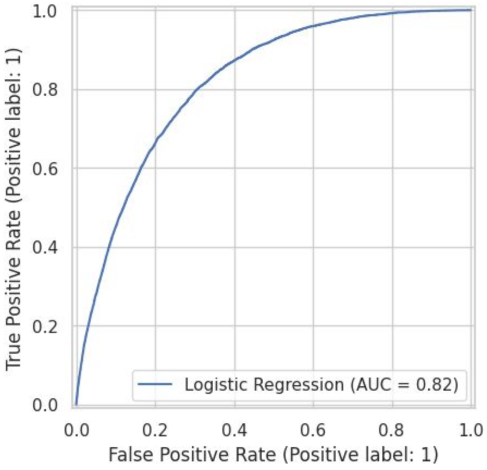
Model	Recall	ROC AUC
LR	0.762626	0.819072
RF	0.783703	0.813351
XGB	0.799264	0.809414





Round 3 - Fine-Tuning for Stability and Trade-Off Balance

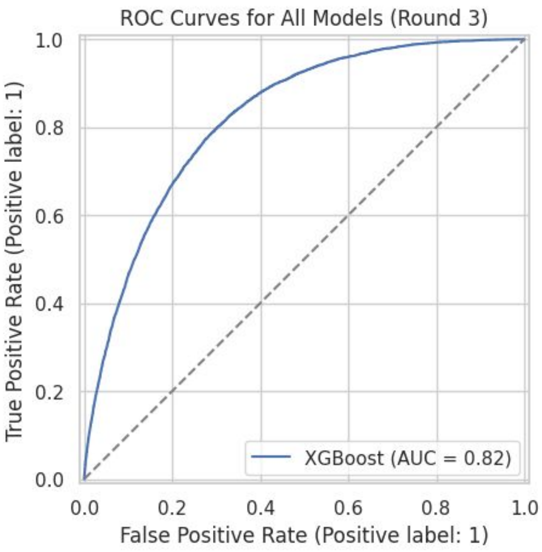
To slightly improve recall while avoiding performance overfitting, and explore more stable hyperparameters.

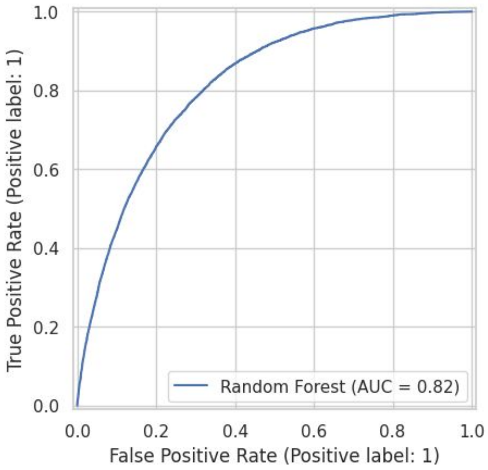


Method

- Continued with **Logistic Regression**, **Random Forest**, and **XGBoost**
- Narrowed the hyperparameter range based on previous round’s results
- Focused again on **recall** as the key evaluation metric
- Applied **GridSearchCV** with fine-tuned settings for faster iteration
- Compared models on validation set using **recall** and **AUC**
- Re-trained the best model on full dataset for final evaluation

Model	Accuracy	Precision	Recall	F1	ROC AUC
LR	0.863568	0.533034	0.167775	0.255218	0.818848
RF	0.863016	0.579228	0.061536	0.111253	0.816175
XGB	0.864948	0.553900	0.157731	0.245541	0.823524





Conclusion

Key Takeaways

- XGBoost achieved the highest recall (~80%) and is preferred when missing diabetics is costly.
- Logistic Regression remains more interpretable and nearly matched XGBoost's performance.
- Model tuning with a recall-first mindset significantly improved minority class detection.

Insights

- Lifestyle and demographic features (e.g., Age, BMI, HighBP) are crucial predictors.
- Trade-offs exist: higher recall comes with slightly lower precision or accuracy.
- Depending on healthcare context, different models may be prioritized.

Final Choice

- We recommend XGBoost for deployment in early diabetes screening tasks.
- Future work may include threshold optimization and SHAP-based interpretability.

Model Building & Comparison & Selection part2

Model Overview

Model Selection

We selected three additional supervised learning models to compare and optimize, with a focus on recall-based performance:

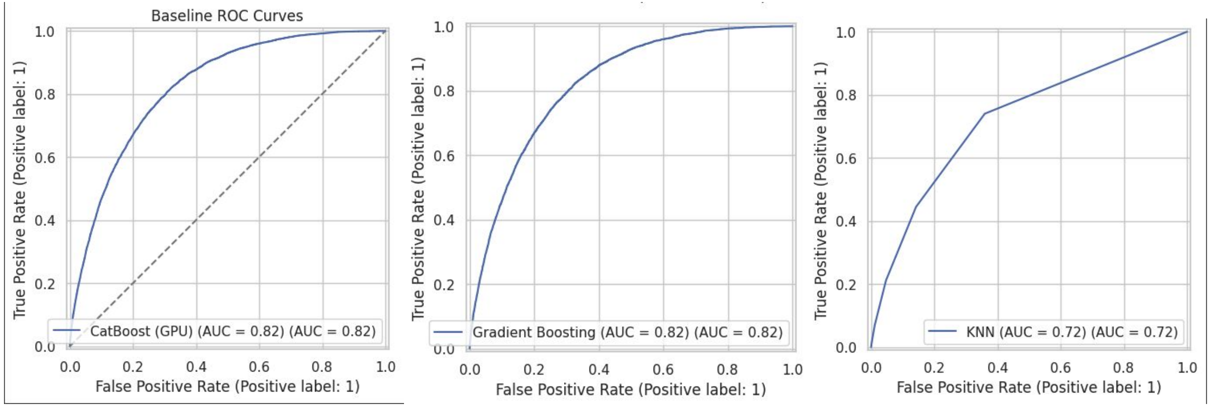
- **K-Nearest Neighbors (KNN):** A simple, non-parametric model that classifies based on proximity in feature space — often effective for recall in imbalanced datasets.

- **Gradient Boosting:** An ensemble method that builds trees sequentially to reduce errors — known for strong accuracy and generalization in tabular data.

- **CatBoost (GPU):** A gradient boosting algorithm optimized for categorical features and GPU acceleration — delivers fast, accurate performance with minimal preprocessing.

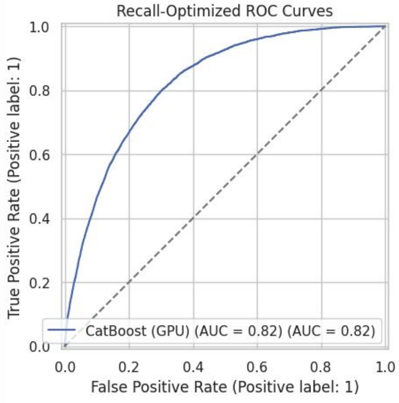
Baseline

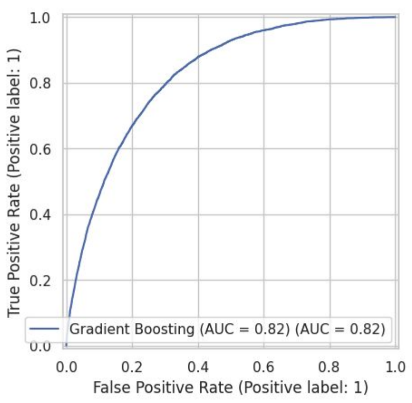
- KNN had the highest recall
- Gradient Boosting and CatBoost had higher overall accuracy and AUC
- CatBoost was the top performer in AUC

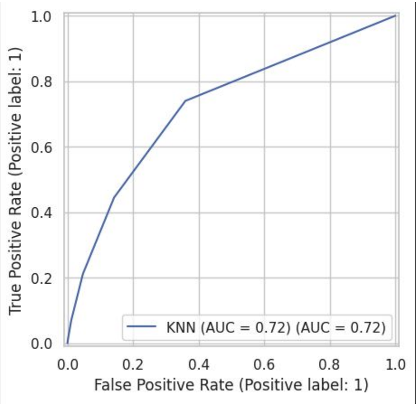


Model	Accuracy	Precision	Recall	F1	ROC AUC
KNN	0.849417	0.419826	0.211487	0.281279	0.722380
GB	0.864278	0.545681	0.154619	0.240961	0.822608
CatBoost	0.865086	0.557613	0.153346	0.240541	0.824206

Recall Optimized Results





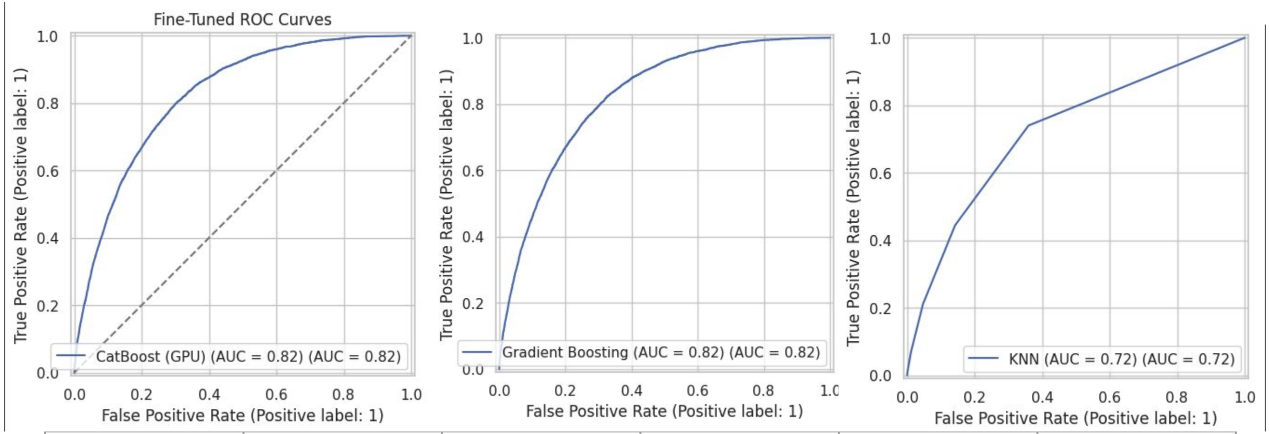


Model	Accuracy	Precision	Recall	F1	ROC AUC
KNN	0.849417	0.419826	0.211487	0.281279	0.722380
GB	0.864298	0.545954	0.154619	0.240988	0.822619
CatBoost	0.864869	0.556260	0.148960	0.234992	0.823286

- KNN’s recall did not improve further
- CatBoost and Gradient Boost both retained its balance, but didn’t significantly improve recall either
- indicating possible structural limitations in the dataset

Fine-Tuned Results

- All performance metrics remained consistent



Model	Accuracy	Precision	Recall	F1	ROC AUC
KNN	0.849417	0.419826	0.211487	0.281279	0.722380
GB	0.864298	0.545954	0.154619	0.240988	0.822619
CatBoost	0.864869	0.556260	0.148960	0.234992	0.823286

Conclusion

Key Takeaways

- KNN consistently achieved the highest recall (~21%), making it the most effective at identifying diabetic individuals.
- Gradient Boosting and CatBoost offered stronger overall accuracy and ROC AUC but missed more diabetic cases.
- Tuning hyperparameters with a recall-first mindset did not improve recall.

Insights

- Demographic and lifestyle features (e.g., Age, BMI, HighBP, PhysActivity) remain strong predictors of diabetes.
- There is a clear trade-off between recall and model calibration — higher recall models may sacrifice precision.
- Depending on healthcare context, different models may be prioritized.

Final Choice

- XGBoost still outperforms the models in part 2

Interpretation & Conclusions

Interpreting Our Final Model & Overall Findings

- ***Best Model***
  1. **XGBoost** gave highest recall (~80%) when class weighting was used.
  2. **Logistic Regression** had slightly lower recall but higher interpretability.
  3. Final choice depends on **healthcare context**: if missing a diabetic is costly, XGBoost is favored.
- ***Feature Importance*** *(from ExtraTrees / XGBoost)*
  1. **Age**, **BMI**, **HighBP**, **HighChol**, **PhysActivity**, and composite scores (e.g., *Health_Risk_Score*) ranked highly.
  2. Demographic variables (*Age, Income, Education*) also strongly correlated with diabetes status.
- ***Conclusions***
  1. **Demographics + Lifestyle** both matter; synergy features (e.g., *BMI* × *Age*) improved performance.
  2. **Tuning for recall** can detect more diabetics but lowers overall accuracy—trade-off is context-driven.
  3. The final model can help **health practitioners** target at-risk groups (older, obese, lower-income/education).

✔ ***Next Steps:*** Deploy final XGBoost for early screening or refine threshold. Consider interpretability tools (SHAP) to better explain risk predictions to clinicians.

Questions?