In [163…

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

uci_data = pd.read_csv("./datasets/uci_diabetes_data.csv")
```

Dataset 1

Content

- Age 1.20-65
- Gender 1. Male, 2.Female
- Polyuria 1.Yes, 2.No.
- Polydipsia 1.Yes, 2.No.
- sudden weight loss 1.Yes, 2.No.
- weakness 1.Yes, 2.No.
- Polyphagia 1.Yes, 2.No.
- Genital thrush 1.Yes, 2.No.
- visual blurring 1.Yes, 2.No.
- Itching 1.Yes, 2.No.
- Irritability 1.Yes, 2.No.
- delayed healing 1.Yes, 2.No.
- partial paresis 1.Yes, 2.No.
- muscle sti ness 1.Yes, 2.No.
- Alopecia 1.Yes, 2.No.
- Obesity 1.Yes, 2.No.
- Class 1.Positive, 2.Negative.

Number of Instances: 520

Number of Attributes: 17

For more detailed info:
https://archive.ics.uci.edu/ml/datasets/Early+stage+diabetes+risk+prediction+dataset.

reference essay: https://iopscience.iop.org/article/10.1088/1742-6596/1684/1/012062

In [164…

```
uci_data.head()
```

Out[164…

	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	Itching
0	40	Male	No	Yes	No	Yes	No	No	No	Yes
1	58	Male	No	No	No	Yes	No	No	Yes	No
2	41	Male	Yes	No	No	Yes	Yes	No	No	Yes
3	45	Male	No	No	Yes	Yes	Yes	Yes	No	Yes
4	60	Male	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes

In [165…

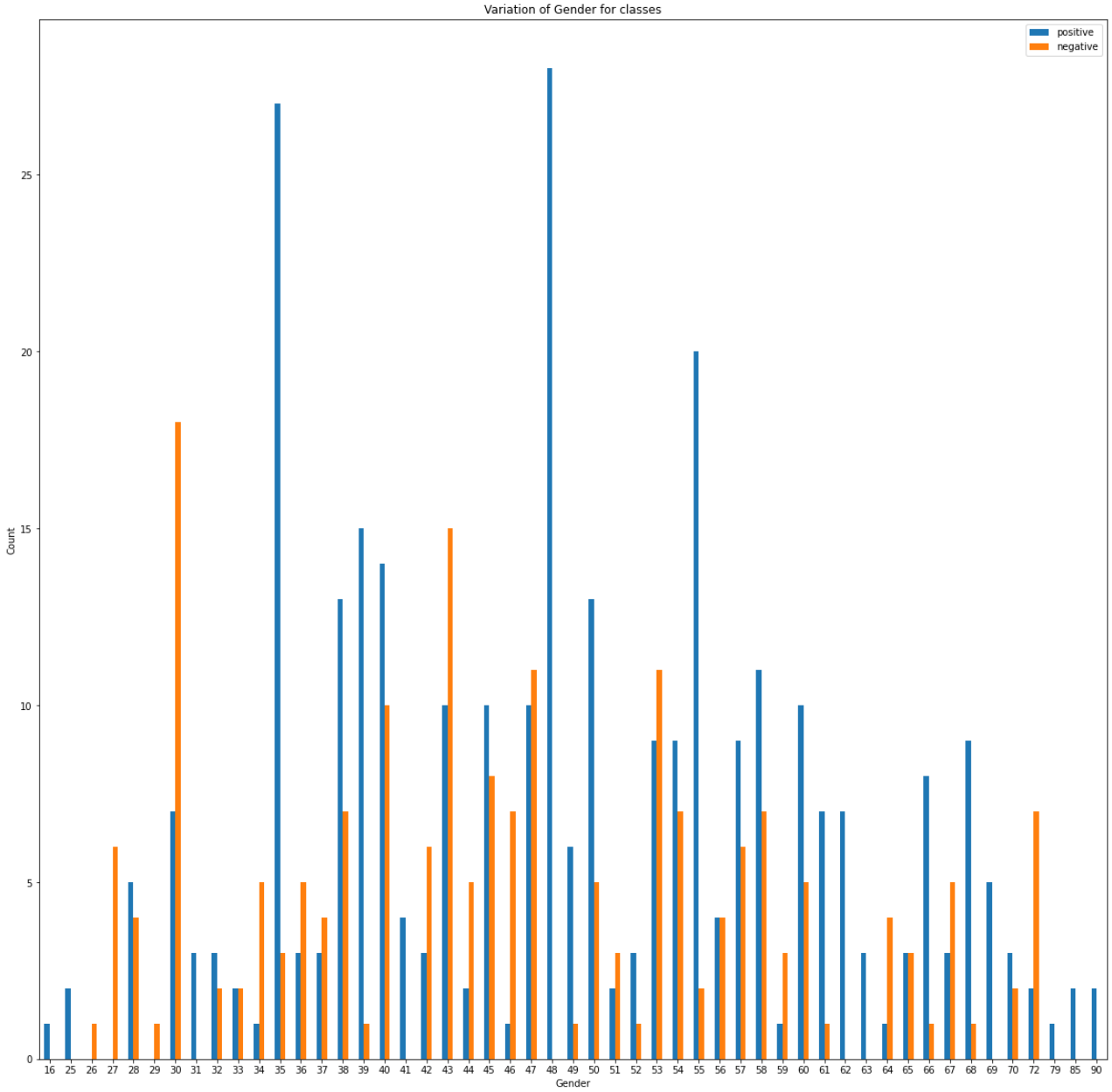
```
# format data
uci_data = uci_data.replace(to_replace =["Yes", "Positive"], value=1).replace(to
uci_data.head()
```

Out[165…

	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	Itching
0	40	Male	0	1	0	1	0	0	0	1
1	58	Male	0	0	0	1	0	0	1	0
2	41	Male	1	0	0	1	1	0	0	1
3	45	Male	0	0	1	1	1	1	0	1
4	60	Male	1	1	1	1	1	0	1	1

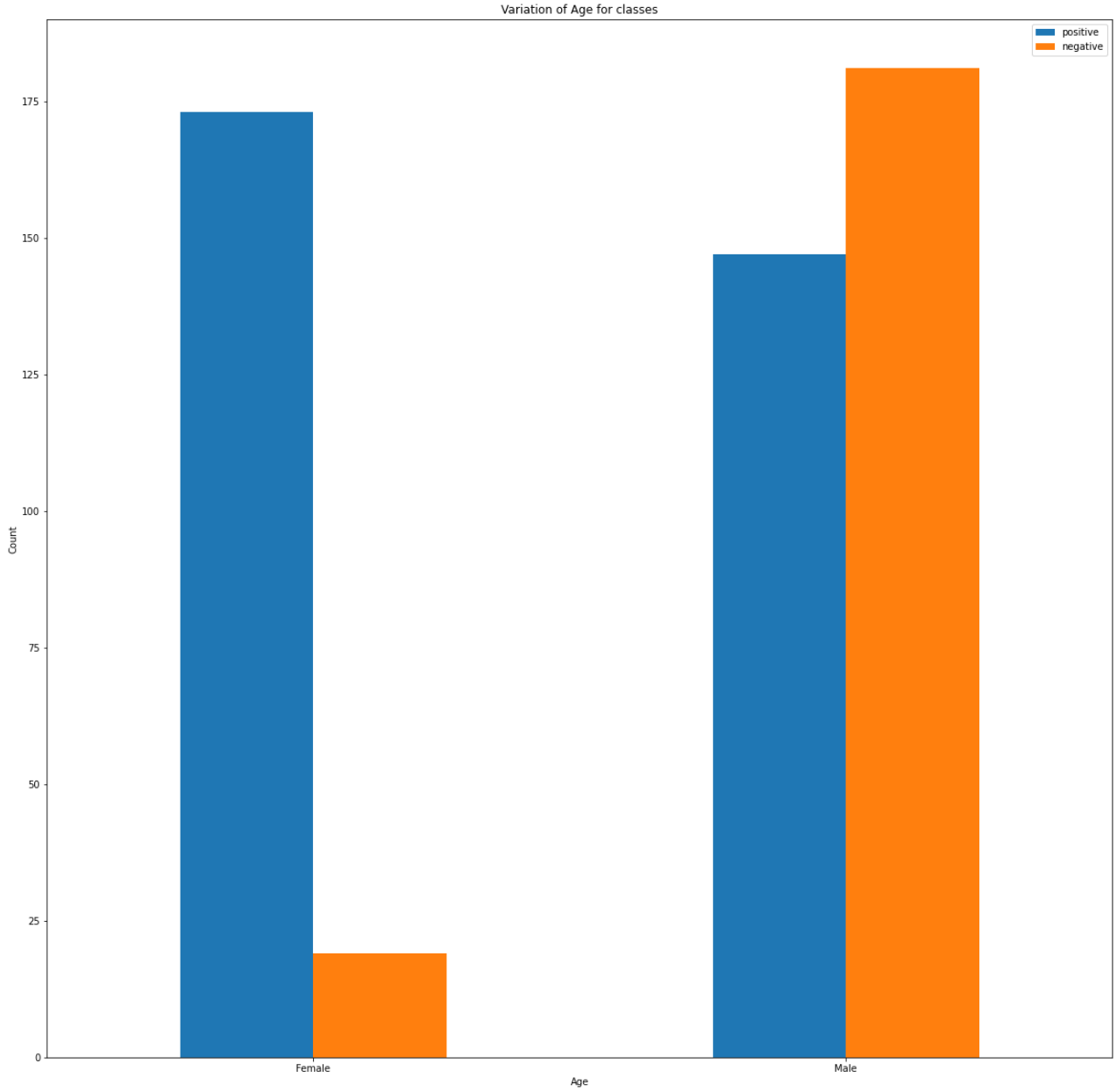
In [166…

```
positive = uci_data.loc[uci_data['class'] == 1]
negative = uci_data.loc[uci_data['class'] == 0]
number_positive_each_age = positive.groupby('Age')['class'].count()
number_negative_each_age = negative.groupby('Age')['class'].count()
result = pd.DataFrame(dict(positive = number_positive_each_age, negative = numbe
result.plot.bar(figsize=[20,20])
plt.xticks(rotation=360)
plt.title('Variation of Gender for classes')
plt.ylabel('Count')
plt.xlabel('Gender');
plt.show()
```



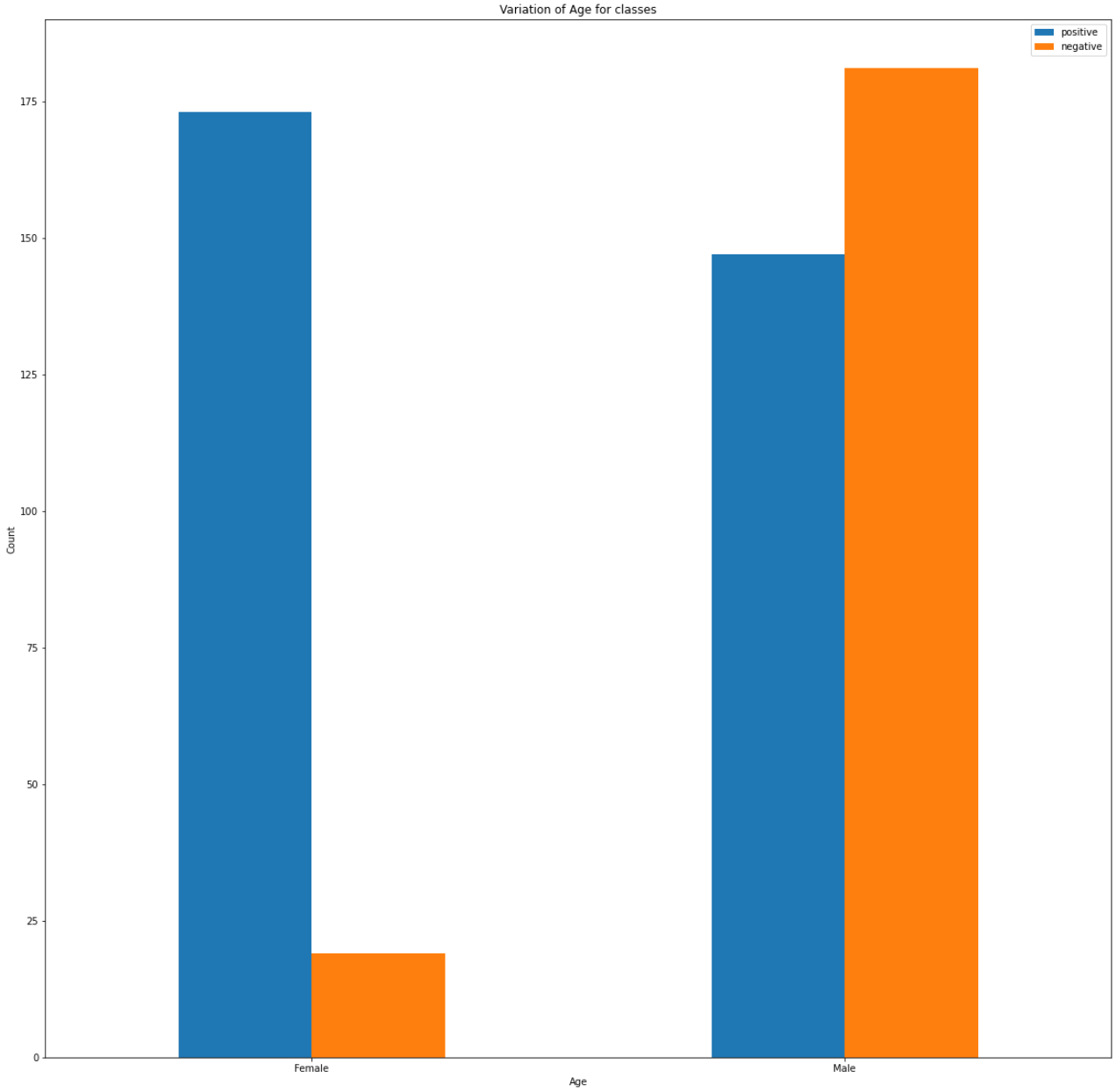
In [167…

```
number_positive_gender = positive.groupby('Gender')['class'].count()
number_negative_gender = negative.groupby('Gender')['class'].count()
result = pd.DataFrame(dict(positive = number_positive_gender, negative = number_
result.plot.bar(figsize=[20,20])
plt.xticks(rotation=360)
plt.title('Variation of Age for classes')
plt.ylabel('Count')
plt.xlabel('Age');
plt.show()
```



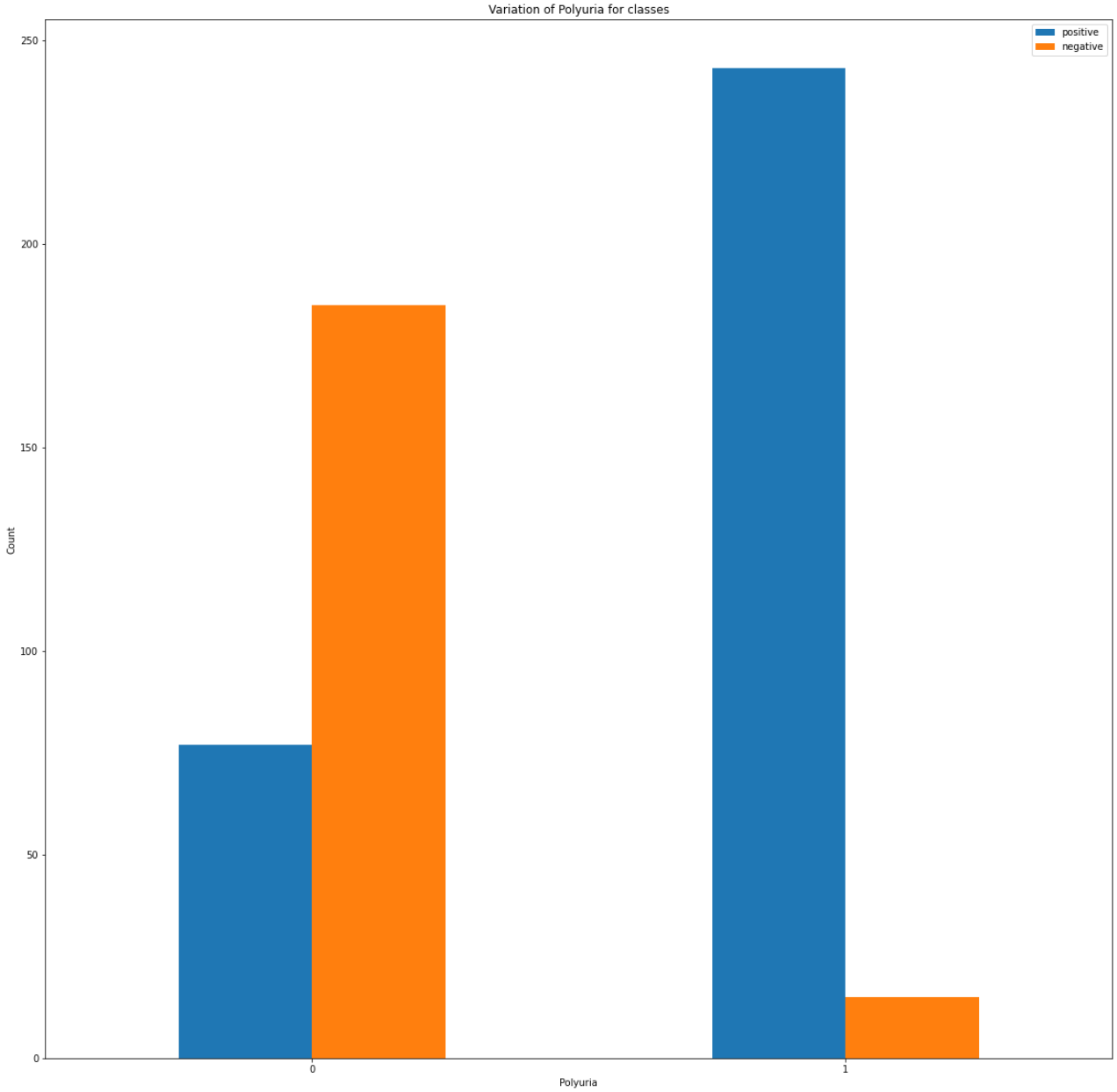
In [168…

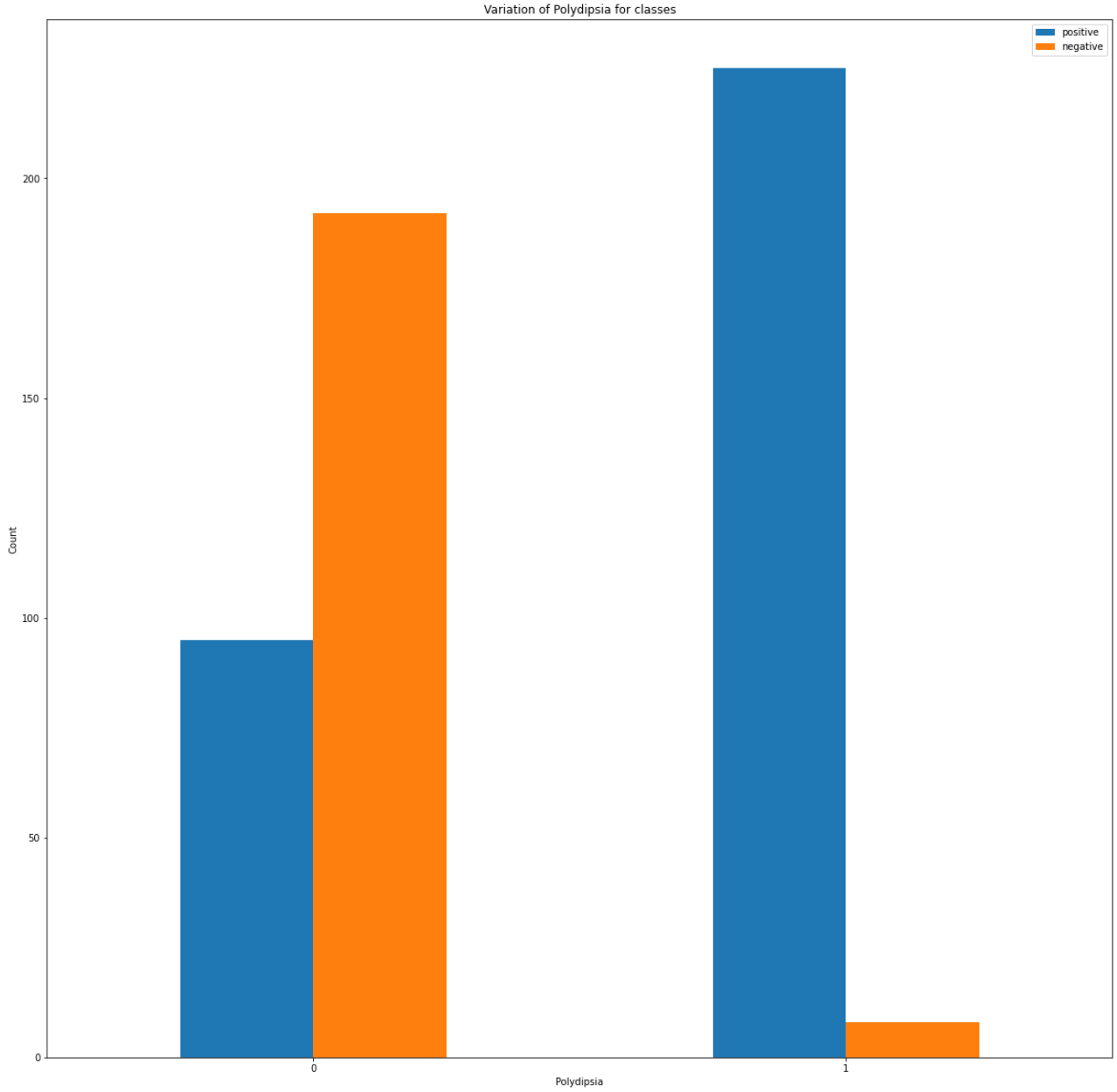
```
number_positive_gender = positive.groupby('Gender')['class'].count()
number_negative_gender = negative.groupby('Gender')['class'].count()
result = pd.DataFrame(dict(positive = number_positive_gender, negative = number_
result.plot.bar(figsize=[20,20])
plt.xticks(rotation=360)
plt.title('Variation of Age for classes')
plt.ylabel('Count')
plt.xlabel('Age');
plt.show()
```

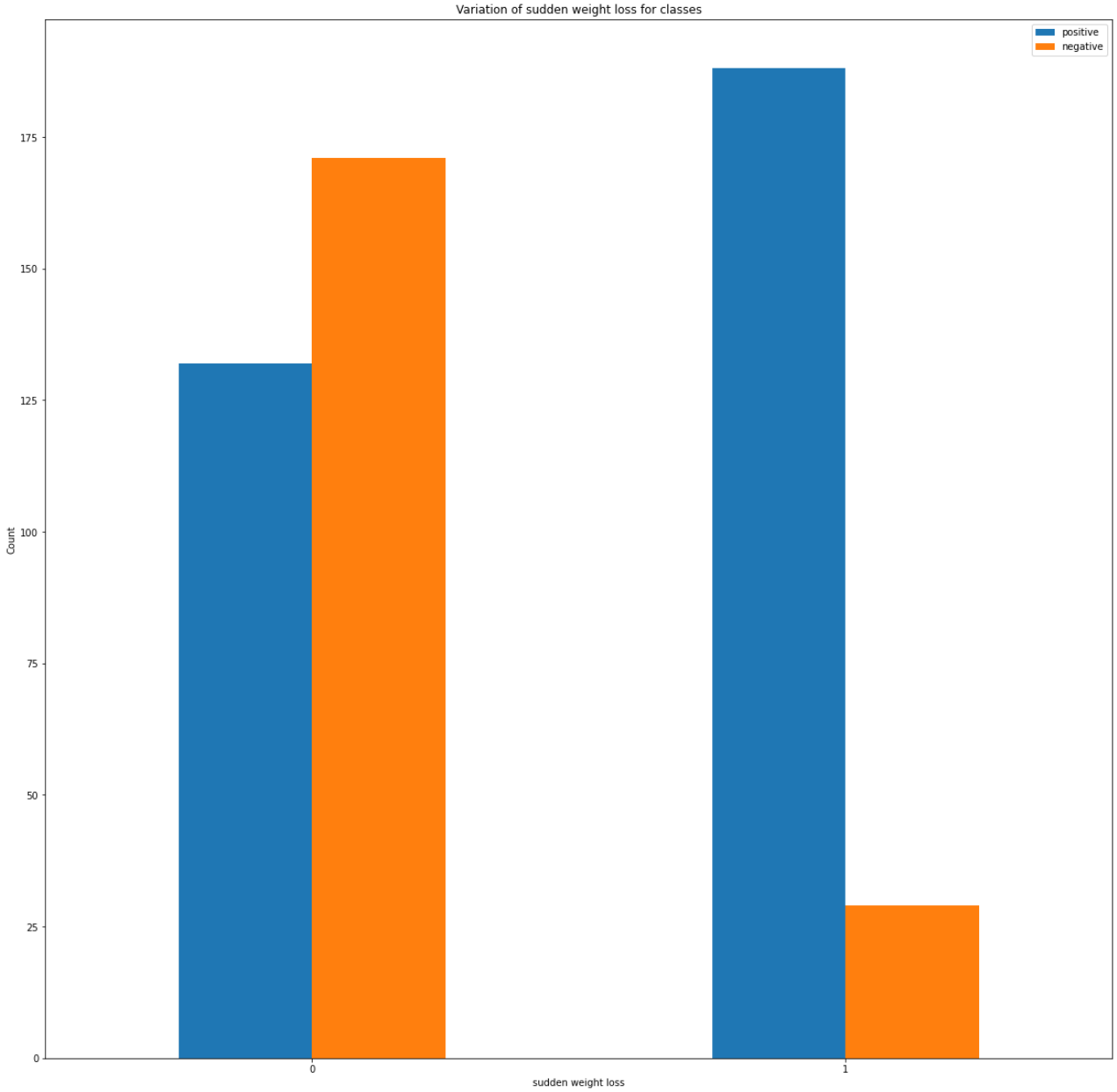


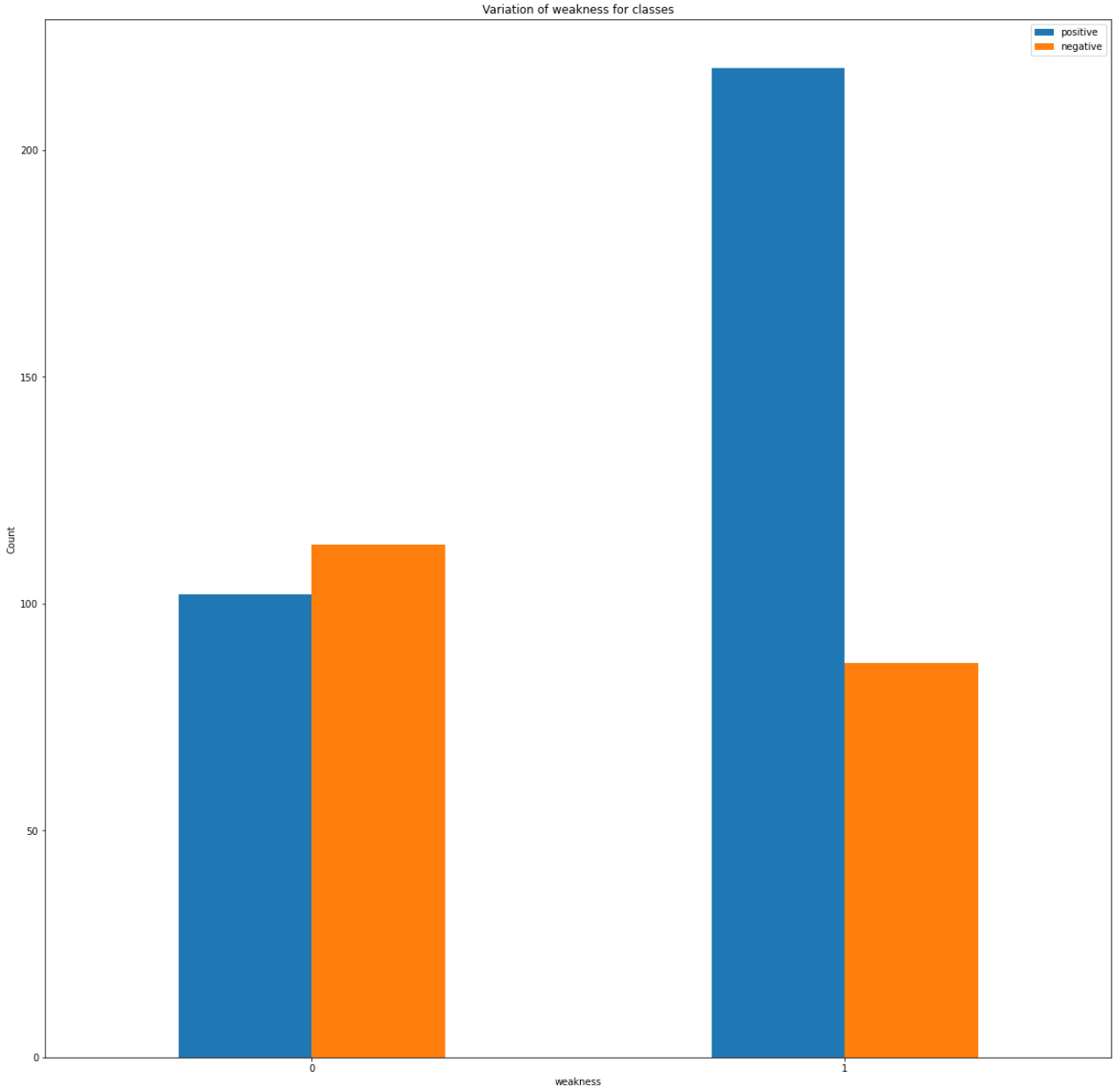
In [169…

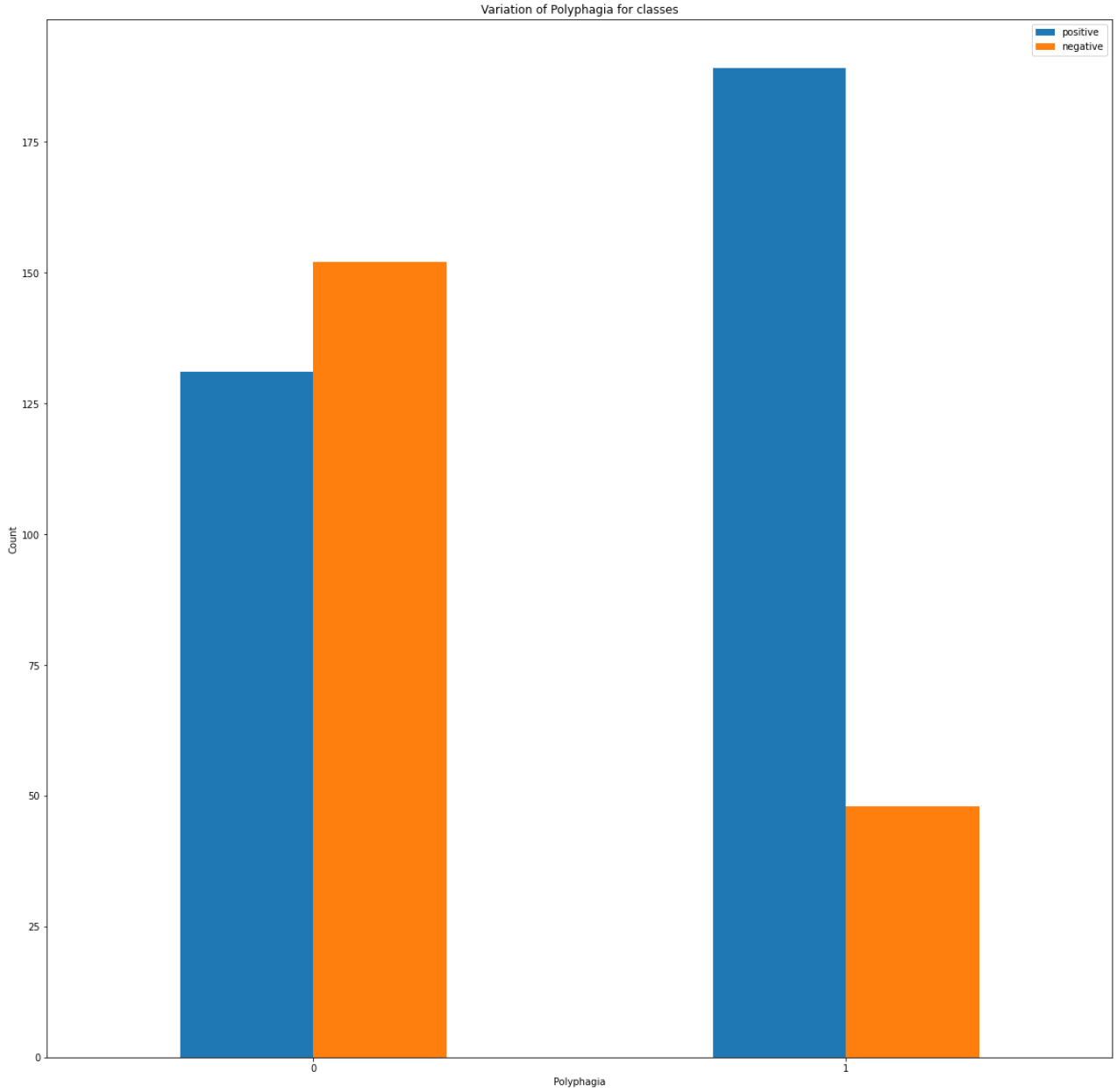
```
# check the distribution of each feature
for i in range(2, 16):
    number_positive_feature = positive.groupby(uci_data.columns[i])['class'].cou
    number_negative_feature = negative.groupby(uci_data.columns[i])['class'].cou
    result = pd.DataFrame(dict(positive = number_positive_feature, negative = nu
    result.plot.bar(figsize=[20,20])
    plt.xticks(rotation=360)
    plt.title('Variation of ' + uci_data.columns[i] + ' for classes')
    plt.ylabel('Count')
    plt.xlabel(uci_data.columns[i]);
    plt.show()
```

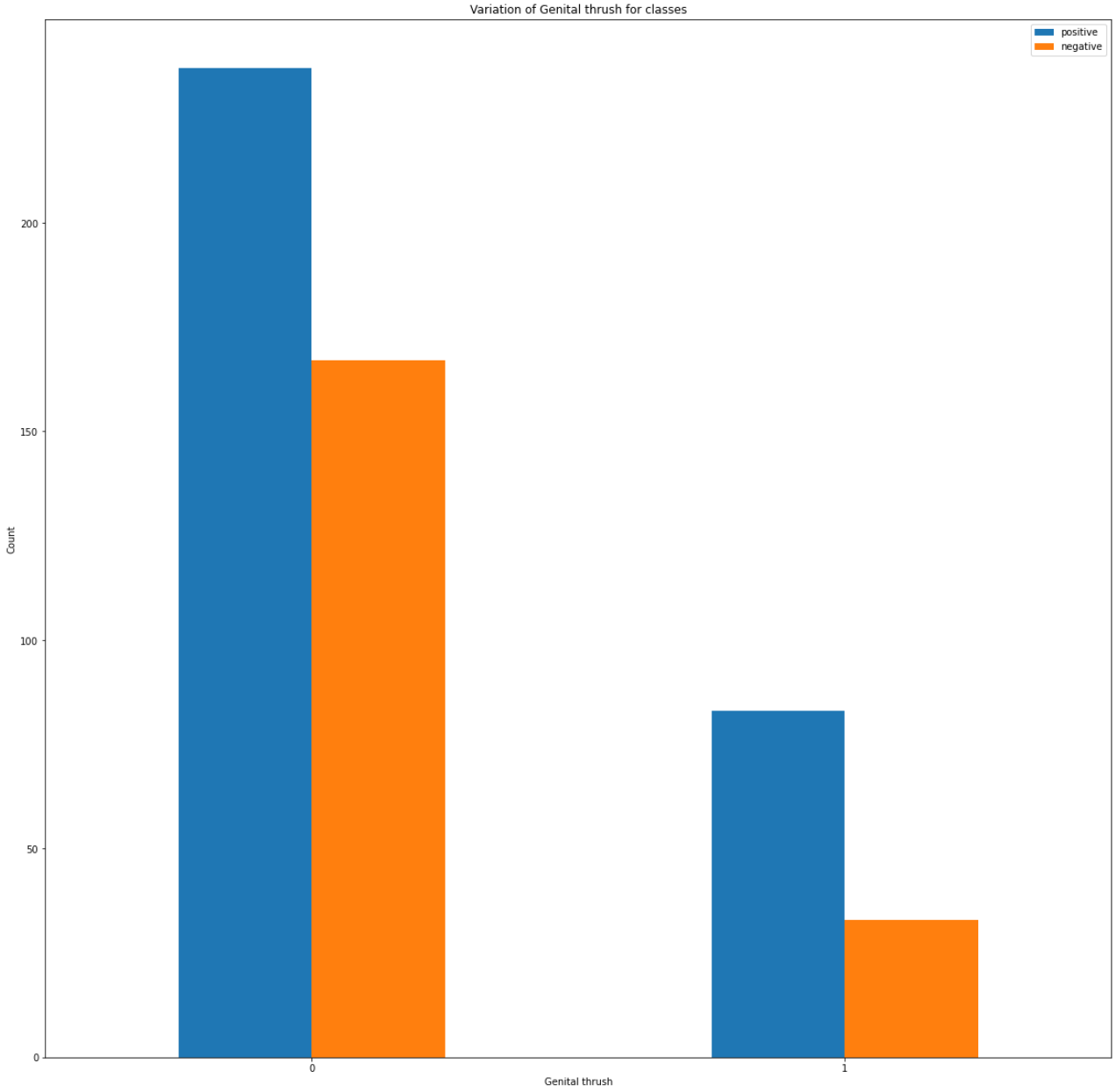

Variation of Polyuria for classes
positive
negative
250
200
150
100
50
0
Count
0
1
Polyuria

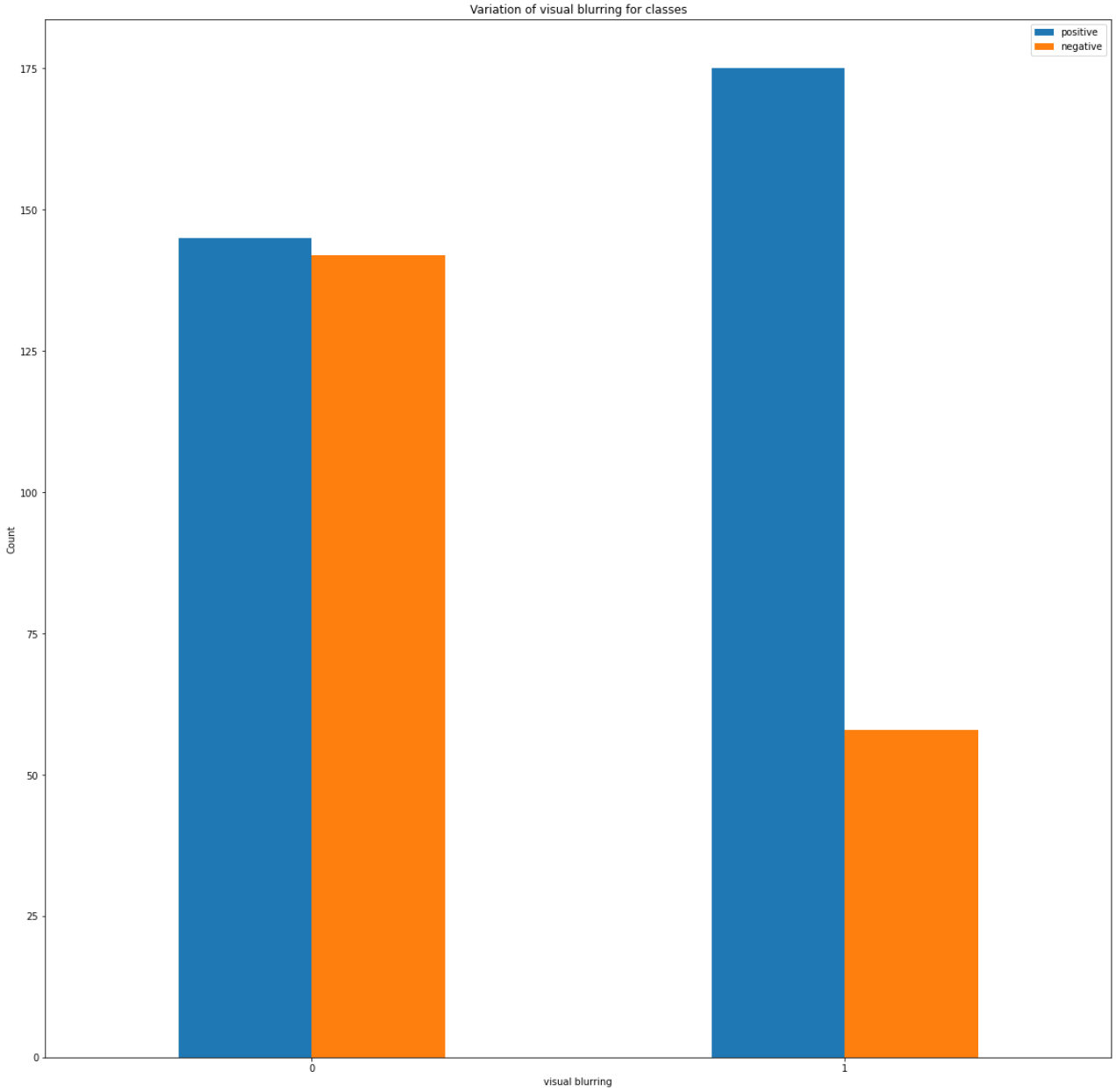

Variation of Polydipsia for classes
positive
negative
Count
0
50
100
150
200
0
1
Polydipsia

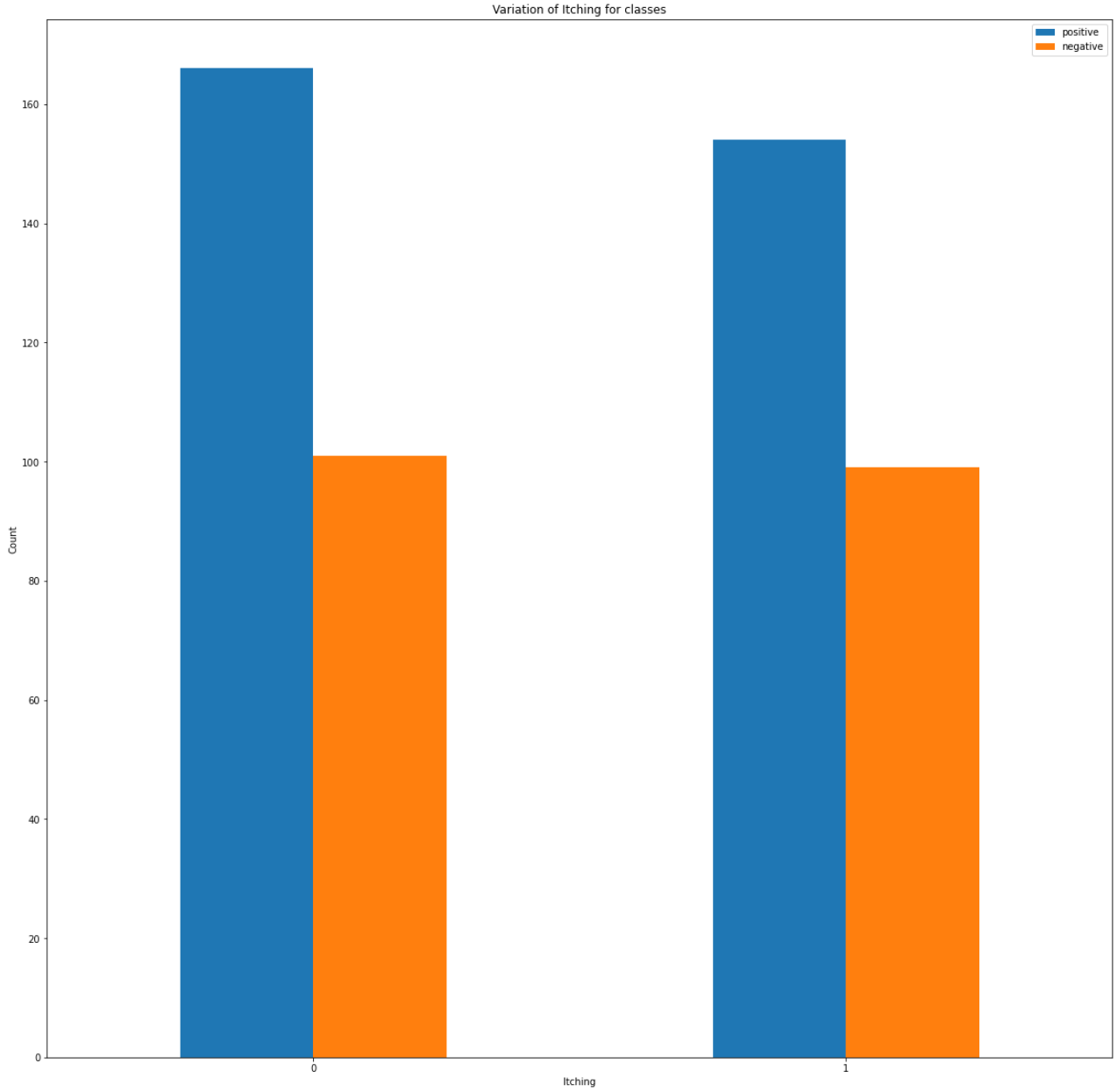

Variation of sudden weight loss for classes
positive
negative
175
150
125
100
75
50
25
0
Count
0
1
sudden weight loss

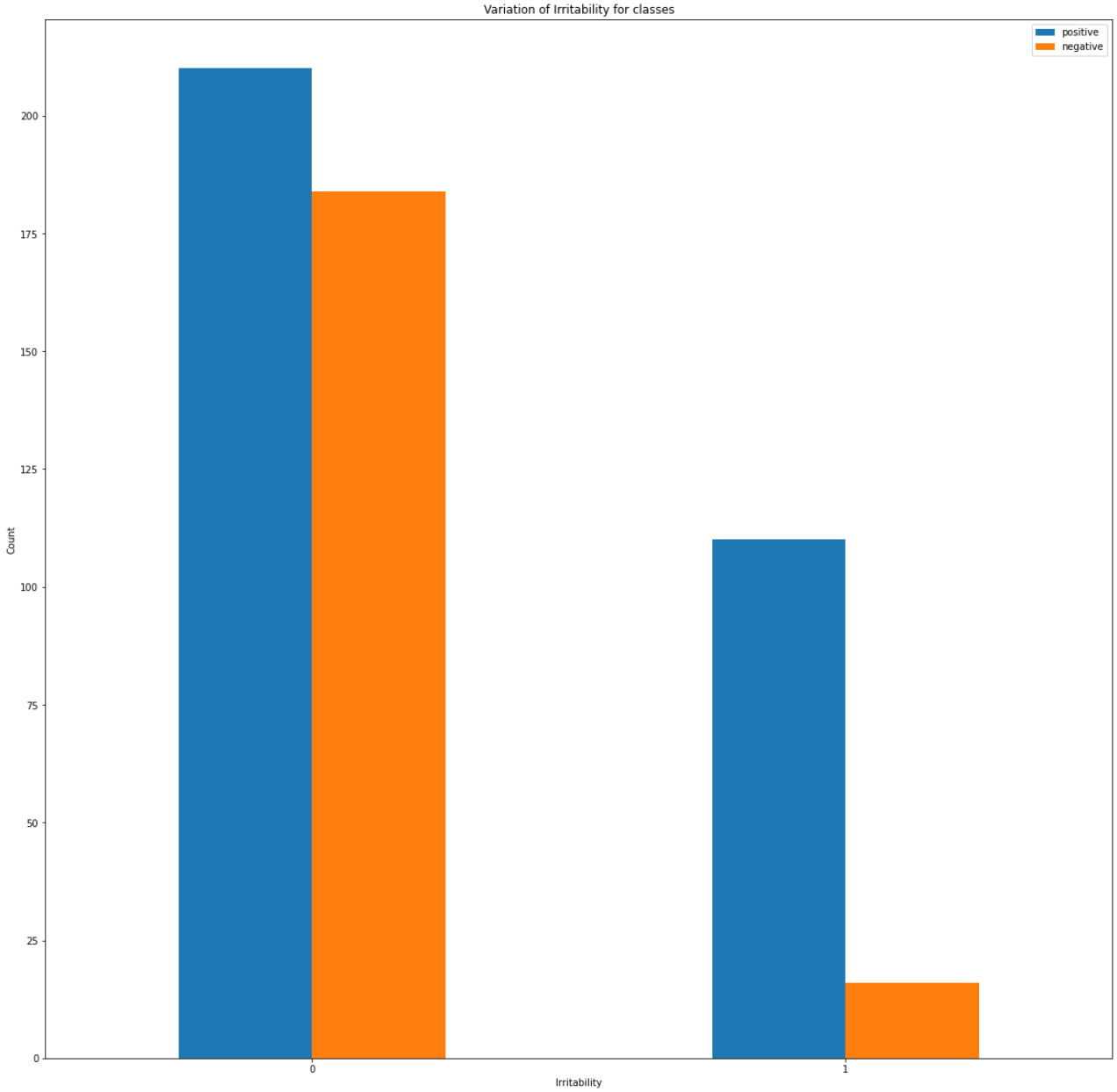

Variation of weakness for classes
positive
negative
200
150
100
50
0
Count
0
1
weakness

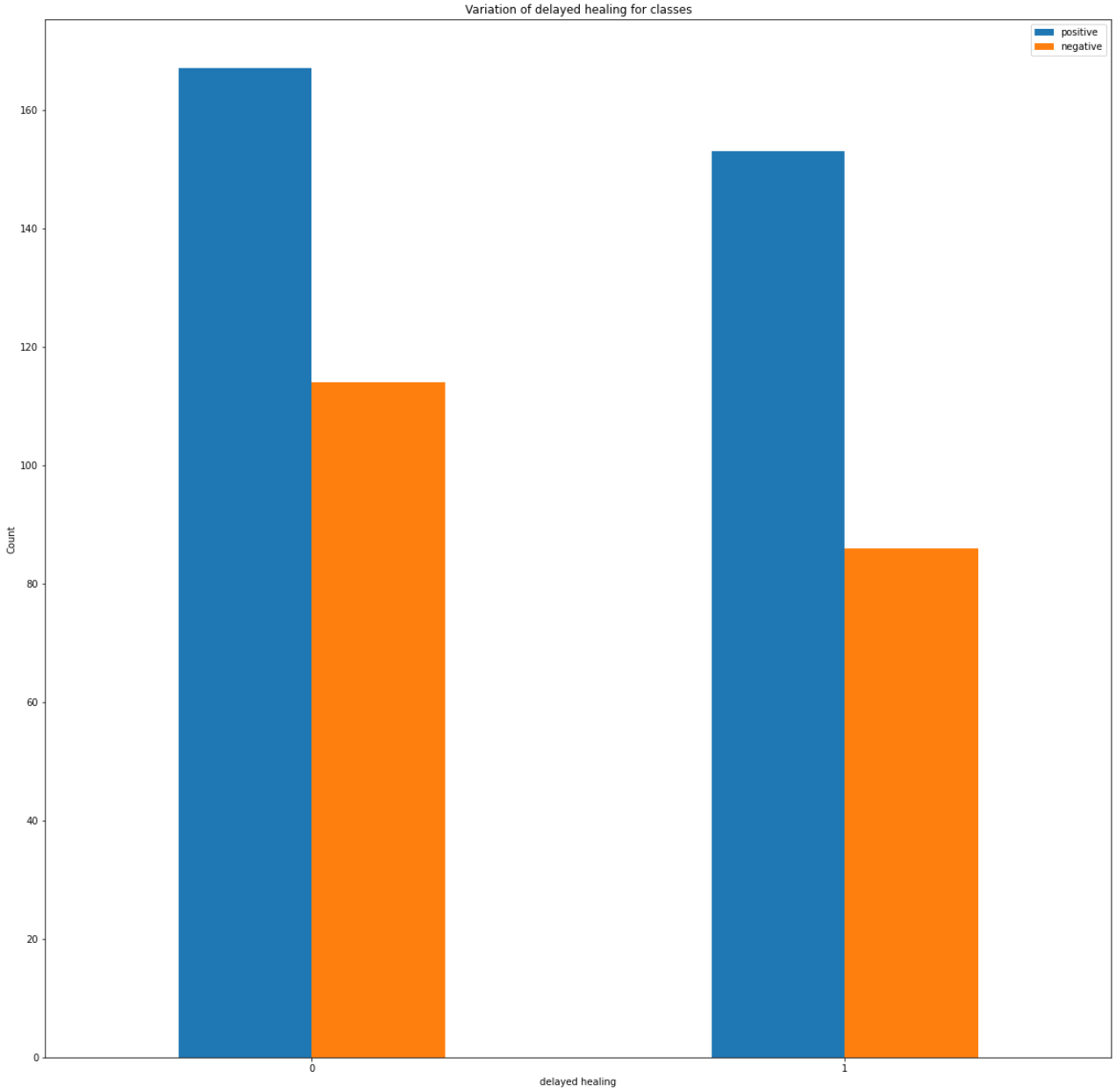

Variation of Polyphagia for classes
positive
negative
175
150
125
100
75
50
25
0
Count
0
1
Polyphagia

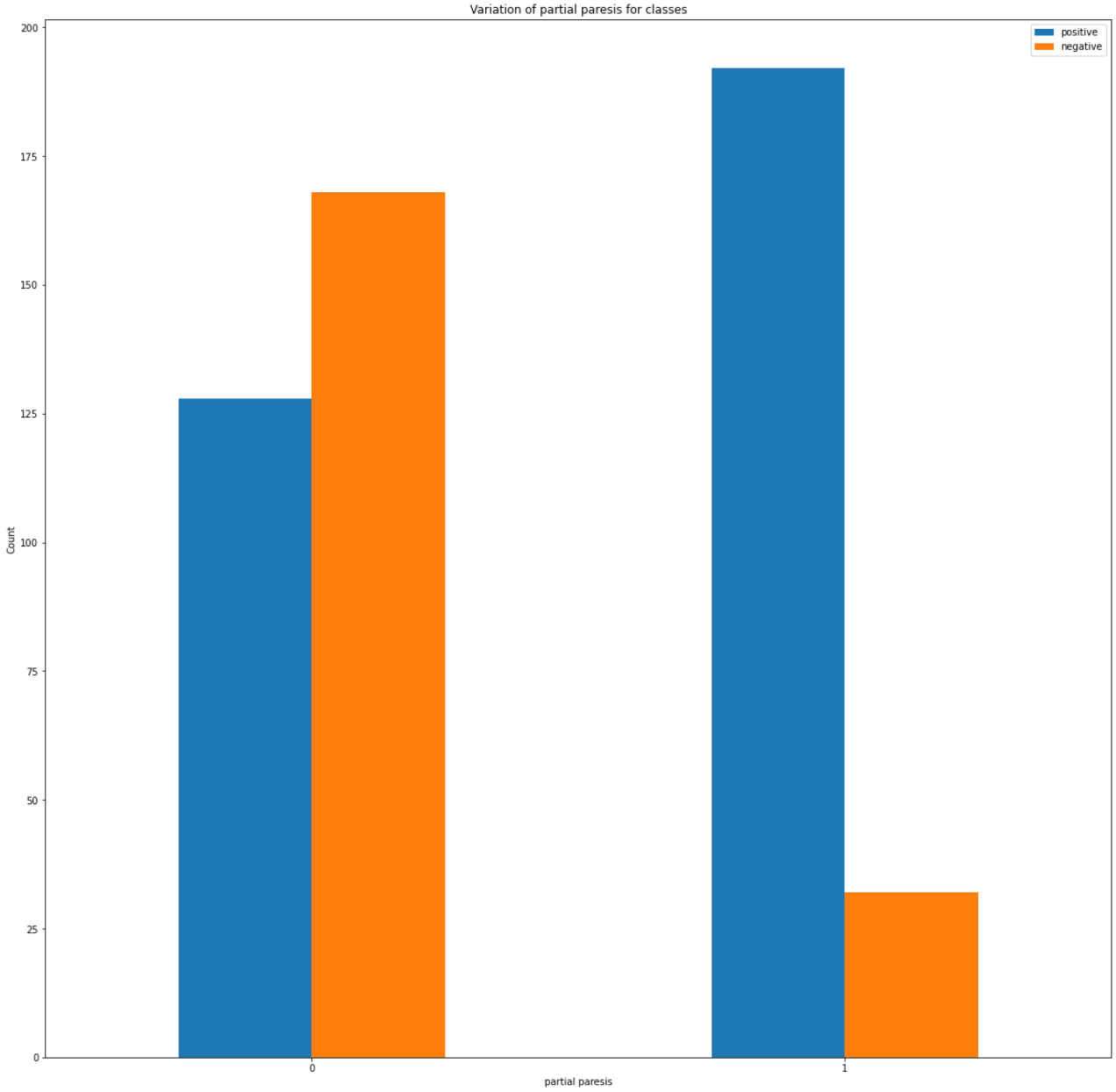

Variation of Genital thrush for classes
positive
negative
Count
0
50
100
150
200
0
1
Genital thrush

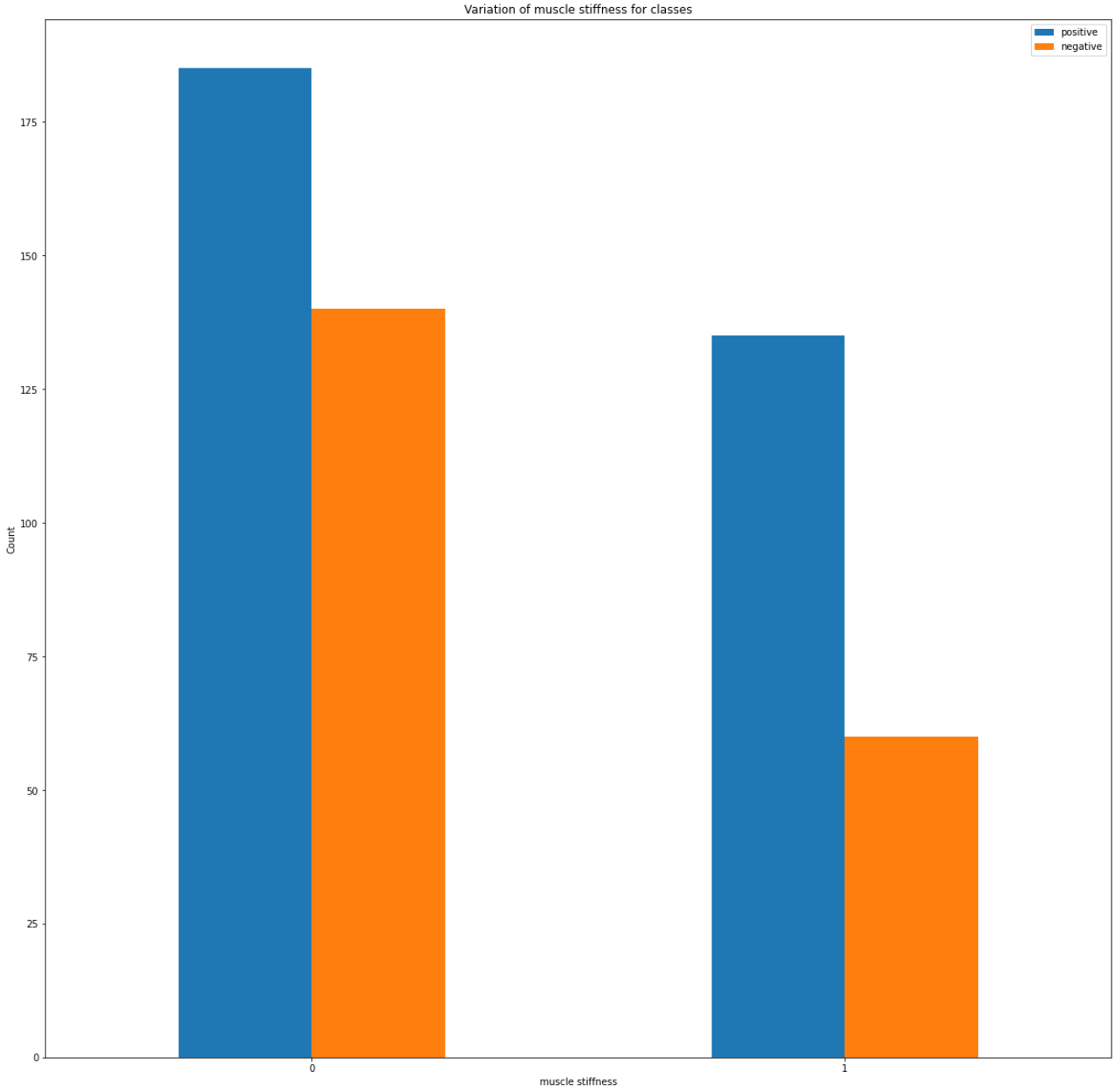

Variation of visual blurring for classes
positive
negative
175
150
125
100
75
50
25
0
Count
0
1
visual blurring

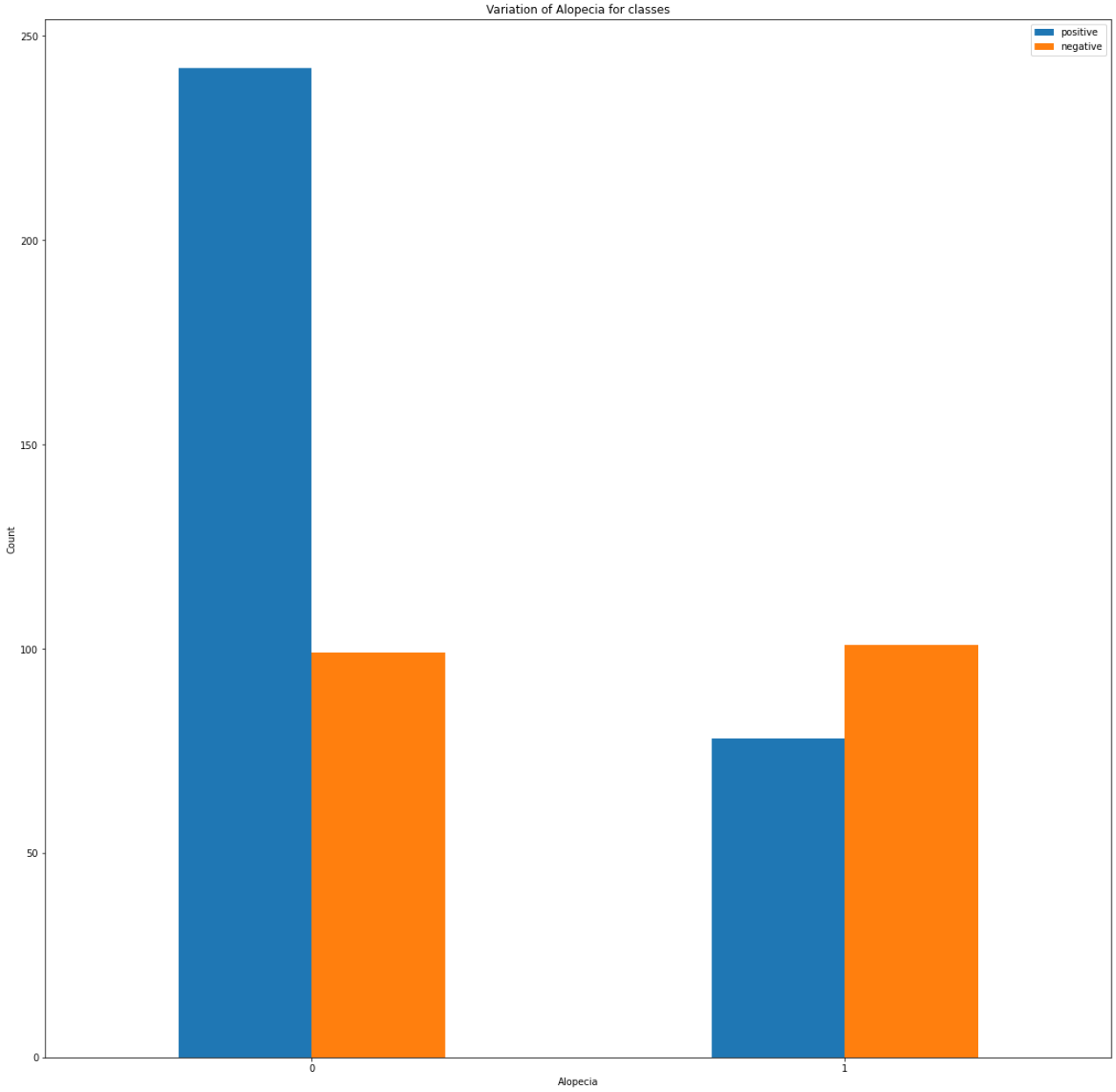

Variation of Itching for classes
positive
negative
160
140
120
100
80
60
40
20
0
Count
0
1
Itching

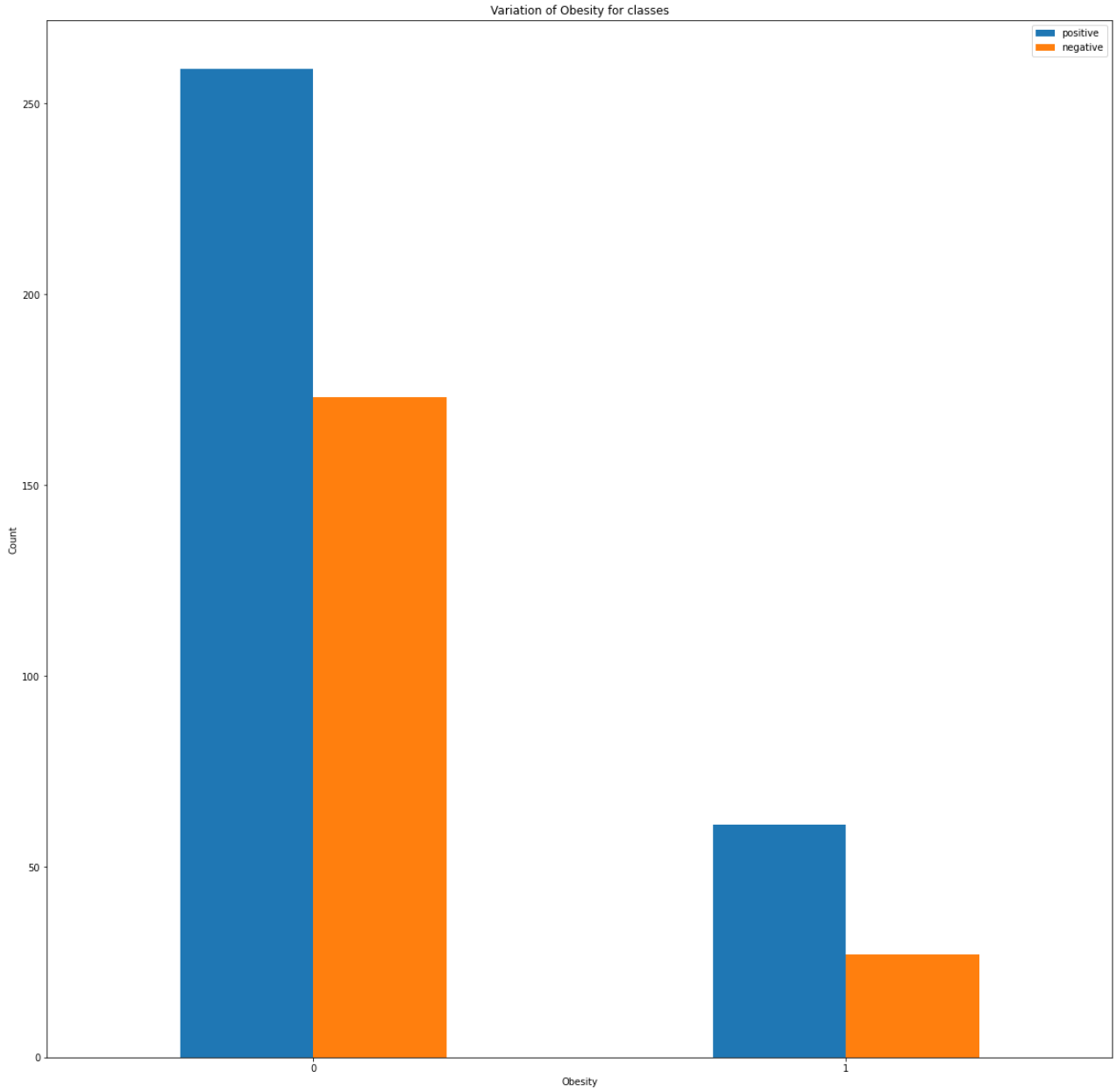

Variation of Irritability for classes
positive
negative
Count
0
25
50
75
100
125
150
175
200
0
1
Irritability


Variation of delayed healing for classes
positive
negative
160
140
120
100
80
60
40
20
0
Count
0
1
delayed healing


Variation of partial paresis for classes
positive
negative
Count
200
175
150
125
100
75
50
25
0
0
1
partial paresis


Variation of muscle stiffness for classes
positive
negative
Count
0
25
50
75
100
125
150
175
0
1
muscle stiffness


Variation of Alopecia for classes
positive
negative
250
200
150
100
50
0
Count
0
1
Alopecia



Dataset 2

Content

- Pregnancies Pregnancies Insulin BMI Age Glucose BloodPressure DiabetesPedigreeFunction Outcome
- Insulin
- BMI
- Age
- Glucose
- BloodPressure
- DiabetesPedigreeFunction
- Outcome

Number of Instances: 2000

Number of Attributes: 7

For more detailed info:https://www.kaggle.com/johndasilva/diabetes

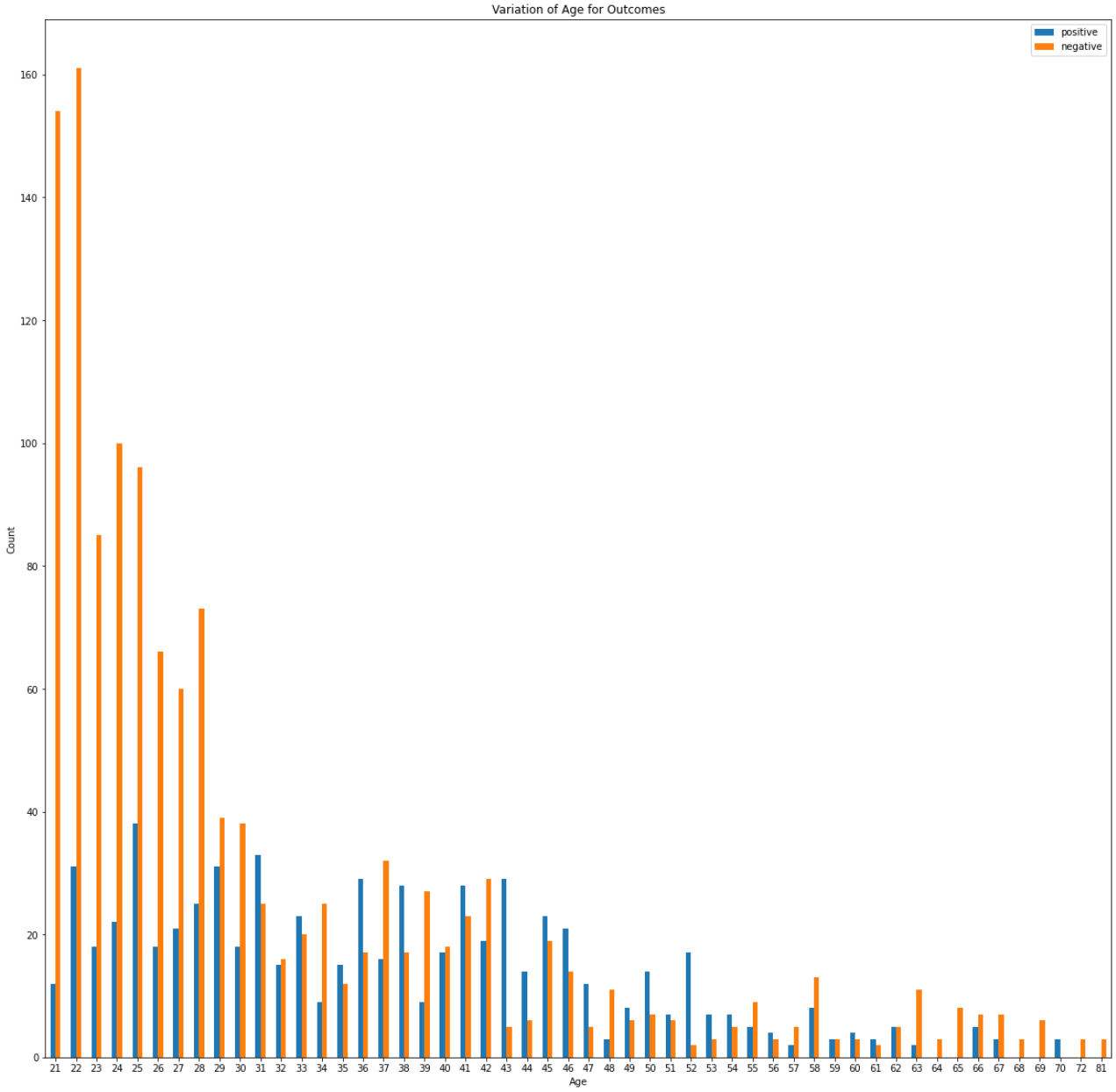
In [170…
```
frankfurt_data = pd.read_csv("./datasets/Frankfurt_diabetes.csv")
```

In [171…
```
frankfurt_data.head()
```

Out[171…

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction
0	2	138	62	35	0	33.6	0.127
1	0	84	82	31	125	38.2	0.233
2	0	145	0	0	0	44.2	0.630
3	0	135	68	42	250	42.3	0.365
4	1	139	62	41	480	40.7	0.536

In [172…
```
positive = frankfurt_data.loc[frankfurt_data['Outcome'] == 1]
negative = frankfurt_data.loc[frankfurt_data['Outcome'] == 0]
number_positive_each_age = positive.groupby('Age')['Outcome'].count()
number_positive_each_age
number_negative_each_age = negative.groupby('Age')['Outcome'].count()
number_negative_each_age
result = pd.DataFrame(dict(positive = number_positive_each_age, negative = numbe
result.plot.bar(figsize=[20,20])
plt.xticks(rotation=360)
plt.title('Variation of Age for Outcomes')
plt.ylabel('Count')
plt.xlabel('Age');
plt.show()
```



In [173…
```
frankfurt_data.isnull().sum()
```

Out[173…
```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

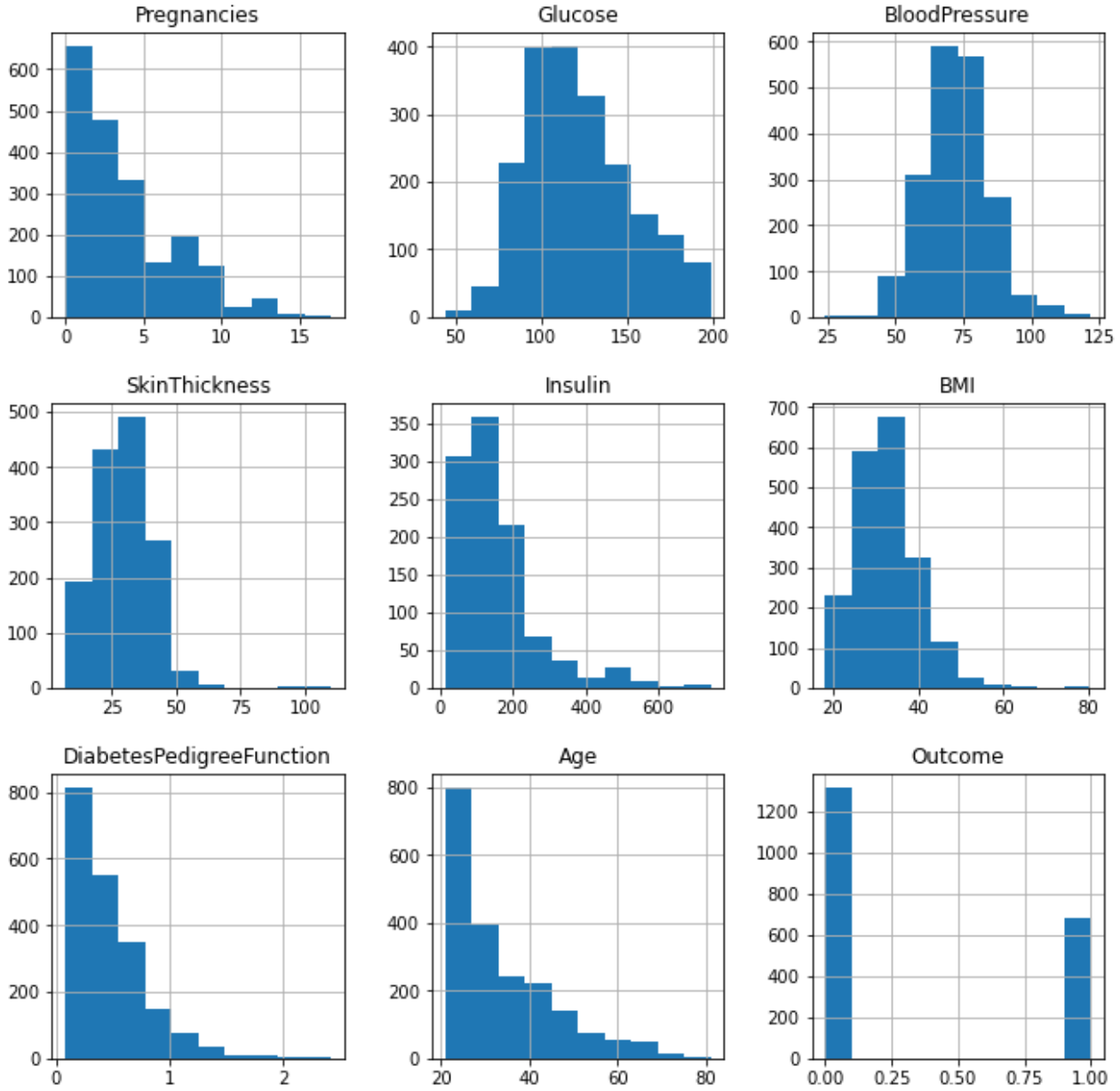
In [174…
```
frankfurt_data[['Glucose','BloodPressure','SkinThickness','Insulin','BMI']] = fr
frankfurt_data.isnull().sum()
```

Out[174…
```
Pregnancies                 0
Glucose                    13
BloodPressure              90
SkinThickness             573
Insulin                   956
```

```
BMI                         28
DiabetesPedigreeFunction     0
Age                          0
Outcome                      0
dtype: int64
```

In [175…
```
frankfurt_data.hist(figsize = (11,11))
```

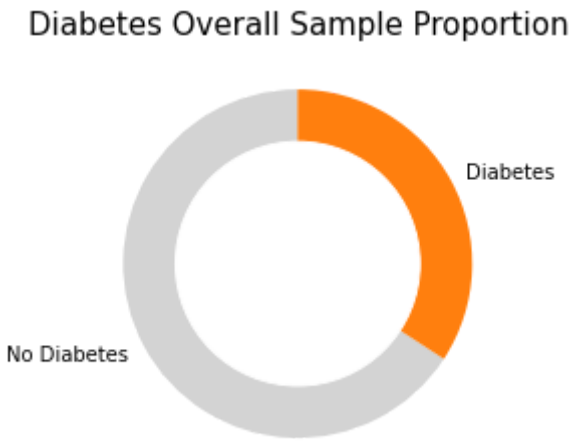
Out[175…
```
array([[<AxesSubplot:title={'center':'Pregnancies'}>,
        <AxesSubplot:title={'center':'Glucose'}>,
        <AxesSubplot:title={'center':'BloodPressure'}>],
       [<AxesSubplot:title={'center':'SkinThickness'}>,
        <AxesSubplot:title={'center':'Insulin'}>,
        <AxesSubplot:title={'center':'BMI'}>],
       [<AxesSubplot:title={'center':'DiabetesPedigreeFunction'}>,
        <AxesSubplot:title={'center':'Age'}>,
        <AxesSubplot:title={'center':'Outcome'}>]], dtype=object)
```



In [176…
```
frankfurt_data['Glucose'].fillna(frankfurt_data['Glucose'].mean(), inplace = Tru
frankfurt_data['BloodPressure'].fillna(frankfurt_data['BloodPressure'].mean(), i
frankfurt_data['SkinThickness'].fillna(frankfurt_data['SkinThickness'].mean(), i
frankfurt_data['Insulin'].fillna(frankfurt_data['Insulin'].mean(), inplace = Tru
frankfurt_data['BMI'].fillna(frankfurt_data['BMI'].mean(), inplace = True)
```

In [177…

```
import matplotlib.pyplot as plt
labels = ['No Diabetes','Diabetes']
colormap = {'lightgrey','tab:orange'}
frankfurt_data['Outcome'].value_counts().plot.pie(startangle=90, colors=colormap
plt.title("Diabetes Overall Sample Proportion", fontsize=15)
plt.ylabel('')
circle = plt.Circle((0,0),0.7,color="white")
p = plt.gcf()
p.gca().add_artist(circle)
plt.show()
```

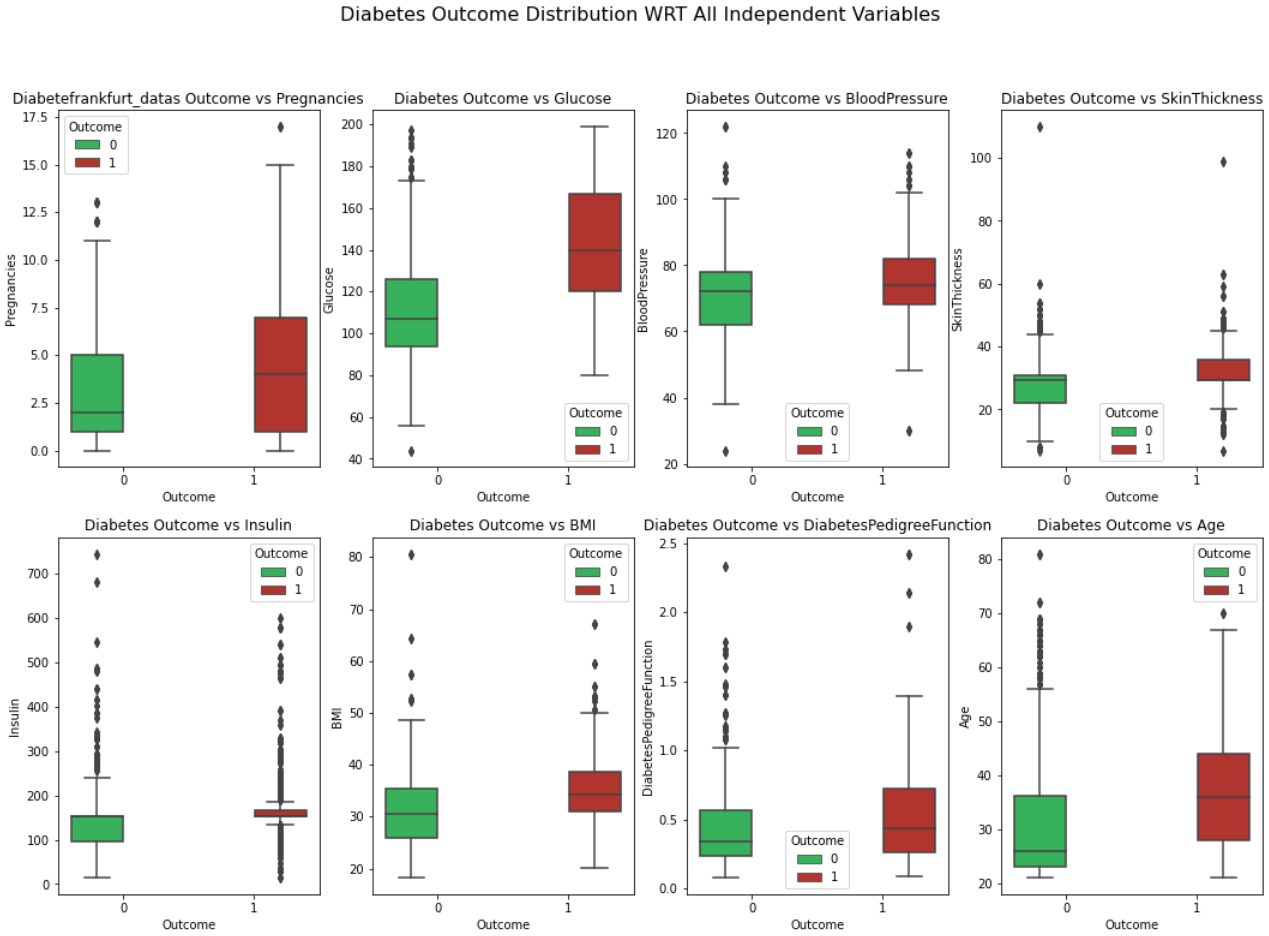


In [178…

```
import seaborn as sns

fig, axes = plt.subplots(2, 4, figsize=(18, 12))
fig.suptitle('Diabetes Outcome Distribution WRT All Independent Variables', font

sns.boxplot(ax=axes[0, 0], x=frankfurt_data['Outcome'], y=frankfurt_data['Pregna
axes[0, 0].set_title("Diabetefrankfurt_datas Outcome vs Pregnancies", fontsize=1

sns.boxplot(ax=axes[0, 1], x=frankfurt_data['Outcome'], y=frankfurt_data['Glucos
axes[0, 1].set_title("Diabetes Outcome vs Glucose", fontsize=12)

sns.boxplot(ax=axes[0, 2], x=frankfurt_data['Outcome'], y=frankfurt_data['BloodP
axes[0, 2].set_title("Diabetes Outcome vs BloodPressure", fontsize=12)

sns.boxplot(ax=axes[0, 3], x=frankfurt_data['Outcome'], y=frankfurt_data['SkinTh
axes[0, 3].set_title("Diabetes Outcome vs SkinThickness", fontsize=12)

sns.boxplot(ax=axes[1, 0], x=frankfurt_data['Outcome'], y=frankfurt_data['Insuli
axes[1, 0].set_title("Diabetes Outcome vs Insulin", fontsize=12)

sns.boxplot(ax=axes[1, 1], x=frankfurt_data['Outcome'], y=frankfurt_data['BMI'],
axes[1, 1].set_title("Diabetes Outcome vs BMI", fontsize=12)

sns.boxplot(ax=axes[1, 2], x=frankfurt_data['Outcome'], y=frankfurt_data['Diabet
axes[1, 2].set_title("Diabetes Outcome vs DiabetesPedigreeFunction", fontsize=12

sns.boxplot(ax=axes[1, 3], x=frankfurt_data['Outcome'], y=frankfurt_data['Age'],
axes[1, 3].set_title("Diabetes Outcome vs Age", fontsize=12)
```

Out[178…

```
Text(0.5, 1.0, 'Diabetes Outcome vs Age')
```



Some other datasets:

- https://www4.stat.ncsu.edu/~boos/var.select/diabetes.html
- https://www.kaggle.com/uciml/pima-indians-diabetes-database