Early Prediction of Diabetes Risk: A Machine Learning Approach

Mayank Chauhan (23AIML008) 22 July 2025

**Project Report**

# Project Summary

This project focuses on developing a machine learning model to predict diabetes risk using the ”diabetes-data.csv” dataset, which contains health-related attributes for 768 patients. The dataset includes features such as glucose levels, BMI, and age, with the target variable indicating whether a patient has diabetes. The project involves loading and exploring the dataset, addressing data quality issues like unrealistic zero values, and applying data preprocessing techniques such as scaling and balancing with SMOTE. A K-Nearest Neighbors (KNN) classifier was implemented, with hyperparameter tuning to optimize performance. The final model achieved a test accuracy of 77.55%, with precision, recall, and F1-scores indicating a balanced but improvable performance. Visualizations like scatter matrices and heatmaps provided insights into feature relationships, guiding feature selection. The project highlights the importance of data cleaning, balancing, and evaluation to build a reliable predictive model for clinical use, with recommendations for future improvements.

# Step 2 - Load Your CSV

## Q1. What is the purpose of loading the dataset at the beginning of your project? In your own words, explain why we load the dataset first. What does it allow you to do in the rest of the project?

Loading the dataset at the beginning of the project is the foundational step. It’s like opening your book before you can read it. By loading the data first, we bring the information into our programming environment, making it accessible for all subsequent tasks. This allows us to perform critical operations such as:

* **Data Exploration:** We can examine the data to understand its structure and content.
* **Data Cleaning:** We can identify and fix any errors or missing values.
* **Analysis and Visualization:** We can analyze the data and create charts to uncover patterns and insights.
* **Model Building:** We can use the data to train a machine learning model to make predictions.

## Q2. What kind of information do you think is stored in this dataset? List at least three types of information (columns) you expect to find in the diabetes dataset based on its name. What kind of questions could this data help you answer?

Based on the dataset’s name, ”diabetes-data.csv”, I expect to find various health-related indicators that are com- monly associated with diabetes. Three types of information (columns) I would expect are:

1. **Glucose:** Blood sugar level, a primary indicator for diabetes.
2. **BMI (Body Mass Index):** A measure of body fat based on height and weight, which is a known risk factor.
3. **Age:** Age is a significant factor in the likelihood of developing diabetes. This data could help answer questions like:
   * What is the relationship between glucose levels and the likelihood of having diabetes?
   * Are older individuals at a higher risk of diabetes?
   * Does a higher BMI correlate with a higher incidence of diabetes?

## Q3. If the dataset did not load properly, what possible problems could you check? List 2-3 things you would troubleshoot if the data didn’t load - for example, internet connection, broken link, etc.

If the dataset failed to load, I would troubleshoot the following:

1. **Internet Connection:** A stable internet connection is required to download the dataset from a URL.
2. **URL Validity:** I would check if the provided URL is correct and if the link is still active.
3. **File Format:** I would ensure that the file being loaded is in the expected format (e.g., CSV) and that the reading function is appropriate for that format.

# Step 3 - First Five Rows of Data

## Q1. What are the names of the columns (features) you saw in the dataset? List all the column names you saw. What do you think each one means?

The columns in the dataset are:

* + **Pregnancies:** Number of times pregnant.
  + **Glucose:** Plasma glucose concentration a 2 hours in an oral glucose tolerance test.
  + **BloodPressure:** Diastolic blood pressure (mm Hg).
  + **SkinThickness:** Triceps skin fold thickness (mm).
  + **Insulin:** 2-Hour serum insulin (mu U/ml).
  + **BMI:** Body mass index (weight in kg/(height in m)ˆ2).
  + **DiabetesPedigreeFunction:** A function that scores the likelihood of diabetes based on family history.
  + **Age:** Age in years.
  + **Outcome:** The final result, where 1 indicates the person has diabetes and 0 indicates they do not.

## Q2. What kind of values do you notice in the first few rows? Are they numbers, text, or something else?

The values in the first few rows are all numerical. Specifically, I see both whole numbers (integers) for columns like ’Pregnancies’ and ’Age’, and decimal numbers (floats) for columns like ’BMI’ and ’DiabetesPedigreeFunction’. There is no text data in the initial rows.

## Q3. Which 2 or 3 columns do you think might be most useful for predicting whether a person has diabetes?

I believe the following columns will be most useful for predicting diabetes:

1. **Glucose:** This is a direct measure of blood sugar, a key diagnostic criterion for diabetes.
2. **BMI:** Obesity is a major risk factor for type 2 diabetes, so BMI is likely to be a strong predictor.
3. **Age:** The risk of developing diabetes increases with age.

## Q4. Do you notice anything strange, surprising, or possibly wrong in the first few rows of the data?

Yes, I noticed some potentially strange values. Several columns, including **Glucose**, **BloodPressure**, **SkinThick- ness**, **Insulin**, and **BMI**, have values of 0. These zero values are unrealistic for a living person and likely represent missing or erroneous data that will need to be addressed.

## Q5. Based on the first few rows, what kind of questions would you like to answer using this data?

Based on the initial data, I would like to explore questions such as:

* + Is there a threshold for **Glucose** or **BMI** levels above which the likelihood of having diabetes increases dramatically?
  + Does the number of **Pregnancies** have a significant impact on the risk of developing diabetes?
  + How does the **DiabetesPedigreeFunction** score correlate with the actual **Outcome**?

# Step 4 - Check Dataset Structure

## Q1. How many records (rows) are in the dataset?

There are **768 rows** in the dataset. Each row represents the health data of a single individual.

## Q2. How many columns are there, and what does each one seem to represent?

There are **9 columns** in the dataset. Each column represents a specific health-related attribute or feature:

* + **Pregnancies:** Number of pregnancies
  + **Glucose:** Blood sugar level
  + **BloodPressure:** Blood pressure reading
  + **SkinThickness:** A measure related to body fat
  + **Insulin:** Insulin level
  + **BMI:** Body Mass Index
  + **DiabetesPedigreeFunction:** A score indicating genetic predisposition to diabetes
  + **Age:** The person’s age
  + **Outcome:** Whether the person has diabetes (1) or not (0)

## Q3. Did any column have missing data? How do you know?

Based on the initial check of non-null counts, it appears that no columns have missing data, as each column shows 768 non-null values out of 768 total entries. However, as noted earlier, the presence of zero values in columns where it is biologically impossible suggests that these zeros might be placeholders for missing data.

## Q4. What are the different types of data in the columns? Why does that matter?

The columns contain two main types of numerical data:

* + **Integers (int64):** Whole numbers, found in columns like ’Pregnancies’, ’Glucose’, ’BloodPressure’, ’SkinThickness’, ’Insulin’, ’Age’, and ’Outcome’.
  + **Floats (float64):** Decimal numbers, found in columns like ’BMI’ and ’DiabetesPedigreeFunction’.

Knowing the data types is crucial because it determines the types of mathematical operations we can perform and helps us identify if the data is stored in an appropriate format for analysis.

## Q5. Based on the structure, do you think this dataset is ready to use for analysis? Why or why not?

No, I do not think the dataset is ready for analysis. While the structure is sound in terms of rows and columns with no explicitly missing values, the presence of unrealistic zero values in several key features like ’Glucose’ and ’BloodPressure’ indicates that the data is not yet clean. These zero values need to be investigated and handled before the data can be used for reliable analysis or model building

# Step 5 - Summary Statistics

## Q1. Which column has the highest average (mean) value? What does this tell you about that feature?

The **Insulin** column has the highest average (mean) value. This indicates that, on average, the insulin levels in this dataset are numerically larger than the values of other features. This could be due to the units of measurement for insulin (mu U/ml).

## Q2. Do you notice any columns that have 0 as the minimum value? Is that realistic?

Yes, the columns **Glucose**, **BloodPressure**, **SkinThickness**, **Insulin**, and **BMI** all have a minimum value of 0. This is not realistic for a living person. A blood pressure or glucose level of 0, for instance, is not compatible with life. These zero values are likely errors or representations of missing data.

## Q3. Which columns show the most variation (look at the ”std” or standard deviation)? Why does variation matter?

The **Insulin** column shows the most variation, as indicated by its high standard deviation. High variation means the values in that column are spread out over a wider range. Understanding variation is important because features with high variance can sometimes dominate the learning process of a machine learning model if the data is not scaled.

## Q4. Based on the 25%, 50%, and 75% values (percentiles), which columns seem to have balanced distribu- tions?

The **BloodPressure** and **BMI** columns appear to have relatively balanced distributions. The 25th, 50th (median), and 75th percentiles are reasonably close to each other, suggesting that the data is not heavily skewed and is more symmetrically distributed around the median.

## Q5. Which features do you think will be most important in predicting the outcome (diabetes or not)? Why?

Based on the summary statistics and general medical knowledge, I believe **Glucose** and **BMI** will be the most important features. Higher glucose levels are a direct indicator of diabetes, and a high BMI is a well-known risk factor. The statistics likely show a significant difference in the mean and distribution of these features between the two outcome groups.

# Step 6 - Create a Safe Copy and Prepare Important Columns

## Q1. Why is it a good idea to make a separate copy of your dataset before making changes?

It’s a good practice to create a copy of the dataset to preserve the original data. This acts as a safety net, allowing you to revert to the original state if you make any irreversible errors during data cleaning or preprocessing. It ensures the integrity of the source data remains intact.

## Q2. What do you think could happen if we don’t focus on specific columns for cleaning and just use all the columns together?

If we don’t focus on specific columns for cleaning, we might introduce errors or noise into our model. Some columns may not have issues with unrealistic values, and applying a cleaning method (like replacing zeros) to them could be unnecessary or even detrimental. Focusing on specific, problematic columns ensures that our cleaning process is targeted and effective.

# Step 7 - Checking for Missing Values

## Q1. Why is it important to check for missing values in your dataset before doing analysis or prediction?

Checking for missing values is crucial because most machine learning algorithms cannot handle them. Missing data can lead to biased or inaccurate analysis and can cause models to fail or produce unreliable predictions. Identifying and handling missing values is a critical step in data preprocessing.

## Q2. Which columns in your dataset had missing values? How many were missing in each?

While the .isnull().sum() method did not show any explicitly missing (NaN) values, the presence of zeros in the following columns suggests they contain missing information:

* + **Glucose:** 5 missing values
  + **BloodPressure:** 35 missing values
  + **SkinThickness:** 227 missing values
  + **Insulin:** 374 missing values
  + **BMI:** 11 missing values

## Q3. Do you think it’s okay for health-related data like ”Glucose” or ”BloodPressure” to have a value of 0 or be empty?

No, it is not okay for these health-related features to have a value of 0 or be empty. A value of 0 for **Glucose**, **BloodPressure**, or **BMI** is biologically impossible for a living person and indicates a data quality issue, likely representing a missing measurement.

## Q4. If you ignored missing values completely and trained a machine learning model, what problems might happen?

Ignoring missing values (in this case, the unrealistic zeros) would lead to a poorly performing model. The model would learn from incorrect data, leading to biased and inaccurate predictions. For example, it might incorrectly associate a value of 0 for ’Glucose’ with a non-diabetic outcome, which is medically nonsensical.

# Step 8 - Visualizing Data Distributions with Histograms

## Q1. What can you say about the distribution of the ’Glucose’ and ’Blood Pressure’ values?

The histograms for **Glucose** and **BloodPressure** show that the data for both features is approximately normally distributed, resembling a bell curve. However, both histograms also show a spike at the value 0, which confirms the presence of the unrealistic zero values that need to be addressed.

## Q2. Which columns seem to have many zero values (e.g., Insulin, Skin Thickness)? Do you think zero is a valid value for these health measurements? What might zero represent in this dataset?

The columns with a large number of zero values are **Insulin** and **SkinThickness**. A zero value is not a valid measurement for these features in a living person. In this dataset, the zero values likely represent missing data, where the measurement was not recorded for a particular individual.

## Q3. Look at the ’Age’ and ’BMI’ histograms. What age group or BMI range do most patients fall under?

* + **Age:** The ’Age’ histogram is skewed to the right, indicating that most patients in the dataset are in the younger age groups, primarily between 20 and 30 years old.
  + **BMI:** The ’BMI’ histogram appears to be normally distributed, with the majority of patients having a BMI in the range of 30-35.

# Step 9 - Fix Zero Values & Understand Data

## Q1. Why is it important to replace zero values in health-related columns like Blood Pressure, Glucose, or BMI?

It is crucial to replace these zero values because they are not physiologically possible and represent data errors. A BMI of 0, for example, is meaningless and would imply a person has no body mass. Including these zeros in our analysis would skew the results and lead to a machine learning model that learns from and makes predictions based on nonsensical data.

## Q2. What method did you use to replace the zero values, and why is it a good choice?

I replaced the zero values with the **median** of each respective column. The median is a good choice because it is a robust measure of central tendency that is not affected by outliers or extreme values. Using the mean could be misleading if the data is skewed by very high or low values.

## Q3. After cleaning the data, what changes did you observe in the histograms of any two columns?

After replacing the zero values, the histograms for columns like **Glucose** and **Insulin** changed significantly. The prominent spike at zero disappeared, and the distributions now look more realistic and continuous. For instance, the ’Glucose’ histogram now shows a more complete bell-shaped curve without the artificial bar at zero.

## Q4. What would happen if you didn’t fix the zero values? How would it affect future analysis or predictions?

If the zero values were not fixed, any statistical analysis (like calculating the mean) would be inaccurate. A machine learning model trained on this flawed data would learn incorrect relationships between features and the outcome, leading to poor and unreliable predictions. It would essentially be learning from garbage data.

## Q5. Which column do you think has the most unusual values, and why?

The **Insulin** column appears to have the most unusual values. Even after replacing the zeros, the histogram for ’Insulin’ is heavily skewed to the right, with a large number of values clustered at the lower end and a long tail of higher values. This indicates a wide range of insulin levels and the presence of some very high values (outliers), which might be due to specific medical conditions in some patients.

# Step 10 - Check the Size of Dataset

## Q1. How many rows and columns are present in your dataset? What does each row and column represent?

The dataset has **768 rows** and **9 columns**. Each row represents a single patient’s health record, and each column represents a different health feature, such as ’Glucose’, ’BMI’, or ’Age’.

## Q2. Why is it important to check the shape of a dataset before doing any analysis or machine learning?

Checking the shape of the dataset is important to get a basic understanding of its size. It tells us if we have a sufficient amount of data to work with and helps verify that the data has been loaded correctly, without any missing rows or columns.

## Q3. If the shape showed fewer columns than expected, what possible issues could have caused this?

If the shape showed fewer columns than expected, it could be due to issues like:

* + An error in the CSV file itself, such as incorrect delimiters causing columns to be merged.
  + A mistake in the code used to load the data, where some columns were unintentionally excluded.

## Q4. Imagine you are working with a dataset that has 5,000 rows and only 2 columns. Do you think this is good enough for training a model? Why or why not?

While 5,000 rows is a good number of samples, having only 2 columns (features) might not be sufficient for building a robust model, depending on the complexity of the problem. With only two features, the model might not have enough information to capture the underlying patterns in the data, potentially leading to underfitting. More features would likely provide a more comprehensive view and improve the model’s predictive power.

## Q5. If your dataset had 0 rows after loading, what would be your first steps to fix the issue?

If the dataset had 0 rows after loading, my first steps would be to:

1. Verify the file path or URL to ensure it is correct.
2. Check the file itself to make sure it is not empty.
3. Examine the code used to load the data for any errors that might be preventing it from reading the rows correctly.

# Step 11 - Check How Many People Have Diabetes

## Q1. How many people in the dataset have diabetes and how many do not?

In the dataset:

* + **500** people do not have diabetes (Outcome = 0).
  + **268** people have diabetes (Outcome = 1).

## Q2. Which group is larger in the dataset - people with diabetes or without diabetes?

The group of people **without diabetes** is larger than the group with diabetes.

## Q3. Why do you think it’s important to check the number of diabetic and non-diabetic cases before building a model?

It’s important to check the balance between the classes because a significant imbalance can lead to a biased model. If one class is much larger than the other, the model might simply learn to predict the majority class all the time to achieve high accuracy, while failing to identify the minority class, which is often the class of interest.

## Q4. If one group is much bigger than the other, what kind of problem could that cause in prediction?

A large imbalance can cause the model to have a high accuracy but a low predictive power for the minority class. For example, if 90% of the data is non-diabetic, a model that always predicts ”non-diabetic” would be 90% accurate but completely useless for its intended purpose of identifying diabetic individuals.

## Q5. How would you describe this dataset to someone who wants to build a diabetes prediction app?

I would describe this as a valuable but imbalanced dataset. It contains relevant health indicators for 768 patients, with about one-third of them having diabetes. While useful, the class imbalance needs to be addressed using techniques like SMOTE to ensure the resulting prediction model is fair and effective at identifying both diabetic and non-diabetic individuals.

# Step 12 - Plot Scatter Matrix

## Q1. Which two features (columns) in the scatter matrix seem to have the strongest positive relationship?

From the scatter matrix, **SkinThickness** and **BMI** appear to have a noticeable positive relationship. The points in their scatter plot tend to form an upward trend, suggesting that as one value increases, the other tends to increase as well.

## Q2. Do you notice any features that do not seem related to others?

**Pregnancies** and **BloodPressure** do not seem to have a strong linear relationship with most other features. The scatter plots involving these variables show a more random, cloud-like distribution of points, indicating a weak correlation.

## Q3. Were there any unusual points (outliers) visible in the scatter matrix?

Yes, there are visible outliers, particularly in the **Insulin** column. There are several data points with very high insulin values that are far from the main cluster of data. These outliers could be genuine extreme values or data entry errors and might need further investigation.

## Q4. What do the histograms on the diagonal line of the scatter matrix tell you about the data?

The histograms on the diagonal show the distribution of each individual feature. For example:

* + The histogram for **Age** is skewed to the right, showing a concentration of younger patients.
  + The histogram for **BMI** is roughly bell-shaped, indicating a normal distribution.

## Q5. Based on the scatter matrix, which features do you think might be most useful for predicting diabetes?

Based on the visual patterns, **Glucose** seems to be a very useful feature. While the scatter matrix doesn’t show the ’Outcome’ directly, the separation in other plots (like the pairplot with hue) will likely show a clearer distinction based on Glucose levels.

# Step 13 - Plot Scatter Matrix (With ’hue=outcome‘)

## Q1. Which features show a clear difference between people with and without diabetes?

The pairplot with hue=’Outcome’ clearly shows that **Glucose** is the most distinguishing feature. The distributions for diabetic (Outcome=1) and non-diabetic (Outcome=0) individuals are visibly separated, with the diabetic group having significantly higher glucose levels. **BMI** and **Age** also show some separation between the two groups, but it is less pronounced than with Glucose.

## Q2. Is there any feature that does not help much in separating diabetic and non-diabetic individuals?

**BloodPressure** and **SkinThickness** do not seem to be very effective at separating the two groups. The distributions for diabetic and non-diabetic individuals in the plots for these features largely overlap, suggesting they are less powerful predictors on their own.

## Q3. What extra insight do you gain by using hue=’Outcome’ in the pairplot that you do not get in the plain pairplot?

Using hue=’Outcome’ is incredibly insightful because it allows us to see how the relationship between features differs for the two outcome classes. For example, a plain scatter plot of Glucose vs. BMI just shows a general correlation. But with hue=’Outcome’, we can see that the cluster of points representing diabetic individuals tends to be concentrated in the region of higher Glucose and higher BMI, an insight that is completely hidden in the plain plot.

## Q4. If you had to choose only one pairplot to include in your final project report, which one would you choose and why?

I would choose the pairplot of **Glucose** vs. **BMI** with hue=’Outcome’. This plot provides a clear visual separation between the two classes and highlights the interplay of two of the most significant risk factors for diabetes. It’s easy to understand and powerfully communicates the core relationships in the data.

## Q5. How can the pairplot with hue=’Outcome’ help you decide which features might be important for predicting diabetes?

The pairplot with hue=’Outcome’ helps identify important features by visually showing which variables create the best separation between the classes. If the colored points (representing diabetic and non-diabetic) form distinct, non-overlapping clusters for a particular feature, that feature is likely to be a strong predictor. **Glucose** is the prime example of this in our dataset.

# Step 14 - Plot a Heatmap

## Q1. Which two features in the dataset show the strongest positive relationship?

The heatmap shows that **Age** and **Pregnancies** have a relatively strong positive correlation. This makes sense, as the number of pregnancies a person has had is likely to increase with their age. Another strong positive relationship is seen between **Glucose** and **Outcome**.

## Q2. Which feature shows the strongest relationship with the Outcome (diabetes)? Why do you think this feature might be important for predicting diabetes?

**Glucose** shows the strongest relationship with the **Outcome**. This is expected, as high blood glucose levels are the primary clinical indicator of diabetes. Therefore, this feature is critically important for prediction.

## Q3. Were there any pairs of features that showed little or no relationship (value close to 0)?

Yes, several pairs showed weak correlations, with values close to 0. For example, the relationship between **BloodPressure** and **DiabetesPedigreeFunction** is very weak. This means that there is no strong linear trend between these two variables; knowing the value of one gives you little information about the value of the other.

## Q4. What insights did the heatmap give you that you could not easily see in the pair plot?

The heatmap provides a concise, quantitative summary of the linear relationships between all pairs of features at a glance. While the pairplot shows the relationships visually, the heatmap gives us the precise correlation coefficient, making it easier to compare the strength of different relationships directly. For example, it’s easier to definitively say that the correlation between ’Glucose’ and ’Outcome’ is stronger than that between ’BMI’ and ’Outcome’ by just comparing the numbers in the heatmap.

## Q5. How can this heatmap help in choosing the right features for building a prediction model?

The heatmap helps in feature selection by highlighting which features are most strongly correlated with the **Outcome**. Features with a higher correlation (either positive or negative) with the target variable are likely to be more useful in a predictive model. Based on the heatmap, **Glucose**, **BMI**, and **Age** would be the top choices for a diabetes prediction model.

# Step 15 - Scaling the Data Before Modeling

## Q1. Why did we need to scale the data before building a machine learning model?

We need to scale the data because different features are measured in different units and have vastly different ranges (e.g., ’Insulin’ vs. ’DiabetesPedigreeFunction’). Many machine learning algorithms, like KNN, are sensitive to the scale of the data and can be biased towards features with larger values. Scaling ensures that all features are treated equally by the model, leading to better performance.

## Q2. Which columns were selected for scaling, and which column was left out? Why?

All the feature columns (Pregnancies, Glucose, etc.) were selected for scaling. The **Outcome** column was left out. This is because the ’Outcome’ is our target variable—the answer we are trying to predict. It is not an input feature, so we do not scale it.

## Q3. What do you think would happen if we skipped the scaling step? How might it affect the model’s performance?

If we skipped scaling, the model’s performance would likely be poor. The features with larger numerical ranges, like ’Insulin’ or ’Glucose’, would dominate the distance calculations in an algorithm like KNN, while features with smaller ranges would be largely ignored. This would lead to a model that is biased and less accurate.

## Q4. After scaling, do the actual values of the features (like BMI or Glucose) still have the same meaning? Explain.

After scaling, the actual numerical values change, but their meaning and the information they carry remain the same. Scaling transforms the values to a common range, but the relative relationships and the ranking of the data points within each feature are preserved. A high scaled value for ’Glucose’ still represents a high blood sugar level relative to other individuals in the dataset.

## Q5. How is the new table after scaling different from the original one? What’s one benefit of creating a separate table instead of replacing the original?

The new table contains the scaled data, where each feature has a mean of 0 and a standard deviation of 1. The original table contains the raw, unscaled values. The benefit of creating a separate table is that it preserves the original data, which might be needed for interpretation or other types of analysis later on. It is a non-destructive workflow that adds a layer of safety.

# Step 16 & 17 - Test Train Data Split & Balancing with SMOTE

## Q1. Why do you think it is important to balance the number of diabetic and non-diabetic cases in the dataset before training a model?

It is important to balance the dataset to prevent the model from becoming biased towards the majority class. If the model is trained on imbalanced data, it may achieve high accuracy by simply predicting the more frequent outcome, while failing to learn the patterns of the minority class (in this case, diabetic individuals), which is often the class we are most interested in predicting correctly.

## Q2. What problems could arise if we did not apply SMOTE and trained the model on unbalanced data?

Without SMOTE, the model would likely have poor **recall** for the diabetic class. It would miss a large number of actual diabetic patients because it would be biased towards predicting the non-diabetic class, which is more prevalent in the training data. For example, it might correctly identify 90% of non-diabetic cases but only 30% of diabetic cases, resulting in a model that is not clinically useful.

## Q3. After applying SMOTE, what changes did you observe in the number of samples for each class?

* + **Before SMOTE:** The training data was imbalanced, with more non-diabetic samples than diabetic ones.
  + **After SMOTE:** The number of samples for the diabetic and non-diabetic classes in the training set became equal. SMOTE achieved this by creating synthetic samples for the minority (diabetic) class.

## Q4. Do you think synthetic (artificial) data created by SMOTE is as useful as real data? Why or why not?

While real data is always preferable, synthetic data created by SMOTE is very useful in situations of class imbalance. SMOTE creates new samples by interpolating between existing minority class samples, making the synthetic data realistic and representative of the minority class. This allows the model to learn the decision boundary more effectively, leading to better performance. However, it’s important to be aware that SMOTE can sometimes create noise if the classes are not well-separated.

## Q5. How does balancing the data help improve the fairness or accuracy of your final machine learning model?

Balancing the data with SMOTE improves the **fairness** of the model by ensuring it pays equal attention to both the majority and minority classes. This leads to a model with better overall performance, particularly in terms of metrics like **recall** and **F1-score**, which are crucial for evaluating performance on imbalanced datasets. It results

in a model that is more reliable and useful in real-world applications where correctly identifying the minority class is critical.

# Step 18 & 19 - Finding the Best K for KNN Classifier

## Q1. Why do we test the KNN model with different values of k (number of neighbors)?

We test different values of ’k’ to find the optimal balance between bias and variance for our model.

* + A very small ’k’ (like 1) can lead to a model that is too sensitive to noise and outliers (**high variance**).
  + A very large ’k’ can lead to a model that is too general and overlooks local patterns (**high bias**).

By testing a range of ’k’ values, we can identify the one that provides the best generalization performance on unseen data.

## Q2. Based on your results, which value of k gave the best performance on the test data? Was this value also the best for training data? Why might they be different?

The best performance on the **test data** was achieved at **k=11**. The training data, however, achieved its highest score at **k=1**. They are different because a ’k’ of 1 perfectly ”memorizes” the training data, resulting in 100% accuracy on the data it has already seen. However, this model does not generalize well to new, unseen test data. The best ’k’ for the test data represents the model that has learned the underlying patterns without overfitting to the training set.

## Q3. Did you notice any k values where the model performed very well on training data but poorly on testing data?

Yes, for **k=1**, the training accuracy was 100%, but the testing accuracy was significantly lower. This large gap indicates **overfitting**. The model has learned the training data too well, including its noise, and is unable to generalize its knowledge to new data.

## Q4. Why is it important to compare both training accuracy and testing accuracy when building a machine learning model?

It is crucial to compare both because it helps us diagnose whether the model is overfitting or underfitting.

* + **High training accuracy, low testing accuracy:** Indicates overfitting.
  + **Low accuracy on both:** Indicates underfitting.
  + **Similar and high accuracy on both:** Indicates a well-generalized model.

Looking only at training accuracy can be very misleading, as it doesn’t tell us how the model will perform in a real-world scenario with new data.

## Q5. Imagine you are recommending a KNN model for predicting diabetes in a real hospital setting. Which value of k would you choose and why? Justify your answer with your testing results.

I would recommend the model with **k=11**. This value yielded the highest accuracy on the test data, which simulates how the model would perform on new patients. While k=1 had perfect training accuracy, its poor performance on the test set makes it unreliable. A k of 11 provides the best balance, demonstrating a good ability to generalize from the training data to make accurate predictions in a real-world setting.

# Step 20 - Plotting Training vs Testing Accuracy (KNN)

## Q1. What do you observe about the training accuracy as the value of K increases?

The training accuracy starts at its highest point (100% for k=1) and generally decreases as the value of ’k’ increases. This is because with a larger ’k’, the model’s decision boundary becomes smoother and less able to fit the fine-grained noise in the training data, leading to a decrease in its ability to ”memorize” the training set.

## Q2. At which value of K does the testing accuracy seem to be the highest? Why might this be a good choice for the final model?

The testing accuracy is highest at **k=11**. This is a good choice for the final model because our primary goal is to have a model that performs well on new, unseen data. High testing accuracy means the model is effective at making correct predictions in real-life terms, such as correctly identifying patients who have diabetes.

## Q3. Is there a big difference between training and testing accuracy at some K values? What could that indicate about the model?

Yes, there is a large gap between training and testing accuracy for low values of ’k’, especially k=1. This large gap is a classic sign of **overfitting**. The model has learned the training data too specifically and has failed to capture the general underlying trend, leading to poor performance on the test set.

## Q4. Why is it important to compare both training and testing accuracy when selecting a machine learning model? What could go wrong if we only looked at one of them?

Comparing both is essential to ensure the model is **generalizable**. If we only looked at training accuracy, we might choose a model that is severely overfit (like k=1) and would fail miserably in a real-world application. If we only looked at testing accuracy without considering the training accuracy, we might not fully understand the model’s learning behavior. The goal is to find a model where both accuracies are high and the gap between them is small.

# Step 21 - Evaluating the Final Model with K-Nearest Neighbors (KNN)

## Q1. What value of K (number of neighbors) did you choose for the KNN model, and why?

I chose **K=11** for the final model. This value was selected because it yielded the highest accuracy on the test set during our hyperparameter tuning process, indicating it is the best value for making predictions on new, unseen data.

## Q2. What is the final accuracy of your model on the test data? Do you think this accuracy is high enough for making real predictions? Why or why not?

The final accuracy on the test data with K=11 is **77.55%**. Whether this is ”high enough” is debatable and depends on the application’s context. For a non-critical screening tool, it might be acceptable. However, for a clinical diagnostic tool, a higher accuracy would be desirable. It’s a good starting point, but there is room for improvement.

## Q3. Is your model performing better on training data or testing data? What does this tell you about the model?

The model performs better on the training data than on the testing data. However, the gap between the training and testing accuracy for K=11 is not excessively large, which suggests that the model is relatively well-balanced and is not severely overfitting.

## Q4. How did applying SMOTE (oversampling) affect your model’s performance?

Applying SMOTE was crucial for improving the model’s performance. By balancing the training data, SMOTE allowed the model to learn the patterns of the minority (diabetic) class more effectively. This likely resulted in a model with better recall for diabetic cases than a model trained on the imbalanced data.

## Q5. If you had more time or resources, what would you try next to improve the model’s accuracy or usefulness?

To improve the model, I would try the following:

* + **Try Different Algorithms:** I would experiment with other classification algorithms like Logistic Regression, Support Vector Machines, or Random Forests to see if they can achieve better performance.
  + **Feature Engineering:** I would explore creating new features from the existing ones that might have stronger predictive power.
  + **Collect More Data:** A larger and more diverse dataset could help build a more robust and accurate model.

# Step 22 - Calculating Precision, Recall, and F1 Score

## Q1. What does your model’s precision score tell you?

The precision score tells us that out of all the patients the model predicted as having diabetes, **69%** actually had diabetes. A higher precision score would mean fewer false alarms (i.e., fewer non-diabetic people being incorrectly told they have diabetes).

## Q2. What does the recall score reveal about your model’s performance?

The recall score of **71%** reveals that the model was able to correctly identify **71%** of all the patients who actually have diabetes in the test set. This means it missed about 29% of the diabetic cases. In a medical context, a high recall is very important to ensure that as many true cases as possible are caught.

## Q3. Why is the F1 score useful in this project?

The F1-score is useful because it provides a single metric that balances both **precision** and **recall**. In a situation like diabetes prediction, we care about both minimizing false alarms (precision) and catching as many true cases as possible (recall). The F1-score gives us a more holistic view of the model’s effectiveness than accuracy alone, especially on an imbalanced dataset.

## Q4. How did SMOTE affect the balance of your training data, and do you think it improved your model’s performance?

SMOTE created a balanced training set by oversampling the minority class. This undoubtedly improved the model’s performance. Without SMOTE, the model would have been heavily biased towards the non-diabetic class, likely resulting in a much lower recall for the diabetic class. Balancing the data forced the model to learn the characteristics of diabetic patients more effectively.

## Q5. Based on your evaluation (precision, recall, F1), what would you suggest to improve this model’s performance further?

Based on the evaluation, I would suggest the following improvements:

* + **Tune the model for higher recall:** In a medical screening context, it might be better to have more false positives than to miss actual cases. We could try to optimize the model’s threshold to favor a higher recall.
  + **Use a more advanced model:** Algorithms like Gradient Boosting or a well-tuned Random Forest might be able to capture more complex patterns in the data and achieve a better balance of precision and recall.
  + **Feature Selection:** We could perform more rigorous feature selection to ensure we are only using the most informative features, which can sometimes lead to a better, less noisy model.