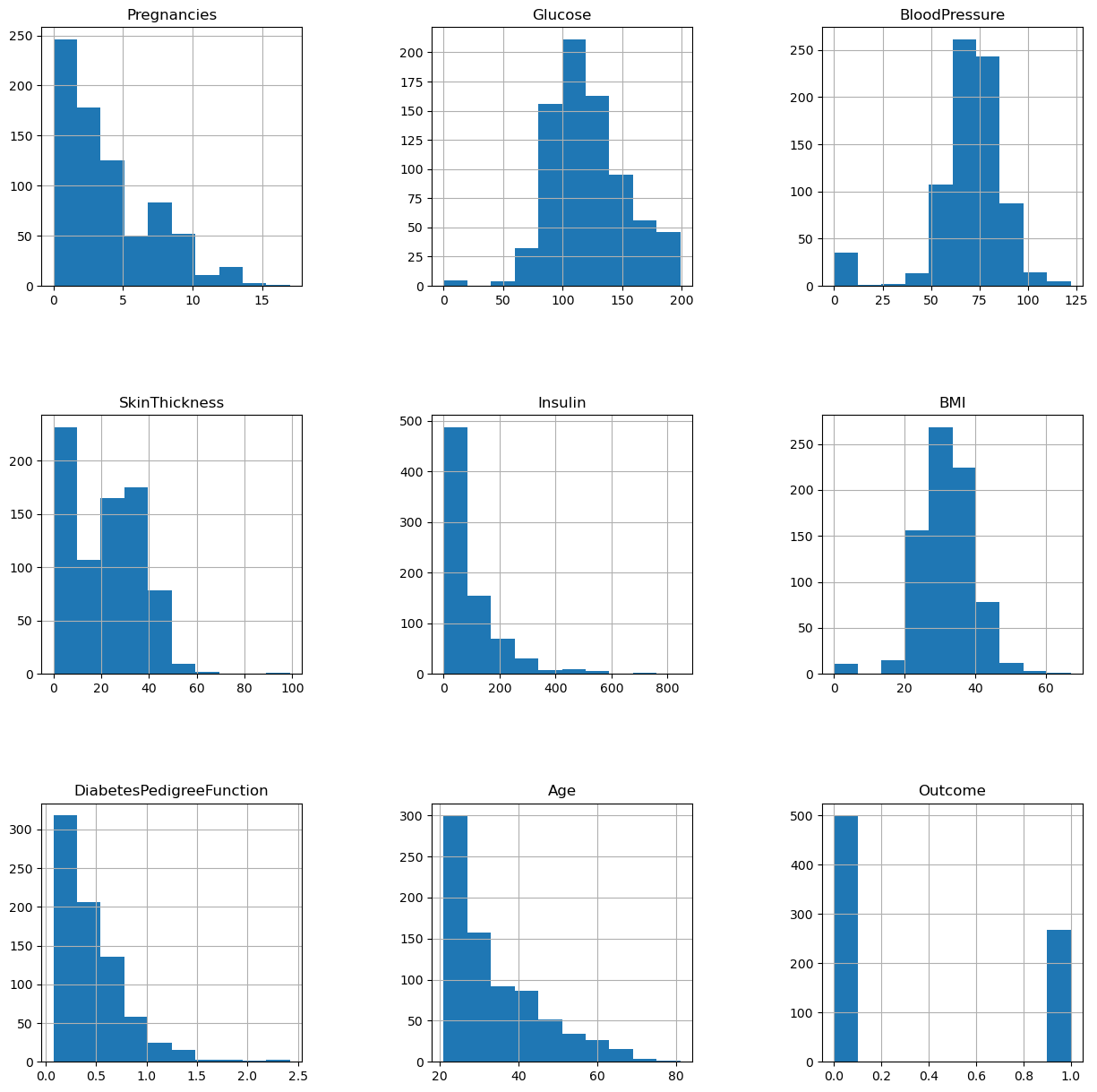
# Diabeties among Indians dataset

Diabeties is a new age problem. It occurs when the body is either unable to produce enough insulin or cannot effectively use the insulin it produces. Diabeties can cause a lot of issues and health complications if not managed properly. There are majorly three types of diabeties. Type 1 which is a condition in which body fails to produce insulin, as they dont have beta cells in the pancreas to produce it. These type is independed depended on insulin. Second type is type 2 diabeties, which is common form of diabeties can devolop at any age. These is due to lack of the beta cells to produce enough insulin or body get insulin resistant due to many factors such as obsity,poor diet etc.. Third catogory is gestational diabeties, this happens in the preanant women who have never had diabetes before but who have high blood sugar levels during pregnancy. It affects both the mother and the baby’s health. Though it usually goes away after giving birth, women who’ve had gestational diabetes have a higher risk of developing type 2 diabetes later in life.

This dataset originates from the National Institute of Diabetes and Digestive and Kidney Diseases. The aim of this dataset is to predict, through diagnostic measures, whether a patient has diabetes.

import pandas as pd  
import matplotlib.pyplot as plt  
  
# Load the data  
df = pd.read\_csv('diabetes.csv')  
  
# Display the first few rows of the dataframe  
print(df.head())  
  
# Display summary statistics  
print(df.describe())  
  
# Fill missing values with the mean of the column  
df = df.fillna(df.mean())  
  
# Plot histograms for each variable  
df.hist(figsize=(15, 15))  
plt.subplots\_adjust(hspace=0.5, wspace=0.5) # Adjust the spacing between subplots  
plt.show()

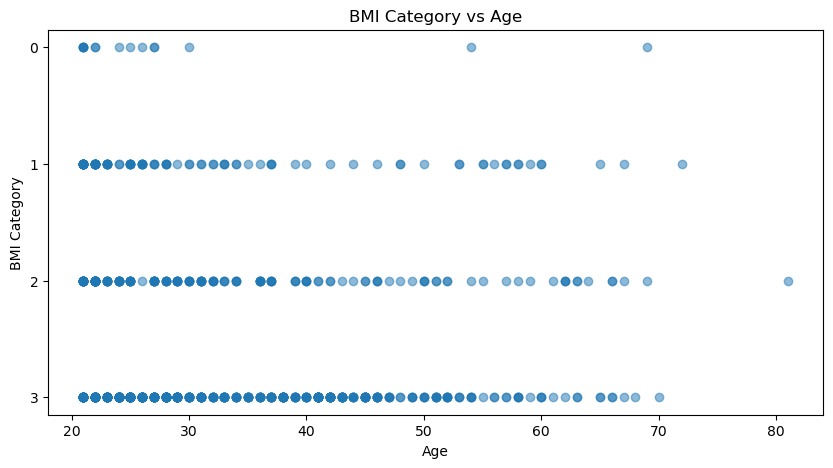
Pregnancies Glucose BloodPressure SkinThickness Insulin BMI \  
0 6 148 72 35 0 33.6   
1 1 85 66 29 0 26.6   
2 8 183 64 0 0 23.3   
3 1 89 66 23 94 28.1   
4 0 137 40 35 168 43.1   
  
 DiabetesPedigreeFunction Age Outcome   
0 0.627 50 1   
1 0.351 31 0   
2 0.672 32 1   
3 0.167 21 0   
4 2.288 33 1   
 Pregnancies Glucose BloodPressure SkinThickness Insulin \  
count 768.000000 768.000000 768.000000 768.000000 768.000000   
mean 3.845052 120.894531 69.105469 20.536458 79.799479   
std 3.369578 31.972618 19.355807 15.952218 115.244002   
min 0.000000 0.000000 0.000000 0.000000 0.000000   
25% 1.000000 99.000000 62.000000 0.000000 0.000000   
50% 3.000000 117.000000 72.000000 23.000000 30.500000   
75% 6.000000 140.250000 80.000000 32.000000 127.250000   
max 17.000000 199.000000 122.000000 99.000000 846.000000   
  
 BMI DiabetesPedigreeFunction Age Outcome   
count 768.000000 768.000000 768.000000 768.000000   
mean 31.992578 0.471876 33.240885 0.348958   
std 7.884160 0.331329 11.760232 0.476951   
min 0.000000 0.078000 21.000000 0.000000   
25% 27.300000 0.243750 24.000000 0.000000   
50% 32.000000 0.372500 29.000000 0.000000   
75% 36.600000 0.626250 41.000000 1.000000   
max 67.100000 2.420000 81.000000 1.000000



In this project, I conducted an analysis to predict whether a woman aged 21 years or older has diabetes, and to identify the factors that make her prone to diabetes. The dataset for this project was sourced from Kaggle and is composed of a CSV file containing 769 rows and 9 columns.

This shows the dataset, on the demography of the dataset. There is a high number of people around age 21-30 than any other catogory. Blood pressure shows a normal distribution and BMI also shows normal distribution. Which signifies most of the people are in the range of overweight (BMI 25-30), another intresting observation is the blood pressure as the mean is around 69 which is quite low for even normal people. Diabeties have a correation with blood pressue and the mean and std deviation is below the normal standered, which need to be explored further.

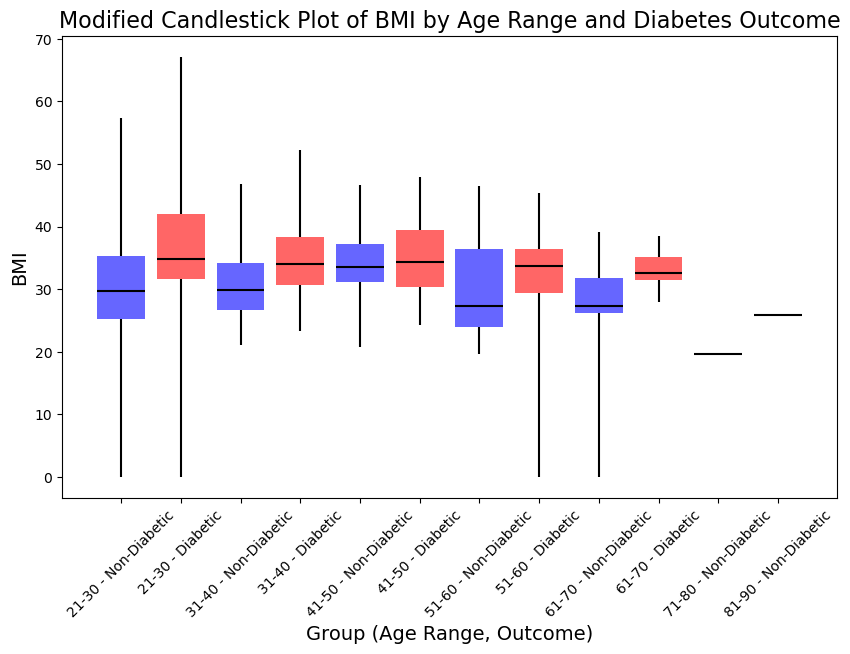
def categorize\_bmi(bmi):  
 if bmi < 18.5:  
 return '0'  
 elif bmi < 25:  
 return '1'  
 elif bmi < 30:  
 return '2'  
 else:  
 return '3'  
  
df['BMI Category'] = df['BMI'].apply(categorize\_bmi)  
plt.figure(figsize=(10, 5))  
plt.scatter(df['Age'], df['BMI Category'], alpha=0.5)  
plt.xlabel('Age')  
plt.ylabel('BMI Category')  
plt.title('BMI Category vs Age')  
plt.show()



As this shows the data set is skewed on the obsity factor, most of the participants are overweight or obsese.

import matplotlib.pyplot as plt  
import seaborn as sns  
import pandas as pd  
  
data = pd.read\_csv('diabetes.csv')  
  
  
  
# Categorize age into bins  
bins = [20, 30, 40, 50, 60, 70, 80, 90]  
labels = ['21-30', '31-40', '41-50', '51-60', '61-70', '71-80', '81-90']  
data['Age Range'] = pd.cut(data['Age'], bins=bins, labels=labels)  
  
# Group by age range and outcome, and calculate the quartiles, min and max of BMI  
summary = data.groupby(['Age Range', 'Outcome'])['BMI'].describe()[['min', '25%', '50%', '75%', 'max']].reset\_index()  
  
# Plot the modified candlestick chart  
fig, ax = plt.subplots(figsize=(10, 6))

for i, row in summary.iterrows():  
 x = i  
 lower\_whisker = row['min']  
 lower\_box = row['25%']  
 upper\_box = row['75%']  
 upper\_whisker = row['max']  
 median = row['50%']  
   
 # Body of the candle  
 ax.bar(x, upper\_box - lower\_box, bottom=lower\_box, color='blue' if row['Outcome'] == 0 else 'red', alpha=0.6)  
   
 # Lower whisker  
 ax.vlines(x, lower\_whisker, lower\_box, color='black')  
   
 # Upper whisker  
 ax.vlines(x, upper\_box, upper\_whisker, color='black')  
   
 # Median line  
 ax.hlines(median, x-0.4, x+0.4, color='black')  
  
# Adding labels and title  
plt.xlabel('Group (Age Range, Outcome)', fontsize=14)  
plt.ylabel('BMI', fontsize=14)  
plt.title('Modified Candlestick Plot of BMI by Age Range and Diabetes Outcome', fontsize=16)  
plt.xticks(range(len(summary)), [f"{row['Age Range']} - {'Non-Diabetic' if row['Outcome'] == 0 else 'Diabetic'}" for \_, row in summary.iterrows()], rotation=45)  
  
# Display the plot  
plt.show()



-The blue bars represent non-diabetic individuals, and the red bars represent diabetic individuals. -The height of each bar represents the interquartile range (IQR) of BMI within each age group. -The whiskers extend from the minimum to the maximum BMI within each age group. -The horizontal line within each bar represents the median BMI.

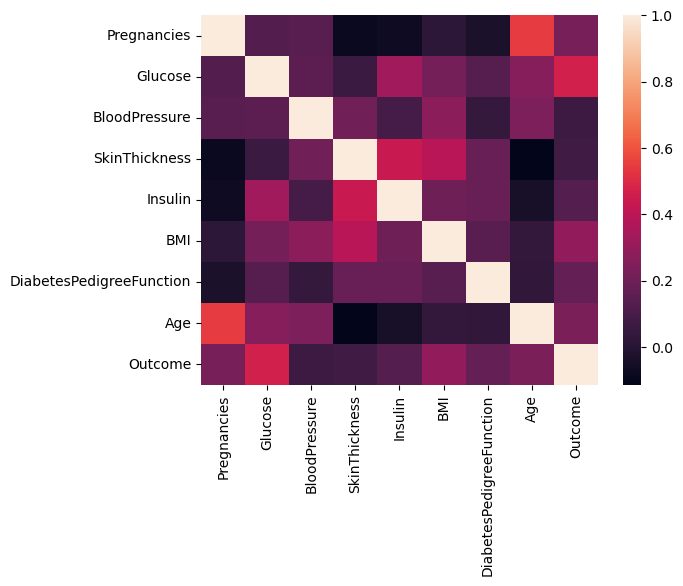
Inferences There is a huge variation among the dataset in the BMI group, even with diabeties.

from sklearn.model\_selection import train\_test\_split  
from sklearn.linear\_model import LogisticRegression  
from sklearn.metrics import accuracy\_score, roc\_auc\_score  
from sklearn.preprocessing import StandardScaler  
  
  
X = df.drop('Outcome', axis=1)  
y = df['Outcome']  
  
# Splitting 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)  
  
# Scaling the data  
scaler = StandardScaler()  
X\_train\_scaled = scaler.fit\_transform(X\_train)  
X\_test\_scaled = scaler.transform(X\_test)  
  
# Training the model with scaled data  
model\_scaled = LogisticRegression(max\_iter=1000)  
model\_scaled.fit(X\_train\_scaled, y\_train)  
  
# Making predictions with the scaled model  
y\_pred\_scaled = model\_scaled.predict(X\_test\_scaled)  
  
# Compute accuracy and AUC-ROC for the scaled model  
print('Accuracy (scaled data):', accuracy\_score(y\_test, y\_pred\_scaled))  
print('AUC-ROC (scaled data):', roc\_auc\_score(y\_test, y\_pred\_scaled))  
  
# Training another model with unscaled data and a different solver  
model\_unscaled = LogisticRegression(solver='liblinear')  
model\_unscaled.fit(X\_train, y\_train)  
  
# Making predictions with the unscaled model  
y\_pred\_unscaled = model\_unscaled.predict(X\_test)  
  
# Compute accuracy and AUC-ROC for the unscaled model  
print('Accuracy (unscaled data):', accuracy\_score(y\_test, y\_pred\_unscaled))  
print('AUC-ROC (unscaled data):', roc\_auc\_score(y\_test, y\_pred\_unscaled))

Accuracy (scaled data): 0.7532467532467533  
AUC-ROC (scaled data): 0.7353535353535354  
Accuracy (unscaled data): 0.7662337662337663  
AUC-ROC (unscaled data): 0.7373737373737373

Running a logistic regression on the dataset to check what is the accuracy and area under curve. As its expected its around 0.75(with scaled and unscaled versions). In this dataset there are some issues, there is no segrregation between type 1 and type 2 diabeties.

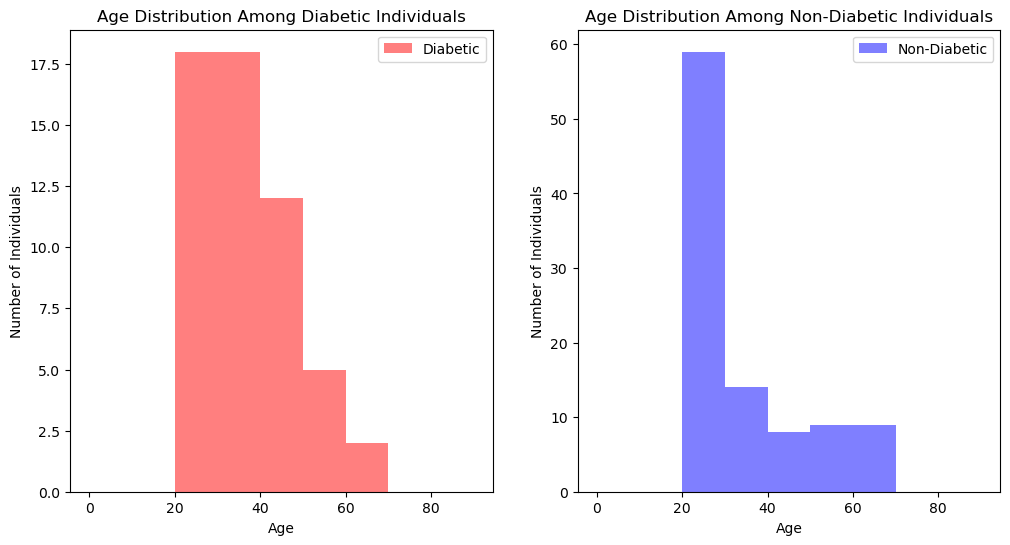
import seaborn as sns   
corr = df.corr()  
print(corr)  
sns.heatmap(corr,   
 xticklabels=corr.columns,   
 yticklabels=corr.columns)

Pregnancies Glucose BloodPressure SkinThickness \  
Pregnancies 1.000000 0.129459 0.141282 -0.081672   
Glucose 0.129459 1.000000 0.152590 0.057328   
BloodPressure 0.141282 0.152590 1.000000 0.207371   
SkinThickness -0.081672 0.057328 0.207371 1.000000   
Insulin -0.073535 0.331357 0.088933 0.436783   
BMI 0.017683 0.221071 0.281805 0.392573   
DiabetesPedigreeFunction -0.033523 0.137337 0.041265 0.183928   
Age 0.544341 0.263514 0.239528 -0.113970   
Outcome 0.221898 0.466581 0.065068 0.074752   
  
 Insulin BMI DiabetesPedigreeFunction \  
Pregnancies -0.073535 0.017683 -0.033523   
Glucose 0.331357 0.221071 0.137337   
BloodPressure 0.088933 0.281805 0.041265   
SkinThickness 0.436783 0.392573 0.183928   
Insulin 1.000000 0.197859 0.185071   
BMI 0.197859 1.000000 0.140647   
DiabetesPedigreeFunction 0.185071 0.140647 1.000000   
Age -0.042163 0.036242 0.033561   
Outcome 0.130548 0.292695 0.173844   
  
 Age Outcome   
Pregnancies 0.544341 0.221898   
Glucose 0.263514 0.466581   
BloodPressure 0.239528 0.065068   
SkinThickness -0.113970 0.074752   
Insulin -0.042163 0.130548   
BMI 0.036242 0.292695   
DiabetesPedigreeFunction 0.033561 0.173844   
Age 1.000000 0.238356   
Outcome 0.238356 1.000000   
  
  
  
  
  


This is a heatmap. To find the correlation between different factors. According to heatmap, features like Pregnancies, Gluecose, BMI, and Age is more correlated with Outcome.

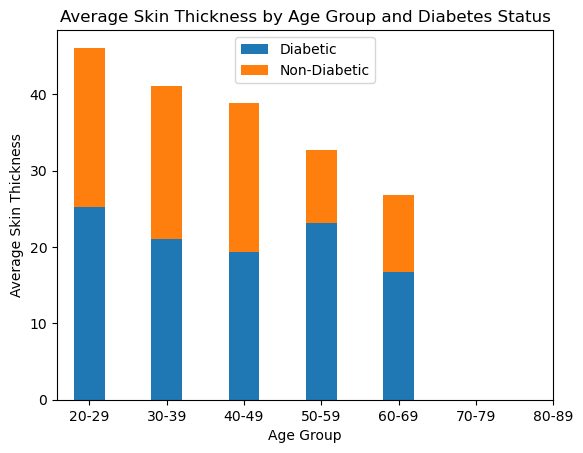
from sklearn.model\_selection import train\_test\_split  
  
# Define the features and the target variable  
X = df.drop(columns=['Outcome']) # Features (all columns except 'Outcome')  
y = df['Outcome'] # Target variable ('Outcome')  
  
# Split the data into training and testing sets  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)  
  
# Now X\_train and y\_train contain the features and target variable for the training set  
# and X\_test and y\_test contain the features and target variable for the testing set

import matplotlib.pyplot as plt  
  
# Split the test data into diabetic and non-diabetic groups  
diabetic\_group = X\_test[y\_test == 1]  
non\_diabetic\_group = X\_test[y\_test == 0]  
  
# Plot the age distribution for the diabetic group  
plt.figure(figsize=(12, 6))  
  
plt.subplot(1, 2, 1)  
plt.hist(diabetic\_group['Age'], bins=range(0, 100, 10), alpha=0.5, color='red', label='Diabetic')  
plt.xlabel('Age')  
plt.ylabel('Number of Individuals')  
plt.title('Age Distribution Among Diabetic Individuals')  
plt.legend()  
  
# Plot the age distribution for the non-diabetic group  
plt.subplot(1, 2, 2)  
plt.hist(non\_diabetic\_group['Age'], bins=range(0, 100, 10), alpha=0.5, color='blue', label='Non-Diabetic')  
plt.xlabel('Age')  
plt.ylabel('Number of Individuals')  
plt.title('Age Distribution Among Non-Diabetic Individuals')  
plt.legend()  
  
# Show the plot  
plt.show()



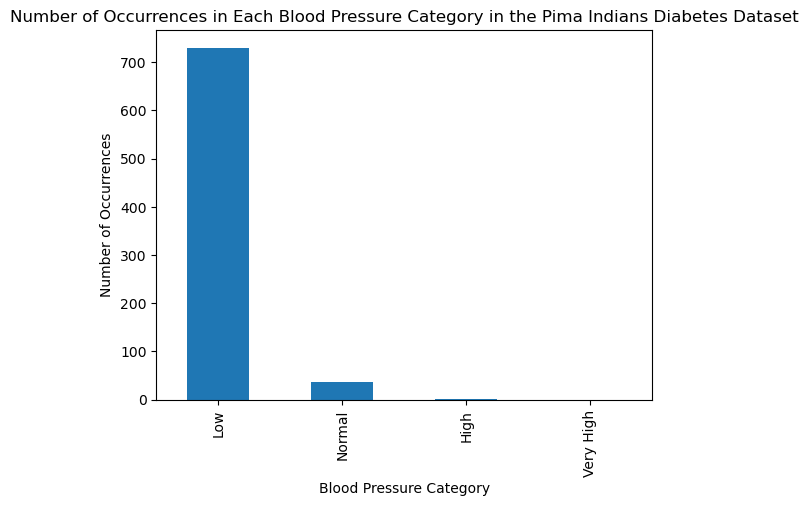
This represents skewness in the data, as it seem most of the people are in 20s who have diabeties in this data set. Which tells most of them might be having type 1 diabeties. Thats why the prediction is off to 0.75

# Define the bins  
bins = [20, 30, 40, 50, 60, 70, 80, 90]  
# Define the labels for the bins  
labels = ['20-29', '30-39', '40-49', '50-59', '60-69', '70-79', '80-89']  
  
# Create a new column 'AgeGroup' by binning the age  
df['AgeGroup'] = pd.cut(df['Age'], bins=bins, labels=labels, include\_lowest=True)  
  
# Separate into diabetic and non-diabetic groups  
diabetic\_group = df[df['Outcome'] == 1]  
non\_diabetic\_group = df[df['Outcome'] == 0]  
  
# Calculate average skin thickness for each age group for diabetic  
diabetic\_avg\_skin\_thickness = diabetic\_group.groupby('AgeGroup')['SkinThickness'].mean()  
  
# Calculate average skin thickness for each age group for non-diabetic  
non\_diabetic\_avg\_skin\_thickness = non\_diabetic\_group.groupby('AgeGroup')['SkinThickness'].mean()  
  
# Plot the data  
x = range(len(labels))  
plt.bar(x, diabetic\_avg\_skin\_thickness, width=0.4, label='Diabetic', align='center')  
plt.bar(x, non\_diabetic\_avg\_skin\_thickness, width=0.4, label='Non-Diabetic', bottom=diabetic\_avg\_skin\_thickness, align='center')  
plt.xlabel('Age Group')  
plt.ylabel('Average Skin Thickness')  
plt.title('Average Skin Thickness by Age Group and Diabetes Status')  
plt.xticks(x, labels)  
plt.legend()  
plt.show()



Skin thickness reduces with age but it get accelarated due to diabeties.

import pandas as pd  
import matplotlib.pyplot as plt  
  
  
  
# Define the bins for age groups  
age\_bins = [20, 30, 40, 50, 60, 70, 80, 90]  
# Define the labels for the age bins  
age\_labels = ['20-29', '30-39', '40-49', '50-59', '60-69', '70-79', '80-89']  
  
# Define the bins for blood pressure categories  
bp\_bins = [0, 90, 120, 140, float('inf')]  
# Define the labels for the blood pressure bins  
bp\_labels = ['Low', 'Normal', 'High', 'Very High']  
  
# Create a new column 'AgeGroup' by binning the age  
df['AgeGroup'] = pd.cut(df['Age'], bins=age\_bins, labels=age\_labels, include\_lowest=True)  
  
# Create a new column 'BloodPressureCategory' by binning the blood pressure  
df['BloodPressureCategory'] = pd.cut(df['BloodPressure'], bins=bp\_bins, labels=bp\_labels, include\_lowest=True)  
  
# Count the number of occurrences in each blood pressure category  
bp\_counts = df['BloodPressureCategory'].value\_counts().sort\_index()  
  
# Plot the data  
bp\_counts.plot(kind='bar')  
plt.xlabel('Blood Pressure Category')  
plt.ylabel('Number of Occurrences')  
plt.title('Number of Occurrences in Each Blood Pressure Category in the Pima Indians Diabetes Dataset')  
plt.show()



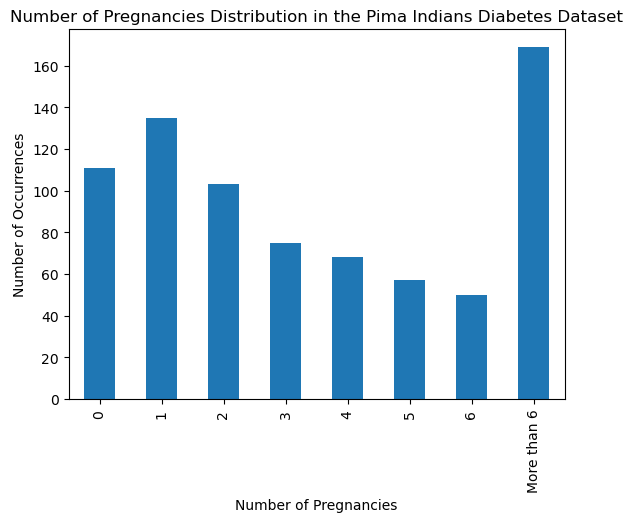
This is also not normal at all, Large percentage of population has low BP. Which would skew data to predict low BP as normal. This needs more understanding why its low in the population.

# Function to categorize the number of pregnancies  
def categorize\_pregnancies(pregnancies):  
 if pregnancies == 0:  
 return '0'  
 elif pregnancies == 1:  
 return '1'  
 elif pregnancies == 2:  
 return '2'  
 elif pregnancies == 3:  
 return '3'  
 elif pregnancies == 4:  
 return '4'  
 elif pregnancies == 5:  
 return '5'  
 elif pregnancies == 6:  
 return '6'  
 else:  
 return 'More than 6'  
  
# Create a new column 'PregnancyCategory' by categorizing the number of pregnancies  
df['PregnancyCategory'] = df['Pregnancies'].apply(categorize\_pregnancies)  
  
# Count the number of occurrences in each pregnancy category  
pregnancy\_counts = df['PregnancyCategory'].value\_counts().sort\_index()  
  
# Plot the data  
pregnancy\_counts.plot(kind='bar')  
plt.xlabel('Number of Pregnancies')  
plt.ylabel('Number of Occurrences')  
plt.title('Number of Pregnancies Distribution in the Pima Indians Diabetes Dataset')  
plt.show()

png

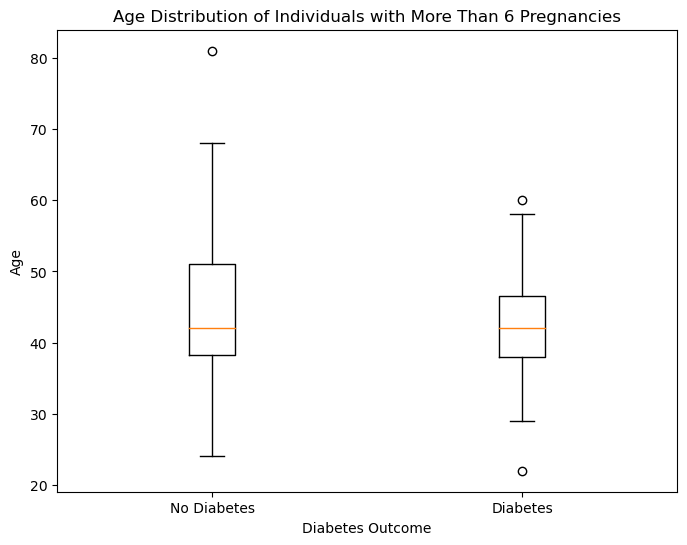
Distribution of pregnancies in the dataset. Maximum pregnancy is 17. Mean is around 3.

df['PregnancyCategory'] = df['Pregnancies'].apply(categorize\_pregnancies)  
  
# Pivot table with PregnancyCategory and Outcome (diabetes vs non-diabetes)  
pivot\_table = pd.pivot\_table(df, values='Age', index='PregnancyCategory', columns='Outcome', aggfunc='count', fill\_value=0)  
  
# Plot the data  
pivot\_table.plot(kind='bar', stacked=False)  
plt.xlabel('Number of Pregnancies')  
plt.ylabel('Number of Occurrences')  
plt.title('Number of Pregnancies Distribution in the Pima Indians Diabetes Dataset')  
plt.legend(['No Diabetes', 'Diabetes'])  
plt.show()



As per the observation in this chart, diabeties is more in the group more than 6, so I wanted to check is it because of the age parameter.

more\_than\_6\_pregnancies = df[df['Pregnancies'] > 6]  
  
# Plot the data using a box plot  
plt.figure(figsize=(8, 6))  
plt.boxplot([more\_than\_6\_pregnancies[more\_than\_6\_pregnancies['Outcome'] == 0]['Age'],  
 more\_than\_6\_pregnancies[more\_than\_6\_pregnancies['Outcome'] == 1]['Age']],  
 labels=['No Diabetes', 'Diabetes'])  
plt.xlabel('Diabetes Outcome')  
plt.ylabel('Age')  
plt.title('Age Distribution of Individuals with More Than 6 Pregnancies')  
plt.show()



Average age for both dataset is similar(around 45) and age plays a huge factor in type 2 diabeties so this is inconclusive regarding pregancy is cause for diabeties.

**Findings**

Age Distribution Among Diabetic and Non-Diabetic Individuals: The analysis of age distribution among diabetic individuals revealed that a significant proportion of the individuals in the dataset are in their 20s. This suggests that many of these individuals might have Type 1 diabetes, which usually develops at a young age. This is an unexpected finding as Type 2 diabetes is more common in the general population. This skewness in age distribution might be impacting the predictive accuracy of our models.

Skin Thickness by Age and Diabetes Status: The average skin thickness was calculated for each age group, separated by diabetes status. The results showed a decrease in skin thickness with age, with a more pronounced decrease for individuals with diabetes. This finding supports existing research which indicates that skin thickness can be impacted by diabetes.

Blood Pressure Categories: A large proportion of individuals in the dataset were found to have low blood pressure. This was unexpected, as high blood pressure is a common condition in individuals with diabetes. This unusual distribution might skew the data and impact the accuracy of our predictive models. Further research is needed to understand why blood pressure is low in this population.

Number of Pregnancies: The distribution of the number of pregnancies was explored. The maximum number of pregnancies recorded was 17, with a mean of around 3. The prevalence of diabetes increased among individuals who had more than 6 pregnancies. However, the average age of individuals with more than 6 pregnancies was similar for both diabetic and non-diabetic groups. Given that age is a significant factor for Type 2 diabetes, it was inconclusive whether pregnancy is a cause for diabetes.

**Conclusion**

This analysis has revealed several intriguing findings, such as the large proportion of individuals in their 20s with diabetes and the unexpected low blood pressure among the population. While these findings are insightful, further research is needed to understand their implications fully.

In conclusion, analyzing the Pima Indians Diabetes Dataset has provided valuable insights that could guide further research and potential healthcare interventions. However, more work is needed to fully understand the complex interplay of factors that contribute to the onset and management of diabetes in this population.