HAN_UNIVERSITY
OF APPLIED SCIENCES

CRISP DM
DIABETES REPORT

Group:

Jeron Kamp | 2136591

Sofiia Nedbailo | 165631

Teacher:

Witek ten Hove

Table of Contents

Introduction

With the increasing number of advances in healthcare and technology, the ability to analyze causes and predict cases has become a great opportunity for different diseases and medical implications. Diabetes currently has between a 71 and 80 % chance of being correct according to (source). With the dataset provided by Kaggle (source) it will be investigated if it is possible to improve this.

In this project, there will be an analysis of a dataset (source) in which different statistics are compared to having diabetes or not. This analysis will be based on a Crisp MD model.

Description of Diabetes Dataset

The dataset selected for the subsequent CRISP-DM (Cross-Industry Standard Process for Data Mining) Report pertains to diabetes.

In this report, we will provide an overview of diabetes, a chronic medical condition characterized by elevated levels of glucose (sugar) in the bloodstream. This condition arises due to the body's inability to effectively regulate insulin, a hormone responsible for controlling blood sugar levels. The dataset overview specifically addresses data related to the female gender.

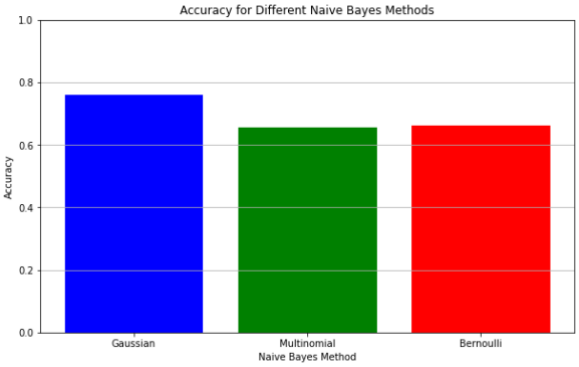
Within the dataset under observation, there are included several variables related to diabetes. These variables are pivotal in our comprehensive overview of the dataset and include the following:

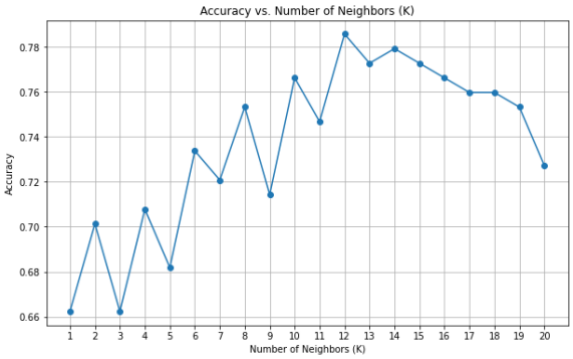
- Pregnancies: This variable records the number of pregnancies a person has experienced.
- Glucose: Indicates the blood glucose level, an important factor in diabetes management. The normal fasting blood glucose level for a healthy adult is typically between 70 and 100 milligrams per deciliter (mg/dL). However, this can vary depending on the specific context and individual factors.
- Blood Pressure: Records the individual's blood pressure readings. Normal blood pressure for adults is typically considered to be around 120/80 mm Hg. However, ideal blood pressure can vary slightly depending on age and individual health. In the dataset, the diastolic category is used to represent the value of the brood pressure.
- Skin Thickness: Measures the thickness of the skin, which can be relevant in diabetes-related assessments.
- Insulin: Captures information about insulin levels, a critical hormone in blood sugar regulation. Normal insulin levels can vary depending on factors like fasting or non-fasting state and the specific assay used to measure it. In a fasting state, normal insulin levels may be around 5-15 µU/mL.
- BMI (Body Mass Index): Reflects the individual's body mass index, which is associated with diabetes risk. A healthy BMI for adults is usually considered to be between 18.5 and 24.9. However, it's essential to note that BMI is a rough measure of body composition and does not account for factors like muscle mass.
- Diabetes Pedigree Function: Provides insight into the genetic predisposition for diabetes. This function is used to assess the genetic predisposition for diabetes but does not have a specific "normal" value. Higher values may indicate a higher genetic risk.
- Age: Records the age of the individuals in the dataset, a significant factor in diabetes diagnosis and management.
- Outcome: Indicates the presence or absence of diabetes, serving as our target variable for analysis. In the dataset 0 represents no diabetes, and 1 represents diabetes.

These variables collectively form the foundation for our comprehensive analysis and insights into the diabetes dataset.

Data Preparation

For the Data Preparation the first thing that was done to the dataset was ETL (extract, transform and load). But after the first try it was quickly noticed that the data is not usable with a simple transformation. This was because the different column had different numbers and values which made the values very weird. After looking at the data a bit more, together with visualization seen in the next chapter Data Modelling, there were some interesting findings. After using KNN and Naïve Bayes on the untransformed dataset there were different results as seen in the graphs.





Knn shows close to 79% with a K of 12. Naïve bayes shows the highest value by using a Gaussian method which is 76%

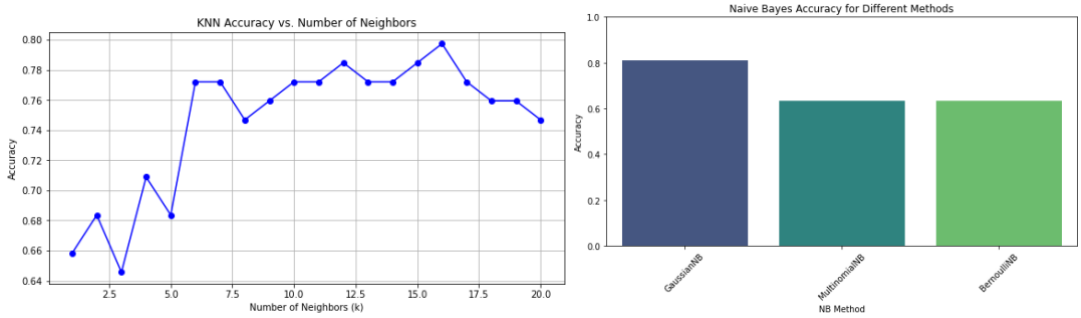
That is why for the Data preparations the evaluation from the data modelling will go further into what can be done towards the data preparation. This is used for the second phase in which the changes to the dataset will give different results.

For future investigation, in the dataset the value BMI reflects the individual's body mass index, which is associated with diabetes risk. A healthy BMI for adults is usually considered to be between 18.5 and 24.9. What we decided to do is to make 3 columns for each BMI range, which varies from normal to overweight range, and obese range.

Data Cleaning

In the Data Cleaning phase of our project, we focused on preparing the raw dataset for further analysis and modeling. Data cleaning is a critical step in the data preprocessing pipeline, as it ensures that the data is accurate, consistent, and free from errors.

To find the best way to clean the data we tried different methods. The first method was removing the 0 values from the different rows except the pregnancies and outcome. The reason we already thought that this would not be the best method is because there are a lot of 0 values. Removing the 0 values will remove about halve rows. After that we used an imputation strategy replacing it with the mean. This gives better results. Only with skinthickness and insulin did it gives weird result. We found the best result came from removing 0 values from SkinThickness and Insulin and replace the rest with mean.



Data Modelling

In the Data Modeling phase of our project, we embark on a transformative journey to enhance the predictive power of our machine learning models. Our objective is to ensure that our models are not only accurate but also well-informed by the underlying data. Central to this endeavor is the utilization of exploratory data analysis techniques, including the judicious application of boxplots and heatmaps.

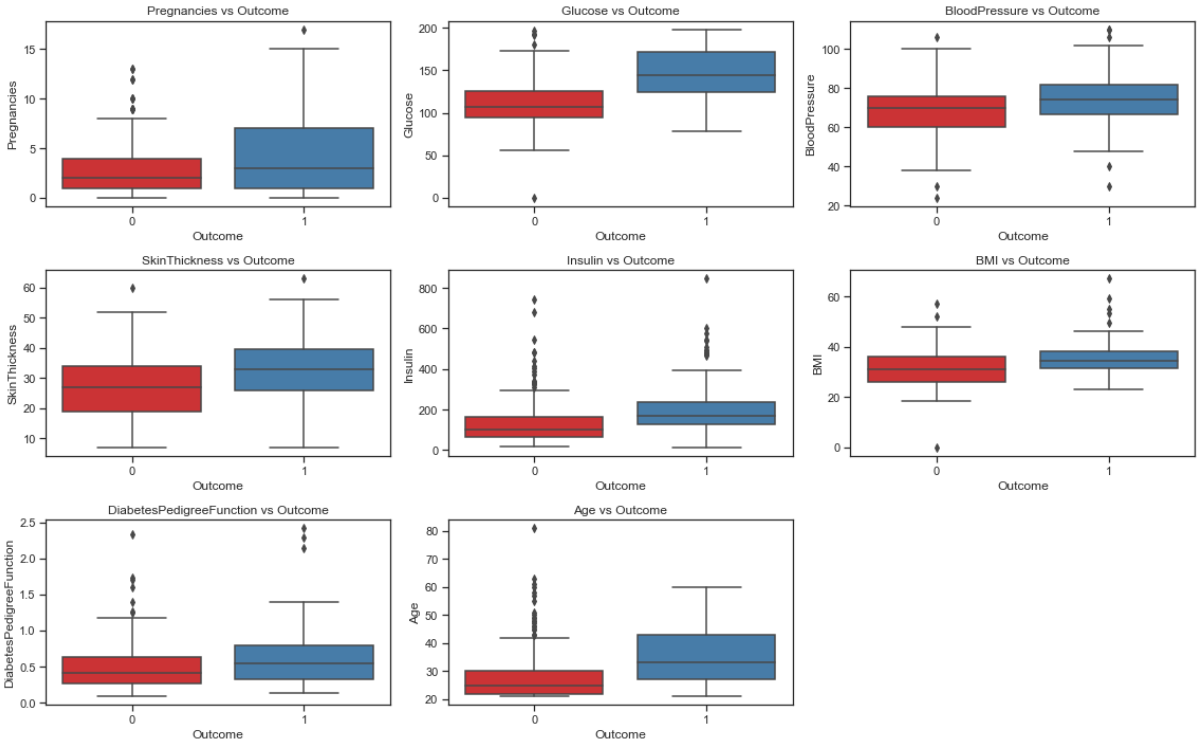
The journey begins with the recognition that raw data, although valuable, often requires refinement and augmentation to extract meaningful insights and bolster predictive performance. It is here that boxplots and heatmaps play pivotal roles as indispensable tools for visualizing and comprehending the intricate relationships within our dataset.

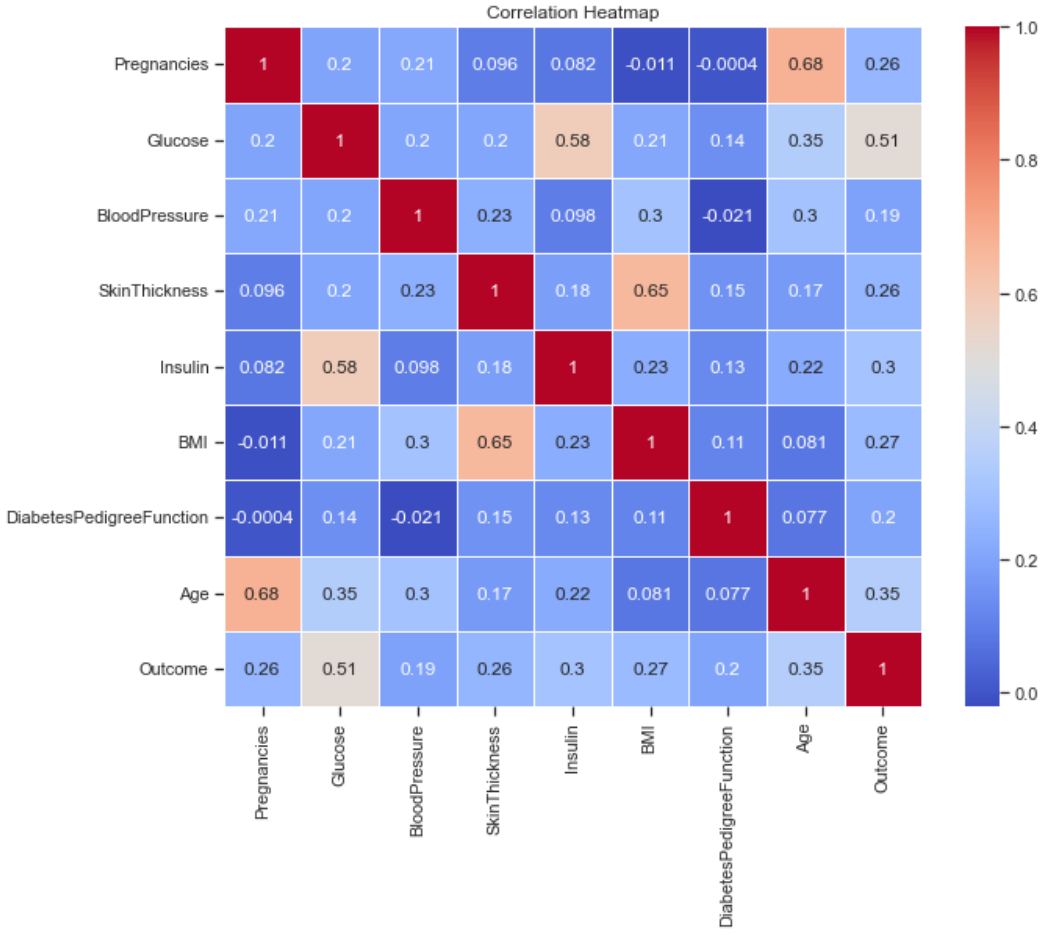
To get a clear view of the correlation between the outcome and other variables, boxplots were used to visualize them. In our exploration, we employ boxplots to discern patterns and disparities across various attributes. By comparing boxplots between different classes or categories within our dataset, we gain valuable insights into the factors that differentiate one group from another. This knowledge guides us in making informed decisions about feature engineering and selection, ultimately influencing the performance of our machine learning models.

With heatmaps, we unveil the hidden intricacies of inter-feature relationships. The color-coded gradients allow us to identify variables that are highly correlated, revealing potential redundancies or multicollinearity. Moreover, they empower us to discern patterns of association that could influence model performance, guiding us toward data transformations or feature selection strategies.

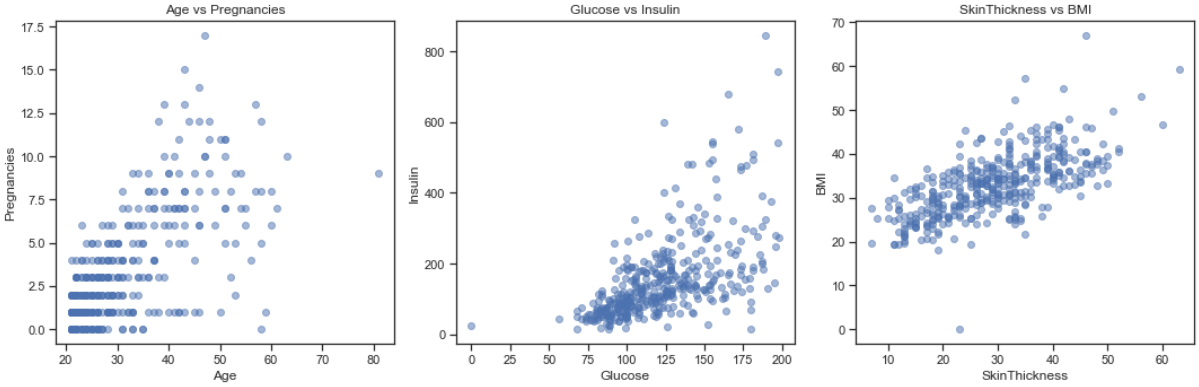
By leveraging these tools, we aim to refine our dataset, enhance feature engineering, and ultimately optimize the performance of our machine learning models. Through careful analysis and interpretation, we endeavor to make strategic modifications that will contribute to the success of our project.

In the sections that follow, we delve into the specific techniques and insights gained from our exploration using boxplots and heatmaps. We will demonstrate how these visualizations guide us in refining the dataset, making informed decisions, and ultimately building predictive models that are both robust and reliable.

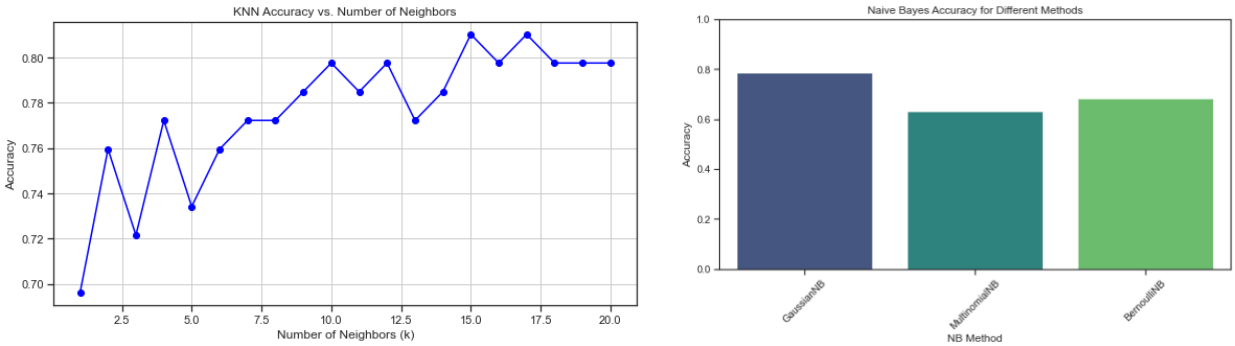




In the boxplots can be seen that there is an increase in all the data when the subject has diabetes. to find out if there is correlation between the different variables a heatmap was made. In the Heatmap there are three datapoints that are interesting.



In the three scatterplots it shows that it looks like there is a correlation between them. Looking at the data that makes sense, for example how older you are the more pregnancies u can have, so it would make sense that it would increase.



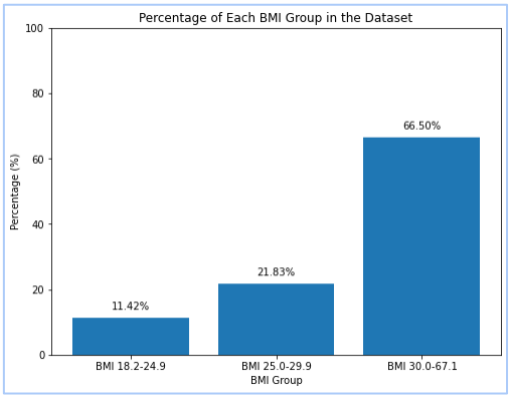
After experimenting with the dataset, the most interesting result came from excluding ‘Age’ and SkinThickness’ from the dataset. With a K of 15 and 17 it shows an 81 % accuracy score, which is a 2% improvement from the original 79%.

BMI Range Overview | Analysis

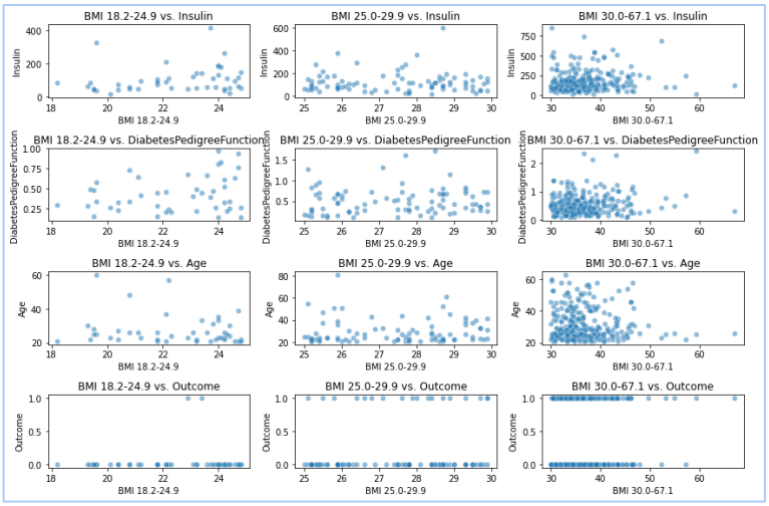
In the next section, we'll take a closer look at our dataset. We're particularly interested in what happens when we split the BMI column into three categories: Normal BMI, Overweight BMI, and Obese BMI. This split will help us better understand how these BMI categories relate to other data points. By organizing the BMI data in this way, we hope to uncover interesting patterns and connections. We want to see how each BMI category affects different aspects of our dataset. This step allows us to focus on specific details and gain a clearer picture of how BMI interacts with other variables.

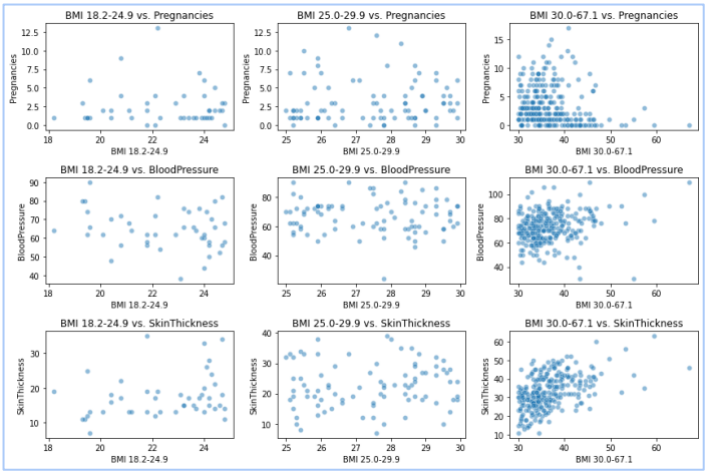
In short, breaking down BMI into these three categories helps us study the data in a more organized and insightful manner. Our goal is to discover valuable insights that enhance our understanding of the dataset and its implications.

After conducting an overview of the BMI range in the dataset, it becomes evident that the 'big overweight' category occupies the largest percentage within our dataset. This observation underscores the significance of this BMI category in our data.



In the subsequent stages of our analysis, we will delve deeper into this dataset. We will conduct visualizations, such as scatterplots, and examine the correlations between each BMI range and other variables. This exploratory process will help us identify which variables exhibit the most significant correlations with BMI indexes. These insights will be further investigated and analyzed in subsequent sections of this report.





- Insulin Correlation

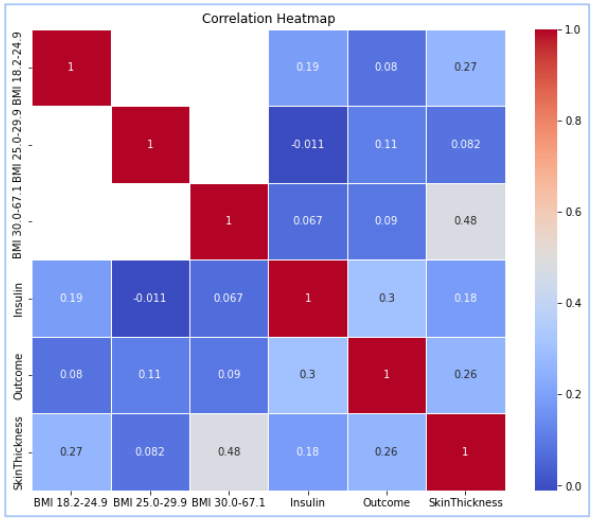
For each BMI range column, there is a notable and rapid increase in insulin levels, indicating a substantial correlation with the BMI index. This suggests a strong association between BMI and insulin levels within our dataset.

- Diabetes (Outcome) Correlation

Our analysis reveals that as the BMI range exceeds the normal levels, specifically in the 25.00-29.9 range, there is a noteworthy increase in the correlation with diabetes among female subjects. In contrast, individuals with BMI levels within the normal range (18.2-24.9) exhibit a comparatively lower correlation with diabetes.

- Skin Thickness Correlation

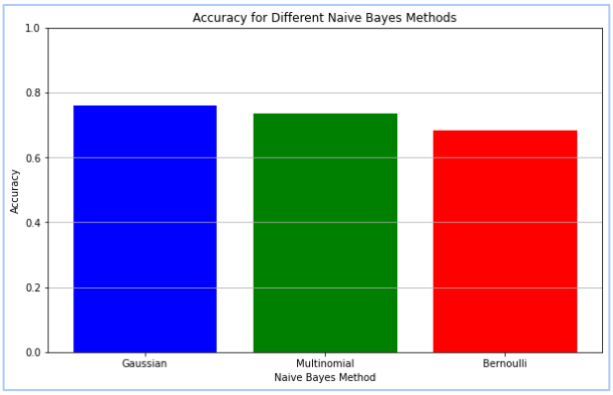
We have observed a correlation between different BMI range levels and skin thickness. Notably, for the 25.0-29.9 and 30.0-67.1 BMI ranges, there is minimal variation in the correlation with skin thickness. However, individuals with BMI levels within the normal range (18.2-24.9) tend to exhibit smaller skin thickness, suggesting a distinct relationship between BMI and skin thickness in this group.

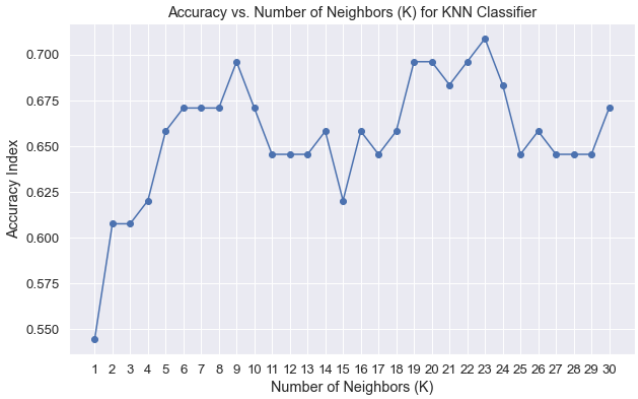


After generating the heatmap, a noteworthy insight emerged: the highest correlation within our dataset was observed between the BMI (highest range) and skin thickness, with a correlation coefficient of 0.48. This finding suggests a meaningful relationship between these two variables that holds the potential for predictive analysis.

The correlation coefficient of 0.48 signifies a moderate positive correlation, indicating that as BMI in the highest range increases, there tends to be a corresponding increase in skin thickness. While this correlation does not imply causation, it does provide a valuable foundation for further investigation and predictive modeling.

This discovery opens the door to exploring predictive models that leverage BMI in the highest range as a potential predictor of skin thickness. Careful consideration can help us uncover the underlying factors contributing to this correlation and its implications for our research.





A Naive Bayes index of 0.76 suggests a reasonably strong performance of the Naive Bayes classification model when applied to a cleaned dataset with three BMI columns per range. This index represents the accuracy or effectiveness of the model in making predictions or classifications.

In this context:

- A Naive Bayes index of 0.76 indicates that the model correctly predicts the outcomes (or classes) approximately 76% of the time. This suggests that the model is fairly accurate in its predictions.

- A value of 0.76 is generally considered to be a good performance, especially in classification tasks. It suggests that the model is capable of making reliable predictions and capturing patterns in the data.

- However, it's important to note that the effectiveness of the model may depend on the specific problem and dataset. It's always a good practice to further evaluate the model using other metrics (such as precision, recall, and F1-score) and perform cross-validation to ensure its generalizability.

In summary, a Naive Bayes index of 0.76 is a positive indication of the model's performance, but it should be accompanied by a comprehensive evaluation to assess its overall quality and suitability for the specific task at hand.

Achieving an accuracy of 70%, the K-Nearest Neighbors (KNN) model demonstrates reasonable performance in predicting the target variable. While this accuracy is satisfactory for some applications, it leaves room for potential improvement. Consideration of specific project goals and potential trade-offs with other metrics is advisable.

Evaluation

Throughout the course of this project, our team embarked on a dynamic journey of data exploration, cleansing, mining, and modeling. This expedition was far from a linear path; instead, it resembled an exhilarating rollercoaster ride filled with meaningful twists and turns.

We gained valuable insights into the intricacies of data cleansing, discovering innovative techniques to enhance the quality and integrity of our dataset. We grappled with questions such as what to clean, what to retain, and how to optimize data for more profound analytical depth.

This venture not only deepened our understanding of data cleaning but also underscored its pivotal role in paving the way for rigorous analysis. We learned how meticulous data preparation can lay the foundation for more accurate and insightful modeling, setting the stage for informed decision-making.

Data Cleaning and Its Impact:

Our investigation shed light on the dynamic interplay between data cleaning and the performance of machine learning models, particularly K-Nearest Neighbors (KNN) and Naive Bayes. We observed how the choice of retaining or removing outliers and zero values in the dataset influenced the accuracy and reliability of our models. This crucial step in data preprocessing demonstrated its potential to significantly affect the outcomes of our analysis.

The Role of Data Ranges:

Furthermore, we explored the effects of segmenting variables into distinct ranges, particularly with the BMI variable. Dividing it into groups—Normal BMI, Overweight BMI, and Obese BMI—allowed us to witness shifts in model performance. This segmentation unveiled nuanced patterns and correlations, enriching our understanding of how different BMI categories interacted with other dataset elements.

Valuable Insights from Visualizations:

Throughout this investigation, visualizations played a pivotal role. We leveraged various visualization techniques to uncover patterns, trends, and relationships within the data. These visualizations provided a comprehensive and intuitive means of exploring data dynamics, enabling us to make informed decisions in data cleaning and modeling.

An Abundance of Experience:

Undoubtedly, this research journey offered a wealth of practical experience. Our team engaged in hands-on data cleaning, analysis, and modeling, enhancing our expertise in the field of data science. We navigated the intricate terrain of data preprocessing and leveraged it as a powerful tool to optimize our models.

In summary, this investigation has been a valuable exploration of the intricate relationship between data cleaning, variable segmentation, visualization, and model performance. It not only contributed to our understanding of the dataset but also provided a rich learning experience in working with real-world data, a testament to the importance of thorough data preparation in data science endeavors. Through our diverse approaches to enhancing the dataset, we uncovered intriguing insights and relationships that shaped our data modeling strategies, demonstrating the profound impact of thoughtful data preparation on the outcomes of data-driven projects.

Appendix

Codes

1. Code for boxplot

```
import pandas as pd

import matplotlib.pyplot as plt


# Assuming your dataset is loaded into a DataFrame named 'df'

df = pd.read_csv(filename.csv')


# List of variables (excluding "Outcome")

variables = ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI',
'DiabetesPedigreeFunction', 'Age']


# Create a boxplot for each variable

for var in variables:

    plt.figure(figsize=(8, 6))

    df.boxplot(column=var, by='Outcome')

    plt.title(f'Boxplot of {var} by Outcome')

    plt.xlabel('Outcome')

    plt.ylabel(var)

    plt.show()
```

2. Code for data cleaning

```
import pandas as pd

import numpy as np

from sklearn.impute import SimpleImputer


# Load the dataset
```

```
file_path = 'diabetesinfo.csv'
df = pd.read_csv(file_path)

# Columns to impute and remove 0 values
columns_to_impute = ['Glucose', 'BloodPressure', 'BMI', 'DiabetesPedigreeFunction', 'Age']
columns_to_remove_zero = ['Insulin', 'SkinThickness']

# Impute missing values with the mean for selected columns
imputer = SimpleImputer(strategy='mean')
df[columns_to_impute] = imputer.fit_transform(df[columns_to_impute])

# Remove rows with 0 values in specific columns
for column in columns_to_remove_zero:
    df = df[df[column] != 0]

# Save the cleaned dataset to a new CSV file
cleaned_file_path = 'diabetesinfo_cleanednew.csv'
df.to_csv(cleaned_file_path, index=False)

print("Cleaning completed. Cleaned dataset saved to 'diabetesinfo_cleaned.csv'.")
```

3. Code for Knn/NB

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB, MultinomialNB, BernoulliNB
from sklearn.metrics import accuracy_score, classification_report
```

```
# Assuming your dataset is loaded into a DataFrame named 'df'
# Replace the following line with your own dataset loading if needed
df = pd.read_csv('diabetesinfo.csv')

# Excluding 'Pregnancies' from the list of features
features = ['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age']

# Split the dataset into features (X) and target (y)
X = df[features]
y = df['Outcome']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Initialize lists to store accuracy values for different Naive Bayes methods
methods = ['Gaussian', 'Multinomial', 'Bernoulli']
accuracy_scores = []

# Loop through different Naive Bayes methods
for method in methods:
    if method == 'Gaussian':
        nb = GaussianNB()
    elif method == 'Multinomial':
        nb = MultinomialNB()
    elif method == 'Bernoulli':
        nb = BernoulliNB()
    nb.fit(X_train, y_train)
```

```
    y_pred_nb = nb.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred_nb)
    accuracy_scores.append(accuracy)

# Create a bar graph to visualize accuracy for different Naive Bayes methods
plt.figure(figsize=(10, 6))
plt.bar(methods, accuracy_scores, color=['blue', 'green', 'red'])
plt.title('Accuracy for Different Naive Bayes Methods')
plt.xlabel('Naive Bayes Method')
plt.ylabel('Accuracy')
plt.ylim(0, 1.0)
plt.grid(axis='y')
plt.show()

# Identify the best Naive Bayes method
best_method = methods[np.argmax(accuracy_scores)]
print(f"The best Naive Bayes method is {best_method} with accuracy {max(accuracy_scores):.2f}")

# Train the Naive Bayes classifier with the best method
if best_method == 'Gaussian':
    best_nb = GaussianNB()
elif best_method == 'Multinomial':
    best_nb = MultinomialNB()
else:
    best_nb = BernoulliNB()
best_nb.fit(X_train, y_train)
y_pred_best_nb = best_nb.predict(X_test)
```

```
# Evaluate the best Naive Bayes model
best_accuracy_nb = accuracy_score(y_test, y_pred_best_nb)
print(f"\nBest Naive Bayes Classifier ({best_method}):")
print(f"Accuracy: {best_accuracy_nb:.2f}")
print(classification_report(y_test, y_pred_best_nb))
```

4. Code for replacing 0 values with the mean

```
import pandas as pd

# Assuming your dataset is stored in a CSV file called 'diabetesinfo.csv'
df = pd.read_csv('diabetesinfo.csv')

# List of columns to exclude (Pregnancies and Outcome)
exclude_columns = ['Pregnancies', 'Outcome']

# Iterate through columns, excluding 'Pregnancies' and 'Outcome'
for column in df.columns:
    if column not in exclude_columns:
        # Calculate the mean excluding 0 values
        mean_value = df[column][df[column] != 0].mean()

        # Replace 0 values with the mean
        df[column] = df[column].replace(0, mean_value)

# Save the updated dataframe to a new CSV file
df.to_csv('diabetesinfo_updated.csv', index=False)
```

5. Code for Scatterplots with 0 in the Dataset

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
```

```
"""
Created on Mon Sep 25 14:10:30 2023

@author: sonyanedbaylo
"""

# import matplotlib.pyplot as plt
# import pandas as pd
# from sklearn.datasets import load_diabetes
# import numpy as np

# # Load the diabetes dataset
# diabetes = pd.read_csv('diabetes.csv')

# variables = ["Pregnancies", "Glucose", "BloodPressure", "SkinThickness", "Insulin", "BMI",
"DiabetesPedigreeFunction", "Age", "Outcome"]

# # Create scatter plots for each variable
# for variable in variables:
#     plt.figure(figsize=(8, 6))
#     plt.scatter(diabetes[variable], diabetes["Age"], alpha=0.5)
#     plt.title(f"Scatter Plot of {variable} vs Age")
#     plt.xlabel(variable)
#     plt.ylabel("Age")
#     plt.grid(True)
#     plt.show()

# diabetes['Age'].mean()
```

```
#     print("Mean of Age: " + str(diabetes['Age'].mean()))


#Second try


import matplotlib.pyplot as plt

import pandas as pd

from sklearn.datasets import load_diabetes

import numpy as np


# Load the diabetes dataset

diabetes = pd.read_csv('diabetes.csv')


variables = ["Pregnancies", "Glucose", "BloodPressure", "SkinThickness", "Insulin", "BMI",
"DiabetesPedigreeFunction", "Age", "Outcome"]


# Create scatter plots for each variable

for variable in variables:

    plt.figure(figsize=(8, 6))

    plt.scatter(diabetes[variable], diabetes["Age"], alpha=0.5)

    plt.title(f"Scatter Plot of {variable} vs Age")

    plt.xlabel(variable)

    plt.ylabel("Age")

    plt.grid(True)

    plt.show()


print("Mean of Age: " + str(diabetes['Age'].mean()))
```

Sources:

Zhou, H., Xin, Y., & Li, S. (2023). A diabetes prediction model based on Boruta feature selection and ensemble learning. *BMC Bioinformatics*, *24*(1). https://doi.org/10.1186/s12859-023-05300-5