Predicting occurrence of diabetes with behavior risk factor survey data from CDC

Link to the dataset: https://archive.ics.uci.edu/dataset/891/cdc+diabetes+health+indicators

Sigma Dynamics
Team lead: Shan Jiang
Analyst: Rui Ma
Analyst: Cassandra Morgan
Analyst: Gabriel Pascual

Table of Contents

Abstract/Executive Summary

The Centers for Disease Control and Prevention (CDC) aims to identify any relationships between diabetes and an individual's lifestyle. This agency has tasked our team, the Sigma Dynamics, to develop a machine learning (ML) model to predict a patient's diabetes risk classification (diabetes and healthy) by using the collected healthcare/demographic statistics along with the health-related survey data from The Behavioral Risk Factor Surveillance System (BRFSS). This dataset consists of a total of 1,014,720 demographic observations, 1,268,400 observations for lifestyle behaviors, health status, and disease history, and 761,040 healthcare access observations.

For the first research question: which machine learning model achieves the highest performance, we tested and developed three separate ML models (logistic regression, random forest, and neural network) on the CDC survey data. To find which ML model is the best to predict a patient's diabetes risk classification, we calculated the following performance measures of the three models: accuracy, sensitivity, specificity, F1-score and area under the curve (AUC) derived from receiver operating characteristic (ROC) curve. Among these performance measures, we prioritize sensitivity because this is a medical diagnosis ML model and we want the rate of misclassification of true diabetic patients as low as possible. AUC is a secondary prioritized measure because it can show the overall performance and identify the model with the best trade-off. Among the three models, the neural network outperformed the other two models with a close to 90% sensitivity (neural network: 88%; Logistic regression: 77%; random forest: 78%) and higher AUC (neural network: 0.83; Logistic regression: 0.82; random forest: 0.82).

For the second research question: what are the key features or factors associated with diabetes that differ between male and female patients, we developed the three aforementioned ML models on the male and female subsets, separately. Not surprisingly, the neural network still outperformed the other two models in terms of sensitivity in both male (neural network: 89%; Logistic regression: 76%; random forest: 76%) and female (neural network: 87%; Logistic regression: 77%; random forest: 79%) subsets, and it also slightly outperformed in terms of AUC. To assess which features contribute most to diabetes classification in males and females from the three models, we used permutation importance evaluated by the changes in AUC derived from ROC. This method reveals the drop in model performance when a feature's values are randomly shuffled, indicating how much the model relies on that feature. By calculating the permutation-based importance measures for the features in both male and female subsets for each model, and subtracting the importance measure of females from males, we were able to obtain the importance difference index between males and females (*importance difference index = importance measure in males - importance measure in females*). This index reflects the differed associations of the features with diabetes between males and females. A positive index indicates the feature is more important for males, whereas a negative index indicates the feature is more important for females. We observed that age is strongly associated with diabetes in males but not in females (importance difference index: 0.0336 for Logistic regression, 0.0021 for random forest and 0.0362 for neural network), whereas body mass index (BMI) is strongly associated with diabetes in females but not in males (importance difference index: -0.0078 for Logistic regression, -0.0077 for random forest and -0.0134 for neural network). In addition, different diseases were also revealed differed associations: coronary heart disease is more associated to males, whereas high blood pressure and high cholesterol are more associated to

females. These findings support the need for sex-specific risk models, as feature importance varies between males and females.

Project Plan

CDC description

The Centers for Disease Control and Prevention (CDC) serves as the national public health agency for the United States. This organization's mission is to protect the overall public health of America.

Since the CDC is a public health organization, most of its information is publicly available online. One of its purposes is to provide health guidelines and disease statistics to educate those who are susceptible to any health threats prevalent in the U.S. or abroad. Additionally, the CDC conducts research on disease threats on a global scale to develop strategies and treatments to counteract these outbreaks. This public health organization conducts ongoing research to create tactics for effective disease control/prevention, improve health treatments, identify risk factors for new or existing diseases, and promotes healthy habits for individuals with existing health conditions. Furthermore, the CDC plays a critical role in the training and preparation of public health workers and leaders through their career/training programs.

CDC Overview:

Organization - Centers for Disease Control and Prevention (CDC)

Budget - $9.683 billion (FY 2025)

Total Employee Count - 11,814

Global Employee Count - over 1,700 health professionals all over 60 countries, 1,300 local staff, and 400 staff from the U.S.

Key Leaders:

Susan Monarez, PhD - Acting Director, First Assistant to the Director, Principal Deputy Director

Debra Houry, MD MPH - Deputy Director for Program and Science/Chief Medical Officer

Nina Witkofsky - Deputy Director of Public Affairs/Acting Director of Communications

Sara Patterson - Office of the Chief Operating Officer (OCOO)

Matthew Buzzelli - Office of the Chief of Staff

Top Competitors:

National Health Service (NHS)

Mayo Clinic

World Health Organization (WHO)

MedlinePlus

NHS inform

Analysis opportunity

The Centers for Disease Control and Prevention (CDC) aims to identify any relationships between diabetes and an individual's lifestyle. This agency has tasked our team, Sigma Dynamics, to develop a machine learning model to predict a patient's diabetes risk classification (diabetes, pre-diabetes, healthy) by using the collected healthcare/demographic statistics along with the health-related survey data from the Behavioral Risk Factor Surveillance System (BRFSS). This dataset consists of a total of 1,014,720 demographic observations, 1,268,400 observations for lifestyle behaviors, health status, and disease history, and 761,040 healthcare access observations. Our efforts will contribute to the long-term goal of preventing diabetes in patients through early detection based on an individual's lifestyle factors.

Research questions

This project aims to support the CDC's efforts in understanding and predicting diabetes prevalence by leveraging machine learning techniques applied to their survey data. To guide our investigation and model development, we focus on the following two research questions:

RQ1: Which machine learning model achieves the highest performance, particularly in sensitivity, for predicting diabetes using the CDC dataset?

Given the public health implications of missed diagnoses, our primary goal is to identify a machine learning model that maximizes **sensitivity (recall)**—the ability to correctly identify individuals with diabetes. We will benchmark several supervised learning algorithms, including logistic regression, random forests, and neural networks. Model performance will be evaluated across multiple metrics, with a focus on sensitivity to ensure the model is effective in flagging potential cases of diabetes for further screening or intervention.

RQ2: What are the key features or factors associated with diabetes that differ between male and female patients in the CDC dataset?

We aim to explore **feature importance and potential sex-based differences** in the risk factors associated with diabetes. We will identify which features most strongly contribute to diabetes predictions and assess how these features differ between male and female patients. This analysis will provide insight into potential disparities and inform targeted prevention or outreach strategies.

Hypothesis

H1. The best machine learning model is robust even if we subdivide the dataset into male and female-specific subsets.

We will compare the performances of different machine learning models on all samples, male only samples, and female only samples respectively. We expect the best model will outperform other models with either all samples set or sex-specific sample subsets. We will test this hypothesis in terms of different performance measuring metrics, especially sensitivity.

H2. Different features/factors contribute to the occurrence of diabetes in male and female populations.

There has been scientific research showing different causes and clinical manifestations between male and female diabetic patients (https://pmc.ncbi.nlm.nih.gov/articles/PMC10163139/). In this capstone project, we expect to observe features/factors contributing to the occurrence of diabetes in male and female populations differently.

Data

The Behavioral Risk Factor Surveillance System (BRFSS) is a health-related telephone survey that is collected annually by the CDC. Each year, the survey collects responses from over 400,000 Americans on health-related risk behaviors, chronic health conditions, and the use of preventative services. It has been conducted every year since 1984. For this project, the data collected for the year 2015 was used. This dataset contains three different versions depending on the sample sizes and the definition of predicting variable diabetes. For this project, the version with the largest sample size and a clear binary definition of diabetes will be used (diabetes_binary_health_indicators_BRFSS2015.csv). It has a total of 21 features, which can be broken down into the following categories.

Demographic attributes

This dataset contains demographic characteristic features for individual samples, including sex, age, education, income and body mass index (BMI).

Life habit

The following life habit features were included: smoking, consume fruits, consume vegetables, heavy drinking and physical activity.

General health

The following health-related features were included: general health, mental health, physical health, and difficulty walking.

Disease history

The following disease history related features were included: high blood pressure, high cholesterol, stroke, heart disease or attack.

Health coverage

The following health coverage related features were included: health insurance and not seeing a doctor because of cost.

Measurements

It is important to consider what is being measured as well as what influential factors are present in our analysis. Some of the measurements derived from the collected CDC data include demographic features, lifestyle behaviors, health status, disease history, and healthcare access. The demographic features such as sex, age, income, and body mass index (BMI) correlate with disease presence and risk. Lifestyle behaviors such as smoking, heavy drinking, physical activity, and the daily consumption of fruits/vegetables all influence the risk of diabetes. Health status including general health, mental health, physical health, and difficulty walking captures an

individual's overall well-being which could be affected by diabetes. Disease history including pre-existing conditions could increase the risk of developing diabetes. Lastly, healthcare access indicates whether an individual has health insurance and if they may have avoided seeing a doctor due to the cost of medical care, which could leave an individual more prone to developing diabetes.

The Centers for Disease Control and Prevention (CDC) has provided a dataset using both demographic/healthcare measurements and survey data from the Behavioral Risk Factor Surveillance System (BRFSS) to analyze key factors and gain valuable insights. By using the given indicators/measurements gathered from multiple sources, this data allows for a deeper analysis to capture the relationship between an individual's lifestyle and diabetes.

Methodology

In exploratory data analysis, we will perform a pairwise Pearson correlation analysis between numerical features and calculate pairwise Jaccard similarity coefficients between categorical features to find possibly redundant features. For example, whether fruit lovers are more likely to be vegetable lovers? If the two features are highly correlated (i.e. Jaccard index > 0.9), only one feature will be retained for follow-up analyses to reduce calculation burden, and the retained feature will be used as an agent for the removed feature.

For research question 1, we will compare multiple machine learning models, including logistic regression, random forest, and neural network, to evaluate their performances based on all sample super-set, the male subset, and the female subset. Due to a higher percentage of features being categorical in the dataset, to further reduce the computation burden, we will apply latent class analysis to find latent features to represent categorical features for the samples. We will

evaluate based on accuracy, sensitivity, specificity, and F1-score. Of these performance measuring metrics, we especially care about sensitivity because we don't want our machine learning model to miss any patients to delay their treatments.

For research question 2, we will determine the most important features/factors for either the male or female subset. After building a model, we will extract the coefficients or indices that can quantify the importance of these features. For example, log-odds coefficients from a logistic regression or Gini importance indices, which measure the mean decrease in impurity from a random forest. In addition, we will also measure the rank changes of the features between males and females. A higher rank change index indicates the feature plays a more different role between males and females.

Computational Methods and Outputs

We will use logistic regression, random forest, and neural network to answer RQ1: Which machine learning model achieves the highest performance, particularly in sensitivity, for predicting diabetes using the CDC dataset?

First, we must choose the proper model performance measure. The relevant choices are precision score, recall score, and F1. Recall score measures the model's ability to find all relevant instances of a class in a data set. Precision score measures what proportion of the risk factors identified were actually relevant. Since Recall and Precision Scores are complementary, they may be balanced by the use of the F1 Score. Regarding our first question: Which machine learning model achieves the highest performance, particularly in sensitivity, for predicting diabetes using the CDC dataset? We will be identifying individuals for diabetes screening. Failing to identify an individual with diabetes could delay treatment which is a much worse

outcome than the cost of a false positive: making an extra trip to the doctor for further examination. So for the first question, false negatives would be more critical and we should choose Recall as our metric for comparing models. For our second question: What are the key features or factors associated with diabetes that differ between male and female patients in the CDC dataset? We should use the F1 Score to balance Precision and Recall since the risk factors should be both identified and identified correctly.

Other than the performance metric, both research questions will follow similar procedures. For our Logistic model, we will preprocess the data by scaling and treating outliers and multicollinearity. Our Random Forest and Neural Network models will use unprocessed data. For the Random Forest and Neural network, tuning is critical. For Random Forest, we will use Random Search Cross Validation using a Random Hyperparameter Grid. For our Neural Network models, we will tune the most impactful hyperparameters: number of hidden layers, number of neurons, and the learning rate.

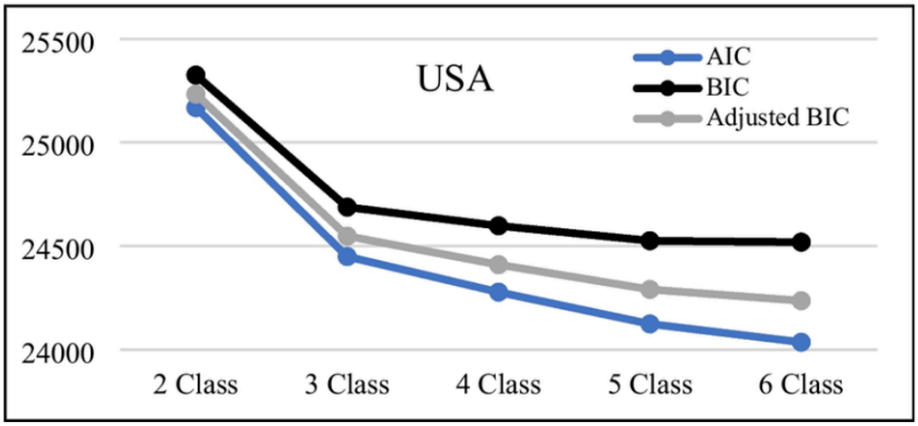
If the performance on Random Forest and Neural Network models are still underperforming, we will look at class imbalance for our first research question. According to the CDC, 11% of the US population has diabetes (which means that 89% don't). Since we have a mix of numerical and categorical data, we can use SMOTE-NC, which stands for Synthetic Minority Oversampling Technique. NC stands for numerical and categorical. This technique is an algorithm to generate new synthetic data of the minority class.

By these steps, we will have our best models and our best measure of which is best, as well as our best answer for the features unique to each gender.

Output Summary

For our exploratory data analysis, we will output a heatmap showing the Pearson Correlation such as this:

When we are choosing the number of classes for Latent Class analysis, we will output an elbow plot like this:



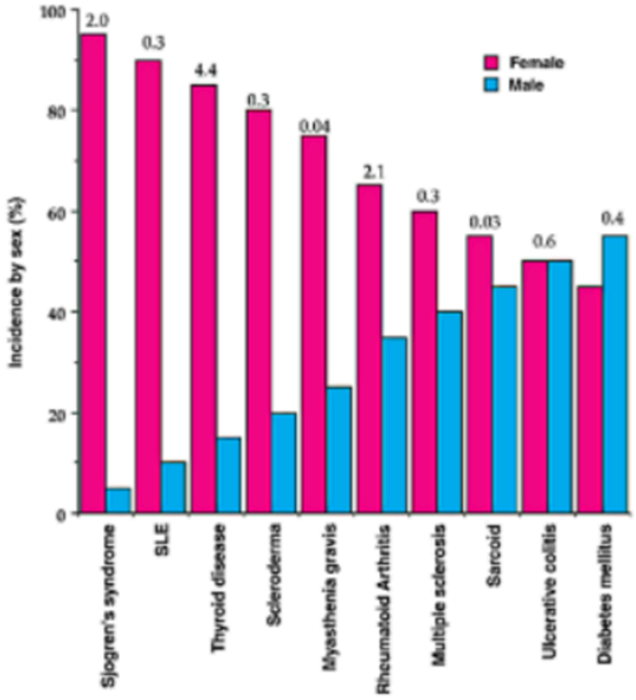
Elbow Plots from latent class analysis (LCA): Germany, Korea, and USA

To compare models to answer question 1, we will output a table such as:

ML Method	Recall (Train)	Recall (Test)
Logistic Regression	0.78	0.74
Decision Tree	0.98	0.65
Random Forest	0.92	0.80
Support Vector Machine	0.76	0.72
K-Nearest Neighbors	0.85	0.70
Gradient Boosting	0.90	0.82
XGBoost	0.91	0.84
Neural Network	0.95	0.78

We can use the built-in “feature importance” method in our random forest to give us a ranked list of the features. We could run it on the male patients and on the female patients, the plot the importance of the features in a double bar chart like the one below:

Figure 1: The sex distribution of the major autoimmune diseases.



Campaign Implementation

Diabetes is a chronic killer. According to the Orlando Clinical Research Center, diabetes "kills more people every year than breast cancer and AIDS combined. Complications from diabetes can vary. However, the most prevalent comorbid conditions include kidney disease, amputations, blindness, cardiovascular disease, obesity, hypertension, hypoglycemia, dyslipidemia, and risk of heart attack or stroke."

Each of our research questions could benefit researchers investigating diabetes. The answer to question one, "Which machine learning model achieves the highest performance, particularly in sensitivity, for predicting diabetes using the CDC dataset?" would give researchers ideas about

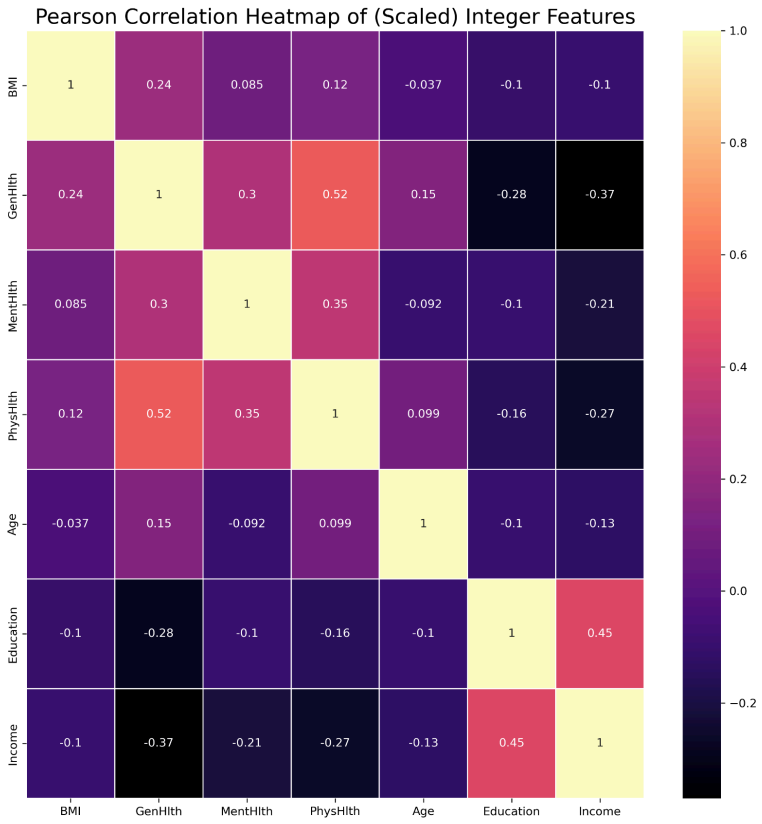
which model would be best for their case use. The answer to question two, “What are the key features or factors associated with diabetes that differ between male and female patients in the CDC dataset?” could suggest avenues for new research focusing on gender specific: symptoms, progression of the disease and prognosis. Patients and diagnosticians could also benefit from the answer to question 2, by showing which factors are more likely to be present in a woman with diabetes versus a man with it.

Exploratory Data Analysis

The version of the data we used for the project from the diabetes dataset from the CDC is the one with the largest sample size, and target variable is binary (healthy or diabetes). We performed exploration with respect to four aspects (subheadings).

1. Correlations Between the Features

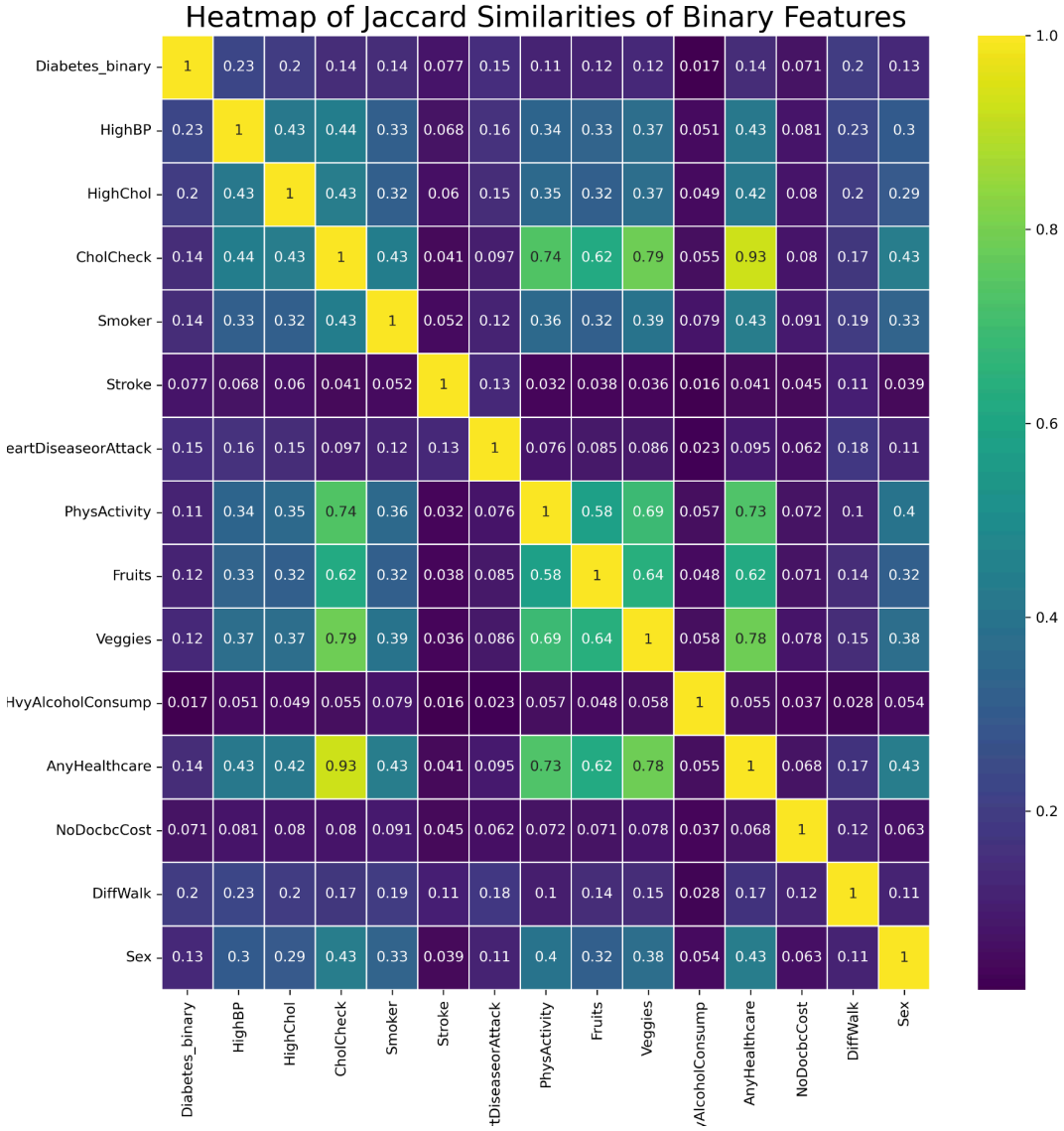
We explore the correlations between different features in order to identify highly related features. For example, whether fruit lovers are also very likely to be vegetable lovers? If the features are highly correlated, it is not necessary to include both of them, and one feature can be used as a surrogacy to the other. This reduction can avoid redundancy and also lessen computation burden.



Above is a heatmap of the Pearson correlations for our numerical data. Our data falls into two distinct categories: binary data and numerical data. We calculated the Pearson correlations for the numerical data. If two features are highly correlated, then there is a danger of multicollinearity causing issues in our models later on. However, according to our exploration of the data using the Pearson correlations, there are no features that are significantly highly correlated. This implies there is not a relatively large danger of multicollinearity should we use them in our models.

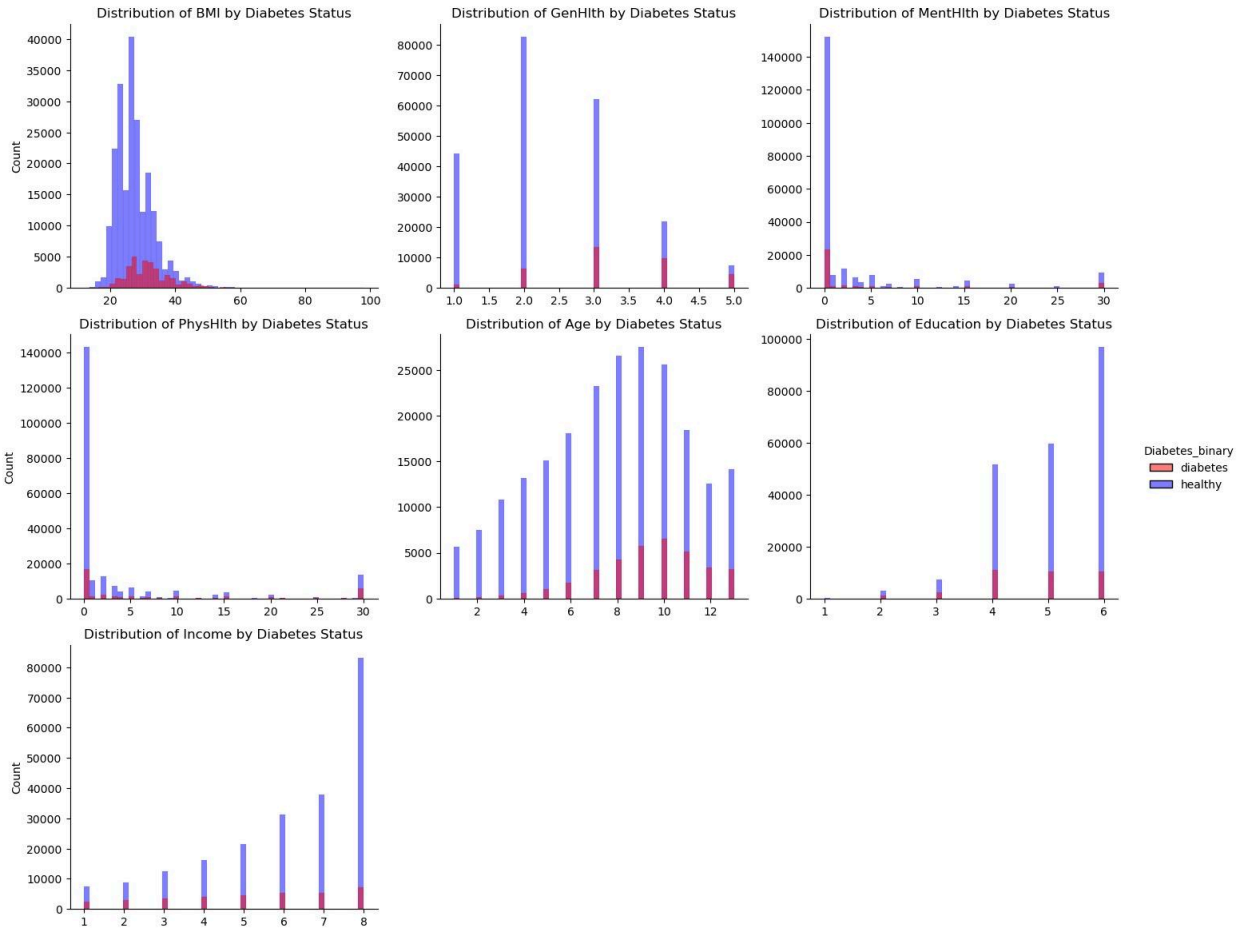
Because our data falls into two distinct categories, binary data and numerical data, we chose to use Jaccard similarity on the numerical data. The data for the binary features came from a phone survey performed by the CDC. The survey was called the Behavioral Risk Factor Assessment Surveillance System.

According to the data exploration of these binary features, the only two features with a Jaccard similarity greater than .9 were cholesterol check and any healthcare. Due to the fact that we believe they are truly measuring different things, we will keep those two features, despite the fact that they have a Jaccard similarity above the .9 threshold. None of the other binary features exhibited a Jaccard similarity greater than our .9 threshold, so they will also remain in the models. From this analysis, we are confident that the binary features we analyzed will not be too similar as to cause problems in our models later. We have a heatmap of the Jaccard similarity matrix calculated from our binary features below.



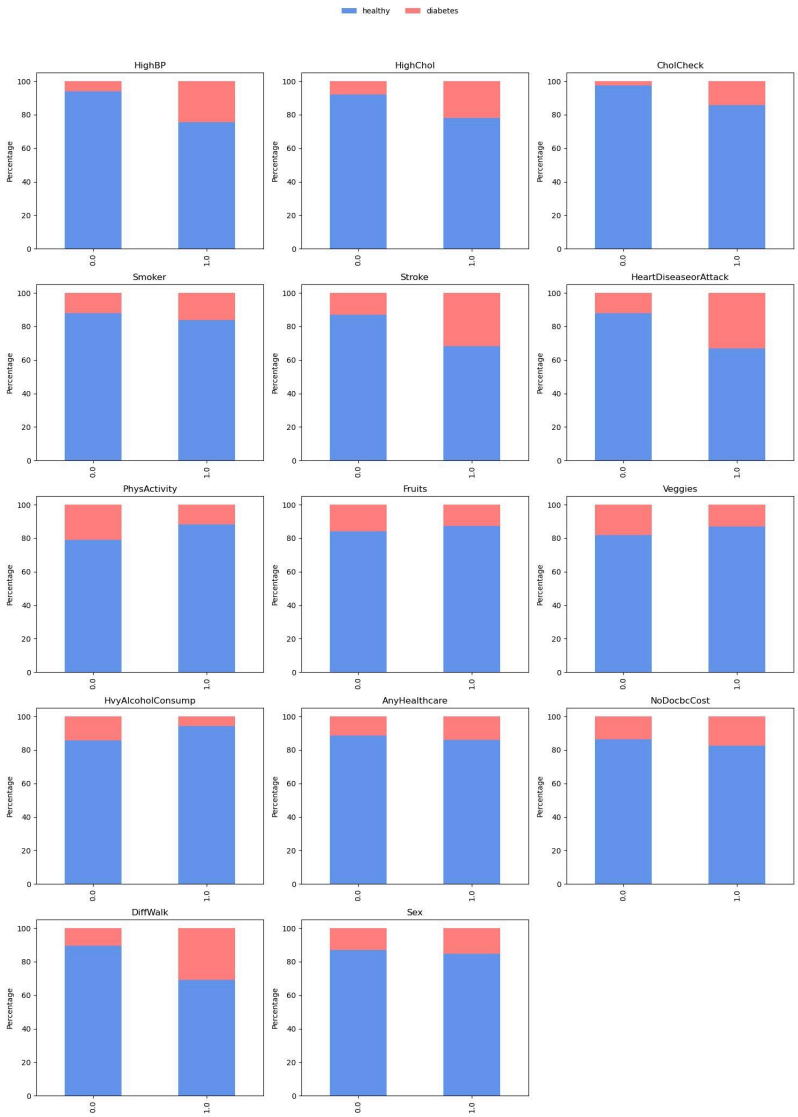
2. Feature differences between diabetic patients and healthy controls

We explore the feature differences between diabetic patients and healthy controls in order to identify features which may be good indicators of diabetes in machine learning models. For example, if the distributions of a numerical feature are different between diabetic patients and healthy controls, or the positive percentages of a binary feature are different between diabetic patients and healthy controls.



The figure above shows various distributions of each numerical feature in the CDC dataset, by diabetes status. The distribution of BMI for healthy individuals appears to have a general mean at 25 compared to those with diabetes with a slightly greater average which lies around 30. This indicates a greater average BMI for people with diabetes. Another observation is that the distributions for general health for diabetes patients mainly lie on the good to poor general health rating scale (3 to 5), whereas healthy patients lie on the excellent to good scale (1 to 3). Both the mental health and physical health distributions for diabetes/healthy patients appear to behave in a similar manner in that most of the data for both features fall at 0. This suggests that both groups, diabetes or healthy, report more individuals with a very minimal amount of poor mental health days or physical injury/illness days. The distribution of age for the diabetic group is shifted more to the right compared to the healthy group which means that the average age is higher among

individuals with diabetes. The majority of individuals in the diabetes group appear to have a piecewise distribution which remains constant from the 4 to 6 range. On the other hand, the healthy group has a strong right-skewed distribution with a gradual increase. This implies that the education level is lower for diabetics compared to healthy individuals. Finally, the distribution of income for diabetics is uniform while the distribution for healthy patients is skewed to the right. This shows that healthy individuals have a higher income scale than the diabetic patients.

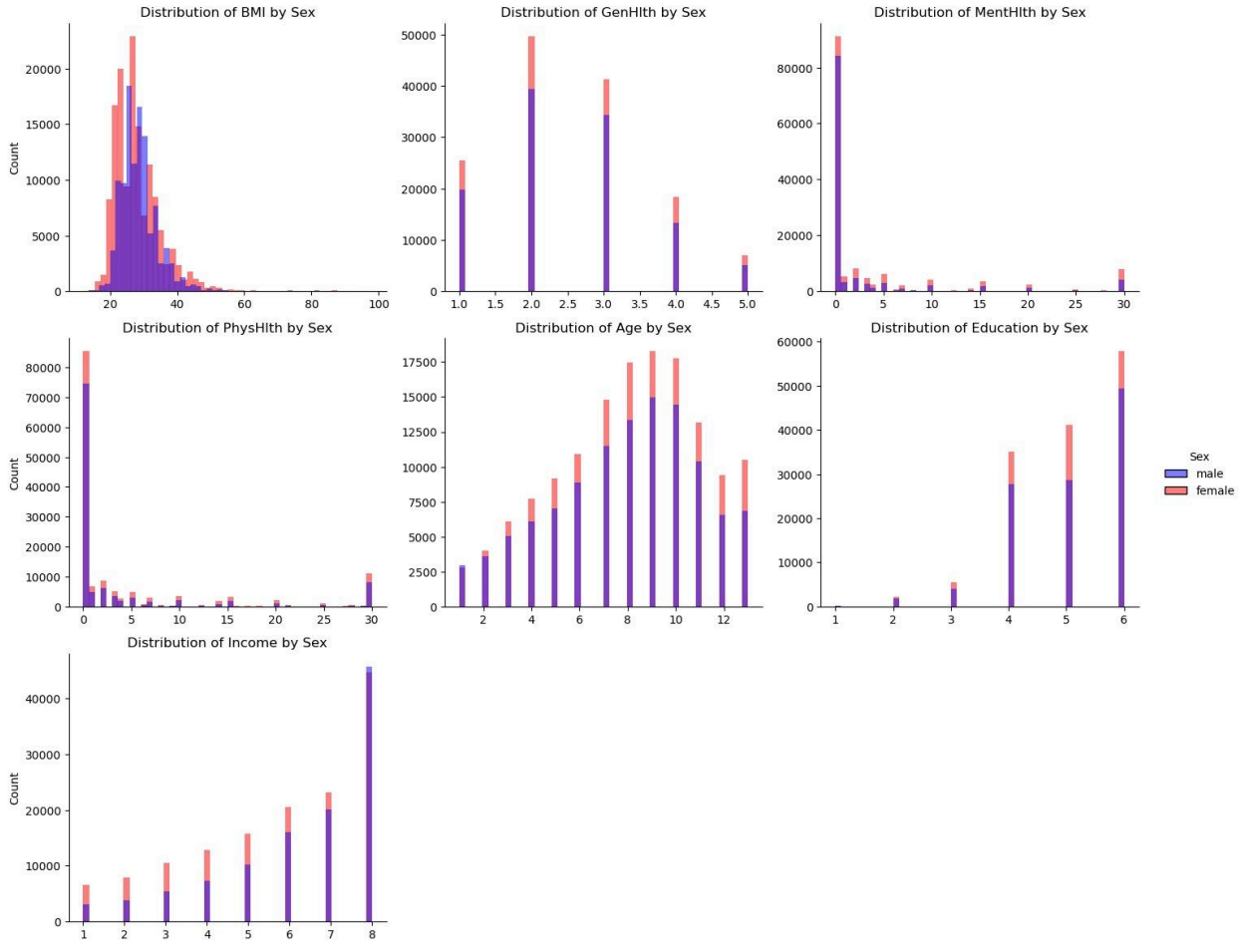


This figure above illustrates the percent differences of binary variables between diabetes status (healthy/diabetes). The HighBP (high blood pressure) variable appears to have a significant difference with around 15% of diabetics reporting no high blood pressure as opposed to the 25% that report high blood pressure. Even though the sizes of each distribution differs due to a greater number of healthy individuals, it is evident that diabetics are more likely to report having high blood pressure than not. The HighChol (high cholesterol) variable demonstrates a similar trend where more diabetics have reported having high cholesterol than not. The CholCheck (cholesterol check) variable distribution appears to have more diabetics who have had a cholesterol check in five years (15%) compared to diabetics that haven't (10%). The smoker variable seems to have more diabetics who reported smoking at least 100 cigarettes in their life (15%) compared to diabetics that haven't (10%). The stroke variable distribution displays 30% of diabetics who have had a stroke versus 15% that have not had one. The HeartDiseaseorAttack binary feature distributions report more diabetics (around 35%) claiming to have had coronary heart disease or myocardial infarction while fewer diabetics (15%) have not. The percent distribution for the PhysActivity (physical activity) shows that around 20% of diabetics had no physical activity in the past 30 days while a mere 10% have. This suggests that more diabetics perform less physical activity. For the fruit and veggie consumption distributions, more diabetics (15%) have reported not consuming fruits/vegetables one or more times in a day compared to the other diabetic individuals who do (10%). The HvyAlcoholConsump (heavy alcohol consumption) attribute is observed to have a greater percentage of diabetics (15%) who have not had seven to fourteen drinks per week compared to those that do (5%). The AnyHealthcare variable appears to have slightly more diabetics (5%) that have any kind of health care coverage than those who do not. The NoDocbcCost feature distributions reveal that there is a slight

percentage increase (5%) in diabetics who have not seen a doctor in the past twelve months because of cost compared to diabetics that have not had this issue. The DiffWalk (difficulty walking) attribute reveals that a greater percentage of diabetics (30%) have difficulty walking compared to the diabetic individuals that don't have any difficulty (20%). Finally, the Sex variable shows a 5% increase in diabetics in males as opposed to females.

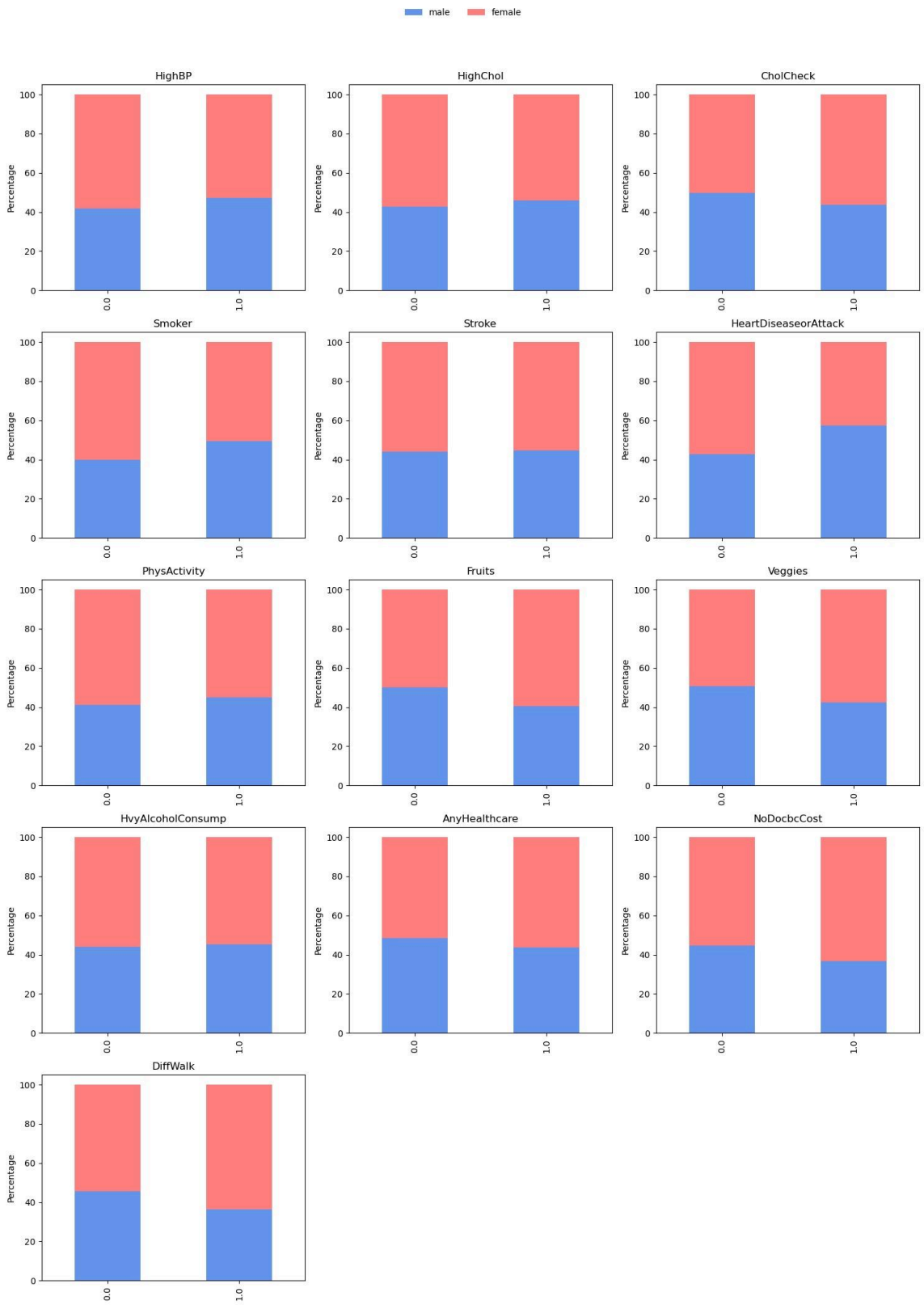
3. Feature Differences between Males and Females

We explore the feature differences between males and females in order to identify potential features which may contribute to the susceptibilities of diabetes in sex-specific manners.



The figure above shows the distributions of numerical variables in the CDC dataset, segmented by sex (male in blue, female in red). The BMI distribution for males and females is similar,

peaking around 25-30, but males have a slightly wider spread toward higher BMI values (up to 90), while females are more concentrated between 20 and 40. For general health (GenHlth), both sexes show a similar pattern, with most individuals rating their health between 1 (excellent) and 3 (good), though females have a slightly higher count in the 2-3 range. Mental health (MentHlth) and physical health (PhysHlth) distributions are also comparable between sexes, with most individuals reporting 0 days of poor mental or physical health. However, females report slightly more days in the 1-10 range for mental health. The age distribution shows males and females peaking around age categories 8-10 (55-69 years), but females have a slightly higher count in the oldest age group (13, 80+ years). Education levels are similar, with most individuals in both groups having a college education (4-6 range), though females have a slightly higher count at the highest education level (5,6). Lastly, the income distribution shows both sexes having a right-skewed pattern, with most individuals in the lower income brackets (1-5), but males have a much higher count in the highest income bracket (8, $75,000 or more).

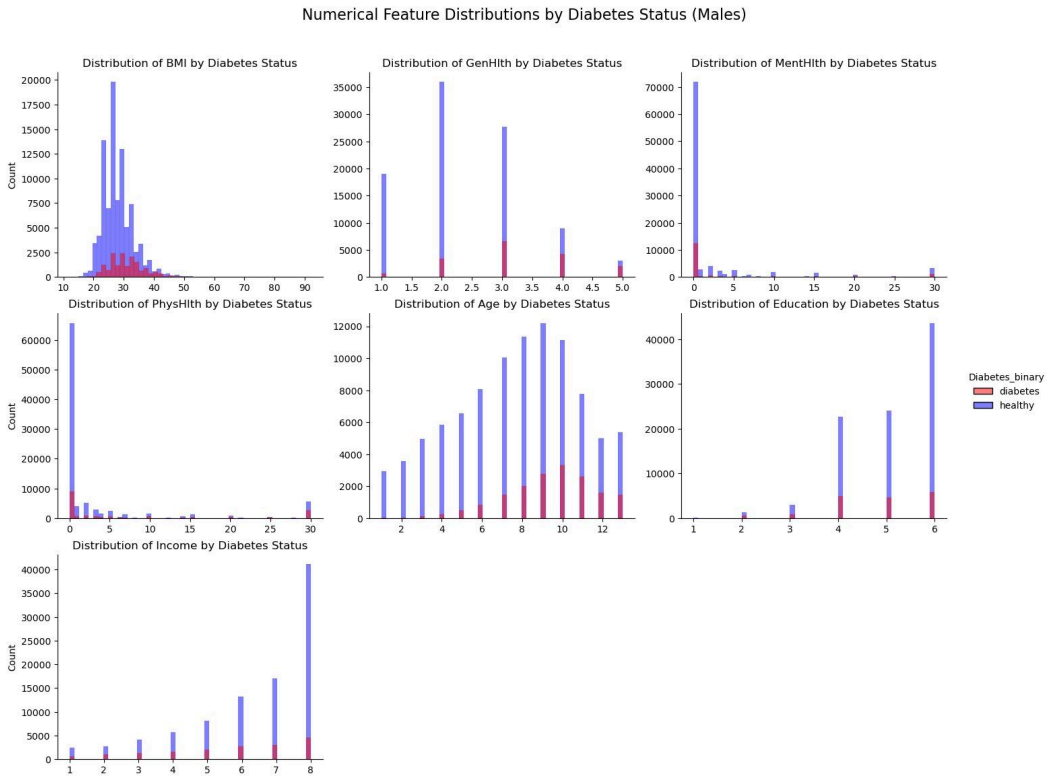

male
female
HighBP
HighChol
CholCheck
Smoker
Stroke
HeartDiseaseorAttack
PhysActivity
Fruits
Veggies
HvyAlcoholConsump
AnyHealthcare
NoDocbcCost
DiffWalk
Percentage
0.0
1.0

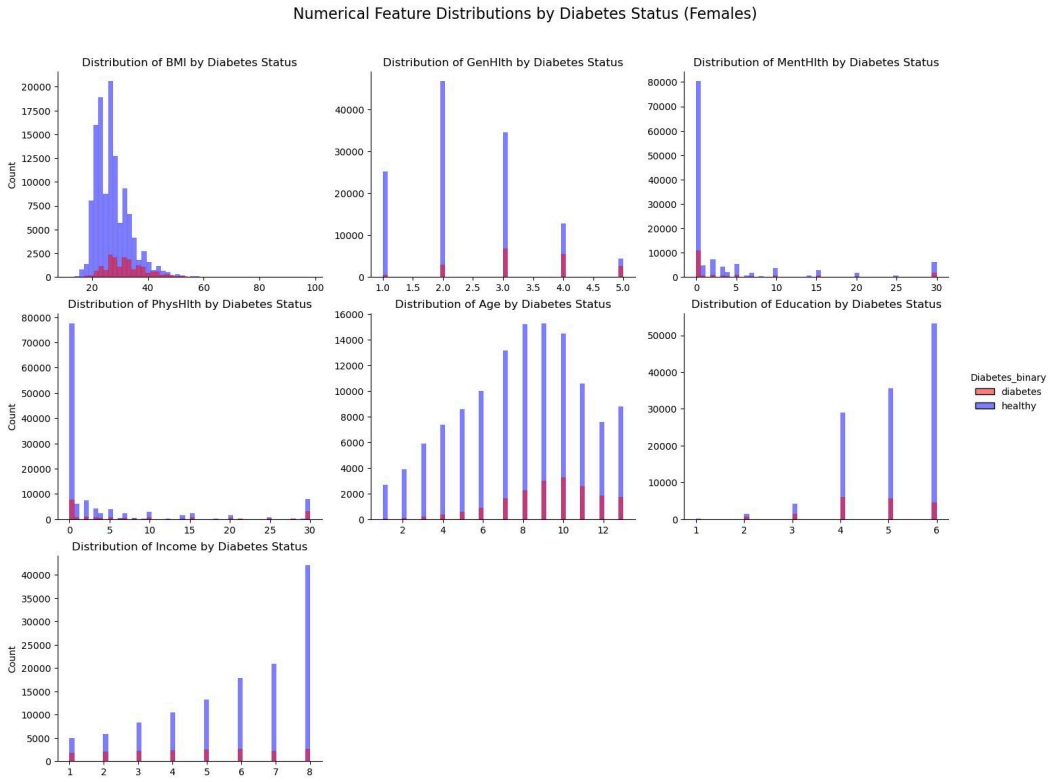
The figure above compares the percentage differences in binary variables between males and females within the CDC dataset. For HighBP (high blood pressure), males exhibit a 5% higher prevalence than females, indicating a slightly greater tendency among men to be affected by this condition. HighChol (high cholesterol) follows a similar trend, with males showing a 4% higher incidence than females, suggesting a notable trending difference in cholesterol levels between the sexes. In CholCheck (cholesterol check in the past 5 years), males are 10% more likely not to have undergone a check than females, pointing to a potential gap in preventive health screening practices among men. Smoking (Smoker) shows a significant difference, with males being 10% more likely to be smokers than females, reflecting a stronger association of tobacco with male demographics. Stroke prevalence shows no difference, with an equal 50% split between males and females who have not experienced a stroke, indicating a balanced distribution of this condition across sexes. HeartDiseaseorAttack (coronary heart disease or myocardial infarction) demonstrates a substantial disparity, with males having a 20% higher prevalence than females, underscoring a considerably greater risk of cardiovascular events among men. Physical activity (PhysActivity) indicates that males participate 4% more frequently than females, suggesting a slight edge in physical engagement among men. For eating fruit habits, females exceed males by 10%, indicating a notable preference or habit among women for fruit intake. Veggie consumption also favors females, who exceed males by 10%, highlighting a stronger inclination toward vegetable consumption among women. Heavy alcohol consumption (HvyAlcoholConsum) remains nearly equal between males and females, with only a negligible difference, suggesting similar drinking patterns across sexes. Anyhealthcare coverage shows females at a 5% higher rate than males, implying slightly greater access or utilization of healthcare services among women. NoDocbcCost (inability to see a doctor due to cost) is 9% higher among females than

males, indicating a greater financial barrier to medical care for women. Lastly, DiffWalk (difficulty walking) is 10% higher in females than males, pointing to a more pronounced challenge with mobility among women in the dataset.

4. Feature differences between diabetic patients and healthy controls stratified by sex

Finally, we explored the feature differences between diabetic patients and healthy controls in males and females respectively. This exploration enables us to identify the most likely diabetic contributors in males and females separately, which can guide us to answer question 2.

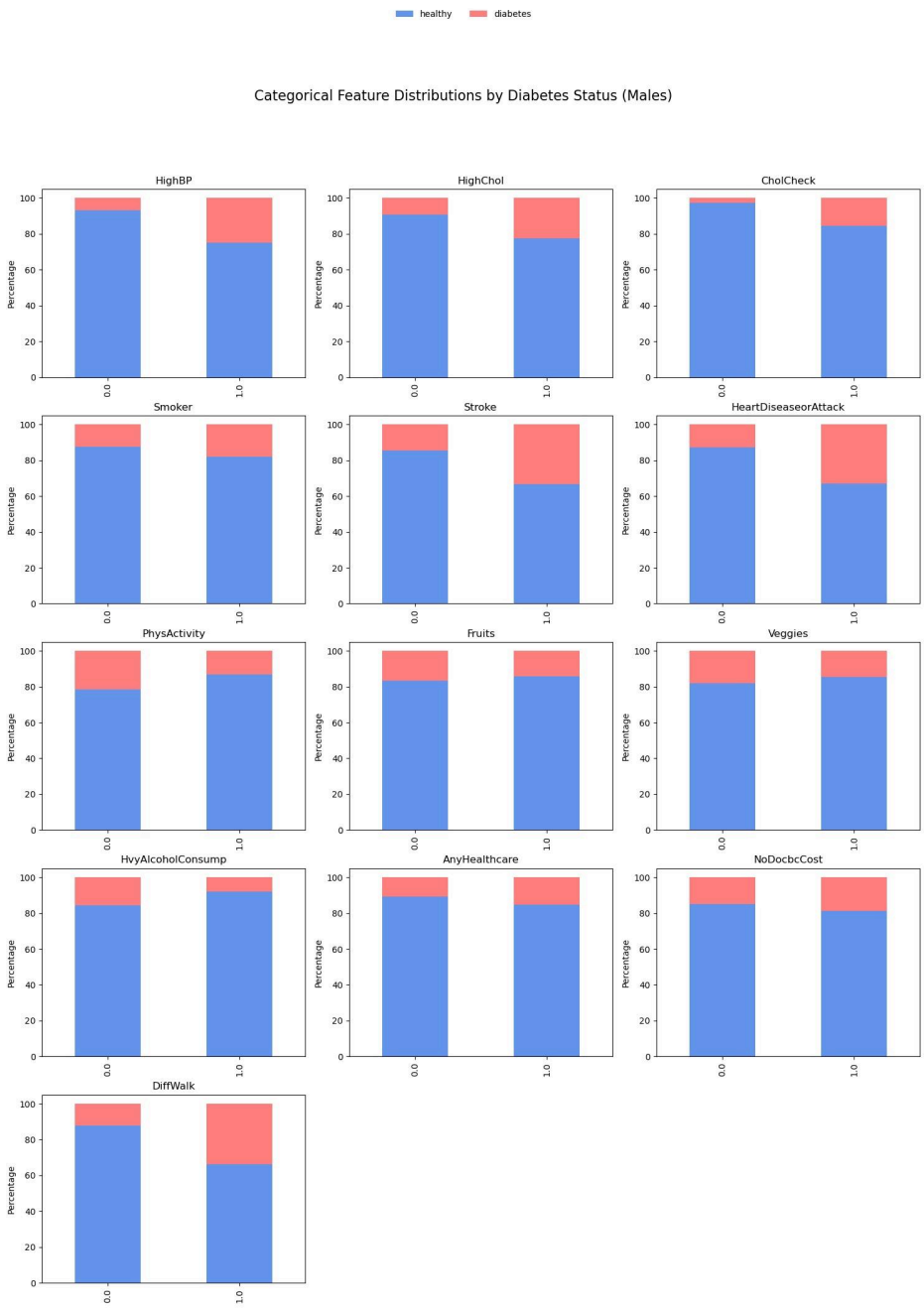


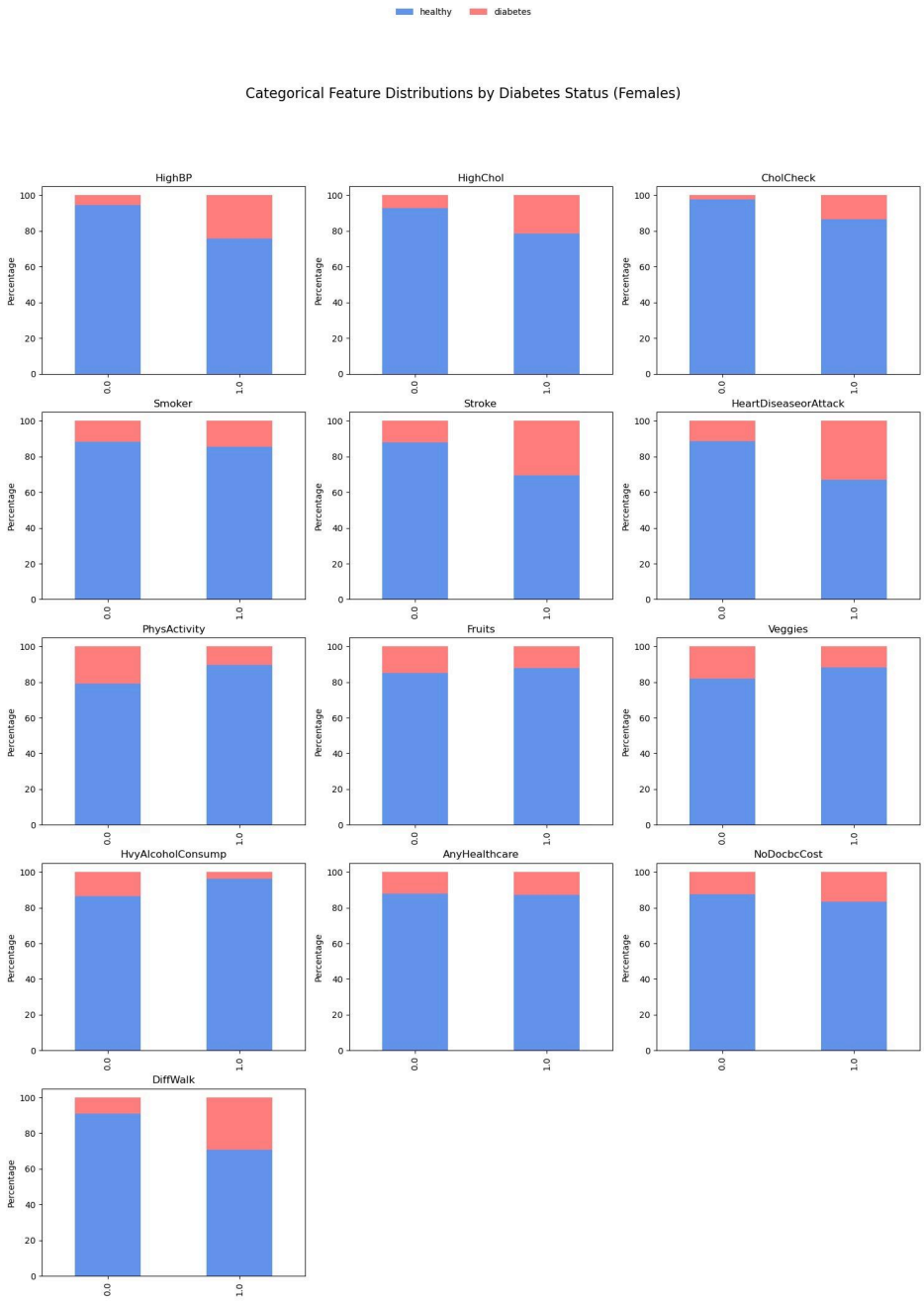


For the numerical features as shown above, we observed several clear distributional shifts between diabetic and non-diabetic groups in both sexes. Body Mass Index (BMI) consistently appeared higher among individuals with diabetes, with this difference more pronounced in females. Age-related variables also skewed older in the diabetic group, aligning with the established association between age and Type 2 diabetes. In females, features such as BMI, systolic blood pressure, and cholesterol-related measures displayed tighter and more separated distributions across diabetes status, suggesting these features may have stronger predictive power in this subgroup. In contrast, these same variables in males often exhibited broader or overlapping distributions, potentially limiting their usefulness in male-specific predictive models.

Furthermore, several features displayed non-normal or multimodal distributions—especially within the diabetic group—indicating underlying complexity or the presence of subpopulations.

These patterns suggest that linear models may be insufficient to capture the full scope of associations within the data. Instead, nonlinear approaches such as decision trees, ensemble methods (e.g., Random Forests, XGBoost), or neural networks may be more appropriate due to their ability to model interactions and non-additive effects.





The analysis of categorical variables as shown above, using stacked bar plots, revealed additional patterns. Features related to health behavior and access to care—such as physical activity levels, general health perception, and frequency of healthcare visits—differed meaningfully between diabetic and non-diabetic individuals. These differences were visible in both sexes but again tended to be more pronounced among females. For example, a larger proportion of diabetic

females reported poor general health, infrequent physical activity, or more frequent mental health issues, while non-diabetic females more often reported healthier behaviors and better perceived health. In males, while similar trends existed, the distributions were generally flatter, suggesting less distinction across diabetes status for many categorical variables.

These findings suggest that behavioral and perception-based features, especially those captured by multi-level categorical variables, may contribute more strongly to diabetes prediction in females than in males. The presence of multiple response levels also supports the use of models that handle categorical data natively and effectively, without requiring excessive preprocessing or one-hot encoding. Tree-based models are particularly advantageous in this context, as they can naturally accommodate such complexity.

Taken together, the exploratory findings from both numerical and categorical variables indicate that the contribution of individual features to diabetes prediction is likely to differ between males and females, both in magnitude and relevance. This reinforces the importance of building and evaluating separate machine learning models for each sex. It also motivates the use of feature interpretability tools, such as SHAP values or permutation importance, to identify which variables drive predictions in each subgroup. Features that are highly informative in one sex may be weak or irrelevant in the other, and failing to account for these differences could compromise model accuracy and equity.

Overall, this exploratory analysis offers critical guidance for model design, feature selection, and interpretability in the context of sex-specific diabetes risk modeling. These insights will inform the next phase of our work, where we build, train, and evaluate separate machine learning models to further investigate and compare feature contributions across sexes.

Methodology

RQ1: Which machine learning model achieves the highest performance, particularly in sensitivity, for predicting diabetes using the CDC dataset?

1. Machine learning methods to be used:

Logistic Regression: A supervised machine learning algorithm used for classification problems. It's a linear model that estimates the log odds of diabetes presence based on input features. It requires preprocessing steps such as scaling numerical variables (e.g., BMI, Age) and one-hot encoding categorical variables to handle multicollinearity and ensure model stability.

Random Forest: A machine learning algorithm that uses many decision trees to make better predictions. Its ensemble method comprises multiple decision trees, which predict diabetes by averaging tree outputs. It uses Gini importance to assess feature impact and employs random search cross-validation to tune hyperparameters like the number of trees and the maximum depth, accommodating unprocessed data with encoded categoricals.

Neural Network: A machine learning algorithm that uses many decision trees to make better predictions. It is a non-linear model with configurable hidden layers and neurons, optimized via backpropagation and learning rate adjustments. Tuning focuses on the number of layers, neuron count, and learning rate, with data preprocessed similarly to logistic regression for consistency.

2. How to compare the performances to identify the outperforming model which will be applied in question 2?

Models will be trained and evaluated using 10-fold cross-validation, randomly splitting the dataset into 10 equal folds, training on nine folds, and testing the remaining fold each iteration to prevent overfitting. Performance metrics include:

Accuracy: The proportion of correct predictions. Accurate positive and true negative are divided by the total.

Sensitivity: The ability to correctly identify diabetic cases, prioritized to minimize false negatives given the public health impact of missed diagnoses. True positive is divided by the total with the condition.

Specificity: The proportion of non-diabetic cases correctly identified. True negative divided by the total without the condition.

F1-Score: It provides a balanced measure and harmonic mean of precision and recall. Given the dataset's class imbalance (approximately 11% diabetes prevalence per CDC data), we will apply SMOTE-NC (Synthetic Minority Oversampling Technique for Numerical and Categorical data) to generate synthetic samples for the minority class if needed. The model demonstrating the highest sensitivity across the all-sample, male, and female subsets will be selected for RQ2 analysis.

$$\text{F1 Score} = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

RQ2: What are the key features or factors associated with diabetes that differ between male and female patients in the CDC dataset?

1. How to extract the feature importances from either male or female subset?

The original features will be used to build the models, and afterwards, the coefficients/indices will be extracted to deduce the importance of each feature and their contribution to predicting our binary target variable, diabetes status. The first method we will use is logistic regression. This provides log-odds coefficients for each attribute which is used to identify the strength of the relationship between a feature and the probability of diabetes in patients. In order to analyze the associated features for both genders, we will fit the logistic regression model on male/female subsets then extract and compare the log-odds coefficients. The next machine learning algorithm to be used is random forest. This model consists of decision trees that predict diabetes status by evaluating various features and splitting the data based on the values that reduce impurity. Gini importance indices measure the mean decrease in impurity to measure the importance of an attribute. The prediction is made using the overall results of each decision tree to determine classification (diabetes/healthy). Two separate models will be used for each gender so that Gini importance scores can be compared to find predictive features in males versus females. We will utilize random search cross validation by using a random hyperparameter grid in order to perform hyperparameter tuning for the random forest model. The last machine learning model we will build is a neural network. This method consists of forward propagation which is when features are passed through multiple layers (input,hidden,output) where linear transformation occurs and is then passed to an activation function to create a non-linear model. Next, backpropagation occurs which is when the model learns by comparing the predicted and true value by using a loss function with the objective of minimizing the loss. The gradients are then computed and used to adjust weights to minimize the loss/error. Finally, the weights are updated opposite in direction of the gradient and the process repeats over a number of epochs. We will fine tune some hyperparameters such as the number of hidden layers, the number of neurons, and

the learning rate. To identify the important features between genders in our neural network model, we can use permutation feature importance for each subset and evaluate the performance of each model.

2. How to measure the rank changes of the features between males and females?

After ranking features by their importance scores in each subset (e.g., highest to lowest coefficients for logistic regression, highest to lowest Gini importance indices for random forest, or highest to lowest permutation-based importance values for neural networks), we will compute the weight change indices as importance differences between the male and female models for individual features. For instance, if the BMI importance is 2 for females but 6 for males, the importance difference is 4 for BMI. A positive importance difference signifies a feature is more associated with males, whereas a negative importance difference signifies a feature is more associated with females, which provides insights into sex-specific risk factors.

Analysis

Data preprocessing

To make sure the data consistency between the follow-up analyses, we built the different machine learning models based on the same preprocessing output. After reading in the raw data, missing values for the individual features and the target variable were examined with the command *dat_df.isnull().sum()*, and any samples with missing value for any columns were removed with the command *dat_df.dropna()*. Fortunately, this is a very clean dataset without any missing value for any feature or target variable.

The dataset include two types of features, numerical and binary. The numerical features include: BMI, GenHlth, MentHlth, PhysHlth, Age, Education and Income. The binary features include: HighBP, HighChol, CholCheck, Smoker, Stroke, HeartDiseaseorAttack, PhysActivity, Fruits, Veggies, HvyAlcoholConsump, AnyHealthcare, NoDocbcCost, DiffWalk and Sex. To less the influences of different ranges from numerical features on follow-up model building and make all numerical features be within the same range, all the numerical features were standardized with Z-score normalization with the command *StandardScaler().fit_transform(dat_df[numVars])*, whereas the binary features were kept unchanged. Afterwards, the preprocessed output was provided as the input for follow-up analyses.

Model Results

RQ1: Which machine learning model achieves the highest performance, particularly in sensitivity, for predicting diabetes using the CDC dataset?

	Logistic Regression	Random Forest	Neural Network
Accuracy	0.73	0.72	0.71 (0.64)
Sensitivity	0.77	0.78	0.8 (0.88)
Specificity	0.73	0.71	0.7 (0.60)
F1-score	0.44	0.44	0.44 (0.41)
AUC	0.82	0.82	0.83

Q1 Logistic regression

To address RQ1, a logistic regression model was implemented using a train-test split approach. Numerical variables were scaled using StandardScaler, and categorical features were adapted as integer arrays. The data was split 70/30, stratified on the target variable, and the class imbalance was addressed using class_weight='balanced'.The confusion matrix revealed 47,612 true negatives, 17,888 false positives, 2,522 false negatives, and 8,082 true positives, achieving a sensitivity of 0.77 in identifying diabetic individuals. The ROC curve confirmed good discriminatory power with an AUC of 0.82. Permutation importance analysis showed GenHlth (0.0591), BMI (0.0278), and Age (0.0336) as top predictors.An odds ratio analysis further clarified the direction and strength of associations between features and diabetes risk. Notably, GenHlth had the highest odds ratio (1.86), followed by BMI (1.62), Age (1.58), HighBP (1.44), and HighChol (1.33), suggesting individuals with poor general health, higher BMI, older age, or high blood pressure/cholesterol have significantly increased odds of having diabetes. Factors with protective capability that reduced the diabetic risk included higher income (0.89), higher education (0.96), and heavy alcohol consumption (0.84), though the latter may reflect behavioral confounding rather than causation. The Sex with an odd ratio of (1.15) indicates a slight increase

in diabetes risk in males, laying the groundwork for further gender-specific analysis in RQ2. These odds ratios support highlighting key risk factors driving true positive identifications.These findings confirm that logistic regression performs well in identifying at-risk patients with a sensitivity that meets public health goals. The model's consistent performance on the test set supports H1 and sets a benchmark for comparing more complex models in future work.

Q1 Random forest

For the random forest classifier, a train test split approach was used along with balanced weights. The balanced weights helped counteract the effect of the imbalanced nature of the data set (most people do not have diabetes). To optimize the random forest, a grid optimization approach was used. Grid search was used to tune the hyper-parameters and optimized for recall. Using this method, we tuned the model and found that the best parameters for our purposes:

```
n_estimators=200,
max_depth=5,
min_samples_split=20,
max_features='log2',
class_weight='balanced',
random_state=123
```

The accuracy score for diabetes predictions using this random forest classifier model was .72. This means that the model was correctly predicting diabetes 72% of the time (total of correct predictions/total number of predictions). The random forest model has a lower accuracy score than the logistic regression model we were using as a baseline. The random forest classifier had a

sensitivity/recall of .78. This implies that it has a relatively low number of false negatives. The specificity was .71. This means that the true negatives are relatively high. Unfortunately, the F1-score is merely .44, which is unfortunately low, but the other models performed similarly on this metric. Finally, the AUC of the ROC of the random forest model was .82 which shows good discriminatory power.

For feature importance, we used permutation tests. This revealed that, for the random forest classifier, the top predictors were: General Health (.041), Age (.024), and High Blood Pressure (.022).

Q1 Neural network

The neural network model was implemented on the male and female superset using an 80/20 split while maintaining the target variable's distribution. Balanced weights were used to handle the imbalanced dataset since the majority of our data contained healthy individuals. The confusion matrix in Appendix C shows 31,313 true negatives, 12,308 false positives, 1,554 false negatives, and 5,561 true positives. A low number of false negatives allows for a high sensitivity for the superset model to correctly detect diabetic patients.

Now, our main priority is to maximize sensitivity since it is crucial in correctly identifying diabetes patients (true positive). The Neural Network model provided the best result in terms of sensitivity (0.80). In this case, the Neural Network model correctly detects 80% of all diabetic patients. The specificity of the Neural Network model is the lowest (0.70) compared to the other two models for detecting healthy patients (true negative). This means a greater likelihood for the Neural Network model to misclassify healthy individuals as diabetics (false positive). Even though this may be the case, it would be an acceptable trade-off to misclassify healthy

individuals (false positive) instead of diabetic individuals (false negative). The F1-score is maintained at a low score of 0.44 for all models. Additionally, the Neural Network model contains the highest AUC score of the ROC is (0.83) which means a better overall ability for the model to classify between diabetic and healthy patients.

To find baseline important features in the overall neural network model, we used the permutation feature importance technique. The outcome of this analysis revealed that the top predictors from the male and female superset neural network model were GenHlth(0.054), BMI(0.040), and Age(0.035).

RQ2: What are the key features or factors associated with diabetes that differ between male and female patients in the CDC dataset?

The two tables below display the performance results from the male and female subset models for Logistic Regression, Random Forest, and Neural Network.

Performance measures in the male subset:

	Logistic Regression	**Random Forest**	**Neural Network**
Accuracy	0.72	0.71	0.60
Sensitivity	0.76	0.76	0.89
Specificity	0.71	0.70	0.55
F1-score	0.45	0.44	0.40
AUC	0.81	0.80	0.81

Performance measures in the female subset:

	Logistic Regression	Random Forest	Neural Network
Accuracy	0.75	0.72	0.67
Sensitivity	0.77	0.79	0.87
Specificity	0.74	0.71	0.64
F1-score	0.43	0.43	0.41
AUC	0.83	.83	0.84

Q2 Logistic regression

For RQ2, the logistic regression model was applied separately to male and female subsets to investigate gender-specific robustness and feature variations. Performance metrics revealed :

Male Performance: Sensitivity = 0.759, AUC = 0.810

Female Performance: Sensitivity = 0.766, AUC = 0.830

indicating a slightly higher discriminatory power for females. The importance of permutation analysis highlighted distinct gender differences, with BMI and HighBP being more influential for females and age-dominant for males. The difference in permutation importance scores (male - female): Age showed a significant increase (+0.036) for males, while BMI decreased (-0.013) for males, indicating greater relevance for females. High Blood Pressure and High Cholesterol were also more impactful for females, whereas smoking status exhibited a negative importance for males, possibly due to interactions with other health behaviors. Odds ratio analysis further elucidated these patterns: for males, Age (OR = 1.58) and GenHlth (OR = 1.86) were the most significant, linking older age and poor general health to higher diabetes risk; for females, BMI

(OR = 1.62) and HighBP (OR = 1.44) were key, emphasizing the roles of body mass index and hypertension. The odds ratios observed are consistent with the variations in permutation importance; these results support Hypothesis 2 (H2), demonstrating that males and females exhibit distinct patterns of diabetes risk factors. While logistic regression contributed valuable results, its linear nature may overlook non-linear relationships identified during exploratory data analysis (EDA). Future studies could employ tree-based or neural network models to capture these complexities.

Q2 Random forest

Having split the dataset into male and female subsets, we used the random forest classifier model on each one to determine differences between male and female diabetes patients. Afterwards, we performed feature importance analysis again to identify which features have more or less predictive power between sexes.

Male Performance Metrics (Random Forest): Sensitivity = 0.76, AUC = 0.80

Female Performance Metrics (Random Forest): Sensitivity = 0.79, AUC = .83

The permutation tests for feature importance revealed that age and general health score were more important for males and that BMI and high blood pressure was more important for females when it comes to predicting diabetes performance using the random forest model.

The random forest model's male-female subset analysis revealed that there are substantial differences between diabetes predictive ability within the model (at least in terms of recall).

Additionally, the feature importance permutation analysis confirms that there are differences between importance of particular features between male and female patients. The random forest provides a baseline of differences between sexes that may be expanded upon during the neural net analysis.

Q2 Neural network

Two separate neural network models have been created for male and female subsets. Each model contains an 80/20 split where the proportion of classes in the training and test sets stay the same. To handle class imbalance, the parameter class_weight='balanced' is used for equal optimization in both classes. The confusion matrices for the male model and female model in Appendix F show a similar amount of false positives. The female model contains more true negatives and true positives which means that there are more diabetes/healthy female samples in the data. The male model has less false negatives than the female model which induces a better recall.

Male Performance: Sensitivity = 0.89, AUC = 0.81

Female Performance: Sensitivity = 0.87, AUC = 0.84

The neural network model for the male subset produces the lowest accuracy but highest sensitivity value across all models. This means that this model correctly detects 89% of all diabetic patients in the male subset. Similar to the male and female neural network model, the male model has the smallest specificity (0.55) compared to the other two male subset models which means the model misclassifies 45% of healthy individuals as diabetic (false positives).

Among the three male models, both the neural network and logistic regression models have the same AUC score of 0.81. Therefore, both the neural network and logistic regression models have similar classification abilities. As for the female subset model, it carries the smallest accuracy and F1-score. However, this model accurately identifies 87% of all diabetic patients in the male subset. This female model also has the lowest specificity (0.64) compared to the logistic regression and random forest female models, meaning it misclassified 36% of healthy patients as diabetic (false positives). Lastly, the neural network female model holds the highest AUC score (0.84) compared to the logistic regression and random forest female subset models which leads to a more favorable classification performance.

Finally, we applied permutation feature importance to each male and female subset neural network model to extract and compare their top predictors. The difference between the top predictors for the male and female subset models show that Age is a significantly strong predictor for males while BMI, HighBP, and HighChol was found to be more important for females. Overall, the performance of the male and female neural network models showed slight differences in recall between the two gender models, but displayed a notable difference when compared to the male and female superset model. The important predictors of the male model (Appendix K) and female model (Appendix L) proves that there are differences in the features associated with diabetes for male and female patients. In the end, the neural network model outperformed the logistic regression and random forest models based on its performance metrics for all subsets of the diabetes data.

Conclusion

After building logistic regression, random forest and neural network models from the male and female subsets, computing feature importance measures, and comparing the feature importance measures between males and females to generate the feature importance change index by subtracting the female importance measure from male importance measure for any given feature, we compiled a feature importance change index heatmap for the three models after sorting the averaged feature importance change indices from high to low.

Data Visualization

RQ1: Which machine learning model achieves the highest performance, particularly in sensitivity, for predicting diabetes using the CDC dataset?

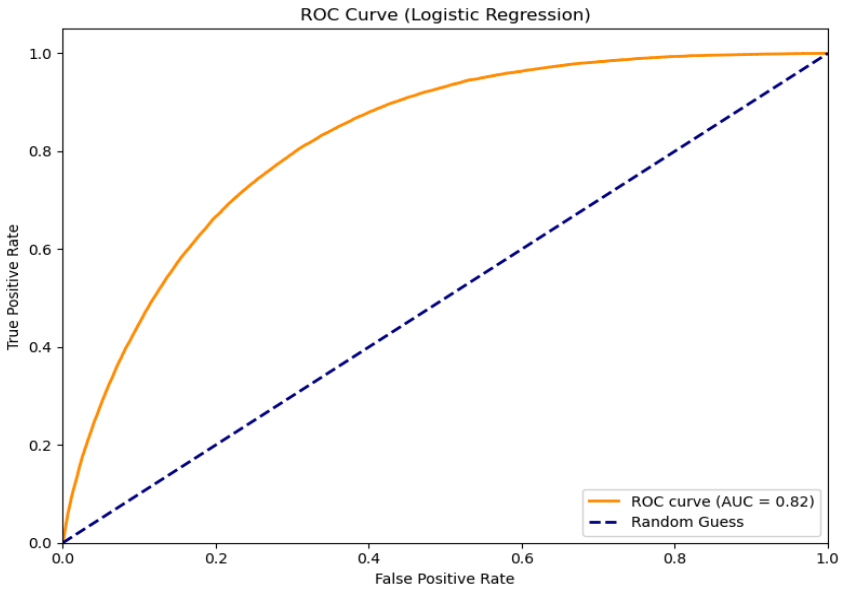
We compared the performances of three machine learning models (Logistic regression, random forest and neural network) in terms of accuracy, sensitivity, specificity, and F1-score. We prioritize sensitivity because these are disease prediction models, and we don't want the models to miss too many diabetic patients to delay their treatments. Below is a table summarizing the performance metrics between the three models. The neural network outperforms the other two models in terms of sensitivity. The accuracy of the neural network is lower than those of the other two models due to its higher probability of misclassifying healthy controls as diabetic patients (lower specificity). Nevertheless, these misclassifications do not harm the healthy controls and even encourage them to live healthier lives to avoid the occurrence of diabetes.

	Logistic Regression	**Random Forest**	**Neural Network**
Accuracy	0.73	0.72	0.71 (0.64)
Sensitivity	0.77	0.78	0.8 (0.88)
Specificity	0.73	0.71	0.7 (0.60)
F1-score	0.44	0.44	0.44 (0.41)
AUC	0.82	0.82	0.83

Then, we compared the areas under the curve (AUC) derived from the receiver operating characteristic (ROC) between the three models. Because ROC runs false positive rate (the rate of misclassifying healthy controls as diabetic patients) against true positive rate (the rate of correctly identifying diabetic patients), it can show the overall performance and identify the

model with the best trade-off. The following graphs are the ROC curves for the three models, respectively. For the confusion matrices of the three models, please refer to Appendices A-C.

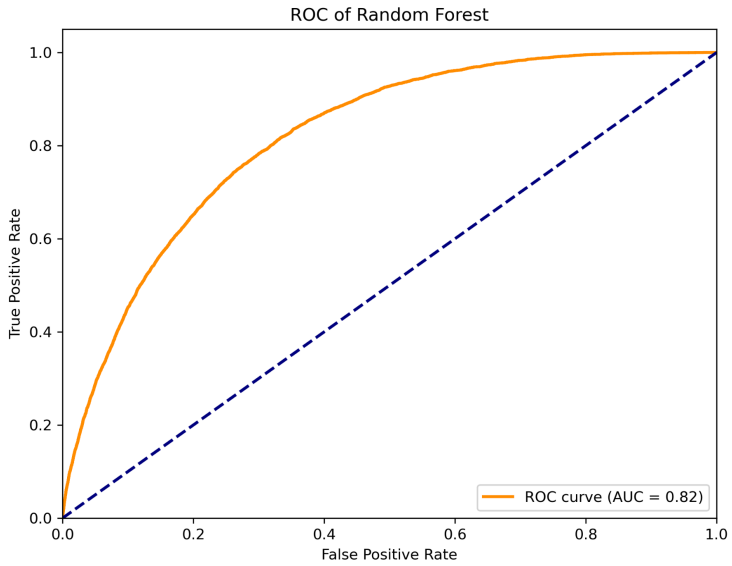
ROC curve for Logistic Regression:



In addition, we also generated a table for the odds ratios of the features in Logistic regression:

Feature	Odds ratio	Feature	Odds ratio	Feature	Odds ratio
HighBP	1.44	PhysActivity	0.98	MentHlth	0.97
HighChol	1.33	Fruits	0.97	PhysHlth	0.94
CholCheck	1.28	Veggies	0.98	DiffWalk	1.04
BMI	1.62	HvyAlcoholConsump	0.84	Age	1.58
Smoker	1	AnyHealthcare	1.02	Education	0.96
Stroke	1.03	NoDocbcCost	1.01	Income	0.89
HeartDiseaseorAttack	1.08	GenHlth	1.86	Sex	1.15

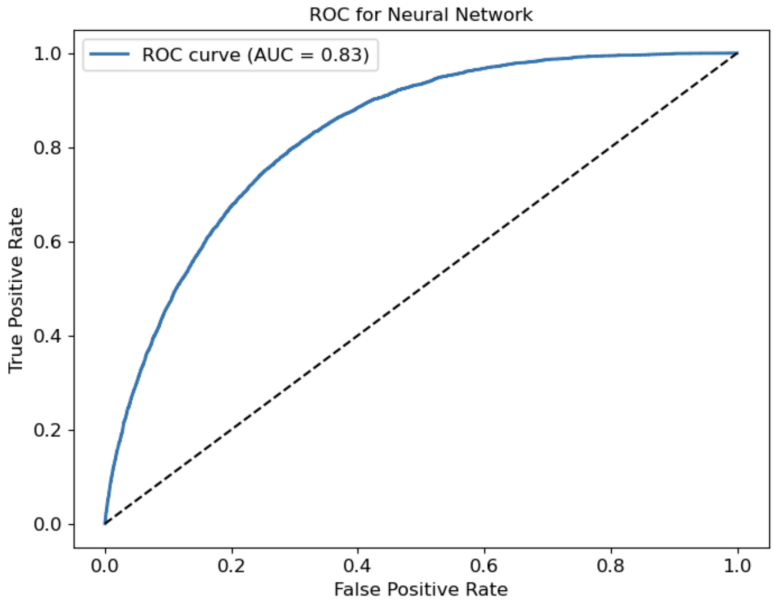
ROC curve for Random Forest:



Parameters for Random Forest: (Used Hyperparameter Grid Search):

```
{'class_weight': 'balanced', 'max_depth': 5, 'max_features': 'log2',
'min_samples_split': 20, 'n_estimators': 200}
```

ROC curve for Neural Network:



AUC values derived from these ROCs were also listed in the table above. The neural network outperforms the other two models and shows the best overall performance in terms of AUC. The neural network model outperformed the other two models in terms of the key measure of sensitivity and the overall performance. Therefore, we will pay more attention to the outcomes from the neural network in the second question in this project.

RQ2: What are the key features or factors associated with diabetes that differ between male and female patients in the CDC dataset?

Before identifying which features are more associated to male or female diabetic patients, we compared the model performances in male and female subsets separately.

Performance measures in the male subset:

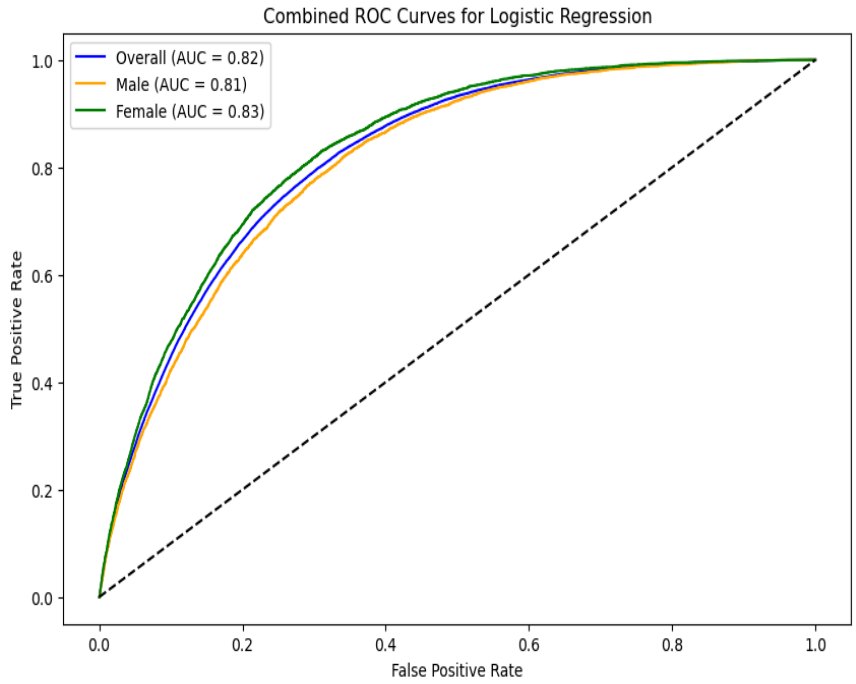
	Logistic Regression	**Random Forest**	**Neural Network**
Accuracy	0.72	0.71	0.60
Sensitivity	0.76	0.76	0.89
Specificity	0.71	0.70	0.55
F1-score	0.45	0.44	0.40
AUC	0.81	0.80	0.81

Performance measures in the female subset:

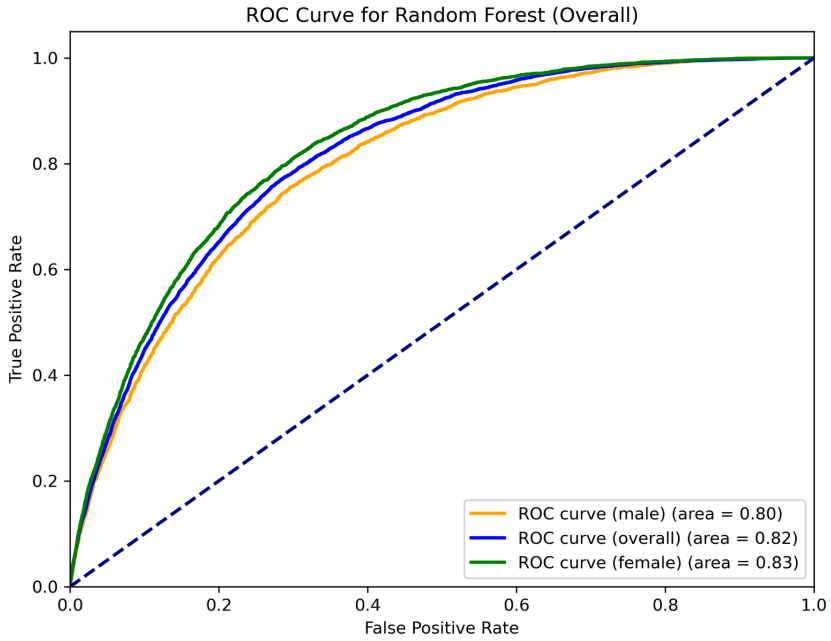
	Logistic Regression	Random Forest	Neural Network
Accuracy	0.75	.72	0.67
Sensitivity	0.77	.79	0.87
Specificity	0.74	.71	0.64
F1-score	0.43	.43	0.41
AUC	0.83	.83	0.84

The following graphs are the ROC curves for the three models in the male and female subsets, respectively. For the confusion matrices of the three models, please refer to Appendices D-F.

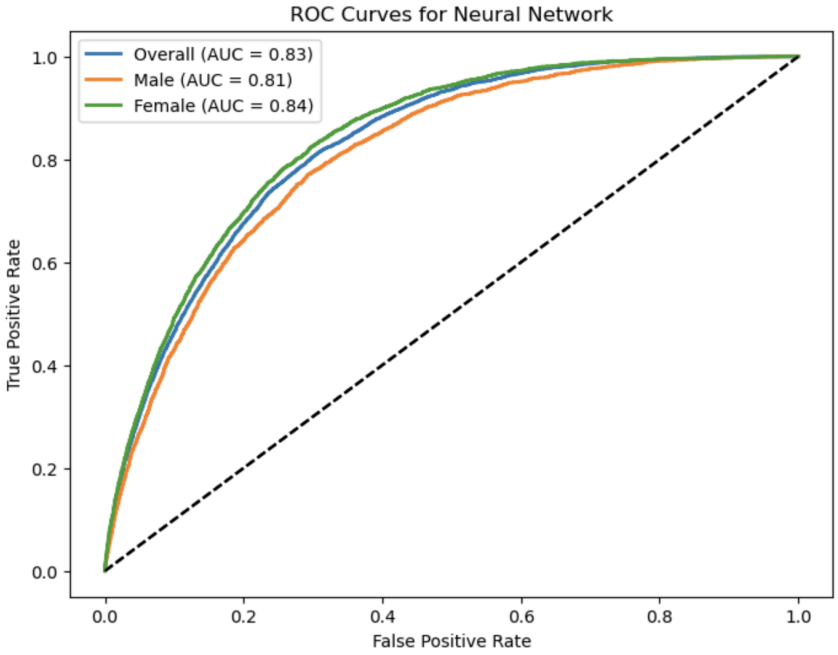
ROC curves from male and female subsets for Logistic regression:



ROC curves from male and female subsets for random forest:

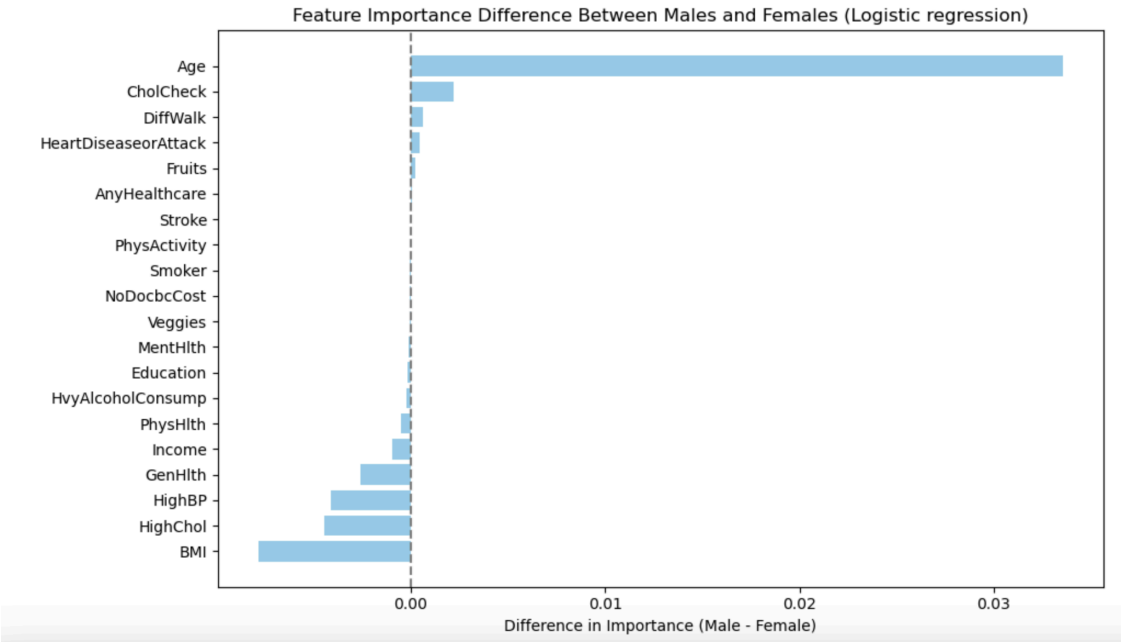


ROC curves from male and female subsets for neural network:

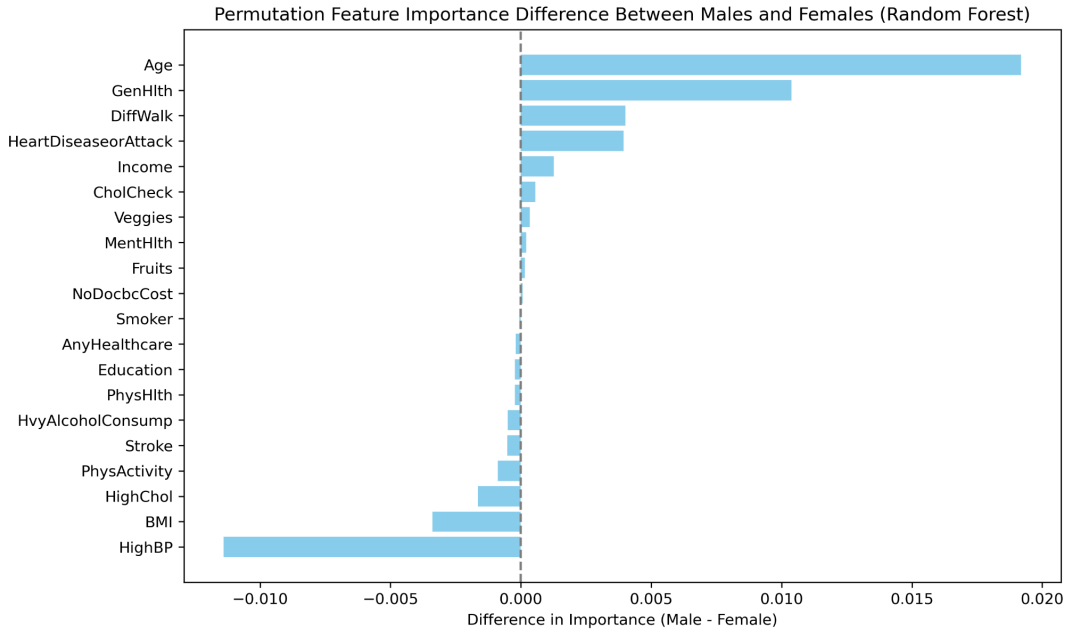


To assess which features contribute most to diabetes classification in males and females from the three models, we used permutation importance evaluated by the changes in AUC derived from ROC. This method reveals the drop in model performance when a feature's values are randomly shuffled, indicating how much the model relies on that feature. For the ranked bar plots of the permutation feature importance scores from either male or female subset for the three models, please refer to Appendices G-L. Then, we compared the change of importance score for each feature by subtracting the score of female from the score of male to get the following plots for the three models, respectively.

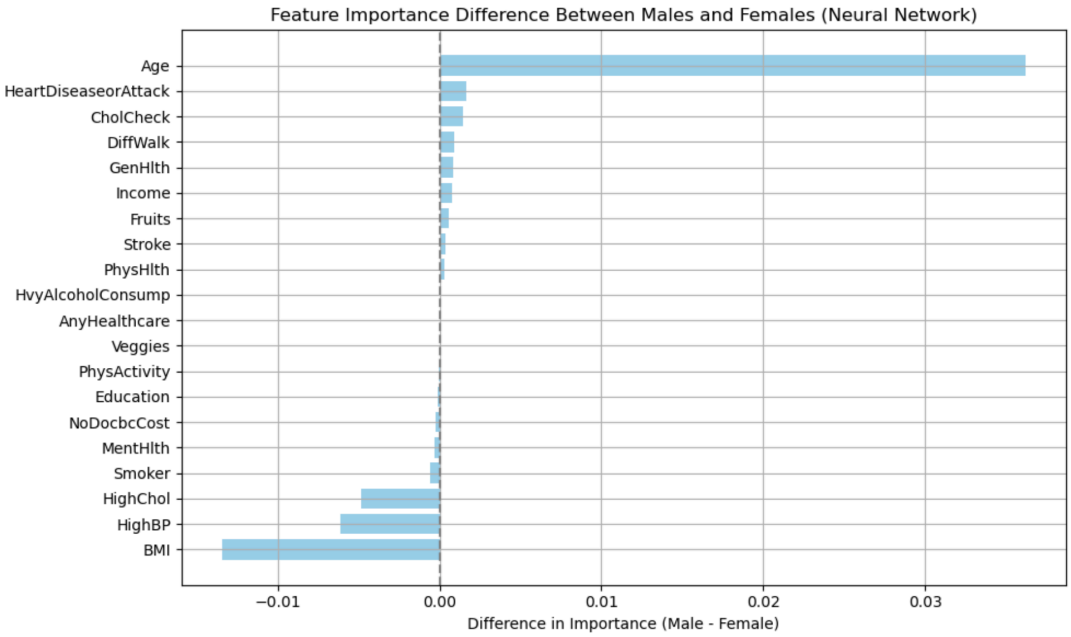
Logistic regression:



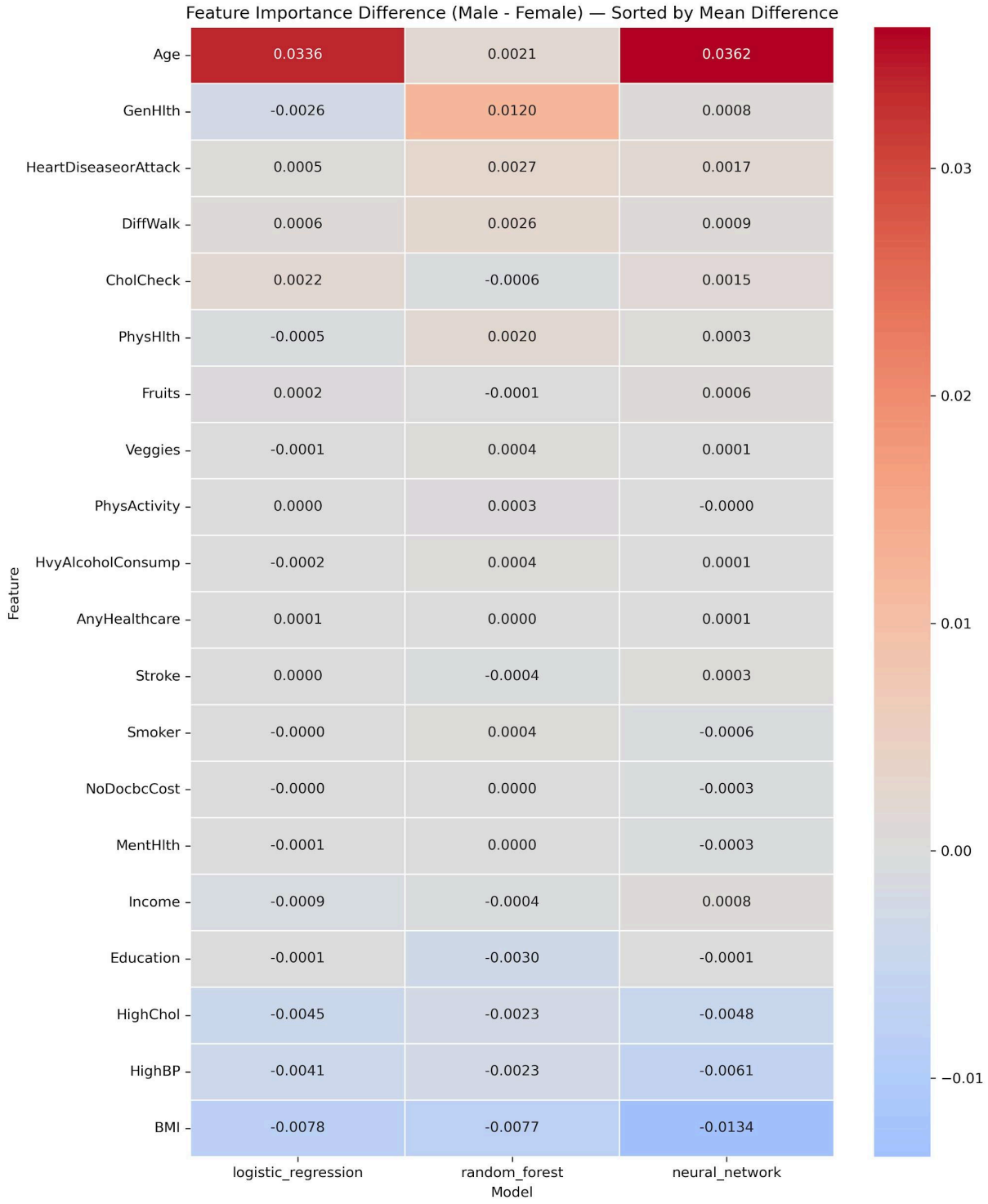
Random forest:



Neural network:



Finally, we integrated the male-female importance difference measures across the three models into one heatmap:



From the results of the changes of importance scores:

- Age was significantly more important in males than in females, showing an increase in importance scores for both Logistic regression and neural network. This suggests that age-related risk patterns for diabetes are more predictive among men.

- Conversely, BMI was more predictive in females across the three applied models. This could reflect different physiological or behavioral responses to weight between the sexes.

- High Blood Pressure and High Cholesterol were also more important for predicting diabetes in females than in males.

- Interestingly, some features, such as stroke and smoker, show discrepant associations with either male or female, possibly due to their weak effects or interactions with other health behaviors.

- Other factors like General Health, Income, and Alcohol Consumption had similar but low-to-moderate contributions across sexes.

These findings support the need for sex-specific risk models, as feature importance varies between males and females. Visualizing these differences helps clinicians and model developers tailor predictions and interventions accordingly.

Ethical Recommendations

The application of Machine learning methods in diabetes risk prediction enhances early detection and targeted intervention in the general public health. However, in the meantime, it also introduces ethical problems, which may include but are not limited to fairness, privacy, transparency, and responsible use of all different aspects. According to Quinn (2020, Chapter 1), ethical decision-making in data science requires prioritizing individual well-being and societal values, especially in health-related applications. The following analysis applies the project with five major ethical theories, Kantianism, act utilitarianism, rule utilitarianism, social contract theory, and virtue ethics, to pursue ethical practices.

Under the Fairness and Algorithmic Bias consideration, biased datasets can result in misdiagnosis for underrepresented populations. In our case, unequal sampling may reduce model accuracy for certain ethnic or socioeconomic groups. Social contract theory emphasizes mutual benefit and equitable treatment (Quinn, 2020, Chapter 2). To uphold fairness, stratified sampling and subgroup validation can be used to ensure the model serves all demographics properly. Rule utilitarianism supports adopting general rules, like bias audits and transparency guidelines, that increase overall happiness and justice when universally applied.

Model Transparency: Healthcare models must be interpretable to respect individuals' rational autonomy (Quinn, 2020, Chapter 2.6). Kantian ethics demands that individuals be treated as ends in themselves, not merely as means. Logistic regression, used as a comparison method with other ML methods in this project, supports this principle through interpretability and clarity. Transparent explanations in the technology method we are using empower patients and providers to make informed decisions, reinforcing public trust and understanding.

Privacy and Responsible Data Stewardship: Using publicly available health data still requires ethical safeguards. Quinn (2020, Chapter 7) emphasizes informed consent and confidentiality. Under social contract theory, data users are obliged to honor mutual rights, including privacy. Virtue ethics also requires that data scientists act with integrity, protecting participant identities even when not legally mandated.

Avoiding Misuse of Statistics: Practices like p-hacking or overfitting for better metrics may yield misleading results. According to act utilitarianism, such actions are unethical if the harms (e.g., misdiagnoses) outweigh the benefits (e.g., improved test accuracy). Kantianism also condemns manipulation that treats patients as mere data points. Instead, developers must ensure honest reporting, proper cross-validation, and transparency to promote the collective good.

Human Autonomy and Decision Support: Predictive models should assist, not replace, human independent judgment. Quinn (2020, Chapter 2.10) states that virtue ethics emphasizes acting with wisdom and compassion. Kantianism reinforces that people must never be used solely as tools. Therefore, models should be solely used to help human beings improve efficiency and accuracy, and encourage shared decision-making between patients and clinicians.

Conclusion:

Using ethical theories like Kantianism, utilitarianism, social contract theory, and virtue ethics to guide this project ensures that this diabetes prediction project meets ethical integrity. Each theory contributes a unique lens: rights (Kantianism), consequences (utilitarianism), fairness (social contract theory), and moral character (virtue ethics). The ethical goal is to empower patients, improve care equity, and respect human dignity while advancing health technology responsibly.

Challenges

When we were training the models, due to the imbalanced sample sizes between diabetic patients and healthy controls (13.93% diabetes v.s. 86.07% healthy controls), the models were not able to reach an ideal sensitivity, which is the prioritized performance measure in this project, because the sample size of diabetic patients is much less than the sample size of healthy controls, and the models are less likely to recognize the true positive than to recognize the true negative consequently. To resolve this issue, we applied cost-sensitive learning, or a balanced weight, to adjust the models to be more sensitive to recognizing diabetic patients than healthy controls. Technically, diabetic patients were assigned a higher class weight and healthy controls a lower class weight in model training. With cost-sensitive learning, the models were able to reach an ideal sensitivity, especially for the neural network (~90%).

Another challenge in the project is the imbalanced sample sizes between the sexes (44.03% males v.s. 55.97% females). One issue brought about by this imbalance is the discrepant model performance measures between the male and female subsets. The performance measures from the female subset are better than the performance measures from the male subset due to the larger sample size. To resolve this issue, we downsampled the female subset to make it almost the same size as the sample size of the male subset. After downsampling, the discrepancy between the performance measures of the male and female subsets is significantly reduced.

Recommendations

The neural network is the outperforming model among the three models (logistic regression, random forest and neural network) we built which can correctly classify true diabetic patients with the highest sensitivity (~90%). However, there is still room to improve the model (~10%). We used a relatively simple neural network model with two hidden layers and a low number of neurons (16 neurons for one hidden layer and 8 neurons for the other) because a complex neural network model is time-consuming and our project is time-limited. This simple model may not be competent to fully capture the characteristics from the large samples because we have a large total sample size (253,680) relative to a small number of neurons. Therefore, this reminds us to wonder whether increasing the number of hidden layers and the number of neurons of the hidden layers will be able to further increase its sensitivity? If it is, how much sensitivity can be increased? This is an interesting question to explore but needs more time and may need a computer with higher capability, such as a computing cluster.

In this project, we divided the samples into the male and female subsets to explore their respective most important features which are associated with the occurrence of diabetes. It is intriguing to consider whether further subdividing the male and female subsets will change the importance measures of the original male and female subsets. For example, if we subdivide the male samples to smoking males and non-smoking males based on the smoker feature, whether the feature importances of either smoking males or non-smoking males will be different than the feature importances of males with smoking and non-smoking status combined. The same thought can be extended to other features in the dataset. However, such subdividing analysis should be taken with caution because the sample size for a certain level of a feature may not guarantee such

analysis. For example, smoking female samples may comprise only a very small proportion of the total samples.

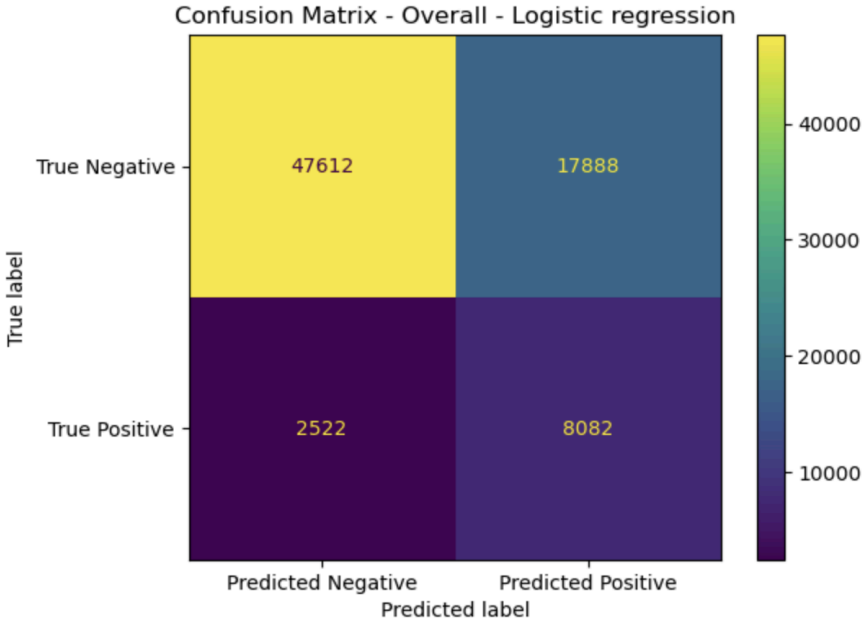
This dataset includes several disease-related features, such as high blood pressure, high cholesterol and heart disease. From the heatmap from data visualization, it can be observed that high blood pressure and high cholesterol are more associated with females to develop diabetes, whereas coronary heart disease is more associated with males to develop diabetes. However, we cannot say that a female with high blood pressure or high cholesterol is more likely to develop diabetes, and a male with coronary heart disease is more likely to develop diabetes. The reason behind this argument is because our machine learning models only provide us with the associations between the features with diabetes but they are not necessarily the causes. We are not sure whether a person with high blood pressure is more likely to develop diabetes, or a diabetic person is more likely to develop to high blood pressure. This brought us some ideas on whether we can leverage data to disentangle the cause-result relationships between diabetes and the other disease-related features.

References

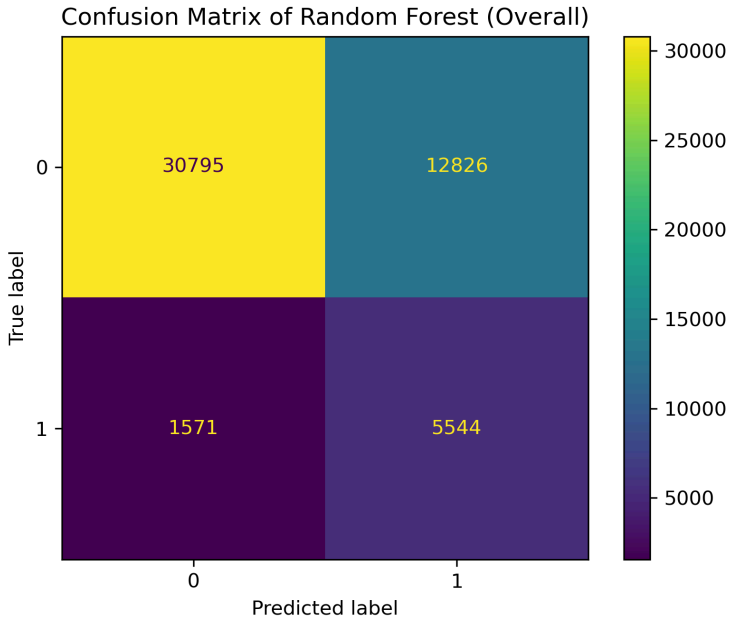
1. ASU. (2025, Spring). Abuse of statistics – General [PowerPoint slides]. DAT 250. Arizona State University.
2. ASU. (2025, Spring). Chapter 2: Introduction to ethical thinking [PowerPoint slides]. DAT 250. Arizona State University.
3. Centers for Disease Control and Prevention. (2020, April 1). Fact sheets | Budget. https://www.cdc.gov/budget/fact-sheets/index.html
4. Centers for Disease Control and Prevention. (2024, June 5). CDC leadership. https://www.cdc.gov/about/leadership/index.html
5. Centers for Disease Control and Prevention. (2025). Archived page: Director of CDC. https://archive.cdc.gov/#/details?url=https://www.cdc.gov/about/leadership/director.html00
6. Quinn, M. J. (2020). Ethics for the information age (8th ed.). Pearson Education.
7. Semrush. (2025). Semrush: CDC.gov competitors. https://www.semrush.com/website/cdc.gov/competitors/

Appendix

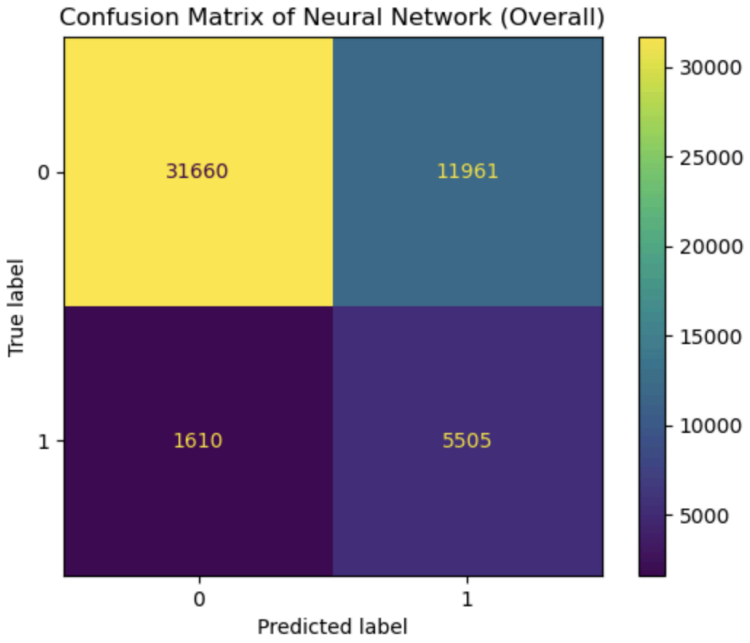
Appendix A. Confusion matrix for Logistic Regression



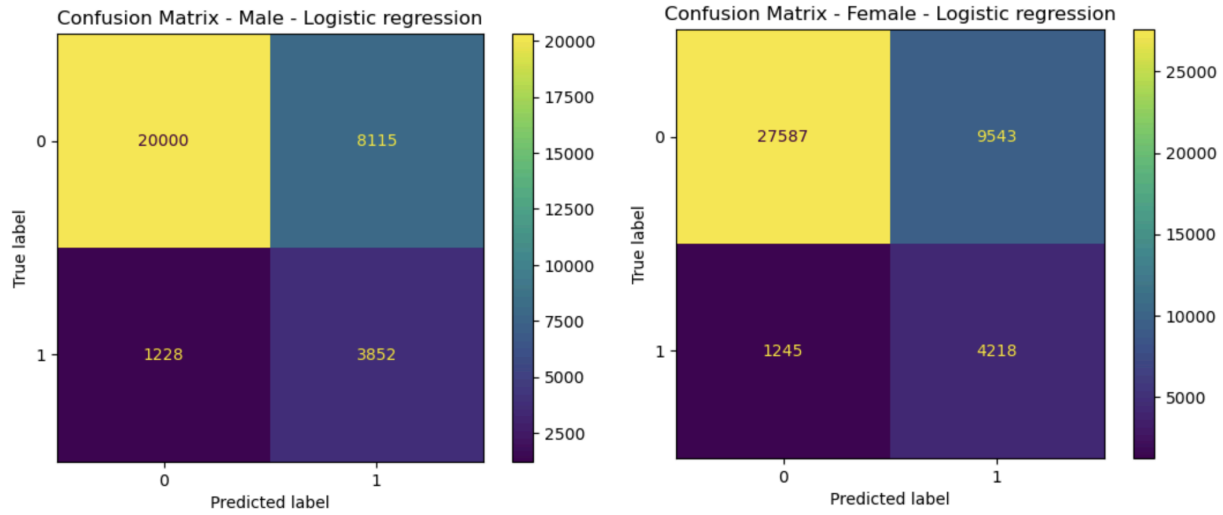
Appendix B. Confusion Matrix for Random Forest



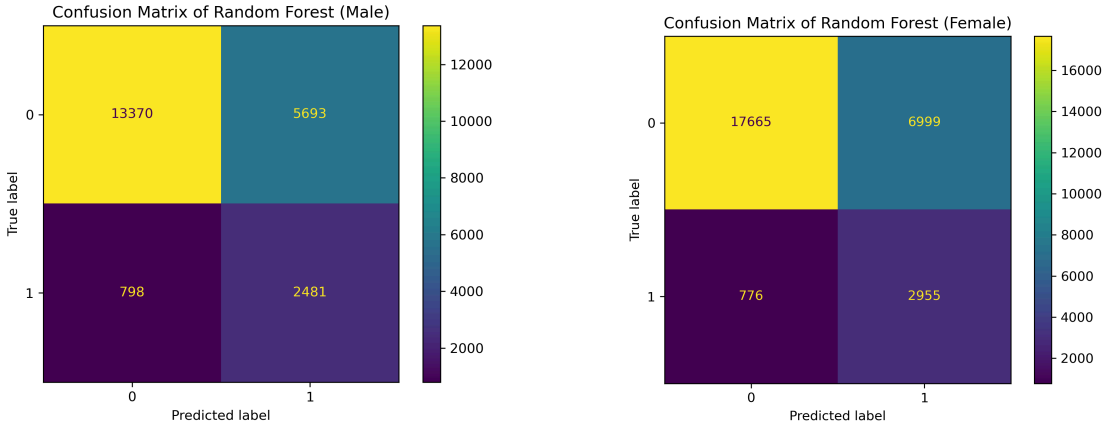
Appendix C. Confusion Matrix for Neural Network



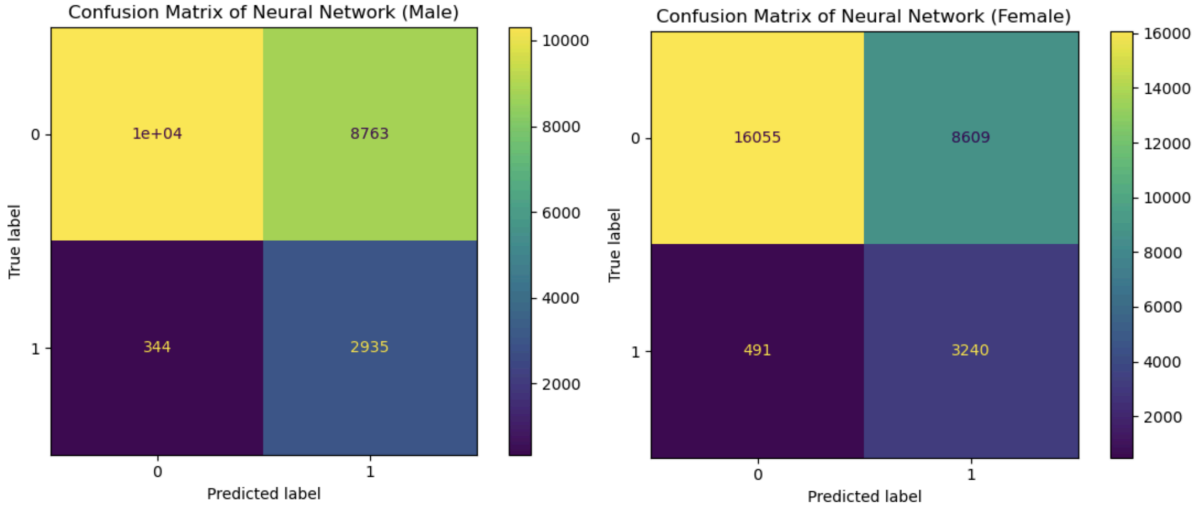
Appendix D. Confusion matrices from male and female subsets for Logistic regression



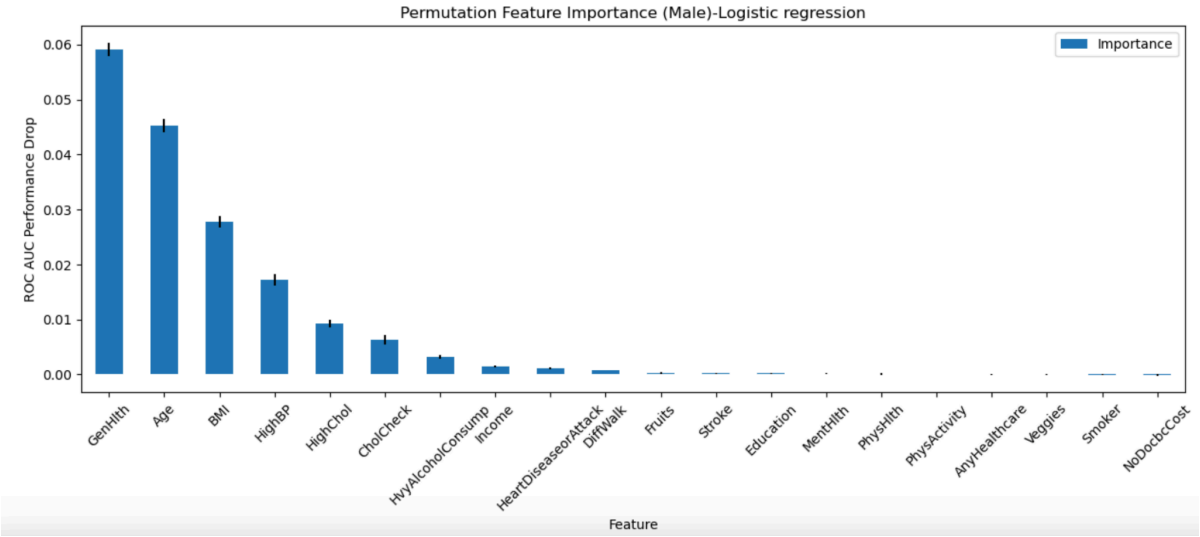
Appendix E. Confusion matrices from male and female subsets for random forest



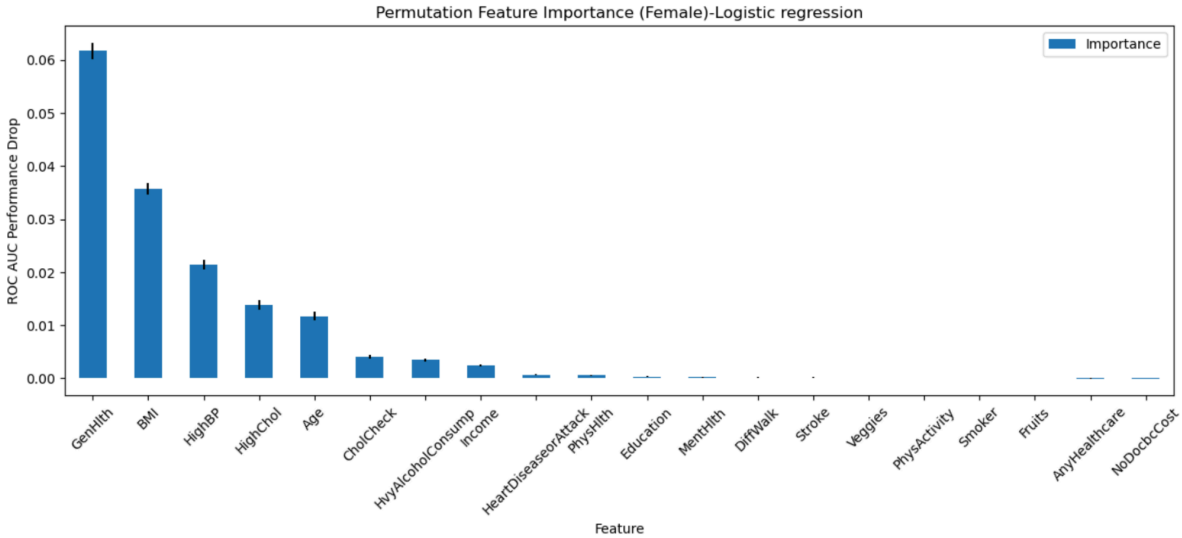
Appendix F. Confusion matrices from male and female subsets for neural network



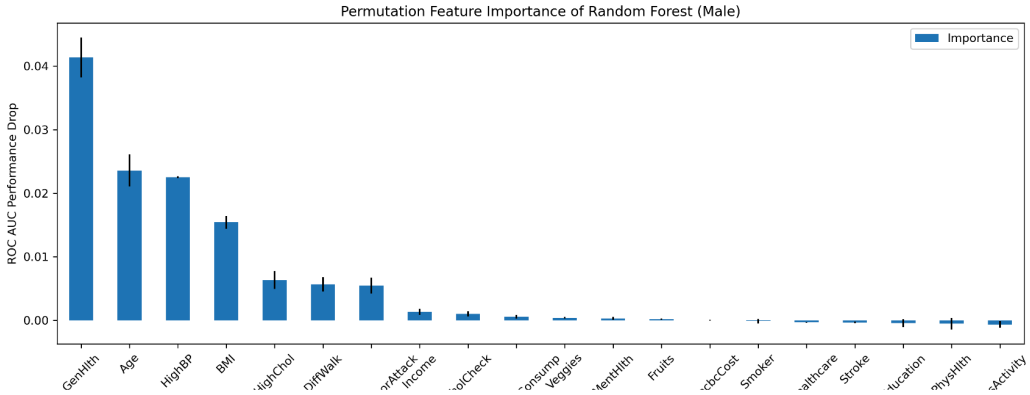
Appendix G. Ranked bar plot of the permutation feature importance score from the male subset for Logistic regression



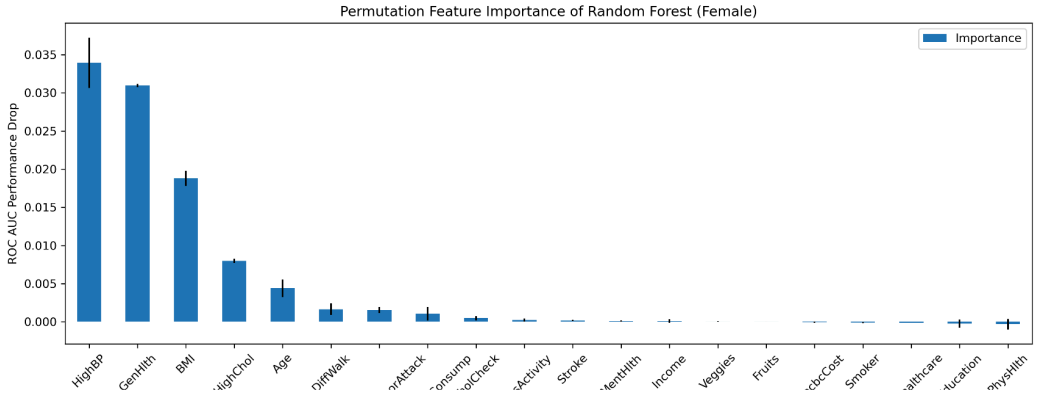
Appendix H. Ranked bar plot of the permutation feature importance score from the female subset for Logistic regression



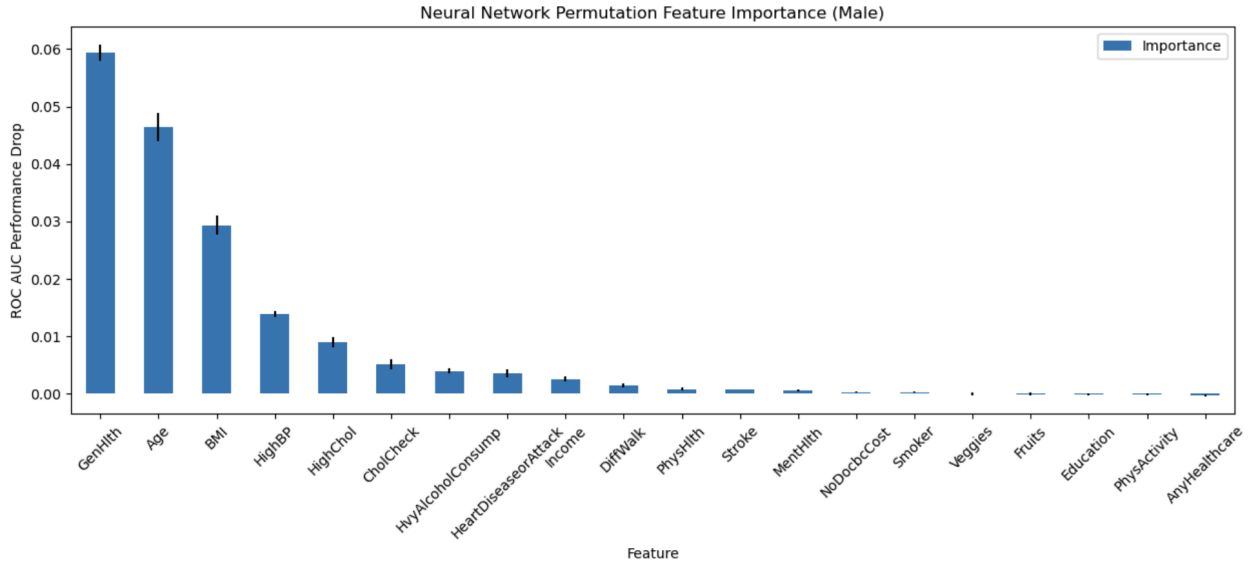
Appendix I. Ranked bar plot of the permutation feature importance score from the male subset for random forest



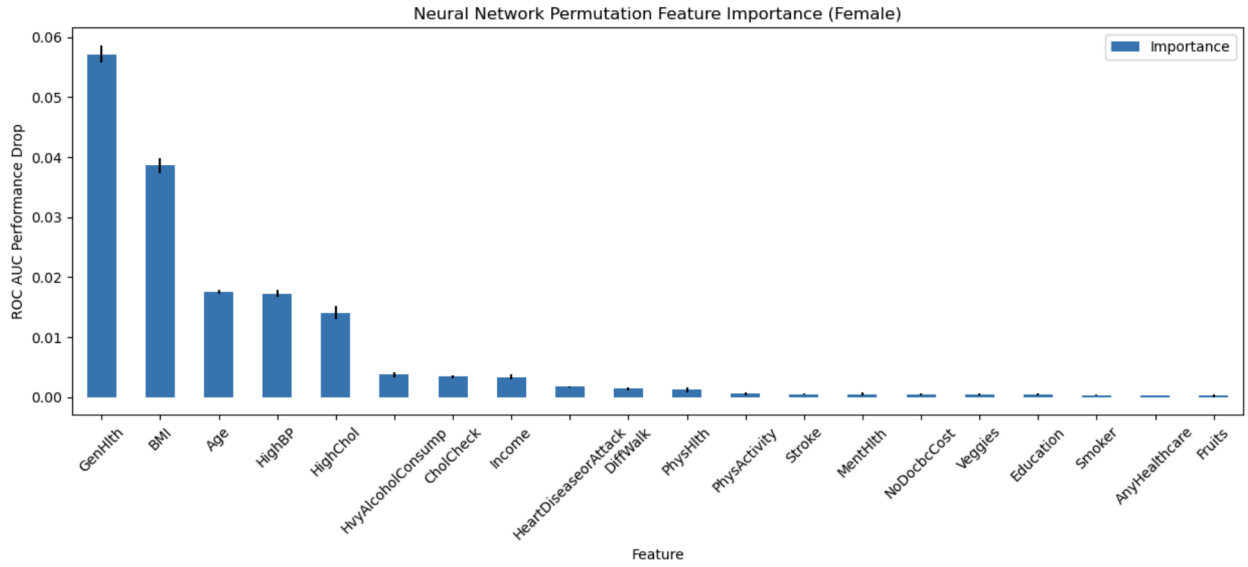
Appendix J. Ranked bar plot of the permutation feature importance score from the female subset for random forest



Appendix K. Ranked bar plot of the permutation feature importance score from the male subset for neural network



Appendix L. Ranked bar plot of the permutation feature importance score from the female subset for neural network



Code

1. Preprocessing

   Jaccard Similarity:

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import statistics

from sklearn.compose import ColumnTransformer

from sklearn.preprocessing import OneHotEncoder

from sklearn.pipeline import Pipeline

from sklearn.preprocessing import StandardScaler

from sklearn.model_selection import train_test_split, RandomizedSearchCV

#from keras.datasets import fashion_mnist

from sklearn.inspection import permutation_importance

from sklearn.metrics import classification_report, confusion_matrix, roc_auc_score,
roc_curve, auc
```

```
from sklearn.metrics import accuracy_score, precision_score, ConfusionMatrixDisplay,
recall_score, f1_score

from sklearn.decomposition import PCA

import plotly.express as px

from sklearn.ensemble import RandomForestClassifier

from sklearn.model_selection import GridSearchCV

from imblearn.pipeline import Pipeline

from imblearn.over_sampling import SMOTE, SMOTENC



dat_df = pd.read_csv("diabetes_binary_health_indicators_BRFSS2015.csv", header=0,
na_values='?', skipinitialspace=True)

#df.columns = ["Column A", "Column B"]

#numerical variable

numVars = ['BMI','GenHlth','MentHlth','PhysHlth','Age','Education','Income']

catVars = [x for x in dat_df.columns if x not in numVars]

dat_df
```

```
#scale numerical variables Z-score transformation

scaler = StandardScaler()

scaled_dat_df = dat_df.copy()

scaled_dat_df[numVars] = scaler.fit_transform(dat_df[numVars])




#Separates Binary and Non-binary columns from dataframe



def create_binary_dataframe(df):
    binary_columns = []
    for column in df.columns:
        if df[column].isin([0, 1]).all():
            binary_columns.append(column)


    if binary_columns:
        return df[binary_columns].copy()
    else:
```

```
        return pd.DataFrame() # Return an empty DataFrame if no binary columns


binary_df_diabetes = create_binary_dataframe(scaled_dat_df)


def create_nonbinary_dataframe(df):
    nonbinary_columns = []
    for column in df.columns:
        if not df[column].isin([0, 1]).all():
            nonbinary_columns.append(column)


    if nonbinary_columns:
        return df[nonbinary_columns].copy()
    else:
        return pd.DataFrame() # Return an empty DataFrame if no binary columns
nonbinary_df_diabetes = create_nonbinary_dataframe(scaled_dat_df)
print(nonbinary_df_diabetes)
```

```
def jaccard_similarity(col1, col2):
    intersection = np.logical_and(col1, col2).sum()
    union = np.logical_or(col1, col2).sum()
    return intersection / union if union > 0 else 0


def jaccard_similarity_matrix(df):
    num_cols = len(df.columns)
    similarity_matrix = pd.DataFrame(index=df.columns, columns=df.columns)


    for i in range(num_cols):
        for j in range(i, num_cols):
            col1 = df.iloc[:, i]
            col2 = df.iloc[:, j]
            similarity = jaccard_similarity(col1, col2)
            similarity_matrix.iloc[i, j] = similarity
            similarity_matrix.iloc[j, i] = similarity # Matrix is symmetric
    return similarity_matrix
```

```
df = pd.DataFrame(binary_df_diabetes)

similarity_matrix = jaccard_similarity_matrix(binary_df_diabetes)

print(similarity_matrix)



similarity_matrix = similarity_matrix[similarity_matrix.columns].astype(float)



#Plotting Jaccard Similarity Matrix

fig, ax = plt.subplots(figsize=(12,12))



sns.heatmap(similarity_matrix, annot=True, cmap="viridis", linewidths=.5)

plt.title("Heatmap of Jaccard Similarities of Binary Features", fontsize = 20)

plt.show()
```

Pearson Correlation Matrix:

```
pearson_correlation_matrix_diabetes = nonbinary_df_diabetes.corr()

print(pearson_correlation_matrix_diabetes)



fig, ax = plt.subplots(figsize=(12,12))



sns.heatmap(pearson_correlation_matrix_diabetes, annot=True, cmap="magma",
linewidths=.5)

plt.title("Pearson Correlation Heatmap of (Scaled) Integer Features", fontsize = 18)

plt.show()
```

2. Q1

2.1. Logistic regression

```
import numpy as np

import pandas as pd

from sklearn.linear_model import LogisticRegression

from sklearn.preprocessing import StandardScaler
```

```
from sklearn.model_selection import train_test_split

from sklearn.metrics import confusion_matrix, roc_curve, auc, accuracy_score,
precision_score, recall_score, f1_score, ConfusionMatrixDisplay

from sklearn.inspection import permutation_importance

from sklearn.utils import class_weight

import matplotlib.pyplot as plt

import matplotlib.ticker as mticker

import seaborn as sns

from wordcloud import WordCloud



# Select features and target

features = ['HighBP', 'HighChol', 'CholCheck', 'BMI', 'Smoker', 'Stroke',
'HeartDiseaseorAttack',

        'PhysActivity', 'Fruits', 'Veggies', 'HvyAlcoholConsump', 'AnyHealthcare',

        'NoDocbcCost', 'GenHlth', 'MentHlth', 'PhysHlth', 'DiffWalk', 'Age',

        'Education', 'Income']  # For male/female subsets (RQ2)
```

```
features_with_sex = ['HighBP', 'HighChol', 'CholCheck', 'BMI', 'Smoker', 'Stroke',
'HeartDiseaseorAttack',
        'PhysActivity', 'Fruits', 'Veggies', 'HvyAlcoholConsump', 'AnyHealthcare',
        'NoDocbcCost', 'GenHlth', 'MentHlth', 'PhysHlth', 'DiffWalk', 'Age',
        'Education', 'Income', 'Sex']  # For RQ1 (includes 'Sex')  # 'Sex' added for RQ1
X = scaled_dat_df[features]
y = scaled_dat_df['Diabetes_binary']
gender = scaled_dat_df['Sex']  # 0 = Female, 1 = Male


# Split data by gender
male_idx = gender == 1
female_idx = gender == 0
X_male, y_male = X[male_idx], y[male_idx]
X_female, y_female = X[female_idx], y[female_idx]


# Scale features
scaler = StandardScaler()
```

```
X_scaled = scaler.fit_transform(X)

X_male_scaled = scaler.fit_transform(X_male)

X_female_scaled = scaler.fit_transform(X_female)



# Define features and target

X_rq1 = scaled_dat_df[features_with_sex]

y = scaled_dat_df['Diabetes_binary']



# 70/30 Train-Test Split (Stratified)

X_train, X_test, y_train, y_test = train_test_split(

    X_rq1, y, test_size=0.3, random_state=42, stratify=y

)



# Scale numeric features (fit only on training data)

scaler = StandardScaler()

X_train_scaled = scaler.fit_transform(X_train)

X_test_scaled = scaler.transform(X_test)
```

```
# Logistic Regression model

log_reg = LogisticRegression(class_weight='balanced', max_iter=1000)

log_reg.fit(X_train_scaled, y_train)



# Predict on test set

y_pred_test = log_reg.predict(X_test_scaled)



# Confusion Matrix and ROC/AUC

cm_overall = confusion_matrix(y_test, y_pred_test)

fpr_overall, tpr_overall, _ = roc_curve(y_test, log_reg.predict_proba(X_test_scaled)[:,
1])

auc_overall = auc(fpr_overall, tpr_overall)



# Odds Ratios

odds_ratios = np.exp(log_reg.coef_[0])

features_rq1 = features_with_sex
```

```
# Calculate odds ratios and 95% confidence intervals

import statsmodels.api as sm



# Refit logistic regression model using statsmodels for detailed summary

X_sm = sm.add_constant(X_rq1_scaled)  # Add intercept

model_sm = sm.Logit(y, X_sm).fit(disp=False)

# ROC Curve for Overall

plt.figure(figsize=(10, 6))

plt.plot(fpr_overall, tpr_overall, label=f'Overall (AUC = {auc_overall:.2f})', color='blue')

plt.plot([0, 1], [0, 1], 'k--')

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('Overall ROC Curves for Logistic Regression')

plt.legend()

plt.show()



# Get odds ratios and confidence intervals
```

```
odds_ratios = np.exp(model_sm.params)

conf = model_sm.conf_int()

conf.columns = ['2.5%', '97.5%']

conf_exp = np.exp(conf)



# Combine into a single table

odds_table = pd.DataFrame({

    'Feature': ['Intercept'] + features_rq1,

    'Odds Ratio': odds_ratios.round(4),

    'CI Lower (2.5%)': conf_exp['2.5%'].round(4),

    'CI Upper (97.5%)': conf_exp['97.5%'].round(4)

})



print("\nOdds Ratios for Logistic Regression (Overall with Sex):")

print(odds_table.to_string(index=False))

# Permutation importance for overall model
```

```
perm_importance_overall = permutation_importance(log_reg, X_rq1_scaled, y,
scoring='roc_auc', random_state=42, n_repeats=10)

importance_overall = perm_importance_overall.importances_mean

std_overall = perm_importance_overall.importances_std

import matplotlib.pyplot as plt

from sklearn.metrics import ConfusionMatrixDisplay



# Overall Confusion Matrix Plot

fig, ax = plt.subplots()

disp_overall = ConfusionMatrixDisplay(confusion_matrix=cm_overall,
display_labels=[0, 1])

disp_overall.plot(ax=ax, values_format='d', cmap='viridis', colorbar=True)

ax.set_title("Confusion Matrix - Overall - Logistic regression")

ax.set_xticklabels(['Predicted Negative', 'Predicted Positive'])

ax.set_yticklabels(['True Negative', 'True Positive'])

plt.show()
```

2.2. Random forest

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import statistics

from sklearn.compose import ColumnTransformer

from sklearn.preprocessing import OneHotEncoder

from sklearn.pipeline import Pipeline

from sklearn.preprocessing import StandardScaler

from sklearn.model_selection import train_test_split

import tensorflow as tf

#from keras.datasets import fashion_mnist

from tensorflow.keras import layers

from sklearn.inspection import permutation_importance
```

```
from sklearn.metrics import classification_report, confusion_matrix, roc_auc_score,
roc_curve, auc, recall_score

from sklearn.metrics import accuracy_score, confusion_matrix, f1_score

from sklearn.decomposition import PCA

import plotly.express as px

from sklearn.ensemble import RandomForestClassifier

from sklearn.model_selection import GridSearchCV



#Overall Data

dat_df = pd.read_csv("diabetes_binary_health_indicators_BRFSS2015.csv", header=0,
na_values='?', skipinitialspace=True)

#df.columns = ["Column A", "Column B"]

#numerical variable

numVars = ['BMI','GenHlth','MentHlth','PhysHlth','Age','Education','Income']

catVars = [x for x in dat_df.columns if x not in numVars]

dat_df = dat_df.dropna()

scaler = StandardScaler()
```

```
scaled_dat_df = dat_df.copy()

scaled_dat_df[numVars] = scaler.fit_transform(dat_df[numVars])

#Male and Female superset

X = scaled_dat_df.drop('Diabetes_binary', axis=1)

y = scaled_dat_df['Diabetes_binary']



#Overall Random Forest Model

rf_model_grid_combined = rf_model_grid.fit(X_train, y_train)

y_pred_combined = rf_model_grid_combined.predict(X_test)



accuracy_combined = accuracy_score(y_test, y_pred_combined)

print(accuracy_combined)



#Important combined features

from sklearn.inspection import permutation_importance

from sklearn.metrics import get_scorer
```

```
from scikeras.wrappers import KerasClassifier


def scorer(model, X, y):
    y_pred = model.predict(X)
    return roc_auc_score(y, y_pred)


perm = permutation_importance(rf_model_grid_combined, X_test, y_test, n_repeats=3,
                              random_state=0, scoring=scorer)


importance = pd.DataFrame({'Feature': X_train.columns,
                           'Importance': perm["importances_mean"],
                           'Standard Deviation': perm["importances_std"]})


combined_importance = importance.sort_values('Importance',ascending=False)


print(importance[['Feature','Importance','Standard Deviation']].to_string(index=False))
```

```
#Graphing Feature Importance

combined_importance.plot(figsize=(15,5),x='Feature',y='Importance',kind="bar",yerr='St
andard Deviation')

plt.title("Permutation Feature Importance (Overall)")

plt.ylabel("ROC AUC Performance Drop")

plt.xticks(rotation=45)

plt.savefig("Permutation Feature Importance of Random Forest (Overall)", dpi = 300)

plt.show()

from sklearn import metrics

actual_combined = np.random.binomial(1,.9,size = 1000)

predicted_combined = np.random.binomial(1,.9,size = 1000)



confusion_matrix_combined = metrics.confusion_matrix(y_test, y_pred)



cm_display_combined = metrics.ConfusionMatrixDisplay(confusion_matrix =
confusion_matrix_combined, display_labels = [0, 1])
```

```
cm_display_combined.plot()

plt.title("Confusion Matrix of Random Forest (Overall)")

plt.savefig("Confusion Matrix of Random Forest (Overall)", dpi = 300)

plt.show()
```

2.3. Neural network

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import statistics

from sklearn.compose import ColumnTransformer

from sklearn.preprocessing import OneHotEncoder

from sklearn.pipeline import Pipeline

from sklearn.preprocessing import StandardScaler

from sklearn.model_selection import train_test_split
```

```
import tensorflow as tf

from tensorflow.keras import layers

from sklearn.inspection import permutation_importance

from sklearn.metrics import classification_report, confusion_matrix, roc_auc_score,
roc_curve, auc

from sklearn.metrics import accuracy_score

from sklearn.decomposition import PCA

import plotly.express as px

dat_df = pd.read_csv("diabetes_binary_health_indicators_BRFSS2015.csv", header=0,
na_values='?', skipinitialspace=True)



#numerical variables

numVars = ['BMI','GenHlth','MentHlth','PhysHlth','Age','Education','Income']



#categorical variables

catVars = [x for x in dat_df.columns if x not in numVars]

dat_df = dat_df.dropna()
```

```
#scale numerical variables Z-score transformation

scaler = StandardScaler()

scaled_dat_df = dat_df.copy()

scaled_dat_df[numVars] = scaler.fit_transform(dat_df[numVars])

#Male and Female superset

X = scaled_dat_df.drop(['Diabetes_binary','Sex'], axis=1)

y = scaled_dat_df['Diabetes_binary']

#Overall Neural Network Model

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=123)

mf_model = tf.keras.Sequential()

mf_model.add(tf.keras.layers.Input(shape=(X_train.shape[1],)))

mf_model.add(tf.keras.layers.Dense(16, activation='relu'))

mf_model.add(tf.keras.layers.Dense(8, activation='relu'))

mf_model.add(tf.keras.layers.Dense(1, activation='sigmoid'))

#Class weight for Overall Neural Network Model

from sklearn.utils import class_weight
```

```
y_train = np.array(y_train).astype('int32').flatten()



weights =
class_weight.compute_class_weight(class_weight='balanced',classes=np.array([0,1]),y=y
_train)

cw = {0: weights[0], 1: weights[1]}

#Overall Neural Network Model with balanced weight

mf_model.compile(optimizer='adam', loss='binary_crossentropy',

                 metrics=['accuracy'])

mf_model.fit(X_train,y_train,epochs=20, class_weight=cw)

#Overall Neural Network Confusion Matrix

import matplotlib.pyplot as plt

import numpy

from sklearn import metrics



y_probs = mf_model.predict(X_test)

y_pred = (y_probs >= 0.5).astype(int)
```

```
confusion_matrix = metrics.confusion_matrix(y_test, y_pred)

cm_display = metrics.ConfusionMatrixDisplay(confusion_matrix = confusion_matrix,
display_labels = [0, 1])

cm_display.plot()

plt.title("Confusion Matrix of Neural Network (Overall)")

plt.show()

# Overall Neural Network Performance Metrics

from sklearn.metrics import accuracy_score, recall_score, f1_score, roc_curve,
confusion_matrix

y_probs = mf_model.predict(X_test)

y_pred = (y_probs >= 0.5).astype(int)



cm_mf = confusion_matrix(y_test, y_pred)

print("Confusion Matrix (male and female): ")

print(cm_mf)

accuracy = accuracy_score(y_test,y_pred)

sensitivity = recall_score(y_test,y_pred, pos_label=1)
```

```
specificity = recall_score(y_test, y_pred, pos_label=0)

f1 = f1_score(y_test, y_pred)

print("Accuracy:", accuracy)

print("Sensitivity:", sensitivity)

print("Specificity:", specificity)

print("F1:", f1)

# Overall Neural Network ROC plot function

def plot_roc_overall(y_truth, y_prob, ax):

    FPR, TPR, thresholds = roc_curve(y_truth, y_prob)

    AUC = np.trapz(TPR, FPR)

    ax.step(FPR, TPR, linewidth=2,label='Overall (AUC = ' + str(round(AUC,2)) + ')')

    ax.plot([0,1],[0,1],'--', color = 'black')

    fs = 10

    ax.set_xlabel('False Positive Rate', fontsize=fs)

    ax.set_ylabel('True Positive Rate', fontsize=fs)

    ax.tick_params(axis='both', labelsize=fs)

    ax.set_title('ROC Curves for Neural Network')
```

```
    ax.legend(loc='upper left',fontsize=fs)

# Overall ROC plot for Neural Network Model

fig, ax = plt.subplots(figsize=(8,6))

plot_roc_overall(y_test, y_probs, ax)

plt.show()

#Important features for both male and female (Overall)

from sklearn.inspection import permutation_importance

from sklearn.metrics import get_scorer

from scikeras.wrappers import KerasClassifier



def scorer(model, X, y):

    y_pred = model.predict(X)

    return roc_auc_score(y, y_pred)

perm = permutation_importance(mf_model, X_test, y_test, n_repeats=3,

                              random_state=0, scoring=scorer)

importance = pd.DataFrame({'Feature': X_train.columns,

                           'Importance': perm["importances_mean"],
```

```
'Standard Deviation': perm["importances_std"]})

mf_importance = importance.sort_values('Importance',ascending=False)

print(mf_importance[['Feature','Importance','Standard
Deviation']].to_string(index=False))

# Feature importance plot (Overall)

mf_importance.plot(figsize=(15,5),x='Feature',y='Importance',kind="bar",yerr='Standard
Deviation')

plt.title("Neural Network Permutation Feature Importance (Male and Female)")

plt.ylabel("ROC AUC Performance Drop")

plt.xticks(rotation=45)

plt.show()
```

3. Q2

    3.1. Logistic regression

```
# Logistic Regression - Male

X_train_m, X_test_m, y_train_m, y_test_m = train_test_split(X_male_scaled, y_male,
test_size=0.3, random_state=42)

y_train_m = np.array(y_train_m).astype('int32').flatten()

weights_male = class_weight.compute_class_weight(class_weight='balanced',
classes=np.array([0, 1]), y=y_train_m)

cw_male = {0: weights_male[0], 1: weights_male[1]}

log_reg_male = LogisticRegression(class_weight=cw_male, max_iter=1000)

log_reg_male.fit(X_train_m, y_train_m)

y_pred_m = log_reg_male.predict(X_test_m)

cm_m = confusion_matrix(y_test_m, y_pred_m)

fpr_m, tpr_m, _ = roc_curve(y_test_m, log_reg_male.predict_proba(X_test_m)[:, 1])

auc_m = auc(fpr_m, tpr_m)



# Logistic Regression - Female

X_train_f, X_test_f, y_train_f, y_test_f = train_test_split(X_female_scaled, y_female,
test_size=0.3, random_state=42)
```

```
y_train_f = np.array(y_train_f).astype('int32').flatten()

weights_female = class_weight.compute_class_weight(class_weight='balanced',
classes=np.array([0, 1]), y=y_train_f)

cw_female = {0: weights_female[0], 1: weights_female[1]}

log_reg_female = LogisticRegression(class_weight=cw_female, max_iter=1000)

log_reg_female.fit(X_train_f, y_train_f)

y_pred_f = log_reg_female.predict(X_test_f)

cm_f = confusion_matrix(y_test_f, y_pred_f)

fpr_f, tpr_f, _ = roc_curve(y_test_f, log_reg_female.predict_proba(X_test_f)[:, 1])

auc_f = auc(fpr_f, tpr_f)



# Permutation Importance

perm_importance_m = permutation_importance(log_reg_male, X_test_m, y_test_m,
scoring='roc_auc', random_state=42, n_repeats=10)

perm_importance_f = permutation_importance(log_reg_female, X_test_f, y_test_f,
scoring='roc_auc', random_state=42, n_repeats=10)
```

```
importance_male = perm_importance_m.importances_mean

importance_female = perm_importance_f.importances_mean

std_male = perm_importance_m.importances_std

std_female = perm_importance_f.importances_std



# Raw importance difference calculation

importance_diff = importance_male - importance_female

importance_diff_df = pd.DataFrame({

    'Feature': features,

    'Importance_Male': importance_male,

    'Importance_Female': importance_female,

    'Difference (Male - Female)': importance_diff

})

print("\nRaw Importance Differences (Male - Female):")

print(importance_diff_df.sort_values(by='Difference (Male - Female)',

ascending=False).to_string(index=False))
```

```
# Permutation Feature Importance Charts

# Sort features and importance by male importance

sort_indices = np.argsort(importance_male)[::-1]

features_sorted = [features[i] for i in sort_indices]

importance_male_sorted = importance_male[sort_indices]

std_male_sorted = std_male[sort_indices]

#std_female_sorted = std_female[sort_indices]

sort_indices = np.argsort(importance_female)[::-1]

features_sorted = [features[i] for i in sort_indices]

importance_female_sorted = importance_female[sort_indices]

std_female_sorted = std_female[sort_indices]

# Male Confusion Matrix Plot

fig, ax = plt.subplots()

disp_male = ConfusionMatrixDisplay(confusion_matrix=cm_m, display_labels=[0, 1])

disp_male.plot(ax=ax, values_format='d', cmap='viridis', colorbar=True)

ax.set_title("Confusion Matrix - Male - Logistic regression")

ax.set_xticklabels(['Predicted Negative', 'Predicted Positive'])
```

```
ax.set_yticklabels(['True Negative', 'True Positive'])

plt.show()



# Female Confusion Matrix Plot

fig, ax = plt.subplots()

disp_female = ConfusionMatrixDisplay(confusion_matrix=cm_f, display_labels=[0, 1])

disp_female.plot(ax=ax, values_format='d', cmap='viridis', colorbar=True)

ax.set_title("Confusion Matrix - Female - Logistic regression")

ax.set_xticklabels(['Predicted Negative', 'Predicted Positive'])

ax.set_yticklabels(['True Negative', 'True Positive'])

plt.show()

# Combined ROC Curve for Overall, Male, and Female

plt.figure(figsize=(10, 6))

plt.plot(fpr_overall, tpr_overall, label=f'Overall (AUC = {auc_overall:.2f})', color='blue')

plt.plot(fpr_m, tpr_m, label=f'Male (AUC = {auc_m:.2f})', color='green')

plt.plot(fpr_f, tpr_f, label=f'Female (AUC = {auc_f:.2f})', color='red')

plt.plot([0, 1], [0, 1], 'k--')
```

```
plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('Combined ROC Curves for Logistic Regression')

plt.legend()

plt.show()

# Plot for Male

df_male = pd.DataFrame({'Feature': features_sorted, 'Importance':
importance_male_sorted, 'Standard Deviation': std_male_sorted})

df_male.plot(figsize=(15, 5), x='Feature', y='Importance', kind="bar", yerr='Standard
Deviation')

plt.title("Permutation Feature Importance (Male)-Logistic regression")

plt.ylabel("ROC AUC Performance Drop")

plt.xticks(rotation=45)

plt.show()

# Plot for Female

df_female = pd.DataFrame({'Feature': features_sorted, 'Importance':
importance_female_sorted, 'Standard Deviation': std_female_sorted})
```

```
df_female.plot(figsize=(15, 5), x='Feature', y='Importance', kind="bar", yerr='Standard Deviation')

plt.title("Permutation Feature Importance (Female)-Logistic regression")

plt.ylabel("ROC AUC Performance Drop")

plt.xticks(rotation=45)

plt.show()

# Step 1: Merge Male and Female Importance DataFrames

df_diff = pd.merge(df_male, df_female, on='Feature', suffixes=('_Male', '_Female'))



# Step 2: Calculate difference (Male - Female)

df_diff['Difference'] = df_diff['Importance_Male'] - df_diff['Importance_Female']



# Step 3: Sort by absolute difference

# Step 3: Sort by raw difference (positive to negative)

df_diff_sorted = df_diff.sort_values(by='Difference', ascending=False)
```

```
# Step 4: Plot the difference

plt.figure(figsize=(10, 6))

plt.barh(df_diff_sorted['Feature'], df_diff_sorted['Difference'], color='skyblue')

plt.xlabel("Difference in Importance (Male - Female)")

plt.title("Feature Importance Difference Between Males and Females (Logistic
regression)")

plt.axvline(x=0, color='gray', linestyle='--')

plt.gca().invert_yaxis()  # Show largest difference at the top

plt.tight_layout()

plt.show()

# Calculate metrics

def calculate_metrics(y_true, y_pred):

    cm = confusion_matrix(y_true, y_pred)

    tn, fp, fn, tp = cm.ravel()

    accuracy = accuracy_score(y_true, y_pred)

    precision = precision_score(y_true, y_pred)

    recall = recall_score(y_true, y_pred)
```

```
    specificity = tn / (tn + fp) if (tn + fp) > 0 else 0
    f1 = f1_score(y_true, y_pred)
    sensitivity = recall  # alias for clarity
    return {
        'Accuracy': accuracy,
        'Precision': precision,
        'Recall': recall,
        'Sensitivity': sensitivity,
        'Specificity': specificity,
        'F1': f1
    }


# Metrics
metrics_overall = calculate_metrics(y, y_pred_overall)
metrics_male = calculate_metrics(y_test_m, y_pred_m)
metrics_female = calculate_metrics(y_test_f, y_pred_f)
```

```
print("Overall Metrics:")

for k, v in metrics_overall.items():

    print(f"{k}: {v:.4f}")



print("\nMale Metrics:")

for k, v in metrics_male.items():

    print(f"{k}: {v:.4f}")



print("\nFemale Metrics:")

for k, v in metrics_female.items():

    print(f"{k}: {v:.4f}")
```

3.2. Random forest

```
#Male and Female

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=123)
```

```
#Male subset

male_df = scaled_dat_df[scaled_dat_df['Sex'] == 1.0].copy()

#Female subset

female_df = scaled_dat_df[scaled_dat_df['Sex'] == 0.0].copy()

from sklearn.utils import resample

female_downsampled = resample(female_df, replace=False,

                    n_samples=len(male_df),

                    random_state=123)

female_downsampled

#Male

X_male = male_df.drop(['Diabetes_binary','Sex'], axis=1)

y_male = male_df['Diabetes_binary']



#Female

X_female = female_df.drop(['Diabetes_binary','Sex'], axis=1)

y_female = female_df['Diabetes_binary']
```

```
#Female downsampled

X_female_ds = female_downsampled.drop(['Diabetes_binary','Sex'], axis=1)

y_female_ds = female_downsampled['Diabetes_binary']



#Male

X_train_male, X_test_male, y_train_male, y_test_male = train_test_split(X_male,
y_male, test_size=0.2, random_state=123)



#Female

X_train_female, X_test_female, y_train_female, y_test_female =
train_test_split(X_female, y_female, test_size=0.2, random_state=123)



#Female downsampled

X_train_female_ds, X_test_female_ds, y_train_female_ds, y_test_female_ds =
train_test_split(X_female_ds, y_female_ds, test_size=0.2, random_state=123)



#Male
```

```
X_train_male, X_test_male, y_train_male, y_test_male = train_test_split(X_male,
y_male, test_size=0.2, random_state=123)


#Female

X_train_female, X_test_female, y_train_female, y_test_female =
train_test_split(X_female, y_female, test_size=0.2, random_state=123)


#Female downsampled

X_train_female_ds, X_test_female_ds, y_train_female_ds, y_test_female_ds =
train_test_split(X_female_ds, y_female_ds, test_size=0.2, random_state=123)


#GridSearch

grid_search = GridSearchCV(RandomForestClassifier(), param_dist, cv=5, n_jobs=-1,
scoring = 'recall')

grid_search.fit(X_train, y_train)

print(grid_search.best_params_)

#Random Forest

rf_model_grid = RandomForestClassifier(
```

```
    n_estimators=200,
    max_depth=5,
    min_samples_split=20,
    max_features='log2',
    class_weight='balanced'
)
#Male Subset Random Forest
rf_model_grid_male = rf_model_grid.fit(X_train_male, y_train_male)
y_pred_proba_male = rf_model_grid_male.predict_proba(X_test_male)[:, 1]
y_pred_male = rf_model_grid_male.predict(X_test_male)
accuracy_male = accuracy_score(y_test_male, y_pred_male)
print(accuracy_male)
recall_male = recall_score(y_test_male, y_pred_male, average='binary')
print(recall_male)
specificity_male = recall_score(y_test_male, y_pred_male, average='binary', pos_label =
0)
print(specificity_male)
```

```
f1_score_male = f1_score(y_test_male, y_pred_male, average='binary')

print(f1_score_male)

auc_male = roc_auc_score(y_test_male, y_pred_proba_male)

print(auc_male)

fpr_male, tpr_male, thresholds_male = roc_curve(y_test_male, y_pred_proba_male)

roc_auc_male = auc(fpr_male, tpr_male)

plt.figure(figsize=(8, 6))

plt.plot(fpr_male, tpr_male, color='orange', lw=2, label=f'ROC curve (area =
{roc_auc_male:.2f})')

plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')  # Random classifier

plt.xlim([0.0, 1.0])

plt.ylim([0.0, 1.05])

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('ROC Curve Random Forest (Male)')

plt.legend(loc="lower right")

plt.savefig('ROC Curve for Random Forest (Male)', dpi = 300)
```

```
plt.show()

from sklearn.datasets import make_classification

from sklearn.inspection import permutation_importance

import shap

from sklearn.inspection import permutation_importance

from sklearn.metrics import get_scorer

from scikeras.wrappers import KerasClassifier



def scorer(model, X, y):

    y_pred = model.predict(X)

    return roc_auc_score(y, y_pred)



perm = permutation_importance(rf_model_grid_male, X_test_male, y_test_male,

n_repeats=3,

                              random_state=0, scoring=scorer)



importance = pd.DataFrame({'Feature': X_train_male.columns,
```

```
                    'Importance': perm["importances_mean"],

                    'Standard Deviation': perm["importances_std"]})


male_importance = importance.sort_values('Importance',ascending=False)


print(male_importance[['Feature','Importance','Standard
Deviation']].to_string(index=False))

male_importance.plot(figsize=(15,5),x='Feature',y='Importance',kind="bar",yerr='Standar
d Deviation')

plt.title("Permutation Feature Importance of Random Forest (Male)")

plt.ylabel("ROC AUC Performance Drop")

plt.xticks(rotation=45)

plt.savefig("Permutation Feature Importance of Random Forest (Male)", dpi = 300)

plt.show()

#Female Subset Random Forest

rf_model_grid_female = rf_model_grid.fit(X_train_female, y_train_female)

y_pred_proba_female = rf_model_grid_female.predict_proba(X_test_female)[:, 1]
```

```
y_pred_female = rf_model_grid_female.predict(X_test_female)

accuracy_female = accuracy_score(y_test_female, y_pred_female)

print(accuracy_female)



recall_female = recall_score(y_test_female, y_pred_female, average='binary')

print(recall_female)

specificity_female = recall_score(y_test_female, y_pred_female, average='binary',
pos_label = 0)

print(specificity_female)

f1_score_female = f1_score(y_test_female, y_pred_female, average='binary')

print(f1_score_female)

auc_female = roc_auc_score(y_test_female, y_pred_proba_female)

print(auc_female)

fpr_female, tpr_female, thresholds_female = roc_curve(y_test_female,
y_pred_proba_female)

roc_auc_female = auc(fpr_female, tpr_female)

plt.figure(figsize=(8, 6))
```

```
plt.plot(fpr_female, tpr_female, color='green', lw=2, label=f'ROC curve (area =
{roc_auc_female:.2f})')

plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')  # Random classifier

plt.xlim([0.0, 1.0])

plt.ylim([0.0, 1.05])

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('ROC Curve for Random Forest (Female)')

plt.legend(loc="lower right")

plt.savefig('ROC Curve for Random Forest (Female)', dpi = 300)

plt.show()

#Important female features

from sklearn.inspection import permutation_importance

from sklearn.metrics import get_scorer

from scikeras.wrappers import KerasClassifier



def scorer(model, X, y):
```

```
    y_pred = model.predict(X)

    return roc_auc_score(y, y_pred)


perm = permutation_importance(rf_model_grid_female, X_test_female, y_test_female,
n_repeats=3,
                              random_state=0, scoring=scorer)


importance = pd.DataFrame({'Feature': X_train_female.columns,
                           'Importance': perm["importances_mean"],
                           'Standard Deviation': perm["importances_std"]})


female_importance = importance.sort_values('Importance',ascending=False)


print(importance[['Feature','Importance','Standard Deviation']].to_string(index=False))

female_importance.plot(figsize=(15,5),x='Feature',y='Importance',kind="bar",yerr='Stand
ard Deviation')

plt.title("Permutation Feature Importance of Random Forest (Female)")
```

```
plt.ylabel("ROC AUC Performance Drop")

plt.xticks(rotation=45)

plt.savefig("Permutation Feature Importance of Random Forest (Female)", dpi = 300)

plt.show()



#Combining Male and Female Sets

sorted_df_male_importance = male_importance.sort_values(by='Feature')

sorted_df_female_importance = female_importance.sort_values(by='Feature')

merged_df_importance = pd.merge(left=male_importance, right=female_importance,
on='Feature', how='inner')

difference_importance_df = merged_df_importance.drop(columns=['Standard
Deviation_x','Standard Deviation_y'], axis=1, inplace=False)

difference_importance_df.head()

difference_importance_df.columns = ['Feature', 'Male', 'Female']

difference_importance_df.head()
```

```
#Difference Between Male and Female Feature Importances

import matplotlib.pyplot as plt

import pandas as pd



# Feature importance data



df = pd.DataFrame(difference_importance_df)

df["Difference"] = df["Male"] - df["Female"]



# Sort by raw difference (not by absolute value)

df_sorted = df.sort_values("Difference", ascending=False)



# Plot

plt.figure(figsize=(10, 6))

bars = plt.barh(df_sorted["Feature"], df_sorted["Difference"], color='skyblue')
```

```
plt.xlabel("Difference in Importance (Male - Female)")

plt.title("Permutation Feature Importance Difference Between Males and Females
(Random Forest)")

plt.axvline(x=0, color='gray', linestyle='--')

plt.gca().invert_yaxis()  # Highest difference at the top

plt.tight_layout()

plt.savefig("Permutation Feature Importance Difference Between Males and Females
(Random Forest)", dpi = 300)

plt.show()

y_pred_proba_combined = rf_model_grid_combined.predict_proba(X_test)[:, 1]

y_pred = rf_model_grid_combined.predict(X_test)

fpr_combined, tpr_combined, thresholds_combined = roc_curve(y_test,
y_pred_proba_combined)

roc_auc_combined = auc(fpr_combined, tpr_combined)



#Graphing ROC Curves
```

```
plt.figure(figsize=(8, 6))

plt.plot(fpr_male, tpr_male, color='orange', lw=2, label=f'ROC curve (male) (area =
{roc_auc_male:.2f})')

plt.plot(fpr_combined, tpr_combined, color='blue', lw=2, label=f'ROC curve (overall)
(area = {roc_auc_combined:.2f})')

plt.plot(fpr_female, tpr_female, color='green', lw=2, label=f'ROC curve (female) (area =
{roc_auc_female:.2f})')

plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')  # Random classifier

plt.xlim([0.0, 1.0])

plt.ylim([0.0, 1.05])

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('ROC Curve for Random Forest (Overall)')

plt.legend(loc="lower right")

plt.savefig('ROC Curve for Random Forest (Overall)', dpi = 300)

plt.show()
```

```
#Confusion Matrix Female

actual_combined = np.random.binomial(1,.9,size = 1000)

predicted_combined = np.random.binomial(1,.9,size = 1000)


confusion_matrix_female = metrics.confusion_matrix(y_test_female, y_pred_female)


cm_display_female = metrics.ConfusionMatrixDisplay(confusion_matrix =
confusion_matrix_female, display_labels = [0, 1])


cm_display_female.plot()

plt.title("Confusion Matrix of Random Forest (Female)")

plt.savefig("Confusion Matrix of Random Forest (Female)", dpi = 300)

plt.show()


#Confusion Matrix Male
```

```
from sklearn import metrics

actual_male = np.random.binomial(1,.9,size = 1000)

predicted_male = np.random.binomial(1,.9,size = 1000)



confusion_matrix_male = metrics.confusion_matrix(y_test_male, y_pred_male)



cm_display_male = metrics.ConfusionMatrixDisplay(confusion_matrix =

confusion_matrix_male, display_labels = [0, 1])



cm_display_male.plot()

plt.title("Confusion Matrix of Random Forest (Male)")

plt.savefig("Confusion Matrix of Random Forest (Male)", dpi = 300)

plt.show()
```

3.3. Neural network

```
#Male subset
```

```
male_df = scaled_dat_df[scaled_dat_df['Sex'] == 1.0].copy()

#Female subset

female_df = scaled_dat_df[scaled_dat_df['Sex'] == 0.0].copy()

#Male

X_male = male_df.drop(['Diabetes_binary','Sex'], axis=1)

y_male = male_df['Diabetes_binary']

#Female

X_female = female_df.drop(['Diabetes_binary','Sex'], axis=1)

y_female = female_df['Diabetes_binary']

#Male

X_train_male, X_test_male, y_train_male, y_test_male = train_test_split(X_male,
y_male, test_size=0.2, random_state=123)



#Female

X_train_female, X_test_female, y_train_female, y_test_female =
train_test_split(X_female, y_female, test_size=0.2, random_state=123)

#Male model
```

```
male_model = tf.keras.Sequential()

male_model.add(tf.keras.layers.Input(shape=(X_train_male.shape[1],)))

male_model.add(tf.keras.layers.Dense(16, activation='relu'))

male_model.add(tf.keras.layers.Dense(8, activation='relu'))

male_model.add(tf.keras.layers.Dense(1, activation='sigmoid'))

#Female model

female_model = tf.keras.Sequential()

female_model.add(tf.keras.layers.Input(shape=(X_train_female.shape[1],)))

female_model.add(tf.keras.layers.Dense(16, activation='relu'))

female_model.add(tf.keras.layers.Dense(8, activation='relu'))

female_model.add(tf.keras.layers.Dense(1, activation='sigmoid'))

#Female model (balanced weights)

from sklearn.utils import class_weight

y_train_female = np.array(y_train_female).astype('int32').flatten()

weights_female =
class_weight.compute_class_weight(class_weight='balanced',classes=np.array([0,1]),y=y
_train_female)
```

```
cw_female = {0: weights_female[0], 1: weights_female[1]}

cw_female = {0: 0.5, 1: 5}

#Compile the female model with balanced weights

female_model.compile(optimizer='adam', loss='binary_crossentropy',

                metrics=['accuracy'])

female_model.fit(X_train_female,y_train_female,epochs=20,class_weight=cw_female)

# Female Neural Network Performance Metrics

from sklearn.metrics import accuracy_score, recall_score, f1_score, roc_curve

y_probs_female = female_model.predict(X_test_female)

y_pred_female = (y_probs_female >= 0.5).astype(int)



cm_female = confusion_matrix(y_test_female,y_pred_female)

print("Confusion Matrix (female): ")

print(cm_female)

accuracy_female = accuracy_score(y_test_female,y_pred_female)

sensitivity_female = recall_score(y_test_female,y_pred_female, pos_label=1)

specificity_female = recall_score(y_test_female, y_pred_female, pos_label=0)
```

```
f1_female = f1_score(y_test_female, y_pred_female)

print("Accuracy:", accuracy_female)

print("Sensitivity:", sensitivity_female)

print("Specificity:", specificity_female)

print("F1:", f1_female)

# Female Neural Network Confusion Matrix

import matplotlib.pyplot as plt

import numpy

from sklearn import metrics

y_probs_female = female_model.predict(X_test_female)

y_pred_female = (y_probs_female >= 0.5).astype(int)

confusion_matrix_female = metrics.confusion_matrix(y_test_female, y_pred_female)

plt.figure(figsize=(6,5))

cm_display_female = metrics.ConfusionMatrixDisplay(confusion_matrix =
confusion_matrix_female, display_labels = [0, 1])

cm_display_female.plot(colorbar=False)

plt.title("Confusion Matrix of Neural Network (Female)")
```

```
plt.tight_layout()

plt.show()

# ROC curve for female model

fig, ax = plt.subplots(figsize=(8,6))

plot_roc(y_test_female, y_probs_female, ax)

plt.show()

#Important female features

from sklearn.inspection import permutation_importance

from sklearn.metrics import get_scorer

from scikeras.wrappers import KerasClassifier



def scorer(model, X, y):

    y_pred = model.predict(X)

    return roc_auc_score(y, y_pred)



perm = permutation_importance(female_model, X_test_female, y_test_female,

n_repeats=3,
```

```
                    random_state=0, scoring=scorer)


importance = pd.DataFrame({'Feature': X_train_female.columns,
                    'Importance': perm["importances_mean"],
                    'Standard Deviation': perm["importances_std"]})


female_importance = importance.sort_values('Importance',ascending=False)
# Plot of important female features
female_importance.plot(figsize=(15,5),x='Feature',y='Importance',kind="bar",yerr='Stand
ard Deviation')
plt.title("Neural Network Permutation Feature Importance (Female)")
plt.ylabel("ROC AUC Performance Drop")
plt.xticks(rotation=45)
plt.show()
# Female Neural Network ROC plot function
def plot_roc_female(y_truth, y_prob, ax):
    FPR, TPR, thresholds = roc_curve(y_truth, y_prob)
```

```
    AUC = np.trapz(TPR, FPR)

    ax.step(FPR, TPR, linewidth=2,label='Female (AUC = ' + str(round(AUC,2)) + ')')

    ax.plot([0,1],[0,1],'--', color = 'black')



    fs = 10

    ax.set_xlabel('False Positive Rate', fontsize=fs)

    ax.set_ylabel('True Positive Rate', fontsize=fs)

    ax.tick_params(axis='both', labelsize=fs)




    ax.set_title('ROC Curves for Neural Network')

    ax.legend(loc='upper left',fontsize=fs)


# Male model (balanced weights)

from sklearn.utils import class_weight
```

```
y_train_male = np.array(y_train_male).astype('int32').flatten()



weights_male =
class_weight.compute_class_weight(class_weight='balanced',classes=np.array([0,1]),y=y
_train_male)

cw_male = {0: weights_male[0], 1: weights_male[1]}

cw_male = {0: 0.5, 1: 5}

#Compile the male model with balanced weights

male_model.compile(optimizer='adam', loss='binary_crossentropy',

                   metrics=['accuracy'])

male_model.fit(X_train_male,y_train_male,epochs=20,class_weight=cw_male)

# Male Neural Network Performance Metrics

from sklearn.metrics import accuracy_score, recall_score, f1_score, roc_curve

y_probs_male = male_model.predict(X_test_male)

y_pred_male = (y_probs_male >= 0.5).astype(int)



cm_male = confusion_matrix(y_test_male, y_pred_male)
```

```
print("Confusion Matrix (male): ")

print(cm_male)



accuracy_male = accuracy_score(y_test_male,y_pred_male)



sensitivity_male = recall_score(y_test_male,y_pred_male, pos_label=1)



specificity_male = recall_score(y_test_male, y_pred_male, pos_label=0)



f1_male = f1_score(y_test_male, y_pred_male)

print("Accuracy:", accuracy_male)

print("Sensitivity:", sensitivity_male)

print("Specificity:", specificity_male)

print("F1:", f1_male)

# Male Neural Network Confusion Matrix

import matplotlib.pyplot as plt
```

```
import numpy

from sklearn import metrics



y_probs_male = male_model.predict(X_test_male)

y_pred_male = (y_probs_male >= 0.5).astype(int)



confusion_matrix_male = metrics.confusion_matrix(y_test_male, y_pred_male)



cm_display_male = metrics.ConfusionMatrixDisplay(confusion_matrix =

confusion_matrix_male, display_labels = [0, 1])



cm_display_male.plot()

plt.title("Confusion Matrix of Neural Network (Male)")

plt.show()

#Important male features

from sklearn.inspection import permutation_importance

from sklearn.metrics import get_scorer
```

```
from scikeras.wrappers import KerasClassifier


def scorer(model, X, y):
    y_pred = model.predict(X)
    return roc_auc_score(y, y_pred)


perm = permutation_importance(male_model, X_test_male, y_test_male, n_repeats=3,
                              random_state=0, scoring=scorer)


importance = pd.DataFrame({'Feature': X_train_male.columns,
                           'Importance': perm["importances_mean"],
                           'Standard Deviation': perm["importances_std"]})


male_importance = importance.sort_values('Importance',ascending=False)


print(male_importance[['Feature','Importance','Standard
Deviation']].to_string(index=False)
```

```
# Plot of important male features

male_importance.plot(figsize=(15,5),x='Feature',y='Importance',kind="bar",yerr='Standar
d Deviation')

plt.title("Neural Network Permutation Feature Importance (Male)")

plt.ylabel("ROC AUC Performance Drop")

plt.xticks(rotation=45)

plt.show()

# Male Neural Network ROC plot function

def plot_roc_male(y_truth, y_prob, ax):

    FPR, TPR, thresholds = roc_curve(y_truth, y_prob)



    AUC = np.trapz(TPR, FPR)

    ax.step(FPR, TPR, linewidth=2,label='Male (AUC = ' + str(round(AUC,2)) + ')')

    ax.plot([0,1],[0,1],'--', color = 'black')



    fs = 10

    ax.set_xlabel('False Positive Rate', fontsize=fs)
```

```
    ax.set_ylabel('True Positive Rate', fontsize=fs)
    ax.tick_params(axis='both', labelsize=fs)



    ax.set_title('ROC Curves for Neural Network')
    ax.legend(loc='upper left',fontsize=fs)
# ROC curve for male,female, and overall models
fig, ax = plt.subplots(figsize=(8,6))
plot_roc_overall(y_test, y_probs, ax)
plot_roc_male(y_test_male, y_probs_male, ax)
plot_roc_female(y_test_female, y_probs_female, ax)
plt.show()
# Feature importance difference plot between males and females
import matplotlib.pyplot as plt
import pandas as pd



# Feature importance data
```

```
data = {
    "Feature": [
        "GenHlth", "BMI", "HighBP", "Age", "HighChol", "HvyAlcoholConsump",
"CholCheck",
        "Income", "HeartDiseaseorAttack", "MentHlth", "DiffWalk", "PhysHlth",
"PhysActivity",
        "Education", "Stroke", "Smoker", "NoDocbcCost", "Veggies", "AnyHealthcare",
"Fruits"
    ],
    "Female": [
        0.058459, 0.039956, 0.018232, 0.015464, 0.013666, 0.003603, 0.003124,
        0.002254, 0.001412, 0.000963, 0.000828, 0.000774, 0.000349,
        0.000319, 0.000282, 0.000233, 0.000180, 0.000062, 0.000021, -0.000139
    ],
    "Male": [
        0.059273, 0.026509, 0.012096, 0.051672, 0.008858, 0.003717, 0.004597,
        0.003047, 0.003098, 0.000628, 0.001704, 0.001101, 0.000337,
```

```
        0.000182, 0.000629, -0.000328, -0.000087, 0.000120, 0.000097, 0.000446
    ]
}

df = pd.DataFrame(data)
df["Difference"] = df["Male"] - df["Female"]

# Sort by raw difference (not by absolute value)
df_sorted = df.sort_values("Difference", ascending=False)

# Plot
plt.figure(figsize=(10, 6))
bars = plt.barh(df_sorted["Feature"], df_sorted["Difference"], color='skyblue')
plt.xlabel("Difference in Importance (Male - Female)")
plt.title("Feature Importance Difference Between Males and Females (Neural Network)")
plt.axvline(x=0, color='gray', linestyle='--')
plt.gca().invert_yaxis()  # Highest difference at the top
```

```
plt.tight_layout()

plt.grid()

plt.show()
```

4. Conclusion

```
# Load the uploaded CSV file

file_path = 'male_female_importance_difference.csv'

df = pd.read_csv(file_path)

# Add a new column for mean difference across the three models

df['mean_difference'] = df[['logistic_regression', 'random_forest',

'neural_network']].mean(axis=1)



# Sort the dataframe by the mean difference

df_sorted = df.sort_values(by='mean_difference',

ascending=False).drop(columns='mean_difference')



# Set 'Feature' as index again
```

```
df_heatmap_sorted = df_sorted.set_index('Feature')



# Create the sorted heatmap

plt.figure(figsize=(10, 12))

sns.heatmap(df_heatmap_sorted, annot=True, cmap='coolwarm', center=0, linewidths=0.5,
fmt=".4f")

plt.title('Feature Importance Difference (Male - Female) — Sorted by Mean Difference')

plt.ylabel('Feature')

plt.xlabel('Model')

plt.tight_layout()

plt.savefig('feature_importance_difference.tiff', dpi=300, format='tiff')

plt.show()
```