Lawrence Technological University

INT 7623
Data Science for Business
Final Project

Diabetes Prediction using K-means algorithm

-Sridevi Anandan
Keerthana Godala

Introduction

According to World Health Organization (2011) diabetes is a chronic, metabolic disease that can be identified by high level of blood glucose, which after a period of time may lead to severe damage. Early detection and intervention are crucial for effectively managing diabetes and reducing its associated complications.

Dataset:

The Diabetes prediction dataset is a collection of medical and demographic data from patients, along with their diabetes status (positive or negative), it has attributes such as age, gender, body mass index (BMI), hypertension, heart disease, smoking history, HbA1c level, and blood glucose level.

The dataset used in this project is comprehensive, and the EDA will provide valuable insights into the relationships between different health indicators and their role in predicting diabetes onset.

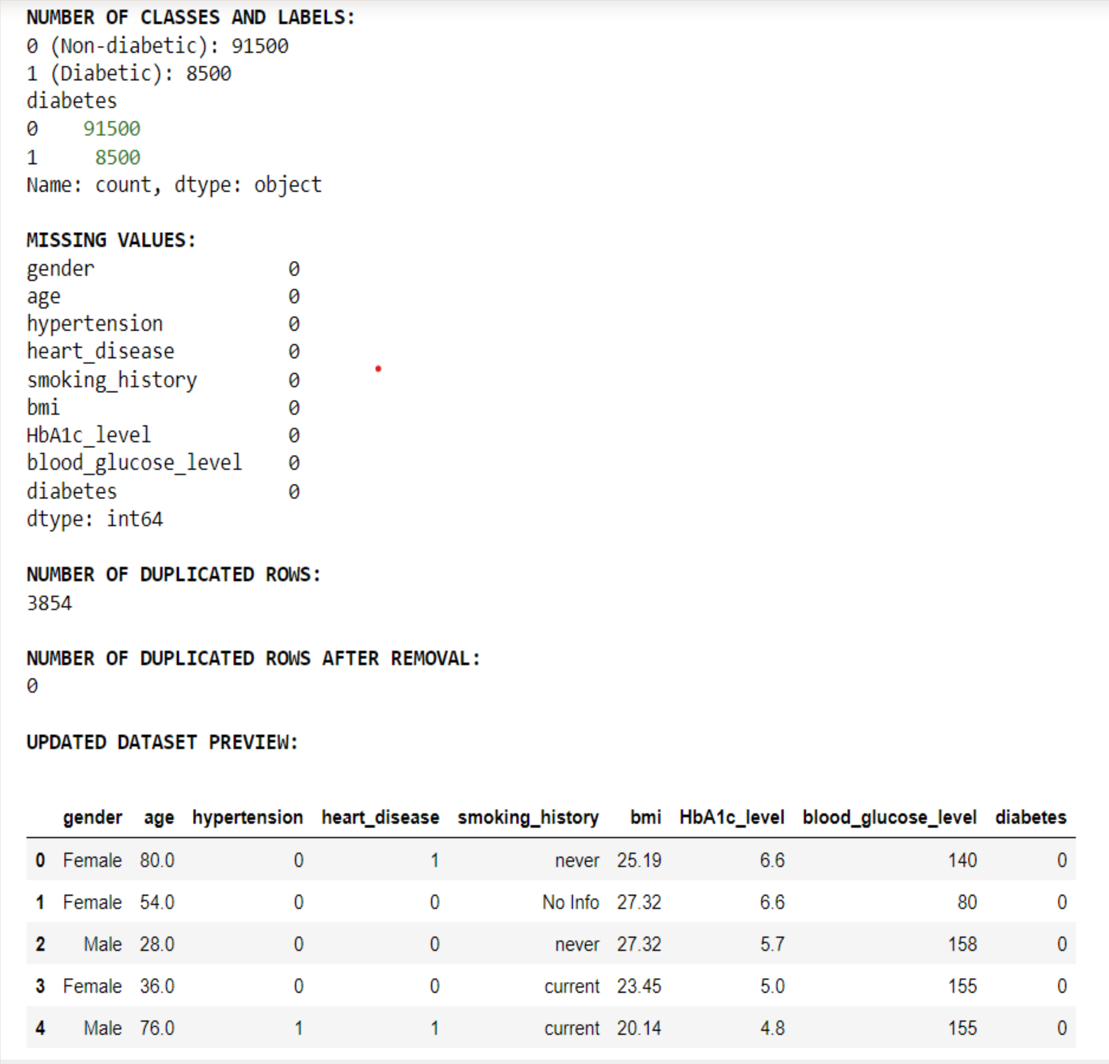
Key details of the dataset:

Size: The dataset consists of a certain number of instances, each representing a patient.

Number of Measurements: It includes several features such as age, gender, body mass index (BMI), hypertension, heart disease, smoking history, HbA1c level, and blood glucose level.

Type of Measurements: The features in the dataset are a mix of categorical and numerical variables, reflecting various aspects of a patient's medical and demographic profile.

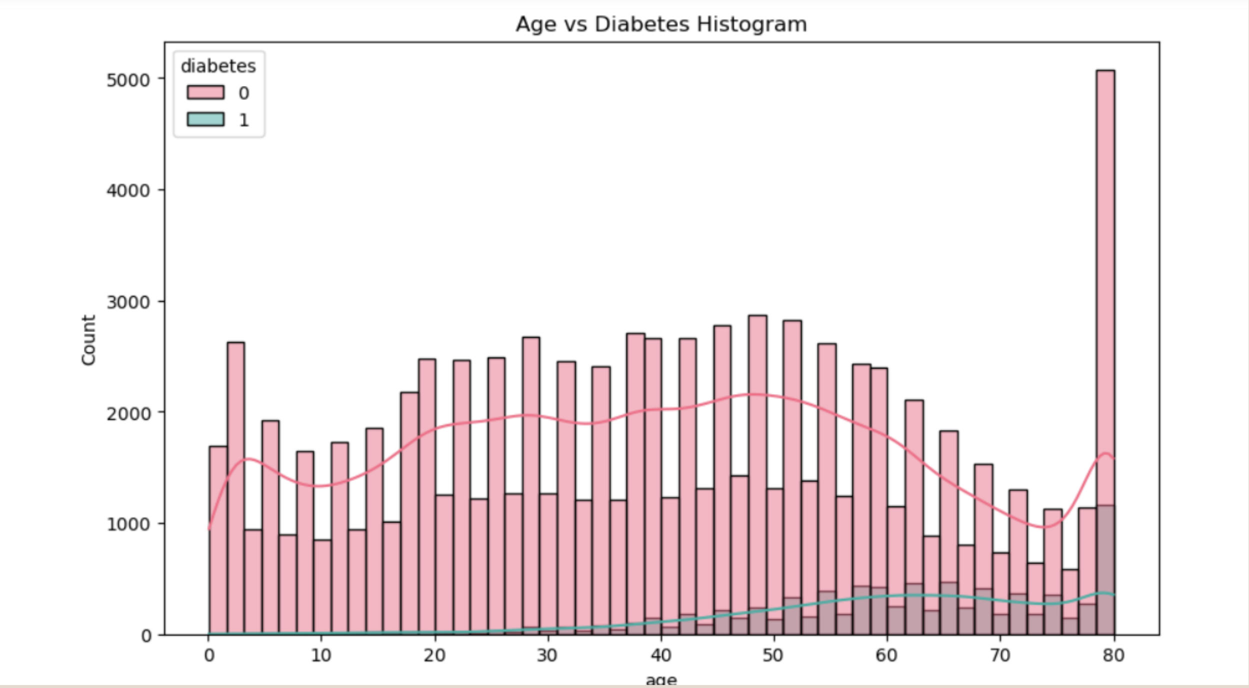
Number of Classes and Labels: The target variable is the diabetes status, which has two classes: positive or negative. Patients are labelled based on whether they have been diagnosed with diabetes or not.

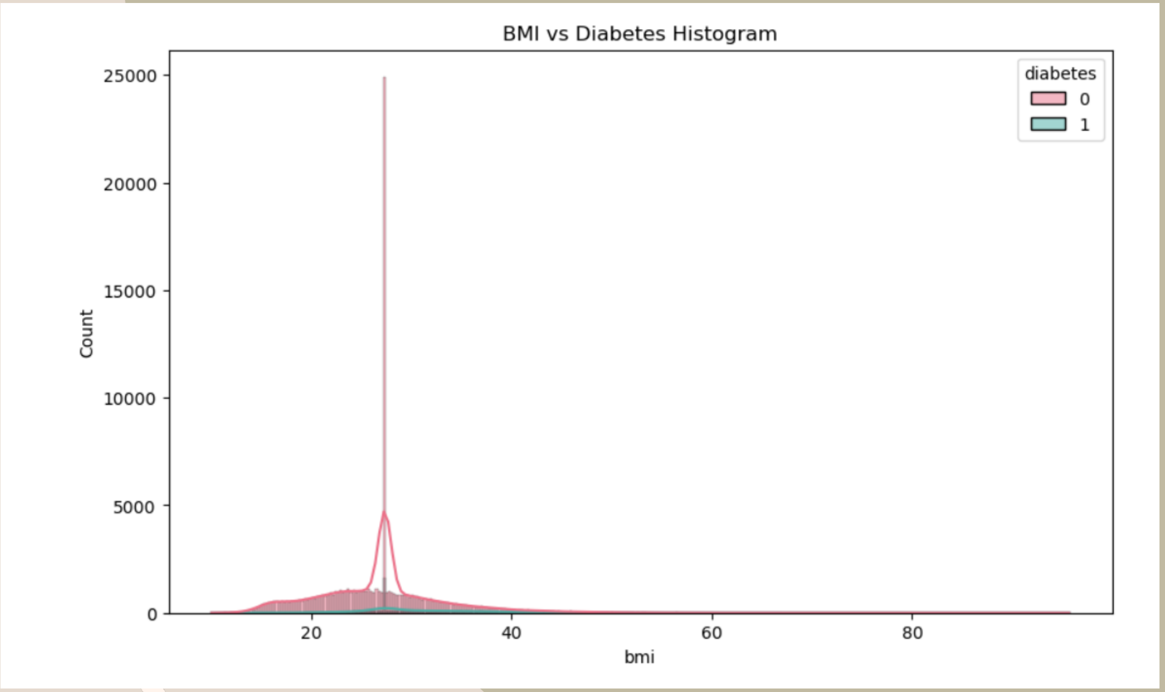


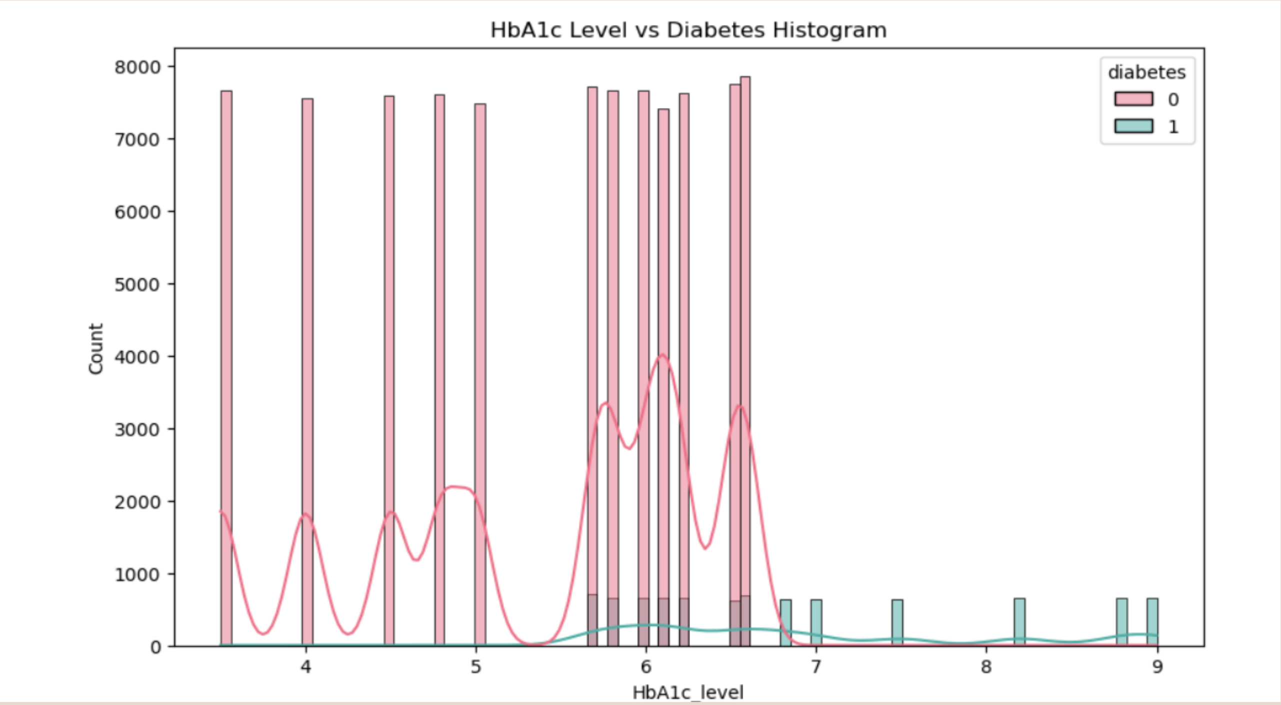
```
NUMBER OF CLASSES AND LABELS:
0 (Non-diabetic): 91500
1 (Diabetic): 8500
diabetes
0    91500
1     8500
Name: count, dtype: object

MISSING VALUES:
gender                 0
age                    0
hypertension           0
heart_disease          0
smoking_history        0
bmi                    0
HbA1c_level            0
blood_glucose_level    0
diabetes               0
dtype: int64

NUMBER OF DUPLICATED ROWS:
3854

NUMBER OF DUPLICATED ROWS AFTER REMOVAL:
0

UPDATED DATASET PREVIEW:
```

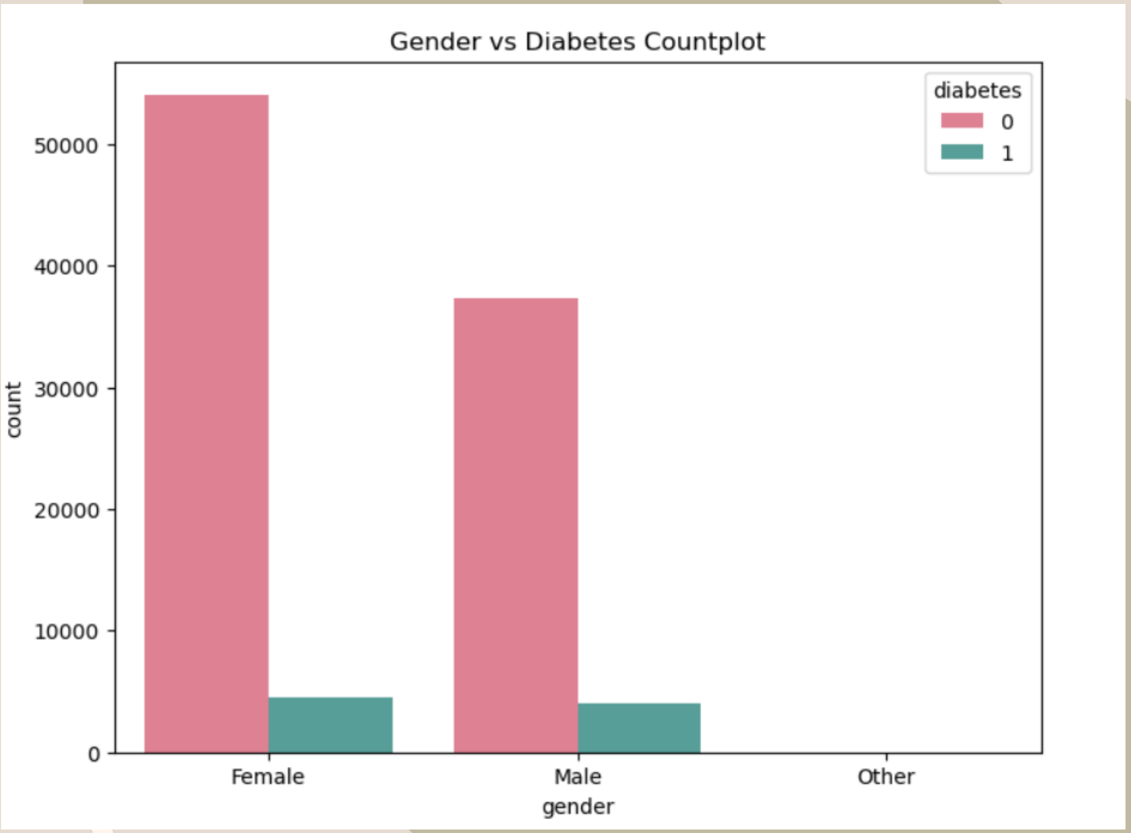
	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
0	Female	80.0	0	1	never	25.19	6.6	140	0
1	Female	54.0	0	0	No Info	27.32	6.6	80	0
2	Male	28.0	0	0	never	27.32	5.7	158	0
3	Female	36.0	0	0	current	23.45	5.0	155	0
4	Male	76.0	1	1	current	20.14	4.8	155	0

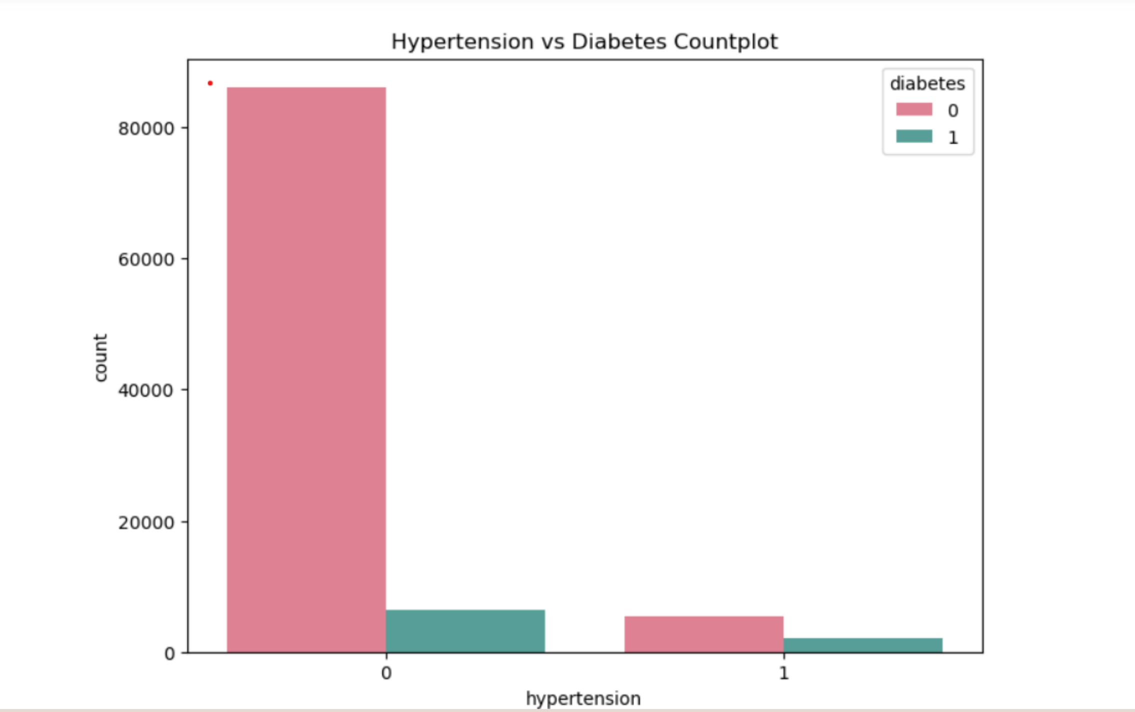
Loading the data and visualizing the insights

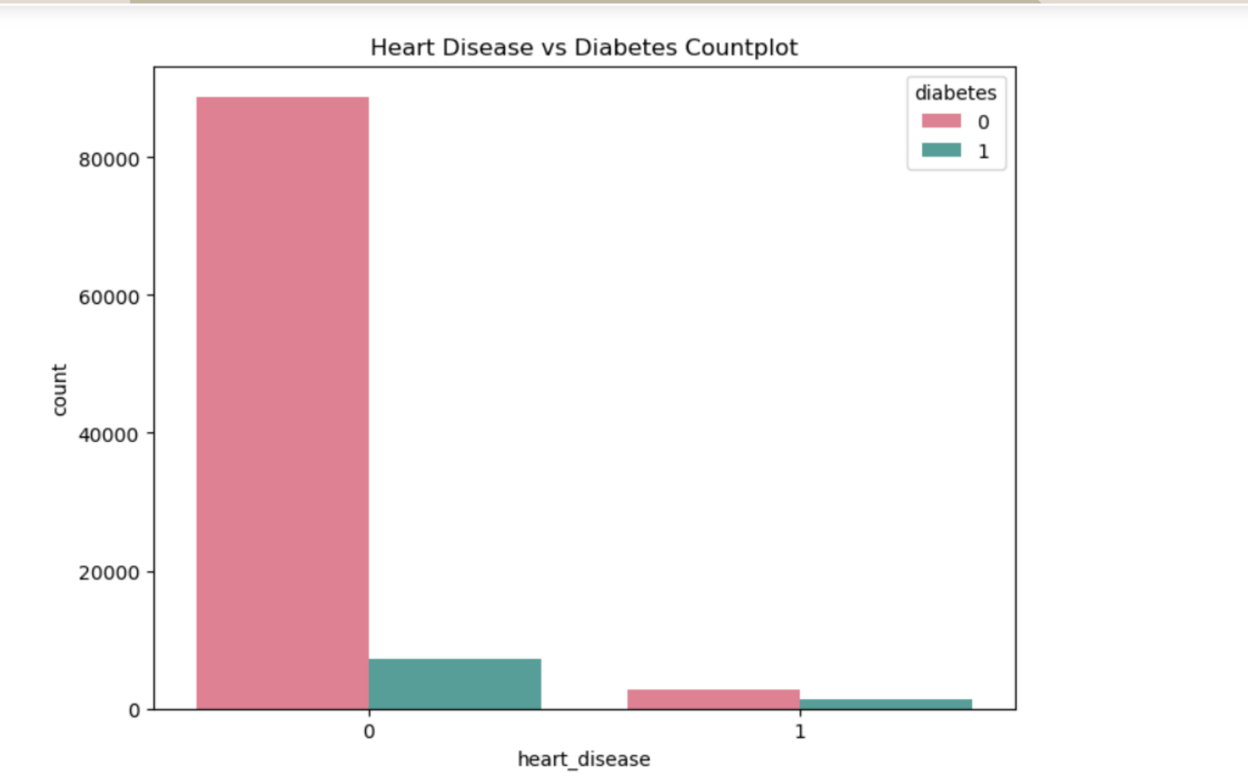





HbA1c Level vs Diabetes Histogram
diabetes
0
1
Count
0
1000
2000
3000
4000
5000
6000
7000
8000
4
5
6
7
8
9
HbA1c_level


Gender vs Diabetes Countplot
diabetes
0
1
count
0
10000
20000
30000
40000
50000
Female
Male
Other
gender

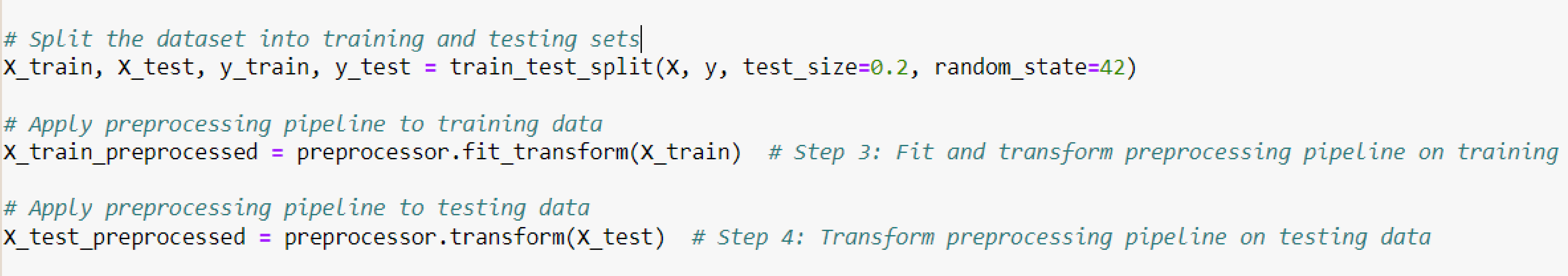

Hypertension vs Diabetes Countplot
diabetes
0
1
count
80000
60000
40000
20000
0
0
1
hypertension


Heart Disease vs Diabetes Countplot
diabetes
0
1
count
80000
60000
40000
20000
0
0
1
heart_disease

Preparing the dataset

- CONVERT CATEGORICAL VARIABLES INTO NUMERICAL REPRESENTATIONS USING TECHNIQUES LIKE ONE-HOT ENCODING OR LABEL ENCODING.
- CLEAN THE DATA BY IMPUTING MISSING VALUES USING STRATEGIES SUCH AS MEAN, MEDIAN, OR MODE IMPUTATION, OR DROP ROWS/COLUMNS WITH MISSING VALUES DEPENDING ON THE CONTEXT.
- DETECT AND HANDLE OUTLIERS BY EITHER REMOVING THEM OR TRANSFORMING THEM USING TECHNIQUES SUCH AS WINSORIZATION OR ROBUST SCALING.
- STANDARDIZE NUMERICAL FEATURES: SCALE NUMERICAL FEATURES TO HAVE A MEAN OF 0 AND A STANDARD DEVIATION OF 1 USING TECHNIQUES LIKE Z-SCORE STANDARDIZATION OR MIN-MAX SCALING. THIS STEP ENSURES THAT ALL VARIABLES ARE ON THE SAME SCALE, WHICH IS IMPORTANT FOR K-NN.
- SCALE NUMERICAL FEATURES TO A RANGE BETWEEN 0 AND 1 USING MIN-MAX SCALING.

Data partitioning



```
# Split the dataset 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)

# Apply preprocessing pipeline to training data
X_train_preprocessed = preprocessor.fit_transform(X_train)  # Step 3: Fit and transform preprocessing pipeline on training

# Apply preprocessing pipeline to testing data
X_test_preprocessed = preprocessor.transform(X_test)  # Step 4: Transform preprocessing pipeline on testing data
```

In this code:

- X contains all the features except for the target variable 'diabetes'.
- y contains only the target variable 'diabetes'.
- The train_test_split function is used to split X and y into training and testing sets.
- The parameter test_size=0.2 specifies that 20% of the data will be used for testing, while the remaining 80% will be used for training. random_state=42 ensures reproducibility of the split.
- Now you have X_train, X_test, y_train, and y_test containing the respective sets for training and testing your machine learning model.

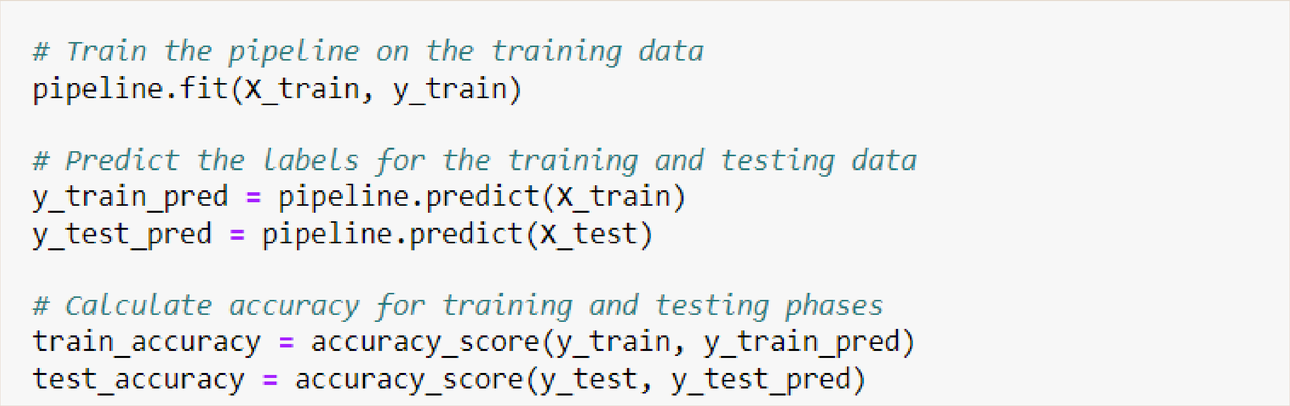
Different values of K

- In our dataset, there might be clear distinctions between individuals with and without diabetes. Therefore, starting with *k*=2. It allows us to explore whether there are indeed two main clusters in the data.

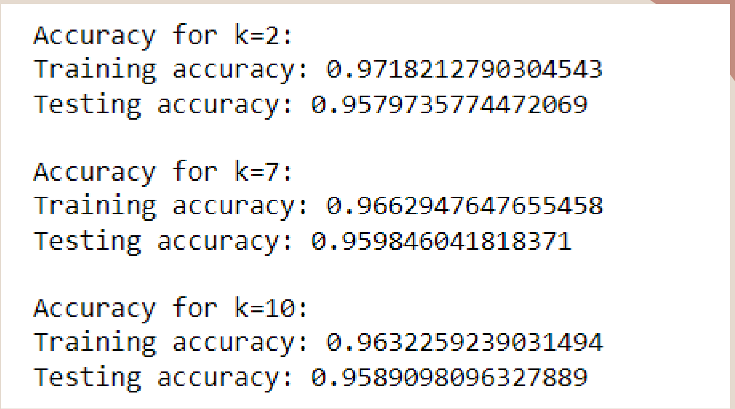
- With *k*=7, we can investigate whether there are distinct subgroups within the dataset that may correspond to different risk profiles or disease . since diabetes prediction is a multifactorial problem influenced by various medical and demographic factors, a larger value of *k* helps in capturing more complex relationships between features.

- Choosing *k*=10 provides an even broader perspective on the clustering patterns within the data. By increasing *k* to 10, we aim to explore the possibility of more refined clusters and potentially uncover hidden patterns or relationships among the features that contribute to diabetes prediction.

Training phase and testing phase:



```
# Train the pipeline on the training data
pipeline.fit(X_train, y_train)

# Predict the labels for the training and testing data
y_train_pred = pipeline.predict(X_train)
y_test_pred = pipeline.predict(X_test)

# Calculate accuracy for training and testing phases
train_accuracy = accuracy_score(y_train, y_train_pred)
test_accuracy = accuracy_score(y_test, y_test_pred)
```

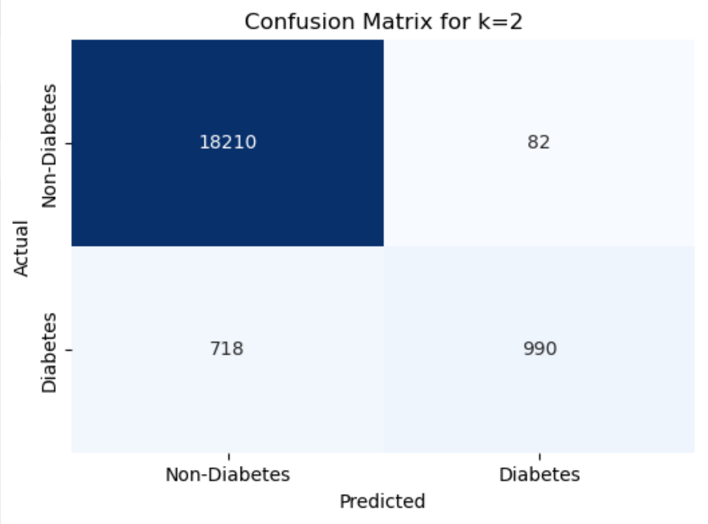


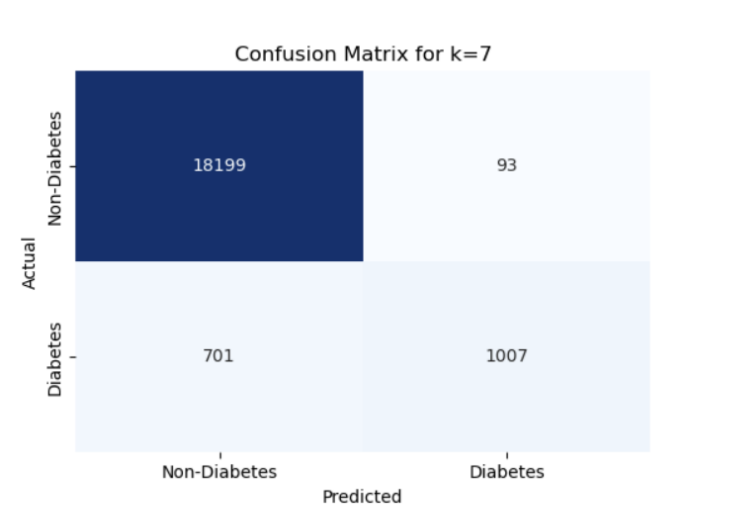
```
Accuracy for k=2:
Training accuracy: 0.9718212790304543
Testing accuracy: 0.9579735774472069

Accuracy for k=7:
Training accuracy: 0.9662947647655458
Testing accuracy: 0.959846041818371

Accuracy for k=10:
Training accuracy: 0.9632259239031494
Testing accuracy: 0.9589098096327889
```

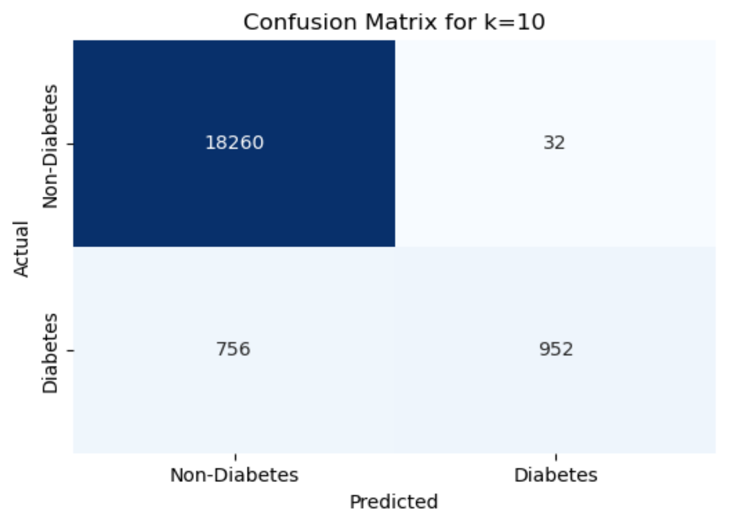
In the testing phase, the trained K-NN model is evaluated on a separate dataset called the testing dataset. The testing dataset contains instances that were not used during the training phase and serves as an independent measure of the model's performance. The accuracy metric provides insights into how well the model generalizes to unseen data.

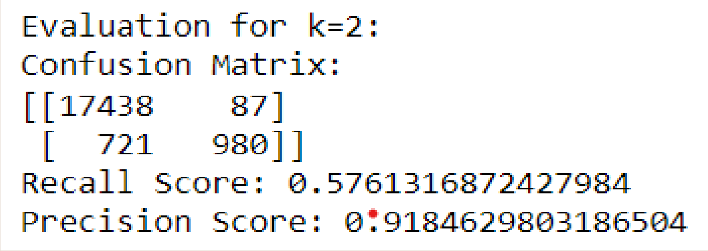
Based on the testing accuracies, *k*=7 performs better than the other values of *k*.

Evaluation Phase

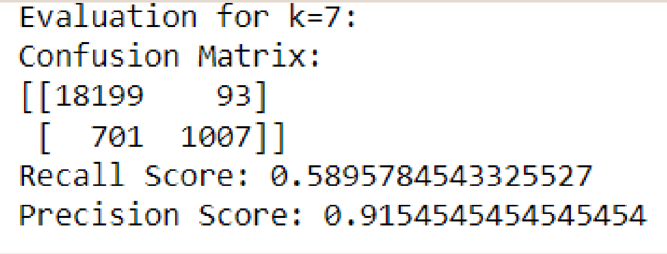




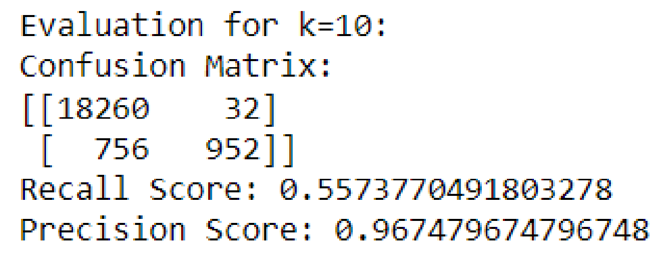




```
Evaluation for k=2:
Confusion Matrix:
[[17438    87]
 [  721   980]]
Recall Score: 0.5761316872427984
Precision Score: 0.9184629803186504
```



```
Evaluation for k=7:
Confusion Matrix:
[[18199    93]
 [  701  1007]]
Recall Score: 0.5895784543325527
Precision Score: 0.9154545454545454
```



```
Evaluation for k=10:
Confusion Matrix:
[[18260    32]
 [  756   952]]
Recall Score: 0.5573770491803278
Precision Score: 0.967479674796748
```

Best model and conclusion:

- To determine the best model, we need to understand the significance of each metric. In medical diagnosis, missing a positive case (false negative) can be detrimental as it means not identifying a patient who needs treatment. Therefore, based on the importance of recall the fact that *k*=7 achieves higher recall while still maintaining a relatively high accuracy, k=7 appears to be the preferred model for diabetes prediction.

- Performance Evaluation: The KNN models, particularly the k=7 model, exhibit high accuracy, with precision and recall metrics indicating their ability to accurately classify individuals into diabetic and non-diabetic categories.

- Model Selection: Among the different k values considered, the k=7 KNN model emerges as the preferred choice based on its balanced performance in terms of precision, recall, and accuracy.

- Utility in Clinical Practice: The KNN models offer practical utility in clinical practice by providing reliable predictions for diabetes diagnosis.

- Further Enhancements: Strategies such as feature selection, data augmentation, model ensemble, and continuous monitoring can be implemented to improve predictive accuracy, robustness, and interpretability in real-world healthcare settings.

Thank you