In [50]:
```
# importing all the libraraies
import numpy as np
import pandas as pd
from sklearn.preprocessing import  StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
```

importing dataset

In [46]:
```
df=pd.read_csv("C:/Users/user/Downloads/archive (8).zip")
df
```

Out[46]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunctic
0	6	148	72	35	0	33.6	0.62
1	1	85	66	29	0	26.6	0.35
2	8	183	64	0	0	23.3	0.67
3	1	89	66	23	94	28.1	0.16
4	0	137	40	35	168	43.1	2.28
...	...	...	...	...	...	...	
763	10	101	76	48	180	32.9	0.17
764	2	122	70	27	0	36.8	0.34
765	5	121	72	23	112	26.2	0.24
766	1	126	60	0	0	30.1	0.34
767	1	93	70	31	0	30.4	0.31

768 rows × 9 columns

In [ ]:
```
1. Pregnancies:
Description: The number of times the patient has been pregnant.
Significance: This feature might provide insights into how pregnancy influence
especially gestational diabetes. A higher number of pregnancies may correlate
2. Glucose:
Description: The plasma glucose concentration (mg/dL) after a 2-hour oral gluc
Significance: This is a critical measure for diabetes, as elevated glucose lev
impaired glucose metabolism, which is a hallmark of diabetes.
3. BloodPressure:
Description: The diastolic blood pressure (mmHg).
Significance: High blood pressure (hypertension) can be a risk factor for diab
making this an important feature.
4. SkinThickness:
Description: The thickness of the skinfold (mm) at the triceps.
Significance: Skinfold thickness is related to body fat. Increased body fat ca
which is linked to the development of Type 2 diabetes.
5. Insulin:
Description: The insulin level (mu U/ml) in the blood.
Significance: Insulin is the hormone that regulates blood sugar. High insulin
which is a common factor in the development of diabetes.
6. BMI (Body Mass Index):
Description: The body mass index is a measure of body fat based on weight and
Significance: Higher BMI values generally indicate obesity, which is a signifi
Obesity is associated with increased insulin resistance.
7. DiabetesPedigreeFunction:
Description: A function that represents the genetic history of diabetes (a fam
Significance: This feature captures genetic predisposition. A higher value ind
which increases the likelihood of developing the disease.
8. Age:
Description: The age of the patient in years.
Significance: Age is an important factor, as the risk of developing Type 2 dia
especially if they have other risk factors like obesity or a sedentary lifesty
9. Outcome:
Description: A binary variable indicating whether the patient has diabetes (1)
Significance: This is the target variable in the dataset. It indicates whether
This is what you're trying to predict in a classification task.
```

Exploratory Analysis

In [3]:
```
df.head()
```

Out[3]:

	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

In [4]:
```
df.tail()
```

Out[4]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunctic
763	10	101	76	48	180	32.9	0.1
764	2	122	70	27	0	36.8	0.3
765	5	121	72	23	112	26.2	0.2
766	1	126	60	0	0	30.1	0.3
767	1	93	70	31	0	30.4	0.3

In [5]:
```
df.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 [6]:
```
df.shape
```

Out[6]:
```
(768, 9)
```

In [7]:
```
df.columns
```

Out[7]:
```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

In [9]:
```
df.describe()
```

Out[9]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Diabetes
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	

In [10]:
```
df["Outcome"].value_counts()
```

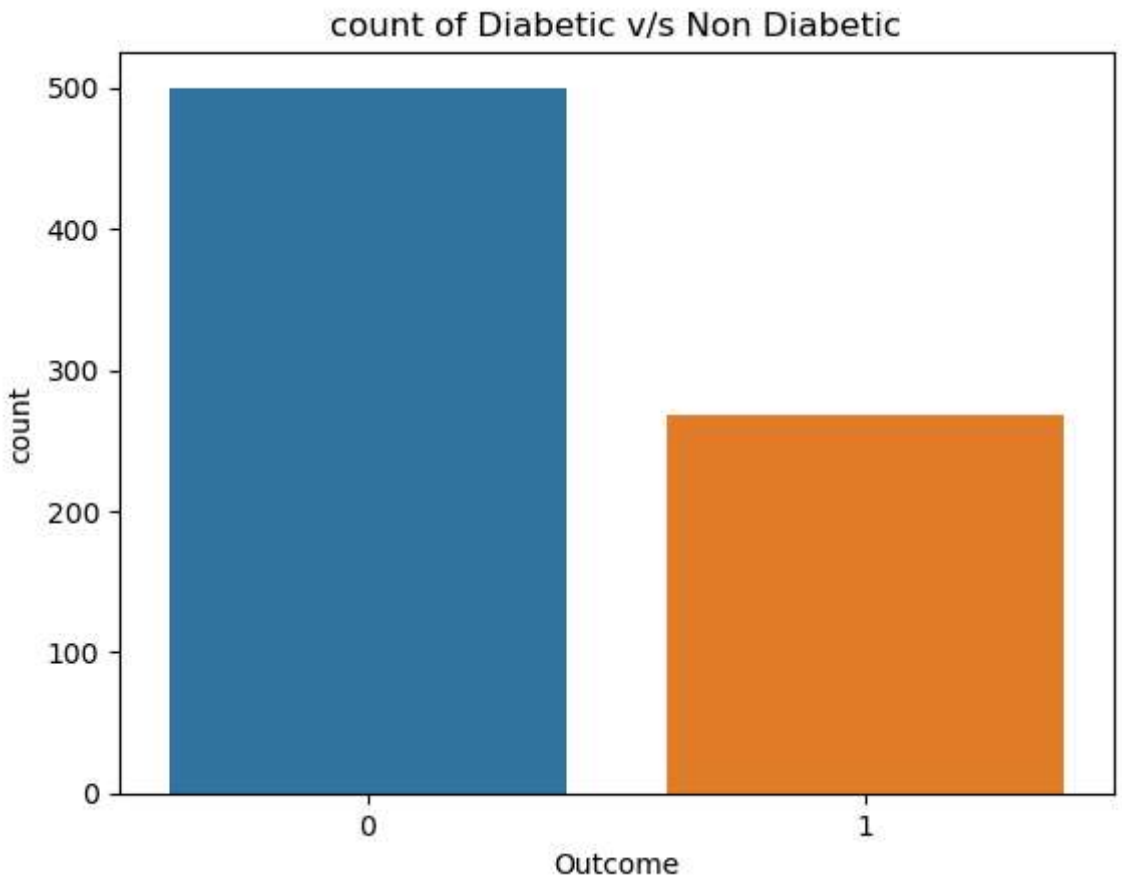
Out[10]:
```
Outcome
0    500
1    268
Name: count, dtype: int64
```

In [12]:
```
df.groupby("Outcome").mean()
```

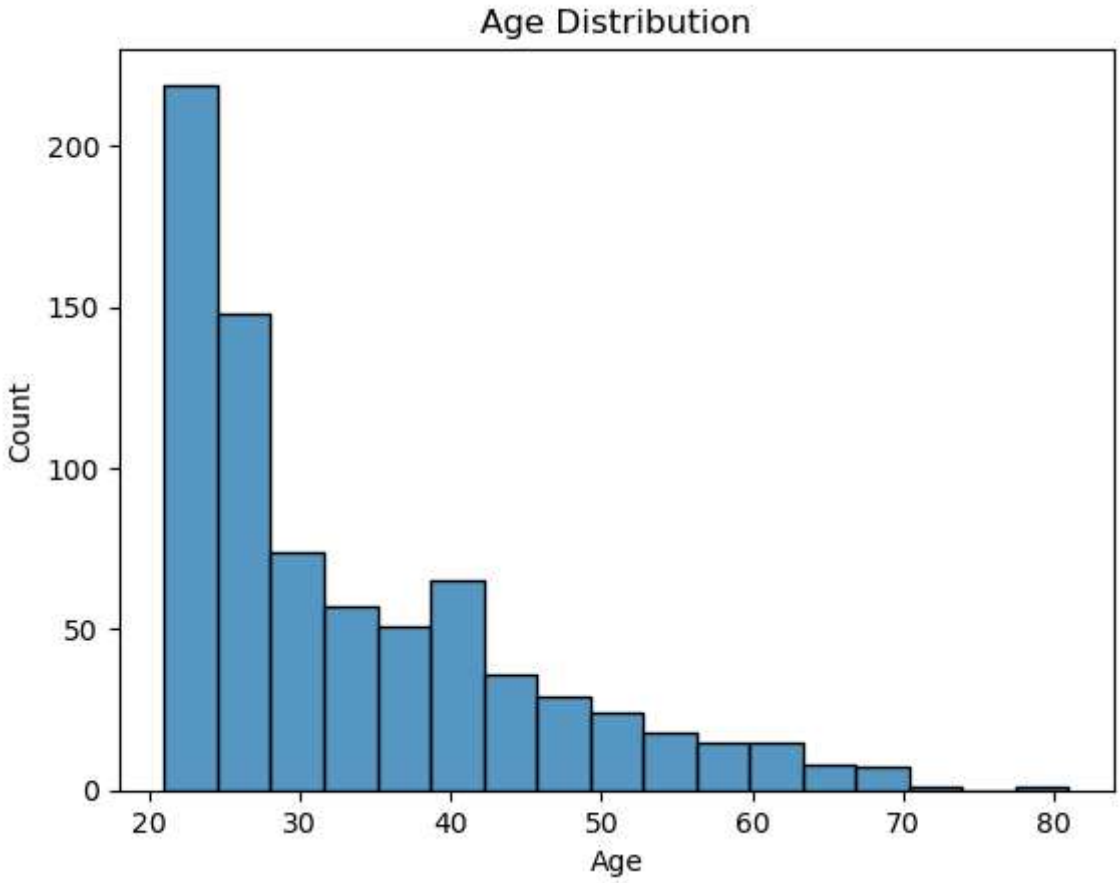
Out[12]:

Outcome	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Diabet
0	3.298000	109.980000	68.184000	19.664000	68.792000	30.304200	
1	4.865672	141.257463	70.824627	22.164179	100.335821	35.142537	

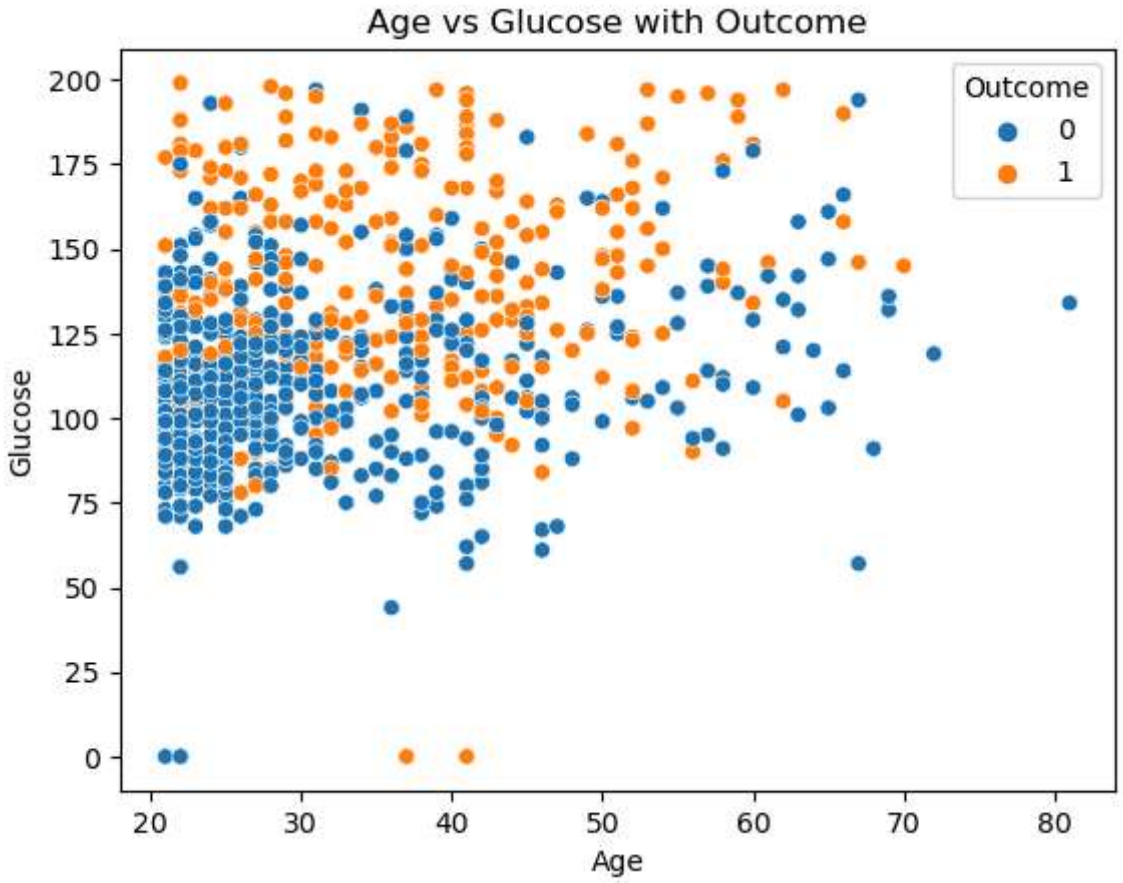
In [54]:
```
sns.countplot(x="Outcome",data=df)
plt.title("count of Diabetic v/s Non Diabetic")
plt.show()
```



In [53]:
```
sns.histplot(x="Age",data=df)
plt.title("Age Distribution")
plt.show()
```



In [55]:
```
sns.scatterplot(x='Age', y='Glucose', hue='Outcome', data=df)
plt.title('Age vs Glucose with Outcome')
plt.show()
```



In [56]:
```
sns.pairplot(data=df,hue="Outcome")
plt.show()
```

```
C:\Users\user\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarnin
g: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

In [4]:
```
df.rename(columns={"Pregnancies": "preganancy",
                   "Glucose":"glucose",
                   "BloodPressure":"bp",
                    "SkinThickness":"st",
                   "Insulin":"insulin",
                   "BMI":"bmi",
                   "DiabetesPedigreeFunction":"history",
                   "Age":"age",
                    "Outcome":"result"},inplace=True)
```

In [5]:
```
df
```

Out[5]:

	preganancy	glucose	bp	st	insulin	bmi	history	age	result
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows × 9 columns

In [6]:
```
x=df.drop(["result"],axis=1)
```

In [7]:
```
y=df["result"]
```

In [38]:
```
x
```

Out[38]:

	preganancy	glucose	bp	st	insulin	bmi	history	age
0	6	148	72	35	0	33.6	0.627	50
1	1	85	66	29	0	26.6	0.351	31
2	8	183	64	0	0	23.3	0.672	32
3	1	89	66	23	94	28.1	0.167	21
4	0	137	40	35	168	43.1	2.288	33
...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63
764	2	122	70	27	0	36.8	0.340	27
765	5	121	72	23	112	26.2	0.245	30
766	1	126	60	0	0	30.1	0.349	47
767	1	93	70	31	0	30.4	0.315	23

768 rows × 8 columns

In [39]:
```
y
```

Out[39]:
```
0      1
1      0
2      1
3      0
4      1
      ..
763    0
764    0
765    0
766    1
767    0
Name: result, Length: 768, dtype: int64
```

In [ ]:
```
#Scaling the dataset
```

In [8]:
```
scaler=StandardScaler()
```

In [9]:
```
scaler.fit_transform(x)
```

Out[9]:
```
array([[ 0.63994726,  0.84832379,  0.14964075, ...,  0.20401277,
         0.46849198,  1.4259954 ],
       [-0.84488505, -1.12339636, -0.16054575, ..., -0.68442195,
        -0.36506078, -0.19067191],
       [ 1.23388019,  1.94372388, -0.26394125, ..., -1.10325546,
         0.60439732, -0.10558415],
       ...,
       [ 0.3429808 ,  0.00330087,  0.14964075, ..., -0.73518964,
        -0.68519336, -0.27575966],
       [-0.84488505,  0.1597866 , -0.47073225, ..., -0.24020459,
        -0.37110101,  1.17073215],
       [-0.84488505, -0.8730192 ,  0.04624525, ..., -0.20212881,
        -0.47378505, -0.87137393]])
```

In [ ]:
```
#spliting the dataset
```

In [21]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,stratify=y,ra
```

In [14]:
```
# Building the kodel using SVC
```

In [22]:
```
model=SVC(kernel="linear")
```

In [24]:
```
model.fit(x_train,y_train)
```

Out[24]:
```
SVC(kernel='linear')
```
(https://scikit-learn.org/1.6/modules/generated/sklearn.svm.SVC.html)

In [26]:
```
# Testing the accuracy of the test data
```

In [27]:
```
y_train_predict=model.predict(x_train)
train_accuracy=accuracy_score(y_train_predict,y_train)
```

In [28]:
```
print("The training accurcay is:",train_accuracy)
```

```
The training accurcay is: 0.7915309446254072
```

In [ ]:
```
#Testing the accuracy of training data
```

In [29]:
```
y_pred=model.predict(x_test)
test_accuracy=accuracy_score(y_pred,y_test)
print("The test accuracy is..",test_accuracy)
```

```
The test accuracy is.. 0.7207792207792207
```

In [ ]:
```
#checking outside data
```

In [41]:
```
input_data=(4,110,92,0,0,37.6,0.191,30)
my_data=np.array(input_data)
my_data_reshap=my_data.reshape(1,-1)
std_data=scaler.transform(my_data_reshap)
std_data
```

```
C:\Users\user\anaconda3\Lib\site-packages\sklearn\utils\validation.py:2739: U
serWarning: X does not have valid feature names, but StandardScaler was fitte
d with feature names
  warnings.warn(
```

Out[41]:
```
array([[ 0.04601433, -0.34096773,  1.18359575, -1.28821221, -0.69289057,
         0.71168975, -0.84827977, -0.27575966]])
```

In [42]:
```
testing=model.predict(std_data)
testing
```

```
C:\Users\user\anaconda3\Lib\site-packages\sklearn\utils\validation.py:2739: U
serWarning: X does not have valid feature names, but SVC was fitted with feat
ure names
  warnings.warn(
```

Out[42]:
```
array([0], dtype=int64)
```

In [43]:
```
if testing[0]==0:
    print("not diabetic")
else:
    print("yes diabetic")
```

```
not diabetic
```

In [ ]: