In [1]:
```
import pandas as pd
import seaborn as sns
```

In [3]:
```
df = pd.read_csv("diabetes.csv")
df
```

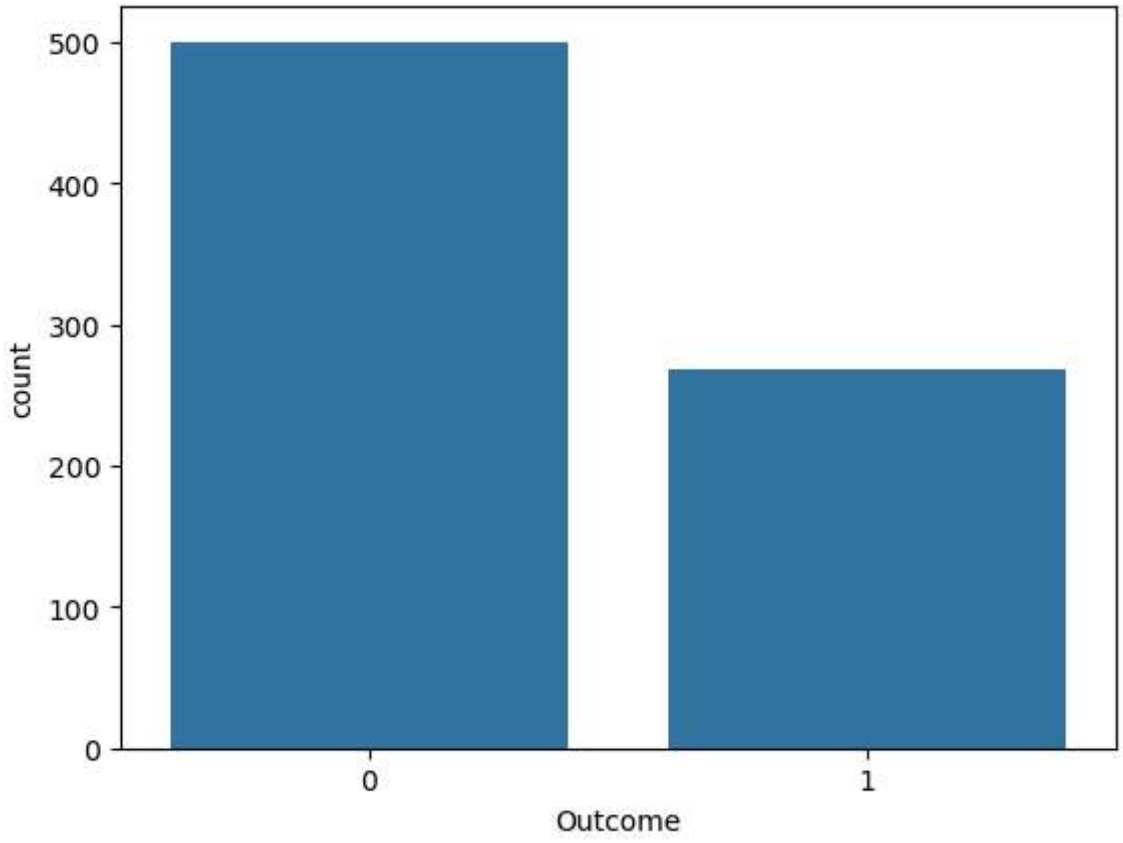
Out[3]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Pedigree	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 × 9 columns

In [5]:
```
x = df.drop('Outcome', axis = 1)
y = df['Outcome']
```

In [7]:
```
sns.countplot(x=y);
```



In [9]:
```
y.value_counts()
```

Out[9]:
```
Outcome
0    500
1    268
Name: count, dtype: int64
```

In [11]:
```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
x_scaled = scaler.fit_transform(x)
```

In [15]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x_scaled,y,random_state=50)
```

In [17]:
```
x.shape
```

Out[17]:
```
(768, 8)
```

In [19]:
```
x_train.shape
```

Out[19]:
```
(576, 8)
```

In [21]:
```
x_test.shape
```

Out[21]:
```
(192, 8)
```

In [23]:
```
from sklearn.neighbors import KNeighborsClassifier
```

In [25]:
```
knn =  KNeighborsClassifier(n_neighbors = 5)
```

In [27]:
```
knn.fit(x_train, y_train)
```

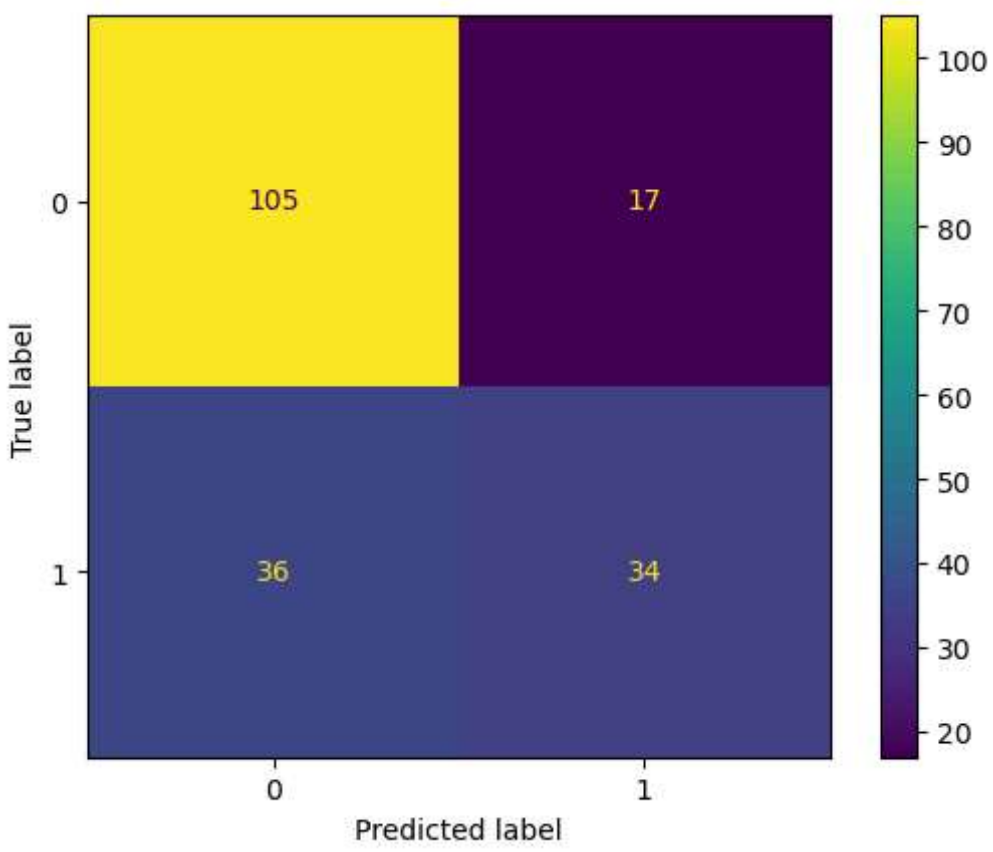
Out[27]:
```
KNeighborsClassifier
KNeighborsClassifier()
```

In [29]:
```
from sklearn.metrics import accuracy_score , ConfusionMatrixDisplay
from sklearn.metrics import classification_report
```

In [31]:
```
y_pred = knn.predict(x_test)
```

In [33]:
```
ConfusionMatrixDisplay.from_predictions(y_test,y_pred)
```

Out[33]:
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x12c54b8b620
>
```



In [35]:
```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.74      0.86      0.80       122
           1       0.67      0.49      0.56        70

    accuracy                           0.72       192
   macro avg       0.71      0.67      0.68       192
weighted avg       0.72      0.72      0.71       192
```

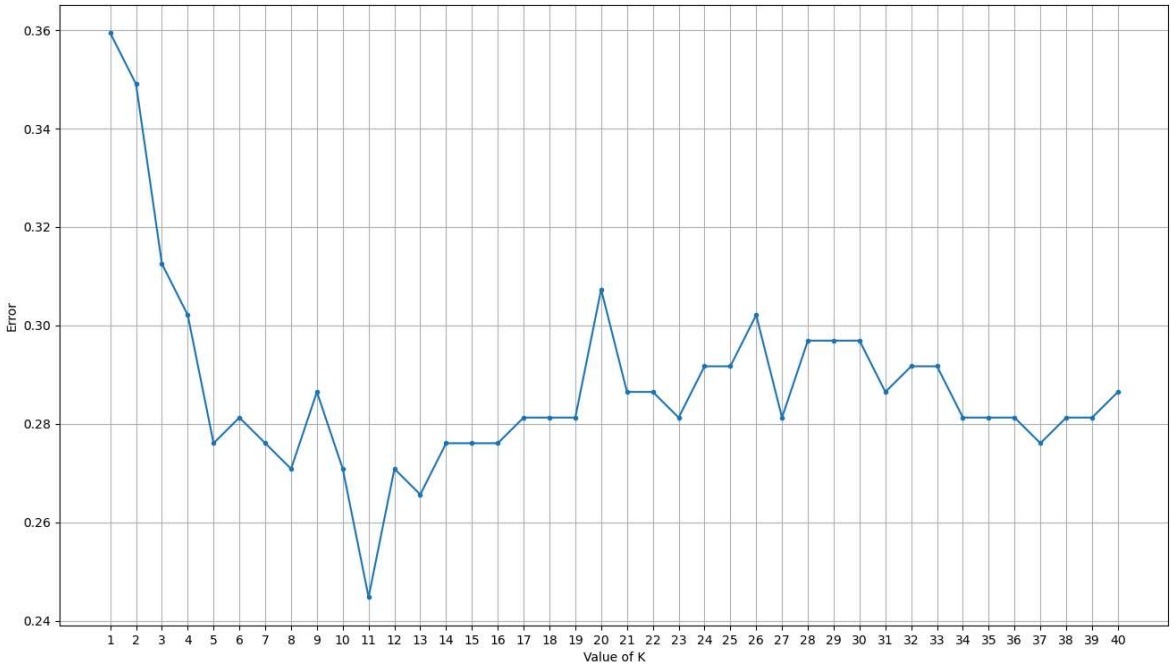
In [37]:
```
import matplotlib.pyplot as plt
import numpy as np
```

In [39]:
```
error = []
for k in range (1,41):
    knn = KNeighborsClassifier(n_neighbors = k)
```

```
    knn.fit(x_train, y_train)
    pred=knn.predict(x_test)
    error.append(np.mean(pred!=y_test))
```

In [43]:
```
plt.figure(figsize=(16,9))
plt.xlabel('Value of K')
plt.ylabel('Error')
plt.grid()
plt.xticks(range(1,41))
plt.plot(range(1,41),error,marker='.')
```

Out[43]: [<matplotlib.lines.Line2D at 0x12c59695550>]



In [45]:
```
knn = KNeighborsClassifier(n_neighbors = 33)
```

In [47]:
```
knn.fit(x_train, y_train)
```

Out[47]:
```
KNeighborsClassifier
KNeighborsClassifier(n_neighbors=33)
```

In [49]:
```
y_pred=knn.predict(x_test)
```

In [51]:
```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.72      0.89      0.80       122
           1       0.68      0.39      0.49        70

    accuracy                           0.71       192
   macro avg       0.70      0.64      0.64       192
weighted avg       0.70      0.71      0.68       192
```

In [ ]: