"Diabetes Patients Prediction Insights"

Diabetes

Diabetes is a chronic disease characterized by elevated levels of blood glucose resulting from the body's inability to properly produce or use insulin, a hormone responsible for regulating blood sugar levels.

There are primarily three types of diabetes:
- **Type 1 diabetes**, which is typically diagnosed in childhood
- **Type 2 diabetes**, which is more common and usually develops in adulthood.
- **Gestational diabetes**, temporary form of diabetes that occurs during pregnancy.

Project Summary

- Dataset source: Insights from the National Institute of Diabetes and Digestive and Kidney Diseases Dataset (.csv)
- Objective: To develop a diagnostic prediction model for identifying diabetes in patients using specific diagnostic measurements.
- Target audience: Female patients of Pima Indian heritage, aged at least 21 years old.
- Variables: Medical predictor variables (independent) and "Outcome" (dependent)

Insights from the Dataset

- Being **overweight** or **obese** and being over the **age** of **30** are significant risk factors for diabetes. Maintaining a healthy weight and regular checkups can reduce the risk.
- The **number of pregnancies** a woman experiences correlates with the likelihood of diabetes. Regular blood sugar monitoring is crucial, especially with increasing pregnancies.
- Diabetes Pedigree Function (DPF) assesses diabetes risk based on age and **family history**. Higher DPF values indicate a greater likelihood of diabetes development.
- There's a strong relationship between **glucose levels and diabetes**. Levels above 140 indicate, the person is considered to have **impaired glucose tolerance**, Glucose is a major diabetes risk factor. Factors influencing high glucose include stress, diet, and certain medications.
- Increased skin thickness often indicates **insulin resistance**. Excess insulin in the body leads to fat storage, causing thickening.

ADAMYA AGGARWAL
(Data Analyst)

"Let's Dive into the Code: Exploring Diabetes Dataset through Analysis"

"Diabetes Patients Prediction : Diagnostic Measurements and Outcomes"

Objective of this Project :-

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether a patient has diabetes, based on certain diagnostic measurements included in the dataset.

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

"Import Diabetes Dataset from CSV File"

In [2]:
```
data= pd.read_csv("diabetes.csv")
data
```

Out[2]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction
0	6	148	72	35	0	33.6	0.627
1	1	85	66	29	0	26.6	0.351
2	8	183	64	0	0	23.3	0.672
3	1	89	66	23	94	28.1	0.167
4	0	137	40	35	168	43.1	2.288
...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171
764	2	122	70	27	0	36.8	0.340
765	5	121	72	23	112	26.2	0.245
766	1	126	60	0	0	30.1	0.349
767	1	93	70	31	0	30.4	0.315

768 rows × 9 columns

"Dataset Information Summary"

In [3]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype  
---                            ──────────────  ─────
 0   Pregnancies               768 non-null    int64  
 1   Glucose                   768 non-null    int64  
 2   BloodPressure             768 non-null    int64  
 3   SkinThickness             768 non-null    int64  
 4   Insulin                   768 non-null    int64  
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64  
 8   Outcome                   768 non-null    int64  
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

In [4]:
```
data.shape
```

Out[4]:
```
(768, 9)
```

"Attributes in Diabetes Dataset: Understanding the Meaning and Significance"

- Pregnancies: Represents the number of times the patient has been pregnant.
- Glucose: Represents the patient fasting blood sugar level (measured in mg/dL). High glucose levels can be indicative of diabetes.
- BloodPressure: Represents the patient blood pressure (measured in mm Hg). High blood pressure can be a risk factor for diabetes.
- SkinThickness: Represents the thickness of the patient skinfold (measured in mm) at a certain location on the body.
- Insulin: Represents the patient insulin level (measured in µU/ml). Abnormal insulin levels can be associated with diabetes.
- BMI: Represents the patient Body Mass Index, measure of body fat based on height and weight. High BMI is often associated with an increased risk of diabetes.
- DiabetesPedigreeFunction: Represents the genetic risk of diabetes based on family history.
- Age: Represents the age of the patient in years. The risk of diabetes often increases with age.
- Outcome: This is the target column. It indicates whether a patient has diabetes or not, where '1' indicate the presence of diabetes, and '0' indicate the absence of diabetes.

In [5]:
```
data.head()
```

Out[5]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	A
0	6	148	72	35	0	33.6	0.627	
1	1	85	66	29	0	26.6	0.351	
2	8	183	64	0	0	23.3	0.672	
3	1	89	66	23	94	28.1	0.167	
4	0	137	40	35	168	43.1	2.288	

"Count Duplicate Rows for Each Column in the DataFrame"

In [6]:
```
for i in data.columns:
    print(i,":", data.duplicated().sum())
```

```
Pregnancies : 0
Glucose : 0
BloodPressure : 0
SkinThickness : 0
Insulin : 0
BMI : 0
DiabetesPedigreeFunction : 0
Age : 0
Outcome : 0
```

"Statistics Summary of the Dataset"

In [7]:
```
data.describe()
```

Out[7]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPe
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	

"Number of Unique Values in Each Column of the Dataset"

In [8]:
```
data.isnull().mean()
```

Out[8]:
```
Pregnancies                 0.0
Glucose                     0.0
BloodPressure               0.0
SkinThickness               0.0
Insulin                     0.0
BMI                         0.0
DiabetesPedigreeFunction    0.0
Age                         0.0
Outcome                     0.0
dtype: float64
```

In [9]:
```
data.nunique()
```

Out[9]:
```
Pregnancies                 17
Glucose                    136
BloodPressure               47
SkinThickness               51
Insulin                    186
BMI                        248
DiabetesPedigreeFunction   517
Age                         52
Outcome                      2
dtype: int64
```

"Count of Patients with High Glucose Levels (>140 mg/dL)"

- The plasma glucose level <140 mg/dL is considered normal

In [10]:
```
high_glucode= data["Glucose"][data["Glucose"]>140]
high_glucode.count()
```

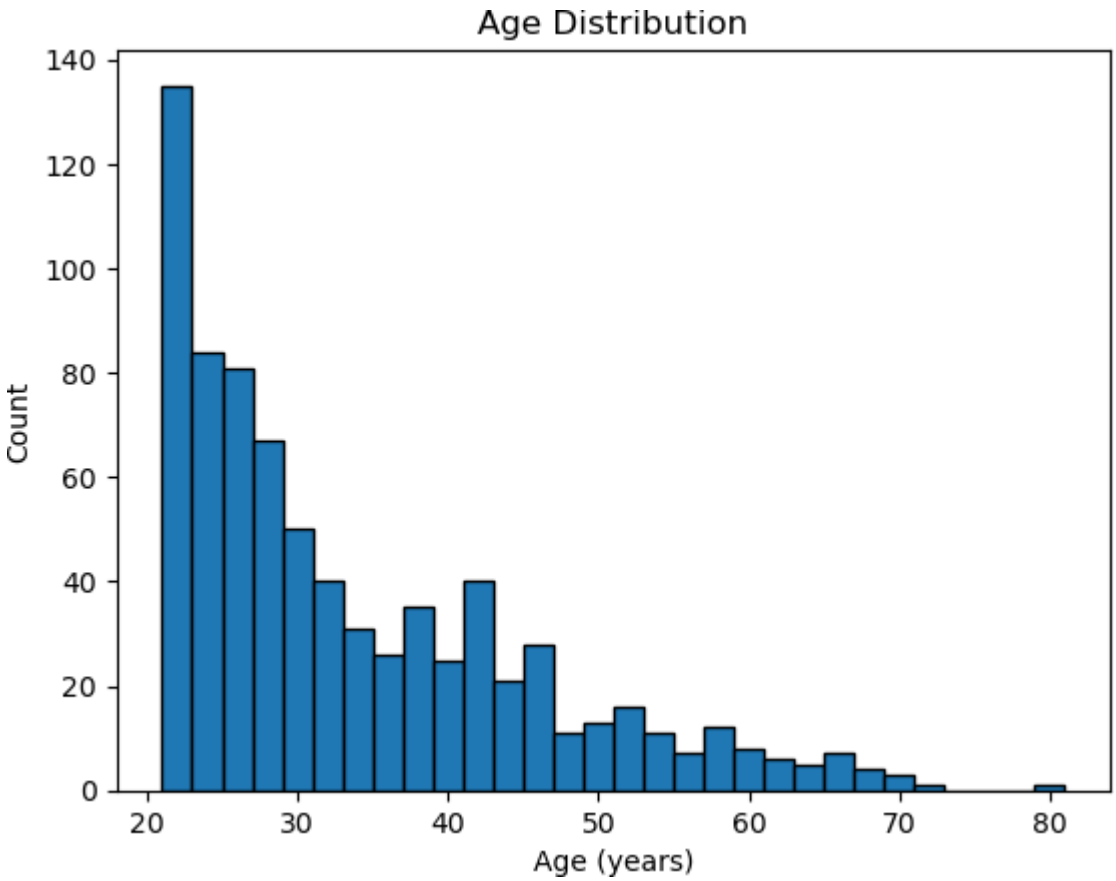
Out[10]:
```
192
```

"Univariate Analysis"

"Age Distribution: Visualizing the Spread of Ages in the Dataset using a Histogram"

- Insights: Age is a significant factor in diabetes risk, as it often increases with age. This histogram can help identify age groups with a higher occurrence of diabetes.

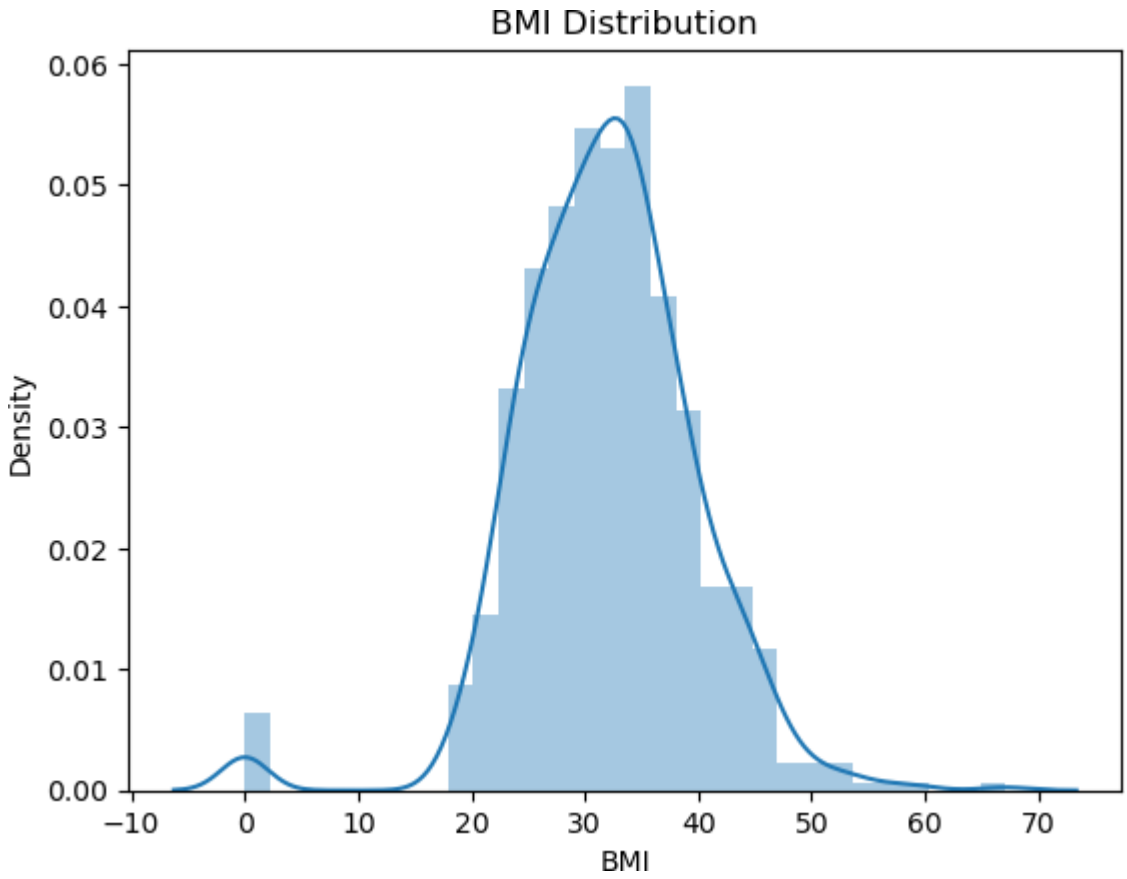
In [11]:
```
plt.hist(data['Age'], bins=30, edgecolor='black')
plt.title('Age Distribution')
plt.xlabel('Age (years)')
plt.ylabel('Count')
plt.show()
```



"BMI Distribution: Exploring Body Mass Index Distribution"

- Insights: This can help in understanding the occurrence of different body mass index categories (e.g., underweight, normal weight, overweight, obese) among the individuals.

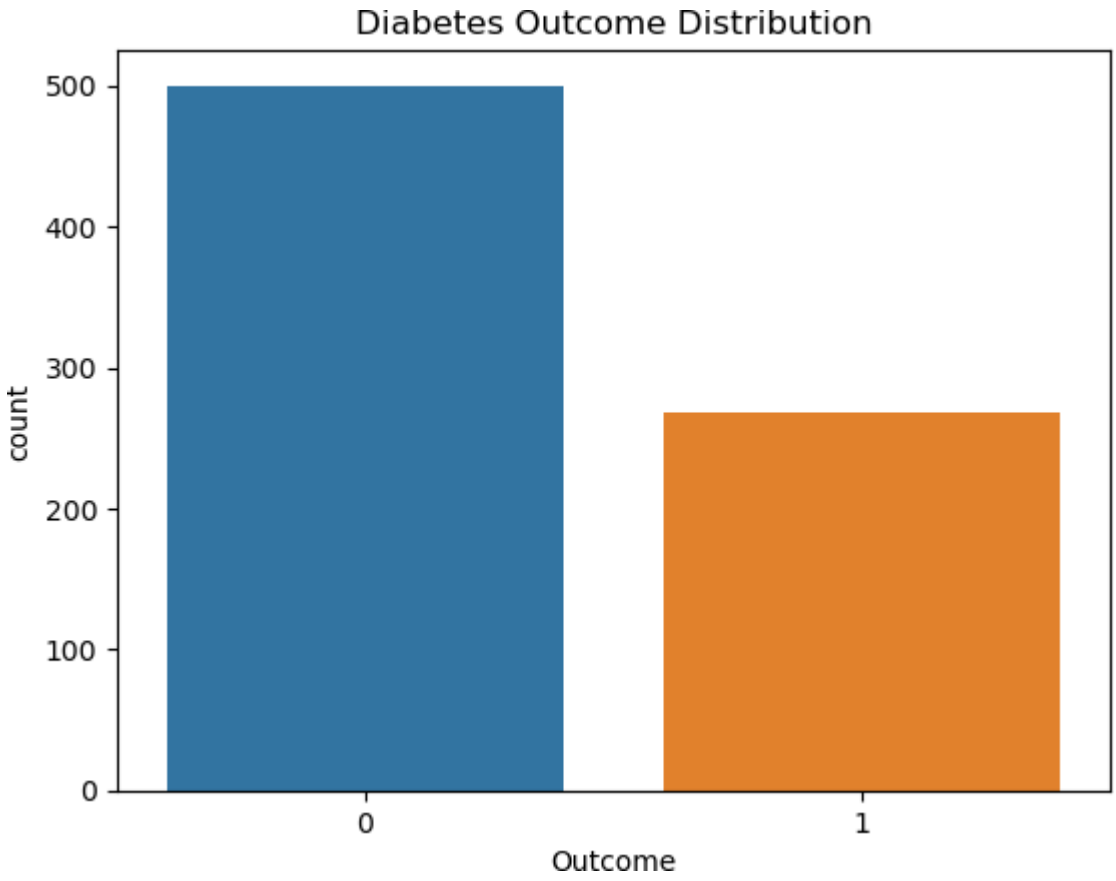
In [12]:
```
sns.distplot(data['BMI'], bins=30)
plt.title('BMI Distribution')
plt.show()
```



"Diabetes Outcome Distribution: Count of Patients with and without Diabetes"

- Insights: It provides essential context for predictive modeling efforts, helping us gauge the dataset's balance and identify potential data quality issues.

In [13]:
```
bar= sns.countplot(x='Outcome', data=data)
plt.title('Diabetes Outcome Distribution')
plt.show()
```



In [14]:
```
data["Outcome"].value_counts()
```

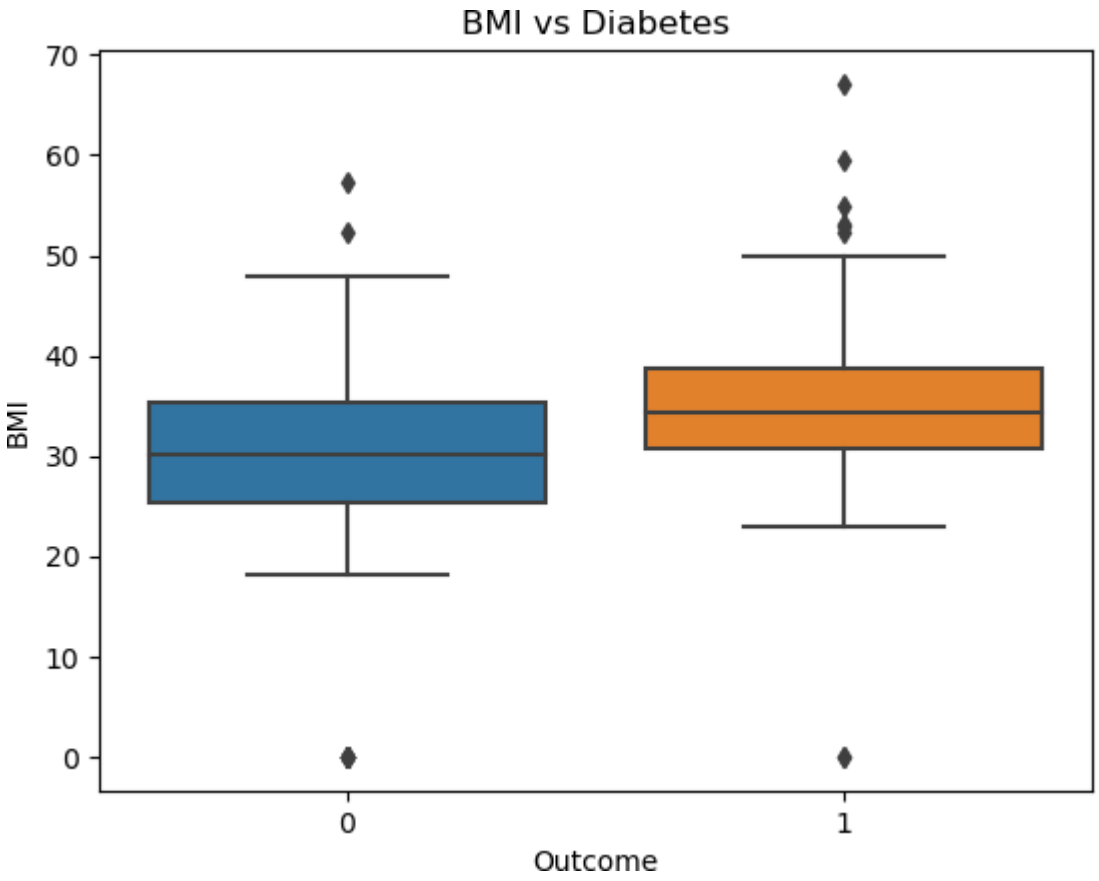
Out[14]:
```
0    500
1    268
Name: Outcome, dtype: int64
```

"Bivariate Analysis"

"BMI vs Outcome: Exploring BMI Differences with a Box Plot"

- Insights: BMI is a differentiating factor between patients with and without diabetes. It provides visual insights into the distribution and characteristics of BMI values within each group and helps us decide whether BMI should be included as a significant feature in our predictive model for diabetes classification.

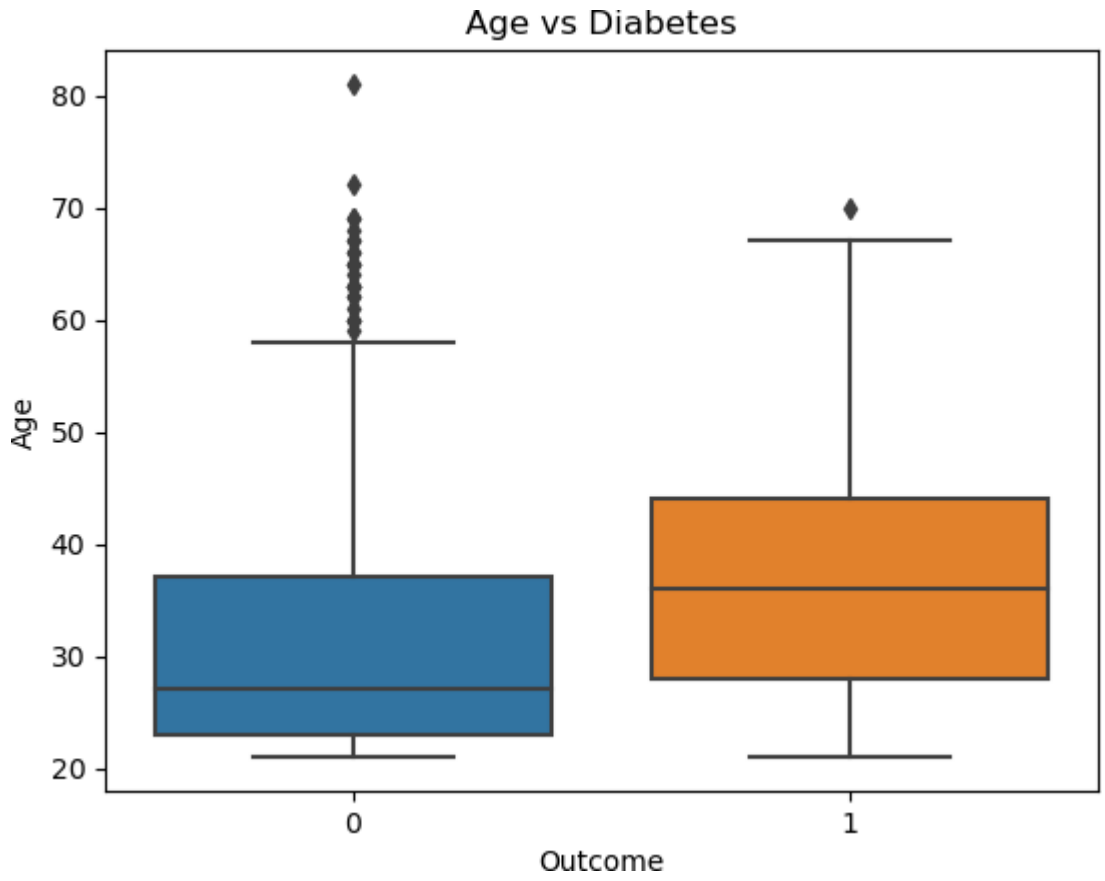
In [15]:
```
sns.boxplot(x='Outcome', y='BMI', data=data)
plt.title('BMI vs Diabetes')
plt.show()
```



"Age vs Outcome: Diabetes Outcomes w.r.t. Age Using a Box Plot"

- Insights: It can help us assess whether there are any age-related patterns or trends that could aid in our classification task. This visualization is an important step in the data exploration process when working on a classification problem like predicting diabetes.

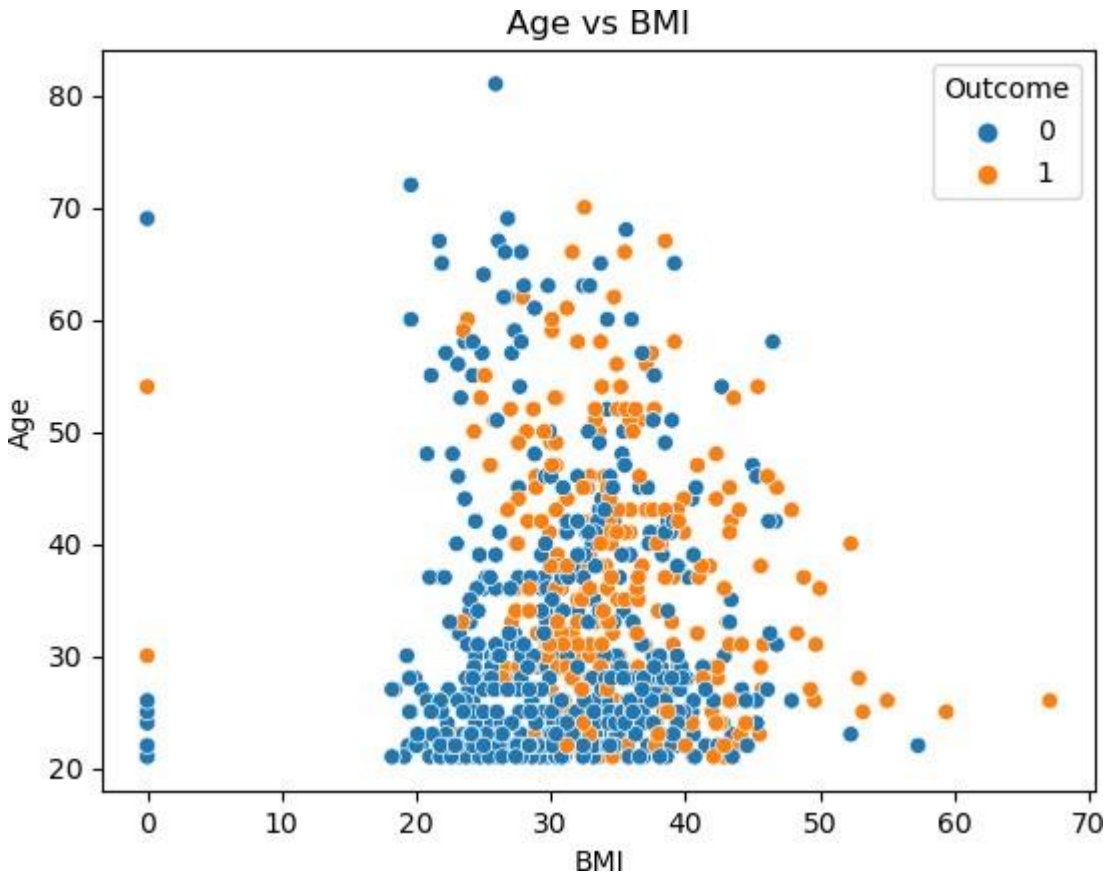
In [16]:
```
sns.boxplot(x='Outcome', y="Age", data=data)
plt.title("Age vs Diabetes")
plt.show()
```



"Age vs BMI: Diabetes Outcomes w.r.t. Age and BMI Using a Scatter Plot"

- Insights: It can provide insights into the distribution, trends, and potential relationships between these two features. These insights can help us to detect outliers and enhance our understanding of how age and BMI contribute to the classification of diabetes risk.

In [17]:
```
sns.scatterplot(x='BMI', y='Age', hue='Outcome', data=data)
plt.title('Age vs BMI')
plt.show()
```



"Pairplot: Exploring Relationships Between Variables with Outcome as the Highlight"

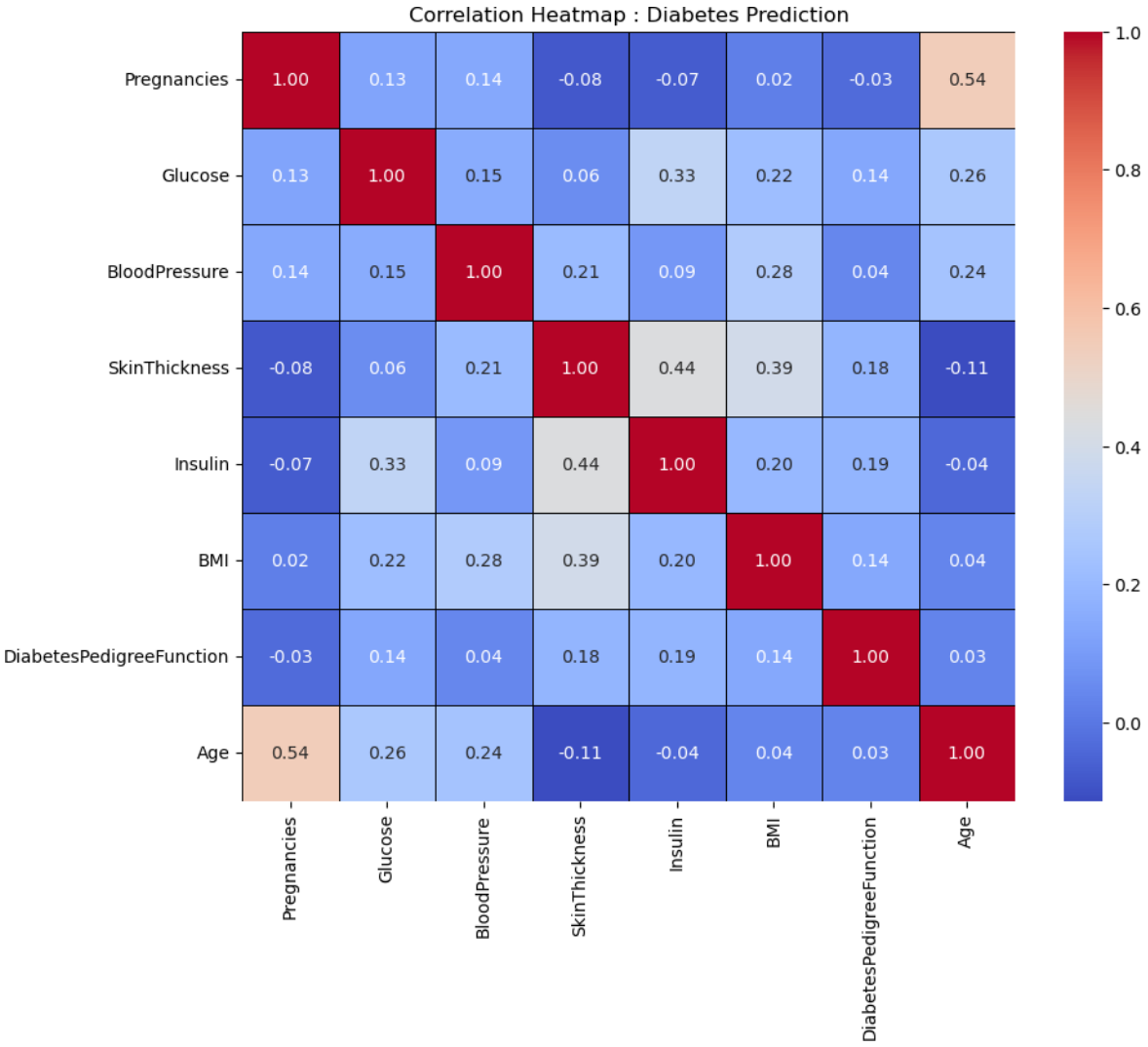
- Insights: Defines how numerical characteristics relate to the presence or absence of diabetes. It can guide feature selection, help us identifing potential patterns, and inform the design of predictive models for diabetes diagnosis.

In [18]:
```
sns.pairplot(data, hue='Outcome')
plt.show()
```

"Correlation Heatmap: Exploring Relationships Between Variables for Diabetes Prediction"

- Correlation coefficients range from -1 to 1, where: -1 indicates a perfect negative correlation. 1 indicates a perfect positive correlation. 0 indicates no linear correlation.
- Numeric values provide the exact correlation coefficient.
- Darker colors suggest stronger correlations. The color (blue or orange) indicates the direction of the correlation.

In [19]:
```
new_data= data.drop('Outcome', axis = 1)
corr_matrix = new_data.corr()
plt.figure(figsize=(10,8))
sns.heatmap(corr_matrix, annot=True, fmt=".2f", linewidths=0.5,cmap="coolwarm",lin
plt.title("Correlation Heatmap : Diabetes Prediction")
plt.show()
```



"Skewness : Indicates whether the data is symmetrically distributed or skewed to one side"

In [20]:
```
data.skew()
```

Out[20]:
```
Pregnancies                0.901674
Glucose                    0.173754
BloodPressure             -1.843608
SkinThickness              0.109372
Insulin                    2.272251
BMI                       -0.428982
DiabetesPedigreeFunction   1.919911
Age                        1.129597
Outcome                    0.635017
dtype: float64
```

"Observations Regarding Skewed Data"

1. Insulin , DPF and Age are highly right skewed and having heavy amount of outliers .
2. Age and Pregnancies are also right skewed with some extreme values .
3. Outcome variable is highly imbalanced (1:0) ##### "Attributes in the Diabetes Dataset: Understanding the Meaning of Positive and Negative Skewness"

- Pregnancies: Means that there are likely more individuals with fewer pregnancies, and there may be some outliers with a higher number of pregnancies.
- Glucose: Indicates that most individuals may have normal or lower glucose levels, with some outliers having higher values.

- BloodPressure: Indicates that most individuals likely have higher blood pressure levels, with some outliers having lower values.
- SkinThickness: Most individuals may have typical skin thickness, with some outliers having thicker skin.
- Insulin: Most individuals may have low insulin levels, while a few outliers may have very high insulin levels.
- BMI: Most individuals may have normal or lower BMIs, with some outliers having higher values.
- DiabetesPedigreeFunction: Most individuals may have low values, while a few outliers may have very high values indicating a strong family history of diabetes.
- Age: Most individuals may be younger, with some outliers being older.
- Outcome: Suggests that there may be more individuals without diabetes (Outcome=0) than with diabetes (Outcome=1) in the dataset.

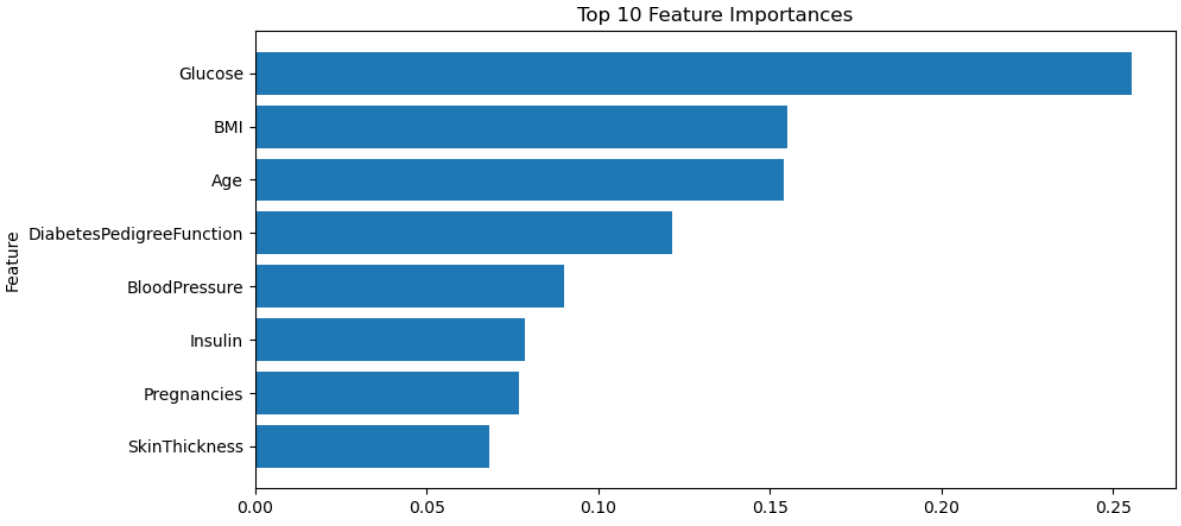
"Plot the Bar Graph to Visualize the Top 10 Most Important Features in the Model"

In [21]:
```
from sklearn.ensemble import RandomForestClassifier
X = data.drop('Outcome', axis = 1)
y = data['Outcome']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_s
model = RandomForestClassifier()
model.fit(X_train, y_train)
```

Out[21]:
```
▾ RandomForestClassifier
RandomForestClassifier()
```

In [22]:
```
top_features = pd.Series(model.feature_importances_, index=X.columns).nlargest(10)

# Plot the top 10 features:
plt.figure(figsize=(10, 5))
plt.barh(top_features.index, top_features.values)
plt.title("Top 10 Feature Importances")
plt.ylabel("Feature")
plt.gca().invert_yaxis()
plt.show()
```



"Logistic Regression Model : Train-Test Split"

In [23]:
```
X = data.drop('Outcome', axis = 1)
y = data['Outcome']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_s
```

In [24]:
```
from sklearn.preprocessing import StandardScaler
Scaler = StandardScaler()
Scaler.fit(X)
X = Scaler.transform(X)
```

In [25]:
```
model = LogisticRegression()
model.fit(X_train, y_train)
print(X.shape, X_train.shape, X_test.shape)
```

```
(768, 8) (614, 8) (154, 8)
```

In [26]:
```
prediction = model.predict(X_train)
accuracy = accuracy_score(y_train, prediction)
print(f"Accuracy on train data: {accuracy*100:.2f}%")
```

```
Accuracy on train data: 77.36%
```

In [27]:
```
y_pred = model.predict(X_test)
accuracy= accuracy_score(y_test,y_pred)
print(f"\nAccuracy on test data: {accuracy*100:.2f}%")
```

```
Accuracy on test data: 74.68%
```

The model is 75% accurate in predicting whether the patient has diabetes or not

---------- **Thank You** ----------
