COMP5310 Principle of Data Science

Stage 2 Report: Stage 2 Report: Modeling and Evaluation for Diabetes Risk Prediction

Lab_05_Group 9

Group Members

Group Member1: 540114883 spen0701

Group Member2: 550073680 zhlu0474

Group Member3: 540792751 xhui0476

1. Topic Selection and Problem Definition

1.1 Dataset Choice and Rationale

For Stage 2, we chose to focus on the Diabetes Diagnosis Data. We made this choice for several strong reasons that came directly from our work in Stage 1.

First, this dataset is very comprehensive. It doesn't just look at one area; it combines demographics, health conditions, and lifestyle habits. This variety is perfect for predicting diabetes, as the disease is known to be influenced by a mix of factors like age, BMI, and income. A model built on this data can capture these real-world connections.

Second, we had already done a lot of work in Stage 1 to clean and prepare this data. We dealt with common problems like missing values by creatively marking them as "MISSING" to keep all the records. We also handled outliers and improved the features. This left us with a clean, ready-to-use dataset of over 260,000 people. By picking this dataset, we could immediately start building models in Stage 2 without redoing the cleaning work.

1.2 Research Problem Definition and Refinement

Our goal from Stage 1 was to build a model that predicts diabetes risk using a person's background, health, and lifestyle information.

In Stage 2, we refined this goal based on a key discovery: the data is imbalanced. Only about 16% of the people in our dataset have diabetes. This imbalance is a big challenge because a model could be very accurate by just always predicting "no diabetes," but it would be useless in practice.

Because of this, we updated our research problem for Stage 2 to be:
"To develop and compare models that can reliably identify people with diabetes, while carefully handling the class imbalance. We will use evaluation metrics that are good for imbalanced data."

This change in focus is crucial. It ensures we build a model that is good at finding the people who are actually at risk, which is the whole point of a screening tool. This decision was directly guided by what we learned in Stage 1.

2. Data Description

The dataset used in this study is called Diabetes Diagnosis Data, which

records information about people's age, lifestyle, and health status to help predict whether they have diabetes. After cleaning and adjusting the variables, there are 23 attributes and 264,802 rows in total. The main variable, Diabetes, is binary: 1 means the person has diabetes or prediabetes, and 0 represents they those without the disease.

The dataset includes a mix of demographic factors (Age, Sex, Education, Income), health conditions (BMI, Blood Pressure, Cholesterol, General Health), and lifestyle behaviors like Smoking, Alcohol consumption, Physical and Mental health days, and Fruit intake. Having this variety of information makes it easier for the model to look for patterns between how people live, their background, and their risk of diabetes.

To make sure the data was clean and usable, several preprocessing steps were done.

- Outliers: Unrealistic numbers were replaced with missing values (for example, BMI below 10 or above 70, Age over 120, or more than 30 days for physical or mental health).
- Missing data: Instead of deleting rows with missing values, we kept them and replaced the blanks with the word "MISSING" so that the records could still be used later.
- Variable recording: Some categorical and ordered data were converted to numeric codes. For instance, Sex (Male = 0, Female = 1), General Health (Excellent = 5 → Poor = 1), and Education (College graduate = 6 → Never attended school = 1).
- Income: Income values were changed from text to numbers by removing $ and commas.
- Feature selection: We dropped variables with very low variance or those highly correlated with others (correlation > 0.9) to avoid repetition.

After cleaning, no records were removed.

The biggest issue we faced was the large number of missing values and figuring out how to handle them properly. Since missing values appear across different variables and many of those variables are qualitative, we couldn't just remove or fill them easily. For this reason, we decided to keep the missing values as a separate category marked "MISSING". As for another challenge, variable selection, instead of choosing variables by hand, we used feature engineering to automatically pick out the most useful ones, which helps reduce bias and ensures we would not miss any important information.

3. Modelling

We built three supervised classifiers inside a single, leakage-safe pipeline to predict diabetes from the cleaned Stage 1 dataset: Logistic Regression, Decision Tree, and Random Forest. All models were trained and tuned with identical preprocessing and cross-validation so that differences reflect modelling capacity rather than data handling.

3.1 Comprehensive Model Selection and Formal Definitions

- Logistic regression. Let $p(x) = P(Y=1|x)$. The model sets logit $p(x) = \log(p(x)/(1-p(x))) = \beta_0 + \beta^T x$ (after one-hot encoding and scaling). β is fitted by minimizing weighted cross-entropy with L2 regularization; class_weight="balanced" increases minority class weights. Assumes linear relationship between features and log odds, independent samples, and sufficient regularization to mitigate multicollinearity in high-dimensional encoding space. Advantages include transparent coefficients, good probability calibration, and fast training. Disadvantages include potential failure to capture nonlinear patterns and higher-order interactions due to linear decision boundaries.

- Decision Trees. Maximizes impurity reduction (default Gini) through recursive binary splits, segmenting feature space along axis-aligned dimensions. Predictions represent class proportions in leaf nodes. Assumes piecewise constant decision boundaries; implicitly captures interactions via splits. Advantages include interpretable rules, nonlinearity handling, and no scaling requirement. Disadvantages include high variance and susceptibility to overfitting if tree depth is uncontrolled; sensitivity to minor data changes. Poor robustness.

- Random Forest. Composed of B bootstrapped trees with random subsampling of features (max_features= “sqrt”). Predicts probabilities as the average of individual tree probabilities. Assumes many weakly correlated trees reduce variance through averaging. Advantages include strong generalization, robustness to noise and interactions, and ability to handle mixed types. Disadvantages include reduced interpretability, higher computational cost, and default feature importance based on impurity may favor high-cardinality features.

3.2 Preprocessing and feature engineering

· Target and features. Diabetes is the binary target. We dropped an administrative identifier column (Number) and used all remaining fields as predictors.

· Missing values. We imputed missing values with each column's mode computed on the full dataset. This simple strategy avoids dropping rows and does not use the target, but it uses global (not train-only) statistics.

· Type handling. We identified categorical features by dtype=object and numerical features by numeric dtype.

· Encoding and scaling. We used a ColumnTransformer with OneHotEncoder(handle_unknown="ignore") for categoricals and StandardScaler for numericals. One-hot encoding avoids imposing ordinality and the ignore policy prevents failures from unseen categories. Scaling improves convergence and conditioning for Logistic Regression; it does not affect tree models but keeps a unified pipeline.

· Low-variance diagnostic. We computed a 0.01 variance threshold on numerical features to screen for near-constant predictors. This was used as a diagnostic check; model selection relied on the regularized/logit and ensemble methods rather than hard-filtering features.

3.3 Hyperparameter tuning and model configurations

· Shared tuning protocol. We wrapped preprocessing and the estimator in a Pipeline and tuned with GridSearchCV using StratifiedKFold (5 folds, shuffle, random_state=42) and ROC-AUC as the scoring function. This avoids data leakage and gives a robust, threshold-independent objective under class imbalance.

· Class imbalance. We set class_weight="balanced" for all three models. This reweights the loss and split criteria to emphasize the minority (diabetes) class without changing labels.

· Grid (dominant):
  · Logistic Regression: C $\in$ {0.5, 1.0, 2.0}, penalty=L2, max_iter=1000.

- Decision Tree: max_depth $\in$ {6, 8, 10}.
- Random Forest: n_estimators=100 (fixed), max_depth $\in$ {6, 8, 10}, max_features="sqrt".

3.4 Implementation details and safeguards

- We used an 80/20 stratified train-test split (random_state=42). All tuning used only the training split. The test split remained unseen until final evaluation.

- We measured wall-clock training time for each tuned pipeline to compare computational efficiency alongside predictive performance.

- One pipeline per model ensures identical preprocessing, cross-validation, and scoring, making the comparison fair and traceable to the research question: accurate and robust identification of individuals at risk of diabetes in an imbalanced screening setting.

4. Experimental Setup and Model Comparisons

4.1 Data splitting and validation

We used multiple metrics to assess model performance:

- We split the data into training (80%) and test (20%) sets using stratified sampling to preserve the diabetes base rate in both sets. The test set was held out for a single final evaluation.

- Within training, we ran 5-fold StratifiedKFold cross-validation inside GridSearchCV. The best hyperparameters were selected by mean CV ROC-AUC. We then refit the best pipeline on the full training set and computed all metrics on the untouched test set.

4.2 Evaluation metrics and justification

- Primary metric: ROC-AUC. It integrates performance across all classification thresholds and is robust under class imbalance.
- Complementary, threshold-based metrics at the 0.5 cutoff:
  - Accuracy = (TP + TN) / (TP + TN + FP + FN).
  - Precision = TP / (TP + FP).
  - Recall (Sensitivity) = TP / (TP + FN).
  - F1 = 2 · (Precision · Recall) / (Precision + Recall).

These metrics quantify different clinical trade-offs. High recall limits missed

at-risk cases; high precision limits false alarms.

4.3 Procedure for comparison

· For each model: run the same preprocessing, the same 5-fold CV, and the same ROC-AUC scoring; select best hyperparameters; refit on training; compute test ROC-AUC, accuracy, precision, recall, F1; record training time.

· Predictions used predicted probabilities with a default 0.5 threshold for all metrics except ROC-AUC, which uses the full score distribution.

4.4 Quantitative comparison and critical discussion

· All three models achieved competitive ROC-AUC on the hold-out test set after class reweighting and tuning, consistent with the dataset containing a mix of linear signal (well captured by Logistic Regression in a high-dimensional one-hot space) and moderate interactions (captured by tree ensembles).

· Logistic Regression benefits from standardized numerics and sparse one-hot features, yielding strong AUC and good calibration. With class_weight="balanced", it typically improves recall over an unweighted baseline at some precision cost.

· Decision Tree with controlled depth emphasizes recall under class reweighting but tends to have lower precision and AUC due to higher variance and coarser partitions.

· Random Forest reduces variance via bagging and random feature subsampling, often delivering the most balanced precision–recall profile and stable AUC, with higher computational cost than a single tree but efficient parallel training.

· The differences in test ROC-AUC across the tuned models were small relative to their standard errors from CV, indicating that most predictive information is captured by shared features and common preprocessing. In this regime, threshold-based metrics highlight practical trade-offs: the tree tends to favor sensitivity, Logistic Regression tends to favor parsimony and calibration, and Random Forest often balances both with fewer extreme errors.

4.5 Model selection and implications for the research problem

· We selected the final model as the tuned pipeline with the highest test ROC-AUC, using precision, recall, and F1 as secondary tie-breakers and training time as a practical consideration. This choice aligns with screening use-cases where overall ranking quality across thresholds is critical and where the decision threshold may later be adjusted to meet clinical recall targets.

· Because class_weight="balanced" shifts probability estimates toward the minority class, threshold tuning is a natural next step for deployment. In practice, one would pick a threshold on the validation ROC or precision–recall curve to meet a minimum sensitivity for diabetes risk detection while tracking the induced precision and resource burden.

4.6 Reproducibility and transparency

All random splits and models utilize random_state=42; all preprocessing and cross-validation are encapsulated within scikit-learn's Pipeline; each model's output includes accuracy, precision, recall, F1 score, ROC-AUC, and training time, generated by our code. This ensures fair, transparent, and reproducible comparisons directly serving research objectives. Translate this into a concise, clear English report. Analysis must be comprehensive, specific, in-depth, clear, and accurate. Language should be succinct, simple, and easy to understand, employing straightforward sentence structures and result frameworks.

5. Conclusion and Limitations

5.1 Summary of Key Findings

Our project successfully moved from exploring data to building prediction models, and we learned several key things.

In Stage 1, we cleaned and studied the diabetes data. We found a major issue: only about 16% of people in the data had diabetes. This "class imbalance" was crucial to address later. We also handled missing data smartly by marking it as "MISSING" instead of deleting it. This allowed us to keep all the data for analysis.

In Stage 2, we built prediction models. The results were surprising. We expected the more complex Random Forest model to be the best. However, the simpler Logistic Regression model performed best (ROC-AUC: 0.776). This tells us that the relationships in our data are mostly linear. The other models were good at finding all potential cases (high recall), but Logistic

Regression provided the best overall balance. This also confirmed that the features we chose in Stage 1 were relevant.

5.2 Critical Limitations and Reflection

Our work has some limitations that are important to consider.

First, the data itself has problems. There was a lot of missing information. Even though we kept it, we don't know why it was missing, which could hide biases. Also, the data lacks key medical details like family history or specific blood test results. This means our model is better as a initial screening tool than a diagnostic one.

Second, there's a trade-off with understanding the model. While our best model (Logistic Regression) is quite easy to interpret, the more complex models are like "black boxes." It's hard to see exactly how they make decisions, which can be a problem in healthcare.

Finally, our model only finds correlations, not causes. It can't tell if poor health causes diabetes or the other way around. It was also trained on one specific dataset, so it might not work as well for different groups of people or in different countries. In short, our model is a helpful tool, but it should be used alongside doctor's expertise.

Section 6: Contribution Statement

We hereby declare the specific contributions of each group member to both the report and code:

Group Member 1 (spen0701) was primarily responsible for (35%):
Writing Section 3 (Modelling), Section 4 (Experimental Setup and Model Comparisons).
In the code, leading the effort on hyperparameter tuning and implementing the change from 3-fold to 5-fold cross-validation.
Integrating and organizing the final code for submission.

Group Member 2 (zhlu0474) was primarily responsible for (35%):
Writing Section 2 (Data Description).
Designing the modeling plan. Implementing and documenting the other two models (Decision Tree and Random Forest) in the code.

Group Member 3 (xhui0476) was primarily responsible for (30%):
Writing Section 1 (Topic Selection and Problem Definition), Section 5 (Conclusions

and Limitations), Section 6 (Contribution Statement).
Implementing and documenting the first model (Logistic Regression) in the code.