**A blue circle with white text and wings in a circle

Description automatically generated**

**Pima Indians Diabetes Dataset**

**Mahsa Dorri**

**Mohammad Reza Mohtat**

**March2024**

**Introduction:**

This project involves using IBM SPSS Modeler software to analyze a dataset originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The goal is to predict whether a patient has diabetes based on various diagnostic measurements. The dataset is specifically focused on a subset of patients with the following criteria: all are females, at least 21 years old, and of Pima Indian heritage. This focus is due to several constraints placed on the selection of instances from a larger database to ensure the dataset is homogeneous and relevant to the study.

The dataset's features include:

**Pregnancies:**

This feature counts the number of pregnancies the patient has had. A higher number might indicate a higher risk of developing diabetes, especially gestational diabetes, which can lead to type 2 diabetes later in life.

**Glucose:**

Measured in mg/dL, this is a key indicator of diabetes. The plasma glucose concentration 2 hours after consuming a glucose-rich beverage during an oral glucose tolerance test (OGTT) helps diagnose diabetes and prediabetes. Normal values are typically less than 140 mg/dL, with higher values indicating potential diabetes.

**Blood Pressure:**

Specifically, diastolic blood pressure, measured in mm Hg. It indicates the pressure in the arteries when the heart rests between beats. High blood pressure (hypertension) is common in people with diabetes, and managing it is crucial for preventing complications.

**Skin Thickness:**

The thickness of the triceps skin fold, measured in millimeters. This measurement helps estimate body fat percentage. A higher value may indicate higher body fat, which is associated with insulin resistance and diabetes risk.

**Insulin:**

The 2-hour serum insulin level, measured in micro-units per milliliter (mu U/ml), following the glucose intake in the OGTT. Insulin is a hormone that helps cells absorb glucose. In type 2 diabetes, cells resist insulin's action, and the pancreas may not produce enough insulin, leading to high blood glucose levels.

**BMI:**

Body Mass Index, a measure of body fat based on weight and height (kg/m^2). A BMI over 25 is considered overweight, and over 30 is obese. Obesity is a major risk factor for type 2 diabetes as it contributes to insulin resistance.

**Diabetes Pedigree Function:**

A synthetic function that provides a measure of the diabetes mellitus history in relatives and the genetic relationship of those relatives to the patient. A higher value indicates a greater genetic predisposition to diabetes.

**Age:**

The age of the patient in years. The risk of type 2 diabetes increases with age, particularly after 45 years. However, it can occur at any age and is increasingly seen in children, adolescents, and younger adults due to rising obesity rates.

**Outcome:**

This is the target variable, indicating whether the patient has diabetes (1) or not (0). It is based on diagnostic criteria such as the glucose concentration after the OGTT.

Together, let's begin our journey using IBM SPSS:

* Data Loading:

1. The initial step involves importing the dataset, which is provided in a CSV (Comma-Separated Values) file format.

A screenshot of a computer

Description automatically generated

* Data Preprocessing:

1. The second step is to address and fill in missing or nonsensical values, such as zero entries in fields where zero is not a meaningful value. This is particularly important for features like blood pressure, Glucose, insulin, and skin thickness where a zero value is biologically implausible and can skew our analysis. By identifying and substituting these zero entries with null values or appropriate imputations, we ensure the integrity and accuracy of our dataset.

A screenshot of a computer

Description automatically generated

1. In this step, we will utilize the "Type" node in IBM SPSS Modeler, which allows us to closely observe and understand the modifications made to our data following each change. Also when we activate the "Type" node, we select the "Read Values" option to display the numerical ranges of our variables.

A screenshot of a computer

Description automatically generated

1. I opt to use the Anderson-Darling method for this analysis. The Anderson-Darling test is highly sensitive to tail behavior, making it particularly effective for identifying deviations in distribution tails. It's versatile, applicable to a wide range of distributions, and offers improved power for detecting discrepancies from specified distributions, especially those with heavy tails. This makes it a robust choice for statistical analysis where tail behavior is critical.We have a "Sim Gen" node, which, when clicked, provides information about variable descriptions and correlations. This node is instrumental in generating insights about the relationships between variables, helping to understand how they interact with each other within the dataset.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

1. Now, we utilize the "Field" node to view graphs and thereby make informed decisions about how to handle outlier data. This node enables us to visually inspect the distribution of data, identify outliers, and assess their impact on the analysis. Based on these visual insights, we can strategize on whether to modify, remove, or retain outliers to ensure the integrity and accuracy of our statistical models.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

1. In the "Auto Data Audit" section, I employ the interquartile range (IQR) method, using the upper and lower quartiles, for outlier detection. After setting this up, I execute the "Run" command to systematically identify and assess outliers within the dataset.

In the audit section, we review the graphs and examine statistical parameters to understand the characteristics of our data better. This process includes analyzing the distribution, central tendency, and variability of each feature. Additionally, we assess the count of valid entries for each feature, identifying any missing or invalid data that might affect the quality and reliability of our analysis.

Given that the counts of valid entries for insulin(51%) and skin thickness(70%) are low, indicating missing or possibly incorrect data, we must address these issues. Handling these deficiencies involves strategies such as imputation, where missing values are replaced with statistical estimates (e.g., mean, median, or mode), or more sophisticated techniques like predictive modeling to estimate missing values based on other variables.

In the "Quality" section, we closely examine the outliers and identify the count of extreme outliers. This analysis helps us understand the extent and impact of these values on our dataset. In the "Action" part, we determine how to handle these outliers and extreme values. The options available for action include:

**None:**This option leaves the data as is, without any modification to outliers or extreme values. It's chosen when outliers are considered to be meaningful or representative of the natural variation in the data.

**Coerce:** Adjusts outliers to a specified threshold close to the rest of the data, while retaining them within the dataset. This approach moderates the impact of outliers without completely removing them.

**Discard:** Eliminates outliers or extreme values from the dataset. This option is used when such values are deemed to distort the analysis or are considered errors.

**Nullify:** Replaces outliers or extreme values with nulls, acknowledging their presence but preventing them from affecting certain statistical calculations.

**Coerce Outliers / Discard Extremes:** A combination approach that coerces outlier values to a less extreme value while completely removing the most extreme outliers from the dataset.

**Coerce Outliers / Nullify Extremes:** This option coerces less extreme outliers to closer to the median or mean values, while replacing the most extreme outliers with null values to mitigate their impact.

As a result of the selections and actions taken, the system provides a model. If you zoom in on this model, it appears as follows:

After processing through the model, when we revisit the "Type" node, we observe that the range of values for the features has changed. This alteration reflects the impact of our data cleaning and preprocessing steps, such as handling outliers, imputing missing values, or transforming variables.

A screenshot of a computer

Description automatically generated

1. Now, we move on to addressing missing values by utilizing the "Impute Missing Values" section, choosing to convert them into blanks and null values. This modification takes place within the "Data Audit" node, which is designed to identify and handle missing values in our dataset. By transforming missing entries to blanks and nulls, we prepare our data for more accurate analysis, ensuring that the absence of information is properly accounted for and doesn't skew our results. This step is critical in data preprocessing, as it directly impacts the quality of the data and the validity of any conclusions drawn from subsequent analyses.For missing values in insulin and glucose, we apply a Random Uniform imputation to preserve their distributions. Blood pressure's missing values are imputed with the mean, considering its relatively stable distribution. These methods ensure minimal bias while maintaining data integrity for accurate analysis.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Now, we proceed to scaling the data, a crucial step that ensures our model correctly interprets the magnitudes and distances between values. This normalization or standardization process adjusts the scales of different features to a common range, making them directly comparable.

A screenshot of a computer

Description automatically generated

Having completed the data cleaning process, which includes handling missing values, outliers, and scaling, we're now well-prepared to move on to the next step of our analysis.

1. For the next step, we'll split the data into training and testing sets with a ratio of 80:20. This division prepares us for model building, ensuring that we have a robust dataset for training the model and a separate dataset for evaluating its performance.

10) Now that we have three models—logistic regression, random forest, and SVM (Support Vector Machine)—we will use the "Analysis" node to compare their performances. This comparison helps us determine which model best suits our data.

A computer screen shot of a network

Description automatically generated

* Based on the prediction accuracies provided, logistic regression predicts correctly 82% of the time, while random forest and SVM (Support Vector Machine) have accuracies of 78.85% and 77.88%, respectively. This indicates that, in this case, logistic regression slightly outperforms both the random forest and SVM models.
* In our analysis, the Support Vector Machine (SVM) model achieved the highest accuracy when configured optimally. As shown in the image, the SVM model reached an accuracy of 82.69%

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

1. I added a neural network model using a Multilayer Perceptron (MLP) with one hidden layer in IBM SPSS Modeler. The model was trained on the training set (80% of the data) and evaluated on the test set (20% of the data). The neural network achieved an accuracy of 80.77%, comparable to other models tested. This demonstrates the neural network's efficacy in predicting diabetes within the dataset.

A screenshot of a computer

Description automatically generated

1. I implemented the K-Nearest Neighbors (KNN) model, which automatically selects the optimal number of neighbors (k) from a range between 2 and 20. The KNN model was trained and evaluated, achieving an accuracy of 77.88%. This accuracy indicates that KNN provides a viable alternative, though slightly less accurate than other models tested.

A screenshot of a computer

Description automatically generated

1. we evaluated the Random Forest model, which initially achieved an accuracy of 75%. This accuracy is significantly lower than expected, indicating potential issues with the model or data preprocessing. Therefore, the Random Forest model requires further testing and optimization to ensure reliable performance.
2. CHAID Model (Chi-squared Automatic Interaction Detection) model achieved an accuracy of 72.12%. This indicates that CHAID is less effective in predicting diabetes compared to other models tested.

A screenshot of a computer

Description automatically generated

1. The QUEST (Quick, Unbiased, Efficient Statistical Tree) model achieved an accuracy of 79.81%, demonstrating moderate performance in this dataset.

A screenshot of a computer

Description automatically generated

1. C&R Tree Model: The Classification and Regression Tree (C&R Tree) model performed well, achieving the highest accuracy of 83.65%. This suggests that C&R Tree is particularly effective for this prediction task.

A computer screen shot of a network

Description automatically generated

1. C5.0 Model achieved an accuracy of 78.85%, indicating a solid performance, though slightly lower than some other models.

A screenshot of a computer

Description automatically generated

**Summary of Model Accuracies**

* Logistic Regression: 82%
* Random Forest: 75% (requires further testing)
* SVM (Support Vector Machine): 81.73%
* Neural Network: 80.77%
* KNN (K-Nearest Neighbors): 77.88%
* CHAID: 72.12%
* QUEST: 79.81%
* C&R Tree: 83.65%
* C5.0: 78.85%

The C&R Tree model provides the highest accuracy in this dataset, followed closely by logistic regression and SVM.