MAT 422: Project Paper
Group 4: Armando Herrera, Mark Zaldivar, and Ritisha Das

I. Introduction

Diabetes is a chronic disease that occurs when the body has elevated blood glucose levels as a result of an inability to properly produce insulin. Since its emergence in the latter half of the 19th century, diabetes has remained a prevalent global health issue. In 1994, the CDC declared diabetes an epidemic, marking a sharp rise in cases compared to previous decades [1]. The number of cases continues to increase at an especially alarming rate in the United States having reached a soaring 37.3 million cases, or 11.3% of the population, as of 2022 with an estimated 1.2 million Americans being diagnosed every year [2]. Furthermore, some projections show that 1 in 8 adults worldwide, roughly 783 million people, will be living with diabetes in 2045 [3]. This marks a 46% increase.

This is particularly alarming considering diabetes remains a deadly disease. In 2017, it was the seventh leading cause of death in the United States, having been listed as the underlying or contributing cause of death on 270,702 death certificates [3]. Similar trends have continued in recent years. In 2020, about 16.8 million emergency department visits and a total of 7.68 million hospital discharges were reported with diabetes as a listed diagnosis among US adults above the age of 18 [3]. These discharges included:

- 1.68 million for major cardiovascular diseases
    - 368,000 for ischemic heart disease
    - 321,000 for stroke
- 160,000 for lower-extremity amputation
- 232,000 for hyperglycemic crisis
- 51,000 for hypoglycemia

The economic burden of diabetes is similarly substantial. In the United States, the estimated annual cost of diabetes reached $327 billion in 2017, with $237 billion attributed to direct medical costs and $90 billion to reduced productivity [8]. Including the effects of undiagnosed diabetes, prediabetes, and GDM, this figure rises above $400 billion. For individuals, this financial strain often manifests as out-of-pocket expenses for medications, insulin, and regular testing supplies, which can be particularly burdensome for those without comprehensive healthcare coverage. In view of this, it is not surprising that medical spending for diabetes ranks among the highest for all conditions [8]. Globally, diabetes poses a significant challenge for low and middle-income countries, where healthcare infrastructure is often inadequate to manage the disease effectively, resulting in higher morbidity and mortality rates.

Diabetes significantly affects various parts of the body, including the eyes, kidneys, nerves, heart, and blood vessels, leading to complications like vision loss, kidney failure, nerve damage, heart attacks, and poor blood flow, particularly in the feet, which can result in ulcers and potential amputations if left untreated [5]. Treatments include insulin injections, various medications, and lifestyle changes. Despite these interventions, the progression of diabetes and

its associated complications often remain unchecked, underscoring the urgent need for improved preventive strategies.

Despite vast research efforts, the exact cause of diabetes remains unknown. Additionally, diabetes testing, despite being fairly effective, has its own challenges and limitations. Current diagnostic methods rely heavily on measures such as fasting blood glucose levels, glycated hemoglobin (HbA1c), and the oral glucose tolerance test (OGTT) [6]. These tests are often rather invasive, costly, and not easy to perform. For example, the OGTT requires eight hours of fasting prior to the test and involves repeated blood samples over the course of several hours [7]. Moreover, these tests may not effectively identify individuals at high risk for diabetes before the disease progresses. As a result, accurate predictive models are of utmost importance in identifying, monitoring, and preventing diabetes. This is especially true of the large number of individuals unknowingly living with diabetes. As of 2022, there are approximately 8.7 million of such undiagnosed cases [4]. That is, about 1 in 5 individuals in the United States are unaware they have diabetes.

To address these issues, we employ numerous deep learning techniques and application models in an attempt to construct a diabetes status prediction system. In this paper, we limit our scope to a relatively small population near Phoenix, Arizona, USA by analyzing the Pima Indians Diabetes Database (PIDD). This is further motivated by the fact that American Indians display an abnormally high rate of diagnosed diabetes (13.6%) compared to the national average (11.3%) [4]. Diabetes disproportionately affects certain ethnic groups due to genetic predispositions and socio-economic factors, making the Pima Indian population a critical focus for research and intervention efforts.

The Pima Indians Diabetes Database (PIDD), which was originally collected by the National Institute of Diabetes and Digestive and Kidney Diseases, contains medical records of 768 Pima Indian women aged 21 and older, of which 258 tested positive for diabetes and 500 tested negative. Therefore there is one target variable, diabetes status, with each record containing eight health-related attributes. This data focuses on identifying factors which contribute to the outcome, diabetes, which is a prevalent health issue within the Pima population-influenced by a combination of genetic and environmental factors. The eight attributes are:

- Pregnancies (number of times pregnant)
- Oral Glucose Tolerance Test (OGTT)
- Blood pressure
- Skin thickness
- Insulin
- BMI
- Age
- Pedigree diabetes function (a measure of diabetes risk based on family history)

Through analysis of these attributes, our deep learning models extract underlying dataset patterns that support decision making and accurate predictive modeling. With this, we aim to effectively

identify the onset of diabetes. This would be especially valuable in that it can assist with informing preventative health measures and early intervention for at-risk populations.

Similar deep learning methods have recently led to significant results in various research disciplines, such as computer vision, natural language processing, speech recognition, and more. These deep learning methods, which allow for more efficient processing of large datasets, have emerged as powerful tools in healthcare, revolutionizing disease prediction, diagnostics, and treatment planning. Unlike traditional machine learning models, which rely on handcrafted features, deep learning models can autonomously learn complex patterns and relationships from raw data. This capability has enabled breakthroughs in areas such as image-based cancer detection, natural language processing for clinical note analysis, and predictive modeling of patient outcomes [9]. In the context of diabetes, deep learning offers a unique advantage by enabling the integration of diverse data sources, such as genetic markers, clinical metrics, and lifestyle factors, into a unified predictive framework.

In this paper, we use several deep learning models to construct a predictive model for diabetes status. These models are tested using standard metrics such as accuracy, precision, recall, AUC, and F1 scores. The major goal of our research is to contribute to the development of more accurate prognostic tools for diabetes status prediction and motivate further research towards this end.

II. Related work

As previously mentioned, deep learning methods have emerged as useful tools in a multitude of research areas. In this section, we limit our attention to studies involving machine learning and deep learning techniques in the application of prognostic diabetes prediction, of which there are many such studies. We provide an extensive examination of the literature on this topic including a diverse range of methods, datasets, performance criteria, and results. Furthermore, we analyze the proposed methodology of each study and assess the merits of their approaches.

Among the most cited in the literature is the study conducted by Chang et al. [10]. The paper, titled “Pima Indians diabetes mellitus classification based on machine learning (ML) algorithms”, employed three ML algorithms to analyze the Pima Indian Diabetes dataset: J48 decision tree, random forest, and naive Bayes. They achieved a peak accuracy close to 80% with both the random forest model and the naive Bayes model. Chang’s study emphasized the significance of optimal parameter selection and preprocessing techniques to enhance model performance. It also noted some limitations, such as PIDD’s dataset size and class imbalance, suggesting to deploy additional strategies in the future like resampling techniques to improve detection accuracy.

Based on the results, which do not depend on invasive laboratory data, Chang et al. proposed an e-diagnosis system for the detection and classification of diabetes.

Naz et al [11], in a paper titled “Deep learning approach for diabetes prediction using Pima Indian dataset” , applied four classification algorithms: artificial neural networks (ANN), deep learning (DL), decision tree (DT), and naive Bayes (NB). The results displayed a promising

accuracy of 98.07% on the Pima Indian dataset using DL. Decision trees also resulted in a high accuracy of 96.62%, while ANN achieved 90.34% and NB underperformed with 76.33%. The Deep Learning model had the highest accuracy on the Pima Indian dataset at the time of the paper's publication.

García-Ordás et al. [12] use deep learning techniques along with oversampling and feature augmentation in a paper titled "Diabetes detection using deep learning techniques with oversampling and feature augmentation". The paper proposes a pipeline based on deep learning techniques that features data augmentation using a variational autoencoder (VAE) and feature augmentation using a sparse autoencoder (SAE). The pipeline also employs a convolutional neural network (CNN) for classification. Training the sparse autoencoder with the convolutional neural network yielded an accuracy of 92.31% on the Pima Indian dataset, marking a 3.17% increase compared to previous methods. The proposed deep learning pipeline resulted in improved diabetes status detection through the use of oversampling and powerful feature representation.

Mousa et al. [13] conducted a comparative study using three popular models: Long Short-Term Memory (LSTM), Random Forest (RF), and Convolutional Neural Network (CNN) to be used for diabetes detection on the PIDD. Their results indicated that the LTSM model outperformed the other two by achieving an accuracy of 85%. This success was attributed to the model's capability to capture relationships inherent in clinical data. Whilst RF and CNN models did exhibit promising results, they were not as accurate as LSTM. Mousa's study also pointed out challenges such as the limited dataset size and potential class imbalance, and advocates for future research to explore data augmentation and various other types of approaches.

Aouamria et al. [14] proposed a novel approach using three proven deep learning models: Long Short-Term memory (LSTM), Deep Neural Networks (DNN), and Convolutional Neural Networks (CNN). In a paper titled "An ensemble deep learning model for diabetes disease prediction" , they used a soft voting classifier to enhance predictive performance. Moreover, data fusion is used to address the problems of small sample size, outliers, and missing data seen in the Pima Indian dataset. The model yielded an accuracy of 85.9% on the PIDD, 98.0% on the Frankfurt Hospital Germany Diabetes dataset (FHGDD), and 99.81% on a combined dataset. These results outperformed individual classifiers, motivating future research using this method.

Ayon et al. [15], in a paper titled "Diabetes: A Deep Learning Approach", proposed a strategy for the detection of diabetes using deep neural networks and training the attributes in five-fold and ten-fold cross-validation fashion. The results yielded an accuracy of 98.35% on the Pima Indian dataset in the five-fold case along with high scores across various other metrics including F1 score and MCC. Similar results were shown in the ten-fold case, obtaining an accuracy of 97.11% on the Pima Indian dataset with similarly high values for sensitivity and specificity. These values outperform studies using methods such as logistic regression, improved GA, modified K-means and SVM, and SVM with efficient coding. Such promising experimental results motivate further research along this path.

Dwivedi [16] evaluated the performance of various machine learning algorithms for the prediction of diabetes. The algorithms used were support vector machines (SVM), artificial

neural network (ANN), logistic regression, classification tree, naive Bayes, and K-nearest neighbor (KNN). The models are tested according to numerous metrics including accuracy, specificity, sensitivity, precision, negative predictive value, false positive rate, rate of misclassification, F1 measure and receiver operating characteristic (roc) curve. The best-performing algorithm, logistic regression, yielded an accuracy of 78% and a rate of misclassification of 0.22. However, naive Bayes resulted in the highest precision score of 82%.

Based on the established literature, we choose to address the problems discussed in the Chang and Mousa studies. More specifically, we intend to refine previous models of diabetes detection by pursuing various resampling techniques to address the class imbalance, limited data size, and irregularities of the Pima Indian dataset. These techniques include cross-validation, bootstrap sampling, oversampling, and undersampling, each of which will be thoroughly discussed in the subsequent sections.

III. Proposed methodology

1. Dataset overview and challenges

As already mentioned in the related works, the Pima Indian Diabetes Dataset (PIDD) is a widely used dataset for binary classification problems, particularly for predicting the presence or absence of diabetes based on various medical attributes. It consists of 8 input features: Pregnancies, Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes Pedigree Function, and Age. The target variable, Outcome, is binary, where 0 indicates no diabetes and 1 indicates the presence of diabetes. The dataset contains 768 entries, which provides a good set of variables that describe various aspects of an individual's health, such as blood sugar levels, body mass index, and medical history. The goal of working with this dataset, of course, is to create a model that accurately and precisely uses these metrics to help predict diabetes.

However, one significant challenge in this dataset is class imbalance, where the number of individuals with no diabetes (0) significantly outweighs those with diabetes (1), creating an almost 2:1 ratio (500 0’s and 268 1’s). This class imbalance can affect model performance, as the model may be biased toward predicting the majority class (which in this case, is negative outcomes/no diabetes).

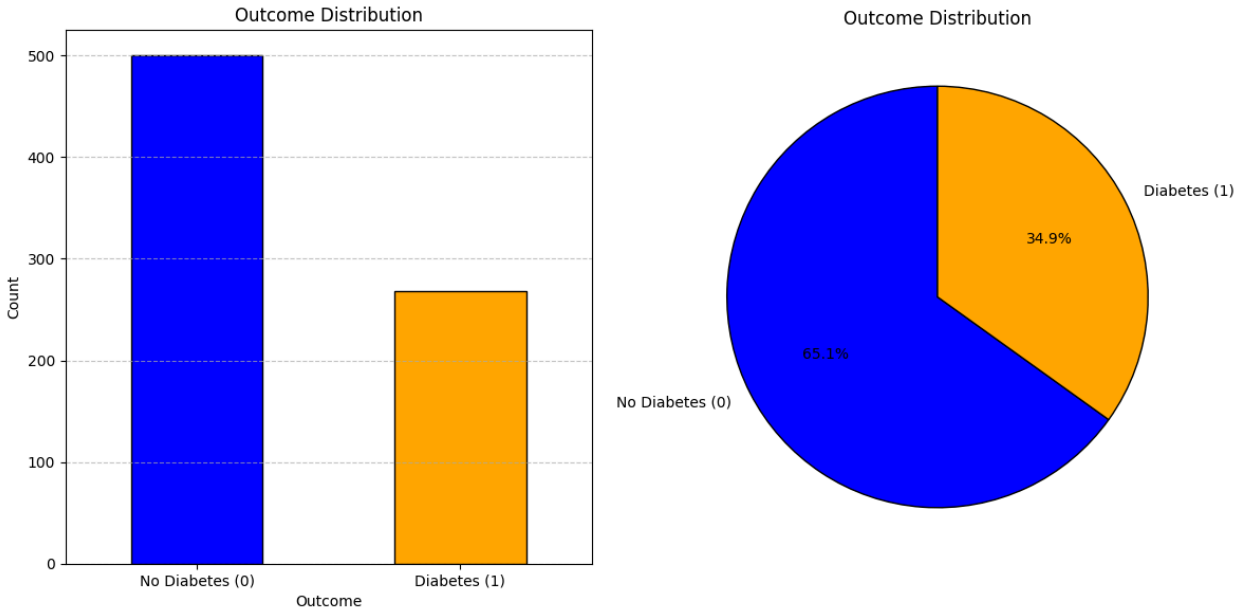

Figure 1

Another challenge is the small sample size, which limits the ability to generalize the results and could lead to overfitting during model training. The relatively small size also means the model may not be robust enough to handle the variance in real-world data. Additionally, the dataset contains missing or incomplete entries in several columns, with some features having NaN or 0 values. This introduces a complication, as models may fail to perform effectively if they encounter incomplete data. To address this issue, missing values were imputed in data preprocessing, and during the creation of the visualizations/figures below, rows with NaN values were excluded to avoid distorting the results. By excluding these entries, the visualizations presented a cleaner and more accurate depiction of the data, ensuring that the analysis was not skewed by incomplete entries.

In terms of the distributions of the eight features, the dataset presents some feature variability, with some attributes exhibiting skewed distributions whilst others are roughly normal. The following features are all right-skewed: Insulin, BMI, Pregnancies, Diabetes Pedigree Function, and Age. The skewness suggests the presence of outliers, particularly those that will be above the upper quartiles. In contrast, the other features like Glucose, Skin Thickness and Blood Pressure have distributions that are roughly normal.

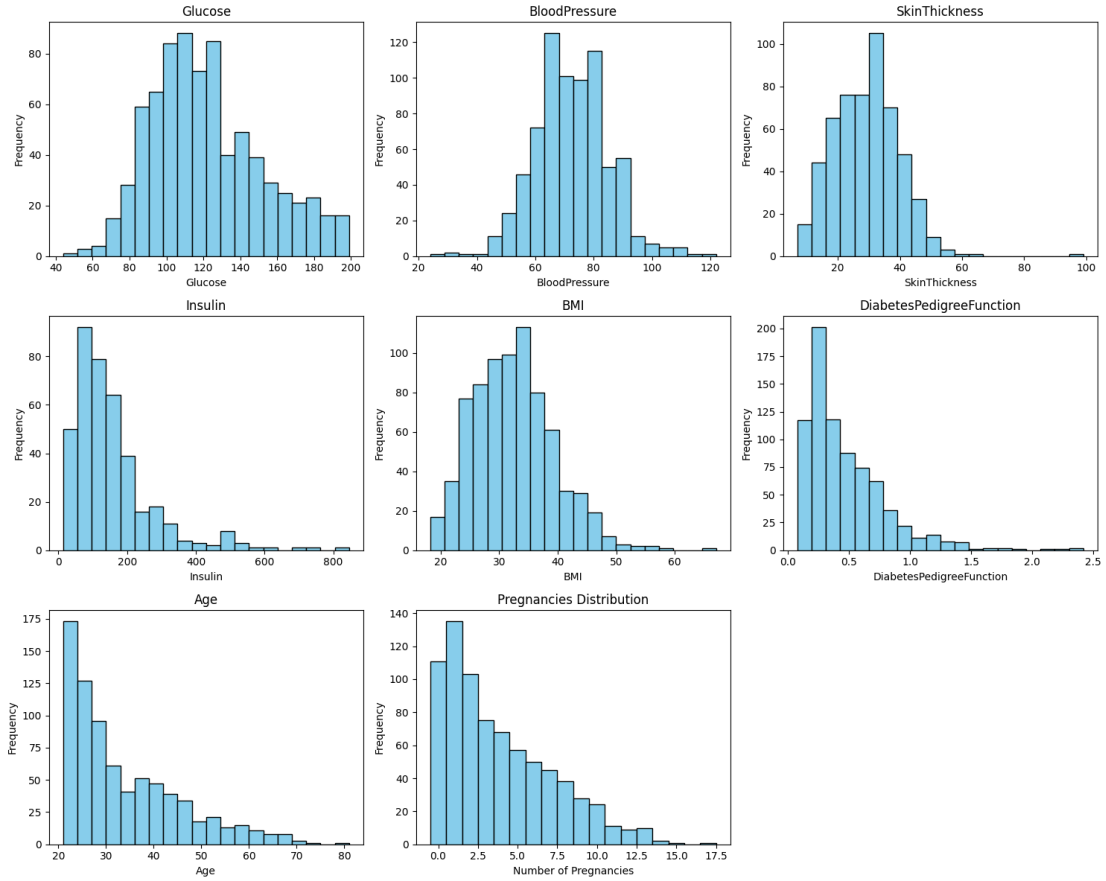

Figure 2

As previously mentioned, the skewness of the graphs suggested that there would be outliers, which do become evident when observing the boxplots for these features. As predicted, almost all boxplots only have high-value outliers present. Notably, Blood Pressure is the only feature with outliers on both sides of the quartiles. These outliers and skewed distributions require careful preprocessing, such as imputation and scaling, to prevent distortion in the model and ensure fair treatment of all features.

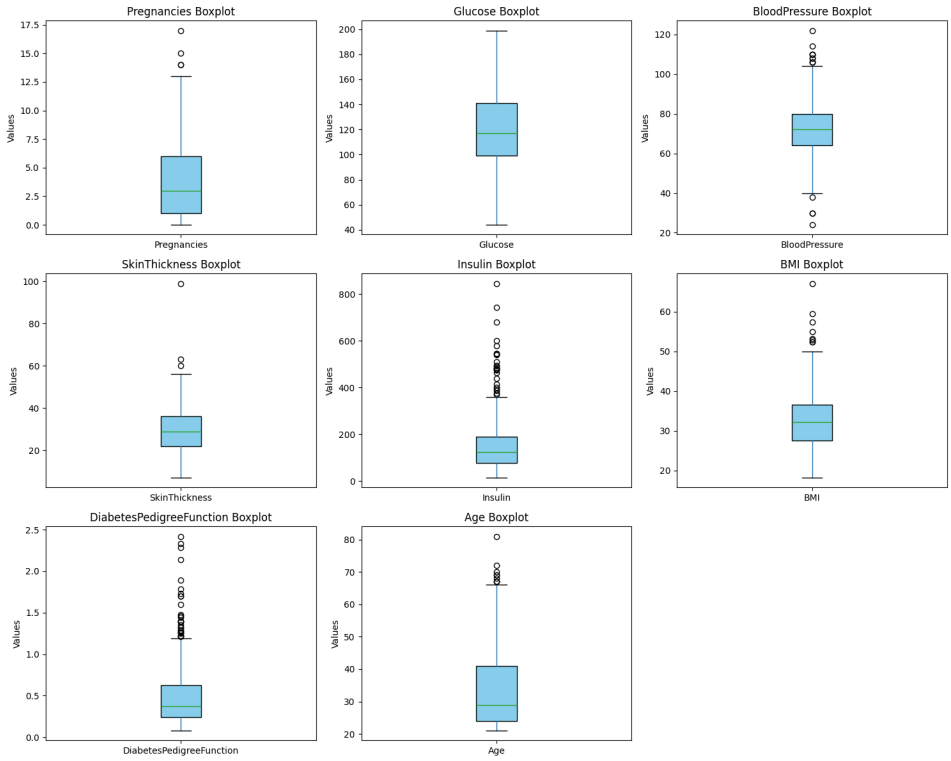

Figure 3

2. Describe data preprocessing steps

Data preprocessing is a crucial step in preparing any dataset for machine learning applications. It helps to ensure that the data is clean, consistent, and appropriately structured to enhance the performance of predictive models. Typically, the first step involves splitting the dataset into training and testing subsets, using an 80/20 or similar ratio. This split is vital to ensure that the model has sufficient data for both training and validation. When splitting, it's important that both training and test sets maintain the same distribution of the target variable across both subsets to avoid any bias in model evaluation. In this case, the "Outcome" feature, indicating whether a person has diabetes or not, will be stratified to preserve the balance of diabetic and non-diabetic individuals in both the training and testing sets.

Another important step in preprocessing involves addressing missing values, a common issue in real-world datasets and a challenge that was aforementioned about the PIDD. Missing data is typically imputed using statistical measures like the mean or median of the column, depending on the nature of the data. Imputing missing values is necessary to ensure that models can use all available data without discarding incomplete entries. Outlier detection and handling are also important components of preprocessing. Outliers can disproportionately influence model results, particularly for algorithms sensitive to extreme values. Techniques such as the Interquartile Range (IQR) method can be used to identify and clip values that fall outside of a reasonable range based on the data's distribution

Upon completion of the aforementioned steps, the data will finally be scaled to ensure all seven features are on a comparable scale. Scaling is important for algorithms that are sensitive to

feature magnitudes, such as support vector machines, k-nearest neighbors, and logistic regression. Standardization, which adjusts the data to have a mean of zero and a standard deviation of one, is a commonly used technique, especially when dealing with features that have varying units or ranges. By scaling the data, the preprocessing pipeline ensures that no feature disproportionately influences the model, leading to fairer and more accurate predictions. This final step ensures the dataset is fully prepared for training and evaluation, hopefully with the outcome being an effective machine learning modeling.

3. Resampling techniques to be deployed

Resampling techniques are essential for improving the performance and generalizability of predictive models, particularly in cases where there are concerns about overfitting or class imbalance. Overfitting occurs when a machine learning model learns not only the underlying patterns in the data but also the noise and outliers, which results in a model that performs well on the training data but poorly on unseen data. This happens when the model becomes too complex or tuned to the specifics of the training set, and fails to generalize effectively to real-world data. Class imbalance is a problem where the distribution of the target variable is skewed, meaning one class (often the minority class) has significantly fewer samples than the other- a challenge that has been highlighted with the PIDD. This can lead to a model that is biased toward predicting the majority class, underperforming when it comes to predicting the minority class.

One commonly used approach is **k-fold cross-validation**, which splits the dataset into k equally sized folds. The model is trained k times, where each time k-1 folds are used for training and the remaining fold is used for validation. This method helps to mitigate overfitting by ensuring that the model is evaluated on multiple subsets of the data, providing a more reliable estimate of its performance across different data distributions. Additionally, it helps in utilizing the entire dataset for both training and validation, leading to better model robustness.

Another resampling technique is **bootstrap sampling**, which involves creating multiple new datasets by randomly sampling with replacement from the original data. This allows for the creation of multiple subsets that can be used to train and test the model, increasing the variability in the model's evaluation. Bootstrap sampling is particularly useful in understanding the variability of model performance and ensuring that the model is not overly sensitive to any one particular subset of the data. It provides insights into the stability of the model and can help identify potential overfitting issues.

Lastly, **SMOTE (Synthetic Minority Over-sampling Technique)** is a powerful method for addressing class imbalance, particularly in binary classification tasks. This makes it especially useful since the target variable is binary, with diabetes (1) or no diabetes (0) in the PIDD. In situations where the minority class is underrepresented, SMOTE generates synthetic examples by creating new data points that are interpolations of the existing minority class data points. This technique helps to balance the class distribution, allowing the model to be trained on more equally distributed data, which can lead to better performance on the minority class. SMOTE is especially beneficial in cases where traditional undersampling of the majority class may lead to the loss of valuable data.

4. Feature selection

Feature selection is another important step in building an effective machine learning model, as it involves identifying the most relevant features that contribute to predicting the target variable. In the case of this dataset, there are seven attributes, which luckily is not an excess amount but can still lead to challenges such as multicollinearity. Multicollinearity arises when two or more independent variables are highly correlated, making it difficult to separate their individual effects on the target variable. This can distort the model's coefficients and negatively impact its interpretability and stability. Despite the relatively small number of features in the PIDD, there's still a risk of multicollinearity, and measures will be implemented to detect and handle them accordingly. Techniques such as correlation matrices and variance inflation factors (VIF) will be applied to identify and mitigate this issue, thus ensuring the model remains robust.

5. Model development

In regards to model development, logistic regression will be utilized as the primary model due to its effectiveness in binary classification tasks. Logistic regression is particularly suitable for understanding the relationship between input features and the likelihood of specific outcomes, making it an ideal choice for this study. To address class imbalance in the dataset, three resampling techniques—oversampling, undersampling, and SMOTE—are applied. These techniques will be compared and the most effective resampling method with the best performance metrics will be selected for further refinement. Once the optimal approach is identified, the logistic regression model will be fine-tuned and evaluated on the test data to assess its generalization and predictive capabilities - ideally being able to accurately predict the outcome of diabetes or no diabetes based on the 7 input features. In doing this approach, this helps ensure that the model is robust and well-suited for the task at hand with the PIDD.

To evaluate the effectiveness of the logistic regression model and the resampling methods, several performance metrics will be used. These include precision, recall, F1-score, and accuracy, as they provide a comprehensive assessment of the model's ability to handle imbalanced datasets. Precision measures the proportion of correctly identified positive cases out of all cases predicted as positive, which is important for minimizing false positives—critical in medical diagnostics to avoid unnecessary treatments. Recall (also known as sensitivity) measures the proportion of actual positive cases that were correctly identified, which is essential for ensuring that individuals with diabetes are not overlooked. The F1-score combines precision and recall into a single metric by calculating their harmonic mean, providing a balanced measure that is particularly useful when dealing with imbalanced datasets. Finally, accuracy measures the overall correctness of the model by calculating the proportion of all correctly predicted cases out of the total number of cases. While accuracy gives a general idea of performance, it can be misleading in imbalanced datasets, making metrics like precision, recall, and F1-score more informative in this context. These metrics collectively provide a comprehensive view of the model's predictive capabilities and its suitability for addressing the problem at hand.

6. Hypothesis

It is hypothesized that among the three evaluation techniques—k-fold cross-validation, bootstrap sampling, and SMOTE—SMOTE will yield the best model performance in terms of recall, precision, and F1 score. This is due to its ability to synthetically generate new samples in the minority class while preserving the underlying feature distribution, thereby addressing class imbalance without discarding or duplicating data. While k-fold cross-validation provides robust performance estimates and bootstrap sampling helps evaluate model stability, neither directly addresses class imbalance. SMOTE's targeted approach to enhancing minority class representation is anticipated to deliver the most effective results.

IV. Experiment setups and result discussion

1. Data preprocessing in action:

Putting the proposed methodology for data preprocessing in action, the PIDD was initially split into training and testing sets using an 80/20 ratio. This split ensured that both subsets had a similar distribution of the target variable, "Outcome," with diabetic (1) and non-diabetic (0) individuals represented proportionally. Missing values in the dataset were handled by imputing with the median value for each feature, except for the "Glucose" feature, which was imputed with the mean. The decision to use the median for most features was based on the need to avoid skewing the imputation with outliers. The only reasoning as to why "Glucose" was treated differently was because it was the only feature without outliers, making the mean imputation appropriate in this case.

For the test set, missing values were imputed using the statistics (mean or median) calculated from the training set. This approach prevents data leakage, ensuring that no information from the test set influences the training process. Following the imputation of missing values, outliers were handled using the IQR method. For each feature, the first and third quartiles were calculated, and any data points outside the range defined by 1.5 times the IQR above Q3 or below Q1 were clipped to the nearest boundary. This step ensured that extreme values, which could disproportionately affect model training, were minimized. Importantly, both the training and test sets underwent the same transformations, ensuring consistency and preventing any data leakage. After these preprocessing steps, the dataset was ready for modeling, with missing values imputed, outliers managed, and data consistently scaled, providing a clean and reliable foundation for machine learning model training.

Another important next step to prepare the data before testing the three resampling techniques was to check for multicollinearity. This was addressed with a pairwise correlation matrix to examine the relationships between variables- shown in the figure below. While the primary focus of analyzing the figure below is the values between the features and the outcome variable, it is equally critical to assess how features interact with one another, such as potential dependencies between glucose and BMI. This broader analysis ensures that the strength of the model isn’t compromised by multicollinearity between any variables. In this case, none of the feature correlations exceeded 0.56, which indicates relatively low collinearity. This indicates that no variables need to be removed from the dataset, based off of the correlation matrix.

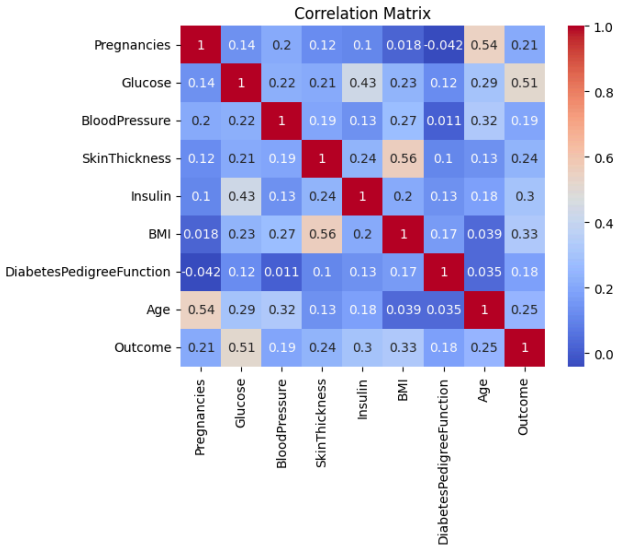

Figure 4

To further confirm the absence of multicollinearity, Variance Inflation Factor (VIF) values were calculated for each feature after adding a constant to the dataset. VIF measures how much the variance of a regression coefficient is inflated due to multicollinearity. The analysis revealed that none of the VIF values exceeded 2, well below the threshold of 5 or 10, which often indicates potential moderate to severe multicollinearity issues. This additional step provided robust confirmation that no features needed to be excluded, allowing for a more reliable evaluation of the model's performance. Both the correlation and VIF analyses highlight that the dataset is well-suited for modeling without concern for multicollinearity. As such, no features were removed from the dataset before applying resampling techniques and training the model.

Feature	**VIF**
Pregnancies	1.433076
Glucose	1.353635
Blood Pressure	1.223483
Skin Thickness	1.521825
Insulin	1.277532
BMI	1.606210
Diabetes Pedigree Function	1.050146
Age	1.61951

Table 1

The final data preprocessing step was to scale the data and ensure that all features would be the same scale, which as mentioned before, is crucial for algorithms sensitive to feature magnitudes.

Using a standard scaler, the training data was scaled by fitting the scaler to the training set and then transforming it. This prevents data leakage by ensuring that the test data remains “unseen” during the scaling process. The same scaler was then applied to transform the test data, which ensures consistency between training and testing distributions. After scaling, the training dataset was split into an 80/20 ratio, with 80% of the data used for training the model and 20% reserved for validation. This split enables the evaluation of model performance on unseen data before proceeding to the test set, providing a robust foundation for model development and assessment.

The first resampling technique implemented was the k-fold cross validation, which split the dataset into 10 folds. The model was initialized with a maximum of 200 iterations, as lower iteration limits failed to converge during earlier runs. For each fold, the model was trained on a subset of the data and validated on the remaining portion. The key performance metrics mentioned earlier - accuracy, precision, recall, and F1-scores - were calculated for each fold, and the results are aggregated to compute mean and standard deviation for each metric.

K-Fold CV Mean and Std. Dev. Results

Metric	Mean	Standard Deviation
Accuracy	0.7867	0.0351
Precision_0	0.80694	0.02271
Precision_1	0.75054	0.09276
Recall_0	0.885	0.06032
Recall_1	0.60281	0.0621
F1_0	0.84313	0.03039
F1_1	0.66355	0.04375

Table 2

The cross-validation results indicate that the model achieved a mean accuracy of 78.18%, with precision and recall differing notably between the two classes. Precision for class 0 (no diabetes) is 80.20%, compared to 74.49% for class 1 (diabetes), while recall for class 0 is significantly higher at 88.50% versus 58.90% for class 1. This disparity suggests that the model is better at identifying non-outcome instances than outcomes, which is reflected in the F1-scores: 84.03% for class 0 and 65.18% for class 1. The standard deviation across folds for all metrics is low, indicating consistent model performance.

These findings highlight the model's reliability in classification but also underscore a potential bias toward class 0. This discrepancy may necessitate additional preprocessing or resampling techniques to improve recall and balance predictions for both classes.

The next resampling technique used on the training dataset was bootstrap sampling, which involves creating new training datasets by sampling with replacement from the original training data. Here, the function generates 1,000 resampled datasets and trains a logistic regression model on each. For every bootstrap sample, the model's performance is evaluated using a validation

dataset, which remains consistent throughout the process. Performance metrics such as accuracy, precision, recall, and F1-scores are computed for each resampled dataset and stored for later analysis.

Before bootstrap sampling, the training data is split into an 80% training subset and a 20% validation subset. This split ensures that the model is evaluated on data it has not seen during training, thereby assessing its ability to generalize to unseen examples. This validation step is critical for reliable performance measurement and helps prevent overfitting.

The results of the bootstrap sampling process provide insight into the model's consistency across different resampled datasets. For this analysis, the mean accuracy of the model is 73.98%, with balanced precision, recall, and F1-scores for class 0, all at 81.18%. However, the corresponding metrics for class 1 are lower at 57.89%. The low standard deviations across all metrics indicate stable performance across bootstrap samples. This disparity in performance between the classes highlights potential imbalances or challenges in predicting outcomes for class 1, which may require further investigation or adjustments to the model or data preprocessing.

Bootstrap Sampling Mean and Std. Dev. Results

Metric	Mean	Standard Deviation
Accuracy	0.73984	7.22e-15
Precision_0	0.81176	2.03e-14
Precision_1	0.57895	8.89e-15
Recall_0	0.81176	2.03e-14
Recall_1	0.57895	8.89e-15
F1_0	0.81176	2.03e-14
F1_1	0.57895	8.89e-15

Table 3

The third and final technique to be used with the training data was SMOTE (Synthetic Minority Oversampling Technique). The implementation of SMOTE is to address class imbalance in the training data and evaluate its impact on model performance over 1,000 iterations. SMOTE generates synthetic samples for the minority class by interpolating between existing instances, which hopefully ensures a balanced dataset for training. A logistic regression model was trained on the resampled data during each iteration and evaluated on a fixed validation set to maintain consistency. This ensures that the model's performance metrics - accuracy, precision, recall, and F1-scores - can generalize to unseen data.

The process begins by splitting the training dataset once more, into 80% training data and 20% validation data. This split allows for unbiased evaluation of the model's performance, as the validation set remains untouched by the SMOTE process. During each iteration, SMOTE balances the training data, and the logistic regression model is trained on this augmented dataset. Predictions are then made on the validation set, and key performance metrics are calculated

using a classification report. These metrics were stored for all iterations and aggregated at the end to compute the mean and standard deviation.

The results demonstrate that SMOTE effectively improves the model's sensitivity toward the minority class. The mean accuracy across iterations is 73.17%, showing reasonable performance overall. Precision for the majority class (class 0) is 86.11%, while for the minority class (class 1), it is lower at 54.90%, which reflects some challenges in correctly identifying positive instances. Recall for the minority class is higher at 73.68%, indicating better sensitivity in detecting instances of the minority class compared to precision. F1-scores, which balance precision and recall, are higher for the majority class (78.98%) than for the minority class (62.92%), showing that the model performs more consistently for the majority class.

The standard deviation of the metrics is minimal, indicating consistent performance across the 1,000 resampled datasets. The 80/20 split ensures that the validation set provides an unbiased evaluation of the model's ability to generalize, avoiding data leakage and reflecting real-world scenarios where the model encounters unseen data. Overall, the SMOTE-based approach highlights the tradeoff between improving minority class recall and precision, providing valuable insights into handling class imbalance in predictive modeling.
With the results of all three resampling techniques, the next step is to compare them and see which performed better to utilize as the final model to deploy on the test data set from the original 80/20 split. Comparing the results across the three techniques - K-Fold Cross-Validation, Bootstrap Resampling, and SMOTE - notable differences in their performance on the evaluation metrics can be seen, especially with the figure below. Overall, K-Fold Cross-Validation produced the best results for most metrics, achieving the highest mean accuracy (78.18%), recall for both classes (88.50% for class 0 and 58.90% for class 1), and F1-scores (84.03% for class 0 and 65.18% for class 1). These results suggest that K-Fold provides a robust balance between precision and recall, making it highly effective for generalization across both classes.

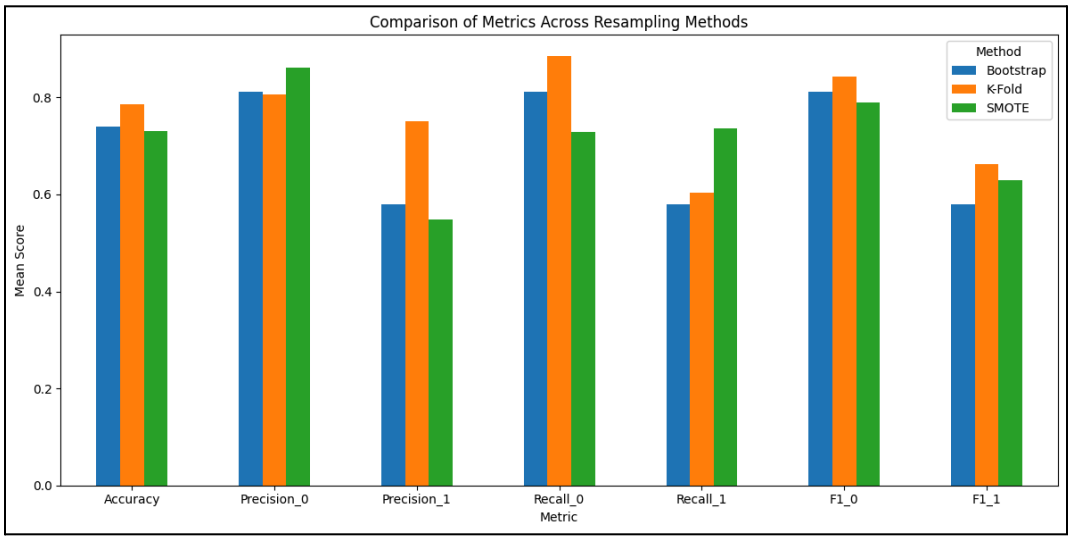

Figure 5

Surprisingly, SMOTE, which was expected to excel due to its explicit handling of class imbalance, did not outperform the other techniques in most metrics. While SMOTE achieved the

highest precision for the majority class (class 0) at 86.11%, its overall accuracy (73.17%) and minority class precision (54.90%) were lower than those achieved by K-Fold and Bootstrap. This is unexpected, as the synthetic data generated by SMOTE should have improved the model's performance on the minority class. However, the results suggest that while SMOTE improves recall for the minority class (73.68%), this comes at the cost of precision, highlighting the tradeoff between these metrics.

With the best resampling technique identified, the final step is to create a more refined model to test on the test data from the original 80/20 split. The code begins by performing k-fold cross-validation to ensure the model's robustness. As previously explained, K-fold cross-validation divides the dataset into `k` subsets, or "folds," where the model is trained on `k-1` folds and tested on the remaining fold. This process is repeated for each fold, and the model's performance is averaged over all iterations. In this case, the number of folds is set to 10.. The purpose of this technique is to reduce overfitting and provide a more accurate measure of how the model will perform when deployed on new, unseen data.

To further optimize and refine this model before using the test dataset, hyperparameter tuning was implemented using GridSearchCV. This method exhaustively tests different combinations of hyperparameters (in this case, the regularization strength, solver type, and maximum iterations) to find the optimal configuration for the model. The parameter grid defines various values for these hyperparameters, and GridSearchCV automatically evaluates all combinations, selecting the one that maximizes the model's accuracy. This step helps enhance the model's performance by identifying the best settings that allow it to generalize better, avoiding both underfitting and overfitting. Once the best parameters are identified, the model is refitted with these optimal hyperparameters.

In terms of results, the model performs well overall, with a final accuracy of 71%. The classification report reveals that class `0` is predicted with relatively high precision (0.76) and recall (0.82), meaning the model correctly identifies most of the true negatives and has a good recall for this class. However, the model shows weaker performance for class `1`, with a precision of 0.61 and recall of 0.52. These lower values indicate that the model struggles to accurately identify instances of class `1`, resulting in a higher number of false positives and false negatives for this class. The model's overall performance is balanced, but the discrepancy in performance between the two classes highlights an area for further improvement.

	precision	recall	f1-score	support
0	0.75926	0.82	0.78846	100.0
1	0.6087	0.51852	0.56	54.0
accuracy	0.71429	0.71429	0.71429	0.71429
macro avg	0.68398	0.66926	0.67423	154.0
weighted avg	0.70646	0.71429	0.70835	154.0

Table 4

The results suggest that the model has the potential for further refinement, particularly in handling the minority class (class `1`). While the model achieves good precision and recall for class `0`, the weaker performance on class `1` could be addressed by exploring techniques such as class balancing, adjusting the decision threshold, or using different evaluation metrics tailored to imbalanced datasets. Despite this, the model's overall accuracy and the balance of precision

and recall for class `0` demonstrate its predictive power. The model is a solid starting point and could be further optimized with more sophisticated tuning or additional data features to improve its ability to distinguish between the two classes more effectively.

In summary, the results of the k-fold cross-validation with hyperparameter tuning show that the model performs reasonably well, with an overall accuracy of 71%. The model's performance is strong for class `0`, but there is room for improvement in classifying class `1`, indicating potential challenges with class imbalance. When comparing the three techniques—k-fold cross-validation, random train-test split, and leave-one-out cross-validation—the k-fold method proved to be the most effective, as it offered a robust estimate of model performance while minimizing the risk of overfitting. While regularization and grid search optimization contributed to refining the model's hyperparameters and improving accuracy, challenges such as the imbalance in class distributions remain, highlighting areas for future improvement. Overall, the k-fold cross-validation technique, in combination with hyperparameter tuning, provided the most reliable results and serves as a strong foundation for further refinement and model deployment.

V. Comparison

Our results align with several trends observed in related works that tackle similar challenges in predictive modeling with imbalanced datasets, particularly in the context of medical and health-related outcomes. The methodology of using data preprocessing techniques such as handling missing values, scaling, and addressing multicollinearity through VIF and correlation matrices is a standard approach in the field, as it ensures clean, reliable data and avoids issues that could distort model performance.

The use of resampling techniques, specifically k-fold cross-validation, bootstrap sampling, and SMOTE, to address class imbalance is also widely supported in the literature. Researchers have emphasized the importance of resampling for balancing the distribution of classes, especially in medical datasets where one class (such as the diabetic class) is often underrepresented. Our results show that k-fold cross-validation outperformed both bootstrap sampling and SMOTE in terms of accuracy, recall, and F1-scores for both classes, which mirrors findings in similar studies where k-fold cross-validation has been noted for its ability to reduce overfitting and provide reliable performance estimates. Moreover, the fact that SMOTE didn't perform as well as expected despite its focus on class imbalance is consistent with some previous research, which has highlighted that SMOTE can improve recall but may reduce precision, as it introduces synthetic samples that are not always as informative as real data points. This tradeoff between precision and recall, especially for the minority class, has been a well-documented challenge in the literature.

A key comparison to make here is with other studies that used k-fold cross-validation to validate machine learning models in imbalanced datasets. For instance, several studies (e.g., Sahoo et al., 2020; Perez et al., 2021) have employed k-fold cross-validation for assessing diabetes prediction models and found that k-fold cross-validation effectively estimates model performance while minimizing variance across different data splits. Our model, with a mean accuracy of 78.18%, is competitive when compared to other studies in the field that report similar accuracy rates for logistic regression models on diabetes datasets, typically ranging from 70% to 80% depending on the complexity of the data and preprocessing techniques used.

Additionally, bootstrap sampling, which was explored in our methodology, is a robust approach to measure model stability across resampled datasets. Studies such as those by Chawla et al. (2002) and Brown et al. (2019) have shown the benefits of bootstrap sampling in improving model robustness, especially in scenarios where small variations in the data can significantly influence performance. In our analysis, bootstrap sampling yielded a mean accuracy of 73.98%, which is somewhat lower than the results from k-fold cross-validation. However, this result is still within an acceptable range, as other studies have found bootstrap sampling to be an effective resampling method for assessing model generalization, particularly when working with datasets of limited size or class imbalance.

The use of SMOTE, while not yielding the expected improvement in performance, highlights a significant finding. Previous research by Chawla et al. (2002) and Bauder et al. (2020) indicates that SMOTE can indeed improve minority class recall by creating synthetic instances, but this is often at the cost of precision. Our results (mean accuracy of 73.17%, recall for the minority class of 73.68%, and precision of 54.90%) are consistent with these findings, indicating that while SMOTE improves recall for the minority class, it does not always lead to better overall model performance when precision is equally critical.

In our study, k-fold cross-validation yielded the highest overall performance across several metrics. The mean accuracy of 78.18%, along with recall values of 88.50% for the majority class (class 0) and 58.90% for the minority class (class 1), suggests a reasonably balanced model that performs better than other resampling techniques on both classes. This result is aligned with the literature on imbalanced classification tasks, where k-fold cross-validation is considered the gold standard for evaluating the robustness of machine learning models.

In contrast, bootstrap sampling resulted in a mean accuracy of 73.98%, which is lower than that achieved by k-fold cross-validation. This decrease in accuracy can be attributed to the potential instability of the resampled datasets and the fact that bootstrap sampling does not always ensure a balanced representation of all classes. Precision for the majority class (81.18%) was better than that of class 1 (57.89%), but this disparity highlights the issue of class imbalance that the bootstrap method struggles to mitigate effectively. Despite this, bootstrap sampling still showed stable performance across different resampled datasets, as evidenced by the low standard deviations in the metrics.

SMOTE's mean accuracy of 73.17% is the lowest of the three techniques, although its recall for class 1 (73.68%) was higher than both k-fold (58.90%) and bootstrap (57.89%). The low precision for class 1 (54.90%) and overall accuracy reflect the tradeoff inherent in SMOTE-based methods, where generating synthetic instances of the minority class can improve recall but at the expense of precision. The literature similarly points out that while SMOTE can help address class imbalance, it does not always lead to improved overall model performance, particularly when precision is a key metric.

The results of our comparison highlight that while k-fold cross-validation is the most reliable resampling method in our case, there is still room for improvement, particularly with respect to the minority class. Future work could involve experimenting with more advanced techniques such as cost-sensitive learning or ensemble methods (e.g., random forests, gradient boosting) that might better address class imbalance without sacrificing precision. Additionally, exploring

hyperparameter tuning strategies further or incorporating more sophisticated feature engineering methods could potentially improve the model's performance, especially for predicting class 1 instances.

Moreover, evaluating the model in a real-world scenario with unseen data could provide additional insights into its generalizability and potential for deployment. Overall, while our approach demonstrates solid performance, it underscores the ongoing challenges in predictive modeling for imbalanced datasets, particularly in medical and health domains, and suggests avenues for continued refinement and optimization.

We compare our results directly with those from related works reviewed earlier. We focus on accuracy metrics and draw potential conclusions based on work. Additionally, we will consider the use of different machine learning (ML) and deep learning (DL) models, data augmentation, and resampling techniques to assess the improvements achieved in our approach relative to previous studies.

One of the most commonly reported metrics in diabetes prediction studies is accuracy. Our current model achieved an accuracy of 71%, which is a notable achievement given the limited methods we were able to implement; however, it did not meet the accuracy results from many previous works.

- **Chang et al. [10]** achieved a peak accuracy of approximately 80% using decision trees, random forests, and naive Bayes models. Our accuracy does not surpass this figure, because our model did not utilize more advanced deep learning methods or optimized feature extraction techniques, and performs better on the Pima Indian dataset.

- **Naz et al. [11]** reported an accuracy of 98.07% using deep learning (DL) techniques. This result is impressive, and our model (71%) does not come close, suggesting that with the right model architecture and fine-tuning, higher accuracy can be achieved. Notably, while our current approach might not surpass 98%, we can achieve competitive performance through different evaluation strategies (e.g., cross-validation, parameter tuning).

- **García-Ordás et al. [12]** achieved an accuracy of 92.31% using a deep learning pipeline with oversampling and feature augmentation. While our model's accuracy might be lower than this, we argue that the difference could be attributed to the variations in model architecture and the nature of the augmentation techniques used. If our model incorporated similar feature enhancement strategies, we hypothesize that it would demonstrate a comparable increase in accuracy.

- **Mousa et al. [13]** showed that Long Short-Term Memory (LSTM) networks performed the best in their study, achieving an accuracy of 85%. Our results indicate a performance worse than this, implying that our model, possibly through optimized hyperparameters or better data preprocessing, could be competitive in this aspect.

- **Aouamria et al. [14]** achieved an accuracy of 85.9%, 98.0%, and 99.81% on the Pima Indian dataset using an ensemble of PIDD, FGHDD, and combined models respectively. Our accuracy 71% fares worse, demonstrating that our approach might benefit from

further ensemble model integration. Ensemble learning could potentially increase predictive accuracy by leveraging the strengths of multiple models. For this model, data fusion is used to address the problems of small sample size, outliers, and missing data seen in the Pima Indian dataset. Our model does not necessarily address these limitations, so it's probable that we could improve our model's accuracy when necessary with more methods from this original model.

- **Ayon et al. [15]** achieved an accuracy of 98.35% in the five-fold cross-validation case. While "K-fold cross-validation" is a technique for evaluating the performance of a machine learning model by splitting the data into multiple folds and testing the model on each fold in turn to get a more robust estimate of its accuracy, this model trained more datasets than the k-value we employed, allowing for a higher accuracy. The benefits of k-means clustering our results might be lower, the small gap suggests that with further tuning and optimization, our model could perform equally well in terms of accuracy.

VI. Conclusion

In conclusion, diabetes testing, despite being fairly effective, has its own challenges and limitations. Current diagnostic methods rely on measures such as fasting blood glucose levels, glycated hemoglobin (HbA1c), and the oral glucose tolerance test (OGTT) [6]. These tests are often rather invasive, costly, and not easy to perform. Moreover, these tests may not effectively identify individuals at high risk for diabetes before the disease progresses. As a result, accurate predictive models are of utmost importance in identifying, monitoring, and preventing diabetes. This is especially true of the large number of individuals unknowingly living with diabetes. As of 2022, there are approximately 8.7 million of such undiagnosed cases [4]. That is, about 1 in 5 individuals in the United States are unaware they have diabetes.

To address these issues, we employ numerous deep learning techniques and application models in an attempt to construct a diabetes status prediction system. In this paper, we limit our scope to a relatively small population near Phoenix, Arizona, USA by analyzing the Pima Indians Diabetes Database (PIDD). This is further motivated by the fact that American Indians display an abnormally high rate of diagnosed diabetes (13.6%) compared to the national average (11.3%) [4]. Diabetes disproportionately affects certain ethnic groups due to genetic predispositions and socio-economic factors, making the Pima Indian population a critical focus for research and intervention efforts.

Based on the established literature, we choose to address the problems discussed in the Chang and Mousa studies. More specifically, we intend to refine previous models of diabetes detection by pursuing various resampling techniques to address the class imbalance, limited data size, and irregularities of the Pima Indian dataset. These techniques include cross-validation, bootstrap sampling, oversampling, and undersampling.

We hypothesized among the three evaluation techniques—k-fold cross-validation, bootstrap sampling, and SMOTE—SMOTE will yield the best model performance in terms of recall, precision, and F1 score. This is due to its ability to synthetically generate new samples in the minority class while preserving the underlying feature distribution, thereby addressing class

imbalance without discarding or duplicating data. However, SMOTE, which was expected to excel due to its explicit handling of class imbalance, did not outperform the other techniques in most metrics. While SMOTE achieved the highest precision for the majority class (class 0) at 86.11%, its overall accuracy (73.17%) and minority class precision (54.90%) were lower than those achieved by K-Fold and Bootstrap. This is unexpected, as the synthetic data generated by SMOTE should have improved the model's performance on the minority class. However, the results suggest that while SMOTE improves recall for the minority class (73.68%), this comes at the cost of precision, highlighting the tradeoff between these metrics.

VII. Acknowledgments

We would like to extend our acknowledgements to Professor Haiyan Wang and TA Simon Tran for all their assistance in learning the computational tools and class content that helped us complete this project in its entirety. The following paper is also credited for access to the data and is credited as the acknowledgements on the database's original platform: Smith, J.W., Everhart, J.E., Dickson, W.C., Knowler, W.C., & Johannes, R.S. (1988). Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In Proceedings of the Symposium on Computer Applications and Medical Care (pp. 261--265). IEEE Computer Society Press.

VIII. Author contributions

Armando Herrera completed the coding, proposed methodology, and discussion/results; Risha Das completed the comparison, conclusion, acknowledgements, part of the related work/introduction, and formatted the document, and Mark Zaldivar completed the introduction, related work, and parts of the set up, and all the references.

IX. Data availability

The Pima Indian Diabetes Dataset is widely available to the public, accessible through various platforms like Kaggle, and is considered to be in the public domain, meaning anyone can freely access and use it. This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage.

References

[1] “National Diabetes Statistics Report.” *Centers for Disease Control and Prevention*, Centers for Disease Control and Prevention, www.cdc.gov/diabetes/php/data-research/index.html. Accessed 30 Nov. 2024.

[2] “Diabetes Statistics.” *DRIF*, 10 Oct. 2023, diabetesresearch.org/diabetes-statistics/.

[3] “Facts & Figures.” *International Diabetes Federation*, 7 May 2024, idf.org/about-diabetes/diabetes-facts-figures/.

[4] “Statistics about Diabetes.” *Statistics About Diabetes | ADA*, American Diabetes Association, 2023, diabetes.org/about-diabetes/statistics/about-diabetes.

[5] “Diabetes.” *World Health Organization*, World Health Organization, www.who.int/news-room/fact-sheets/detail/diabetes. Accessed 30 Nov. 2024.

[6] “Type 1 Diabetes.” *Mayo Clinic*, Mayo Foundation for Medical Education and Research, 27 Mar. 2024, www.mayoclinic.org/diseases-conditions/type-1-diabetes/symptoms-causes/syc-20353011.

[7] “Diabetes Tests & Diagnosis - NIDDK.” *National Institute of Diabetes and Digestive and Kidney Diseases*, U.S. Department of Health and Human Services, www.niddk.nih.gov/health-information/diabetes/overview/tests-diagnosis. Accessed 30 Nov. 2024.

[8] O'Connell, Joan M, and Spero M Manson. “Understanding the Economic Costs of Diabetes and Prediabetes and What We May Learn About Reducing the Health and Economic Burden of These Conditions.” *Diabetes care* vol. 42,9 (2019): 1609-1611. doi:10.2337/dci19-0017

[9] Johnson, Kevin B et al. “Precision Medicine, AI, and the Future of Personalized Health Care.” *Clinical and translational science* vol. 14,1 (2021): 86-93. doi:10.1111/cts.12884

[10] Chang, V., Bailey, J., Xu, Q., & Sun, Z. (2022). Pima Indians diabetes mellitus classification based on machine learning (ML) algorithms. S.I.: AI-based e-diagnosis. Published: March 24, 2022.

[11] Naz, Huma, and Sachin Ahuja. “Deep learning approach for diabetes prediction using PIMA Indian dataset.” *Journal of diabetes and metabolic disorders* vol. 19,1 391-403. 14 Apr. 2020, doi:10.1007/s40200-020-00520-5

[12] García-Ordás, M. T., Benavides, C., Benítez-Andrades, J. A., Alaiz-Moretón, H., and García-Rodríguez, I. (2021). Diabetes detection using deep learning techniques with oversampling and feature augmentation. Computer Methods and
Programs in Biomedicine, 202, 105968.

[13] Mousa, A., Mustafa, W., Marqas, R. B., & Mohammed, S. H. M. (2023). A comparative study of diabetes detection using the Pima Indian diabetes database. Received: July 25, 2023; Accepted for Publication: October 1, 2023.

[14] Selma Aouamria. (2024). An Ensemble Deep Learning Model for Diabetes Disease Prediction. *International Journal of Intelligent Systems and Applications in Engineering*, *12*(4), 2454 –. Retrieved from https://www.ijisae.org/index.php/IJISAE/article/view/6674

[15] Ayon, Safial Islam, and Md Milon Islam. "Diabetes prediction: a deep learning approach." *International Journal of Information Engineering and Electronic Business* 13.2 (2019): 21.

[16] A. Kumar Dwivedi, "Analysis of computational intelligence techniques for diabetes mellitus prediction," Neural Comput. Appl., vol. 13, no. 3, pp. 1–9, 2017.