



Diabetes Prediction

A PRESENTATION BY YEN LU, STEPHANIE AYALA, BRYAN HAMILTON-BROWN, ADEBOLA SHELBY & NEBIAT BEYENE

The Contents

01

Project Overview

02

Data Cleaning & Preprocessing

03

Handling Class Imbalance & Feature Scaling

04

Neural Network Model

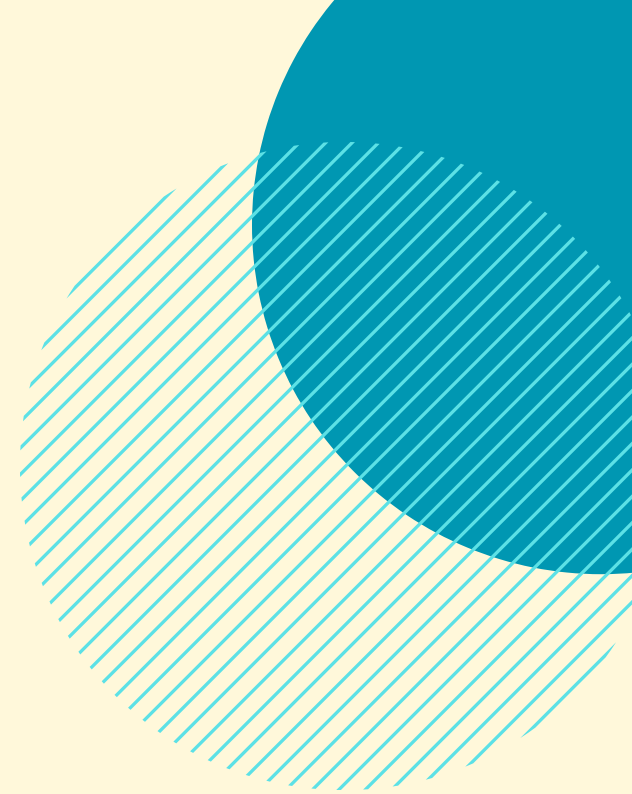
05

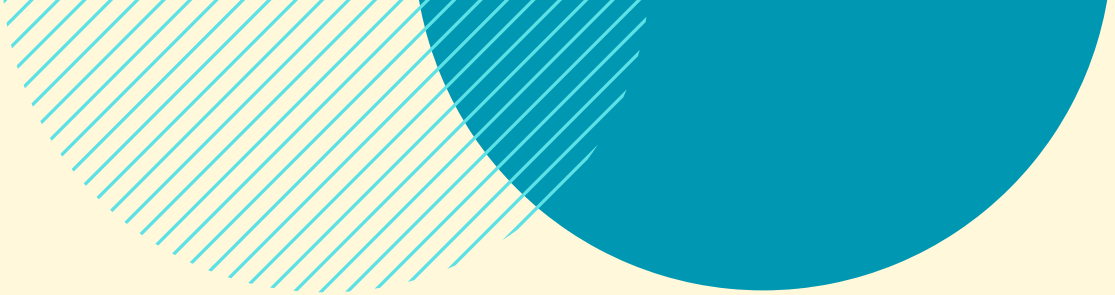
Compilation, Training & Evaluation

Project Overview

The purpose of this project is to predict if a person has diabetes using survey data collected from the Behavioral Risk Factor Surveillance System (BRFSS), which is a health-related telephone survey conducted annually by the CDC.

Annually, the survey gathers responses from over 400,000 Americans regarding health-related risk behaviors, chronic health conditions, and the utilization of preventative services.






Data Cleaning

Step 1

Step 2

Step 3

Step 4



Narrow down the dataset from 350 columns to 25

Drop Null Values

Choose 25 Health Indicators/Variables

Creating a Binary Dataset for Diabetic vs. Non-Diabetic



Data Preprocessing

- Feature and Target Separation:
 - Features (X) were separated from the target variable (y).
 - The target variable is the Diabetes_binary column.
- One-Hot Encoding:
 - Categorical columns (if any) were identified and one-hot encoded to convert them into numerical format suitable for machine learning models.
- Train-Test Split:
 - The dataset was split into training (70%) and testing (30%) subsets using a stratified approach to preserve class distribution.

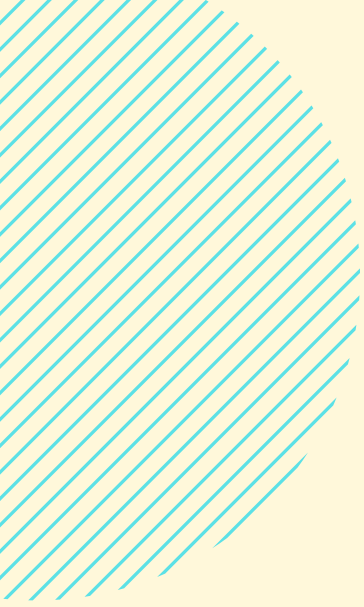


Handling Class Imbalance

The dataset was balanced using the SMOTETomek technique:

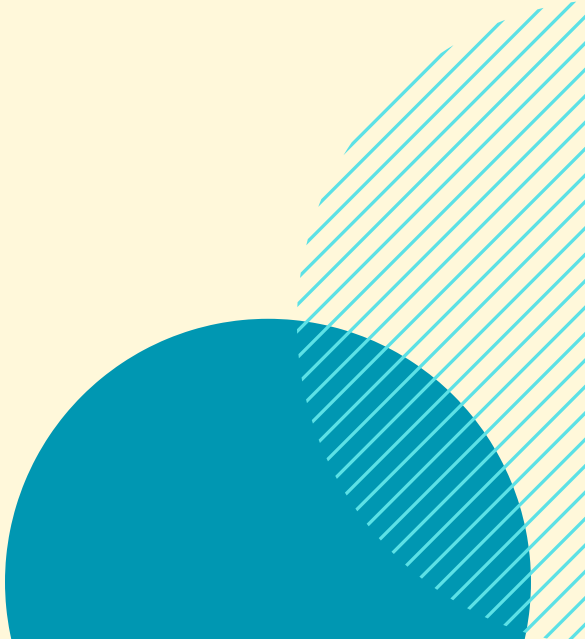
SMOTE (Synthetic Minority Oversampling Technique): Synthetic examples of the minority class were created.

Tomek Links: Instances causing class overlap were removed to improve class separation. This resulted in a balanced training dataset, crucial for improving the performance of the model on imbalanced data.



Feature Scaling

We standardized all features using StandardScaler to normalize feature values, ensuring faster and more stable model training.





Neural Network Model

An artificial neural network was implemented using TensorFlow and Keras. The architecture includes:

- Input Layer: Matches the number of features in the dataset.
- Hidden Layers:
 - Three dense layers with ReLU activation for non-linear transformations.
 - Batch normalization layers for stabilizing and accelerating training.
 - Dropout layers (50% rate) to prevent overfitting.
- Output Layer: A single neuron with a sigmoid activation function for binary classification.

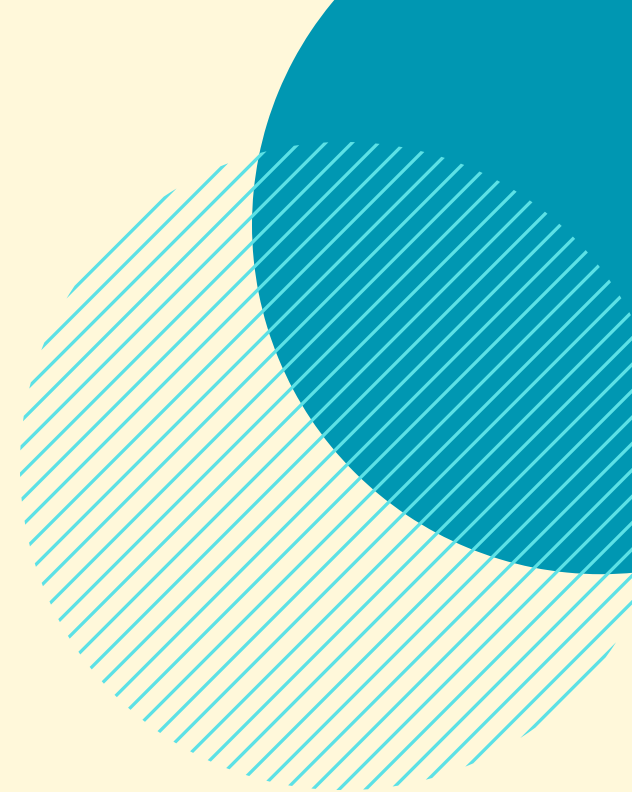
Compilation and Training

The model was compiled with:

- Loss Function: `binary_crossentropy` to measure prediction error.
- Optimizer: Adam with a learning rate of 0.001 for efficient gradient descent.
- Metrics: Accuracy was used as the performance metric.

Training Details:

- The model was trained for 20 epochs with a batch size of 32.
- Validation was performed on the test dataset to monitor performance.



Evaluation

The model's performance was evaluated using:

- Accuracy: Measures overall predictive correctness.
- Classification Report: Includes precision, recall, F1-score, and support for both classes.

Results

Logistic Regression:

- Overall Accuracy: 74%
- Precision – Non-diabetics: 75%; Diabetics: 73%
- Recall – Non-diabetics: 72%; Diabetics: 76%

Neural Network:

- Overall Accuracy: 75%
- Precision – Non-diabetics: 78%; Diabetics: 73%
- Recall – Non-diabetics: 70%; Diabetics: 80%



Thank You