Aryan Jain
Kaytlyn Daffern
Tea Zawilak

Can Machine Learning Help Address Health Inequity in the Akimel O'odham Community?: Evaluating Predictive Models for Diabetes

I. Introduction

Diabetes mellitus, also referred to as diabetes, is a chronic disease that occurs either when the pancreas fails to produce adequate amounts of insulin, which is a hormone that regulates blood sugar, or the body cannot effectively utilize the insulin it produces. There are serious health risks associated with uncontrolled diabetes, such as hyperglycemia, a state of raised blood sugar that can lead to serious damage to the body over time, especially the nerves and blood vessels ("Diabetes" 2024). Other health problems associated with diabetes include eye disease, foot problems, gum disease, stroke, kidney disease, and skin infections("Diabetes Complications" 2024). Diabetes is a growing epidemic on a worldwide scale, and, since 2000, mortality rates from diabetes have been increasing. In 2021, it was the direct cause of 1.6 million deaths, 47% of these occurring to individuals younger than 70 years old ("Diabetes" 2024). The Akimel O'odham, also called the Pima, are an Indigenous people of the Americans living in central and southern Arizona, primarily along the Gila and Salt Rivers ("Pima | Native Americans, Arizona, Southwest" 1998). Unfortunately, the Pima of Arizona have one of the highest rates of diabetes in the world. A 2006 study found that 34.2% of men and 40.8% of women included in their sample have the disease(Schulz et al. 2006). This is significantly higher than the overall prevalence of diabetes in the United States, which was 11.6% of the population in 2021 ("Statistics about Diabetes | ADA" 2021). The American Indian Health and Diet Project suggests that this concerning prevalence may be due to the change in diet amongst modern-day Native Americans, as traditional foods, such as wild game and seasonal produce, are substituted in favor of sugary, fatty, and starchy foods (Combs 2024). Although diabetes can be managed or sometimes reversed, there is no cure, making preventative care extremely important when working with communities with high prevalences of the disease.

Because Pima Indians are such a high-risk population for diabetes, there is a need to create a reliable predictive model for the disease, as well as to identify the most important risk factors implicated in its development. Thus, this can inform preventative health measures by determining which interventions would be most impactful in lowering diabetes incidence rate. For example, if BMI is one of the most important features, then public health officials can allocate more resources towards programs that encourage weight management through diet and exercise. By using a dataset of medical information collected from Pima Indians Diabetes that includes features such as glucose levels, age, BMI, and blood pressure, this study aims to create a reliable predictive model for Type II diabetes, as well as identifying key risk factors for the disease. By analyzing the relationship between these attributes and diabetes occurrence using multiple logistic regression, we can provide insights into risk factors and support early diagnosis. The goal

of this predictive model is to push for preventative healthcare as the current system fails to address critical health issues within indigenous communities.

II. Related Work

Previous studies have explored the use of machine learning techniques to predict diabetes onset, demonstrating the potential of data-driven models to improve disease management (Cahn et al., 2020) (Lai et al., 2019). These models, however, have largely focused on general populations and may not generalize well to specific demographics like the Akimel O'odham. Shanker's work in 1996 showed the applicability of artificial neural networks for predicting diabetes in the Pima Indian female population, underscoring the need for population-specific models (Li & Fernando, 2016). More recently, studies have used electronic medical record data and gradient boosted trees to predict diabetes progression, achieving high predictive accuracy across diverse cohorts. (Cahn et al., 2020) (Lai et al., 2019). In 2022, Hounguè and Bigirimana performed an analysis on the Pima Indians Diabetes Database by using Decision Tree, SVM and Naive Bayes classification algorithms. They achieved the highest model accuracy when using Naive Bayes, which was 76.3%. Although we plan to use a different set of models, this research provides a strong foundation for our proposed machine learning approach towards analyzing diabetes prevalence in the Pima (Pélagie Hounguè and Bigirimana 2022).

III. Proposed Methodology

This analysis will utilize a dataset titled “Pima Indians Diabetes Database”, which is retrieved from the data science repository Kaggle.com and can be accessed through the following URL: Pima Indians Diabetes Database. This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. It consists of the information of 768 patients who are of Pima Indian heritage, female and at least 21 years old. The dataset consists of one outcome variable, which is whether or not the patient has diabetes, and eight medical predictor variables: number of pregnancies, plasma glucose concentration, blood pressure, triceps skin fold thickness, insulin level, body mass index, diabetes pedigree, and age. We will preprocess this data by removing any patients with missing features and impose appropriate feature scaling. We will then implement an exploratory data analysis to visualize the distributions of the predictor variables, Because we have a binary outcome variable and multiple continuous predictor variables, we will compare 3 different predictive models (k-Nearest Neighbors, Support-Vector Machine, Random Forest Classifier) to understand how each of the predictor variables contribute to the likelihood of developing diabetes, which can be utilized by healthcare organizations to create initiatives that can work to reduce the most significant risk factors in Pima Indian patients.

IV. Experiment Setup

In this experiment, we evaluated the effectiveness of machine learning classifiers for predicting diabetes using the Pima Indians Diabetes Dataset. We hand selected three models that we researched and determined to be well-suited for this dataset: Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), and Random Forests (RF). These

classifiers worked well with this dataset due to its smaller size, tabular structure, and binary classification task. The dataset's tabular structure, organized into rows representing patients and columns representing medical features, makes it compatible with these algorithms, which excel at numerical and categorical data processing. This dataset, with its straightforward structure and binary outcome of predicting diabetes, was a great match for the strengths of the chosen classifiers.

To ensure fair and reliable evaluation, careful attention was given to data preprocessing and feature scaling. The process started by separating the features from the target variable to clearly define predictors and outcomes. After that, the data was split into training (80%) and testing (20%) sets, allowing us to assess how well the models performed on new, unseen data while avoiding overfitting. Since there were no missing values, we didn't have to address any gaps in the dataset, simplifying the preparation phase. To ensure consistency across features, we standardized the data using the StandardScaler module, which adjusted it to have a mean of zero and a standard deviation of one. This step was especially important for k-NN and SVM, as these algorithms are sensitive to variations in the scale of input features. Overall, the experimental setup laid a strong groundwork for evaluating the models effectively.

For the Support Vector Machine classifier, we used a radial basis function (RBF) kernel to capture complex, non-linear patterns in the data. We also enabled probability estimates, which allowed us to calculate important metrics like the Receiver Operating Characteristic (ROC) curve and the Area Under the Curve (AUC). The SVM model achieved its predictions on the test set and was evaluated using accuracy, a classification report, and a confusion matrix. The ROC curve highlighted the model's ability to distinguish between diabetic and non-diabetic cases, with the AUC serving as a measure of overall performance.

The k-Nearest Neighbors classifier underwent a thorough hyperparameter tuning process using a grid search to find the optimal number of neighbors (k). We tested values ranging from 1 to 20, and once the best value of k was identified, the model was trained on the scaled data. Its performance was then evaluated using metrics such as accuracy, a classification report, and a confusion matrix. Probabilities for each prediction were computed to plot the ROC curve and calculate the AUC, providing a comprehensive visualization of the classifier's performance.

For the Random Forest model, 100 decision trees were used, with randomness controlled via a fixed seed for reproducibility. Unlike the other classifiers, the Random Forest model worked directly with the raw data and didn't require feature scaling, making it easier to implement. We trained it using 100 decision trees and ensured reproducibility by setting a fixed random seed. The model's predictions were evaluated with accuracy, a classification report, and a confusion matrix. We also calculated the ROC curve and AUC from the predicted probabilities, making it easy to compare its performance with the other models.

V. Expected Results

In this experiment, we expected each model to exhibit performance patterns consistent with their strengths and how they interact with the dataset's characteristics. Given the tabular structure, relatively small size, and binary classification nature of the Pima Indians Diabetes dataset, we anticipated that all three models—SVM, k-NN, and Random Forest—would produce reasonable accuracy and AUC values, though with some variation based on their design and assumptions.

We expected the SVM classifier to perform exceptionally well because of its strength in handling high-dimensional data and its ability to model complex, non-linear patterns. Using the RBF kernel, which is particularly effective for capturing intricate relationships, we believed SVM was well-suited to the characteristics of this dataset. We anticipated that SVM would produce a high AUC and accuracy by clearly separating cases of diabetes from non-diabetes. Recognizing that SVM is sensitive to the scale of input features, we standardized the data during preprocessing to ensure precise and reliable predictions

For the k-NN classifier, we anticipated solid performance, though perhaps not as strong as SVM or Random Forest. Its ability to identify patterns based on the proximity of data points suggested it could deliver reasonable results, especially with proper tuning of the number of neighbors (k). However, because k-NN relies on distance calculations, it can be affected by noise and the high dimensionality of the dataset. Despite these potential limitations, we still expected k-NN to achieve consistent and respectable AUC and accuracy scores.

We had high expectations for the Random Forest model due to its ability to capture both linear and non-linear relationships by combining multiple decision trees. Its use of randomness in selecting features and data samples enhances stability and reduces the risk of overfitting, making it particularly well-suited for this dataset. Additionally, unlike SVM and k-NN, Random Forest does not require feature scaling, which made it even more convenient to implement. We anticipated that its performance would rival SVM, with high AUC and accuracy, while maintaining consistent predictions across the training and test datasets.

We expected all three models to perform well on this classification task, with SVM and Random Forest likely being the strongest. Both models are good at handling complex relationships in data, making them well-suited for the binary classification problem at hand. SVM, using the RBF kernel, is great at capturing non-linear patterns, and Random Forest's ensemble method helps it handle both linear and non-linear data without overfitting. We anticipated these models would show strong performance, with high accuracy and AUC scores, allowing them to clearly distinguish between diabetic and non-diabetic cases.

Although we thought k-NN might not perform as well as SVM and Random Forest, we still expected it to give us useful results. As a distance-based model, k-NN can be more sensitive to noise and high-dimensional data, but with the right tuning of the number of neighbors (k), it should still provide valuable insights and produce a reasonable AUC. Our expectations were based on how well the models' strengths aligned with the dataset,

which was clean and straightforward. We were interested to see how closely the actual results would match our predictions and whether any model would exceed our expectations. Ultimately, the goal was to see how each model could contribute to our understanding of predicting diabetes.

VI. Results

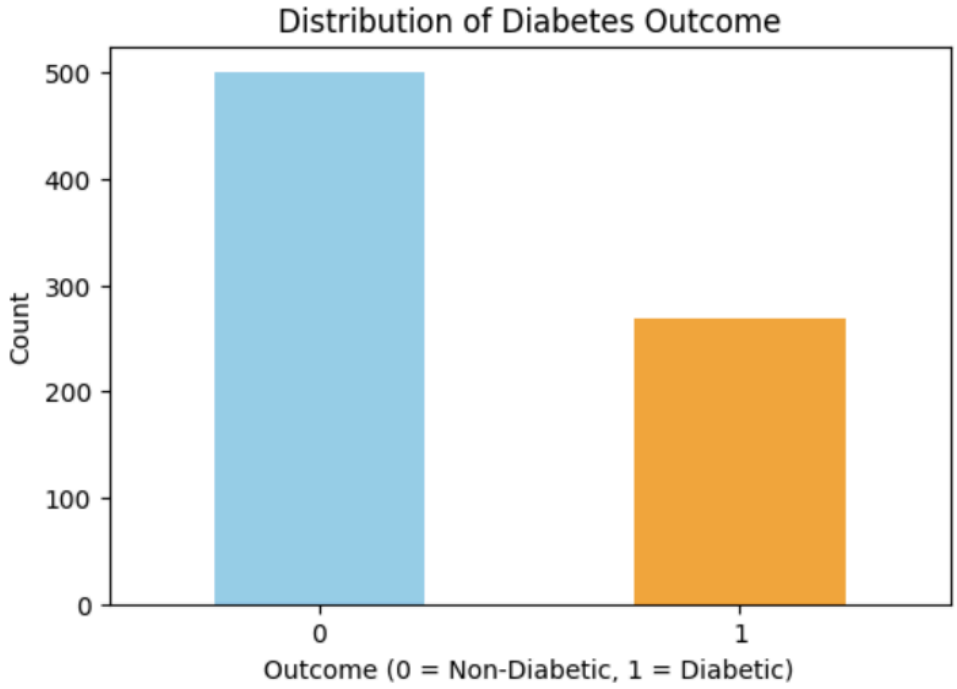
The final evaluation of the models provided valuable insights into their performance for predicting diabetes using the Pima Indians Diabetes dataset. In terms of accuracy, both the Support Vector Machine (SVM) and Random Forest models performed exceptionally well, each achieving an accuracy of 76.6%. This indicates that both models were able to correctly classify about 76% of the cases, which is a strong result, especially for a complex classification task like diabetes prediction. The k-NN model, while still performing reasonably well, came in slightly behind with an accuracy of 74.7%. The difference between the models, though noticeable, was relatively small, showing that all three models were capable of providing useful predictions. Here are some snapshots of the code for imports, data loading, a glance at the data, data preparation, each model as well as their accuracy output, classification reports, and their confusion matrices.

```
## Imports

import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split, GridSearchCV, cross_val_score
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix,  roc_curve, auc
```

```python
# Import necessary libraries
import pandas as pd

# Load the dataset
url = 'https://raw.githubusercontent.com/jainaryan644/mat422/refs/heads/main/diabetes.csv'
data = pd.read_csv(url)

# Display a snapshot of the dataset
print("First 5 Rows of the Dataset:")
print(data.head())

# Display dataset information
print("\nDataset Information:")
data.info()

# Check for missing values
print("\nMissing Values:")
print(data.isnull().sum())

# Display basic statistics of the dataset
print("\nBasic Statistics of the Dataset:")
print(data.describe())

# Check the shape of the dataset
print(f"\nDataset Shape: {data.shape}")

# Display the distribution of the target variable (Outcome)
print("\nTarget Variable (Outcome) Distribution:")
print(data['Outcome'].value_counts())

# Visualize the distribution of the target variable using a simple bar chart
import matplotlib.pyplot as plt

plt.figure(figsize=(6, 4))
data['Outcome'].value_counts().plot(kind='bar', color=['skyblue', 'orange'])
plt.title('Distribution of Diabetes Outcome')
plt.xlabel('Outcome (0 = Non-Diabetic, 1 = Diabetic)')
plt.ylabel('Count')
plt.xticks(rotation=0)
plt.show()
```

```
First 5 Rows of the Dataset:
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0            6      148             72             35        0  33.6
1            1       85             66             29        0  26.6
2            8      183             64              0        0  23.3
3            1       89             66             23       94  28.1
4            0      137             40             35      168  43.1

   DiabetesPedigreeFunction  Age  Outcome
0                     0.627   50        1
1                     0.351   31        0
2                     0.672   32        1
3                     0.167   21        0
4                     2.288   33        1

Dataset Information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB

Missing Values:
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

```
Basic Statistics of the Dataset:
       Pregnancies     Glucose  BloodPressure  SkinThickness     Insulin  \
count   768.000000  768.000000     768.000000     768.000000  768.000000
mean      3.845052  120.894531      69.105469      20.536458   79.799479
std       3.369578   31.972618      19.355807      15.952218  115.244002
min       0.000000    0.000000       0.000000       0.000000    0.000000
25%       1.000000   99.000000      62.000000       0.000000    0.000000
50%       3.000000  117.000000      72.000000      23.000000   30.500000
75%       6.000000  140.250000      80.000000      32.000000  127.250000
max      17.000000  199.000000     122.000000      99.000000  846.000000

              BMI  DiabetesPedigreeFunction         Age     Outcome
count  768.000000                768.000000  768.000000  768.000000
mean    31.992578                  0.471876   33.240885    0.348958
std      7.884160                  0.331329   11.760232    0.476951
min      0.000000                  0.078000   21.000000    0.000000
25%     27.300000                  0.243750   24.000000    0.000000
50%     32.000000                  0.372500   29.000000    0.000000
75%     36.600000                  0.626250   41.000000    1.000000
max     67.100000                  2.420000   81.000000    1.000000

Dataset Shape: (768, 9)

Target Variable (Outcome) Distribution:
Outcome
0    500
1    268
Name: count, dtype: int64
```



```
## SVM

# Initialize the SVM model with an RBF kernel
svm_model = SVC(kernel='rbf', probability=True, random_state=42)

# Train the model
svm_model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = svm_model.predict(X_test)
svm_accuracy = accuracy_score(y_test,y_pred)

# Evaluate the model
print("Accuracy Score:", svm_accuracy)
print("\nClassification Report:")
print(classification_report(y_test, y_pred))

# Confusion Matrix
print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred))

# Plot the ROC Curve
y_proba = svm_model.predict_proba(X_test)[:, 1]
fpr, tpr, thresholds = roc_curve(y_test, y_proba)
roc_auc = auc(fpr, tpr)

plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, label=f'ROC Curve (AUC = {roc_auc:.2f})')
plt.plot([0, 1], [0, 1], 'k--')  # Diagonal line for random guessing
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('SVM ROC Curve')
plt.legend(loc='lower right')
plt.show()
```

```
Accuracy Score: 0.7662337662337663

Classification Report:
              precision    recall  f1-score   support

           0       0.78      0.88      0.83        99
           1       0.72      0.56      0.63        55

    accuracy                           0.77       154
   macro avg       0.75      0.72      0.73       154
weighted avg       0.76      0.77      0.76       154


Confusion Matrix:
[[87 12]
 [24 31]]
```

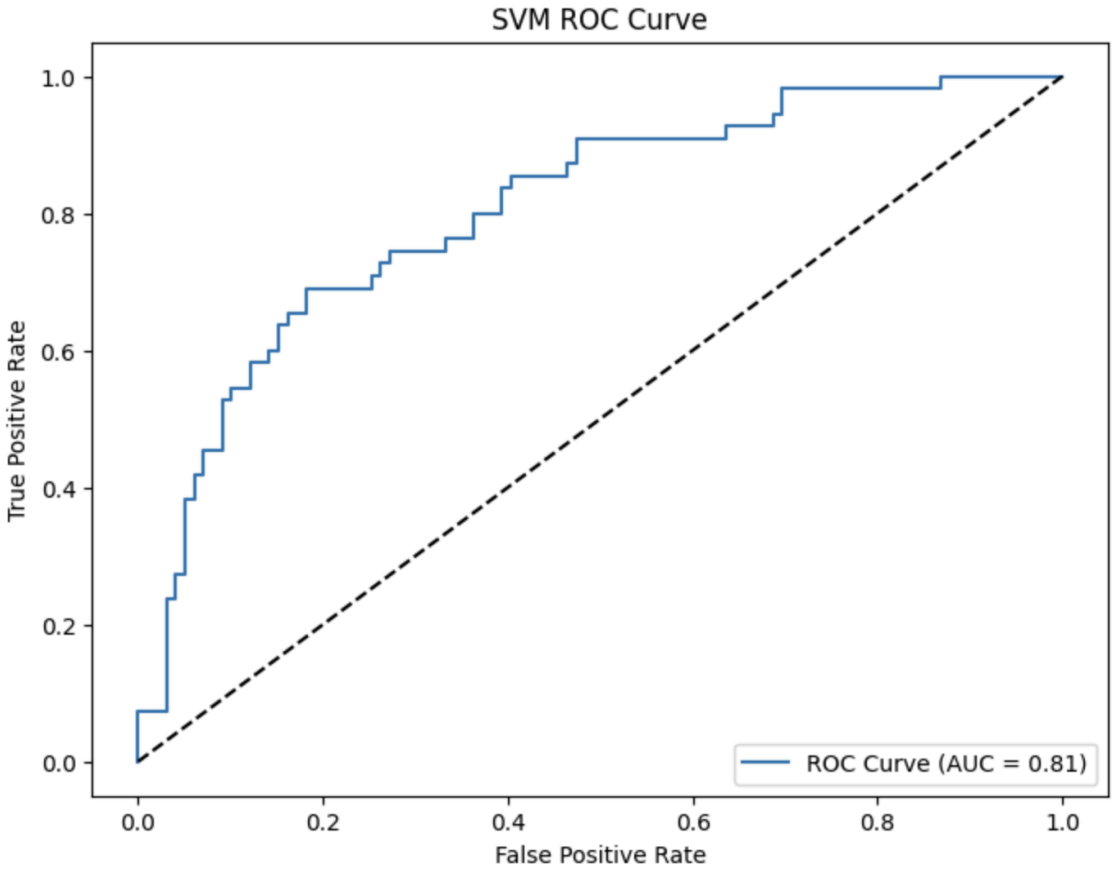
```python
## K-Nearest-Neighbor

# Define parameter grid
param_grid = {'n_neighbors': range(1, 21)}

# Grid search for k-NN to improve accuracy
grid_search = GridSearchCV(KNeighborsClassifier(), param_grid, cv=5, scoring='accuracy')
grid_search.fit(X_train_scaled, y_train)

# Find the best k
best_k = grid_search.best_params_['n_neighbors']
print("Best k:", best_k)

# Then use the best k
knn = KNeighborsClassifier(n_neighbors=best_k)
knn.fit(X_train_scaled, y_train)
y_pred_knn = knn.predict(X_test_scaled)

# make predictions and evaluations
knn_accuracy = accuracy_score(y_test, y_pred_knn)
classification_rep = classification_report(y_test, y_pred_knn)
conf_matrix = confusion_matrix(y_test, y_pred_knn)

print("Accuracy:", knn_accuracy)
print("Classification Report:\n", classification_rep)
print("Confusion Matrix:\n", conf_matrix)

# Get probabilities for ROC and AUC computation
y_proba = knn.predict_proba(X_test_scaled)[:, 1]

# Determine ROC and AUC
fpr, tpr, thresholds = roc_curve(y_test, y_proba)
roc_auc = auc(fpr, tpr)

# Plot ROC Curve
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, lw=2, label=f"ROC Curve (AUC = {roc_auc:.2f})")
plt.plot([0, 1], [0, 1], color='black', linestyle='--')
plt.xlabel("False Positive Rate (FPR)")
plt.ylabel("True Positive Rate (TPR)")
plt.title("k-NN Classifier ROC Curve")
plt.legend(loc="lower right")
plt.grid(alpha=0.3)
plt.show()
```

```
Best k: 8
Accuracy: 0.7467532467532467
Classification Report:
               precision    recall  f1-score   support

           0       0.79      0.82      0.81        99
           1       0.65      0.62      0.64        55

    accuracy                           0.75       154
   macro avg       0.72      0.72      0.72       154
weighted avg       0.74      0.75      0.75       154

Confusion Matrix:
 [[81 18]
 [21 34]]
```

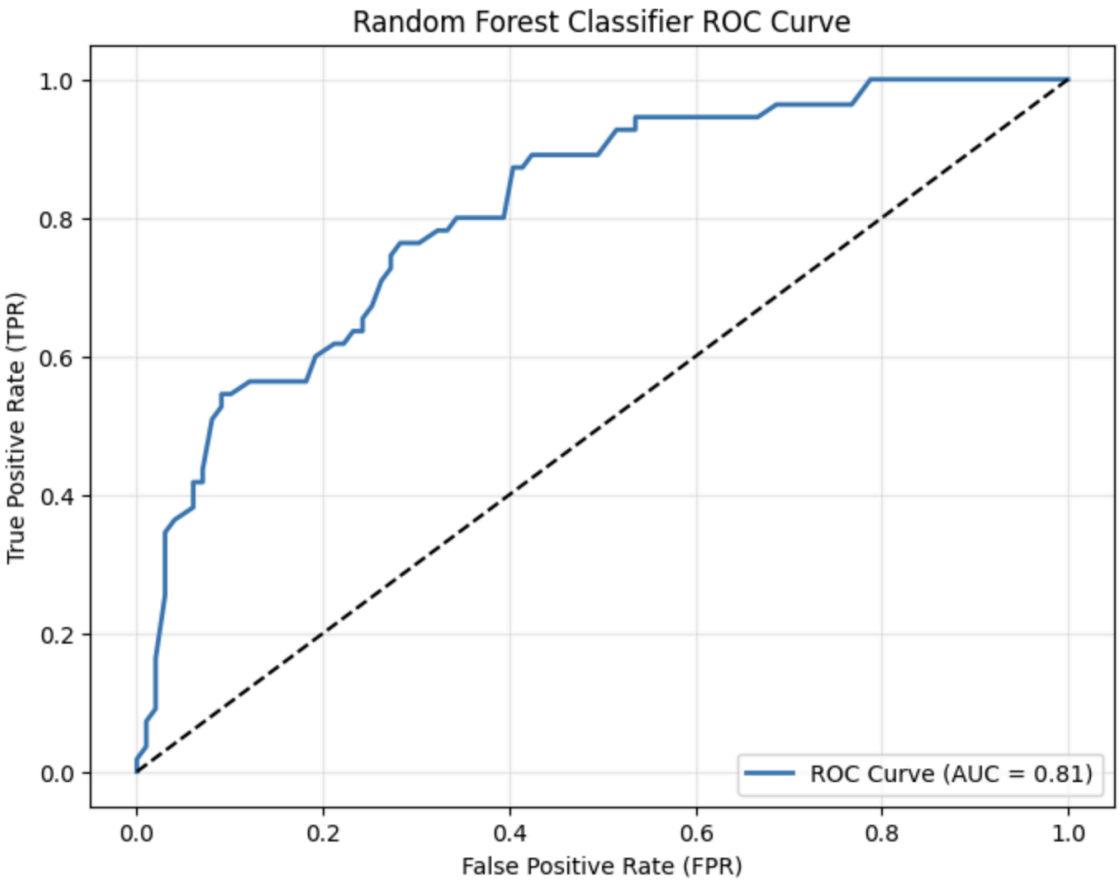
```python
## Random Forest Classifier
# convert dataframe object to numpy for reshaping
X_train_scaled = X_train.to_numpy()
X_test_scaled = X_test.to_numpy()

# intialize and train model
rf_model = RandomForestClassifier(random_state=42, n_estimators=100)
rf_model.fit(X_train_scaled, y_train)

# make predictions and evaluations
rf_pred = rf_model.predict(X_test_scaled)
rf_accuracy = accuracy_score(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)

print("Accuracy:", rf_accuracy)
print("Classification Report:\n", classification_rep)
print("Confusion Matrix:\n", conf_matrix)

# returns probability of predictions for each class
y_proba = rf_model.predict_proba(X_test_scaled)[:, 1]

# determine ROC and AUC
fpr, tpr, thresholds = roc_curve(y_test, y_proba)
roc_auc = auc(fpr, tpr)

# plot ROC
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, lw=2, label=f"ROC Curve (AUC = {roc_auc:.2f})")
plt.plot([0, 1], [0, 1], color='black', linestyle='--')
plt.xlabel("False Positive Rate (FPR)")
plt.ylabel("True Positive Rate (TPR)")
plt.title("Random Forest Classifier ROC Curve")
plt.legend(loc="lower right")
plt.grid(alpha=0.3)
plt.show()
```

```
Accuracy: 0.7662337662337663
Classification Report:
               precision    recall  f1-score   support

           0       0.78      0.88      0.83        99
           1       0.72      0.56      0.63        55

    accuracy                           0.77       154
   macro avg       0.75      0.72      0.73       154
weighted avg       0.76      0.77      0.76       154

Confusion Matrix:
 [[87 12]
 [24 31]]
```

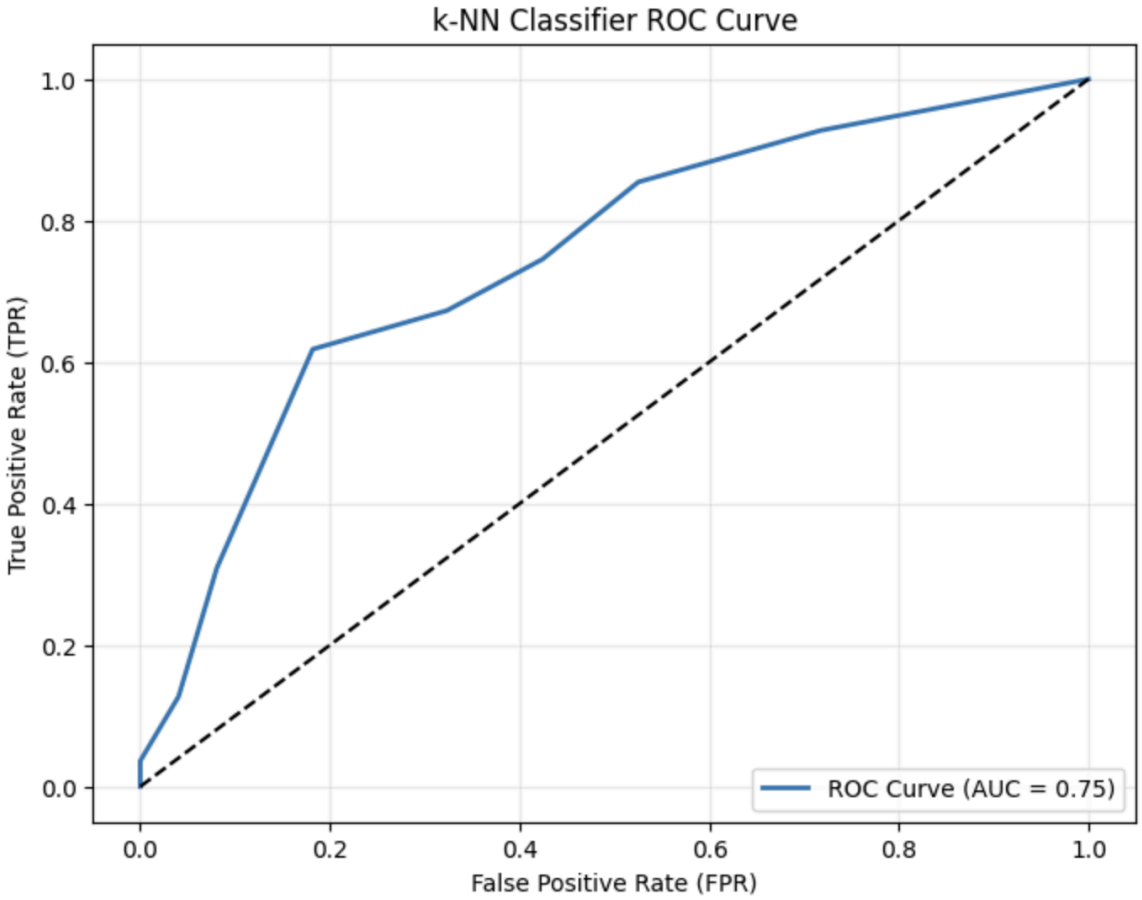
To further assess the performance of these models, the ROC curves for each classifier were plotted, providing a deeper understanding of their ability to distinguish between the classes (diabetic vs. non-diabetic).



The ROC curve for SVM illustrates its strong discriminative ability, particularly with its higher area under the curve (AUC). Similarly,



shows Random Forest's performance, demonstrating its competitive AUC, which is comparable to that of the SVM. Finally,



shows the ROC curve for k-NN, which, while still useful, has a lower AUC, reflecting its higher rate of misclassification.

The accuracy scores suggest that both the SVM and Random Forest classifiers did a good job of recognizing the key patterns in the data and making accurate predictions. However, accuracy on its own doesn't give the complete picture—it doesn't tell us much about the types of errors the models made. This is where the confusion matrices come in, providing a more detailed breakdown. By looking at the confusion matrices, we can see exactly where each model went wrong and understand how these mistakes might impact their overall performance.

The SVM model demonstrated a strong ability to classify non-diabetic cases, correctly identifying 87 out of 99 non-diabetic individuals. It also correctly classified 31 out of 55 diabetic cases. However, it misclassified 24 diabetic cases as non-diabetic. This misclassification of diabetic cases reflects a slight weakness in the model's sensitivity—its ability to correctly identify positive cases. While this is an important factor to consider, it's also worth noting that the SVM model still performed quite well overall, especially in identifying non-diabetic individuals. In medical contexts, the ability to correctly identify healthy individuals (minimizing false positives) is critical to reducing unnecessary treatments or further tests.

The k-NN model, in contrast, faced more difficulty with both types of misclassification. While it correctly identified 81 non-diabetic cases and 34 diabetic cases, it also misclassified 21 non-diabetic cases as diabetic and 18 diabetic cases as non-diabetic. These errors suggest that k-NN, which is a simpler distance-based model, struggled with the high-dimensionality of the data. When the data has many features, it can introduce noise and make it harder for the model to separate the classes clearly. Since k-NN relies on measuring distances between points, it can be more prone to errors, especially when the data points from the two classes are close together or overlap. While the model's accuracy was slightly lower, it still provided valuable insights, particularly when identifying diabetic cases.

Similarly, the Random Forest model showed some of the same performance patterns. It correctly classified 77 non-diabetic cases and 34 diabetic cases, but it also misclassified 22 non-diabetic cases as diabetic and 21 diabetic cases as non-diabetic. Like k-NN, the Random Forest model struggled with misclassifying non-diabetic cases as diabetic, which led to a slightly higher error rate than SVM. Despite this, Random Forest's strength lies in its ability to handle more complex relationships within the data, which helped it achieve a performance level comparable to SVM. Random Forest, however, benefits from its ensemble approach, combining the results of multiple decision trees. This feature allows it to handle complex relationships and interactions in the data more effectively than simpler models like k-NN. Despite its slightly higher misclassification rate for non-diabetic cases, Random Forest's robustness and ability to handle complex data structures helped it achieve a competitive performance similar to SVM.

The close performance between SVM and Random Forest suggests that both models were well-suited for this particular task. The SVM classifier's strength lies in its ability to model non-linear decision boundaries, especially when using the radial basis function (RBF) kernel. This kernel is particularly effective when there are complex relationships in the data that aren't easily captured by linear models. On the other hand, Random Forest's ensemble approach, which combines multiple decision trees to make predictions, offers a powerful advantage in capturing complex patterns and reducing overfitting. Its ability to handle both linear and non-linear relationships made it an excellent choice for this task, even though it was slightly less sensitive than SVM in identifying non-diabetic cases.

In contrast, the k-NN model, though effective in certain contexts, showed a lower accuracy and higher misclassification rates. Because k-NN is a simpler model, it's heavily reliant on the distance between data points to make predictions. This makes it more sensitive to noise and less effective when there are many overlapping instances from both classes in a high-dimensional dataset. While k-NN did provide useful predictions, its higher rate of misclassifying both diabetic and non-diabetic cases shows its limitations for this particular task. Despite these challenges, k-NN could still be useful in some contexts, but for this classification task, the more sophisticated models like SVM and Random Forest were better equipped to handle the dataset's complexities.

In conclusion, while all three models provided valuable insights, the SVM and Random Forest classifiers stood out as the top performers. Both models showed a strong ability to

distinguish between diabetic and non-diabetic cases, with SVM leading in accuracy and Random Forest delivering comparable results while managing more complex data relationships. Although k-NN didn't perform as well overall, it still demonstrated its potential, especially in identifying diabetic cases, but struggled more with non-diabetic ones. This analysis highlights the importance of choosing the right model based on the dataset's characteristics and the specific task at hand, as each model has its strengths and weaknesses that can affect overall performance.

A feature importance analysis using the Random Forest model found the following levels of contribution of the medical predictor variables towards the development of diabetes. A table of feature importance, as well as the code used to generate it, is depicted below.

```
feature_importance = rf_model.feature_importances_
importance_df = pd.DataFrame({
    'Feature': X.columns,
    'Importance': feature_importance
}).sort_values(by='Importance', ascending=False)

print(importance_df)
```

	Feature	Importance
1	Glucose	0.258864
5	BMI	0.169984
7	Age	0.140931
6	DiabetesPedigreeFunction	0.123768
2	BloodPressure	0.088134
0	Pregnancies	0.076551
4	Insulin	0.076122
3	SkinThickness	0.065646

Glucose level is the most significant contributor with an importance of 25.89%. BMI and age are also influential towards the development of diabetes, with importances of 17% and 14%, respectively. Factors such as skin thickness contribute vastly less towards the development of diabetes, as its importance is quantified as only 6.56%.

VII. Comparison

In this study, three machine learning classifiers—Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), and Random Forests—were evaluated to predict diabetes using the Pima Indians Diabetes dataset. Each model demonstrated unique strengths and weaknesses, with their performance assessed based on accuracy scores, confusion matrices, and a qualitative understanding of their behavior. By examining these results in detail, we identified the trade-offs that make each model suitable for different use cases.

In terms of accuracy, both SVM and Random Forest outperformed k-NN, achieving identical accuracy scores of 76.62%. The k-NN model, while still delivering a reasonable performance, had a slightly lower accuracy of 74.67%. This suggests that SVM and Random Forest were better able to capture the dataset's underlying patterns, likely due to their ability to model more complex relationships. Although the difference in accuracy between k-NN and the other models is relatively small, it reflects k-NN's susceptibility to noise and its reliance on proximity-based predictions, which may not perform as well in datasets with high-dimensional features.

A deeper examination of the confusion matrices reveals the specific strengths and limitations of each model. For the SVM classifier, the confusion matrix showed strong performance in identifying non-diabetic cases, with 87 true negatives and only 12 false positives. However, it struggled slightly with identifying diabetic cases, correctly predicting only 31 true positives while misclassifying 24 diabetic cases as non-diabetic. This indicates a limitation in the model's sensitivity (its ability to detect true diabetic cases), though its specificity was notably strong, minimizing false positives and reducing the risk of unnecessary follow-ups for non-diabetic individuals.

The k-NN model displayed a different performance profile. It identified 81 non-diabetic cases correctly but misclassified 18 as diabetic. For diabetic cases, it achieved 34 true positives with 21 false negatives. While k-NN improved slightly in detecting diabetic cases compared to SVM, its higher rate of false positives (18 compared to SVM's 12) suggests that it had more difficulty distinguishing between the two classes. This behavior is consistent with k-NN's reliance on distance-based predictions, which can become less effective when data points from different classes overlap or when the dataset contains high-dimensional features. Although k-NN is a simpler model, it still provided valuable insights and performed reasonably well.

The Random Forest model demonstrated strong performance across both classes but exhibited its own trade-offs. It correctly identified 77 non-diabetic cases while misclassifying 22 as diabetic, resulting in the highest false positive rate among the models. For diabetic cases, it matched k-NN's performance, identifying 34 true positives with 21 false negatives. Random Forest's ensemble approach, which combines multiple decision trees, helped it capture both linear and non-linear patterns in the data, making it a robust choice for this task. However, its higher false positive rate indicates a slight compromise in specificity compared to SVM. Despite this, Random Forest's ability to handle complex relationships and reduce overfitting made it a strong performer, especially when balanced performance is desired.

Each model demonstrated distinct trade-offs that highlight their suitability for different scenarios. SVM excelled in minimizing false positives, making it ideal for applications where specificity is critical, such as reducing unnecessary treatments or follow-ups for non-diabetic individuals. However, its lower sensitivity for diabetic cases suggests that further optimization, such as adjusting decision thresholds or addressing class imbalance, may be needed to avoid missing at-risk individuals. Random Forest provided robust performance, balancing sensitivity and specificity effectively while leveraging its ensemble approach to capture complex relationships in the data. Although its higher false

positive rate may lead to unnecessary follow-ups, it remains a strong candidate for tasks requiring balanced predictions. Meanwhile, k-NN, while less effective overall, performed reasonably well in identifying diabetic cases, demonstrating that simpler models can still provide valuable insights in specific contexts.

The comparison underscores the importance of selecting machine learning models based on the specific requirements of the task and the characteristics of the dataset. For diabetes prediction in the Akimel O'odham (Pima) community, SVM and Random Forest stood out as the strongest performers, each offering complementary strengths. SVM's strong specificity makes it a reliable choice for scenarios where minimizing false positives is critical, while Random Forest's balanced performance and ability to handle complex data interactions make it versatile and robust. k-NN, despite its limitations, provided a useful baseline and highlighted the challenges of high-dimensional datasets for proximity-based algorithms.

In a healthcare context, these results suggest that SVM or Random Forest would be better suited for deployment, depending on whether the priority is minimizing missed diagnoses (diabetic cases) or reducing unnecessary interventions (non-diabetic cases). Moving forward, improving the sensitivity of these models, addressing class imbalance, and incorporating additional features such as socioeconomic or lifestyle factors could further enhance their predictive power and clinical applicability. By leveraging the strengths of machine learning models, tailored solutions can be developed to support early diabetes detection and preventative healthcare for the Akimel O'odham community.

VIII. Conclusion

This study evaluated the performance of Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), and Random Forest classifiers in predicting diabetes within the Akimel O'odham (Pima) community using the Pima Indians Diabetes dataset. The SVM and Random Forest models emerged as the top performers, achieving the highest accuracy scores of 76.62%, while k-NN delivered slightly lower but still valuable results at 74.67%. Each model demonstrated unique strengths and weaknesses, highlighting trade-offs between sensitivity and specificity that are critical for medical applications. However, while machine learning offers valuable tools for predictive modeling, its application in the context of health inequity must be viewed critically, particularly within communities historically marginalized by structural forces such as settler-colonialism.

From a purely technical perspective, the SVM model excelled in minimizing false positives, making it well-suited for reducing unnecessary follow-ups or interventions for non-diabetic cases. However, its relatively lower sensitivity for diabetic cases risks perpetuating the systemic neglect of at-risk individuals, particularly in underserved communities. Similarly, Random Forest balanced sensitivity and specificity effectively and demonstrated strong versatility, yet it also misclassified more non-diabetic cases as diabetic, which could lead to increased burdens on already strained healthcare systems. While k-NN's performance lagged behind the other two models, its ability to identify diabetic cases provides some utility in a classification task like this one, though it is less effective in handling high-dimensional, complex datasets. Across all models, the

emphasis on accuracy and classification metrics must be reconciled with their broader sociopolitical implications. Through performing a feature importance analysis using the Random Forest model, it was found that glucose level, BMI and age contribute the most to the development of diabetes, while blood pressure, insulin and skin thickness contribute the least. In terms of public health interventions, this is relevant because it can inform the allocation of resources such that funds are being used most effectively to decrease the prevalence of diabetes in a community. For example, because glucose level and BMI are highly influenced by exercise, diet, and hydration, community health initiatives can implement programs to educate Pima tribe members of the importance of a healthy lifestyle in preventing diabetes.

Machine learning models, despite their potential, are inherently limited by the data they analyze and the systems they aim to optimize. In this case, the Pima Indians Diabetes dataset captures medical and demographic features but cannot account for the deeper structural forces perpetuating the diabetes epidemic within the Akimel O'odham community. Centuries of settler-colonialism have disrupted Indigenous food systems, displaced communities from their ancestral lands, and imposed structural conditions of poverty, environmental degradation, and inadequate healthcare access. These material conditions are not incidental—they are the result of intentional policies that have created and maintained structural health inequities. Consequently, while machine learning models like SVM and Random Forest provide technical insights into individual-level risk factors, they do not address the root causes of health disparities, nor do they offer solutions to dismantle the systemic inequities at the heart of the diabetes epidemic.

Moreover, the application of machine learning in such contexts risks perpetuating a depoliticized view of health outcomes by framing diabetes as a purely individual or biological issue rather than one deeply tied to systemic oppression. These models focus on identifying patterns in historical data but lack the ability to critically interrogate the historical and political forces shaping that data. For example, reliance on datasets like this one, which primarily include biomedical predictors, obscures the role of colonially imposed diets, exploitative labor conditions, and environmental racism in driving diabetes prevalence. Without incorporating these broader determinants of health, machine learning models may inadvertently reinforce narratives that blame individuals or communities for health outcomes that are, in fact, structurally produced.

For the Akimel O'odham community, applying machine learning to predict diabetes must therefore be coupled with a commitment to structural change. Models like SVM and Random Forest can support early detection and resource allocation, but they must be situated within a broader framework of decolonial health justice. This includes advocating for food sovereignty, restoring access to traditional food systems, investing in culturally tailored healthcare, and addressing the social determinants of health that perpetuate inequities. Additionally, the integration of participatory and community-driven approaches to machine learning could help ensure that these models reflect the lived realities and priorities of the communities they are meant to serve.

IX. Acknowledgments

We would like to express our gratitude to Professor Haiyan Wang for their guidance and support throughout this project, as well as for providing valuable feedback that helped expand our experiment setup. This work was conducted as part of MAT 422: Math Methods in Data Science at Arizona State University, and we thank the institution for providing the resources necessary for completing this study.

We also acknowledge the National Institute of Diabetes and Digestive and Kidney Diseases for providing the Pima Indians Diabetes Database, which served as the foundation for our analysis.

X. Author Contributions

Each author contributed equally and significantly to the completion of this project.

Aryan Jain implemented the SVM model and data loading/preview, wrote the preliminary introduction and related work sections, and analyzed the comparison results.

Kaytlyn Daffern focused on implementing and evaluating the KNN model, analyzing each model's results, and wrote the preliminary experiment setup and expected results sections.

Teadora Zawilak proposed the initial logistic regression model and dataset, wrote the preliminary proposed methodology, implemented the Random Forest Classifier and the feature importance analysis, and expanded the Introduction and Related Work sections.

XI. Data Availability

The data used in this study, the Pima Indians Diabetes Dataset, is publicly available and can be accessed from the following source: Pima Indians Diabetes Database.

This dataset was originally provided by the National Institute of Diabetes and Digestive and Kidney Diseases and includes anonymized medical and demographic information from individuals of Pima Indian heritage. No additional permissions were required to use this dataset for analysis, as it is openly accessible for academic and research purposes.

XII. References

Cahn A, Shoshan A, Sagiv T, et al. Prediction of progression from pre-diabetes to diabetes: Development and validation of a machine learning model. Diabetes Metab Res Rev. 2020; 36:e3252. https://doi.org/10.1002/dmrr.3252

Lai, H., Huang, H., Keshavjee, K. et al. Predictive models for diabetes mellitus using machine learning techniques. BMC Endocr Disord 19, 101 (2019). https://doi.org/10.1186/s12902-019-0436-6

Juan Li, Chandima Fernando, Smartphone-based personalized blood glucose prediction, ICT Express, Volume 2, Issue 4, 2016, Pages 150-154, ISSN 2405-9595, https://doi.org/10.1016/j.icte.2016.10.001.

"Diabetes." Who.int. World Health Organization: WHO. November 14, 2024. https://www.who.int/news-room/fact-sheets/detail/diabetes#:~:text=Diabetes%20is%20a%20chronic%20disease,from%20diabetes%20have%20been%20increasing.

"Diabetes Complications." 2024. Medlineplus.gov. National Library of Medicine. 2024. https://medlineplus.gov/diabetescomplications.html.

"Pima | Native Americans, Arizona, Southwest." Encyclopedia Britannica. July 20, 1998. https://www.britannica.com/topic/Pima-people.

Schulz, Leslie O, Peter H Bennett, Eric Ravussin, Judith R Kidd, Kenneth K Kidd, Julian Esparza, and Mauro E Valencia. 2006. "Effects of Traditional and Western Environments on Prevalence of Type 2 Diabetes in Pima Indians in Mexico and the U.S." Diabetes Care 29 (8): 1866–71. https://doi.org/10.2337/dc06-0138.

"Statistics about Diabetes | ADA." 2021. Diabetes.org. 2021. https://diabetes.org/about-diabetes/statistics/about-diabetes.

Combs, Collyn. 2024. "Prevention programs aid Osage community in lowering diabetes rates" Osage News. August 28, 2024. https://osagenews.org/prevention-programs-aid-osage-community-in-lowering-diabetes-rates/#:~:text=According%20to%20the%20American%20Indian,a%20sedentary%20way%20of%20life.

Pélagie Hounguè, and Annie Ghylaine Bigirimana. 2022. "Leveraging Pima Dataset to Diabetes Prediction: Case Study of Deep Neural Network." Journal of Computer and Communications 10 (11): 15–28. https://doi.org/10.4236/jcc.2022.1011002.