Predicting the Risk of Diabetes using Logistic Regression

1. Summary

We attempt to develop a logistic regression model to predict whether a patient has diabetes or not using the Diabetes Health Indicators dataset sourced from the UCI machine learning repository [@CDC2023]. We employ Random Over-Sampling Examples (ROSE) to balance the data and a tuned Least Absolute Shrinkage and Selection Operator (LASSO) regression model to classify patients who are at high risk of developing diabetes, using 5 out of the 21 risk factors provided in the dataset. Recall was used to measure the classifier's performance, as the consequences of false negatives would be more severe than false positives for this task. The area under the receiver operating characteristic (ROC) curve (AUC) was also chosen to evaluate the model's effectiveness in distinguishing between the two classes compared to random guessing.

2. Introduction

Diabetes is a chronic condition characterized by high blood sugar levels, resulting from excess buildup of glucose in the bloodstream [@Mayo2024]. Diabetes is linked to a variety of complications, including retinopathy, cardiovascular disease, stroke, and increased susceptibility to infections [@Papatheodorou2018]. Individuals with diabetes often experience significant reductions in their years of healthy life [@Ong2023]. In 2021, diabetes-related conditions resulted in over 2 million deaths worldwide and the prevalence of diabetes continues to rise, with its growth only accelerating in the 21st century [@WHO2024]. The true mortality rate may be higher than current estimates suggest, as many cases of diabetes go undiagnosed [@Stokes2017]. However, advancements in screening and detection methods have the potential to reduce the number of undiagnosed cases [@Fang2022]. As such, accurate diagnosis in the early stages is crucial since interventions can be administered to prevent the progression of diabetes [@Mayo2024]. This project aims to develop a classification model to predict diabetes status based on several health indicators. By using various data preprocessing strategies, we seek to improve the accuracy of diabetes detection using publicly available health data. Ultimately, the goal is to answer the following question: **Can we develop a classification model that can predict whether a person will have diabetes more accurately than random guessing?**

The dataset used in this project is sourced from the CDC Diabetes Health Indicators dataset on the UCI machine learning repository [@CDC2023], which contains various demographic and lifestyle-related features that may influence the likelihood of an individual developing diabetes. The dataset contains 23 features, of which 21 can be used as predictors in a classification model. The total 23 features are:

- `ID`: Patient identification number. This feature was removed from the publicly available dataset.
- `Diabetes_binary`: Diabetes/pre-diabetes (1) or no diabetes (0). This is the target feature.
- `HighBP`: High blood pressure (1) or not (0)
- `HighChol`: High cholesterol (1) or not (0)
- `CholCheck`: Cholesterol check in past 5 years (1) or no check (0)
- `BMI`: Body mass index
- `Smoker`: Have smoked at least 100 cigarettes in lifetime (1) or not (0)
- `Stroke`: Had a stroke in the past (1) or not (0)
- `HeartDiseaseorAttack`: Had coronary heart disease or myocardial infarction (1) or not (0)
- `PhysActivity`: Physical activity in the last 30 days (1) or not (0)
- `Fruits`: Consume fruit 1 or more times per day (1) or not (0)
- `Veggies`: Consume vegetables 1 or more times per day (1) or not (0)
- `HvyAlcoholConsump`: Having more than 14 drinks per day for adult men and 7 drinks for women, yes (1) or no (0)
- `AnyHealthcare`: Have any kind of health care coverage (1) or not (0)
- `NoDocbcCost`: Could not see a doctor in the past 12 months due to cost (1) or not (0)
- `GenHlth`: General health rating on scale of 1 - 5
  - 1 = Excellent
  - 2 = Very good
  - 3 = Good
  - 4 = Fair
  - 5 = Poor
- `MentHlth`: Number of days where mental health was not good in the last 30 days (1 - 30)
- `PhysHlth`: Number of days where physical health was not good in the last 30 days (1 - 30)
- `DiffWalk`: Serious difficulty walking or climbing stairs (1) or not (0)
- `Sex`: Male (1) or female (0)

- `Age`: Age based on 13-level scale (See codebook _AGEG5YR for more information)
  - 1 = Age 18-24
  - 9 = 60-64
  - 13 = 80 or older
- `Education`: Education level based on 6-level scale:
  - 1 = Never attended school/only kindergarten
  - 2 = Grades 1 through 8
  - 3 = Grades 9 through 11
  - 4 = Grade 12 or GED
  - 5 = College 1 year to 3 years
  - 6 = College 4 years or more
- `Income`: Income based on 8-level scale (See codebook INCOME2 for more information)
  - 1 = Less than $10,000
  - 5 = Less than $35,000
  - 8 = $75,000 or more

The primary objective is to classify individuals into diabetic/high risk of diabetes (`Diabetes_binary = 1`) or non-diabetic (`Diabetes_binary = 0`) categories using predictive modelling.

3. Method and Results

The project follows a structured approach to data exploration, preprocessing, feature selection and classification modeling.

Analysis workflow:

First, the dataset is obtained from an external source and loaded into R. Then, the raw dataset is inspected for completeness and correctness. This includes checking for missing and unique values in each feature. Categorical variables (e.g. age, smoking status, high blood pressure) are converted into factors to facilitate the analysis.

Moreover, the dataset is highly imbalanced, with more non-diabetic cases than diabetic ones. To address this, the Random Over-Sampling Examples (ROSE) technique is applied to generate synthetic data points to balance the dataset.

Visualizations (bar plots, density plots) are generated to explore the relationships between health indicators and diabetes status. Trends between factors such as `BMI` and `HighBP` with the target variable `Diabetes_binary` are examined. The dataset is split into 75% training data and 25% testing data to build and evaluate the LASSO regression model. According to @Sivakumar2024, the most commonly used train-test splits in the literature are 70:30 and

80:20. However, using too little or too much training data can lead to issues such as underfitting or overfitting. As such, we chose a 75:25 split as a reasonable compromise that has also demonstrated high accuracy in logistic regression models (@Sivakumar2024). The results of our analysis are visualised as an ROC curve and a confusion matrix, along with a LASSO feature coefficient plot.

3.1. Loading Data from Original Source on the Web

The packages used in this analysis are `tidyverse`, `tidymodels`, `glmnet`, `patchwork`, `ROSE`, `vcd`, and `FSelectorRcpp`. The dataset of interest can be acquired from the source by running `dataset_download.py` located in the `~/work/src/` directory. This script will fetch the raw dataset (UCI ID: 891) from the UC Irvine machine learning repository and then write the result into a .csv file (`cdc_diabetes_health_indicators.csv`) located in the `~/work/data/raw/` directory.

3.2. Exploratory Data Analysis (EDA)

Analysis Workflow:

All features in the dataset are checked for the following attributes:

- `NA_Count`: Number of "NA" values within each variable; if they exist, they would need to be replaced or removed.
- `Distinct_Count`: Number of possible values for each variable; primarily to check if the variables are numerical, categorical or binary (only 2 possible values).
- `Current_Data_Type`: Current data type for each variable; primarily to ensure that they are in the appropriate format we require, as the data type may be lost during `read_csv()`.

Table 1: Summary of Missing Values, Distinct Counts of each Variable, and Data Types in the Raw Diabetes Dataset

	NA_Count	Distinct_Count	Current_Data_Type
HighBP	0	2	double
HighChol	0	2	double
CholCheck	0	2	double
BMI	0	84	double
Smoker	0	2	double
Stroke	0	2	double
HeartDiseaseorAttack	0	2	double
PhysActivity	0	2	double
Fruits	0	2	double
Veggies	0	2	double

	NA_Count	Distinct_Count	Current_Data_Type
HvyAlcoholConsump	0	2	double
AnyHealthcare	0	2	double
NoDocbcCost	0	2	double
GenHlth	0	5	double
MentHlth	0	31	double
PhysHlth	0	31	double
DiffWalk	0	2	double
Sex	0	2	double
Age	0	13	double
Education	0	6	double
Income	0	8	double
Diabetes_binary	0	2	double

Given the initial check, the following observations can be made using Table 1:

- None of the columns have `NA` values.
- `BMI` is the only numerical variable, with the rest being categorical or binary. The `BMI` feature may cause issues with certain feature selection techniques. As such, we will look to bin this feature into discrete, categorical values later on.
- All variables are treated as `double` in the original dataset, and thus every variable except for `BMI` will need to be converted to the factor data type.

We now look to check for class imbalances in the dataset. It is important to have a balanced the dataset for our machine learning model to reduce bias towards the majority class, improve the model's ability to generalize to unseen data, and increase model training efficiency.

Table 2: Class Distribution of Diabetes_binary in the Raw Diabetes Dataset

Diabetes_binary	Count	Proportion
0	218334	0.860667
1	35346	0.139333

In the original `cdc_diabetes_health_indicators.csv` dataset, approximately 86% of individuals do not have diabetes, resulting in heavily imbalanced classes (Table 2).

3.3. Preprocessing: Wrangling, Cleaning, and Balancing Data from Original Format

Analysis Workflow:

From our initial EDA, we saw that there is a large class imbalance in the dataset. To address this, we use the `ROSE()` function to create a balanced version of the data by undersampling the majority class and oversampling the minority class. The size of the balanced dataset will be equal to that of the original `cdc_diabetes_health_indicators.csv`.

Table 3: Class Distribution of Diabetes_binary in the Balanced Dataset

Diabetes_binary	Count	Proportion
0	126884	0.5001734
1	126796	0.4998266

After balancing, the distribution of individuals in both categories of `Diabetes_binary` is roughly 50 - 50 (Table 3). A summary of the class distribution before and after ROSE can be seen in Table 4 below.

Table 4: Comparison of Class Distribution Before and After Balancing with ROSE

Diabetes_binary	Original_Count	Original_Proportion	Balanced_Count	Balanced_Proportion
0	218334	0.860667	126884	0.5001734
1	35346	0.139333	126796	0.4998266

The balanced dataset, `balanced_raw_diabetes_df`, will be saved in the `~/work/output/` directory and then split into training (`diabetes_train`) and testing (`diabetes_test`) sets for machine learning. These two datasets will be saved in the `~/work/data/processed` directory as .RDS files to maintain their data types.

From our EDA, we also noted that the `BMI` feature being numerical will be an issue in feature selection. Here, we decide to bin the `BMI` feature into discrete categories corresponding to the Centers for Disease Control and Prevention (CDC) BMI categories for adults (@CDC2024). Table 5 below shows the CDC's categorization (`BMI_Category`, `BMI_Range` in (kg/m^2)) and how the BMI feature will be binned in the dataset under `BinnedBMI` (@CDC2024).

Table 5: CDC BMI Categories, Ranges and Corresponding Binned BMI Values in the Dataset

BMI_Category	BMI_Range	BinnedBMI
Underweight	Less than 18.5	1
Healthy Weight	18.5 to less than 25	2
Overweight	25 to less than 30	3
Class 1 Obesity	30 to less than 35	4
Class 2 Obesity	35 to less than 40	5

BMI_Category	BMI_Range	BinnedBMI
Class 3 Obesity (Severe Obesity)	40 or greater	6

Table 6 shows the resulting counts and percentages of individuals in the dataset that correspond to each binned BMI value.

Table 6: Count of each Binned BMI Value in the Dataset

BinnedBMI	Count	Proportion
1	2712	0.0106906
2	59118	0.2330416
3	86826	0.3422658
4	57719	0.2275268
5	27118	0.1068985
6	20187	0.0795766

There appears to be a sizable portion of people who are categorized as obese (`BinnedBMI` = 4, 5, 6), with this group making up about 41% of the total. Only a slim majority of individuals have a BMI that is not considered obese (59%) (Table 6).

3.4. Feature Selection

Analysis Workflow:

The dataset contains 21 usable features along with the target (Table 1). However, using every feature in the dataset would not be optimal as having a large number of features in our model could lead to a risk of overfitting, difficulty in interpretation and increased computation time. Thus, we want to select only a subset of relevant predictors from the 21 total risk factors in the dataset.

Bar plots are created below to visually determine correlation between each categorical variable and the target variable `Diabetes_binary`.

At a glance, it appears that `HighBP`, `HighChol`, `CholCheck`, `Stroke`, `HeartDiseaseorAttack`, `HvyAlcoholConsump`, `DiffWalk`, `Age`, `Education`, `Income`, `GenHlth`, and `BinnedBMI` provide the most obvious differences in distribution between their categories and the target (Figure 1). There are 14 features with binary categories and 7 features with multiple categories (Figure 1).

To further narrow down relevant predictors, we conducted independent chi-squared tests to determine whether a significant relationship exists between each categorical variable and the target variable `Diabetes_binary`.

Figure 1: Distribution of Diabetes Binary by Various Variables

Table 7: Chi-squared Statistic, Degrees of Freedom, p-value, and Cramér's V sorted in Descending Order

Variable	Statistic	DF	p_value	Expected_Min	Expected_Max	CramersV
GenHlth	32595.1837	4	0	7775.801687	31460.91	0.4139072
HighBP	27182.1472	1	0	41369.644150	53764.64	0.3779900
BinnedBMI	18483.1383	5	0	1015.147729	32671.83	0.3116837
Age	16201.8203	12	0	1375.522674	14420.00	0.2918154
HighChol	15433.4322	1	0	44893.421350	50238.42	0.2848220
DiffWalk	14177.0819	1	0	24191.105345	70955.11	0.2729851
Income	10471.3545	7	0	4803.833002	27664.59	0.2345998
PhysHlth	10070.4049	30	0	8.497051	53746.14	0.2300646
HeartDiseaseorAttack	8308.6125	1	0	14101.606544	81051.61	0.2089879
Education	6012.3511	5	0	77.473116	35186.20	0.1777659
PhysActivity	4667.1676	1	0	28074.257837	67069.26	0.1566336
Stroke	3088.2364	1	0	6067.394528	89091.39	0.1274251
CholCheck	2406.0841	1	0	2310.698155	92850.70	0.1124899
MentHlth	2037.9585	30	0	8.996878	64606.40	0.1034961
HvyAlcoholConsump	1615.0859	1	0	3932.635320	91227.64	0.0921613
Smoker	1455.6010	1	0	45045.868448	50085.87	0.0874782
Veggies	1144.1008	1	0	20112.020845	75037.02	0.0775587
Fruits	603.4556	1	0	36554.315137	58583.32	0.0563290
Sex	363.5604	1	0	43475.913245	51656.91	0.0437239
NoDocbcCost	346.9351	1	0	8839.932419	86316.93	0.0427203
AnyHealthcare	110.5003	1	0	4293.510092	90866.51	0.0241248

All chi-squared tests yielded very small p-values (< 0.05), indicating significant relationships between each feature and the target (Table 5). However, there is a key limitation to this test: It only tests for statistical significance (i.e., whether an association exists) but does not indicate the strength of that association. In large datasets, chi-squared tests can produce extremely small p-values even for weak associations, making it difficult to determine their practical importance.

To address this, we employed Cramér's V which provides a standardized measure of the strength of association between each categorical variable and the target variable. Cramér's V quantifies how strong the association is on a scale from 0 (no association) to 1 (perfect association). To utilize Cramér's V, both the variables of interest and the target variable should be categorical. More than two unique values are also allowed for the categorical variables [@StatsTest2020].

We defined a Cramér's V value greater than 0.25 to be strongly associated with the target and sufficient to include the corresponding feature in our model. This value is supported in the literature; @Akoglu2018's correlation coefficient guide suggests that a value above 0.25 can be interpreted as a very strong relationship between the variables compared. Similarly, @Dai2021 considered Cramér's V values greater than 0.25 as reflecting a very strong association in their clinical study. Using this threshold, we selected the following features as candidates for inclusion in the model: GenHlth, HighBP, BinnedBMI, Age, HighChol, DiffWalk (Table 5).

One limit of using Cramér's V is that it can only capture linear relationships since it measures the strength of associations between independent categorical variables and the target. To address this, we must also consider non-linear correlations. One metric is Information Gain (IG) which evaluates how much uncertainty is reduced when a particular feature is included in the model. A higher IG value indicates a more informative feature that is useful for prediction. Research has demonstrated the effectiveness of using Information Gain for feature selection as a preprocessing step in various machine learning models, including logistic regression (@Sholeh2024). Table 8 below displays the IG value for each feature in the dataset.

Table 8: Information Gain (IG) Value for Each Feature

Variable	Information_Gain
GenHlth	0.0911453
HighBP	0.0733880
BinnedBMI	0.0501820
Age	0.0455696
HighChol	0.0411368
DiffWalk	0.0384096
Income	0.0279668
PhysHlth	0.0270617
HeartDiseaseorAttack	0.0227078
Education	0.0159997
PhysActivity	0.0123593
Stroke	0.0084526
CholCheck	0.0070007
MentHlth	0.0054180
HvyAlcoholConsump	0.0043927
Smoker	0.0038312
Veggies	0.0030167
Fruits	0.0015875
Sex	0.0009562
NoDocbcCost	0.0009150
AnyHealthcare	0.0002916

Domain knowledge suggests that an information gain value greater than 0.05 is generally considered to be valid for feature selection (@Sholeh2024). This threshold value has been shown to yield high model accuracy and optimal training time across multiple datasets (@Sholeh2024). From Table 8, we can see that GenHlth, HighBP, BinnedBMI have an information gain greater than 0.05. This makes them suitable candidates for inclusion in the model.

Using both Cramér's V and Information Gain analyses, we see overlap between features selected. This is summarized by Table 9 below:

Table 9: Summary of Features Selected using Cramér's V and Information Gain

Cramér.s_V_Variables	Information_Gain_Variables
GenHlth	GenHlth
HighBP	HighBP
BinnedBMI	BinnedBMI
Age	NA
HighChol	NA
DiffWalk	NA

As such, we will be using the following features for our logistic regression model: GenHlth, HighBP, BinnedBMI, Age, HighChol, DiffWalk (Table 9).

3.5. Classification Analysis

Analysis Workflow:

Given the large quantity of binary features and with the task being for classification, a logistic regression model `lr_mod` will be required. Specifically, we will use a Least Absolute Shrinkage and Selection Operator (LASSO) regression model. We chose to use a LASSO model due to its interpretability, efficiency, and reduced overfitting risk.

- `Interpretability`: Each predictor in a LASSO model has an associated coefficient which can be extracted from the model. These coefficients can tell us the direction and magnitude of a predictor's impact on the target variable.
- `Efficiency`: LASSO is effective in handling multi-dimensional data which is useful when we one-hot encode features that result in many feature levels. Additionally, LASSO has been shown to outperform tree-based models in clinical applications, such as predictive diagnoses of medical conditions (@Ping2022).
- `Reduced overfitting risk`: LASSO inherently applies a penalty (L1) to the regression coefficients which shrinks less important features towards zero. This regularization aspect of LASSO reduces the risk of overfitting, especially in high-dimensionality. It also handles multicollinearity better than traditional least squares regression.

In the model pipeline, a v-fold cross-validation split `vfold_cv()` is created in preparation for cross-validation of the dataset, to provide a more effective estimate of the model performance. `lr_recipe` is then created to apply one-hot encoding to the categorical features, and normalization to all predictor variables.

`lambda_grid` is created as a grid of hyperparameters for tuning with the cross-validation set with penalty terms, while `tune_grid()` is used for hyperparameter tuning, specifically tuned for higher recall, as the consequences of false negatives for this task would be more severe. The model that maximizes recall is selected for the finalized workflow `lasso_tuned_wflow`.

3.6. Result of Analysis - Visualization

Analysis Workflow:

The results of our LASSO regression predictions include `lasso_preds`, which provides the predictions for each row, `lasso_probs`, which provides the probability of each classification for each row, and `lasso_metrics`, which displays the following metrics in Table 10:

- `sens`: Sensitivity; true positive rate
- `spec`: Specificity; true negative rate
- `ppv`: Positive predictive value; precision
- `npv`: Negative predictive value
- `accuracy`: Accuracy for all predictions
- `recall`: Recall; true positive rate
- `f_meas`: F-measure; harmonic mean of precision and recall
- `roc_auc_value`: ROC AUC value; measure of the model's effectiveness in distinguishing between classes.

Table 10: Classification Metrics for Lasso Model on Test Set

.metric	.estimator	.estimate
sens	binary	0.7652607
ppv	binary	0.7135545
npv	binary	0.7471112
accuracy	binary	0.7291233
recall	binary	0.7652607
f_meas	binary	0.7385037
roc_auc	binary	0.8013900

The ROC curve and confusion matrix are visualised below (Figure 2, Figure 3).

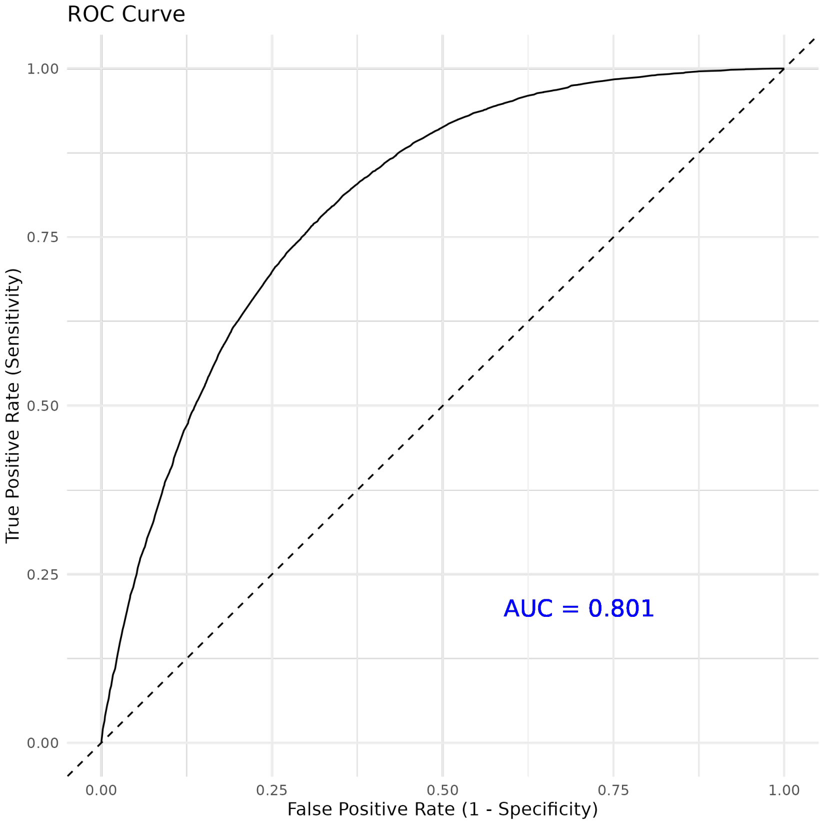

Figure 2: ROC Curve for Lasso Model

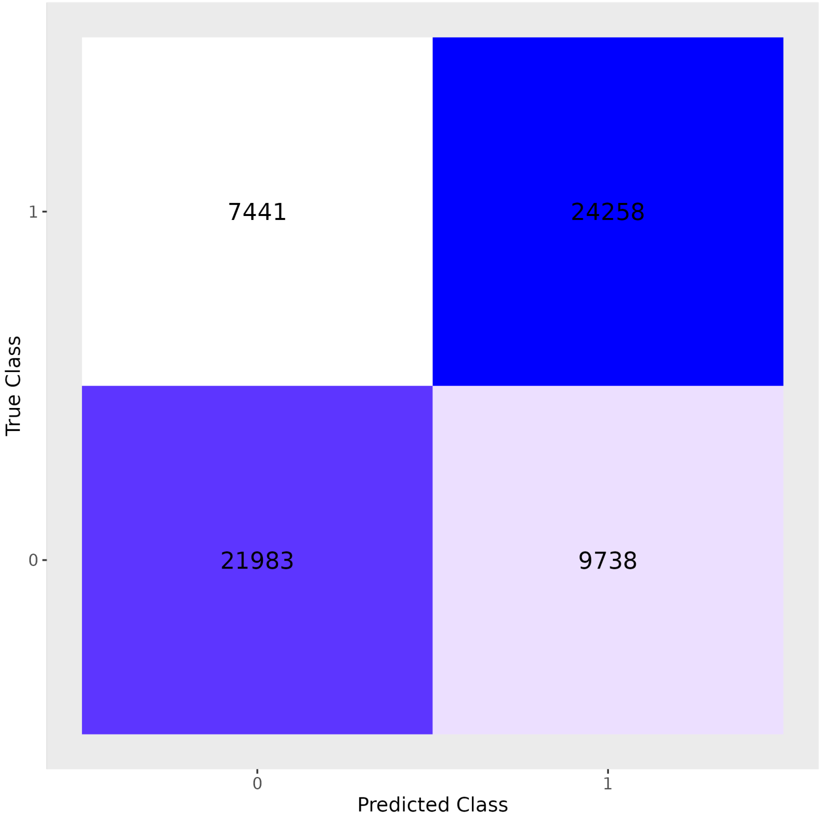

Figure 3: Confusion Matrix for Lasso Model

The confusion matrix displays the quantity of each type of prediction result; The model predicts 24258 true positive, 21983 true negative, 7441 false negative and 9738 false positive cases (Figure 3).

Thus, the recall for the model can be calculated as $\frac{\text{True Positives}}{\text{True Positives+False Negatives}} = 0.7652607$ (7 s.f.). This is the same result as shown in Table 10.

The false negative rate can be calculated as: $\frac{\text{False Negatives}}{\text{True Positives+False Negatives}} = 0.2347393$ (7 s.f.). This is the same result as calculating (1 - `sens`) from Table 10.

From Table 10, the values of particular interest are the `recall` score of 0.7652607 (7 s.f.) and the `roc_auc_value` of 0.80139 (7 s.f.).

Looking into the fitted LASSO model, we can extract the coefficients to see which features are contributing at what magnitudes to the final prediction (Fig 3).

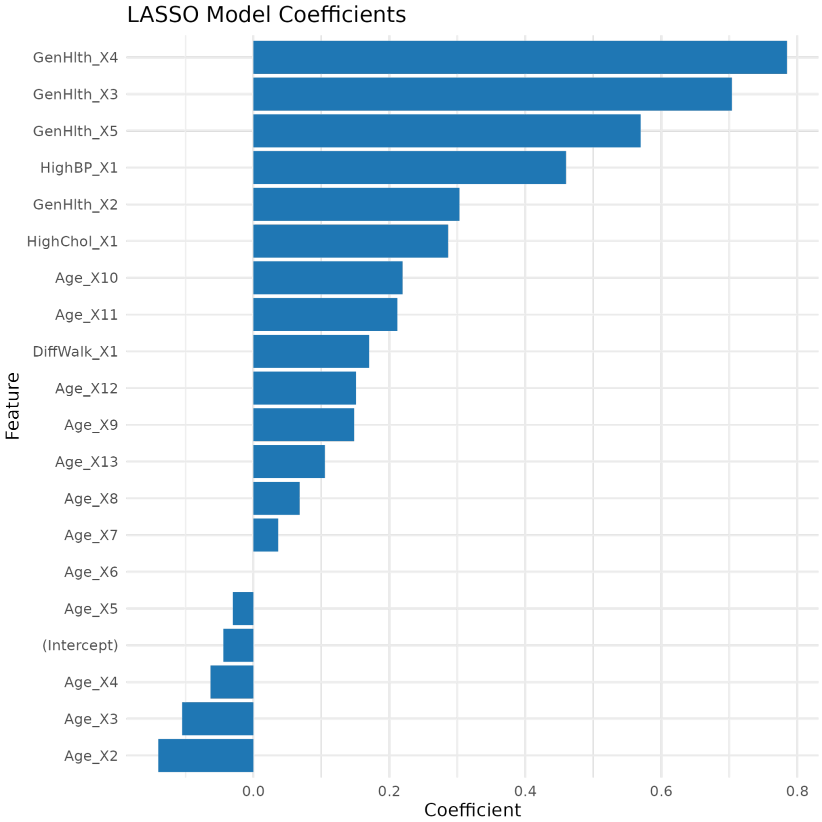

Figure 4: Coefficient Plot for Lasso Model

From Fig. 3, we can see that multiple levels of `GenHlth` along with `HighBP` and `HighChol` are the main features with positive contributions to the LASSO model. On the flip side, lower age levels are the main features with negative coefficients. For the positive features:

- GenHlth: Individuals who classify themselves as 3 - Good, 4 - Fair, and 5 - Poor seem to predict an increase in the target variable (Diabetes_binary)
- HighBP: Individuals with higher than average blood pressure seems to predict an increase in the target variable
- HighChol: Individuals with higher cholesterol levels than average seems to predict an increase in the target variable

For the negative feature: - Age: Individuals who fall under 5 (corresponding to ages 25 to 44) seem to predict a decrease in the target variable

From the LASSO coefficients, we can see that vital signs such as high blood pressure and cholesterol along with overall feelings of health are the strongest predictors for diabetes. Additionally, individuals who are younger appear to be inversely correlated with diabetes prevalence. However, as age increases, the corresponding coefficients increase which suggests that being older is a predictor for diabetes.

4. Discussion

Our model achieved a recall score of 0.7652607 (7 s.f.) on the test set, implying that about 77% of all positive instances of diabetes were correctly classified by the LASSO regression model (Table 6). This suggests that the model is relatively effective at identifying individuals who are at risk of developing diabetes. Additionally, the model achieved an area under the ROC curve of 0.80139 (7 s.f.) on the test set (Table 6). Since the AUC is above 0.5, this indicates that the model can discriminate between diabetic and non-diabetic cases better than random guessing (Figure 3).

We expected the model to perform better than random guessing, which it did achieve. Additionally, we aimed to minimize false negatives which is particularly important in healthcare diagnoses where a false negative case can have serious consequences. For example, a false negative would indicate that the model predicts the patient to not develop diabetes even though they do. This may lead to the patient not getting the treatment or care they need, potentially resulting in health complications and even death. The model had a false negative rate around 23% which is concerning as it indicates a significant risk for missing positive cases leading to unfavourable patient outcomes.

Our model was optimized for recall through hyperparameter tuning, with cross-validation used to evaluate model performance during the process. However, despite our results, the model falls short of what is expected in clinical applications. For example, more complex models in the literature which incorporate advanced feature selection techniques [@Alhussan2023] or Generative Adversarial Networks (GANs) [@Feng2023] can achieve recall and AUC scores upwards of 97%. Our findings serve as a proof of concept for the feasibility of classification models to predict the risk of diabetes based on publicly available health data. Since our model did relatively well compared to random guessing, this indicates potential correlations between health indicators and the likelihood of developing diabetes. With this information, people might be more aware of their health and lifestyle choices. They may be inclined to work harder to reduce cholesterol levels, manage high blood pressure, keep alcohol consumption under control and maintain a healthy lifestyle. Thus, this may help decrease the global mortality rate from diabetes through early interventions and lifestyle changes.

Future directions could include how we can improve our classification method to more accurately predict the risk of diabetes in patients. We can explore more rigorous feature selection techniques or implement other machine learning models such as boosted trees to improve our classification performance. In the context of the healthcare system, we can look to integrate classification models into the diagnosis process to help detect diabetes early to improve patient outcomes.

5. References