

1. Problem Statement and Background

Written by: Thomas D. Robertson (50%), Tania Perdomo Flores (50%)

Statement of the Problem:

This project aims to develop a binary classification model to predict whether a female patient of Pima Indian Heritage has diabetes. The goal is to use a set of health-related variables to accurately classify patients as either diabetic or non-diabetic. This problem is important because early detection of diabetes can lead to better management and prevention of the disease, improving patient outcomes and reducing the long-term health risks associated with diabetes.

Dataset Overview:

The dataset used in this project is hosted on Kaggle and originates from the National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK). It is a subset of a larger dataset, specifically focused on female patients of Pima Indian Heritage, aged 21 and older. The dataset contains several independent variables, such as medical and health metrics, that are used to predict the dependent variable: whether the patient has diabetes or not. The target is binary: 1 for diabetes, 0 for non-diabetic.

Success Measures:

The primary success measure for this project will be the accuracy of the predictive model in classifying the diabetes status of patients. This will be evaluated by comparing the model's predictions to the known outcomes in the dataset. While other metrics (e.g., precision, recall,

ROC) may be useful, the focus here will be on overall accuracy as a proxy for the model's ability to correctly classify diabetes status.

Background and Importance:

Diabetes is a chronic and increasingly prevalent condition that affects millions of people worldwide. It is particularly concerning in the U.S., where diabetes-related complications can significantly reduce quality of life and lead to higher healthcare costs. Current healthcare systems often focus on reactive treatment after diagnosis, but early detection of diabetes could help shift the focus to preventative care. Accurate predictive models can assist healthcare professionals in identifying at-risk individuals early, enabling interventions that can delay or prevent the onset of diabetes and its associated complications.

Related Work:

There has been significant work in the area of diabetes prediction using machine learning, particularly in classifying diabetic versus non-diabetic patients based on medical and lifestyle factors. Various models, including logistic regression, decision trees, and neural networks, have been applied to similar datasets to predict diabetes risk. However, the focus on a specific demographic (Pima Indian Heritage women) provides an opportunity to refine models that might be more tailored to this group's unique health characteristics.

Goal:

By developing a model that accurately predicts diabetes status, this project seeks to contribute to the broader effort of improving early detection methods for diabetes, which could have significant implications for patient care and public health strategies.

2. Data and Exploratory Analysis

Written By: Logan Bolton (10%), Jun Fenghan (10%), Thomas D. Robertson (40%), Tania Perdomo Flores (20%), Kristian Obrusanszki (20%)

The Diabetes Dataset is hosted on Kaggle and originated from the National Institute of Diabetes and Digestive Kidney Diseases. The dataset is a subset of the larger original dataset by using constraining factors to reduce the selection of variable instances. All of the patients within the dataset are females of Pima Indian heritage and are at least 21 years old. There are several independent medical predictor variables and one target outcome (dependent) variable, whether the patient has diabetes or not.

Dataset Variables:

1. Pregnancies: Number of times pregnant
2. Glucose: Plasma glucose concentration 2 hours in an oral glucose tolerance test
3. BloodPressure: Diastolic blood pressure (mm Hg)
4. SkinThickness: Triceps skin fold thickness (mm)
5. Insulin: 2-Hour serum insulin (μ U/ml)
6. BMI: Body mass index ($\text{weight in kg}/(\text{height in m})^2$)
7. DiabetesPedigreeFunction: Diabetes pedigree function
8. Age: Age (years)
9. Outcome: Class variable (0 or 1) 268 of 768 are 1, the others are 0

General data cleaning included outlier removal and missing value imputation, in this dataset the missing values were 0 (excluding pregnancies and diabetic outcome), which is unrealistic for a living patient in any of the above variables.

Libraries Utilized: **Tidyverse**, **readxl**, **caret**, **pROC**, **MICE**, **caret**, and **corrplot**.

Data Cleaning and Preparation:

The dataset used in this project required several steps of cleaning to ensure its suitability for predictive modeling. The main issues addressed during the cleaning process were missing values, outliers, and the treatment of unrealistic data entries. Below is a detailed discussion of the steps taken to clean and preprocess the data, as well as the tools and methods used in the process.

Handling Missing Values:

One of the primary challenges in this dataset was the presence of missing values, particularly in columns such as *Insulin* and *BloodPressure*. In this dataset, missing values were represented by zero (0) for most variables (with the exception of the *Pregnancies* column). Since a value of zero is unrealistic for these variables in the context of a living patient (e.g., having zero insulin levels or zero blood pressure is not a plausible scenario), these entries were treated as missing data rather than valid observations.

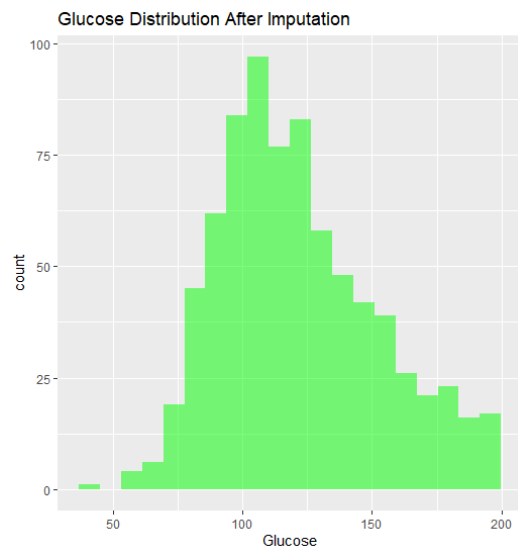
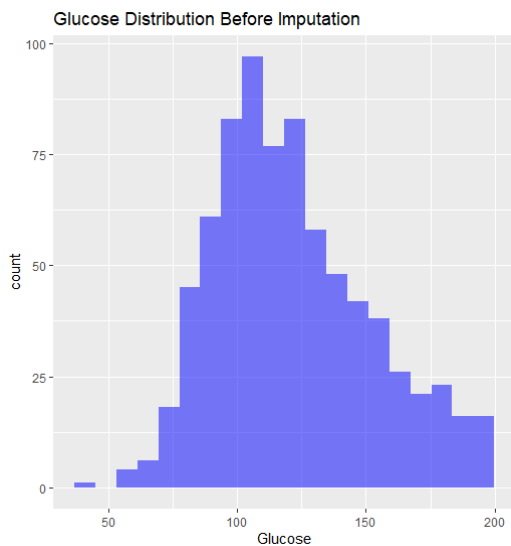
Two methods of imputation were used to address these missing values:

1. **Mean Imputation with Clustering for Insulin:** For the *Insulin* column, a clustering-based imputation strategy was employed. Since insulin levels are likely influenced by other medical metrics, particularly *Glucose*, we used clustering techniques

to group patients into classifications based on glucose levels (ranging from low to high).

The mean insulin values within each glucose group were then used to replace missing insulin values, under the assumption that insulin levels would correlate with glucose levels.

- Multivariate Imputation by Chained Equations (MICE):** The MICE method was applied to impute missing values across other columns in the dataset, except for *Pregnancies* and *Insulin*, which were handled differently. MICE is a more sophisticated technique that generates multiple imputed values for each missing entry by utilizing the relationships between variables in the dataset. This method was preferred for columns where the relationship between variables was complex and could benefit from imputations based on multiple features in the dataset. The goal of the imputation is to keep all replaced values relatively similar to pre-cleaning.

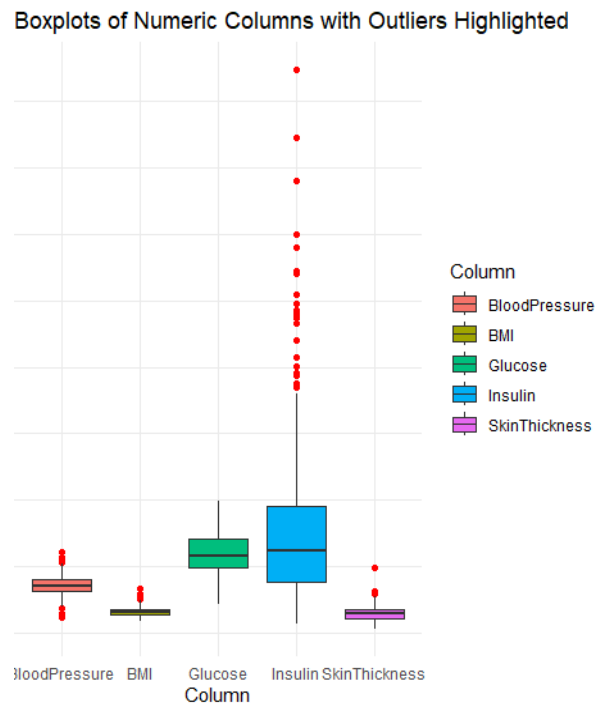
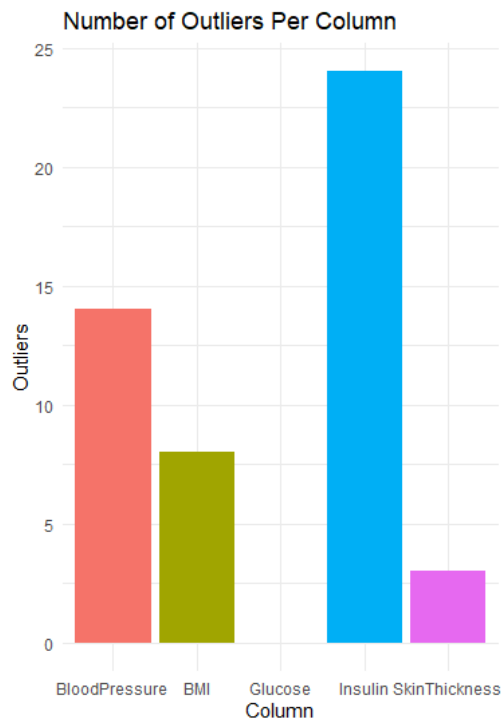


After imputation, both cleaned datasets (one with clustering-based imputation and one with MICE) were used to train logistic regression models to predict the likelihood of diabetes, with the goal of evaluating which cleaning method produced a model with higher predictive accuracy.

Outlier Detection and Handling:

In addition to missing values, several variables in the dataset contained outliers that needed to be addressed. (Outliers were collected after removal of 0 values in select columns)

- The *BloodPressure* column had 14 outliers.
- The *Insulin* column contained the highest number of outliers at 24.
- The *BMI* column had 8 outliers.
- The *SkinThickness* column had 3 outliers.



These outliers were identified using simple statistical methods (e.g., identifying values beyond 1.5 times the interquartile range). In cases where the outliers appeared to be due to data entry errors or extreme values that were not physiologically plausible (0 values for select columns), these were either capped or removed, in terms of that.

Related Data Variables:

Looking at the correlation matrix of data we can see various relationships between features within the dataset (See visualization matrix in MICE R script). Glucose and Outcome show a moderately strong positive correlation, suggesting higher glucose levels results in a positive diabetic diagnosis, which we would expect given blood glucose levels is a direct biological link to glucose cases. BMI and SkinThickness show a strong positive correlation, this is also expected as skinfold thickness can be used to estimate body fat, which links to BMI. Insulin and Glucose has a strong positive correlation, this is also expected as high glucose levels are generally accompanied by elevated insulin levels. Pregnancies also show a moderately positive correlation with age, naturally older people are more likely to have more children.

Data Preprocessing Tools and R Code:

To perform the cleaning, we used the following tools and R packages:

- **Tidyverse** for data manipulation and imputation tasks.
- **mice** package for performing Multivariate Imputation by Chained Equations (MICE) for missing values across the dataset.
- **ggplot2** and **corrplot** for visualizing data distribution, correlations, and variable relationships.
- **Caret** and **pROC** was used for initial model training, cross-validation, and performance measure calculations using confusion matrices and AUC and ROC curves.
- **Readxl** used to import data from the CSV diabetes dataset.

Overview:

The cleaning process involved addressing missing values, outliers, and unrealistic data entries to ensure the dataset was ready for analysis. The imputation methods used (clustering and MICE) provided a way to fill in missing data in a reasonable manner, and outlier detection helped mitigate any potential issues that could arise from extreme values. The cleaned dataset was then used to build predictive models and assess their accuracy. Further evaluations will focus on comparing the performance of models built with different imputation methods to determine the best approach for this specific dataset.

3. Methods

Written By: Logan Bolton (20%), Jun Han (20%), Thomas D. Robertson (20%), Tania Perdomo Flores (20%), Kristian Obrusanszki (20%)

Data Cleaning and Imputation Methods:

In our data cleaning process, we explored two primary methods: **Cluster-based Imputation** and **Multivariate Imputation by Chained Equations (MICE)**. The aim of these methods was to replace unrealistic or missing values in the dataset with categorized or imputed values that would make the data more suitable for our modeling process. Both methods were used to ensure that our dataset was robust before we applied our machine learning models.

Cluster-Based Imputation:

Method: We applied **k-means clustering** to impute missing insulin values based on glucose levels. This method groups data into clusters according to similar glucose levels and assigns the mean insulin values within each cluster to the corresponding missing insulin data points.

Justification: The correlation between glucose levels and insulin usage is well-documented in biological research, making **k-means clustering** a sensible method for imputing insulin values in terms of that. By grouping related data points, this approach utilizes the natural relationship between the two variables (glucose and insulin) to generate more plausible imputed values.

Evaluation: After applying the cluster-based imputation, we observed that most models showed slightly better results in terms of accuracy, specificity, and sensitivity compared to MICE imputation. This suggests that the k-means approach provided a more suitable representation of the missing data, leading to better model performance.

Multivariate Imputation by Chained Equations (MICE):

Method: MICE is a technique that imputes missing values across all columns simultaneously by considering the values of all other variables in the dataset. This method captures complex relationships among variables, generating plausible values for missing data points based on the correlations between features.

Justification: MICE was selected because it considers the interdependence among variables within the dataset. By modeling all variables' relationships, it can reveal hidden correlations, which may be beneficial for datasets like ours with biological data, potentially improving model accuracy.

Evaluation: While MICE worked well in capturing interdependencies, models that used MICE imputation showed slightly lower performance compared to k-means imputation, particularly in terms of **accuracy**. The cause of this difference is still unclear; it could be due to improper implementation of MICE or a lack of strong interdependencies in the dataset. Further exploration is needed to better understand its impact.

Explored Algorithms:

We considered a range of machine learning models to address the binary classification problem of predicting whether a patient is diabetic. These models were chosen based on their suitability for handling binary classification tasks and their ability to manage the complexities of the dataset.

Machine Learning Models Used:

1. **Logistic Regression (Baseline):** This is the standard model for binary classification, predicting probabilities in the (0, 1) range. It served as a baseline model for comparison.
2. **Logistic Regression (LASSO & SMOTE):** This variant incorporates LASSO regularization for feature selection and SMOTE (Synthetic Minority Over-sampling Technique) for handling class imbalance. These modifications improve performance, particularly for imbalanced datasets like ours, where diabetic patients are underrepresented.
3. **Support Vector Machines (SVM):** SVM was explored for its ability to handle high-dimensional data and non-linear relationships. It is particularly useful when data is not linearly separable and can prevent overfitting in high-dimensional spaces.

4. **K-Means Clustering:** Although primarily a clustering algorithm, k-means was explored as a pre-processing step to detect patterns between insulin and glucose levels. It was not used for prediction but rather to better understand data relationships for subsequent models.
5. **Decision Trees:** This model was used to identify important features that split the data effectively. Decision trees are versatile and can handle both linear and non-linear relationships, making them suitable for binary classification.

Models Considered but Not Explored:

We also considered other models, such as Random Forests and Naive Bayes, both of which are well-suited for binary classification problems. Due to time constraints, we did not explore these models fully, but they would be considered for future works.

Summary of Machine Learning Models:

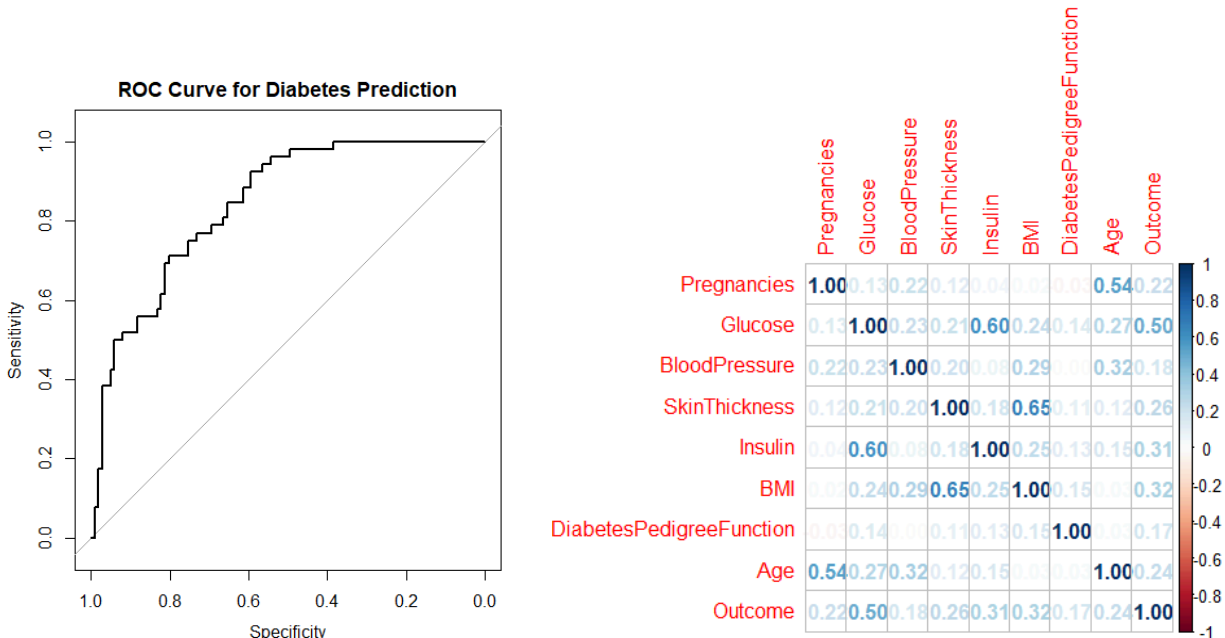
1. Logistic Regression (Baseline)

Logistic Regression is a member of generalized linear models commonly used for binary classification tasks. It predicts probabilities constrained to the $(0, 1)$ interval, making it ideal for predicting outcomes like whether a patient is diabetic or not. This model estimates the probability of an instance belonging to a specific category and serves as an efficient baseline for binary classification problems.

Performance:

- **Area Under the Curve (AUC):** 0.8406

This indicates good performance, establishing Logistic Regression as a reliable starting point for classification tasks.



2. Logistic Regression (LASSO & SMOTE)

This model extends the baseline Logistic Regression by incorporating two techniques: LASSO (Least Absolute Shrinkage and Selection Operator) regularization and SMOTE (Synthetic Minority Over-sampling Technique).

- **LASSO** helps with feature selection by penalizing less important variables, which simplifies the model and enhances interpretability.

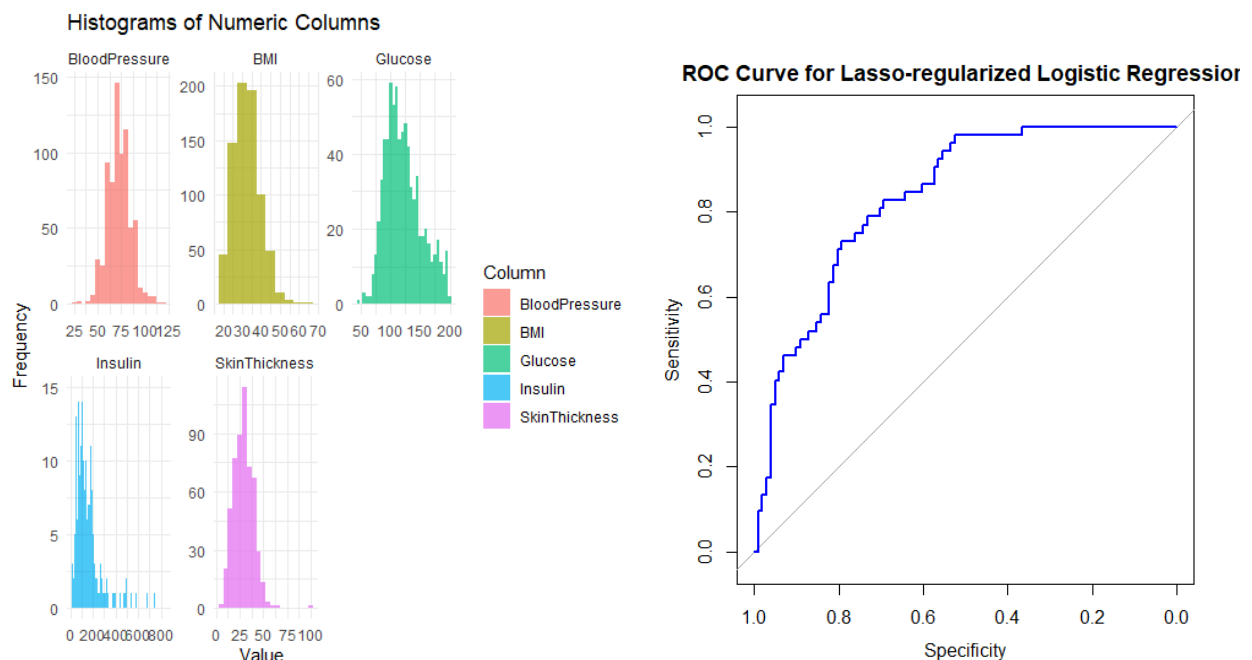
- **SMOTE** addresses class imbalance by generating synthetic examples for the minority class (e.g., diabetic patients), improving the model's ability to accurately classify underrepresented cases.

Together, these techniques aim to enhance predictive performance, particularly in imbalanced binary classification problems, while reducing overfitting.

Performance:

- **Area Under the Curve (AUC): 0.8343**

This is slightly lower than the baseline Logistic Regression model, likely due to the trade-off between mitigating overfitting and addressing class imbalance.



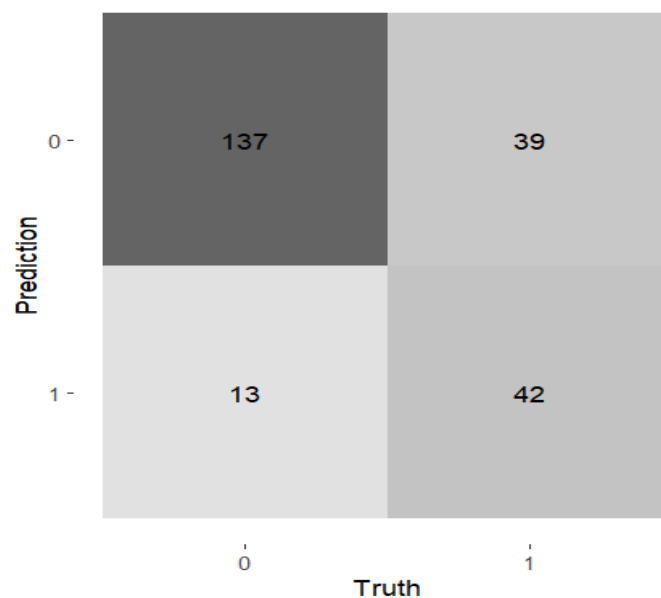
3. Support Vector Machines

Support Vector Machines (SVM) are highly effective for handling high-dimensional data, such as glucose and insulin levels in this dataset. SVM works by identifying the optimal hyperplane that separates different classes in the feature space, making it suitable for both linear and non-linear classification tasks. This capability allows SVM to distinguish between diabetic and non-diabetic patients, even when the features are not linearly separable.

SVM excels in high-dimensional spaces and is particularly good at avoiding overfitting, which makes it a strong choice for complex classification problems.

Key Strengths:

- Handles both linear and non-linear classification tasks.
- Effective in high-dimensional data spaces.
- Good at preventing overfitting.



A confusion matrix for SVM classification. The y-axis is labeled 'Prediction' with values 0 and 1. The x-axis is labeled 'Truth' with values 0 and 1. The matrix cells contain the following counts: (0,0) is 137, (0,1) is 39, (1,0) is 13, and (1,1) is 42. The cells are shaded in different tones of gray: (0,0) is dark gray, (0,1) is medium gray, (1,0) is light gray, and (1,1) is medium-dark gray.

Prediction	0	1
0	137	39
1	13	42
	0	1
	Truth	

4. K-Means Clustering

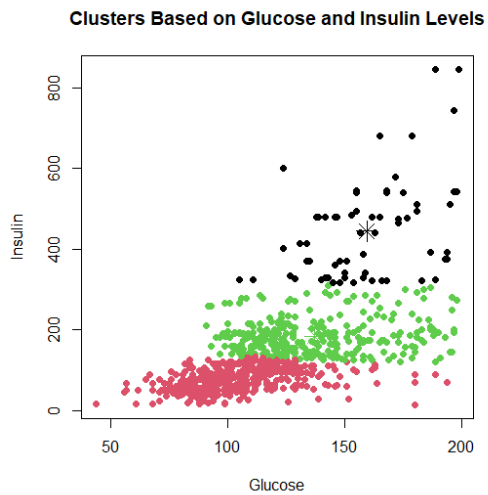
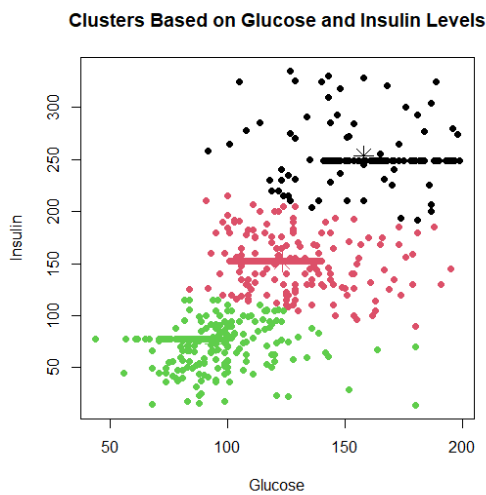
K-Means is an unsupervised clustering algorithm used to group data into distinct clusters based on similarity. For this problem, we focused on the relationship between insulin and glucose levels, specifically analyzing the low-to-high value range after data cleaning.

The K-means algorithm divides the data into three clusters, showing that insulin production can be grouped into three main categories. Although the clusters are not perfectly distinct, there is a clear grouping based on the centroids of the clusters.

While K-Means is not used as a predictive model, it serves as a valuable tool for visual inspection of data clusters. We applied two cleaning methods: K-means cleaning and imputed cleaning, visualizing the results in separate graphs. The imputed cleaning method produced tighter clusters, suggesting that imputation-based cleaning is more effective for models that rely on clustering for downstream analysis.

Key Insights and Performance:

- K-Means is used primarily for cluster visualization, not prediction.
- Imputation-based cleaning resulted in more distinct clustering.



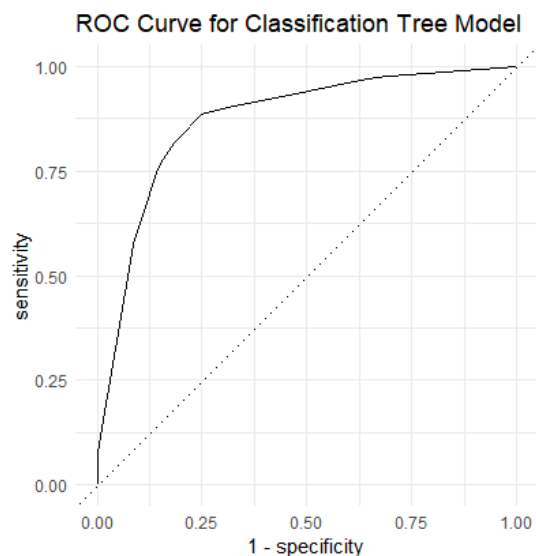
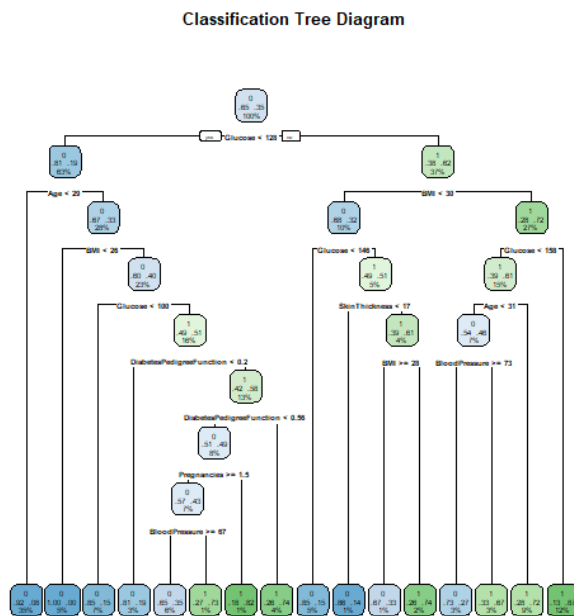
5. Decision Trees: Classification Tree with ROC Curve

The Decision Tree model was trained using the **RPart** engine. It works by recursively splitting the data at key decision points, which allows the model to classify instances based on the most relevant features. For this problem, the tree is used to determine whether a patient is diabetic or non-diabetic.

Decision Trees are versatile and can handle both linear and non-linear relationships in the data. This makes them particularly effective for binary classification tasks like identifying diabetic patients. Additionally, Decision Trees excel at identifying the most important variables for classification, making them a valuable tool for feature selection.

Key Strengths and Performance:

- Effective for both linear and non-linear relationships.
- Good at identifying significant features for classification.



4. Tools

Written By: Thomas D. Robertson (65%), Tania Perdomo Flores (35%)

Tools Used for Data Cleaning, Visualization, and Model Evaluation:

To efficiently clean, visualize, and evaluate the models, we employed a suite of tools and R packages, each selected for their relevance to the problem at hand and their ability to support various stages of data preparation, analysis, and evaluation. Below is a breakdown of the tools used, how they were applied, and the reasons for their selection:

1. Tidyverse (Data Cleaning and Manipulation)

Tool: Tidyverse

Usage: We used the Tidyverse suite of packages for data manipulation tasks such as cleaning, transforming, and imputation. This package provides a cohesive set of functions for data wrangling, allowing us to handle missing values, format data, and prepare it for model training efficiently.

Justification: Tidyverse is widely regarded as a powerful and user-friendly tool for data manipulation. Its syntax is intuitive and works well with other R packages. Given its flexibility and comprehensive set of functions, it was the ideal choice for preparing our dataset for further analysis and modeling.

What Worked Well: Tidyverse's **dplyr** and **tidyr** functions allowed us to efficiently clean and reshape the data, making the entire data wrangling process smoother.

What Could Be Improved: While Tidyverse works well for typical cleaning tasks, handling larger datasets could benefit from optimization in terms of speed and memory usage, particularly in larger datasets with missing values.

2. MICE (Multivariate Imputation by Chained Equations)

Tool: MICE

Usage: We utilized the MICE package to perform Multivariate Imputation by Chained Equations (MICE) to handle missing values across the dataset. This technique imputes missing data by modeling each feature with missing values as a function of the other features in the dataset.

Justification: MICE is particularly useful in cases where missing data may depend on other variables in the dataset. Since we were working with biological data (e.g., insulin and glucose levels), which might have underlying relationships, MICE was a natural choice to preserve the correlations and patterns in the data.

What Worked Well: MICE is effective in capturing interdependencies between variables and generating plausible values for missing data. It enabled us to impute missing insulin values in a manner that preserved the relationships between variables like glucose and insulin.

What Could Be Improved: MICE can be computationally expensive for large datasets, especially with many features. Additionally, the imputation process can sometimes introduce noise if the relationships between variables are not well captured.

3. ggplot2 and corrplot (Data Visualization)

Tools: ggplot2, corrplot

Usage: These visualization tools were used to display data distributions, correlations between variables, and variable relationships. ggplot2 allowed for the creation of custom plots to explore data trends, while corrplot provided a clear, graphical representation of the correlation matrix.

Justification: Visualizing the data was crucial for understanding underlying patterns and relationships before applying models. The flexibility and customization options of ggplot2 made it a good choice for creating a range of plots, from histograms to scatter plots. corrplot was

particularly helpful in displaying the correlation matrix for feature selection and understanding how different variables interact.

What Worked Well: The plots produced by ggplot2 were clear, aesthetically pleasing, and informative, helping us identify trends and anomalies in the dataset. The correlation matrix from corplot provided valuable insight into variable relationships.

What Could Be Improved: Although ggplot2 is highly versatile, complex visualizations with many variables can sometimes become cluttered. It would be beneficial to experiment with interactive visualization tools, such as plotly, for a more dynamic exploration.

4. Caret and pROC (Model Training, Cross-Validation, and Evaluation)

Tools: Caret, pROC

Usage: The Caret package was used for initial model training, cross-validation, and tuning hyperparameters for various machine learning models. The pROC package was used for performance evaluation, particularly for generating ROC curves and calculating the Area Under the Curve (AUC) to assess model performance.

Justification: Caret is a comprehensive package for training and evaluating machine learning models in R. It offers convenient functions for cross-validation, model training, and performance metrics, making it ideal for the iterative process of model development. pROC provides a specialized focus on evaluating classifier performance using ROC curves, a common metric for binary classification tasks.

What Worked Well: Both Caret and pROC streamlined the process of model evaluation and provided easy access to key performance metrics like accuracy, AUC, and ROC curves, which were critical for comparing models.

What Could Be Improved: While Caret offers a wide variety of models, some advanced

techniques, like hyperparameter tuning for more complex models, could require additional customization or external tools beyond Caret.

5. Readxl (Data Import)

Tool: readxl

Usage: This package was used to import the diabetes dataset from a CSV file into R for initial processing.

Justification: readxl is a lightweight package that is specifically designed for importing data from Excel and CSV files, making it easy to load data into R for subsequent analysis.

What Worked Well: The readxl package worked efficiently for importing the dataset, and it was straightforward to use for handling the initial step of loading data.

What Could Be Improved: There are no significant issues with readxl for basic data import tasks. However, for very large files, performance could be enhanced by using more specialized packages such as **data.table**.

6. glm (Logistic Regression Baseline Model)

Tool: glm

Usage: We used the glm function in R to train the baseline logistic regression model for binary classification.

Justification: Logistic regression is a standard, well-understood model for binary classification problems, making it a natural choice for our initial model. It allows us to predict probabilities, which are useful for understanding the likelihood of a patient being diabetic.

What Worked Well: Logistic regression provided a solid baseline for comparison against more complex models. It was easy to implement and interpret the results.

What Could Be Improved: While logistic regression works well for binary classification, its simplicity may not capture non-linear relationships or more complex patterns in the data, which is why more advanced models were explored.

7. e1071 (Support Vector Machines)

Tool: e1071

Usage: The e1071 package was used to implement Support Vector Machines (SVM) for classification tasks. This package allowed us to explore non-linear decision boundaries and better understand the separation of classes.

Justification: SVM is powerful for handling high-dimensional data and can model non-linear relationships, making it appropriate for biological data like glucose and insulin levels, which may not have simple linear separations.

What Worked Well: The e1071 package provided a straightforward implementation of SVM, and the model performed well in distinguishing between the diabetic and non-diabetic classes.

What Could Be Improved: Tuning SVM models can be challenging, especially with high-dimensional data, and the training process can be time-consuming. Exploring other tools for SVM hyperparameter optimization could improve performance.

Tools Considered but Not Used:

- **Random Forests:** We considered using random forests for model evaluation but did not pursue it due to time constraints. Random forests are powerful for binary classification tasks but would require additional tuning and cross-validation for optimal performance.

- **Naive Bayes:** While Naive Bayes is a simple and effective model for binary classification, it assumes independence between features, which may not hold in our biological dataset. Therefore, we chose not to pursue this model further.

Overview:

The combination of tools we used provided a robust workflow for data cleaning, visualization, model training, and evaluation. **Tidyverse** and **mice** facilitated efficient data wrangling and imputation, while **ggplot2** and **corrplot** provided valuable insights through visualization. **Caret** and **pROC** enabled thorough model training and evaluation. Future improvements could include exploring advanced optimization techniques for SVM, considering the use of more interactive visualization tools, and revisiting models like Random Forests and Naive Bayes to expand our analysis.

5. Results

Written By: Thomas D. Robertson (65%), Tania Perdomo Flores (35%)

Explanation of Evaluation Metrics:

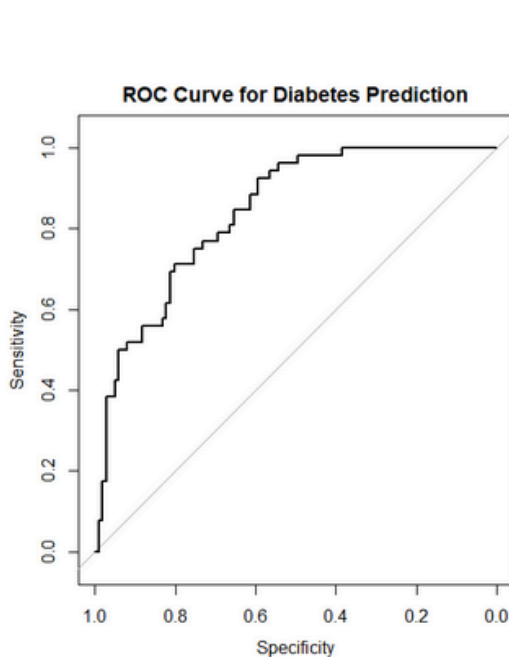
- **Accuracy:** The percentage of all predictions (both diabetic and non-diabetic) that the model correctly identified.
- **Sensitivity (Recall):** The percentage of actual positives (diabetic cases, class 0) correctly identified. High sensitivity reduces false negatives, which is crucial for early detection of diseases like diabetes.
- **Specificity:** The percentage of actual negatives (non-diabetic cases, class 1) correctly identified. High specificity reduces false positives, ensuring that non-diabetic patients are not wrongly classified as diabetic.
- **Positive Predictive Value (PPV / Precision):** The percentage of predicted positives (diabetic predictions) that are actually positive. High precision reduces false positives.
- **Negative Predictive Value (NPV):** The percentage of predicted negatives (non-diabetic predictions) that are actually negative. High NPV reduces false negatives.
- **Balanced Accuracy:** The average of sensitivity and specificity, accounting for imbalance.
- **Kappa:** A statistic that measures the agreement between predicted and actual outcomes, adjusted for random chance.
- **McNemar's Test P-Value:** A test that compares the false positive and false negative rates to check for balance between errors. A p-value > 0.05 indicates balanced errors.
- **Confidence Interval (95% CI):** The range within which the true accuracy of the model is expected to fall, with 95% confidence.

<i>Metrics</i>	<i>Logistic Regression (Baseline)</i>	<i>Logistic Regression (LASSO)</i>	<i>Support Vector Machine (SVM)</i>	<i>Decision Tree</i>
<i>Accuracy</i>	77.32%	76.47%	75.16%	74.68%
<i>95% Confidence Interval</i>	74.11 - 80.31%	68.94 - 82.94%	67.65 - 81.79%	67.05 - 81.33%
<i>Sensitivity (Recall)</i>	86.78%	80.20%	90.00%	80.41%
<i>Specificity</i>	58.87%	69.23%	47.17%	64.91%
<i>Precision</i>	80.46%	83.51%	76.27%	79.59%
<i>Balanced Accuracy</i>	72.82%	74.71%	68.58%	72.66%
<i>McNemar's P-Value</i>	0.0041	0.6171	0.0058	N/A

Detailed Model Analysis:

Logistic Regression (Baseline)

- **Accuracy:** 77.32% – This model offers a solid overall performance, correctly identifying diabetic and non-diabetic patients.
- **Sensitivity (Recall):** 86.78% – The model excels at identifying diabetic cases (class 0), making it strong for detecting patients who are at risk.
- **Specificity:** 58.87% – However, the model struggles with non-diabetic cases, incorrectly labeling a significant portion as diabetic, which results in higher false positive rates.
- **McNemar's Test P-Value:** 0.0041 – The p-value indicates an imbalance between false positives and false negatives, pointing to the need for further model refinement.

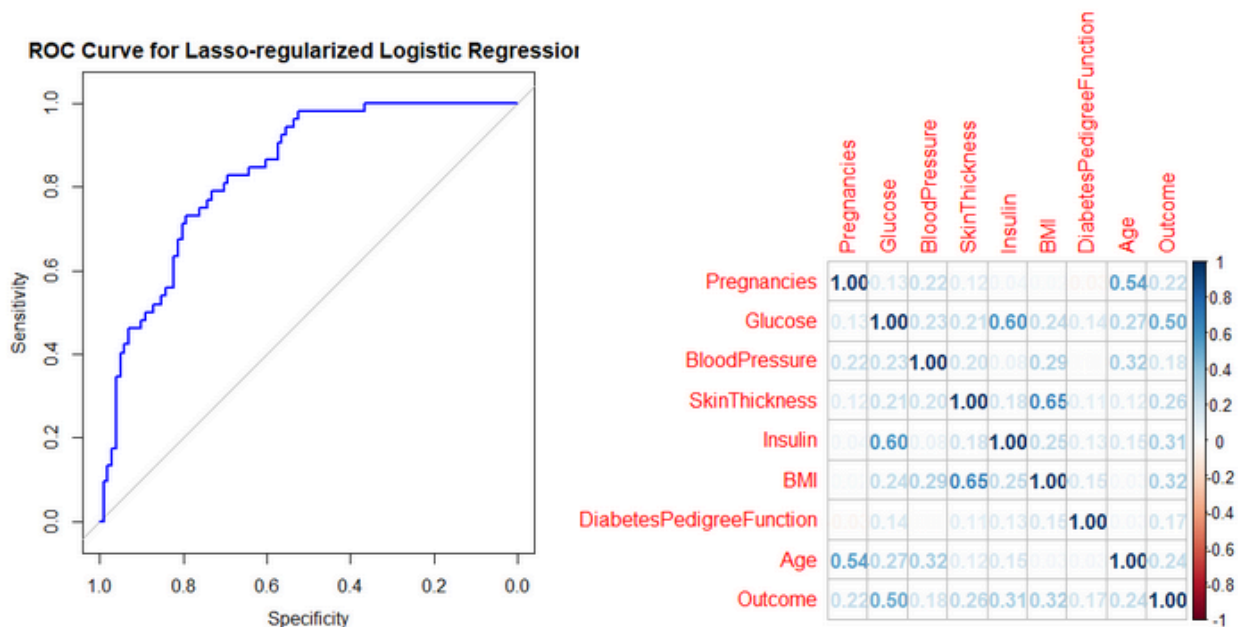


	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Pregnancies	1.00	0.13	0.22	0.12	0.04	0.07	0.00	0.54	0.22
Glucose	0.13	1.00	0.23	0.21	0.60	0.24	0.14	0.27	0.50
BloodPressure	0.22	0.23	1.00	0.20	0.08	0.29	0.00	0.32	0.18
SkinThickness	0.12	0.21	0.20	1.00	0.18	0.65	0.11	0.12	0.26
Insulin	0.04	0.60	0.08	0.18	1.00	0.25	0.13	0.15	0.31
BMI	0.07	0.24	0.29	0.65	0.25	1.00	0.15	0.00	0.32
DiabetesPedigreeFunction	0.00	0.14	0.00	0.11	0.13	0.15	1.00	0.00	0.17
Age	0.54	0.27	0.32	0.12	0.15	0.00	0.00	1.00	0.24
Outcome	0.22	0.50	0.18	0.26	0.31	0.32	0.17	0.24	1.00

Logistic Regression (LASSO & SMOTE)

- **Accuracy:** 76.47% – Slightly lower than the baseline but with improved class balance due to LASSO regularization and SMOTE (Synthetic Minority Over-sampling Technique) for class balancing.
- **Sensitivity (Recall):** 80.20% – Sensitivity is slightly reduced compared to the baseline (86.78%) but still remains strong for detecting diabetic patients.

- **Specificity:** 69.23% – Significant improvement in specificity compared to the baseline (58.87%), reducing false positives.
- **Balanced Accuracy:** 74.71% – This model achieves the highest balanced accuracy, striking a better balance between sensitivity and specificity.
- **McNemar's P-Value:** 0.6171 – The high p-value suggests that the false positives and false negatives are relatively well balanced in this model.



Support Vector Machine (SVM)

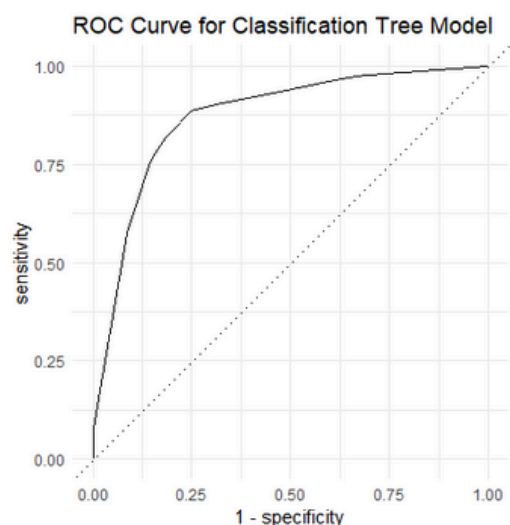
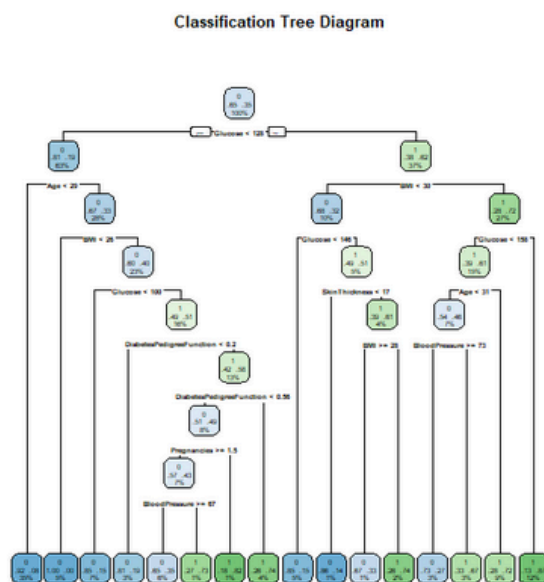
- **Accuracy:** 75.16% – The lowest accuracy among the models, indicating a trade-off for its higher sensitivity.
- **Sensitivity (Recall):** 90.00% – SVM excels at identifying diabetic cases, with the highest sensitivity of all models. This is critical for minimizing false negatives, especially for early detection of diabetes.
- **Specificity:** 47.17% – However, it suffers from the lowest specificity, leading to a high number of false positives.
- **Balanced Accuracy:** 68.58% – The lowest balanced accuracy, indicating a significant imbalance between false positives and false negatives.

- **McNemar's P-Value:** 0.0058 – The low p-value indicates an imbalance between false positive and false negative rates, highlighting the model's tendency to over-predict diabetes.

Prediction	0	137	39
	1	13	42
	Truth	0	1

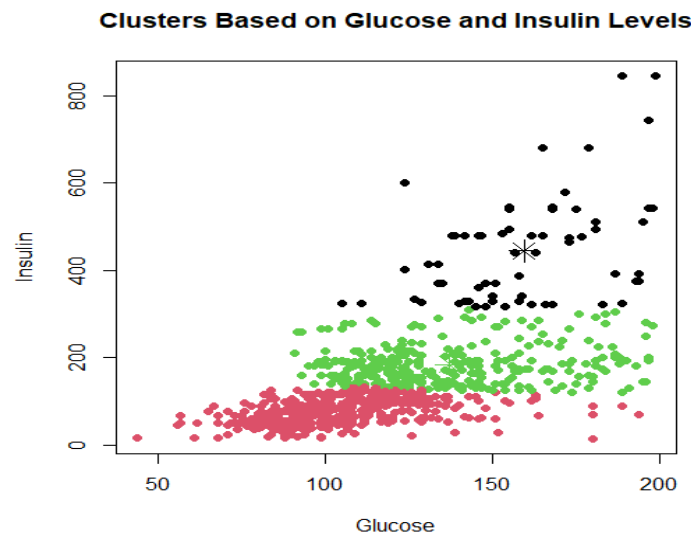
Decision Tree

- **Accuracy:** 74.68% – Slightly lower accuracy than the baseline and LASSO Logistic Regression but offers an acceptable performance overall.
- **Sensitivity (Recall):** 80.41% – A good balance of sensitivity, slightly lower than the baseline but higher than the SVM.
- **Specificity:** 64.91% – Better specificity than the baseline but lower than the LASSO Logistic Regression, reducing false positives to some extent.
- **Balanced Accuracy:** 72.66% – A solid balanced accuracy, offering a more interpretable model without sacrificing too much performance.



Clustering Analysis (K-Means):

While not a direct classification model, K-means clustering showed that imputation methods (e.g., cleaning missing values) contributed to more defined clusters, particularly between insulin and glucose levels. This implies that better data preprocessing and cleaning can improve model accuracy and correlation, enhancing the performance of models relying on these features.



Conclusion and Recommendations:

- **Best Model for Balance:** The **Logistic Regression with LASSO & SMOTE** achieved the highest balanced accuracy (74.71%) and showed the best trade-off between sensitivity and specificity, making it the most reliable model for general use.
- **Best Model for Sensitivity (Recall):** The **Support Vector Machine (SVM)** excels in sensitivity (90.00%), making it ideal when minimizing false negatives is a priority, particularly for early-stage diabetes detection. However, its low specificity (47.17%) makes it unsuitable in contexts where false positives must be minimized.
- **Best for Interpretability:** The **Decision Tree** model, while slightly underperforming compared to the LASSO model, offers excellent interpretability. It provides transparency in decision-making processes, which is beneficial for clinical applications where model transparency is crucial.
- **Considerations:** Although the **Logistic Regression (Baseline)** offers a solid foundation, it shows class imbalance issues that can be improved with techniques like LASSO and SMOTE.

In summary, the **LASSO Logistic Regression** model is the most well-rounded for general usage, while the **SVM** is the best for minimizing false negatives.

6. Summary and Conclusions

Written By: Logan Bolton (20%), Jun Fenghan (20%), Thomas D. Robertson (20%), Tania Perdomo Flores (20%), Kristian Obrusanszki (20%)

High-Level Summary of Results

In this project, we developed and evaluated several machine learning models to predict diabetes in female patients of Pima Indian heritage, using metrics such as accuracy, sensitivity, specificity, and balanced accuracy.

Key Findings:

- **Baseline Logistic Regression:** With strong accuracy (77.32%) and high sensitivity (86.78%), it was effective at detecting diabetes but had low specificity (58.87%), leading to false positives. McNemar's p-value (< 0.05) confirmed class imbalance.
- **LASSO Logistic Regression:** This model balanced sensitivity (80.20%) and specificity (69.23%), resulting in the highest balanced accuracy (74.71%). The slight reduction in sensitivity was offset by improved specificity, making it more reliable for general use.
- **Support Vector Machine (SVM):** SVM excelled in sensitivity (90.00%), minimizing false negatives, but had low specificity (47.17%), leading to high false positives and the lowest balanced accuracy (68.58%). It's ideal for prioritizing recall in early detection.
- **Decision Tree:** Provided balanced performance with sensitivity (80.41%) and specificity (64.91%), offering transparency and interpretability. Its accuracy (74.68%) was slightly lower than the logistic regression models, but its balanced metrics make it valuable for decision-making.

Overall Conclusion: LASSO Logistic Regression emerged as the most reliable model, providing consistent performance across all metrics, suitable for clinical use. For minimizing false negatives, SVM is ideal due to its high sensitivity. The Decision Tree offers transparency with balanced performance, making it a good choice for interpretable decision-making.

Surprising Insights:

- The LASSO model's significant improvement in specificity was unexpected, showing the effectiveness of addressing class imbalance, particularly using SMOTE.
- While SVM's high sensitivity was expected, its poor specificity highlighted the trade-offs of prioritizing recall in imbalanced datasets.

In conclusion, the best model depends on the specific needs of the application, balancing false positives and false negatives. Future work could explore further tuning or combining techniques to optimize performance.

7. Appendix

Github Repository

<https://github.com/TDRobertson/CSC-3220-Group-1-Diabetes-Patients.git>

Diabetes Dataset -

<https://www.kaggle.com/datasets/akshaydattatraykhare/diabetes-dataset?resource=download>